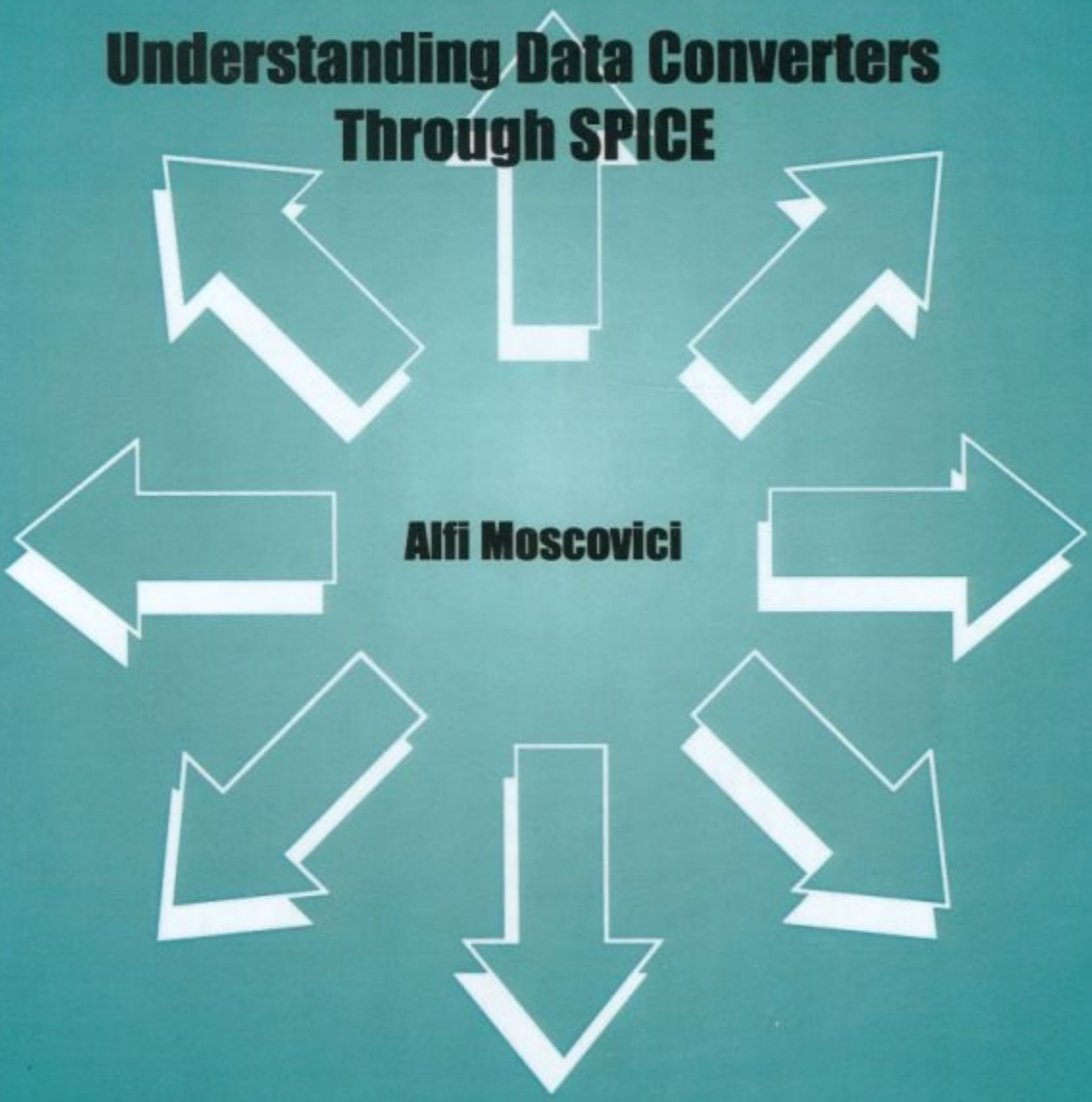
High Speed A/D Converters

Understanding Data Converters Through SPICE



Alfi Moscovici

HIGH SPEED A/D CONVERTERS

Understanding Data Converters Through SPICE

THE KLUWER INTERNATIONAL SERIES IN ENGINEERING AND COMPUTER SCIENCE

ANALOG CIRCUITS AND SIGNAL PROCESSING

Consulting Editor. **Mohammed Ismail**. *Ohio State University*

Related Titles:

ANALOG TEST SIGNAL GENERATION USING PERIODIC ΣΔ-ENCODED DATA STREAMS
B. Dufort, G.W. Roberts
ISBN: 0-7923-7211-5

HIGH-ACCURACY CMOS SMART TEMPERATURE SENSORS
A. Bakker, J. Huijsing
ISBN: 0-7923-7217-4

DESIGN, SIMULATION AND APPLICATIONS OF INDUCTORS AND TRANSFORMERS FOR Si RF Ics
A.M. Niknejad, R.G. Meyer
ISBN: 0-7923-7986-1

DESIGN AND IMPLEMENTATION
B.E. Jonsson
ISBN: 0-7923-7871-7

RESEARCH PERSPECTIVES ON DYNAMIC TRANSLINEAR AND LOG-DOMAIN CIRCUITS
W.A. Serdijn, J. Mulder
ISBN: 0-7923-7811-3

CMOS DATA CONVERTERS FOR COMMUNICATIONS
M. Gustavsson, J. Wikner, N. Tan
ISBN: 0-7923-7780-X

DESIGN AND ANALYSIS OF INTEGRATOR-BASED LOG -DOMAIN FILTER CIRCUITS
G.W. Roberts, V. W. Leung
ISBN: 0-7923-8699-X

VISION CHIP
A. Moini
ISBN: 0-7923-8664-7

COMPACT LOW-VOLTAGE AND HIGH-SPEED CMOS, BiCMOS AND BIPOLAR OPERATIONAL AMPLIFIERS
K-J. de Langen, J. Huijsing
ISBN: 0-7923-8623-X

CONTINUOUS-TIME DELTA-SIGMA MODULATORS FOR HIGH-SPEED A/D CONVERTERS: Theory, Practice and Fundamental Performance Limits
J.A. Cherry, W. M. Snelgrove
ISBN: 0-7923-8625-6

LEARNING ON SILICON: Adaptive VLSI Neural Systems
G. Cauwenberghs, M.A. Bayoumi
ISBN: 0-7923-8555-1

ANALOG LAYOUT GENERATION FOR PERFORMANCE AND MANUFACTURABILITY
K. Larnpaert, G. Gielen, W. Sansen
ISBN: 0-7923-8479-2

CMOS CURRENT AMPLIFIERS
G. Palmisano, G. Palumbo, S. Pennisi
ISBN: 0-7923-8469-5

HIGH SPEED A/D CONVERTERS

Understanding Data Converters Through SPICE

Alfi Moscovici

KLUWER ACADEMIC PUBLISHERS
NEW YORK, BOSTON, DORDRECHT, LONDON, MOSCOW

eBook ISBN: 0-306-46994-4
Print ISBN: 0-7923-7276-X

Created in the United States of America

Visit Kluwer Online at: http://kluweronline.com
and Kluwer's eBookstore at: http://ebooks.kluweronline.com

In loving memory of my mother Marieta

Contents

Acknowledgements

If it seems unduly clear to you, you must have misunderstood what I said.
Alan Greenspan

I would like to thank many of the colleagues that served as catalysts for this work. I am particularly thankful to Dr. R. Schmidt and Dr. G.S. Ostrem for their constructive comments in reviewing the manuscript, to Mr. S. Hisano, Mr. P. Kalthoff and Mr. B. Runco for their stimulating conversations during my stint at S.P.T. Messrs. S. Michaels, S. Sacks, W. Grandner, C. diFazio and Dr. A. Grebene were some of my best mentors. Those friends and numerous others with whom I spent many hours working on A/Ds have made the world of data converters more fascinating to me along the years.

Above all to my wife Michal my daughters Miri and Dana and my dad Yacov go special thanks for being a constant source of encouragement during the many hours that went into preparing this book. The only regret I have is that this work took valuable time away from them.

Finally I would like to thank professor Mohammad Ismail for reviewing the manuscript and Mr. Carl Harris at Kluwer Academic Publishers for taking a chance on this first-time author and to Ms. Deborah Doherty for taking the time to help in formatting the book.

Alfi Moscovici, Colorado Springs, October 2000

Preface

The Analog to Digital Converters represent one half of the link between the world we live in - analog - and the digital world of computers, which can handle the computations required in digital signal processing.
These devices are mathematically very complex due to their nonlinear behavior and thus fairly difficult to analyze without the use of simulation tools. Fortunately the availability of home computers, mathematical and circuit simulation programs can make this task easier. This book attempts to present the subject from the practicing engineer rather then the academic point of view so a practical approach is provided to the topic.

This book is intended to shed some light on the intricacies of various topologies of A/D converters. It is also intended as a learning tool by providing building blocks that can be stacked on top of each other to build higher order systems. The book provides a guide to understanding the various topologies used in A/D converters by suggesting simple models for the blocks used in an A/D converter. The converters discussed through the book constitute a class of devices called undersampled or Nyquist converters. The subject of oversampled A/D converters which are capable of achieving high degree of resolution with small amount of hardware based on a sigma-delta **(Σ−Δ)** modulator is not discussed however. These **Σ−Δ** converters sample the analog input repeatedly and quantize the signal one bit at a time. Since the quantization process of the **Σ−Δ** A/Ds is limited by the nature of their algorithm these converter are mostly used in low speeds applications.

Another reason for this book stems from the user' need to understand the key limitations in converter accuracy. The majority of converters used today are integrated circuits (I.C.) where the user does not have access to internal nodes. Some of the high accuracy converters require laser trim in order to attain higher matching of the internal components. The trim process is a mechanism used to alter a resistor value so it can match another resistor performing a similar function elsewhere in the IC. The trim process damages the resistor physical structure and in time, the matching obtained initially is reduced. The aging process is therefore the reason that A/D performance degrades over time.

Most manufacturers do not supply their customers with extensive models of their A/D converters. Thus the system designer is left without the capability to synthesize his system while performing a "what if" scenario. This invariably causes the system to be over-specified and contributes to the cost escalation of the product. Usually the A/D converters are treated as "black boxes", but a better appreciation of what makes the converters "tick", what makes them operate properly and the effects of parasitic loading on the various nodes can give the reader a better understanding of how to use them. To this end the book is making extensive use of the ubiquitous SPICE program which was developed in the early 1970 as an extension of CANCER and SPICE1 (Simulation Program with Integrated Circuit Emphasis) at Berkeley (ref. 69). In suggesting the use of SPICE the goal is to allow the reader to experiment and learn through a process of trial and error thus minimizing the frustration of system loading in the troubleshooting process. The use of SPICE is a powerful complement to the breadboarding phase in solving problems in almost any circuit. SPICE can help with the test of circuit operation theory, behavioral modeling and component functionality under a variety of temperatures, component tolerances and power supply conditions. The key to the proper use of the SPICE simulator as a tool is to enhance the speed an accuracy of the simulation.

In using SPICE, the analysis of large circuits can prove to be a difficult task even for the computers available today. *Sometimes SPICE aborts the simulation and proclaims the infamous nonconvergence error message* (ref.69). The key ingredient of the SPICE analysis is to be able to observe relevant details of the design in a sensible amount of time. This text will suggest some of the techniques necessary to reduce the analysis time in the face of increased complexity. The main contributor to the simulation length is the number of circuit nodes. For this reason, it is very important to develop a simple behavioral model that represents the circuit response as a

function of its linear as well as nonlinear behavior. The general scope of the macro-model we develop is to have the minimum number of nodes in the circuit, such that the simulation time is minimized.

Several macro-models will be presented along the way illustrating various A/D topologies with the associated simulations making the reader familiar with each topology pitfalls under different transient conditions. The reader will become accustomed with the specific behavior of each of the topologies employed and will be able to recognize the "signature" of the various topologies. The book supplies not only the models for the A/D converters for the SPICE program but describes the physical reasons for the converter's performance.

The tools used in deriving the results presented are:

- TopSpice™ by Penzar – a mixed mode SPICE simulator – version 5.90. The files included in the appendix A were written for this tool. However, most circuit files need only minor adjustments to be used on other SPICE simulators such as PSpice, Hspice, IS_Spice and Micro-Cap IV.
- Mathcad2000 - Professional by Mathsoft. This tool is very useful in performing FFT analysis as well as drawing some of the graphs. Again the mathcad files are included to help the user analyze the data.

The book supplies not only the models for the A/D converters for the SPICE program but describes the physical reasons for the converter's performance.

Chapter 1 examines the terminology used to differentiate between various characteristics of A/D converters. These specifications are examined in both the time and the frequency domain. In this chapter we set the foundation required for the user to understand the A/D manufacturer' data sheet. Since each application focuses on a different aspect we explain the relation between those specifications.

Chapter 2 concentrates on the most basic block used in the topology of any A/D converter. This is accomplished by establishing a model that describes the comparator's behavior for both small and large signal inputs. As a result of the comparator behavior in the time domain under large signal regime distortions are usually observed at its output. These distortions are examined through various examples.

In Chapter 3 we introduce the fastest and easiest converter to understand – the flash A/D. After the concept is reviewed through several examples the its limitations are analyzed in the time and frequency domain. Some of the performance limitations in practical I.C. implemented flash A/D are also illustrated in this chapter.

Possibly the most important block in systems that require fast A/D converters is the track and hold or sample and hold amplifier. This device is the subject of Chapter 4. As in previous chapters we establish a simple model for this component We also examine the matching effects of multiple track and hold channels on the dynamic behavior of time-interleaved A/D converter's performance.

The earliest and very likely the most efficient A/D converter is based on the binary search algorithm. This topology known as the successive approximation A/D is reviewed in Chapter 5. Here the theory of operation of the successive approximation A/D is reviewed followed by examples, which explore the speed/accuracy tradeoffs.

In Chapter 6 we introduce the reader to the concept of folding A/D converters and their dynamic behavior. This A/D converter is a natural evolution from the flash A/D. Several interpolation techniques are presented along with tradeoffs affecting the converter' accuracy.

Chapter 7 - examines the concept of subranging A/D in serial pipeline converters. These converters use several blocks that were already described in the previous chapters. In this topologies however there are several subtle aspects that need to be understood as the various components interact with each other. These relations are analyzed with the help of several examples.

A special case of serial pipeline architecture is explained in Chapter 8. This is the so-called 1.5 bit/stage A/D converter. The algorithm is based on a correction method in which each stage detects three quantization levels and reserves one level for error correction. The chapter illustrates various reasons for signatures observed in this topology.

Chapter 1

A/D Terminology

1. INTRODUCTION

To set a baseline for comparing A/D converter performance we need to define the major static and dynamic parameters. Along the way we will also outline some of the methods used in characterizing these specifications. Before defining the terms it is important to note that some converters have a defined ***minimum sampling speed*** which means that the conversion command is supplied at that minimal rate regardless of the analog input signal rate of change. The main reason for the minimum sampling rate is the droop rate of the track and hold amplifier preceding the A/D. This limitation will be farther explained in chapter 4.

As the name implies, the static specifications are parameters that are measured under DC input conditions (DC input voltage).

In contrast with the static characteristics, the dynamic parameters are measured with analog input signals that vary in time. To make things unambiguous, throughout the book we specify the sampling rate in SPS (samples/second) while the analog input frequency is expressed in Hz.

1.1 The static parameters

The error of an A/D converter is the difference between the theoretical and the actual input voltage required to produce a particular output code. In most applications the user can calibrate the offset and gain errors by subtracting the offset and dividing by the gain thus normalizing this

deviation. The normalized error is called the relative error (as opposed to the absolute error, which is the actual **measured** difference).

Following is a list of terms that define converter static errors.

1.1.1 *Offset Error*

This is the difference between the theoretical and actual input voltage required to obtain the transition from code 0000..00 to code 0000..01.
It should be noted that some manufactures define the offset at the mid-scale of the converter' range (when the input is symmetric around 0V). The assumption is that the converter' transfer curve is described by a straight: $V_{OUT} = V_{IN} \cdot G + V_{OS}$ where G is the slope of the line and V_{OS} is the offset.

1.1.2 *Gain Error*

This parameter indicates the slope difference between the lines connecting the theoretical and actual transitions of the full scale extremes - transition 0000..00 to 0000..01 and transition 1111..10 to 1111..11. Theoretically the line should span from 0.5 LSB above zero where the first transition occurs and 1.5 LSB below the full scale (remember that the last code 1111..11 occurs 1 LSB below full scale and the last code transition is 0.5 LSB below that point). The ratio of the span between the first and the last actual codes and the ideal difference is the gain error (usually expressed as a percentage of full scale).

1.1.3 *Integral Linearity Error (ILE)*

This is the worst case deviation of codes from a straight line connecting the ends of the full scale (from -FS to +FS).
Figure 1 – 1 is an example of the transfer curve for a 3-bit A/D converter. The dotted line represents the theoretical transfer curve of the converter. The heavier line represents the actual input voltage required to obtain the output code transitions shown on the Y-axis. Since we are considering a 3-bit converter, we expect eight output levels $(=2^3)$ – from 000 to 111. Counting from one end of the full scale to the other we observe seven transition points $(=2^3-1)$ – or for the general case 2^N-1. If we assume in our example that the full-scale range is 2V, the corresponding LSB weight is 0.25V(=2V/8)

Next, we illustrate the end-point normalization process for the ILE calculation as it applies to our example. The data is presented in table 1 – 1. This end-point normalization process defines the equation that connects the full-scale extremes as follows:

$$\frac{V_{NORM} - V_{IN}}{V_{IN_{MAX}} - V_{IN_{MIN}}} = \frac{V_{Theor} - V_{Theor_{MIN}}}{V_{Theor_{MAX}} - V_{Theor_{MIN}}} \qquad 1.1$$

where V_{IN} is the measured input voltage, V_{NORM} is the normalized voltage, and V_{Theor} is the theoretical voltage required for that transition.

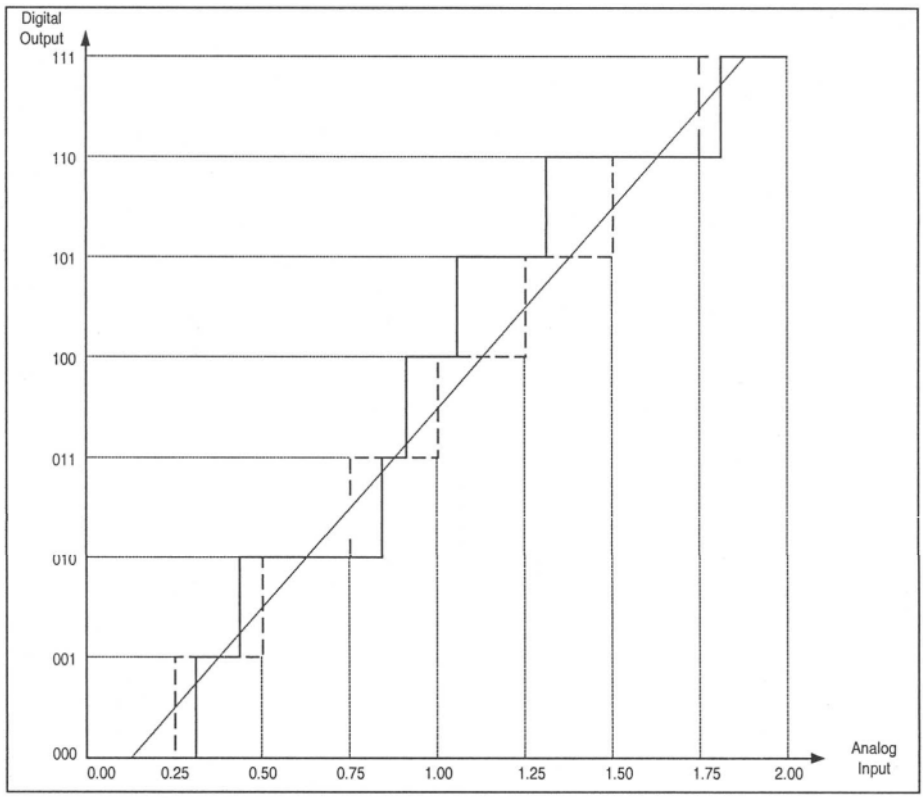


Figure 1-1. Transfer curve for a typical A/D - Digital out vs. analog in

This equation defines a gain – G – and an offset – OS – for the transfer curve. Substituting and extracting the equation for the normalized voltage V_{NORM} we obtain:

$$G = \frac{V_{IN_{MAX}} - V_{IN_{MIN}}}{V_{Theor_{MAX}} - V_{Theor_{MIN}}} \qquad \textit{and} \qquad OS = V_{IN_{MIN}} - G \cdot V_{Theor_{MIN}} \qquad 1.2$$

The normalization process just defined is called end-point normalization since it uses the end points of the full scale for the calculation. Note that by definition the error at the ends is always zero.

The normalized linearity of the device is also shown in table 1-1:

Table 1-1. Example of normalization process

Output Transition Code	Input Voltage equivalent to the Output Transition Code	Theoretical IL	Normalized IL	ILE (normalized IL - theoretical IL)
000 to 001	0.35	0.25	0.250000	+0.000000
001 to 010	0.45	0.50	0.353448	-0.146552
010 to 011	0.85	0.75	0.767241	+0.017241
011 to 100	0.95	1.00	0.870690	-0.129310
100 to 101	1.10	1.25	1.025862	-0.224138
101 to 110	1.35	1.50	1.284483	-0.215517
110 to 111	1.80	1.75	1.750000	+0.000000

Based on the previous definition and equation 1.2 we calculate the gain error to be:

$$G = \frac{1.8 - 0.35}{1.75 - 0.25} = 0.9667 \qquad 1.3$$

resulting in a gain error of approximately 3.3%. The offset – OS – is calculated to be 0.10833 V or an offset error of 0.433 LSBs. The maximum difference from the end point line occurs at the transition 100 to 101 with a deviation of 0.224138 V (=0.896LSBs).

Some manufacturers specify the ILE as a line of the form $OS+G\cdot V_{IN}$ that best approximates the input-output transfer curve in terms of **best square fit**. The idea is to select a line, which minimizes the square of the error for each of the M data points:

$$S = \sum_{i=0}^{M} \left[V_{Theor_i} - G_{BSF} \cdot V_{IN_i} - OS_{BSF} \right]^2 \qquad 1.4$$

where G_{BSF} and OS_{BSF} are the slope and intersection of the linear regression such that the deviation from the line – S - is minimized. In this case the end points of the full scale do not have zero error. Applying this definition to our example and using the theoretical points as reference we calculate the best square fit: G_{BSF}=0.914286 and OS_{BSF}=0.064286. The maximum ILE for the

best square fit occurs at the last code (110 to 111) and is –0.135714V or -0.543 LSBs. The last results can be calculated using any mathematical program that has routines for linear regression.

Obviously the two normalization techniques just defined are significantly different from each other. Using the end point linearity definition the ILE is almost +0.9 LSB while the ILE for the best-fit definition is of 0.54 LSB. This illustrates that the lack of standards in converter specifications may mislead the user in perceiving wide variations in performance due to "specmanship".

Lastly, since the ILE is an indication of the converter transfer characteristics under dynamic conditions it is evident that the best square fit is a better indicator of the converter's harmonic distortion (see section - How does the linearity affect the converter dynamic behavior? - below)

1.1.4 *Differential Linearity Error (DLE)*

This is the actual difference between two adjacent codes minus 1 LSB. In an ideal A/D adjacent code transitions are 1LSB apart resulting in a DLE of 0 LSBs (DLE = 1LSB – 1LSB = 0LSB). A converter is called *monotonic* when it exhibits an increasing output code for an increasing input voltage. When the DLE is non-negative the monotonicity of the converter is guaranteed.

The simplest method used to check the static performance of an A/D is shown in figure 1 – 2.

The test is performed as follows: a linear input voltage with an amplitude equivalent to a fraction of the full-scale range (10-20 LSBs) is supplied to the device under test (DUT). This linear ramp must have the means of being shifted from one end of the full-scale to the other. By changing the ramp's offset, all code transitions can be observed. Using this test method, we notice that the output signal is not a straight line from –FS to +FS but rather a collection of segments shifted in amplitude - see figure 1 – 3. This is equivalent to a modulo operation - the pattern of eight steps repeats and it represents the residue of the overall full scale to the length of the segment presented. As shown in figure 1 – 3 the last three LSBs are utilized to restore the digital output into an analog voltage – constituting a D/A converter.

The D/A performs its conversion by transforming the digital voltage of the bit into a binary weighted current. The three currents are summed into R_T obtaining the D/A output voltage.

The reconstructing D/A amplitude accuracy is not important in this case since the error of interest is shown along the X-axis of the display. Thus, if the resistors are not related in an exactly binary fashion the output shows unequal step heights in the reconstruction on the Y-axis.

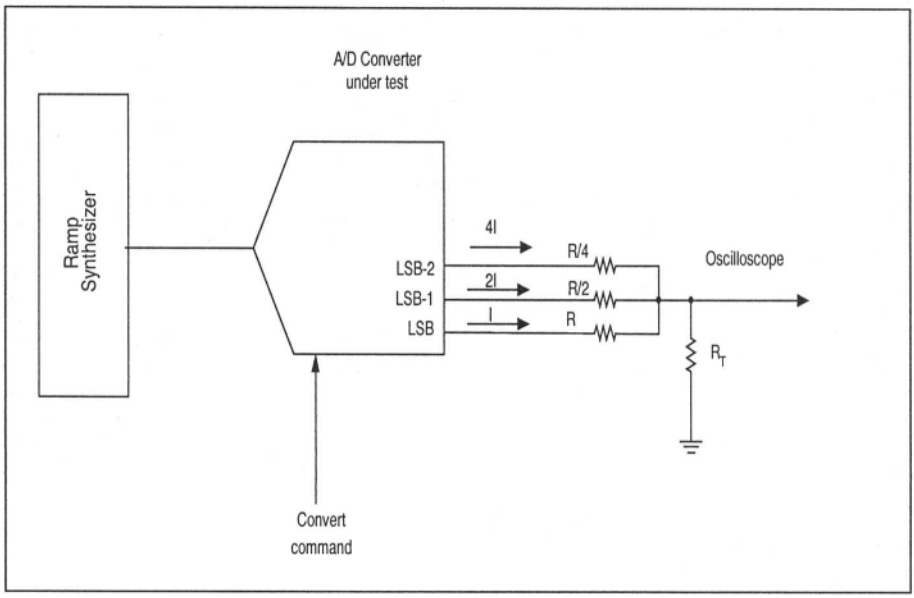


Figure 1-2. Static test for an A/D converter

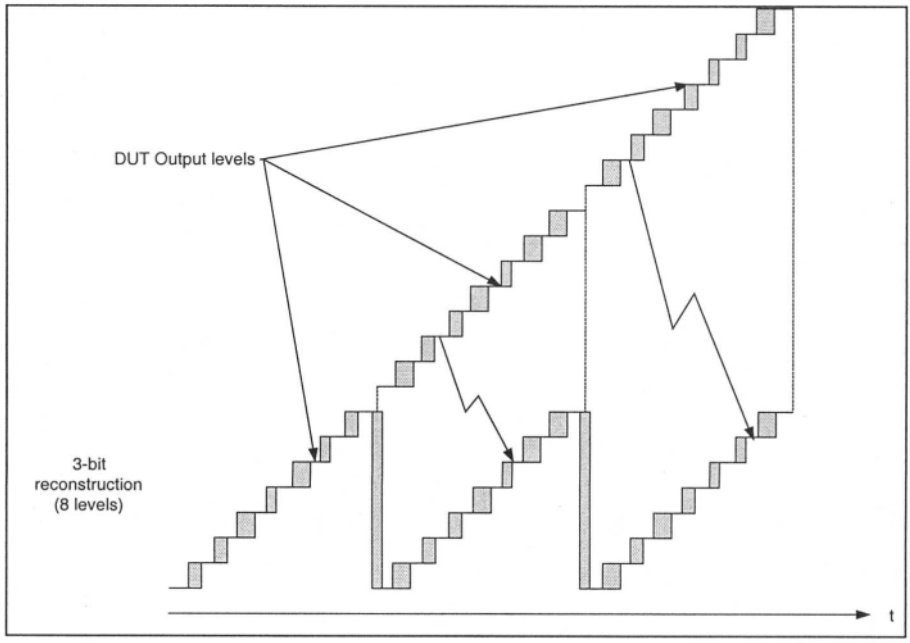


Figure 1-3. A 3-bit reconstruction of the digital output of an A/D

In a typical converter, not all the codes are equal in width. This width variation reflects the linearity errors. ***The linearity errors of an A/D converter are therefore shown along the X-axis.***

Figure 1 – 3 also shows that the code transitions as displayed on an oscilloscope are not very crisp and well defined. These uncertainties (gray areas) on the X – axis represent a dynamic error called jitter (see also dynamic parameters below).

1.2 The dynamic parameters

The dynamic errors of an A/D converter are the result of its behavior under input transient conditions. These errors reflect insufficient bandwidth, slew-rate limitation or settling time of the analog signal path. They are an inevitable consequence of the design tradeoffs between speed, resolution and power conservation. Bandwidth limitation is a dynamic deficiency, which does not cause harmonic distortion. The harmonic distortion is caused by nonlinear phenomena of the circuit such as slew rate limitation or other nonlinear components in the circuit (such as capacitance voltage coefficients, dielectric absorption, etc.).

1.2.1 *Total Harmonic Distortion (THD)*

This is the ratio of the square root of the sum of the squares of the first most significant harmonics (usually from second to the fifth) to the fundamental output signal (V_1). This parameter is generally expressed in dB.

$$THD = 20 \cdot \log\left(\sqrt{\frac{V_2^2 + V_3^2 + V_4^2 + V_5^2}{V^2_1}} \right) \qquad 1.5$$

1.2.2 *Signal-to-Noise Ratio (SNR)*

This is the ratio of the remaining harmonics (not accounted for in the THD) to the fundamental. This parameter is usually expressed in dB.

$$SNR = 20 \cdot \log\left(\sqrt{\frac{V_6^2 + V_7^2 + V_8^2 + \cdots + V_N^2}{V^2_1}} \right) \qquad 1.6$$

1.2.3 *Signal-to-Noise and Distortion Ratio (SINAD or TDE for Total Dynamic Error)*

TDE is the ratio of all harmonics to the fundamental. This parameter is usually expressed in dB.

$$TDE = 20 \cdot \log\left(\sqrt{\frac{V_2^{\,2} + V_3^{\,2} + V_4^{\,2} + \cdots + V_N^{\,2}}{V_1^{\,2}}}\right) \qquad 1.7$$

Since the numerator is usually smaller than the denominator, this is a negative number.
SINAD reflects the ratio of signal to noise and distortion. Since the signal is in the numerator and the noise and distortion are in the denominator this represents the reverse ratio relative to TDE. Therefore the SINAD and TDE are equal in magnitude, and have opposite sign.

The relation between TDE and THD and SNR is given by:

$$TDE = 10 \cdot \log\left(10^{\frac{SNR}{10}} + 10^{\frac{THD}{10}}\right) \qquad 1.8$$

1.2.4 *Spurious-Free Dynamic Range*

This is the difference (in dB) between the RMS input signal and the highest frequency spur at the output of the A/D. Figure 1 - 4 shows an example of how SFDR is measured in an FFT test. In our example the SFDR is approximately 50 dB.

Note: In all equations above the assumption is that the fundamental frequency resides in the first bin of the spectrum. The FFT program in figure 1-11 is set to assure that this is indeed the case.

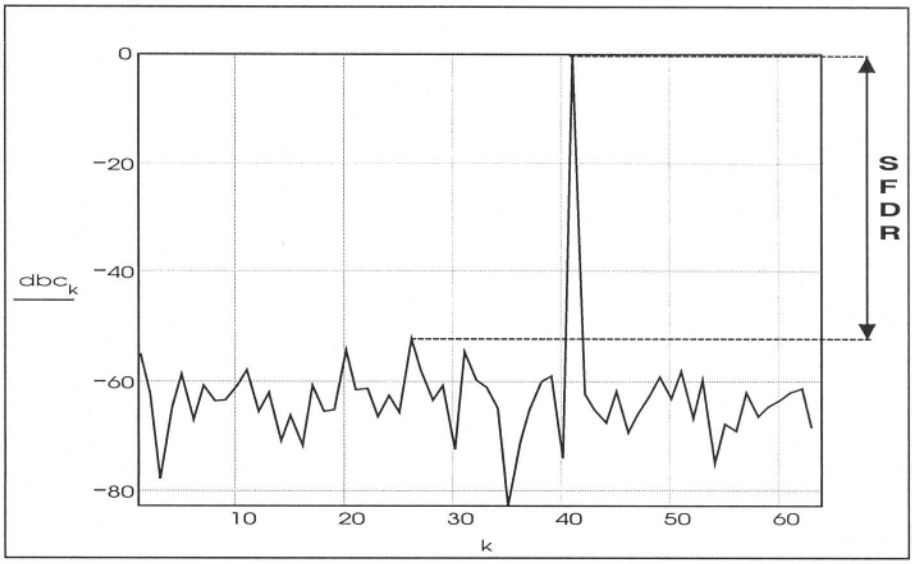


Figure 1-4. Spurious Free Dynamic Range (SFDR) definition

1.2.5 *Aperture uncertainty*

Due to noise, the A/D converter response to the conversion command does not occur at a known time instance. Aperture uncertainty or jitter is defined as the short-term, non-cumulative variation of the significant instants of the sampling signal from their ideal position in time. The error manifests itself as an edge variation of the sampling signal relative to the analog input. A graphical illustration of this phenomenon is shown in figure 1 – 5. Since most high-speed converters employ a track and hold amplifier in front of the A/D, figure 1 – 5 refers to the two regions of operation as the track and hold regions. The sampling instance in figure 1-5 occurs on the falling edge of the track/hold command when the analog voltage is being held prior to being processed by the A/D converter. The highest probability of the sampling edge is shown as the darker region.

There are two reasons for this uncertainty – amplitude noise causing threshold of the sampling device to fluctuate and phase instability of the sampling clock.

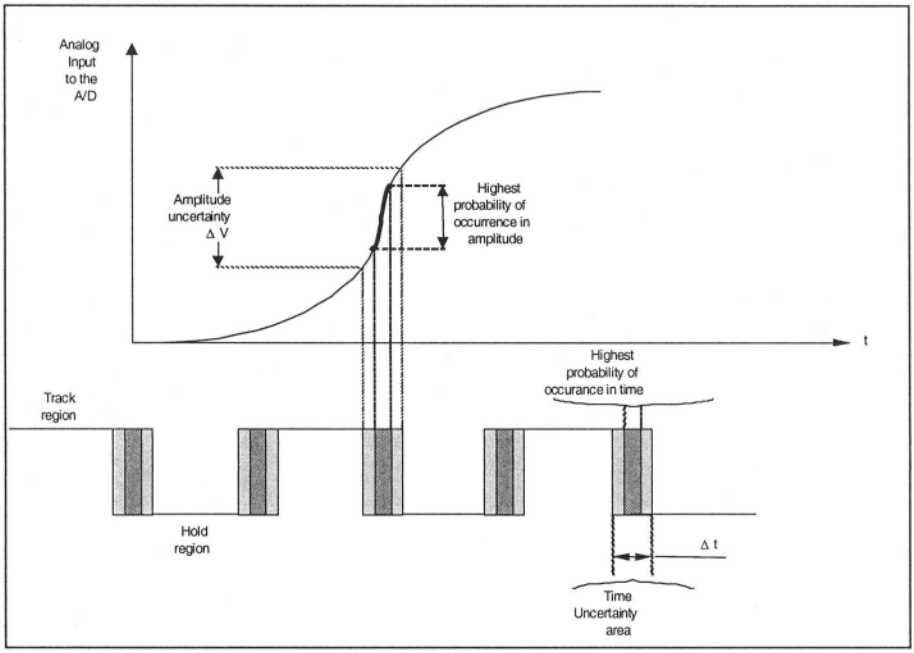


Figure 1-5. Jitter caused timing error

When dealing with high input frequencies the jitter causes noise-like distortions to the sampled signal.

Since aperture uncertainty is a random, noise-like phenomenon, it contributes to a reduction in SNR. References 33 and 37 analyze the contribution of aperture uncertainty to the reduction of SNR.

The following equation predicts the relation between aperture uncertainty and SNR:

$$SNR = 20 \cdot \log\left(\frac{1}{2 \cdot \pi \cdot f \cdot t_a}\right) \qquad 1.9$$

In the equation t_a represents the RMS aperture jitter and f is the input frequency. A graph showing SNR degradation due to aperture uncertainty is shown in Figure 1- 6.

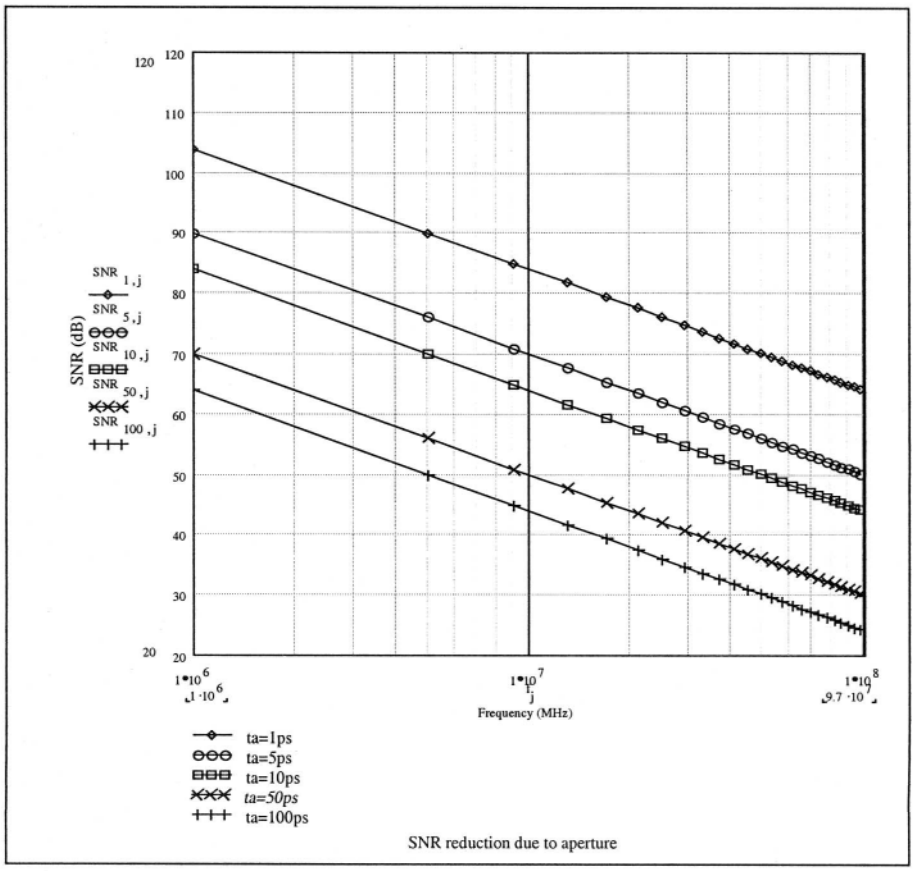


Figure 1-6. SNR reduction due to aperture jitter

The graph shows for example that an A/D with an aperture uncertainty of 50 psec RMS error, tested at an input frequency of 10 MHz will be limited to an SNR of approximately 50 dB. Thus if we test a 10 bit A/D that is expected to have SNR of 62 dB due to quantization error, it is important to limit the jitter to less than 10 psec RMS error:

$$SNR = 20 \cdot \log\left(\frac{1}{2 \cdot \pi \cdot 10 \cdot 10^{6} \cdot 10 \cdot 10^{-12}}\right) = 64.036 \text{ dB} \qquad 1.10$$

1.2.6 *Frequency aliasing*

This is a phenomenon that is a direct artifact of the sampling process. The Nyquist sampling theorem requires that a continuous bandwidth-limited input signal with frequency limited to f_c, is sampled at a rate $f_s > 2f_c$. If this

condition is not observed, the Fourier transform of the sampled signal gets distorted. This occurs because frequency components exceeding the Nyquist frequency are folded back into the input band. This phenomenon makes the reconstruction of the input signal impossible. To mitigate aliasing, it is important to choose the relationship of input to sampling frequency carefully (29,39).
A graphical illustration of the aliasing phenomenon is shown in figure 1 – 7.

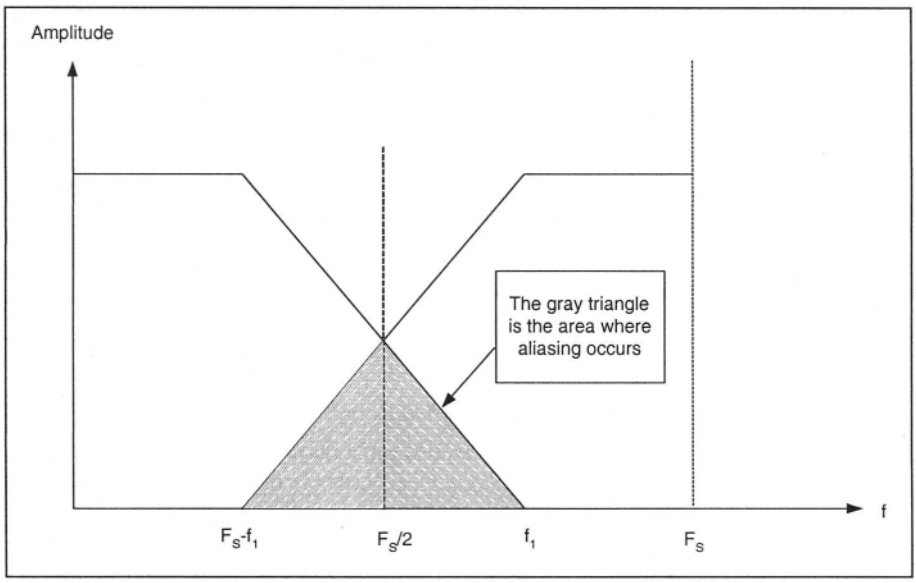


Figure 1-7. Frequencies above Fs/2 are aliased due to the sampling process

Given the sampling frequency of the converter (Fs) and the analog input frequency **(f_{in})** the aliased frequency **(f_a)** can be found as follows:

i. calculate f_{in} / F_s . Name the residue of the division **($R= mod(f_{in}, F_s)$).**
ii. if **$R > Fs /2$** then the aliased frequency f_a will be located at **$Fs - R$** otherwise
iii. f_a will be located at R

For example if the converter sampling frequency is 1GSPS and the input frequency is 450 MHz, the fundamental frequency of the reconstructed output is found at 450 MHz. The second harmonic is expected to be at 900 MHz. Since this frequency is higher than the Nyquist frequency (=500 MHz) and the calculated R=900 MHz it means that the second harmonic will be

aliased. The corresponding frequency for the second harmonic will be found at: 1GHz – 900MHz = 100 MHz.
The third harmonic (=3 · 450 = 1350 MHz) will be found at 350 MHz (R=350 MHz) and therefore the tone will be found at 350 MHz.

1.2.7 How does the linearity affect the converter dynamic behavior?

The sheer operation of the A/D transforming a smooth input ramp into discrete levels through the quantization process creates harmonics. This is known as quantization noise. For a perfectly linear A/D, the quantization noise limits the SNR performance level to:

$$SNR = 6.02 \cdot N + 1.76 \qquad dB \qquad 1.11$$

That is, if the input signal extends to the full-scale range of the converter the noise caused by the quantization process results in 6.02·N +1.76 dB below the fundamental. In the equation, N represents the number of bits. For example in a 10-bit device, the noise floor is 61.96 or approximately 62 dB bellow the RMS value of the input signal. (9,11,12,14,17,19).

Excluding dynamic limitations of the converter such as slew-rate limitations, the Integral Linearity Error (IL or INL) is the major cause of harmonic distortion (THD). The reason is that when a sinusoid is passed through a nonlinear transfer function it gets distorted creating harmonic tones at the output. Usually the INL of an A/D converter is fairly smooth resulting therefore in low order harmonics. The differential nonlinearity of the A/D on the other hand causes high frequency harmonics that are accounted for in the SNR factor. The reason is that unlike the smooth integral nonlinearity, the DLE represents abrupt adjacent transition in the transfer function in the time domain. This is equivalent to a high frequency harmonic content in the frequency domain and thus, results in high order harmonics. Since high harmonics are accounted for in the SNR ratio, DNL cause a reduction in signal-to-noise ratio.

The parameter that measures all nonlinearities of the A/D in aggregate is the TDE or SINAD (signal to noise and distortion) since it encompasses both THD and SNR. Some manufacturers specify an additional parameter called ENOB (effective number of bits). This figure measures the equivalent number of bits in an A/D as a function of input frequency. In effect this is the same equation as the one predicting the noise floor where the number of bits of the converter – N is replaced by ENOB:

$$ENOB = \frac{TDE - 1.76}{6.02} \qquad 1.12$$

The difference between the equations is in the fact that ENOB is always specified as a function of frequency. For example a device with **TDE=62 dB** has approximately 10 effective bits at the specified frequency.

1.2.8 Converter resolution effects on Spurious Free Dynamic Range

The SFDR is a critical parameter especially in communication systems. An analysis of its dependency on the number of bits of the converter has been investigated in references 33, 37 and 53 and it is shown to be: $SFDR \cong 9 \cdot N$. The highest harmonic is proven to occur at harmonic $\pi \cdot 2^N$ for a converter limited by quantization noise.
In other words the largest harmonic of an N bit A/D is lower than the fundamental by 9N dB (assuming a sinusoidal full-scale input signal). This equation predicts that each increase in the number of bits results in a 9-dB reduction in harmonics. An intuitive explanation for this effect is as follows: going from N to **N+1** bits reduces the quantization noise by a factor of 2 (6 dB). In addition, since the quantization frequency is doubled for each additional bit, the frequency of the error is increased. Mathematically, Parseval's formula tells us that the power is conserved in either the time or frequency domain and therefore the doubling of the frequency results in a 3-dB reduction in the harmonic magnitude. Ordinarily converters are not perfectly linear and thus SFDR is limited by lower order harmonics than predicted by the equations above.

Let us summarize the A/D converter errors and their characteristics.

The **static parameters**: offset, gain, INL, DNL are the easiest to test. These parameters are measured under DC input conditions and therefore are constant for a given converter. Figure 1 – 2 illustrates a way of evaluating these DC characteristics and additional methods are described in the references 9,15,18,32.

The **dynamic test methods** for evaluating SNR, THD, SFDR, SINAD (TDE) are discussed at length in references 11, 12, 13, 14, 15, 17, 18, 19, 27, 28, 29, 30, 32, 37 and 38. The dynamic tests however are more elaborate and require significantly more care to accomplish than the static measurements.

The dynamic performance of the A/D analyzed in the following chapters will underline the importance of certain device parameter on the dynamic performance of the entire A/D.

The two test methods available for **dynamic A/D evaluation** can be categorized as follows:

- the beat and envelope tests which are coherent tests and
- the noncoherent windowed tests

A common setup used for A/D coherent dynamic test is shown in figure 1-8.

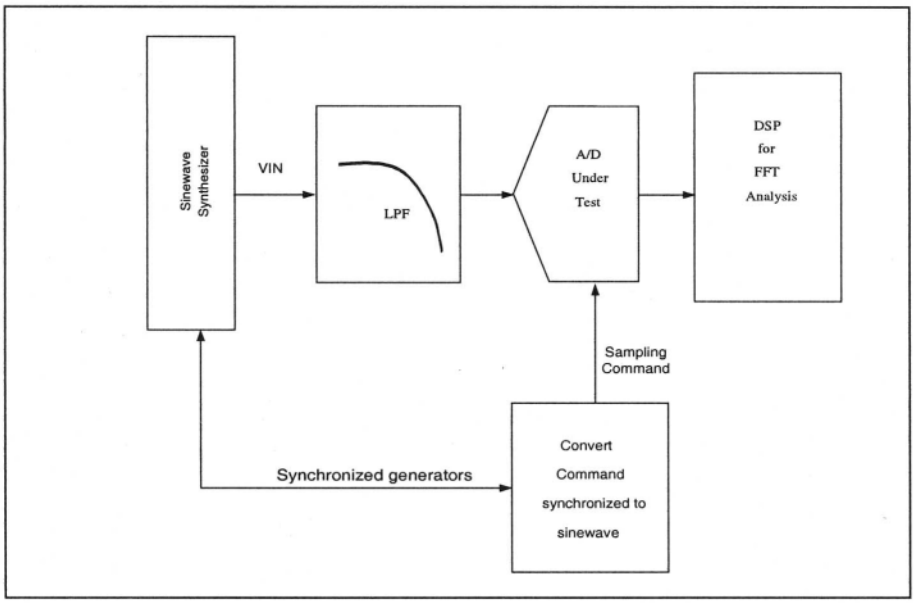


Figure 1-8. Dynamic A/D test set-up

In the beat test method the sinewave-input signal is offset in frequency from the sample frequency. The beat frequency is selected such that on successive cycles of the sampling signal, the output "walks" through the input signal. When the reconstructed output signal is analyzed, the beat frequency is observed. A graphical example of beat frequency test is shown in figure 1-9:

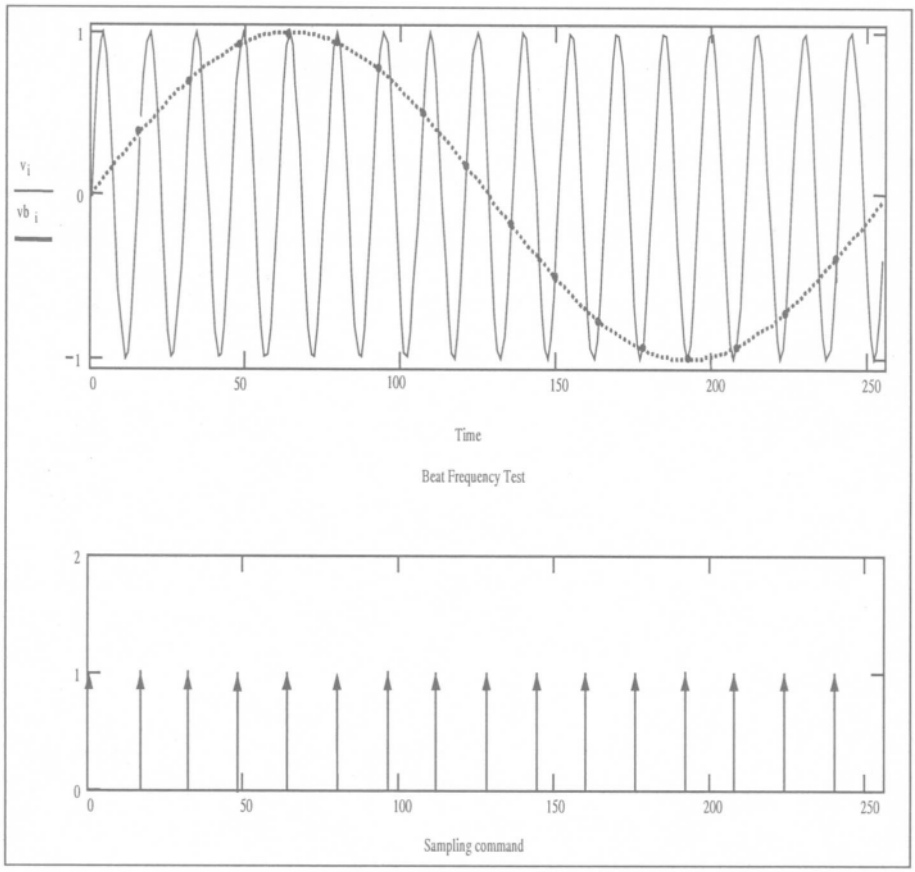


Figure 1-9. Dynamic "beat frequency" test

The test can be performed to observe differential or integral linearity as a function of the converter's input frequency. In this case the samples are taken such that adjacent samples are only one LSB apart from each other. The frequency required for this test is (31):

$$\Delta f = \frac{f_s}{2^N \cdot \pi} \qquad 1.13$$

where $\mathbf{f_s}$ is the sampling frequency, $\mathbf{\Delta f}$ is the beat frequency and N is the converter's number of bits. For example if the device under test is a 10 bit converter with a sample rate of 10 MSPS, then the beat frequency required for 1 LSB change on successive samples is calculated to be 3.108 kHz from the equation above. Since each adjacent sampled point is expected to be

1LSB away from the previous sample the ILE and DLE can be tested as a function of input frequency.

This test is limited to multiples of the beat frequency but requires no special care regarding frequency spillage, which will be explained below.

A variation of the beat test method is the envelope test. This method is more demanding on the converter in the fact that adjacent samples of the input are taken on opposite ends of the full scale. Thus the input signal is tested for example at the positive full scale, followed immediately by a sample at the negative full scale. By making the sample frequency slightly offset from the Nyquist rate this method ensures that the samples are taken on alternate half cycles of the input signal. In performing this test, the A/D converter can be tested during slew-rate conditions and pushed to the settling limit of various internal blocks of the converter.

The envelope and beat test methods are called **coherent test** methods. This is a result of the very tight dependence between the sampling frequency and the analog input frequency. Unless these two signals are completely synchronized during the test (as illustrated in figure 1 – 8), they introduce noise similar to the aperture uncertainty of the converter itself. In fact the jitter between the input and sampling signal will RSS (square root of the sum of the squares) with the aperture uncertainty of the converter (assuming no correlation between the two).

In contrast to the beat and envelope methods, that guarantee no frequency spillage effects, the noncoherent test methods require very careful choice of filters prior to the FFT calculations.

To understand this problem let us consider how the sheer action of sampling of a continuous time signal affects the Fourier transform.

When the Fourier transform of the continuous time signal is known, the Fourier transform of the discrete samples can be obtained by the following operations (39):

1. transformation of the frequency axis according to the relation $\Theta=\omega T$
2. multiplication of the amplitude axis by a factor 1/T
3. summation of an infinite number of replicas of the original spectrum, shifted horizontally by integer multiples of the sampling frequency ω_{sample}.

The result of the summation gives a periodicity in Θ with the period of 2π. Therefore the sampling in the time domain results in periodicity in the frequency domain. The filter's function is to reduce the error caused by the sampling process. The main focus in Fourier analysis is the determination of the Fourier transform $F(\omega)$ of a signal f(t) in term of the segment:

$$f_\tau = f(t) \cdot p_\tau(t)$$

In effect this represents the time window $p_\tau(t)$ during which we look at the signal (42). By definition, we observe the A/D output during a finite time interval. A rectangular window in the time-domain is such that it has a value of 1 inside the window and 0 outside the observation time. In the frequency domain, the rectangular window is the familiar sin(x)/x of a rectangular pulse. To express mathematically the limited observation time we multiply a rectangular pulse equal to the observation time (τ) with the time varying digital output code. The multiplication between the finite observation time and the output of the A/D in the time domain corresponds to a convolution operation in the frequency domain.

If the A/D output does not contain an integer number of cycles during the observed time-window the result will be a **leakage error** in the frequency domain**. The leakage error comes about because the A/D data is not harmonically related to the window length.** Adjusting the window to include an integer number of cycles can eliminate this error. The form that the leakage takes depends on the shape of the window. By changing the window shape in the time domain and reducing the discontinuities at its edges we can significantly reduce the leakage.

Several windows are classically used for the non-coherent test (39). Their main characteristics are outlined below:

- Rectangular filter (with side lobes of –13.5 dB) (39, 43)
- Bartlett window (with side lobes of –27 dB) (39)
- Hann window (with side lobes of –32 dB) (39, 43)
- Hamming window (with side lobes of –43 dB) (39,43)
- Blackman window (with side lobes of –57 dB) (39)
- Kaiser filters (with side lobes -30 to -90 dB depending on the α factor) (39)
- Dolph filters (with side lobes -40 to -80 dB depending on the α factor) (39)

The rectangular window is the standard acquisition window with a weight of 1 inside the window and zero outside the window. The Hamming window is a combination of cosine added to a pedestal and so on. In all above filters the intent is to minimize the width of the main lobe and reduce the size of the side lobes with respect to the main lobe in the frequency domain. In general, the greater the window's bandwidth, the less resolution it provides. On the other hand, as the side lobes decrease, the filter selectivity increases - the ability to distinguish adjacent frequency components. A discussion of these filters is beyond the scope of this book.

The rectangular window while not the best in the frequency domain due to limited attenuation of its side lobes (sin(x) / x), is the easiest to visualize. There is no weighing function that is given to the various time-data and therefore all points have the same weight. This is different than the other filters mentioned above where each time-data-point is given a different weight through the multiplication operation $\mathbf{f(t)\cdot p_\tau(t)}$ to accomplish the windowing function (39 and 43).

1.2.9 Rectangular window – an example

Let us illustrate how the width of rectangular filter can affect the results of the FFT test in a perfect A/D. In our example, we consider a perfect A/D with infinite resolution. The output of the converter is analyzed in a FFT using a 4K-point time record. The time record is then subjected to truncation of the tail data points and padded with zero (this is the operation of adding zeroes to the tail of the sequence). In other words the time record is shorten from 4K points one point at a time and the original data is padded with zeroes. This is equivalent to the shortening of the observation time $\mathbf{p_\tau(t)}$. Plotting the dynamic performance – TDE – as a function of the number of truncated tune records results in the graph outlined in figure 1 - 10.

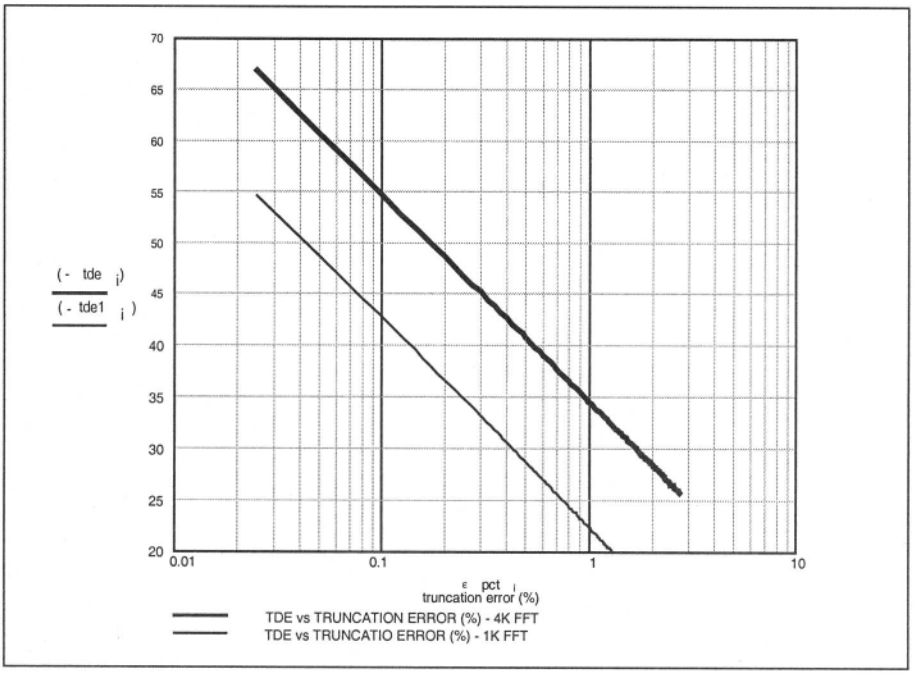


Figure 1-10. The truncation error caused by the rectangular window affects the Signal-to-Noise-and-Distortion (SINAD)

The same graph outlines as well the truncation error for a 1K FFT.
Several things are evident in the graph:

1. The slope of the graph is **–20** dB/decade. For a 4K FFT with a device with infinite resolution and with 0.1% zero padding the **SINAD = - 54.7** dB. This means that if four of the time data points are set to zero (0.1% of 4096 points) the performance is reduced to less than 9 effective bits!
2. For a 1K point FFT with the same infinite resolution A/D, the SINAD is 12 dB lower than the 4K FFT. Again with 0.1% truncation and zero padding limits the SINAD to approximately 42.6 dB (or less than 7 effective bit performance).

This example demonstrates the importance of careful choice in the number of data points for a noncoherent test of A/D and the result of such test when the data is truncated and zero padded. In general when a user is interested in improving the noise floor observability in any A/D converter the time domain record needs to be increased. The noise floor for such a converter is improved to:

$$NoiseFloor = 6.02 \cdot N + 1.76 + 10 \cdot \log\left(\frac{n}{2}\right) \qquad 1.15$$

where n is the number of time points and N is the converter resolution. In other words if a 12 bit converter is used and the expected quantization noise is ≈ **74** dB than with 1024 time points for the FFT we can observe a noise floor of 101 dB **(=6.02· 12 + 1.76 + 10·log[1024/2] = 101.09 dB)**. Doubling the number of time points to 2048 further improves the noise floor by an additional 3 dB.

1.3 The FFT analysis

In the following chapters, all the A/D considered are analyzed using an envelope test to achieve the required coherent test of the device. The resulting time data is subjected to an FFT test using a MATHCAD program. The FFT file is provided below in figure 1 - 11).

The analysis proceeds as follows: the SPICE produced output file is transferred into a file called "FILENAME.DAT". The data file is scanned by the MATHCAD program and examined for the FFT length (expected to be half as long as the number of time-data points). This also establishes the frequency number of bins (frequency resolution). The vector representing the time domain samples are expected in the first column of the data file **FILENAME.DAT**$^{\langle 1 \rangle}$ (remember that Mathcad indexes from "0" not "1") so the first column of your data file is vector $^{\langle 0 \rangle}$ or in Mathcad terminology $\mathbf{d}^{\langle 0 \rangle}$ while the time data points are expected in $\mathbf{d}^{\langle 1 \rangle}$. By searching for the highest

amplitude the fundamental bin is found. The other bins are then amplitude-normalized to the fundamental bin and the results are expressed in dB. Next the reshuffling of the frequency bins occurs (to account for aliasing) and the harmonics bin locations are found and the TDE is calculated. If the number of time data points is small then the frequency resolution is reduced. However since TDE takes into consideration all frequency bins for the TDE calculation the result is accurate regardless of the frequency resolution. The results of the FFT are shown in this example at the end of the figure 1 - 10 as follows:

- the fundamental (harmonic number 1) is as expected at 0 dB and is found in frequency bin number 3.
- 2^{nd} harmonic is at **–32.55** dB relative to the fundamental and is found in frequency bin 6 and so on...
- finally the SINAD and ENOB are calculated to be 21.55 dB and 3.28 bits respectively.

fft.mcd

$td := READPRN("fft812p5.dat")^{(1)}$

$np := length(td)$ $j := 0..np - 1$ number of time data points

$nb := \frac{np}{2}$ $k := 1..nb$ nb is the number of frequency bins

$fd := fft(td)$ frequency domain

$mfd_k := |fd_k|$ magnitude of frequency bin

$big := max(mfd)$ fundamental

$mb_k := if(mfd_k < big, 0, k)$ find the bin of the fundamental

$fb := max(mb)$

$nfd_k := if\left(mfd_k > 10^{-9}, \frac{mfd_k}{big}, 10^{-9}\right)$ normalize amplitude to fundamental and ignore small (-180 db) points for log. operation

$dbc_k := 20 \times \log(nfd_k)$ $bot := min(dbc)$ normalized amplitude

$l := 1..nb$ $m_l := mod(l \times fb, np)$ reshuffle frequency after aliassing

$hb_l := if(m_l > nb, np - m_l, m_l)$

$tde := 10 \times \log\left[\sum_k (nfd_k)^2 - 1\right]$ calculate SINAD (tde)

$l := 1..if(nb > 6, 6, nb)$ $m_l := mod(l \times fb, np)$

Harmonic bin number $l =$	Mag.(dBc) $dbc_{hb_l} =$	Freq. number $hb_l =$
1	0	3
2	-32.55	6
3	-23.51	7
4	-31.25	4
5	-32.57	1
6	-34.64	2

$hb_l := if(m_l > nb, np - m_l, m_l)$

$tde := 10 \times \log\left[\sum_k (nfd_k)^2 - 1\right]$

$tde = -21.55$ total dynamic errror (dBc)

$ENOB := \frac{-tde - 1.8}{6.02}$ $ENOB = 3.28$

Figure 1-11. Mathcad file used for FFT calculations

PROBLEMS

1. With reference to table 1-1: recalculate the 3-bit converter offset, gain and linearity (both end-point and best square fit) if the input voltage at the transition 100 to 101 is:
   i. 0.95V
   ii. 0.88V (note that in this case the converter is non-monotonic).

2. An A/D converter is tested for dynamic performance with a sampling frequency **Fs = 500** MSPS. The analog input frequency is 200 MHz. Find the location of the first 5 harmonics. Is the choice of frequency acceptable for testing the first 5 harmonics? How can the test be improved?
3. A converter has a **THD= 71** dB and **SNR = 73**dB. Find the SINAD.

4. A 14 bit converter is sampled at 20 MSPS. A beat frequency test is required for testing the device for no missing codes:
   i. what is the required input frequency?
   ii. what is the expected SNR and SFDR?
   iii. what is the maximum generator jitter allowed in measuring the device SNR ?

5. An A/D converter is tested at **Fs= 1** GSPS and **fin=437.5** MHz. The harmonic tones are found at the following frequencies and each with the following magnitude:
   i. 125 MHz -62 dB
   ii. 312.5 MHz -75 dB
   iii. 250 MHz -68 dB
   iv. 187.5 MHz -73 dB
   v. 375 MHz -71 dB
   vi. 62.5 MHz -69 dB and
   vii. 500 MHz -66 dB

   Find the location of each harmonic. Find SFDR and SINAD.

Chapter 2

The Comparator

1. INTRODUCTION

The comparator is the most basic component used in the architecture of an A/D converter. In fact, it represents itself the simplest form of an A/D converter. This device compares an analog input to a reference voltage and produces a digital output of zero or one depending on the position of the input relative to the reference. If the reference is connected to the negative terminal of the comparator and the analog input is connected to its positive terminal then when the signal is higher than the reference its digital output (OUT) is high and its complementary output (OUTB) is low. This means that the comparator is an A/D converter with one bit of resolution. We should emphasize that the two terms - resolution and accuracy - should not be confused. **Resolution** represents the number of distinct digital states of a converter while **accuracy** is defined by the precision with which these states are defined. The comparator for example has a resolution of one bit – it produces only two output states. The **accuracy** of the comparator for discriminating between the two states depends on attributes such as gain, slew-rate limitation and input noise. For example, if the output is ECL compatible, the output swing is expected to be approximately 1 V (-0.8V to –1.8 V). With a gain of 60 dB (gain of 1000 V/V) an input of 1 mV is expected to result in the full ECL output swing required. If the gain is less than 60 dB, an undefined output state will result. This can cause the output of the comparator to possibly be in the wrong state or an undefined state.

Figure 2-1(a) illustrates the general topology of a simple comparator implemented in a bipolar technology and its associated symbol – figure 2-1 (b).

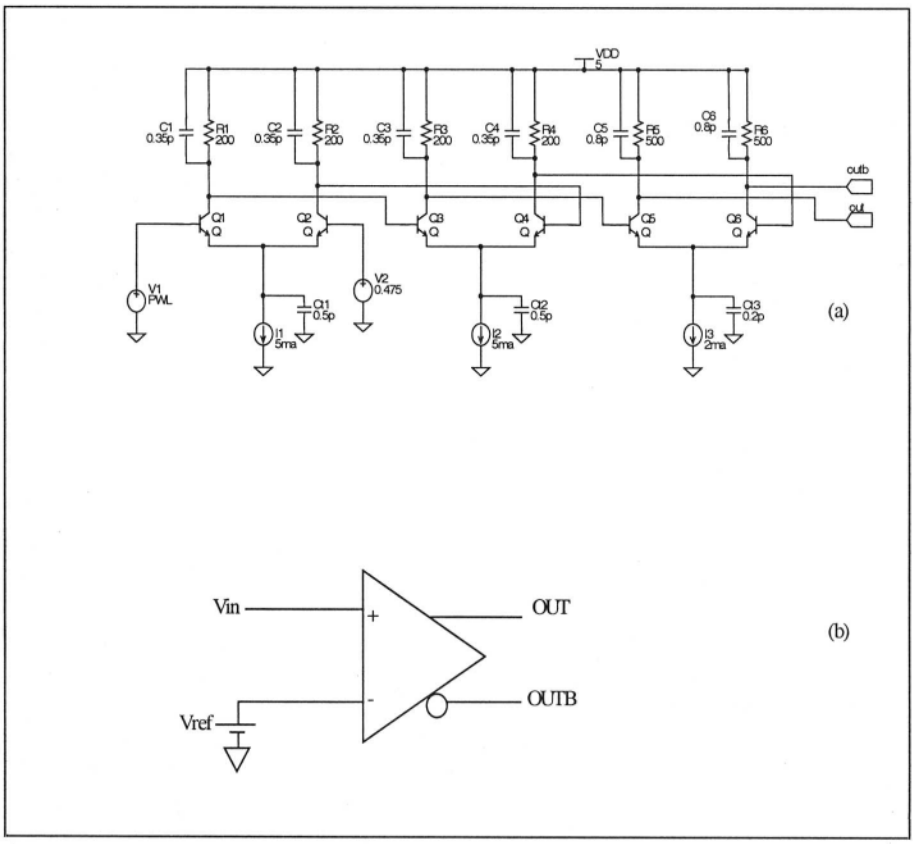


Figure 2-1. An analog comparator: (a) electrical schematic and (b) symbol

The comparator outlined in the figure uses three differential stages. Missing from figure 2 – 1 are the familiar emitter followers of the output stage and the latch circuitry.

The practical limitations of the comparator speed are illustrated by including in figure 2 – 1 the parasitic capacitance of the load in each of the stages. These parasitics are caused by the inherent capacitance of the load resistor to the substrate. In our example we assume a time constant of 70psec **(=0.35pF·200Ω)** in the front two stages and 400psec **(=0.8pF·500Ω)** for the output stage (due to higher load resistance). These time constants are reasonable approximations for the combined parasitics of the diffused resistors, as well as wiring parasitic capacitance to the substrate in a bipolar technology with $\mathbf{f_\tau \approx 15\ GHz}$.

An additional limitation, namely the slew rate at each of the tail current nodes is included with the addition of the capacitance C_{ct} for each stage. With a current of 5mA and a capacitance of 0.5 pF, the emitters' node is limited to a slew rate of:

$$SR = \left(\frac{dV}{dt}\right)_{max} = \frac{I}{C} = 10 V/n\,\text{sec} \qquad 2.1$$

when both transistors of the differential pair may shut off during a transient if driven from a high impedance node.

To examine the comparator' response under various conditions we simulate its transient behavior using the SPICE program and assuming transistor models with $f_\tau = 15$ GHz.
Considering that the transistors themselves have a time constant of only 11.4 psec it becomes obvious that the output followers do not contribute significant delays to the overall response time (the emitter followers are the only stage with no voltage gain).

In general the circuit topology of the comparator varies according to its usage in a specific application. For example, the comparator used in the design of a flash converter is typically very simple. The reason is that for an N bit flash converter the A/D requires 2^N-1 comparators. Thus a 6 bit flash A/D, uses 63 comparators. If the comparator employed utilizes a three stage design, then a total of 189 stages are used $(=63 \cdot 3)$. If each stage consists of 3 transistors (a differential pair and its associated current source), the total number of transistors used is 567 $(=189 \cdot 3)$ without accounting for the bias and other support circuitry. The number of components is therefore an exponential function of the number of bits. It follows then that the more comparators are used in an A/D the simpler their structure must be.

Reference 1, suggests a comparator model as a combination of an amplitude-limiting device followed by a bandwidth-limiting block. While relatively accurate, this analysis does not address the nonlinear behavior caused by input slew rate limitation.

The time-domain response of the comparator is limited by two main mechanisms:

- bandwidth limitation (linear region), and
- slew-rate limitation (nonlinear region).

This states basically that the time-domain limitation occurs **in the analog front end** of the comparator; the digital circuit behavior is usually treated as a simple delay (digital delays are purely additive in the time domain).

1.1 A Comparator Macro-model

Our aim in this chapter is to be able to predict the comparator response using a much simpler model than the actual transistor models. In the next paragraphs, we suggest a comparator model suitable for mathematical analysis and SPICE modeling.

Much work has been done in the industry to develop a simplified macro-model of this complex circuit (1,5,6,7). In developing the model it is important to obtain a method that can predict the comparator' behavior in both the time and the frequency domain.

In order to keep the macro-model simple, we limit its aim to predict only the differential signal behavior. In other words we assume an infinite rejection of the common-mode signals (which contribute only second order terms in the comparator response). The model we are about to consider will be used in subsequent chapters to understand various A/D topologies. The usage of the comparator in the A/D structure is mostly relevant as a differential structure and therefore the CMRR and PSRR aspects have only secondary importance.

Before considering the macro-model we remind the reader the equations describing the output current of a differential pair. For a bipolar differential pair the output current is:

$$I_{out} = I_{tail} \cdot \tanh\left(\frac{Vin}{2 \cdot V_T}\right) \qquad 2.2$$

where V_T represents the familiar thermal voltage (**=26 mV** at room temperature), and I_{tail} is the current source at the emitters connection.

A CMOS differential pair has an output current (see ref.54):

$$I_{out} = I_{SS} \cdot \sqrt{\frac{\beta \cdot Vin^2}{I_{SS}} - \frac{\beta^2 \cdot Vin^4}{4 \cdot I_{SS}^2}} \qquad 2.3$$

where β and I_{SS} are parameters dependent on the process.

Figure 2 – 2 illustrate this current behavior as a function of the input voltage. The graph for the bipolar pair is plotted assuming 50 mV of degeneration for the pair **(=tanh (Vin/0.1)).** The CMOS pair current is scaled with the help of ISS and **β** so the curves can be shown on the same scale and their comparable behavior can be observed.

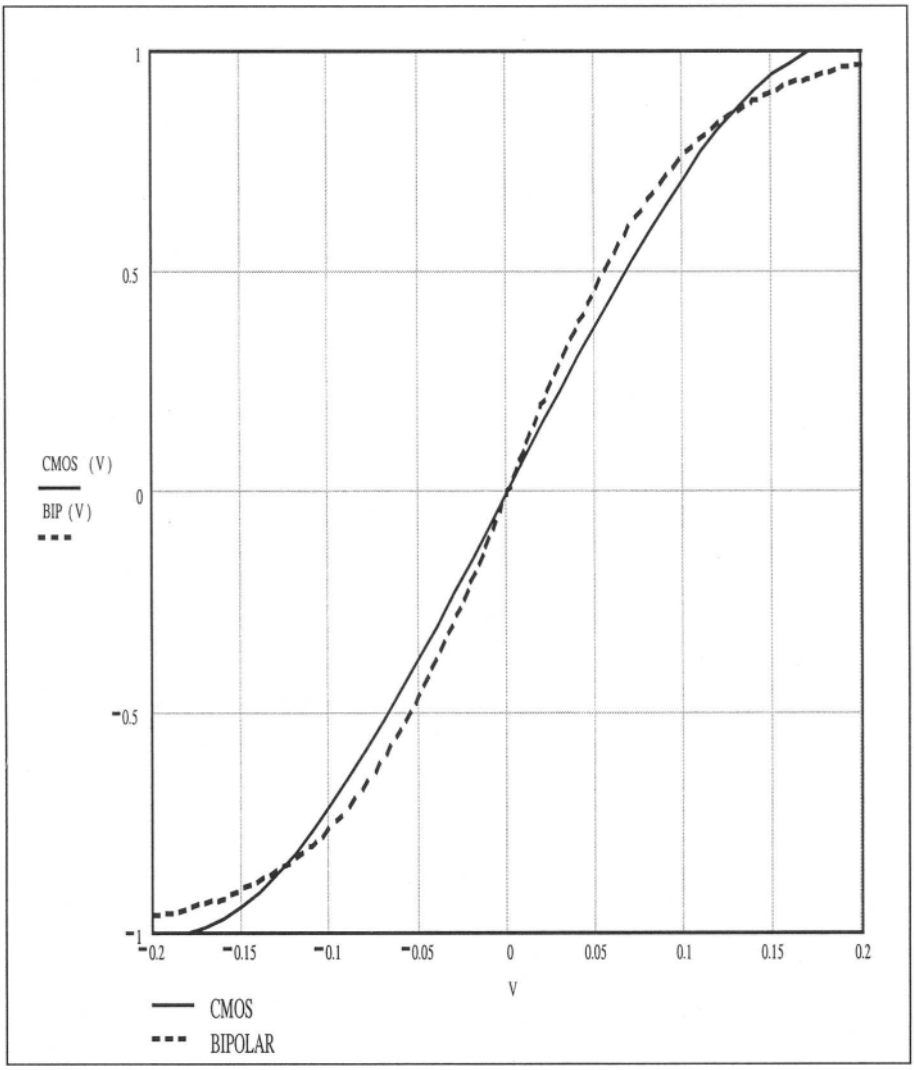


Figure 2-2. Differential pair transconductance curve: CMOS pair (solid line) and Bipolar pair (dashed line)

The first comparator macro-model we consider is intended to model **a bipolar comparator**. Given the base emitter diode equation of the bipolar transistor, the current of the differential front end, can be described by the hyperbolic tangent equation 2.2. The resulting Spice model we propose is based on this simple equation and is shown in the figure 2 - 3.

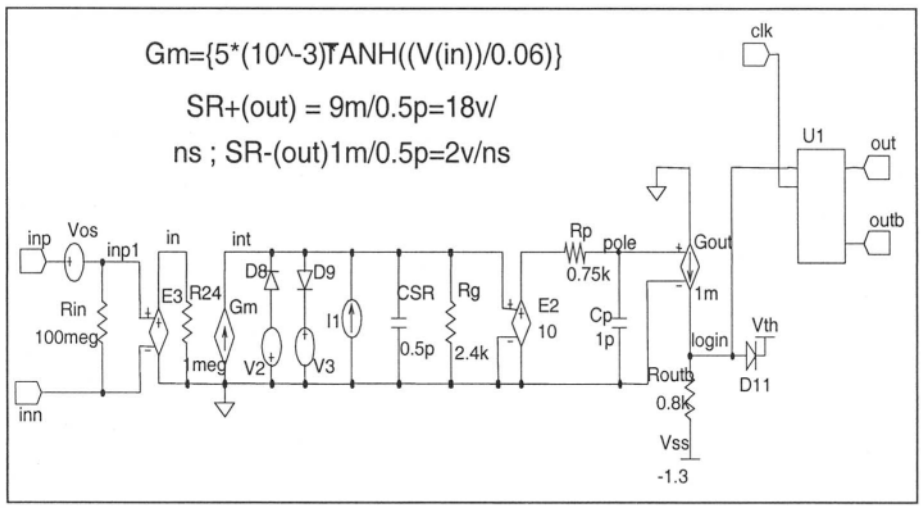


Figure 2-3. The suggested comparator - SPICE model

The reason for using this equation is twofold:

- this is a "well-behaved" equation in the sense of being smooth and having continuous derivatives
- Spice can use this functional description and portray the actual physical device behavior.

The model components are:

- The self-limiting function of a differential pair is obtained by incorporating the nonlinear Gm stage with the help of the hyperbolic tangent function. In the example, the current of the stage is governed by the equation:

$$Gm = 5 \cdot 10^{-3} \cdot \tanh\left(\frac{V_{IN}}{0.06}\right) \qquad 2.4$$

and therefore it is bound between **+5mA** and **–5mA** depending on the controlling input voltage. With the values used, the maximum Gm is

obtained at **Vin = 0** and the equivalent resistance is 60mV / 5mA (=12 ohms).

Having a limiting function at the input is certainly necessary but not sufficient in modeling the slew rate limitation of the stage. That is, if the input voltage causes a current step, the resistor-capacitor combination will allow the current step to settle with the time constant CSR · Rg.

- The diodes D8 and D9 enable the circuit to spend a well defined time in slew limited mode: the time needed for the current discharging the capacitor CSR to switch from one diode to the second. The diodes allow the current to switch almost instantaneously so that the maximum current and the capacitor CSR control the slew rate. In our case the maximum slew rate is:

$$SR = \left.\frac{dV}{dt}\right|_{max} = \frac{I}{C} = \frac{5mA}{0.5PF} = 10^{V}\!/_{n\,\mathrm{sec}} \qquad 2.5$$

- By adding voltage sources in series with each of the diodes (V_2 and V_3), the circuit can be kept in a slew-limited condition for various lengths of time.
- The total gain of the comparator is set by the product Rg · Gm · E2. In the case depicted here the gain is 2.4K · (5mA/60mV) ·10 = 2000 (=66 dB).

  The bandwidth of the comparator is set by two products:
  $\tau_d = R_p \cdot C_p$ (in this case 750 psec) and $t_s = R_g \cdot CSR$ (the second pole - in our example 1.2 nsec). The first pole is equal to the delay time given in the manufacturer data sheet for large input overdrive voltages.
- Finally the latch enable control is accomplished with the help of the clocked latch U1.

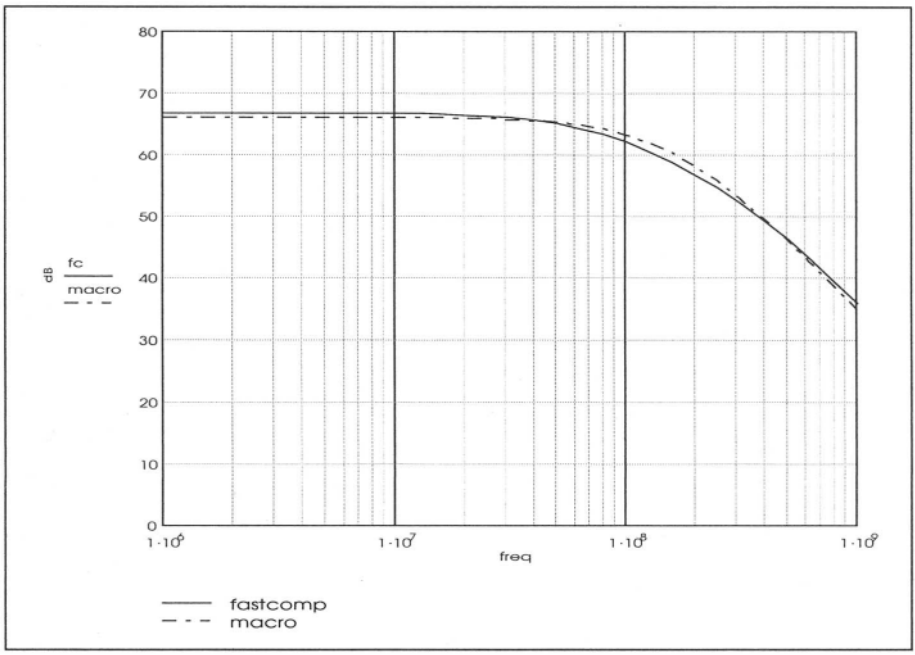


Figure 2-4. Small signal Open-loop gain of the comparators in figure 2-1 and 2-2

To evaluate the accuracy of our behavioral model, it is instructive to simulate in SPICE, the more elaborate comparator of figure 2-1 and compare the response against the model of figure 2-3.

Figure 2-4 illustrates the small signal open loop gain – Bode plot - of our macro-model in figure 2-3 and the circuit of figure 2-1(fastcomp).

Figure 2-4 confirms the close approximation of the macro-model in figure 2-3 to the actual circuit of figure 2-1.

1.2 Comparator delay as a function of overdrive input voltage

We use now the macro-model to analyze the propagation delay of the comparator as a function of input voltage overdrive. For these simulations, we keep the input Vinp constant, with a swing of 1V and input slew rate of 1V/nsec. The negative input node Vinn is changed to obtain the proper overdrive voltage. Figure 2-5 explains the overdrive definition in the time domain.

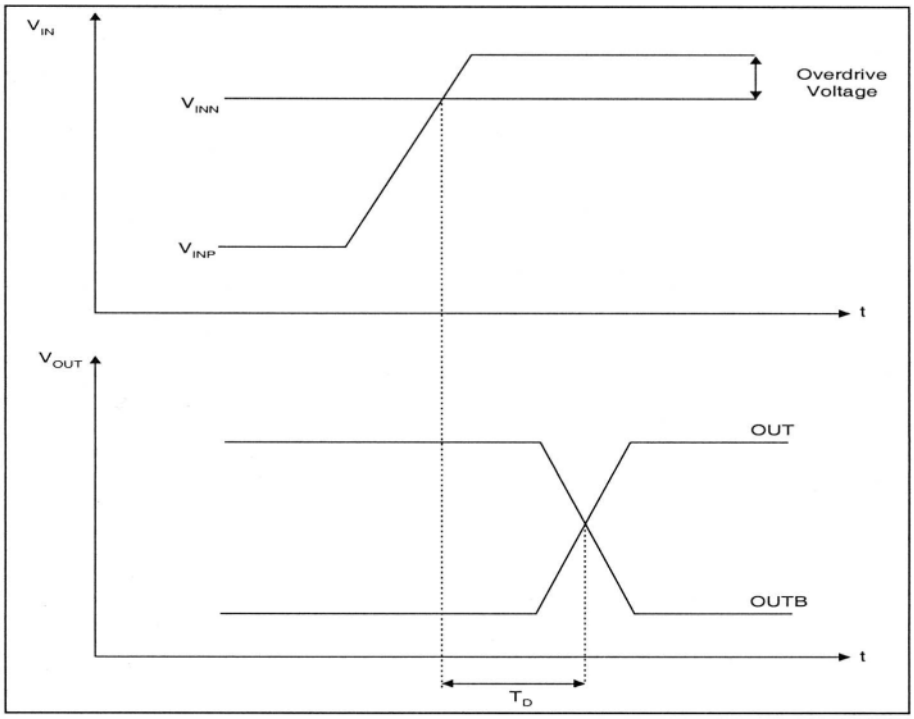


Figure 2-5. Definition of voltage overdrive: (a) Input overdrive and (b) Propagation delay - Td

For example with Vinp moving from **–0.5V** to **+0.5V** and with Vinn held at 0.45V the comparator has an input overdrive voltage of 50 mV.

Using the macro-model we can analyze the comparator performance mathematically. Assuming an input voltage step we can write the equations for the node **int** (figure 2 – 2). Let us assume that the input voltage is a step from **–0.5V** to **+0.5V** with the second input held at 0.45V - or an overdrive voltage of 50 mV. The initial condition is such that Gm has a current of – 5mA (5mA· tanh **(-0.95/0.06) = -5 mA**). This current causes the diode D8 to be "ON" clamping the node **int** to about **-0.7V.** As the analog input switches to a positive voltage, diode D8 turns "OFF" as soon as the capacitor CSR discharges to zero. At the same time the current in the loop reverses direction until D9 turns "ON". During the time that both diodes are "OFF", the circuit is governed by the equation:

$$V_{INT} \cdot \frac{1}{R_g} + CSR \cdot \frac{dV_{INT}}{dt} = Gm \cdot Vin \qquad 2.6$$

This describes the linear region of the circuit. Solving the equation, we obtain:

$$V_{INT}(t) = R_g \cdot 5mA \cdot \tanh\left(\frac{V_{in}(t)}{0.06}\right) + k \cdot e^{-\frac{t}{R_g \cdot CSR}} \qquad 2.7$$

where k is defined by the initial conditions. The time domain equation describing the transient at node **int** becomes:

$$V_{INT}(t) = R_g \cdot 5mA \cdot \tanh\left(\frac{V_{in}(t)}{0.06}\right) + 12.7 \cdot e^{-\frac{t}{R_g \cdot CSR}} \qquad 2.8$$

From this equation, we can find the time t_1 required to switch from one diode being "ON" until the second diode turns "ON". If the input step is instantaneous then the time t_1 is calculated to be:

$$t_1 = R_g \cdot CSR \cdot \ln\left(\frac{12.7}{11.3}\right) = 0.1168 \cdot R_g \cdot CSR \qquad 2.9$$

From node **int** to the latch, the signal is delayed by an amount of time equal to the time constant $R_P \cdot C_P = 750$ psec. This time is actually the majority of delay time while the time t_1 calculated in the equation above is a very short period of time while the current switches from one diode to the next (in our case 140 psec). This represents a small fraction of the overall delay time of the comparator and is the exact period of time that makes the comparator delay dependent on the amount of overdrive. The case just analyzed is a relatively simple case where $V_1(t)$ is a voltage step. To calculate the delay for a different function than a step, the equation becomes significantly less manageable. For this reason, from this point on we will use only the Spice macro-model only rather than an exact symbolic mathematical analysis. A more complex comparison of transient behavior can be obtained through SPICE simulations between circuit of figure 2 – 1 and our macro-model. This comparison establishes the time delay dependency on overdrive conditions.

For this test, lab measurements are performed on a more complex IC device available commercially (SPT9689 - a subnanosecond comparator manufactured in a process with $f_\tau \approx 12\text{-}15$ GHz). The results are summarized in figure 2 – 6 below.

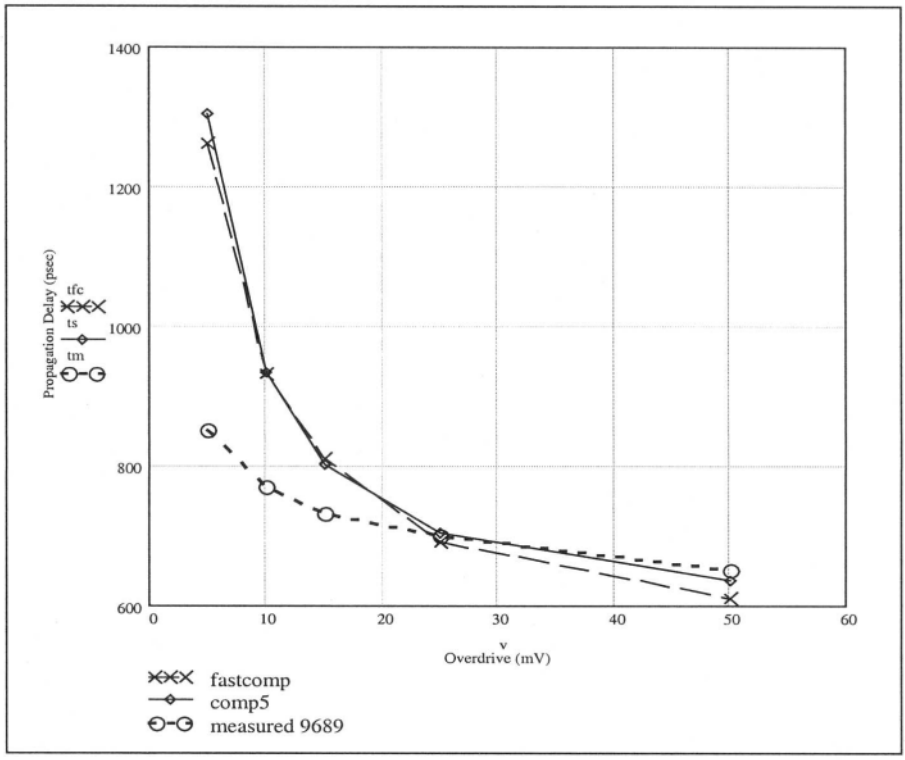


Figure 2-6. Propagation delay vs. input overdrive

In the graph, the horizontal scale is the amount of overdrive voltage in mV, and the vertical scale represents the propagation delay in picoseconds. The various curves are:

- t_{fc} is the propagation delay of the transistor circuit of figure 2 - 1
- t_s is the propagation delay of our macro-model in figure 2 - 3
- t_m the propagation delay measured for the commercial device (SPT9689).

The graph shows that the proposed model is in good agreement with the actual measured lab data for commercial comparator (especially for overdrive voltages of 10 mV or more).

1.3 Delay time versus Slew-Rate

Our next test case studies the comparator behavior under slew rate limitation. For this case we maintain a constant amount of overdrive voltage and vary the input voltage slew rate. The comparator' behavior as a function of slew-rate is presented in figure 2 - 7.

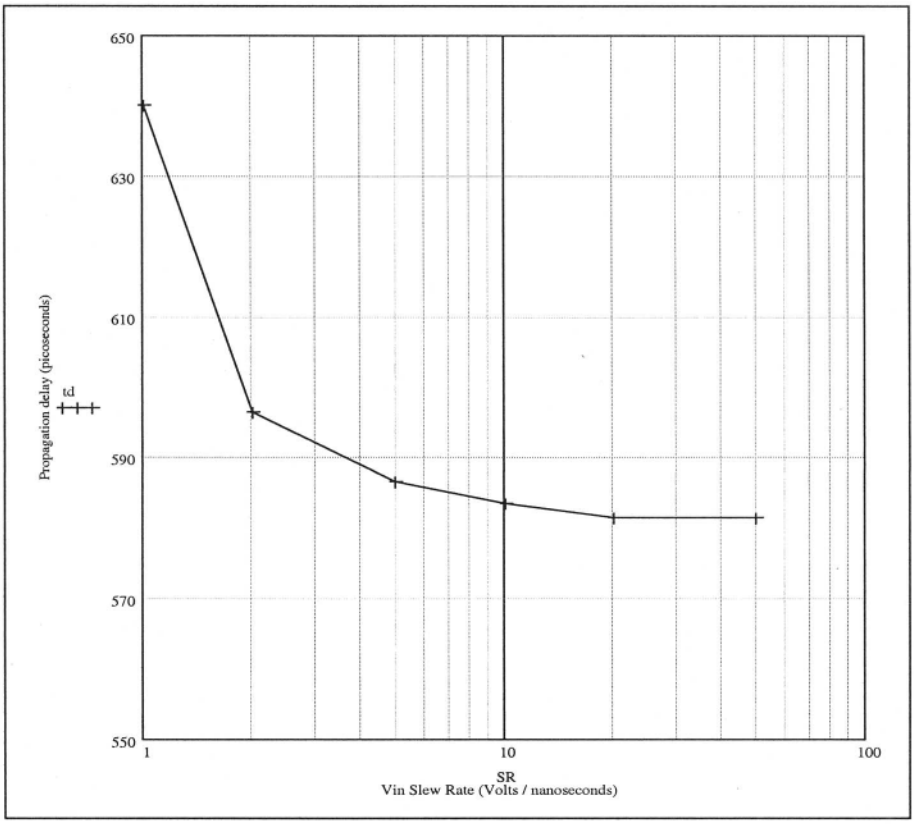


Figure 2-7. Propagation delay vs. input slew-rate

The slew-rate limitation of a differential pair is explained at length in reference 2, which is most likely the best description of a differential amplifier operation.

It is important to note that our macro-model in figure 2 – 3 can be refined to handle nonsymmetrical slew rates of the input signal. To accomplish this, we add a constant current source I_1 in parallel to Gm.

In doing so we obtain a net current with a value I_1+Gm_{MAX}. For example, if we add a constant current source $I_1 = 4mA$ the current extremes become 9 mA in one direction and 1mA in the other **(=4 mA ± 5 mA)**. This results in slew rates of 18 and 2 V/nsec $(=I_{MAX}/CSR)$ accordingly. The consequence of adding this current source is similar to having an offset in the comparator – there is an unbalanced output state due to the constant current even in the absence of input voltage. To counter the effect, we add an offset voltage source in series with one of the inputs. The magnitude of the offset is: $V_{OS} =$ atanh $(I_1 / 0.06)$. In our example the value is atanh $(4 \cdot 10^{-3}/0.06) = 66.7657$

mV. The aspect of nonsymmetrical SR in the macromodel of the comparator and its application will be emphasized in chapter 3.

Having established the comparator behavior under various conditions, we ask the following question: how accurately can the comparator reproduce an input voltage pulse?

The dependence of the comparator on the input voltage level and slew-rate examined above has shown that the output delay is a distorted replica of the input and not a response with a constant propagation delay Td, as we would like.

1.4 Pulse reproduction fidelity

Our next example illustrates the fidelity with which the comparator can replicate an input pulse. To perform this test we proceed as outlined in figure 2 – 8: the positive input is connected to a positive voltage V_{INP} and the negative input voltage V_{INN} is varied so different overdrive conditions are tested. The result is akin to "slicing" the input pulse in time. If the delay of the comparator were constant as intended, the output would have been an exact replica of the input pulse with a constant delay. Since the comparator distorts the pulse behavior and is slower for low overdrive voltages we obtain the result shown in figure 2 - 9.

Figure 2 – 9 represents the input pulse (dark) and the output response (light) predicted by our model. The X-axis of the figure is the input voltage level and the Y-axis represents the delay for that amount of overdrive.

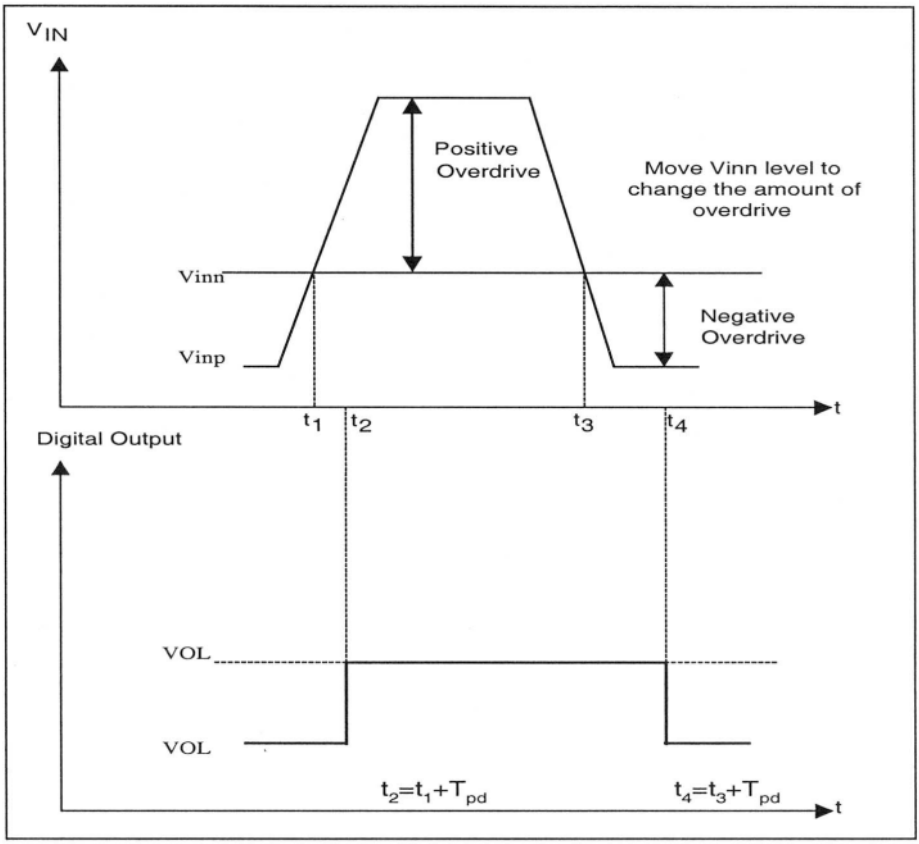


Figure 2-8. Pulse fidelity test

The graph shows that the reconstructed pulse is distorted especially close to the top and bottom of the waveform where the corners become round. The reason is that at the top and at the bottom, the amount of overdrive different for the rising and the falling edges of the signal. For example at t_1 there is more overdrive going up than at time t_3 when the signal is going down. In the middle of the pulse however, where the overdrive voltage exceeds the 15-20 mV of overdrive the delays are almost constant. For this reason the reproduced pulse shows rounded comers due to low overdrive conditions. The distortion in this example is limited to overdrive voltage and not to slew-rate limitation (the input signal has a slew-rate of 1 V/nsec while the comparator is limited to follow slew-rates of less than 10 V/nsec). This example illustrates how a system considered to be linear can distort particularly small signals, while for large signals it is more predictable and linear. This is not a trivial or predictable behavior and while counterintuitive is a very critical consideration in the design of an A/D converter.

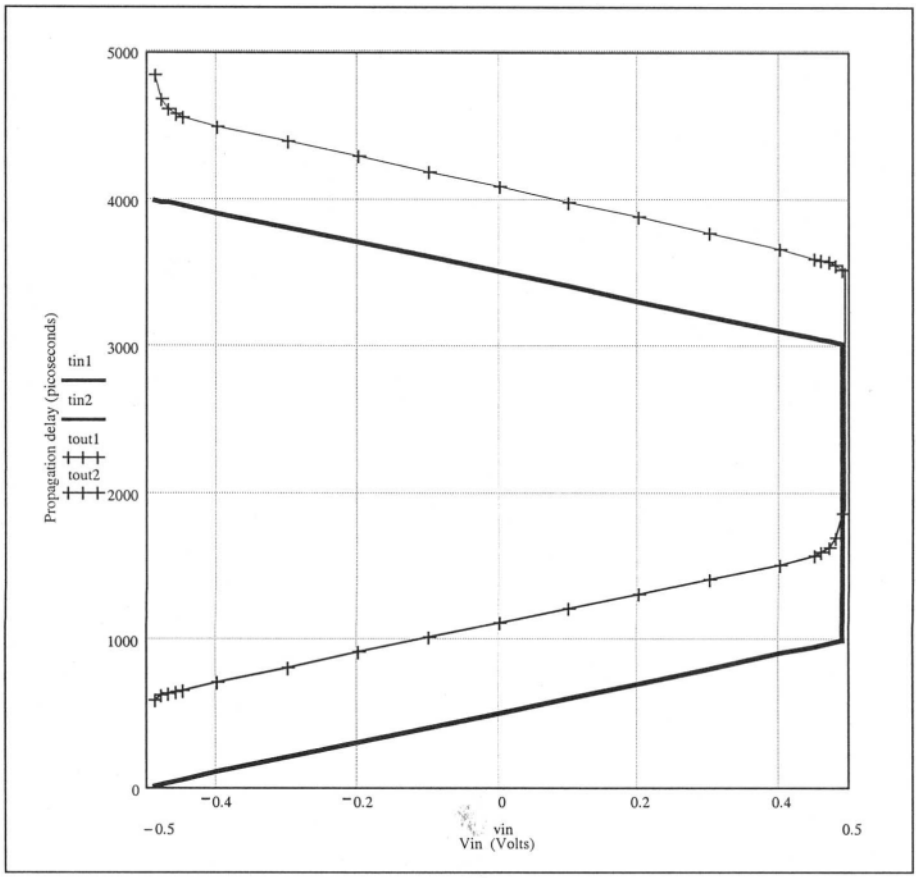


Figure 2-9. The original pulse (dark) is distorted due to the increased delay at low overdrive voltage (light)

To complete our macro-model description we note that most comparators have an additional input available to the user. This is the latch command. The latch command is a logic signal supplied so the output signal is maintained even when the input voltage is removed. In our model this is implemented with U1(see figure 2 – 3). The latch usually employs positive feedback circuitry in order to accelerate the reaction to the latch signal. In this book we will not consider the regeneration time of the latch (see reference 3 for a discussion on this subject). A reasonable rule of thumb for the latch to acquire the signal correctly (assuming realistic amount of gain) is that 5 to 10 mV will be sufficient to drive it to the correct final value given the positive feedback.

Our SPICE simulations have the latch function modeled with an edge-triggered flip-flop, with 10-mV noise margin and propagation delay of 1 picosecond. Obviously this is an unrealistic value for any technology available today, but the aim of this book is to concentrate on the analog front end behavior of the comparator and not delay fluctuations caused by digital blocks.

Finally our model contains the voltage controlled current source G_{OUT}, resistor R_{OUTB} and diode D11 in front of the latch for the level shifting required by U1. This is required so the voltage on the node **pole** is shifted to become an ECL logic level. The voltage V_{SS} constitutes the ECL threshold level.

1.5 A CMOS comparator model

For the sake of completeness, we address now the modeling of a CMOS comparator. The model suggested for the bipolar comparator, used the nonlinear voltage-to-current transfer function of the form:

$$I \cdot \tanh\left(\frac{V_{in}}{0.06}\right) \qquad 2.10$$

The hyperbolic tangent for the bipolar differential pair (see reference 4), is based on the diode behavior (current versus voltage). The 0.06 denominator is merely the term $2 \cdot V_T$ and is an approximation derived by using the thermal voltage V_T (approximately 26 mV at room temperature).

The benefit of using the hyperbolic function as explained above is that it can model a limiting transfer function and at the same time it is a well-behaved function (a continuos function with continues derivatives).

For this reason, an attempt is made to observe the adequacy of modeling a CMOS comparator using a similar model as that seen in figure 2 - 3 but with new component values. Clearly a CMOS transistor has a widely different transfer function than a bipolar transistor (the relation between current and voltage is quadratic rather than exponential) and for the same current the transconductance of a CMOS transistor (gm) is much lower than that of a bipolar transistor (see again figure 2 – 2). To model this difference, the GM used in our CMOS comparator model is:

$$I \cdot \tanh\left(\frac{V_{in}}{0.25}\right) \qquad 2.11$$

To observe the adequacy of the model, a CMOS comparator used in a commercial 10 bit high speed A/D converter manufactured in a 0.8µ CMOS process was evaluated and the propagation delay versus overdrive input voltage was measured. Figure 2 - 10 is a clear indication that the model is a relatively good approximation of the physical CMOS comparator behavior.

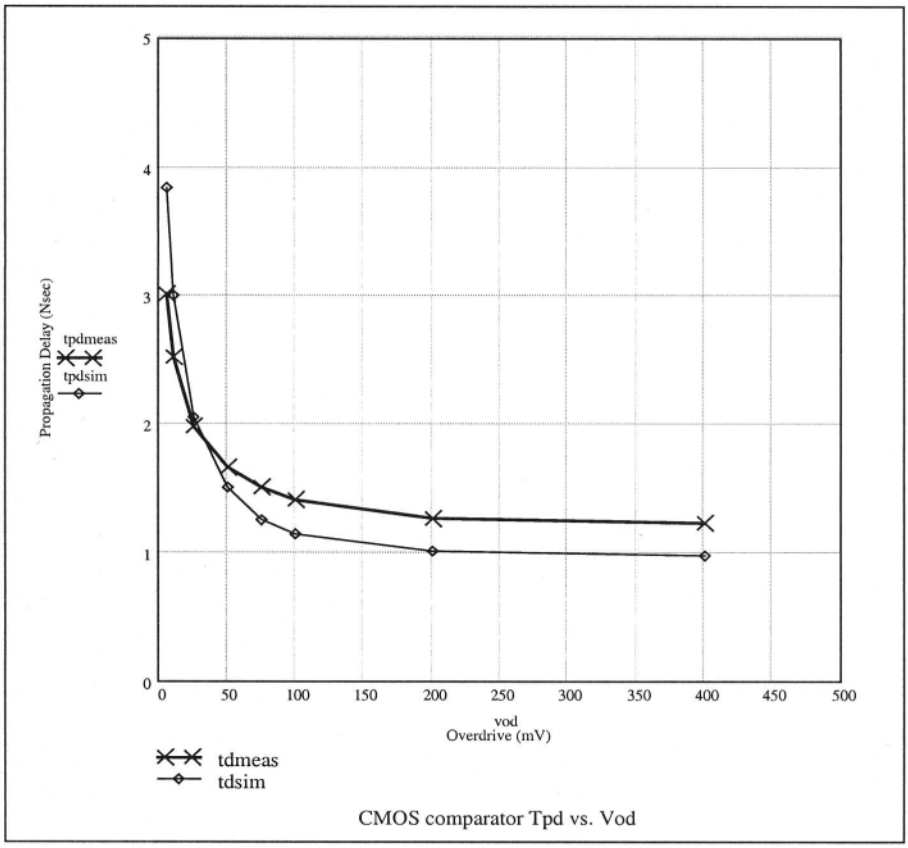


Figure 2-10. CMOS comparator propagation delay vs. overdrive voltage

For this model the component values are:

- CSR = 0.25 pF
- Gm = 1mA · tanh (vin/0.25)
- Rg = 2.25K Rp=1K
- Cp = 1pF
- El=2

The choice of components result from the intent to examine a comparator with a bandwidth BW = 150 MHz, open-loop gain A_{OL} = 25 dB and slew rate SR = 4V/nsec.

The fact that our model closely approximates the transistor model and actual measurements of the commercial device indicates that this is an acceptable macro-model, which can be used both mathematically as well as in Spice simulations. The model describes closely not only the linear behavior (BW) but also the nonlinear effects (SR).

In the next chapters we will take advantage of this model especially to reduce simulation time in Spice while describing various A/D topologies.

SUMMARY

A model was developed to represent the differential behavior of the comparator for both linear (bandwidth limitations) as well as nonlinear conditions (slew-rate limitations). This model is adequate for modeling both bipolar and CMOS comparators. No attempt was made to model the common-mode behavior, as the intention is to give the reader a means of playing "what if scenarios" with A/D converters and observe basic limitation of various topologies rather then secondary order effects.

To model a comparator, the following procedure is used:

- the nonlinear GM in the model has a multiplier for the TANH function equivalent to the front end differential pair tail current
- the slew rate is then modeled by picking a capacitor CSR such that:
  $SR = dV/dt = Itail/CSR$

To lengthen the amount of time of slew rate condition the voltage sources V2 and V3 can be used.

- Rg is chosen to create a pole with CSR such that the time constant of the pole is only a fraction of the pole created by $R_P \cdot C_P$ and the delay associated with it is insignificant.
- El is chosen so the open-loop gain of the comparator is:
  $A_{OL} = G1 \cdot Rg \cdot E1$
- R_p and C_p are chosen to create dominant pole such that $T_D = R_P \cdot C_P$ which is the specified delay time for large overdrive condition in the manufacturer' data sheet.
- The comparator model can be enhanced to account for non-symmetric slew rates on the rise or fall times with the help of a constant current source. When this option is considered the comparator should include a voltage source at the input that accounts for the offset of the circuit. The magnitude of this source is equal to atanh(I_1).

PROBLEMS

1) Using the comparator model of figure 2 – 3 slide the reference voltage on V_{INN} to a sinusoidal input voltage with 1 V p-p signal and frequency of 10 MHz and check the fidelity of the output signal as explained in figure 2 - 8. (Find the phase fidelity at various overdrive levels).

2) Repeat problem 1 when the input frequency is:
   a) f1 = 100 MHz
   b) f2 = 500 MHz
   c) f3 = 1 GHz

3) Repeat problem 1 for the case that the open loop gain of the converter is 46 dB and all poles remain unchanged. Does the result change?

4) Repeat problem 3 for the cases required in 2.

Chapter 3

Flash A/D

1. INTRODUCTION

The flash A/D has the simplest topology among all converters. Of all converters, this is also the fastest. The flash technique is also known as the parallel-approximation.

In this chapter we examine the "signatures" caused by various errors in this converter and its behavior under practical physical conditions such as input frequency, slew-rate conditions, noise, etc. The idea is to learn how these limitations lead to a particular signature and how can the signature be explained physically and mathematically.

To attain the high conversion speed the flash converter uses a parallel array of comparators sampling the analog input simultaneously. Since one comparator is required for each quantization level the number of comparators is doubled for each additional bit of resolution. The drawback of the technique is a significant increase in power dissipation in comparison to other A/D topologies.

An N bit flash converter requires an array of 2^N-1 comparators. The analog input voltage is connected to one input of the comparator array while the other input of each comparator is connected to fixed reference voltages. These references represent 2^N-1 equidistant voltage levels corresponding to the 2^N-1 switching points between the voltage extremes of the A/D converter input range.

The technique has the disadvantage of requiring a large number of comparators, reference resistors, and interconnections, which increase exponentially as the resolution of the converter increases.

A typical A/D is illustrated in figure 3-1.

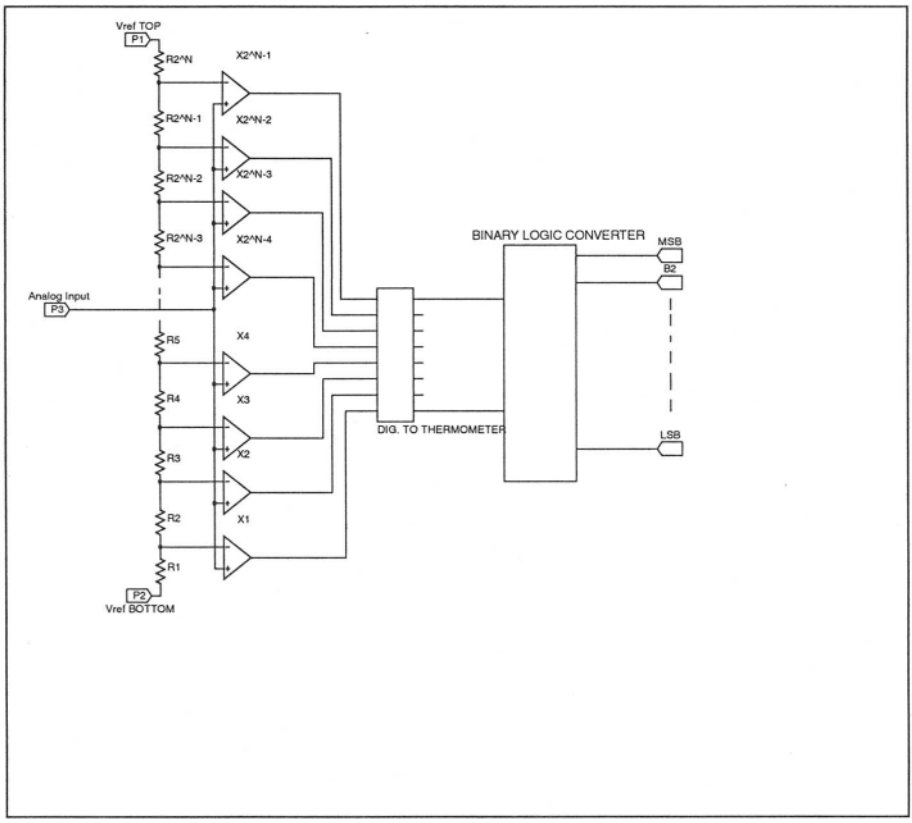


Figure 3-1. Typical flash A/D converter

There are 2^N resistors in the reference ladder supplying 2^N-1 equal voltage levels for the comparators. Assuming no loading by the comparator array – no comparator input current – the voltages supplied by the ladder are equidistant for matched resistors. The voltage difference between two adjacent resistors is equal to:

$$V_{LSB} = \frac{V_{REF\_TOP} - V_{REF\_BOT}}{2^N} \qquad 3.1$$

where V_{REF_TOP} and V_{REF_BOT} represent the two voltage references supplied to the converter.

The converter operates as follows: on the rising edge of the conversion command – clock - the A/D samples the analog input concurrently on all comparators. At that moment each comparator with the analog input below the reference responds with an output of "0", while those with input above the reference respond with an output of "1". On the falling edge of the clock the comparator-array output is latched and its logic output is sent to the **thermometer decoder.** This device detects the location of the boundary transition where "0" to "1" occurs in the comparator array. Theoretically there should be **one-and-only-one** comparator whose reference input is the closest to the analog input making this operation the equivalent of a priority decoder. The next operation in the chain of events is the thermometer to binary decoding operation. As the name implies the output of the thermometer decoder is converted now into a binary word and is directed to the output drivers.

Since the number of comparators is large, in a practical converter, the array is partitioned into several columns. Each column has its own decoder and all columns combine their output into a column decoder where the final binary word is obtained. Each column has its own set of latches contributing to a delay for the column decoder and then an additional delay for the decoder of the columns. These delays occur in a "pipe-line" fashion on subsequent clock edges using the column and decoder latches. In other words, the decoding and all remaining logic is performed on subsequent clock edges while newer analog input samples are being acquired by the comparator array.

To understand the flash A/D concept we illustrate its operation with an example. In this first example we assume that no errors occur in any of the comparators, reference ladder, or decoder.

Assuming a three bit converter with references of $V_{ref\text{-}bottom} = -1V$ and $V_{ref\text{-}top} = +1V$, each reference point in the ladder is greater than the one below by $0.25V\ [\ (V_{ref\text{-}top} - V_{ref\text{-}top})\ ]/\ 2^3 = 2V\ /\ 2^3 = 0.25\ V)$. If we assume for the moment an analog input voltage of 0.2V then the comparator-array and thermometer-decoder respond as outlined in table 3-1:

Table 3-1. Converter code vs. analog input

Analog Input	Comparator in the array	Comparator Reference	Comparator-Array Response	Thermometer Code
	X1	-0.75	1	0
	X2	-0.50	1	0
	X3	-0.25	1	0
	X4	0.00	1	1
0.20	X5	0.25	0	0
	X6	0.50	0	0
	X7	0.75	0	0

In our example, comparators XI through X4 respond with an output logic of "1" (input higher than their references). X5 through X7 respond with an output of "0" (input lower than their references). The thermometer decoder detects that comparator X4 is the only comparator at the transition edge from all "1" to all "0" and therefore its output is at a logic "1" for this comparator only.

A practical implementation of a bipolar thermometer-decoder device is shown in Figure 3-2.

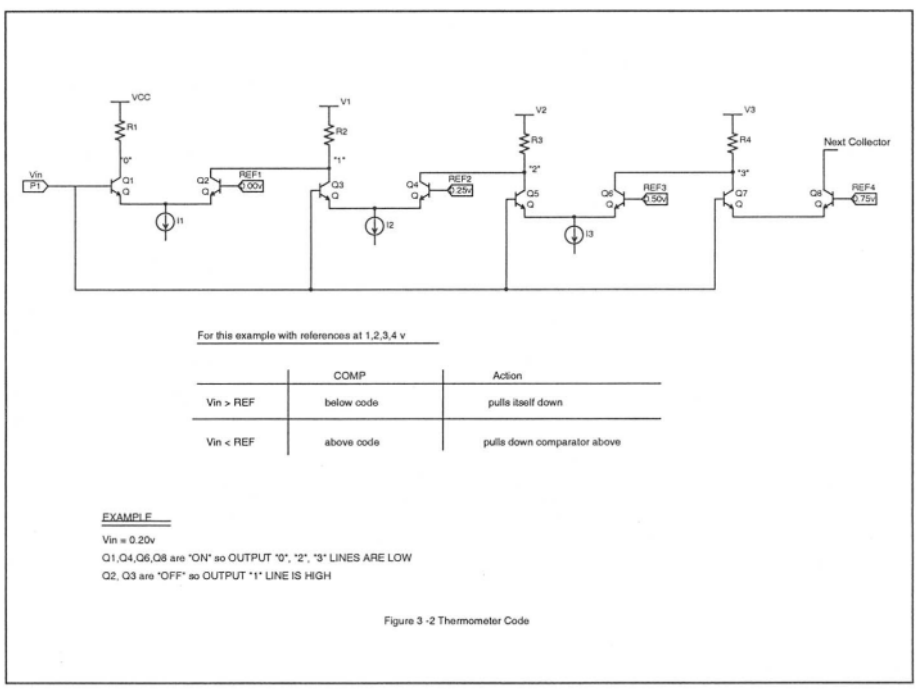


Figure 3-2. Implementation of thermometer decoder - example

In the figure the analog input is equal to 0.2V (as in table 3-1). Transistors Q1 and Q2 represent block X4 in the table while Q3, Q4 are the equivalent of X5 and so on. In this example the comparators and thermometer-decoder are combined in one operation.

For the case depicted in figure 3-2 transistors Q1, Q4, Q6 and Q8 are "ON" resulting in outputs "0", "2" and "3" to be "LOW". Output line "1" however is the only line that is "HIGH" as a result of transistors Q2 and Q3 being "OFF". The result is therefore a ***one-and-only-one of 7 lines*** at a logic "1".

As shown in figure 3-1 the last step in obtaining the A/D output code is the conversion from the thermometer into a binary code. This last function is conceptually trivial and it is usually performed by simple combinatorial logic (OR gates).

The dynamic operation of the logic can be fairly complex however, requiring careful delay analysis of the timing in various signal paths. Errors can result for example from comparators that are not being tripped simultaneously due to layout errors or by metastable states in the latches. In this book we will not elaborate on the subject of logic timing because the digital circuit complexity is limited to an analysis of propagation delays through this combinatorial logic. This task is a fairly mundane given the fact that delays in the digital path are additive. Analysis of various digital-decoding techniques, race factors, metastable conditions in comparator arrays, and latches can be reviewed in references 3,4,8,24,26,34 and 35.

1.1 A practical 3 bit A/D

Having discussed the general operation of a 3-bit converter as a block diagram we proceed to examine the behavior of a practical 3-bit flash converter (see figure 3-3) with the help of spice analysis. The converter is synthesized with components that have real life physical limitations resulting in a realistic dynamic response.

We assume that the reference ladder is comprised of eight 100Ω resistors (R1 through R8) connected to the positive and the negative references (VRT= +1V, VRB= –1 V). These resistor values are quite small, contributing to a fairly high load on the voltage reference (2 V / 800 Ω = 2.5 mA). In practice the ladder tends to have a low resistance value so the comparator array input currents contribute only a fraction of the total ladder current.

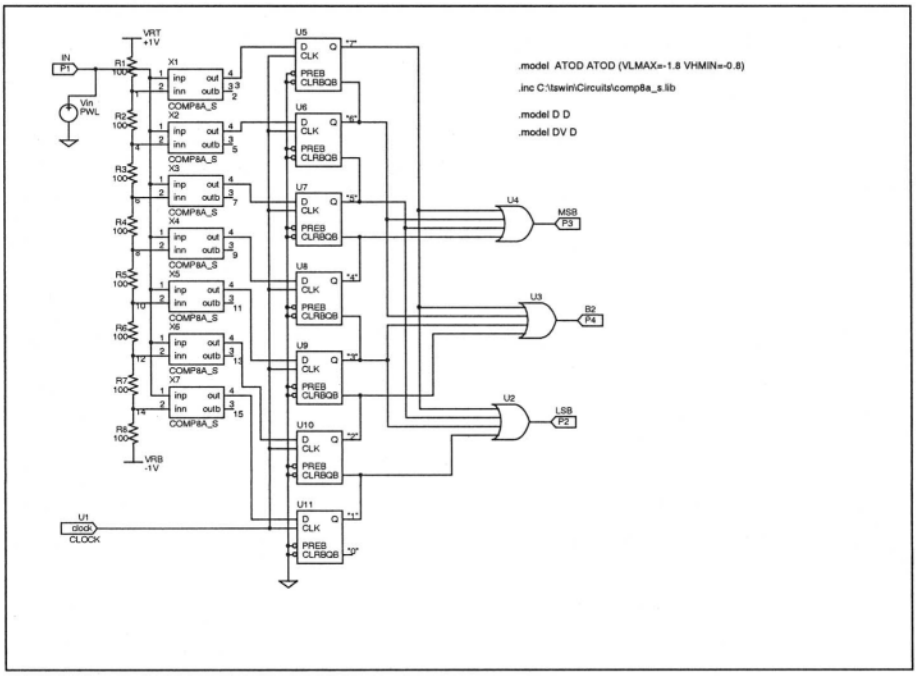


Figure 3-3. 3-bit flash converter

By employing a low resistance ladder, the comparator reference is not affected by either static input currents or dynamic perturbations of the input currents. Since the current gain β is reduced as the frequencies increase, the dynamic base currents increase creating a variable load that changes with frequency. This effect can be modeled by adding a capacitor across the comparator-input terminals. The comparator array used to analyze the converter' operation is similar to the device examined in chapter 2.

The differential input impedance of each comparator is **100 KΩ** contributing to an error of one part in one thousand **(0.1% = 100Ω / 100 kΩ)** which is more than adequate for a 3-bit converter.

The other comparator' parameters are: **CSR=0.4pF, R_g=24K, R_{pole}=0.75K, C_{pole}=1pF** and the input **tanh** function uses again a tail current of 5 mA as studied in chapter 2:

$$G = 5mA \cdot \tanh\left(\frac{Vin}{0.06}\right) \qquad 3.2$$

The comparator differential outputs OUT and OUTB have ECL logic levels and are connected to edge triggered latches U5 through U11 to obtain the thermometer decoder.

With reference to figure 3 – 3 notice that the output (*out*) of the lower comparator is wired-or with the complementary output (outb) of the comparator above. These connections implement the thermometer decoding in our converter so a **one-of-seven** code is achieved when the analog input voltage crosses one of the threshold voltages established by the reference. The comparator-array response "is frozen" in time after the clock signal switches to a logic low. In this example, the thermometer to binary decoding operation is done in a "brute-force" fashion using three OR gates U2, U3 and U4 to attain the three binary output codes.

The major point of interest in the simulation is the behavior of the analog front end of the converter rather than the decoding circuitry. For this reason in our SPICE simulation, the latches propagation delay is set to 1 psec (so the delay contribution of the latch is negligible). Using an input ramp moving from minus to plus full-scale and performing a transient analysis with the SPICE simulator, we obtain the response of the converter illustrated in figure 3 - 4.

It should be noted that the gain of the comparator used in our example is relatively high for a three bit A/D (in excess of 60 dB). A suggested exercise for the reader is to analyze what happens when the comparator gain is reduced (problem 3 – 1).

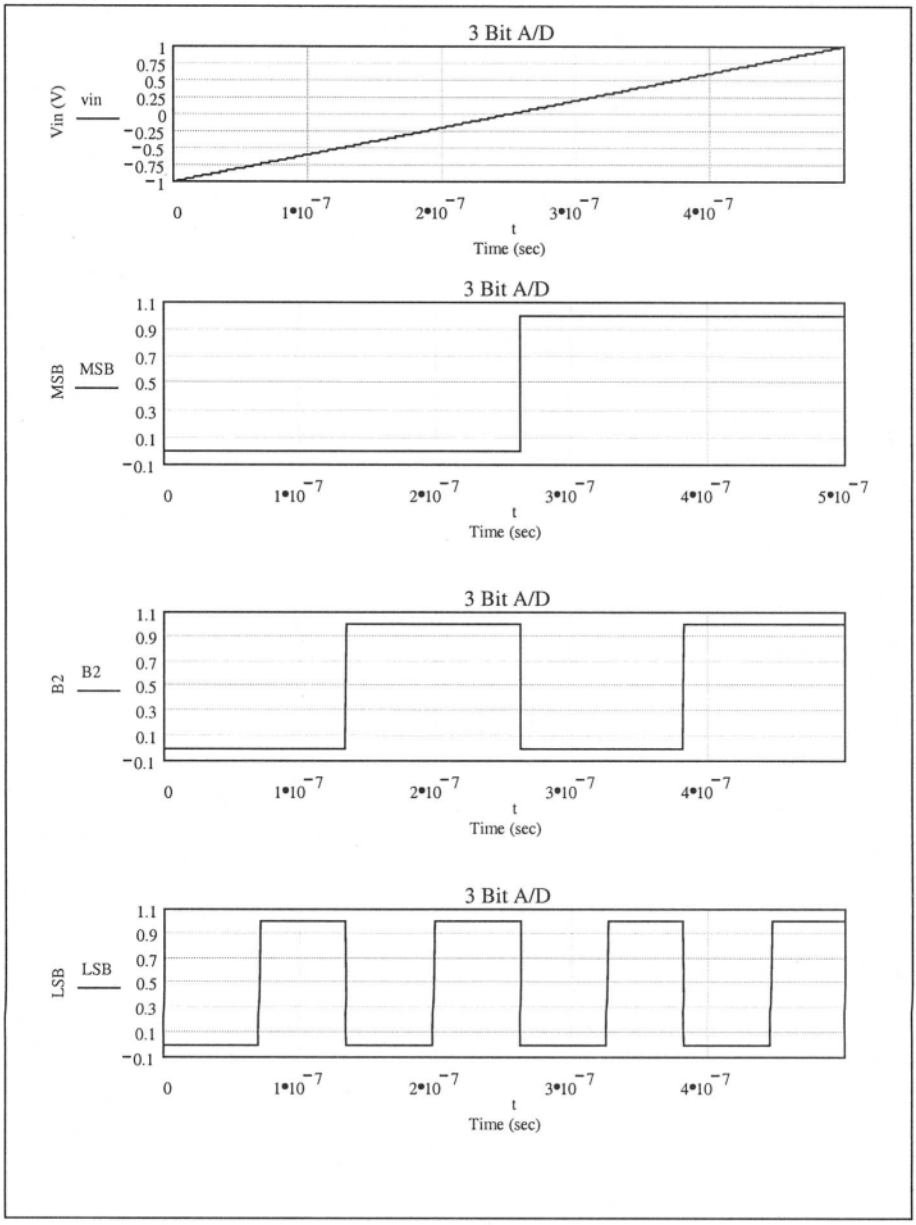


Figure 3-4. The decoding mechanism used for the 3-bit flash A/D

1.2 A 6-bit A/D

Now that the flash conversion technique has been illustrated in a small converter we take a look at a more accurate A/D of the kind more likely to encounter in day-to-day applications. Of course when doubling the resolution of the A/D from three to six bits we increase the number of comparators from $7 (= 2^3 - 1)$ to $63 (=2^6 - 1)$.
Again as in the previous example we use the comparator macro-model proving once more its usefulness in reducing simulation time in SPICE.
Given the size of the comparator array we investigate the converter dynamic performance when the comparator array is partitioned into four columns. As shown in figure 3 – 5, each column uses 16 comparators in the block called 16COMPS. The importance of the partition of the comparator-array into columns will be illustrated by example in the next paragraphs.

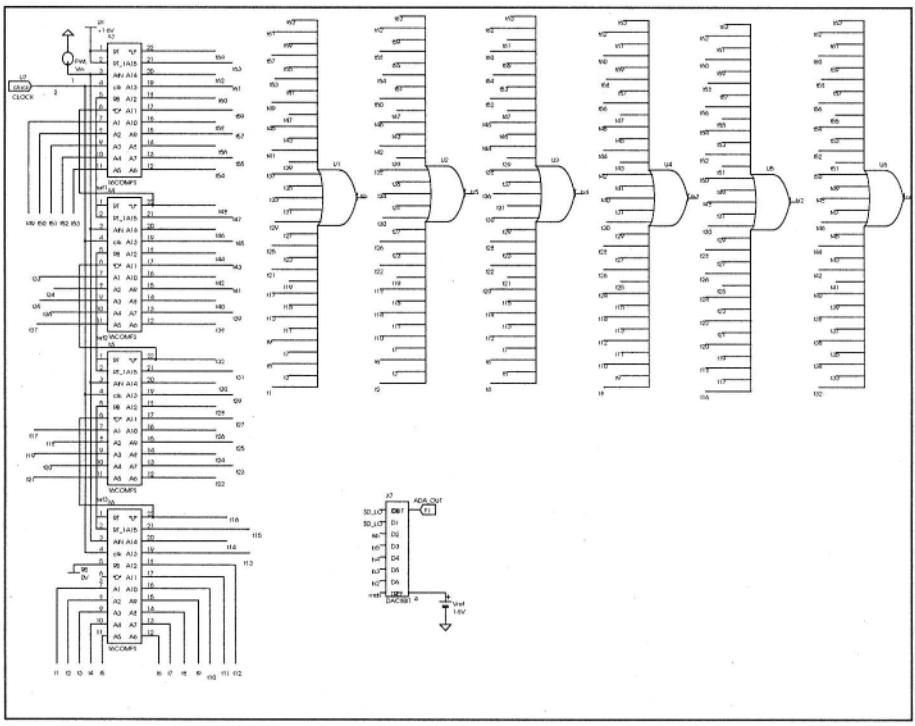


Figure 3-5. 6-bit flash A/D

In the simulations, the comparator and latch are modeled as a single block called C_LAT1. The components used in C_LAT1 are the same components as in the previous chapter. The columns are stacked on top of each other so

the bottom of the first column's reference is connected to the top of the second column reference making it a continuous network. Similarly the top of the last digital output is "wired-or" to the bottom of the comparator of the next column. The outputs of all columns are decoded again as in the previous example using the OR gates U1 through U6. Finally to reconstruct the output data for the FFT analysis we include the D/A – X7 - converter following the A/D.

The dynamic performance of the 6 bit A/D is tested by performing FFT tests at several input frequencies and measuring the SINAD (or TDE) defined in chapter 1. In all our FFT tests we take advantage of the high speed of the comparator by employing a clock with period of 1nsec and a 50% duty cycle. This circuit file is supplied in Appendix A.

Our investigation of dynamic performance consists of an envelope test as explained in chapter 1. The sampling frequency for the device is set at 1GSPS (Giga-samples/second). Since we intend to minimize the computation time the FFT analysis is limited to 16 points. As a benchmark for simulation run time: when using a 450 MHz Pentium III computer with 128 MRAM the transient analysis completes in a little more than one minute (68.4 seconds) illustrating once more the benefit of using the comparator macro-model developed in chapter 2. Simulating the same A/D with actual transistor models would undoubtedly have taken several hours for a similar dynamic test given the large number of nodes.

Several points should be made about the FFT test. With 16 time points the beat frequency is calculated to be **(1000 MHz)/16 = 62.5 MHz.** Using 16 FFT time-points results in only 8 harmonics. Each harmonic bin is 62.5 MHz wide so the frequency resolution is fairly poor. It is interesting to note that at frequencies of approximately 150 MHz, the resulting SINAD is about 6 dB lower than at low frequencies. This means that the effective number of bits – ENOB – is about one bit lower. This represents the large signal bandwidth of the converter. To present the SINAD as a function of analog input frequency we vary the input in increments of 62.5 MHz from 62.5 MHz to 437.5 MHz. The resulting SINAD (assuming no static errors for the 6-bit converter) is illustrated in figure 3 – 6. As expected the SINAD decreases as the input frequency increases due to the limited slew-rate and bandwidth of the comparator.

Note: *remember that having poor frequency resolution in TDE calculation does not limit the accuracy of the result (all harmonics are used for SINAD).*

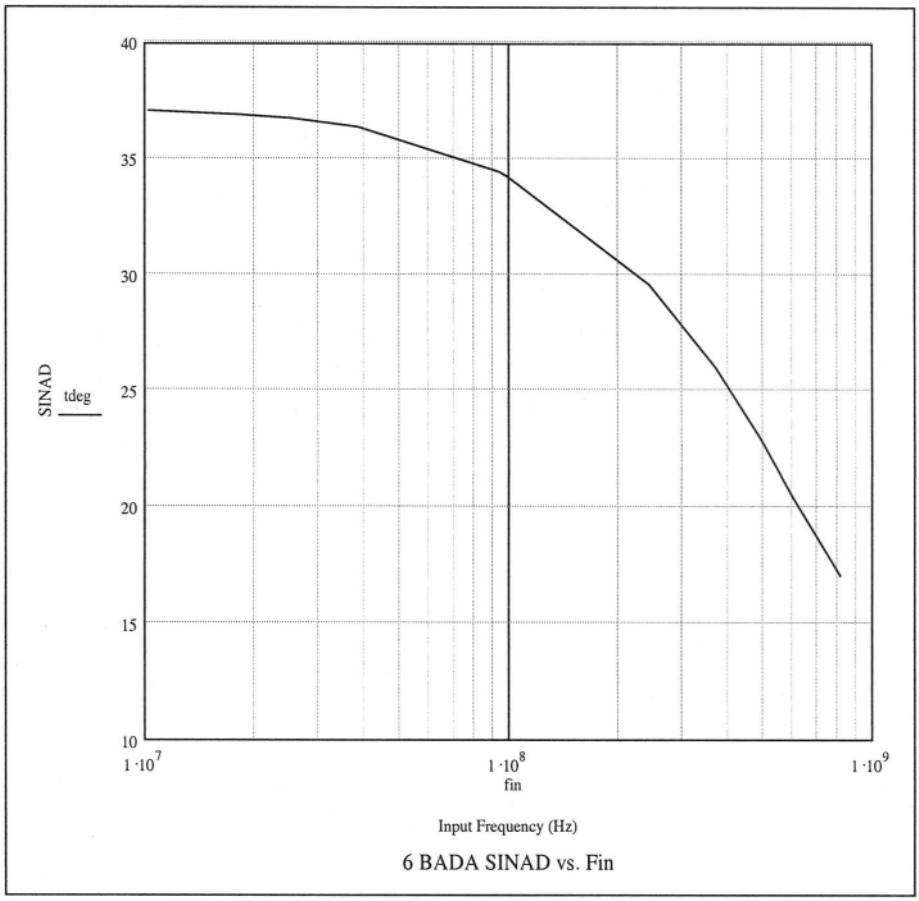


Figure 3-6. Dynamic performance of the 6-bit flash converter vs. input frequency

1.3 ILE / DLE error effects on SINAD

The next topic we consider for the 6-bit converter is its behavior in the presence of linearity errors. This is first observed in the time domain by using an input sinewave. We will show that the linearity error location along the full-scale range makes the converter behave in drastically different manner and that the particular error location may result in vastly different signatures in the time and frequency domain. To see two different cases of linearity errors we plot the reconstructed signal at the converter's output in the presence of comparator offset voltages. In real life, these errors can result from offset errors that are caused by static or dynamic reasons. For example

in a bipolar differential pair, the offset can result from lithographic variations or topographic differences in the transistor's neighborhood. As a rule, this is a static error. In CMOS pairs, the offset can originate from similar reasons or from dynamic motives. For example if the input pair is subjected to large overdrive conditions for long periods of time prior to being switched to the reverse condition it can show large threshold voltage shifts. This makes the comparator exhibit an offset that depends on the overdrive voltage (ref. 44).

In the first case, we examine what happens if an offset error occurs in the comparator detecting the LSB (the bottom comparator in the array). Figure 3-7 (a) shows an exaggerated time domain reconstruction and clearly illustrates that this offset causes the reconstructed sinewave to be distorted at the bottom of the sinusoid. If the exact same error occurs in the comparator array is in the middle of the array (the comparator located at mid-scale) than the time domain distortion is observed at mid-scale as shown in figure 3-7 (b).

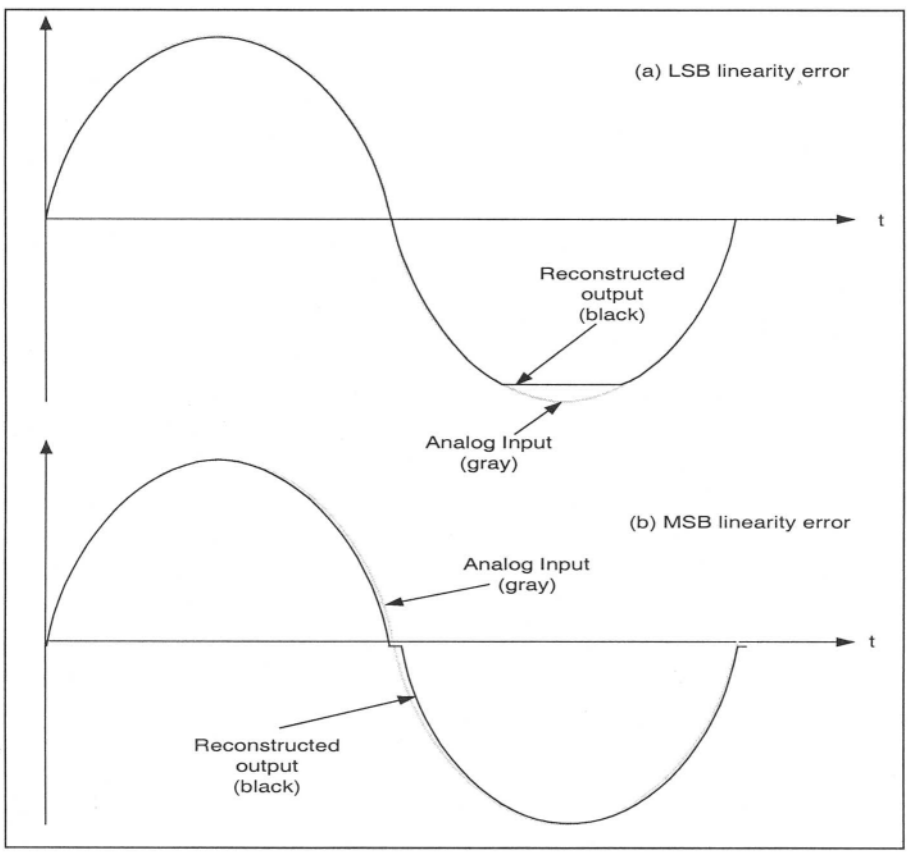


Figure 3-7. Flash A/D distortion caused by comparator offset

The first obvious question is how does the linearity distortion behave in the frequency domain and how does it affect the SINAD? To answer these questions we perform a SPICE transient analysis and analyze the associated FFT. The harmonic content of the FFT analysis for the two cases are summarized in table 3-2:

Table 3-2. Comparator offset effects on SINAD

Harmonic	LSB error	MSB error
H2	-45.87	-51.55
H3	-44.13	-49.84
H4	-48.90	-50.93
H5	-51.53	-53.57
H6	-52.60	-55.27
SINAD	35.22	35.98

From table 3 – 2 we notice that for all practical purposes there is very little difference in SINAD between the two cases (0.76 dB is equivalent to 9.1% difference). The error observed is mostly due to the limited resolution and round off in the FFT calculation. The reason for the harmonic differences is that in the LSB case, the sinewave is distorted at the bottom tip, while in the MSB case it is "chipped" in the middle. Given the time domain difference between the two cases it is reasonable to expect different harmonics distribution. However, since the amount of energy contained in the error is equal in both cases **(=one LSB)** the resulting SINAD is roughly the same.

For comparison purposes the same converter with no DLE errors has a **SINAD=37.03** dB at the same input frequency **(fin=3.90625 MHz).**

From this example we can conclude that in a flash A/D converter, the location of a comparator-offset error affects the SINAD very little. The reason for this independence results from the fact that each comparator determines its output based on its own reference level making the amount of discontinuity in the transfer curves of the two cases very similar.

It will be seen in later chapters that other A/D topologies have significantly different SINAD depending on the error location along the quantization-decision chain.

1.4 Delay between analog-input and clock

Another practical issue that needs consideration in a flash A/D arises from the usage of a large number of comparators in high-resolution converters. This leads to a layout where the distance from the first comparator to the last can be significant. Usually in an I.C. the highest functionality yield is obtained when the layout of the device is such that an

aspect ratio of a square is achieved. This consideration forces the layout of the comparator array to be split into several columns.

An example of this layout is illustrated in figure 3 - 8.

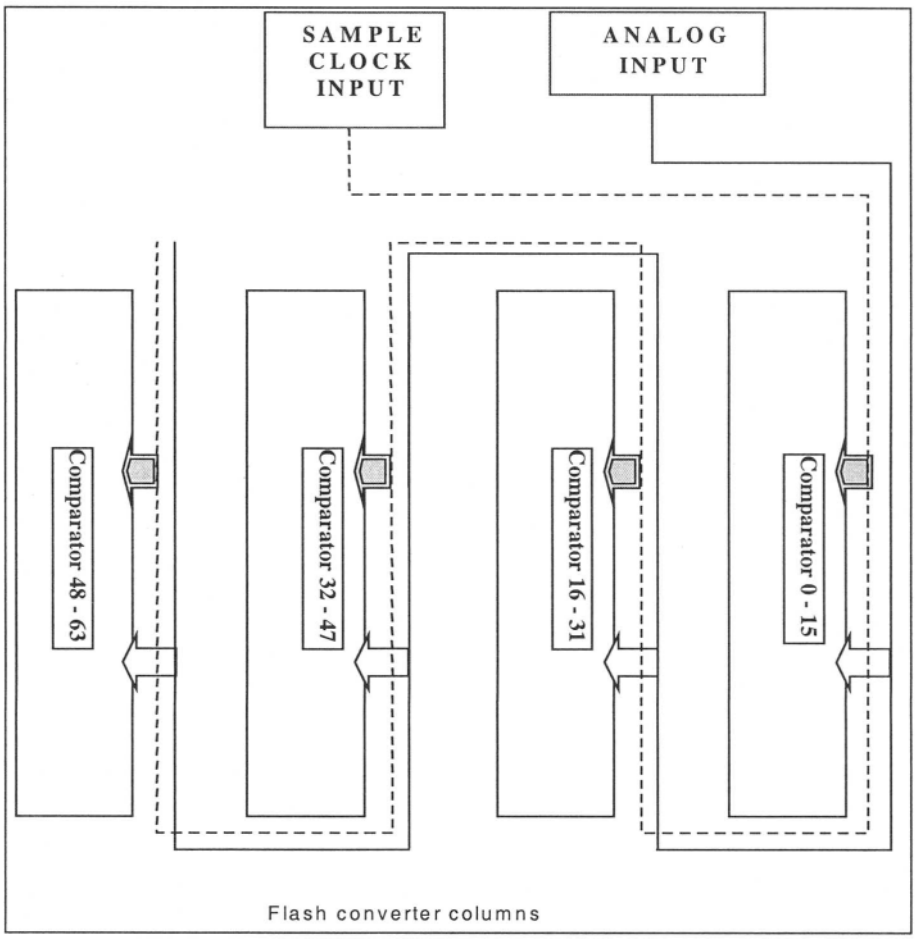


Figure 3-8. Typical layout for an integrated flash A/D converter

The figure illustrates how the comparator array is partitioned for a 6-bit converter. The 63 comparators are divided into four columns each containing 16 comparators. Bearing in mind that long conductor lines on the chip have large parasitic capacitance to the substrate and finite line resistance it stands to reason that the delay of the signals across the chip can be significant in comparison with the frequencies of interest. Therefore the routing of the clock and analog signals in a flash converter needs careful consideration.

Typically the analog input is routed in close proximity to the clock line so minimum delays occur between the two signals. By observing this requirement, the layout guarantees that when the comparator makes a decision the input signal has been delayed for the same amount of time as the latch command.

Similarly the digital decoding of the columns has to take into consideration the delays caused by the parasitics and has to align the decision time for the column so all comparators in the column have sufficient time to propagate their decision. An equivalent constraint is put on the decoding of the columns.

Given the delays between columns due to parasitic capacitance we ask how does this delay affect the dynamic performance (SINAD).

As a practical case we consider a converter using a high-speed BiCMOS process with interconnects in metal or silicided poly material. In our example we assume that the metal layer has a conductor-substrate capacitance of **65 aF / μ^2** (1 aF = 10^{-18} F) and the poly has a conductor-substrate capacitance of **85 aF / μ^2** .

Assuming that the analog input and the clock lines have a width of **2.0μ** and the IC length is 5 mm, the trace length has to take into account not only the die length but also the interconnections needed to access each comparator. With a total conductor length of **9000 μ (=5mm** actual column **length + 4mm** interconnects inside the comparators) we calculate the metal line to have the following characteristics:

Capacitance: **9000μ x 2μ x 65 aF**

$$\rightarrow C_{TOTAL} = 1.17\ pF$$

Resistance: **9000μ/ 2μ = 4500** ; if the trace resistance is **85 mΩ /**

$$\rightarrow R_{TOTAL} = 85 \cdot 10^{-3} \cdot 4500 = 382.5\ \Omega$$

$$\rightarrow Delay = R_{TOTAL} \times C_{TOTAL} = 0.4475\ nsec$$

This column delay of 0.45 nsec is very significant in a converter sampling at 1 GSPS. If the same connection were implemented using the poly conductor, the resulting delay would be 40 – 50 times higher, or 20.6 nsec (poly resistance is higher than metal). The figures above do not take into account any capacitance fringing effects. If fringing is also considered an additional 50% increase in delays is observed resulting in a delay of 0.7 nsec for metal and 32 nsec for poly. This clearly illustrates the reason why this connection cannot be done in poly.

Given delays of 0.7 nsec/column we proceed to examine these effects on the dynamic performance of a 6-bit A/D.

1.4.1 Column delays in 6-bit A/D

To perform the analysis we simplify the problem by assuming that each column has a lumped RC delay of 0.7 nsec. Of course the proper approach would be to simulate the actual condition with distributed RC delays, but for the sake of limiting the simulation time we lump the effect to minimize the number of nodes required. The effect of column delays due to layout is first illustrated in figure 3-9 by simulating our 6-bit flash converter with an input clock with a period of 1nsec and an analog input ramp of 1.6V (=FSR) with a duration of 160 nsec. This is a signal with a slew-rate equivalent to a sinusoidal input with an amplitude of 1V and a frequency of 2 MHz. (both conditions represent an input slew rate of ≈10 V/μsec). To examine the column delays we measure the differential linearity as before. The results of the simulation are outlined in figure 3-9. The two graphs in the figure present a comparison of two devices: one with no delay and the other with 0.7 nsec delays skew between adjacent columns.

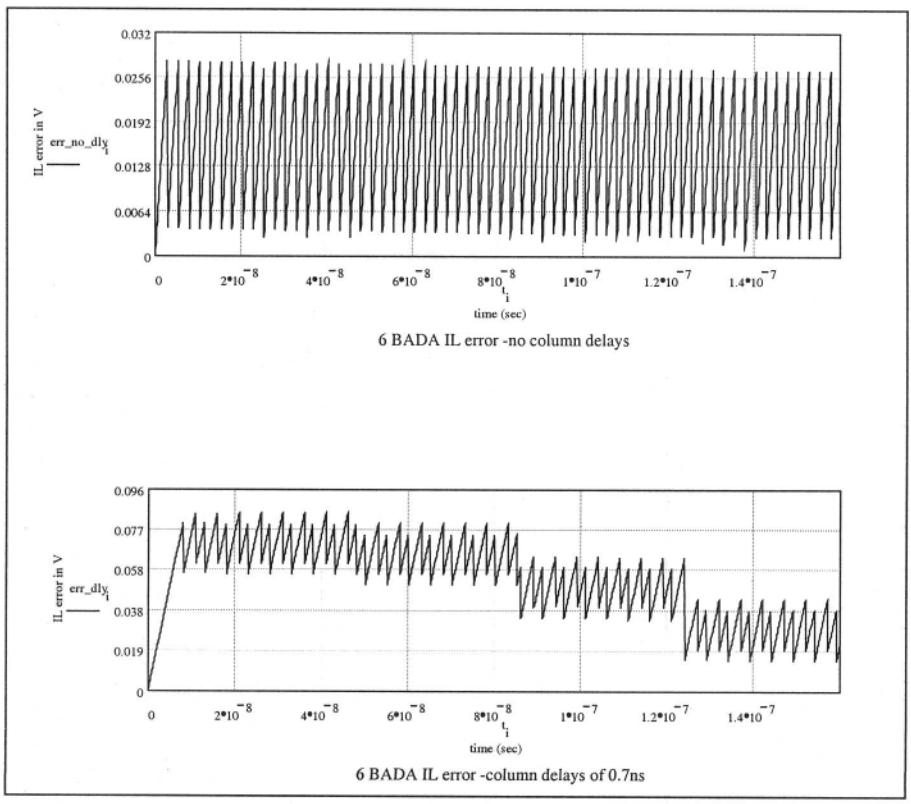


Figure 3-9. IL errors in a 6-bit A/D when the column delays are 0.7 nsec

If no delays were present between the columns we would expect the sawtooth waveform characteristic to the IN-OUT error signal. Since we use in our simulations a limited number of time points encoded at each LSB step we observe that the sawtooth "wobbles" somewhat due to the limited time resolution. The DL observed in figure 3-9 is of course exaggerated and shows four bands along the full scale. This is the result of having three discreet RC time delays between the columns rather than a distributed delay. If the delays were simulated as smooth variations rather than lumped RCs we would observe a smooth delay variation.

It is important to note that performing this test at low clock frequencies and slower input ramps will give results that are almost indistinguishable from the theoretical case with no delays. The faster the analog input the more evident the error. This result also suggests that a high input frequency FFT test which exercises a high slew-rate case is essential in finding this sort of signatures in a flash converters.

Table 3-3 summarizes the differences between a 6-bit A/D with column delays of 0.7 nsec./column and the case without delays. The analog input signal used for the SPICE simulation has a frequency of 3.90625 MHz and the FFT contains 256 data points.

Table 3-3. Harmonic distortion due to column delays in a 6-bit flash A/D

Harmonic	No column delays	With column delays
H2	-62.89	-29.60
H3	-47.41	-46.54
H4	-63.79	-51.53
H5	-50.30	-51.24
H6	-65.13	-52.89
SINAD	37.00	28.43

It is worth emphasizing that even at relatively low input frequency when the input signal has a slew-rate of less than 20 V/ μsec (about three orders of magnitude below the slew rate limit of the comparator) the differences between the two cases (zero and 0.7 nsec delay) is fairly significant. It is also evident from table 3-3 that the major differences between the two cases is in the even harmonics (since the column delay is creating a non-symmetric distortion). More importantly the second harmonic is the limiting factor in the overall dynamic performance. The SINAD difference between the two cases is equivalent to a loss of 1.5 effective bits (approximately 9 dB) at 3.9 MHz.

1.5 Slew-rate limitation

So far we examined the 6-bit converter dynamic performance based on the comparator of chapter 2. Our tests concentrated on the dynamic performance of the converter based on a comparator array with slew-rate limit of **+/- 10 V/nsec.**

Next we investigate the difference in signature caused by non-symmetric slew-rate limitation in the comparator. This may be the result of uneven capacitive loading of the differential pair or unbalanced driving impedance for the pair. To investigate this case, we use the new comparator model shown in figure 3-10.

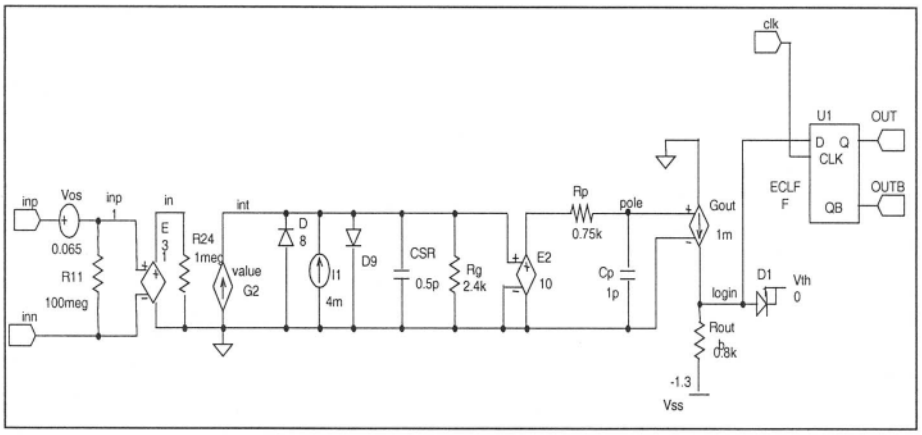


Figure 3-10. Slew-rate limited comparator **(SR+=2V/nsec., SR-=18V/nsec.)**

The major change to the original model is the addition of current source **I1=** 4mA. This change causes the negative slew rate to be limited to 2 V/ nsec while the positive slew rate is increased to 18 V / nsec. In effect the original slew rate of 10 V/nsec is changed by **– 8 V/nsec.** As a result of the addition of I1 the input of the comparator shows an offset voltage of 65 **mV[= atanh (4mA/0.06)]** which is compensated by the voltage source Vos in our model.

1.5.1 6-bit A/D with non-symmetric slew limited comparator

Using the new comparator model we perform the transient simulations for same input frequencies used with the original comparator.

In figure 3-6 we have seen already that at frequencies that are more than two decades below the comparator slew-rate limitation (10V/*nsec*) the original converter lost more than 20 dB in dynamic performance from its performance at low frequency. In other words when the comparator reached its slew rate limitation the distortion of the entire converter has become so bad that the converter lost much of its dynamic range (from approximately 37 dB to less than 15 dB).
For comparison purposes figure 3-11 presents the result of FFT tests on two variations of our 6-bit A/D:

- the original A/D with symmetrical slew-rate of +/- 10 V/nsec and
- an A/D with nonsymmetrical slew-rate of +2 V/nsec and –18 V/nsec.

The slew-rate of the input signal is:

$$SR=\left.\frac{dV}{dt}\right|_{max}=\left.\frac{d(V_{IN}\cdot\sin(\omega\cdot t))}{dt}\right|_{max}=\left.V_{IN}\cdot\omega\cdot\cos(\omega\cdot t)\right|_{max}=2\cdot\pi\cdot V_{IN}\cdot f \qquad 3.3$$

With frequency of 312.5 MHz the input slew rate is approximately 2 V/nsec. The original comparator can follow this input slew-rate but the comparator used for the second case can only react to signals limited to 2 V/nsec. For this reason we can expect that at very low frequencies (much below their slew-rate limit) both converters will have equal SINAD.

In the second case, since the converter will severely distort frequencies above 312 MHz we expect to have significantly reduced SINAD compared with the first case. The two converters studied in our example show equal dynamic performance at input frequency of 3.90625 MHz. It is interesting also to see that even at low input frequencies (< 10 MHz) the slew rate limitation of the comparator reduces the SINAD compared to the original device. This is not an unexpected result given that any amplifier will begin distorting a long way before it reaches its the slew rate limit.

The flash A/D examples examined in this chapter represent the major sources of error for this converter topology. To see the effects of dynamic loading the comparator model can be enhanced by the addition of a small capacitor across INP and INN in figure 3 – 10. With a capacitor of 0.1 pF the dynamic loading of each comparator will be approximately 10 kΩ at 160 MHz or 0.1% (= 10 kΩ || 100 Ω) given the comparator DC resistive ladder (see problem 7).

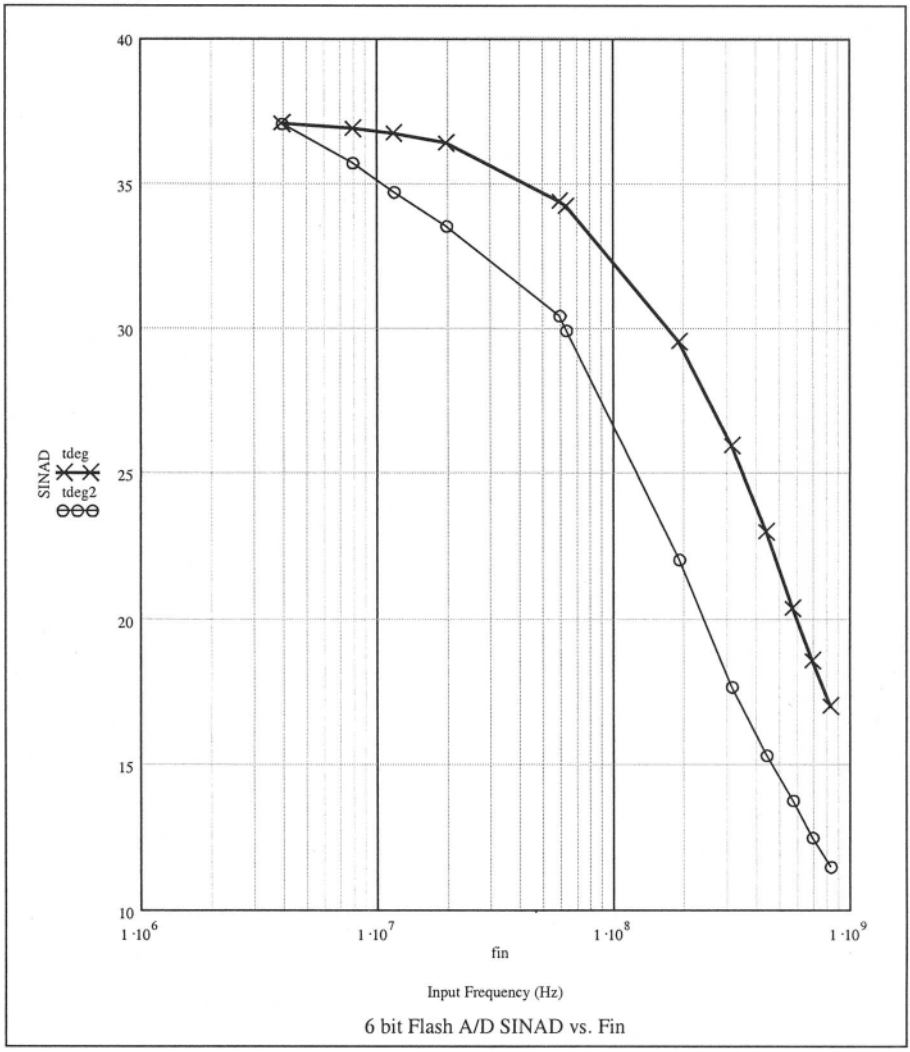


Figure 3-11. Distortion caused by nonsymmetrical slew-rate in the comparators of a flash converter

SUMMARY

- We analyzed the dynamic behavior of A/D converters with static errors by introducing offsets in various comparators. We observed that equal offset errors located at different quantization locations exhibit similar SINAD in the flash A/D. The individual harmonics however reside in different frequency bins for the two cases.

- H. Kimura et al., (70) shows that the dynamic performance of a flash A/D is related to comparator offset distribution – σ_{OS} – as:

$$SNR = \frac{V_{FSR}/\left(2\cdot\sqrt{2}\right)}{\sqrt{\frac{Q^2}{12}+\sigma_{OS}{}^2}} \qquad 3.4$$

  where Q is the quantization level of the converter.

- We tested the effects of propagation delay difference between the analog input signal and the column latch command. This experiment highlighted the reason for preferring certain conductor (metal) connection over others (poly). The column delays underscored once more the importance of performing FFT tests at high input frequencies so high slew rates are considered as delays between adjacent columns of the converter. For this reason it is customary to make a layout called a "binary tree". The binary tree, splits each signal - the analog connection as well as the latch command - as a tree: the "trunk" is connected to the outside world, while each branch connects a column. This technique assures the shortest propagation delay of the respective signals to the columns.

- Finally we tested the effects of nonsymmetrical slew-rate limitations in the comparator on the dynamic performance of the A/D.

PROBLEMS

1. Adapt the comparator model of chapter 2 to obtain an open loop gain of 40 dB while maintaining the BW and slew rate. Use the new comparator model in a 6-bit converter and compare the SINAD performance with figure 3-6.

2. Repeat (1) with open-loop-gain of 66 dB, **SR=+/- 10 V/nsec** and **BW= 1GHz.** In each of these first two exercises only one parameter is changed at a time so its effect can be seen clearly.

3. Consider a 6-bit converter using the CMOS comparator of chapter 1. Perform SINAD simulations at sample rate of 500 MSPS with input frequencies up to Nyquist.

4. Repeat (3) as follows: use an input frequency of 1-5 MHz. Increasing gradually the number of time points used in the transient analysis find what is the minimum number of points required so the SINAD varies by no more than 5% (0.42 dB).

5. Create a 1 LSB error in the ladder at the lowest comparator by changing its resistor value and maintaining the rest of the ladder as before. Using an input ramp observe the IL and DL. Repeat the error but replace the location of the error from bottom of the ladder to the middle of the ladder. Calculate the difference in SINAD between the two cases at an input frequency of 1-5 MHz (using the same number of time points).

6. Repeat the last example of the chapter with a 6-bit converter that has only one comparator with nonsymmetrical slew-rate limitation in the middle of the ladder and calculate SINAD at $\mathbf{f_{in}}$ **= 100 MHz.**

7. Add a capacitance of 0.1 **pF** across **R11= 100 kΩ** in figure 3 – 10 and compare the new flash A/D performance to that of the original A/D of figure 3 – 6 (no capacitance loading on the ladder).

Chapter 4

Track and Hold Amplifier

1. INTRODUCTION

The track and hold amplifier (THA) - also known as the sample and hold (SHA), is a very important component in systems where high speed A/D converters are used.

Having just discussed the flash A/D converter it is appropriate to emphasize that the flash is the only A/D converter that requires no front end sample-and-hold prior to the converter itself. The reason is that the analog input signal is processed simultaneously by all the comparators, which in a flash converter are designed to be very fast. The comparator array outputs are latched as soon as the comparator array has completed the acquisition of the input signal and the latched data is decoded subsequently by the digital decoder. The use of a THA in front of a flash A/D can alleviate the effects caused by delays between adjacent columns illustrated in the previous chapter. That is, the THA can hold the analog input at a constant level while the clock signal propagates through each column without allowing the analog signal to vary while the clock reaches each comparator. In other A/D topologies however, sequential processing takes place and consequently, the maintenance of a constant level at the input is required during the conversion process. Most A/D converters perform the conversion in what is called a "pipeline sequence". This means that a coarse approximation of the analog signal is performed first, followed by progressively finer approximations.

Figure 4 – 1 depicts the most general topology for an A/D converter. The sequence of events taking place is:

- coarse A/D converter performs conversion
- analog equivalent of the coarse converter is reconstructed with the help of the D/A
- error amplifier processes the difference INPUT – COARSE A/D result
- fine A/D conversion
- synchronization unit aligns the coarse and fine code (including digital error correction)
- digital output data presented to the converter' output bus.

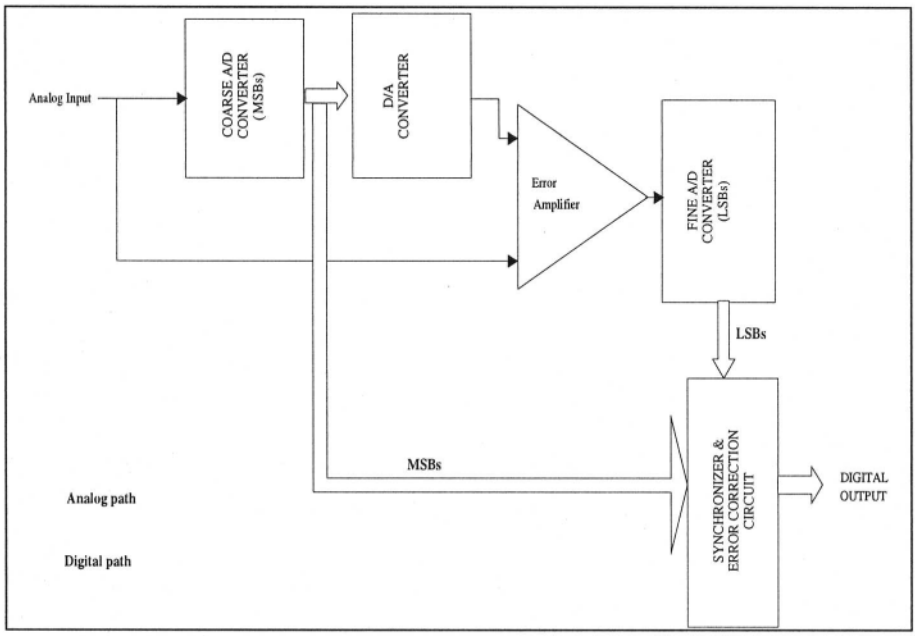


Figure 4-1. Subranging A/D topology

As shown in figure 4 – 1, the coarse digital code is converted into an analog signal after the coarse conversion so the difference between the coarse converter and the analog input can be found in the analog domain. The time required to process the coarse A/D, D/A and the required subtraction is the reason that a THA is needed. As explained above, in the absence of a THA by the time the coarse conversion is completed the analog input has changed resulting in a wrong difference between the analog input and the coarse converter. Employing the THA in front of the converter helps maintain the input signal constant to both the coarse and fine paths so the residue signal is correct at the instance that the error amplifier has settled and the fine conversion is ready to begin.

In order to investigate the THA behavior, we outline briefly the relation of input and output signals relative to the sample command in a THA in the time domain. This is illustrated in figure 4-2.

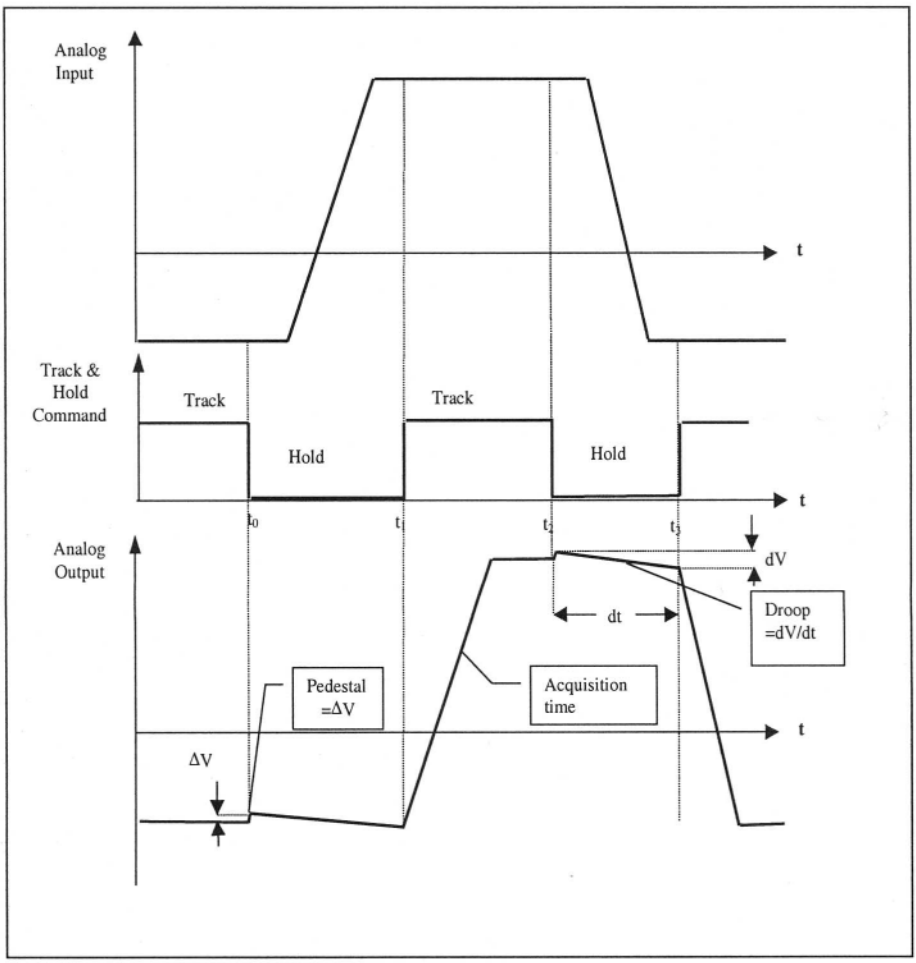


Figure 4-2. Input / Output signal relation relative to T/H command

The figure shows that the THA requires a certain amount of time for the output to reach its final value. This is called the **acquisition time** and is the time needed by the amplifier to settle given its bandwidth and slew-rate limitations. Some of the major errors characteristic to the sampling event are highlighted in figure 4-2. Figure 4-2 does not show the transients associated with the transition from one mode to the next (track-to-hold or hold-to-track mode). By and large these transients from one mode of operation to the next

are different from each other. The reason originates from the fact that the feedback factors around the amplifier are vastly different from each other in the two phases of operation. Let us assume that the amplifier used for the THA has an open loop gain $G(\omega)$. If the track mode has a feedback factor $\beta_T(\omega)$ in the track mode and a feedback factor $\beta_H(\omega)$ in the hold mode than the resulting gain-bandwidth products are $G(\omega)\cdot\beta_T(\omega)$ and $G(\omega)\cdot\beta_H(\omega)$ respectively. Generally $\beta_T(\omega) \neq \beta_H(\omega)$ resulting in different transients in track versus hold mode.

Mathematically, the sampling process of the THA can be described with the help of two step functions of reversed polarity, delayed by an amount of time equal to the hold time Th. Using $\Phi(t)$ to denote a step function in the time domain results in a time domain description of the THA as:

$$h(t) = \Phi(t) - \Phi(t - T_h) \qquad 4.1$$

This equation tells us that the THA output is equal to its input between time 0^+ and **Th** (the hold time) and is zero elsewhere. This equation assumes that the acquisition time is instantaneous and while this is never the case (it is physically impossible to acquire the signal in zero time) our approximation will suffice to illustrate the general THA behavior.

Figure 4 –3 (a) is a graphical representation of the equation above.

The Fourier transform for h(t), $H(\omega)$is shown in figure 4 – 3 (b) and figure 4 – 3 (c) (magnitude $M(\omega)$ and phase $\Theta(\omega)$ respectively).

$$H(\omega) = \frac{-i}{\omega} + \frac{i}{\omega}\cdot e^{-i\cdot\omega\cdot Th} = \frac{\sin(\frac{\omega\cdot Th}{2})}{(\frac{\omega\cdot Th}{2})}\cdot Th\cdot e^{(\frac{-i\cdot\omega\cdot Th}{2})} \qquad 4.2$$

In figure 4-3 the amplitude $M(\omega)$ and the frequency axis ω are normalized by Th. The magnitude $M(\omega)$ shows the familiar sin(x)/x with nulls occurring at intervals of $2\pi/Th$. If the THA had constant amplitude in the frequency domain, associated with a linear phase delay then the output signal would have been a scaled, undistorted replica of the input. Given however the "perfect" THA behavior described by the equations above we notice in figure 4 - 3 (b) that the sample-and-hold actually distorts the input signal. The magnitude response decays gradually and decreases toward zero at a frequency $\omega/Th = +/- 2\pi$ (at $\omega/Th = \pi$ the magnitude is reduced to 0.637·Th). The Bode plot of the THA is the response of a somewhat unusual filter. The magnitude response shows continuous ripples in the frequency domain up to considerable frequencies. The phase response of the THA

shown in figure 4 - 3 (c) is linear and has a slope of $-0.5 \cdot Th$ as a result of the factor $e^{-(i \cdot \omega \cdot Th)/2}$.

As explained above the overall effect of the sample-and-hold process is that the input signal gets distorted and the output deviates from the original form in both phase and magnitude. In this chapter we will examine the consequences of this filter with several examples.

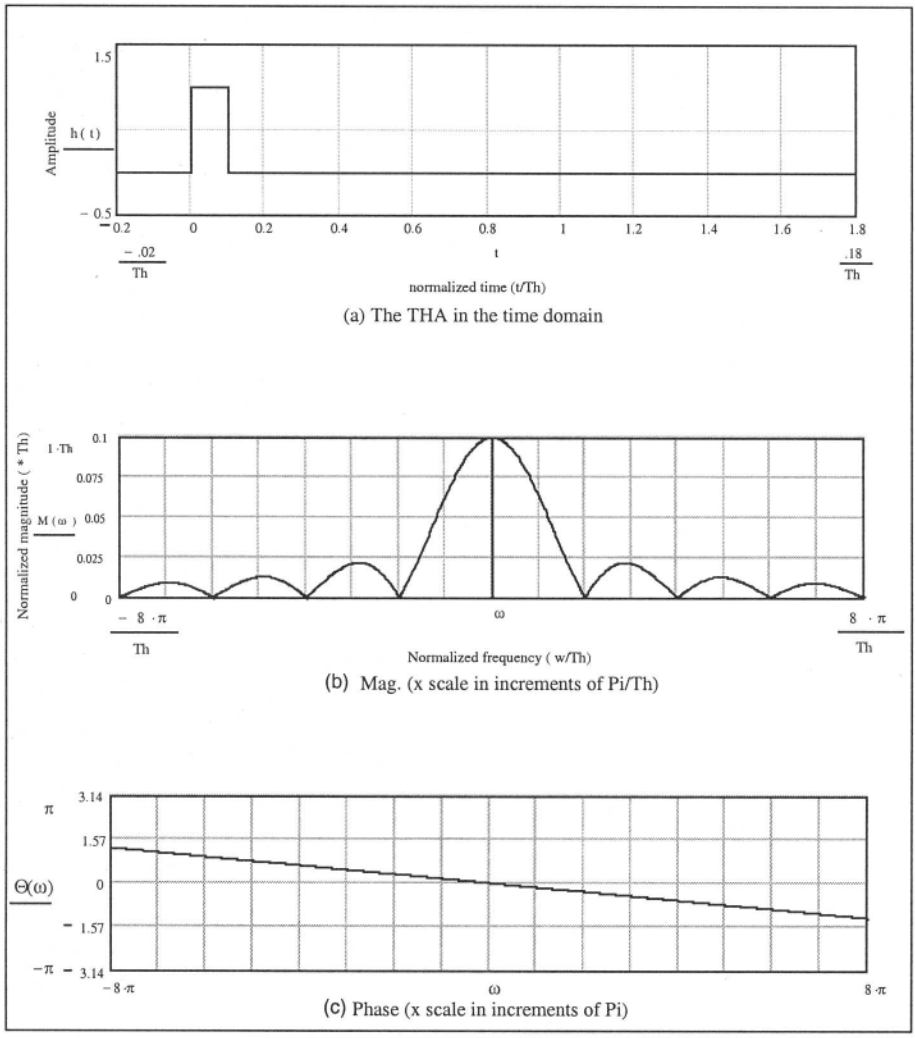


Figure 4-3. Mathematical representation of the Track & Hold function

Let us take a brief look at the THA and observe its frequency spectrum as a function of the number of samples taken on the output of the device. This

is the spectrum that we would observe when connecting the output of the THA to a spectrum analyzer. If the THA had a sinusoidal input and we were observing its output spectrum we would see a very complex picture. As we will show, the spectrum is a strong function of the number of time samples taken during each holding period of the THA. Figure 4 – 4 (a) – (c) represent time domain samples for sinusoidal signals sampled once, four and eight times respectively during each holding period. Figure 4-4 (a) is equivalent to applications where the THA precedes the A/D and the converter's input has ***one-and-only-one time record*** for its conversion. If multiple time samples are taken during the hold period (as would be the case when observing the THA output directly with the spectrum analyzer) than the time domain output would look like figures 4-4 (b) and (c). The frequency domain spectra for each of the conditions depicted in figure 4-4 are illustrated in figure 4-5.

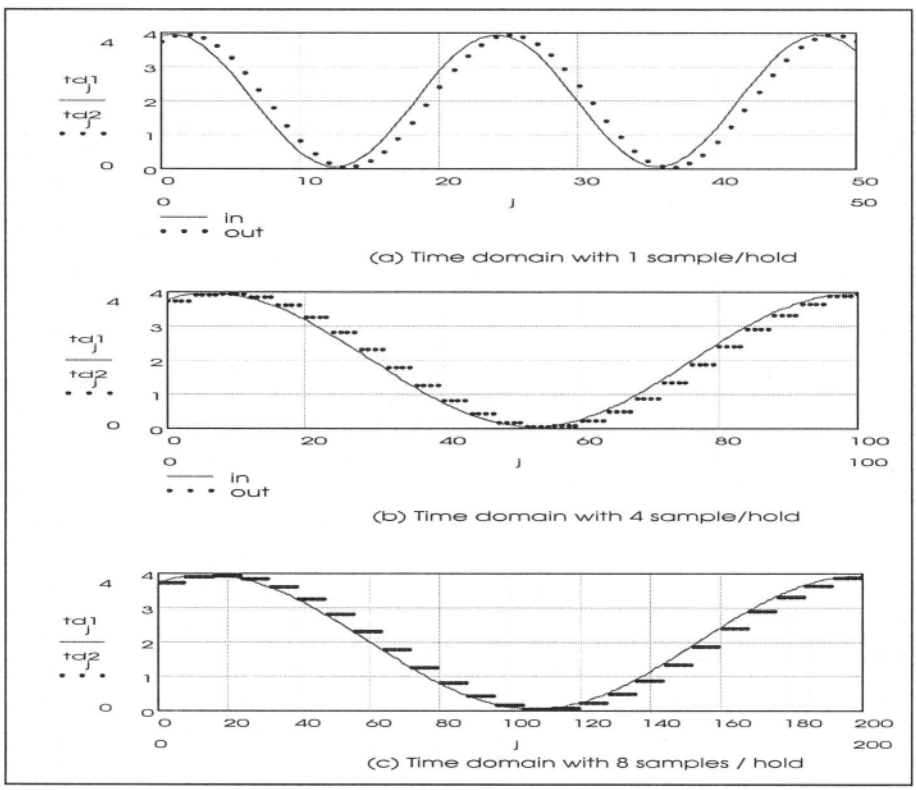


Figure 4-4. The effect of multiple sampling points during single holding period in the time domain

Why does the frequency spectra depend on the number of time records taken during the hold period? The answer is directly related to the sampling process and the fact that the sampling process itself inherently causes aliasing. With a single time point taken during the hold period the resulting output signal is an exact replica of the input signal but delayed by an amount equal to Th. If more data points are taken during each hold period, a deviation of the input sinusoid is seen at the output of the THA as shown in figure 4 – 4 (b) and (c). The changes in the spectrum are the result of having a constant voltage being held (represented by a straight horizontal line) followed by an abrupt change in the time domain to the new voltage being acquired on the next sample. As the number of sampled points is increased during the hold time the output signal approaches a staircase. This progression is clearly illustrated in figures 4 – 4 (b) and (c). The spectrum corresponding to these sudden changes has high frequency components and associated aliasing effects that are a function of the number of sampled points in each holding period. In figure 4 – 4 (b) the held signal is being sampled with four time points during each hold period. Now the distortion becomes more visible and it gets even more distorted in figure 4 – 4 (c) with eight time samples for each held period Th. In effect we are observing the byproduct of double sampling. First the THA samples at the frequency of 1/Th and then each voltage held is sampled multiple times at a higher frequency as shown in figures 4-4 and 4-5 (2, 4 and 8 times the original sampling rate of 25 MSPS used in our example). The spectrum seen with the spectrum analyzer is a similar combination between the sampling frequency of the THA (=1/Th) and the sampling frequency of the analyzer itself which in most cases is uncorrelated to that of the THA.

The associated FFT data, presented in figure 4-5 (a) - (c) shows the expected sin(x)/x shape and the aliasing effects. All data presented in figures 4-4 and 4-5 were obtained by using the simplest THA (presented in the next section) modeled with a simple switch, and a holding capacitor using an analog input frequency Fin = 1.07421875 MHz, sampled with Fs = 25 MHz. The frequencies are the result of using 256 data points for the FFT with one data point per sample. What figure 4 – 5(c) tells us is that our THA with an input frequency of 1.074 MHz and sample frequency of 25 MSPS when observed on the spectrum analyzer will have the following frequency components:

- 1.074 MHz (the original signal – in bin 11 of the FFT)
- 23.925 MHz (alias signal = 25 – 1.074 MHz – in bin = 256 – 11 = 245)
- all aliases in bins 267 (=256+11); 501 (= 256·2 – 11);523 (=256·2 + 11);757 (=256·3 – 11);779 (=256·3 + 11) and 1013 (=256·4–11).

In other words there are four sets **(= 8/2)** of aliased frequencies. In normal A/D applications, the THA samples the input data once per conversion and therefore this aliasing effect does not constitute a problem.

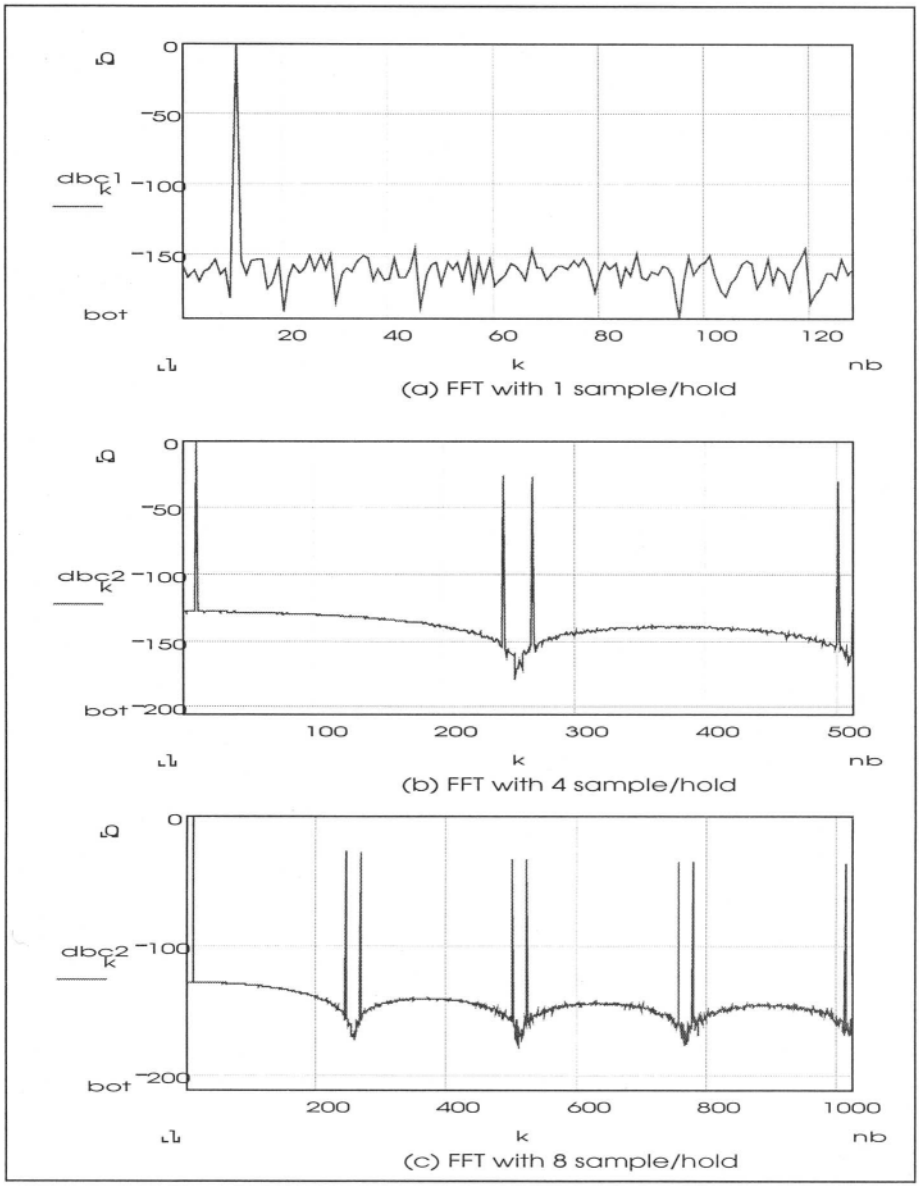


Figure 4-5. The effect of multiple sampling points during single hold period in the frequency domain

1.1 THA - the SPICE model

In the remaining part of this chapter we develop a simple SPICE model to help us understand the various error sources affecting the THA performance.

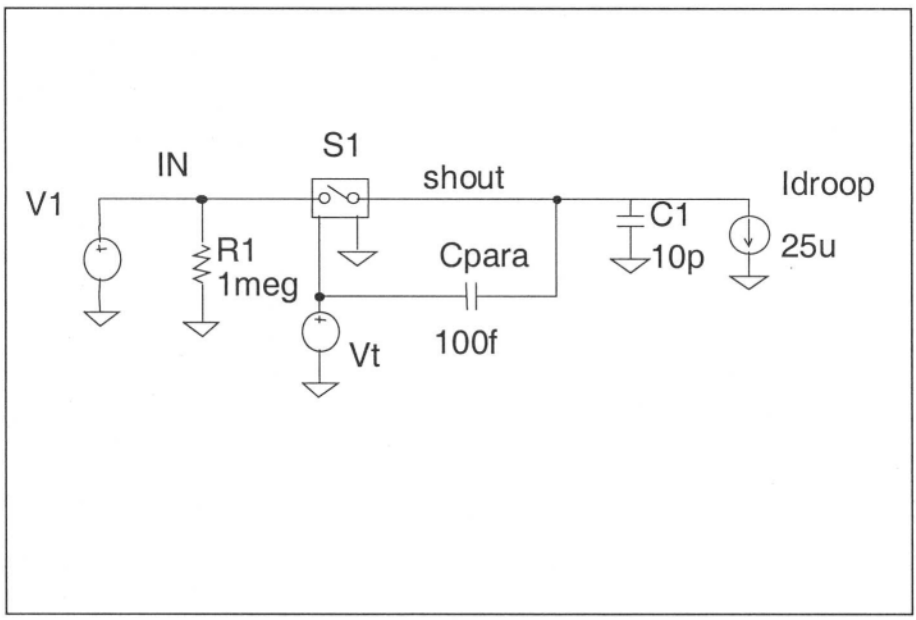


Figure 4-6. Simple THA Spice model

The basic SPICE model we use to model the THA is outlined in figure 4-6. The most relevant components of the figure are:

- a switch controlled by the Track&Hold voltage and
- a capacitor which is the storing element.

The THA operates as follows: when the Vth command takes place (Vth high) the switch closes allowing the capacitor C1 to charge to the input voltage. When the Vth voltage goes low, the switch opens and the holding capacitor maintains the stored charge – the voltage at the instance that the switched opened.

The remaining elements Cpara and Idroop of figure 4-6 represent parasitic effects of the THA.

- Cpara represents the unwanted capacitance that exists between the track and hold command signal and the holding capacitor C1. This parasitic

capacitance causes a charge error on the holding capacitor. The magnitude of the charge error is:

$$Q_{TOTAL} = \Delta V_{TH} \cdot \frac{C_{PARA} \cdot C_1}{C_{PARA} + C_1} \qquad 4.3$$

where ΔV_{TH} is the total swing of the control voltage (V_{TH})measured from the instance in time that the switch opens.
The resulting voltage error (also called pedestal) on the holding capacitor becomes therefore:

$$V_{Pedestal} = \frac{Q_{TOTAL}}{C_1} = \Delta V_{TH} \cdot \frac{C_{PARA}}{C_{PARA} + C_1} \qquad 4.4$$

This error is strictly the result of a voltage division between the holding capacitance and the parasitic capacitance **and is not a function of the voltage rate of change (dV/dt)**.

- The current source called I_{DROOP} in figure 4 - 6 results from loading effects. The load seen by the holding capacitor C1 causes the charge stored on the capacitor to decay over time. This causes an error relative to the initial voltage stored and its magnitude is a function of the length of time that the THA is in the hold mode:

$$V_{DROOP} = \frac{I_{DROOP}}{C_1} \cdot T_{HOLD} \qquad 4.5$$

The droop and the pedestal are two of the most basic errors noticed in THA applications. Neither droop nor pedestal errors cause significant errors when preceding the A/D converter as long as they behave in a linear fashion. For example if the droop current is fixed, the amount of droop is constant from sample to sample (as is usually the case for constant frequency sampling). The result at the output of the A/D is therefore equivalent to a constant offset in the conversion channel (assuming single time point during hold time).

The pedestal error is also equivalent to an offset error as long as the pedestal is invariant with respect to the input voltage. When the charge injection (or pedestal) becomes a function of the input signal however the distortion may become significant, as it will be shown shortly.
In general the errors caused by the sampling process in the A/D itself or the combination of THA and A/D can be grouped as:

- systematic – where for repeated samples the errors replicate and are predictable and
- stochastic – where given the uncertainty of the exact instance when the sample takes place the output signal has a random component.

The error sources mentioned above – droop and charge injection (pedestal) belong to the first category and their effects on the sampled signal can be analyzed intuitively given their predictable nature. By using the THA SPICE model suggested in figure 4 – 6 the reader can prove this statement. In the next few paragraphs we take a look at some parameters affecting the behavior of a THA in less predictable ways than charge injection and droop and we analyze the contribution of these parameters to the overall THA – A/D accuracy. Some of these less intuitive errors just mentioned are caused by the nonlinearity of the sampling switch, parasitic capacitance dependency on the input signal, feedthrough across the switch, variation of the rise/fall time of the hold command and jitter. We will take a brief look at those effects with the help of several examples.

1.2 Errors affecting THA accuracy

1.2.1 Holding capacitor voltage-coefficient

One of the error sources affecting the accuracy of the THA is the dependency of holding capacitor value upon the voltage being held. When capacitors are manufactured in an IC process they exhibit a voltage coefficient due to space charge generated between the electrodes. The plates of the capacitor in an I.C. process are metal-metal, metal-poly or poly-poly materials and are insulated from each other through a thin layer of oxide. The quality of the insulator bears the responsibility for the capacitance variation as a function of stored charge. When the plates are poly-poly materials they exhibit a voltage coefficient of approximately 100 PPM/V or less if both plates are equally doped. If the coefficient is positive the voltage dependency is dominated by accumulation at the surface. In contrast, a negative voltage coefficient indicates a dependency on depletion due to lightly doped plates. For high accuracy converters the voltage coefficient of the capacitor needs to be very low in order to obtain a high degree of linearity and thus low distortion. For example in a 12-bit converter 1 LSB represents one part in 4096 or 244 PPM related to the device's full-scale range. If the holding capacitor varies due the voltage level held then the THA will distort the input causing both static and dynamic errors.

To model this effect we use the ability of SPICE to model the capacitor variation by using a polynomial dependency. In our example we define the holding capacitance using the following equation:

$$C(V) = C_0 + C_1 \cdot V + C_2 \cdot V^2 \qquad 4.6$$

where C_0=10pF, C_1=10 PPM/V and C_2=10 PPM/V^2. Using the circuit of figure 4-6 we evaluate the dynamic performance of the THA. This is illustrated in figure 4-7.

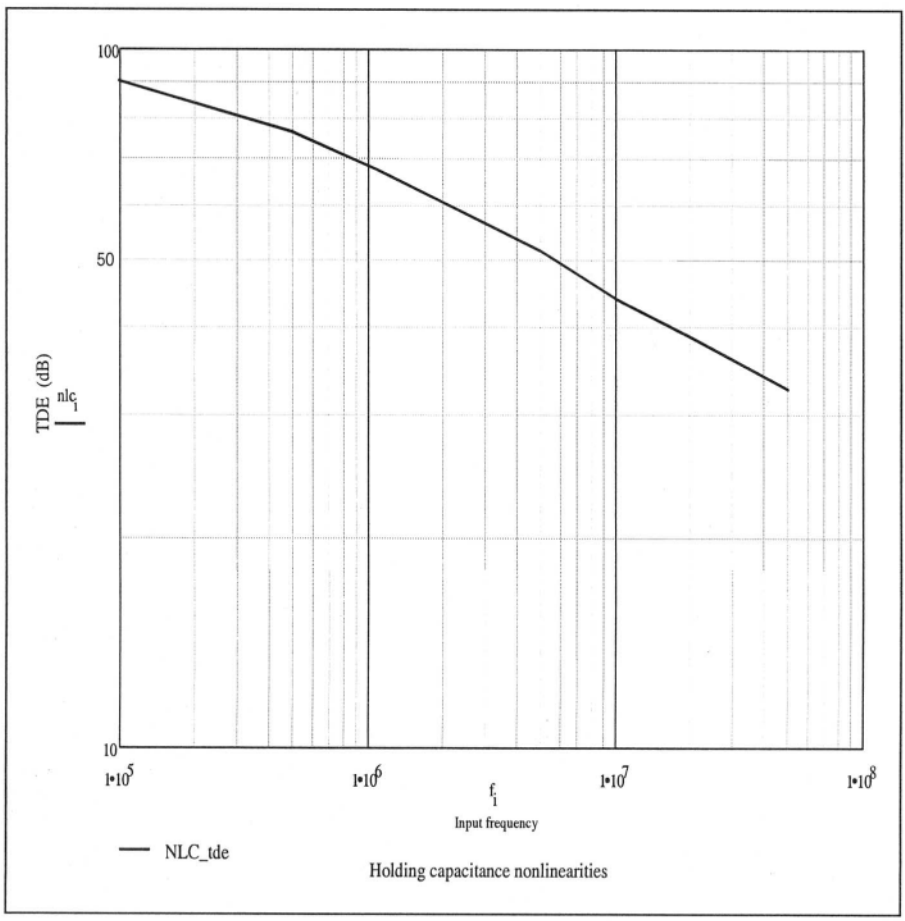


Figure 4-7. Dynamic error caused by voltage coefficient of the holding capacitor

Figure 4 – 7 behavior should not be surprising. As the voltage across the holding capacitor is increased, its capacitance increases governed by the

positive voltage coefficient in the equation above. This causes the THA output to be distorted. The switch resistance and the holding capacitor in effect constitute a passive R-C filter. If the capacitance were constant the filter would have a constant pole frequency. In our case however, the capacitance has components that depend on the voltage across its terminals. The result is a variable capacitance governed by the voltage and therefore the pole frequency of the filter varies as a function of the voltage being held. As the held voltage increases the capacitance increases causing the filter to have a lower frequency pole. Since the effect is not symmetrical with input voltage the output will have mostly even order harmonics. For example, using our SPICE simulation model with the nonlinear capacitor of C(V) equation, a sinusoidal input with a frequency of 4.98 MHz and a sampling frequency of 25 MSPS results in second harmonic of -55.86 dB. A THA using a linear holding capacitor under the same conditions produces second harmonic of -82.7 dB (a difference of almost 30 dB). The third harmonic however for these examples are -73.8 dB and -79.7 dB respectively.

1.2.2 Switch resistance modulation

Next we consider the switch resistance modulation effects on the THA dynamic performance. In a CMOS process the switch is usually implemented with the help of a transfer gate (a PMOS device in parallel with an NMOS). Intuitively the two CMOS devices should have large W/L ratio such that a relatively low switch resistance is obtained. This helps shorten the acquisition time controlled by the time constant $R_{ON} \cdot C1$ (where R_{ON} is the "ON" resistance of the switch). A large device area is associated however with large stray capacitance. The tradeoff is between a reduced resistance (large geometry) and the associated parasitic capacitance (which is also directly proportional to the gate area). Of course increasing the parasitic capacitance affects the $R_{ON} \cdot C1$ pole as well as the pedestal error.

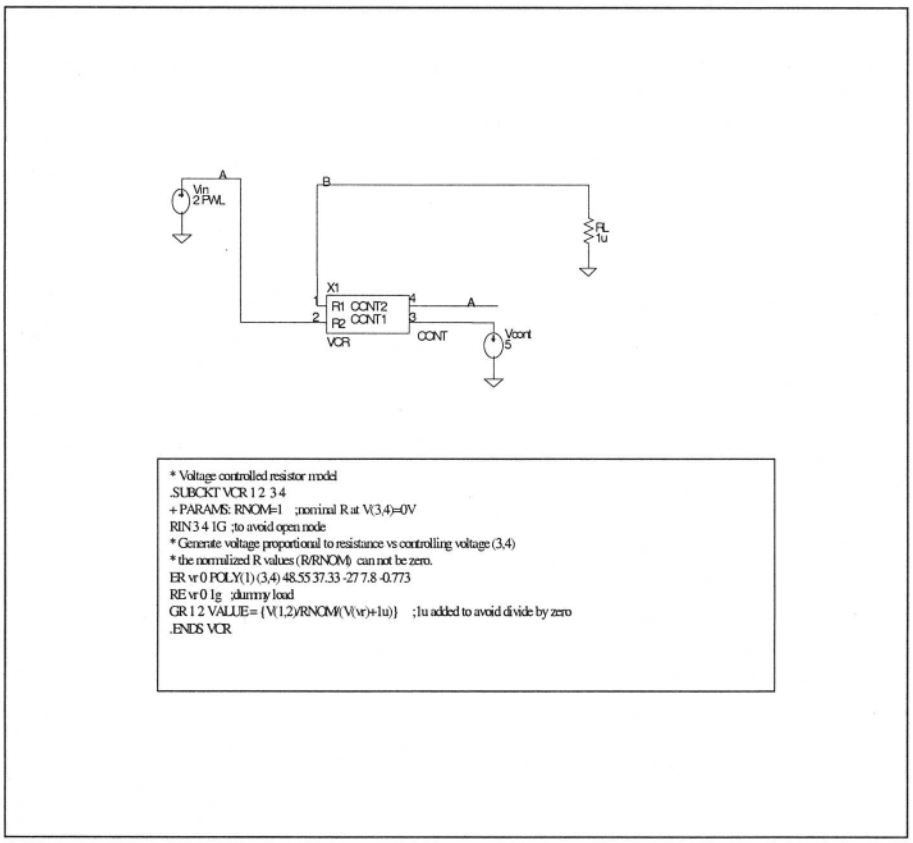
Unfortunately any practical circuit using CMOS transfer gates is affected by switch resistance modulation associated with the input voltage. This is caused by the fact that R_{ON} is a function of V_{GS} of the transistors in the switch:

$$R_{ON} = \left[\mu \cdot C_{OX} \cdot \frac{W}{L} \cdot (V_{T/H} - V_{IN} - V_{Th}) \right]^{-1} \qquad 4.7$$

where μ is the mobility, C_{OX} is the gate oxide capacitance, W and L are the transistor width and length, $V_{T/H}$ is the control voltage of the track/hold signal and V_{Th} is the transistor threshold voltage.

Since this resistance changes with input voltage (it controlled by the source to gate voltage) the track time constant is affected as well. To model this effect we use the behavioral model capability of SPICE with a polynomial approximation.

Modeling the nonlinear resistance characteristic of the switch is by no means a trivial task. In fact, most SPICE simulators do not supply a nonlinear resistor model. A logical approach of accomplishing this task is suggested by P.W. Tuinenga (55). To model a nonlinear resistance Tuninga suggests using a voltage-controlled resistor VCR as illustrated in figure 4-8.




```
* Voltage controlled resistor model
.SUBCKT VCR 1 2 3 4
+ PARAMS: RNOM=1   ;nominal R at V(3,4)=0V
RIN 3 4 1G ;to avoid open node
* Generate voltage proportional to resistance vs controlling voltage (3,4)
* the normalized R values (R/RNOM) can not be zero.
ER vr 0 POLY(1) (3,4) 48.55 37.33 -27 7.8 -0.773
RE vr 0 1g  ;dummy load
GR 1 2 VALUE = {V(1,2)/RNOM/(V(\vr)+1u)}   ;1u added to avoid divide by zero
.ENDS VCR
```

Figure 4-8. Switch resistance modulation - Spice model

The idea behind the implementation of the nonlinear resistor is to obtain a floating voltage-controlled resistance called VCR. This device generates a voltage between terminals 1 and 2 which is proportional to the controlling voltage (nodes 3 and 4).

We illustrate a practical case by using curve fitting to model the transfer gate implemented in a 1μ CMOS process. The curve fitting results in the following approximation:

$$P(V_{CONTROL}) = 48.55 + 37.33 \cdot V_{CONTROL} - 27 \cdot V_{CONTROL}^2 + \\ + 7.8 \cdot V_{CONTROL}^3 - 0.733 \cdot V_{CONTROL}^4$$

Figure 4-9 is the graphical illustration of this equation and it shows the nonlinear behavior of the switch as a function of the control voltage - $V_{control}$ and the analog input voltage - V_{in}. The resistance of the switch shown is the combined resistance of a PMOS in parallel with the NMOS device. As the V_{gs} voltage of the N-channel increase its resistance is lowered while the reverse occurs for the P-channel device. The result is a resistance that has three regions: an almost linear region, followed by a parabolic function and finally a linear decrease in resistance.

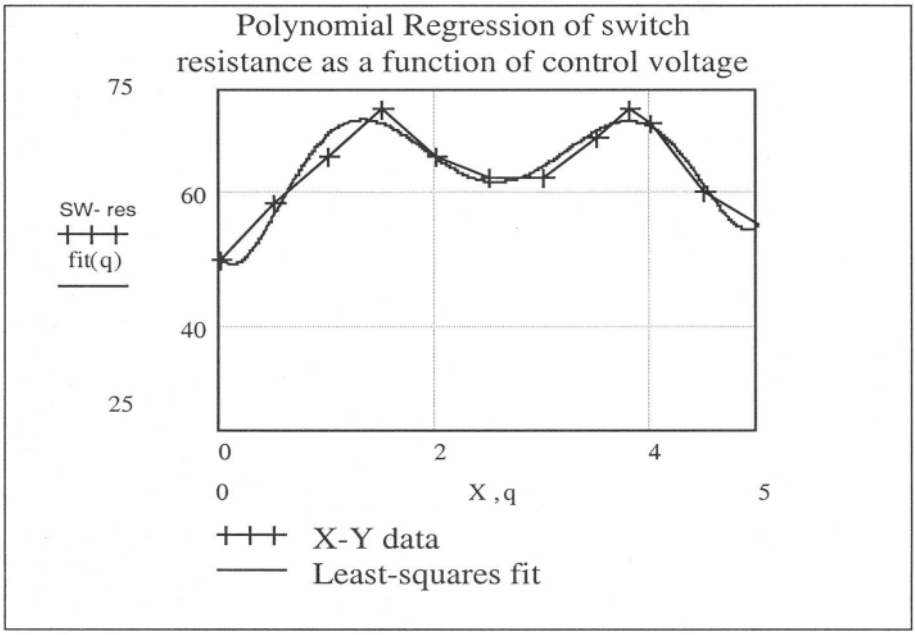


Figure 4-9. Nonlinear switch resistance as observed in CMOS transfer gates

The effect of resistance modulation of the switch on the THA dynamic performance is simulated next using the circuit shown in figure 4 - 10.

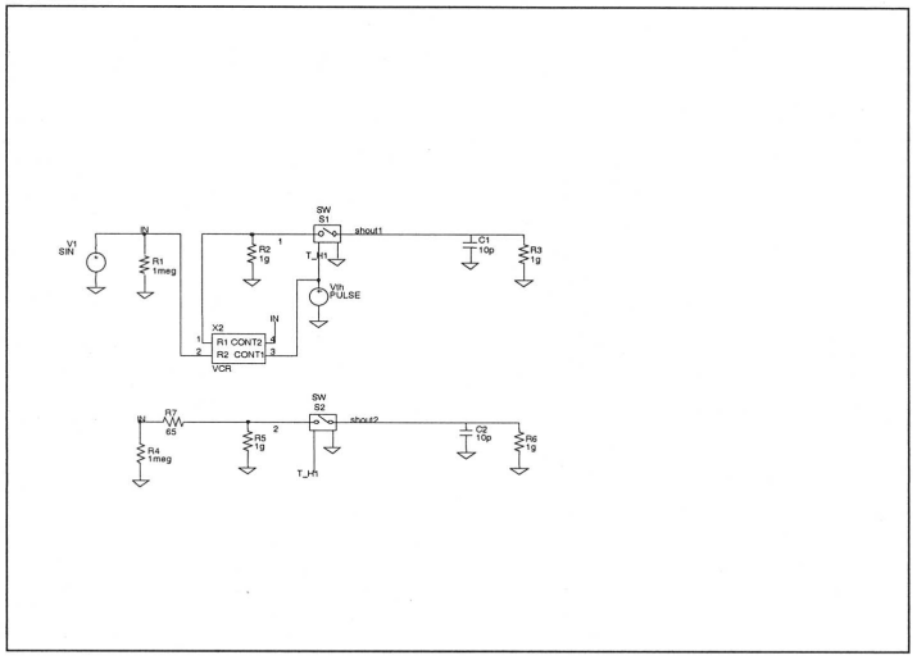


Figure 4-10. Spice schematic for simulation of switch resistance modulation

The figure shows two THA channels:

- top channel using a nonlinear switch resistance and
- bottom channel with a linear resistance having the equivalent RMS value of the nonlinear resistance above (in our example an effective resistance of 65 ohms).

In both channels the switch is modeled as a series combination of a voltage controlled 1 Ω resistor (when the switch is "ON") and voltage controlled resistor (VCR). In the linear channel the resistor R7 replaces the VCR. The VCR has its resistance controlled by the voltage difference Vth – Vin (similar to the gate to source voltage controlling the resistance of the FETs). VCR and R7 together with the associated holding capacitors C1 and C2 control the frequency response of the circuit since these resistors are in series with SW1 and SW2. The resistors VCR and R7 are dominant relative to the switch resistance.

The transient analysis is performed at various input frequencies on the two channels to monitor the differences between a nonmodulated switch resistance and the modulated counterpart in figure 4 - 11.

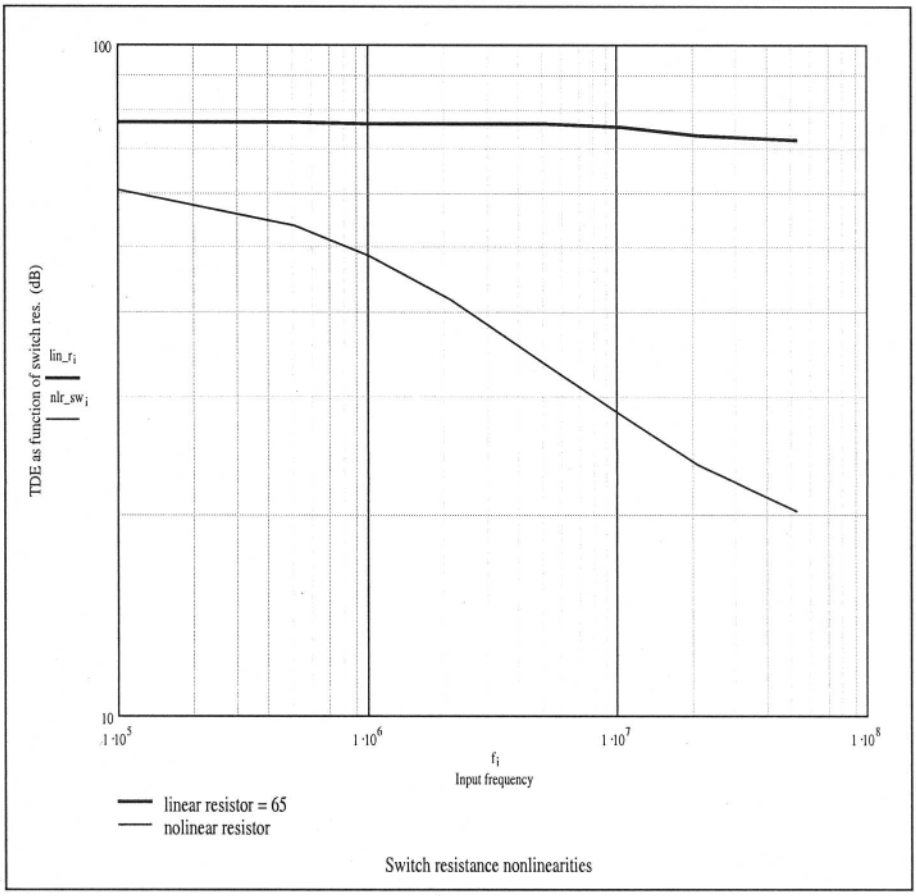


Figure 4-11. TDE of linear vs. nonlinear switch resistance in a THA

Looking at figure 4-11 we observe that the linear resistor outperforms the modulated switch, as expected. Since the circuit at the bottom of figure 4-8 is linear (linear resistor and capacitor) we anticipate that the sampling process will generate no harmonics. The linear resistor with its associated holding capacitor behaves as single pole filter resulting in a linear behavior. On the other hand, the top channel with its nonlinear resistor behaves as a filter with variable frequency response, whose pole location depends on the voltage across the resistor. Consequently, with a low voltage across the resistor, its resistance is low resulting in a high frequency pole. As the voltage across the resistor increases, the pole frequency is decreased. Since the pole frequency is a function of input voltage the overall result is a distortion in the output signal dependent on the input amplitude. Therefore a

low-level input signal passes through a higher frequency filter than a high-level signal approaching the device full-scale–range. A sinewave input signal is therefore going to be more "squashed" on the top of the sinusoid than on the bottom resulting in even harmonics (nonsymmetric) at the output.

Figure 4 - 11 confirms the above explanation but it also shows a slight decrease in dynamic performance at high input frequencies for the linear device. The reason for this behavior is the numerical noise in the simulation, which raises the noise floor. To alleviate this problem a reduction in the simulation time step can be attempted at the cost of increased simulation time. A reasonable rule of thumb is to use a time step ceiling of one tenth of the simulation time step. Of course if finer detail is essential and the required numerical noise needs to be reduced finer time granularity will be needed. Changing the SPICE option card can prove also useful (reducing CHGTOL, TRTOL, or varying the integration method from TRAP to GEAR). More on simulator options can be found in appendix B.

1.2.3 Hold mode feedthrough

Another systematic error in the THA is caused by the input signal being fed across the switch when the THA is in its hold mode. The main reason for the feedthrough is caused by the stray capacitance across the switch. Usually, this capacitance is attributed to fringe effects in the layout and by C_{DS} – drain-source capacitance - of the transistors employed in the transfer gate. The **feedthrough** modeled with a capacitor across the switch **will not cause any nonlinear behavior (harmonic distortion)** but only gain error. This behavior is not intuitive and an explanation is in order.

While in the hold mode, in the presence of a feedthrough capacitance, the output has a component produced by the input. Suppose that the input signal is a sinusoid of a known frequency - F_{in}. At the THA output a signal of the same frequency will be seen due to the capacitor divider (C_1 and C_{FT}).

The sampling process in itself creates new tones due to the sampling frequency - F_s. These tones are observed at multiples of $F_s \pm k \cdot F_{in}$ (where k is an integer). Since the information of interest is contained at the tone F_{in}, the tone resulting from the capacitance divider and the sampling process are only affected by the addition of this divider. A graphical representation is illustrated in figure 4-12.

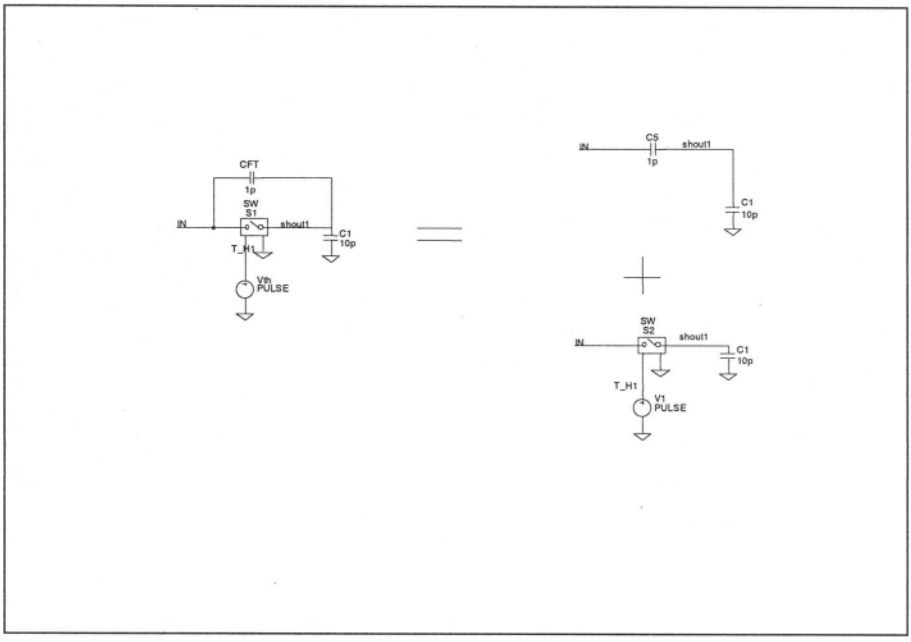


Figure 4-12. Feedthrough error is the superposition of a capacitor divider and a perfect sampling switch

The feedthrough error causes therefore a systematic error whose only relevant component is gain error. No harmonics result however from hold feedthrough.

1.2.4 Sampling instant distortion

Finally we examine a systematic error caused by the rate of change of the hold command. The finite transition time of the hold command – t_r , t_f - creates an input dependent distortion of the sampled voltage. The distortion is caused by the fact that the switch closure instance depends on the voltage difference between the switch' gate and its source. The source is the input voltage while the gate is usually driven by an independent voltage (clock driver). If we assume for the purpose of the analysis that the switch has 0V threshold then the sampling time t_S is defined as the time when the falling clock reaches the voltage V_{AV} (see figure 4-13).

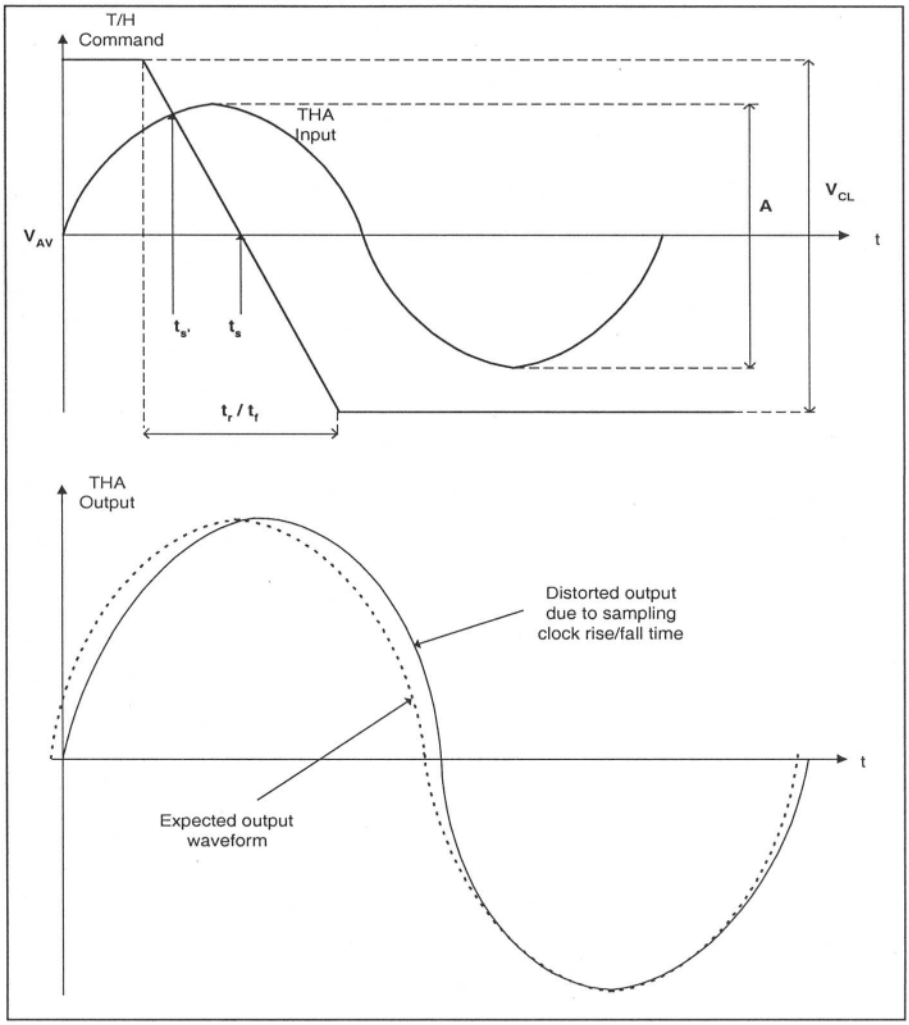


Figure 4-13. Sampling instance distortion due to slow rise / fall time of the T / H command

If the input signal is equal to V_{AV} then the sampling instance is t_S. Otherwise if the input is different than V_{AV} the sampling instance is different – t_S'. This results in a distortion that for a sinusoidal input can be calculated (ref. 71):

$$THD = -20 \cdot \log\left(\frac{V_{CL}}{A \cdot f \cdot t_f}\right) \qquad 4.8$$

where A is the analog input swing, f is the input frequency and t_f is the clock fall time.

To examine this distortion we use the circuit in figure 4-14. Here again we use to channels: one channel simulating the sampling instance distortion and the second channel free of distortion.

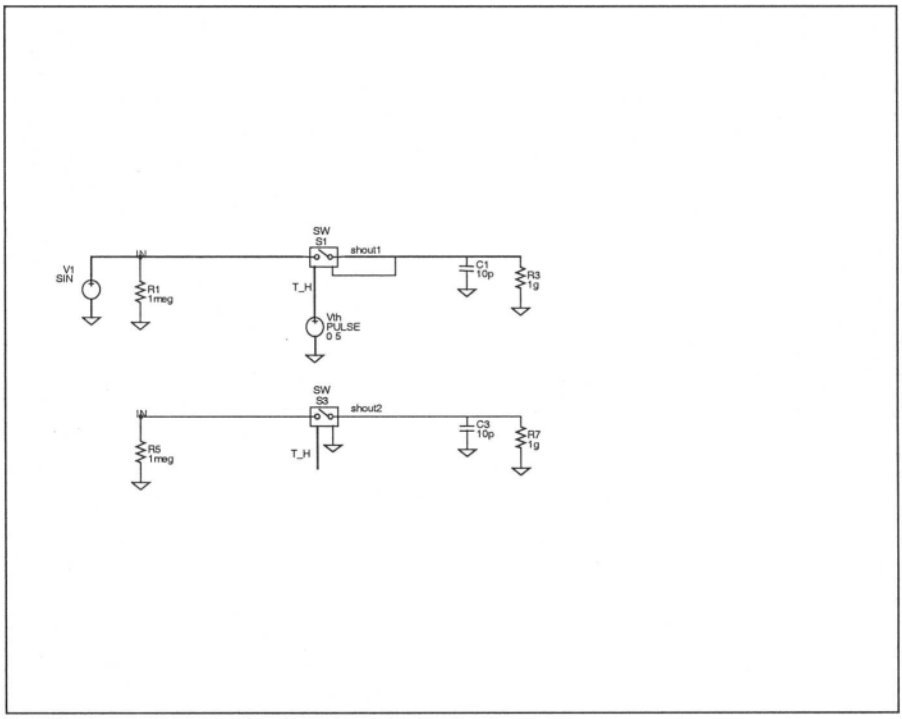


Figure 4-14. Spice diagram for sampling instance dependence

In our model the switch' gate is driven by a voltage source $V_{T/H}$ and its source is connected to the input. The switch is modeled so it opens when the voltage is 0V and turns "ON" when the voltage is 5mV. By performing an FFT analysis on the output voltage "SHOUT1" we obtain the THD as illustrated in figure 4-15. In the analysis used to obtain figure 4-15 the analog input voltage has a swing of 4V, the clock has a swing of 5V and the analog input frequency is ≈ **48.83 kHz.** The clock transition times vary from 1ns to 15 ns keeping the duty cycle at 50%.

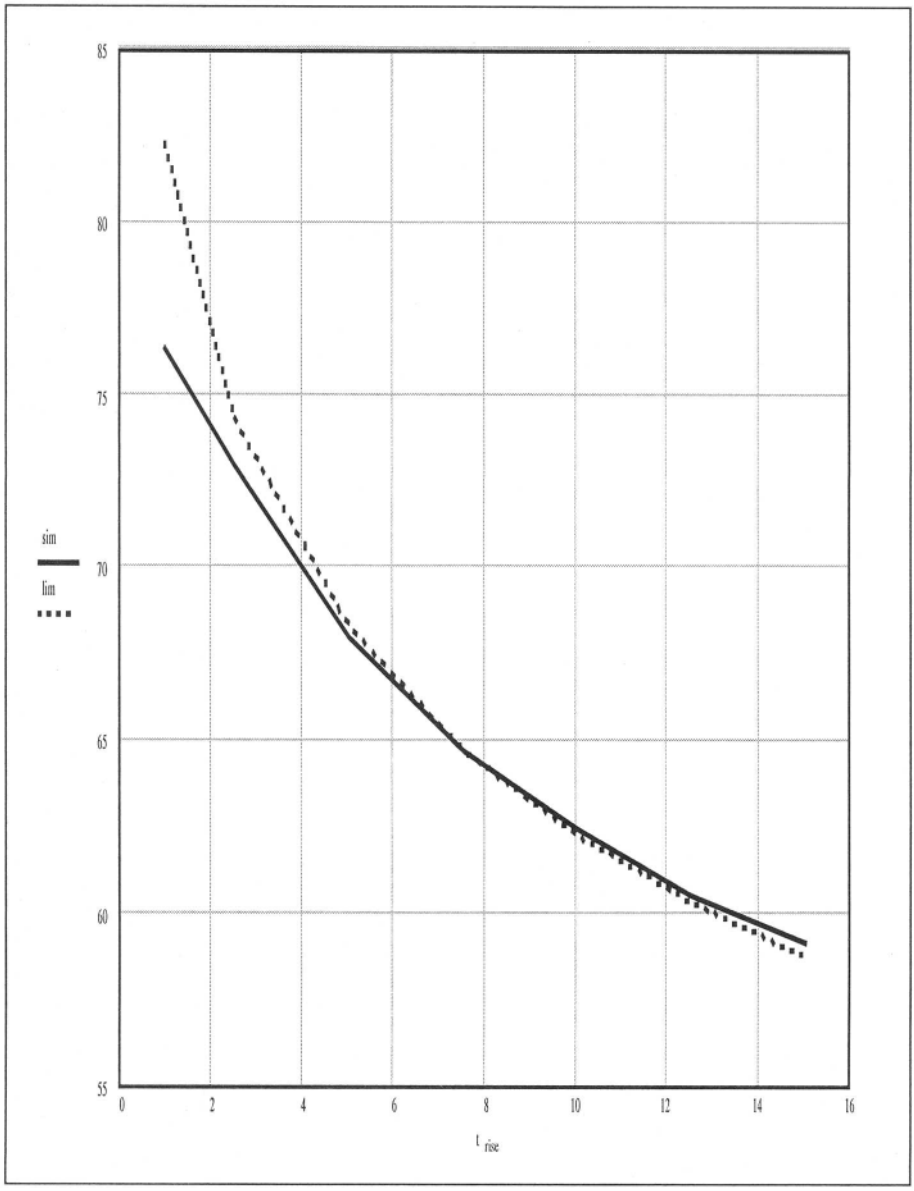


Figure 4-15. Sampling instance effects on TDE solid line simulation, dotted line Lim's equation

Figure 4 – 15 shows a very good correlation to the equation above.

1.2.5 Hold command jitter

In the next section we examine a stochastic error in the THA holding command called jitter. The effects of jitter of the Track/Hold command relative to the input signal are examined by the introduction of a noise source in series to the Vth signal of the circuit as shown in figure 4 - 16. As in the previous case, we perform the spice simulations on two channels: one channel with no errors and the other with jitter related errors. To simulate the jitter, we use a voltage source called V_{NOISE} with its parameters defined in a file called V.DAT. The V.DAT file contains random gaussian noise with a mean of zero and a standard deviation (σ) of 5mV.

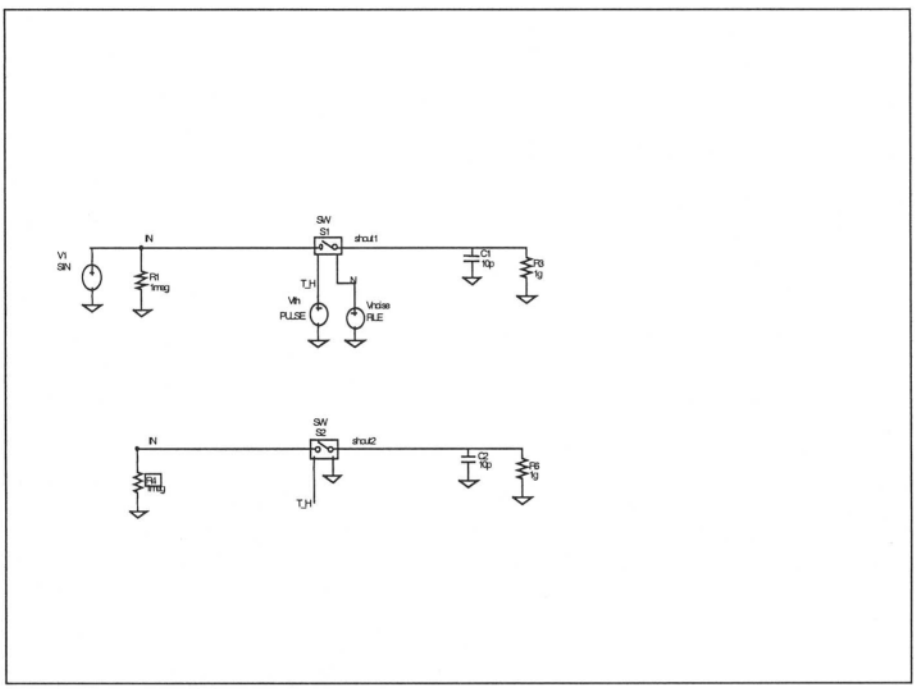


Figure 4-16. Spice schematic used for the simulation of jitter in a THA

The switch closes when the voltage between its control terminals reaches 2V and opens when the voltage is 1.95V. Since the switch activation is a function of the voltage difference between Vth and the noise source we observe the jitter (noise effects) on the actual sampling instance. A graphical representation of the jitter effects is outlined in figure 4 - 17 in the time domain.

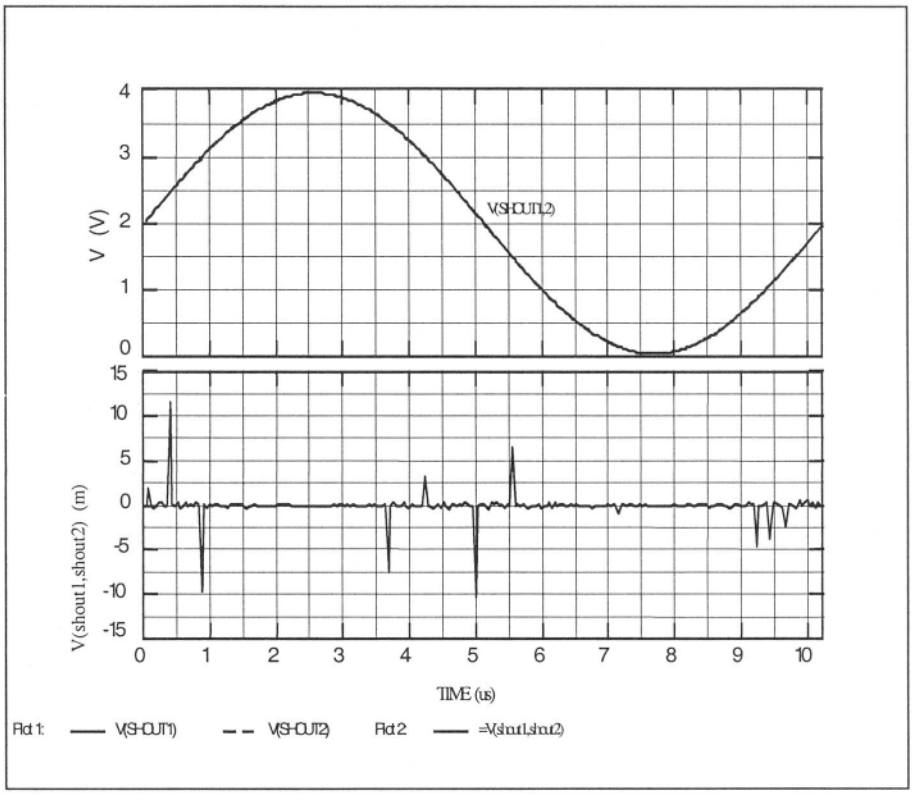


Figure 4-17. Distortion caused by jitter in a THA

The top plot in figure 4 - 17 shows the output voltage of THA for a 4 V p-p signal. The bottom plot is the difference between the two output nodes of THA1 and THA2 (with and without jitter). The difference plot clearly shows that as the input slew rate increases so does the jitter effect. This increase in slew-rate occurs as expected at the sinusoid crossover points. Due to this increase in slew-rate the noise effects are also more predominant at the crossovers.

Simulations for various input frequencies are summarized in figure 4 - 18. Again, with increased input frequency the input slew rate is increased resulting in higher dynamic error as a function of jitter.

Figure 4 - 18 illustrates the same behavior as mentioned in chapter 1 (figure 1 – 5 and 1 - 6). As outlined in figure 1 - 6, our simulations show the same decrease of -20 dB/decade in dynamic performance as a function of input frequency.

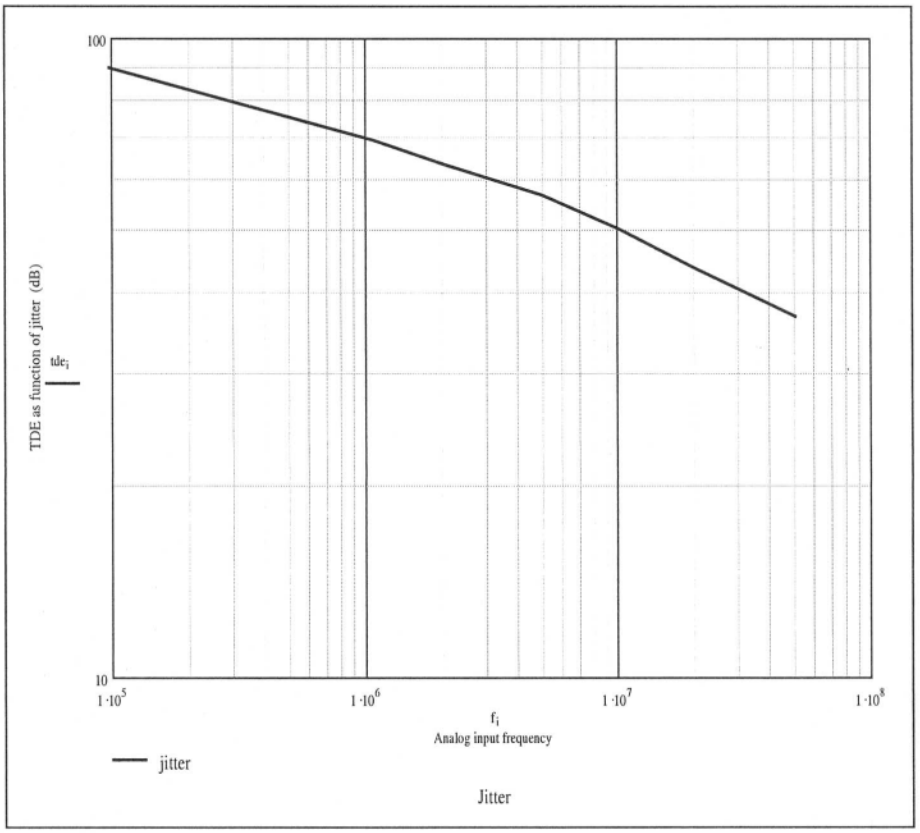


Figure 4-18. TDE in a THA as a function of jitter

SUMMARY

- The simplest model for a track and hold amplifier is a switch and a capacitor. Both the capacitor and/or the switch resistance nonlinearities can be modeled in SPICE with the help of polynomial approximations as shown in this chapter.

- The nonlinearities contribute usually to even harmonics at the THA output.

- Similarly the sampling time instant dependency on the fall time of the T/H command causes nonsymmetric distortions resulting therefore in even harmonics.

- The jitter of the sampling command reduces the THA dynamic performance as a function of the input frequency as shown also in chapter 1.

- The pedestal (charge injection) and feedthrough do not contribute to distortions in a THA when the input signal is sampled only once per hold period.

PROBLEMS

1. Using the SPICE model of figure 4 – 6 find the amount of charge injection with C_{PARA} = 100 FF. By changing the sampling frequency Fs prove that with a constant parasitic capacitance C_{PARA} no distortions effects are observed as a function of frequency.

2. Repeat problem 1 with C_{PARA} = 0 and I_{DROOP} = 10 μA. Find the amount of droop when the hold time is 10 nsec. Perform a frequency analysis varying the sampling frequency Fs and verify that the droop does not cause dynamic errors.

3. Using the SPICE model of figure 4 – 6 connect a feedthrough capacitor of 1 pF across the sampling switch. Perform a transient analysis while sampling the THA at 25 MSPS and vary the input frequency at several frequencies to confirm that the dynamic errors of the THA are not affected by the feedthrough component.

4. Using the SPICE model of figure 4 – 14 change the threshold voltage of the switch to 1V; change the input signal to a swing of 1V and graph the THD as a function of rise/fall time as in figure 4 – 15.

Chapter 5

SAR A/D

1. INTRODUCTION

The Successive Approximation Register (SAR) A/D is probably the most widely used converter in industrial control applications. Its popularity stems from the good ratio of speed/power and the fact that the converter is very compact making it an inexpensive device.

The SAR A/D operation is based on the binary search algorithm. The algorithm is akin to a name search in a telephone book. Without knowing the page that contains the name of interest first, you open the book about midway. If the name is located in the first half of the book then you split the number of pages in half looking in the first or second quarter of the book. As the search advances you, keep halving the remaining number of pages until you reach the relevant page.

The SAR topology requires a single comparator, one D/A and a successive approximation register with an associated digital accumulator. To understand the operations of this converter see figure 5-1a:

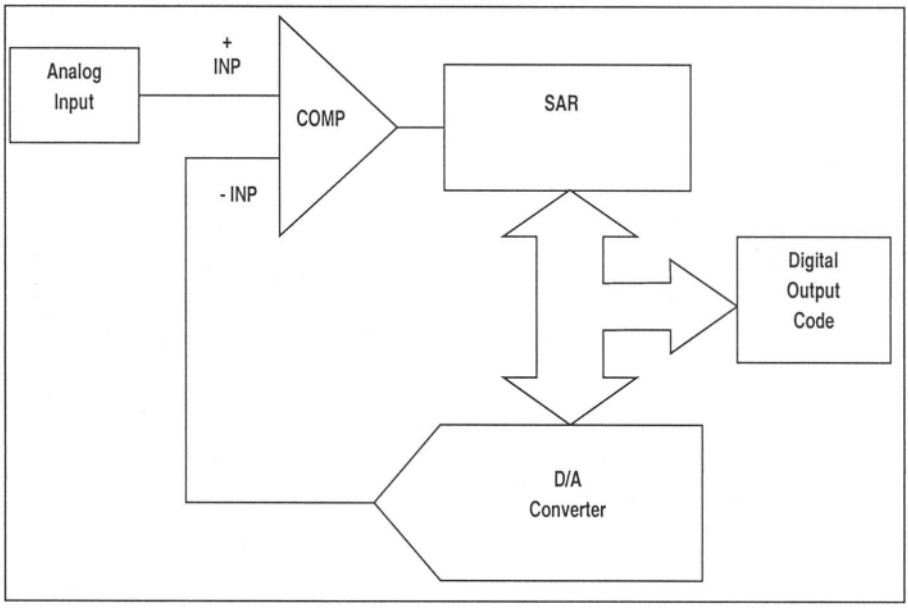


Figure 5-12a. SAR A/D converter - voltage comparison

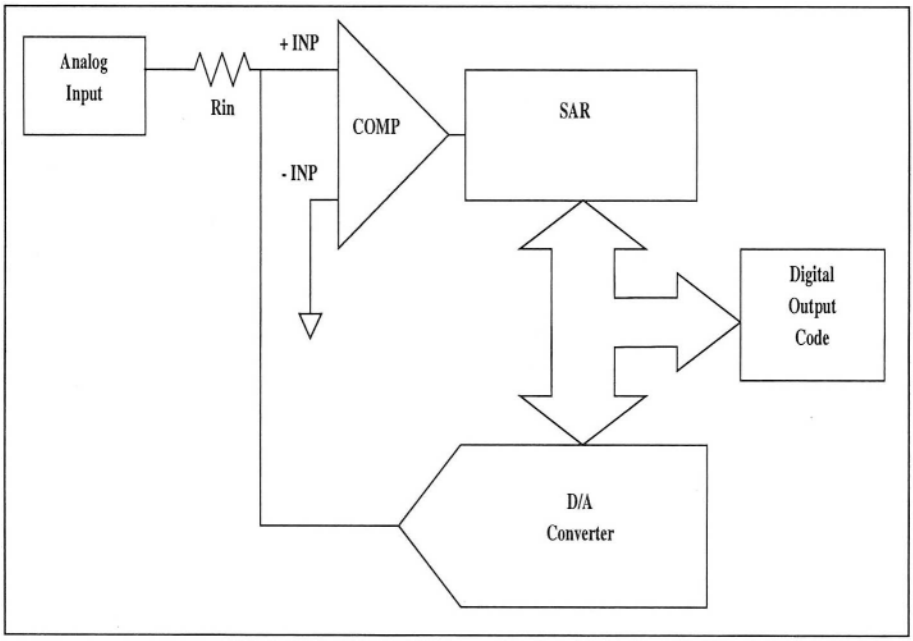


Figure 5-1b. SAR A/D converter - current approach

As the conversion is initiated, the SAR sets the MSB to "1" and the rest of the bits to "0". This causes the DAC to be set at half scale of the converter's range. If the analog input voltage is higher than the MSB's weight of the D/A, the comparator output is set to a "1", the register retains the MSB setting and proceeds with the trial of the next significant bit (B2). The comparator responds again with a "1" if the input voltage is higher than the D/A output voltage or "0" if the reverse is true. Next B3 is tried and the comparator reacts in a similar fashion described for the previous two (more significant) bits. The search continues until the voltage of the D/A converter reaches the analog input to within its specified accuracy.

In effect, the control loop formed by the comparator, SAR, and D/A performs an integration by using the accumulator of the SAR. During the integration process, the SAR continues to accumulate the bit weights until the digital output code represents the best approximation of the analog input. The integration gradient is positive if the analog input is higher than the output of the D/A (comparator output is high) and negative if it is lower (comparator output is low).

As indicated above, the comparison between the analog input and the accumulator of the SAR is performed in two phases:

- a trial phase when the bit of the D/A is compared against the analog input and
- a decision phase: if the output of the comparator is high then the particular bit is set to "1" in the SAR. If the comparator output is low, then the bit that has just been tried is reset to "0" and the next trial begins by setting the following significant bit to "1".

A comparison between the SAR and flash converters reveals the reason for the differences in conversion speed. While the flash converter performs a simultaneous comparison for all bits, in the SAR converter the bits are tested in a serial manner with the MSB first and LSB last in the conversion sequence.

The SAR topology can be accomplished in one of two methods:

- the configuration presented in figure 5-1a which represents a voltage comparison or
- the current approach shown in figure 5-1 b.

In the current approach, the input voltage is converted to a current with the help of resistor Rin. The maximum input current is obtained when the analog input reaches the full scale of the converter. The current is then equal to V_{FS}/Rin. For a current-output D/A the full-scale current has the same value. When the input full-scale voltage is reached, the D/A converter needs

to sink an equal amount of current resulting in a comparator input of zero volt.
Considering the Thevenin equivalent at the comparator input we notice that the comparator decision is based on the voltage $(I_{in} - I_{DAC}) \cdot Rin$.

For a current A/D implementation we use current steering methods where the current source is switched towards the output of the D/A or towards the ground as illustrated in figure 5-2.

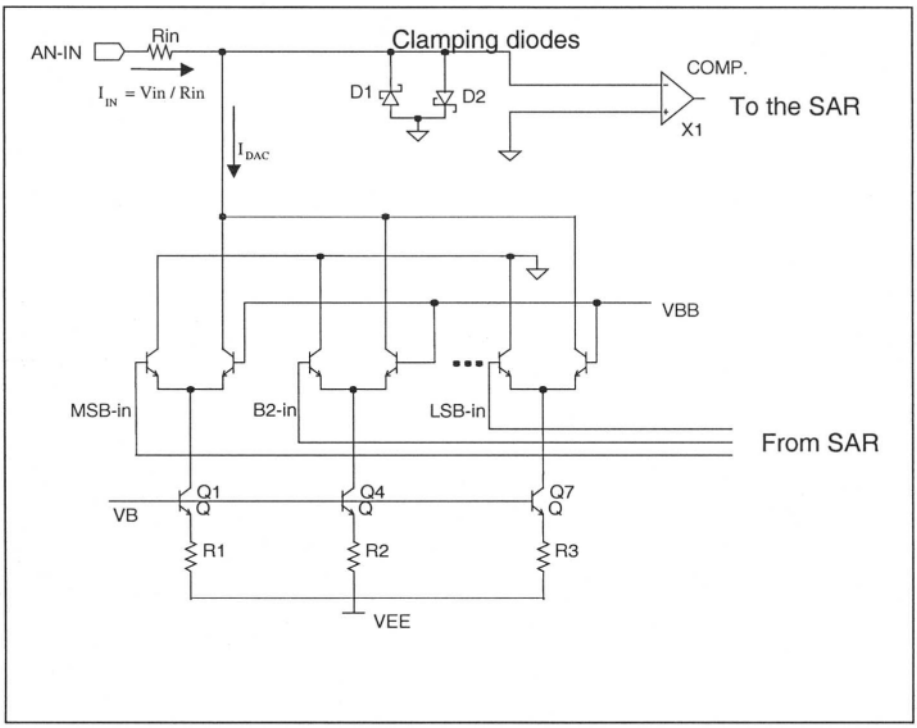


Figure 5-2. Current input SAR A/D

The current and voltage comparison are conceptually the same, however using the current approach has some inherent benefits. One of the benefits in using the current approach is the fact that the common mode voltage at the comparator' input is kept at zero. For this reason, the comparator offset is kept constant throughout the entire input range so no input fluctuations affect the converter linearity. Consequently, the common mode rejection requirement for the comparator is reduced. Another benefit to this approach is that the comparator input (INP-) can be clamped with high-speed diodes

(Dl and D2 in figure 5-2) to ground. This reduces the voltage-input swing and accelerates the settling characteristics of the D/A and the comparator.

Figure 5 – 3 illustrates the D/A output voltage in the time domain for an 8-bit converter using the voltage comparison method. In this example the full-scale range of the converter is 2.048 V resulting in an LSB of 8 mV ($1LSB = 2.048 / 2^8$) and the analog input is 1.0 V.

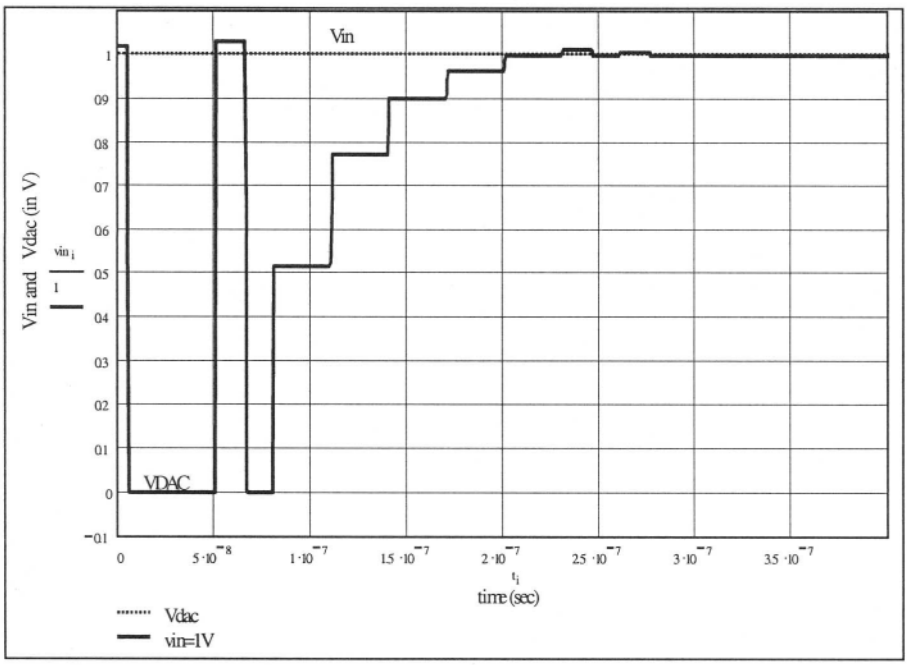


Figure 5-3. The successive approximation process for an 8-bit SAR A/D

Figure 5-3 is obtained using perfectly linear components with instantaneous time response (no bandwidth or slew rate limitation for the comparator and the D/A). Under these conditions for an input voltage of 1 V the A/D equivalent output code is 0.996V (the expected output should be 1V +/- 0.004V or 1V +/- 0.5 LSBs).

The operation of the SAR A/D requires that the analog input is held constant while the conversion takes place (between adjacent conversion start commands). If this condition is not observed the comparator' input resulting from converter's input minus D/A output voltage varies. In that case, the residue in front of the comparator becomes inconsistent from one bit trial to the next during the conversion and the conversion result is incorrect. To guarantee constant residue during the conversion a track and hold amplifier is used.

All simulations performed in this chapter model the THA with a voltage-controlled switch and a holding capacitor of 10 pF as illustrated in figure 5 – 4 similar to the model used in the previous chapter.

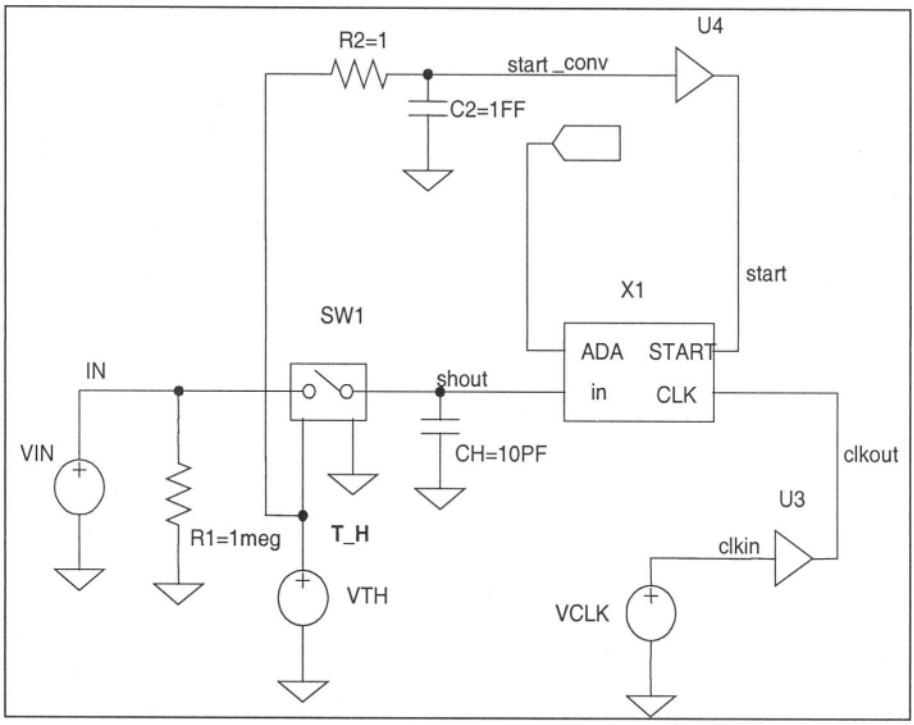


Figure 5-4. SAR A/D macro-model

In the figure, the start conversion is issued as a delayed pulse of the T_H command. The T_H is an analog pulse, whose rise and fall time can be controlled by the user. This voltage is converted into a digital pulse by the buffer U4. The input to the U4 buffer has a user-defined delay – the product R2 · C2. The converter clock used in this example is an analog voltage source – VCLK – that is translated into a digital pulse by the buffer U3. The sampling switch SW1 has a resistance of **1Ω** making the THA time constant equal to 10 psec **(=1Ω · 10 pF).**

X1 is the Spice model defined in the library as MACRO_SAR, which includes the comparator, D/A and SAR logic for the A/D. The inner works of this block are modeled in Spice as illustrated in figure 5 – 5.

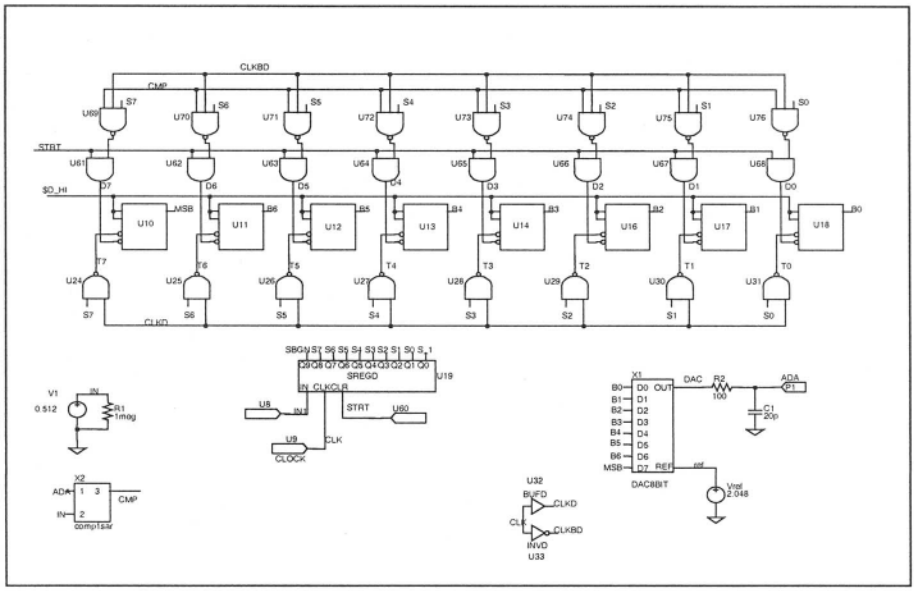


Figure 5-5. Testing the dynamic limitation of the SAR A/D loop

The devices in figure 5 – 5 are the following:

- X1 is the voltage D/A that can respond instantaneously
- R2 and C1 control the D/A voltage output time constant, making it a more realistic device. With the component values used the time constant of the D/A is τ_{DAC}= 2 nsec (=$R_2 \cdot C_1$=100Ω · 20 pF).
- X2 is the comparator modeled below in figure 5 – 6
- U19 is the serial register who's function is to propagate a pulse in a serial fashion from latch to latch such that each bit is tested sequentially as explained above
- The AND gates U24 through U31 allow the latches U10 through U18 to exercise the D/A bits in the sequence provided by U19
- U61 through U76 supply the logic to the latches U10 to U18 to keep or discard the state of each bit trial based on the comparator' decision
- U32 and U33 provide the required delays in the logic for the serial register
- Finally, the port labeled ADA gives the user a means of monitoring the D/A voltage at the input of the comparator.

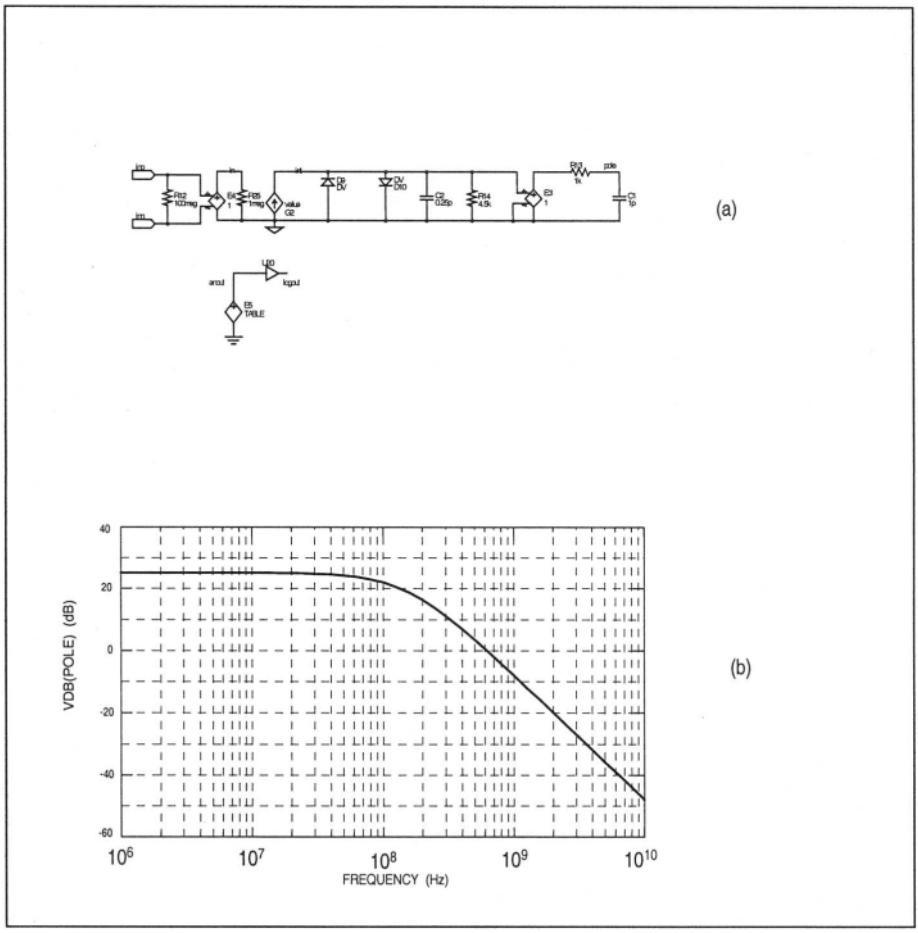


Figure 5-6. CMOS comparator - (a) Spice schematic and (b) Open loop gain

1.1 Physical limitations in a SAR A/D

1.1.1 A/D static errors

Unlike the flash A/D, the SAR topology uses the same comparator throughout the entire input range. Consequently, if the comparator has an offset error it will affect all bits in a similar fashion resulting in an output code with the same offset. Similarly, D/A gain error is also common to all codes thus causing a gain error for the entire converter. The gain and offset

errors do not affect the dynamic behavior of the converter as outlined in chapter 1. In contrast to the gain and offset errors, the only static errors affecting dynamic accuracy are related to the D/A linearity. In the next few sections, we examine the mechanisms affecting the dynamic accuracy in the SAR A/D. Our model for the converter uses a comparator with realistic limitations: slew rate of 4 V/nsec and bandwidth of 100 MHz illustrated in figure 5 - 6. The comparator open loop gain required for an 8-bit device is 54 dB **(=20·log(512),** so an accuracy of 0.5 LSB can be attained). This gain guarantees that the loop is able to correct for errors of more than one part in 2^9.

Assuming that the comparator has infinite common-mode-rejection and that the clock allows sufficient settling time we investigate first the D/A linearity effects on the converter accuracy. The static ILE errors in the converter are a direct result of the D/A linearity errors alone. If the ILE error belongs to the MSB, the first trial and decision of the comparator influences all subsequent decisions. Conversely if the ILE error is situated further down towards the less significant bits the comparator decision are affected from that particular bit on and subsequent bits thus resulting in a smaller (although more frequent) converter error. Since we are concentrating on static parameters, the slew-rate and bandwidth of the loop are not considered for the moment.

We recall now that our first example in figure 5-3 assumed no D/A error and converted the DC input of 1V to an output code of 0.996V. Performing the same conversion with a converter having **+1** LSB error for the MSB results in an output code equivalent to 1.00363V. If the same **+1** LSB error is set at bit B2, the equivalent voltage is 0.99634V. Although the error is relatively small and the output code is within the 0.5 LSB from the input voltage, it is interesting to note that the output depends on the error location. The reason is a direct result of the search algorithm – it depends on the direction from which the comparator input is being approached.

1.1.2 A/D dynamic errors

The SAR A/D converter also has dynamic limitations, since no physical components can transition in zero time. To examine the difference in dynamic behavior for various ILE DAC errors we analyze our 8 bit SAR A/D using a low frequency input signal. Using 128 time sampled points for the FFT and having a sample rate of 2.5 MSPS, results in an input beat frequency of 19.53125KHz **(= 2.5 · 10^6 / 128).** First, we check the dynamic behavior of the converter with an MSB error of 8 mV (equal to a magnitude of 1 LSB). The resulting spectrum for this error is shown in figure 5-7 (a).

Figure 5-7 (b) is the corresponding FFT spectrum for an equal error at the LSB.

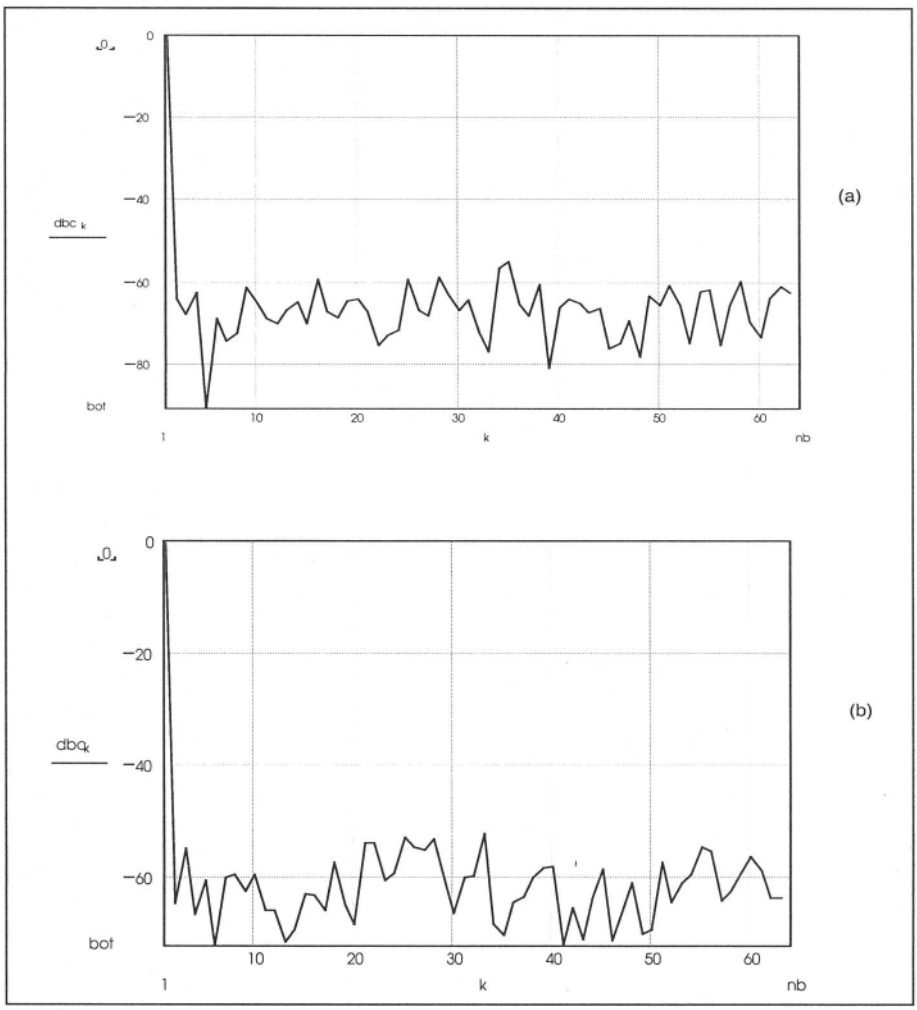


Figure 5-7. ILE effects on dynamic performance

It is interesting to note that the spectrum of the two devices is significantly different. For the case of ILE error at the LSB, the entire noise floor of the FFT is elevated. The FFT average noise floor in figure 5-7 (a) is approximately –70 dB; in contrast, figure 5-7 (b) shows an average on the order of –65 dB.

The SINAD for the MSB error is 46.1 dB while for the corresponding error in the LSB is 40.74 dB. This is a difference of virtually one effective

bit (each additional bit corresponds to 6 dB in dynamic performance as explained in chapter 1). In addition, the harmonics between the two cases are also located in different bins. The highest harmonics for the MSB error case is in bin 35 of the FFT with a magnitude of -54.93 dB and in bin 34 with a magnitude of -56.6 dB.

Many harmonics are between -52 to -53 dB with the worst harmonic in the last bin (64) with a magnitude of -51.45 dB. As expected the distortion caused by the LSB nonlinearity gives rise to a high frequency harmonic. In the MSB error case, the distortion occurs only twice per sinewave cycle. For the LSB case however the distortion occurs 256·2 times per cycle ($2 \cdot 2^8 = 512$; 256 times on the positive slope of the sinewave and an equal amount on the negative slope). For this reason, the noise floor of the FFT is increased and the ENOB is reduced by one effective bit.

It is also interesting to note the differences between the flash converter of chapter 3 and the SAR A/D under similar conditions. We investigated comparable linearity errors for both converters. However, the two topologies produced vastly different dynamic performance. In the flash A/D, the SINAD was almost constant regardless of the error location; in contrast with that, the SAR topology error location affected drastically its dynamic behavior. The reason for this behavior is that each bit in the flash converter is decided independently of all the others by one-and-only-one comparator in the array. In contrast with the flash converter, the SAR algorithm makes the bit decision using a single comparator; therefore if one of the more significant bit decisions is made in error than subsequent decisions attempt to compensate for it.

1.1.3 Loop speed

Another parameter that has a large influence on the converter dynamic accuracy, is the clock period to the SAR. If the clock is too fast compared to the loop bandwidth, then the error of the A/D can become significant. For an 8-bit system the amount of time required for settling (assuming no slew-rate limitations) is approximately 5.54 time constants ($= \tau \cdot \ln 256$). If the clock is faster than the loop speed the converter does not have sufficient time to settle to the correct value. This in turn causes an error in the integration process resulting in an incorrect residue at the comparator input. Of course, the loop requires more time for the MSB decision than the remaining bits. The MSB needs the full 5.5τ while the next bit requires only 4.85τ ($=\tau \cdot \ln 128$) and so on... This suggests the possibility of having a progressively faster clock for less significant bits. In practice, the circuitry needed to synthesize a variable speed clock becomes prohibitively complex without significant gain in converter speed.

We investigate the clock period effects on the loop speed limitation by varying in the clock period in the diagram of figure 5-4. The loop reaction time is limited by two mechanisms:

- the comparator pole - with a time constant of 1 nsec (**=R13·C1** in figure 5-6) and
- the D/A settling time – limited by its time constant of 2 nsec (**=$R_2 \cdot C_1$** in figure 5-5)

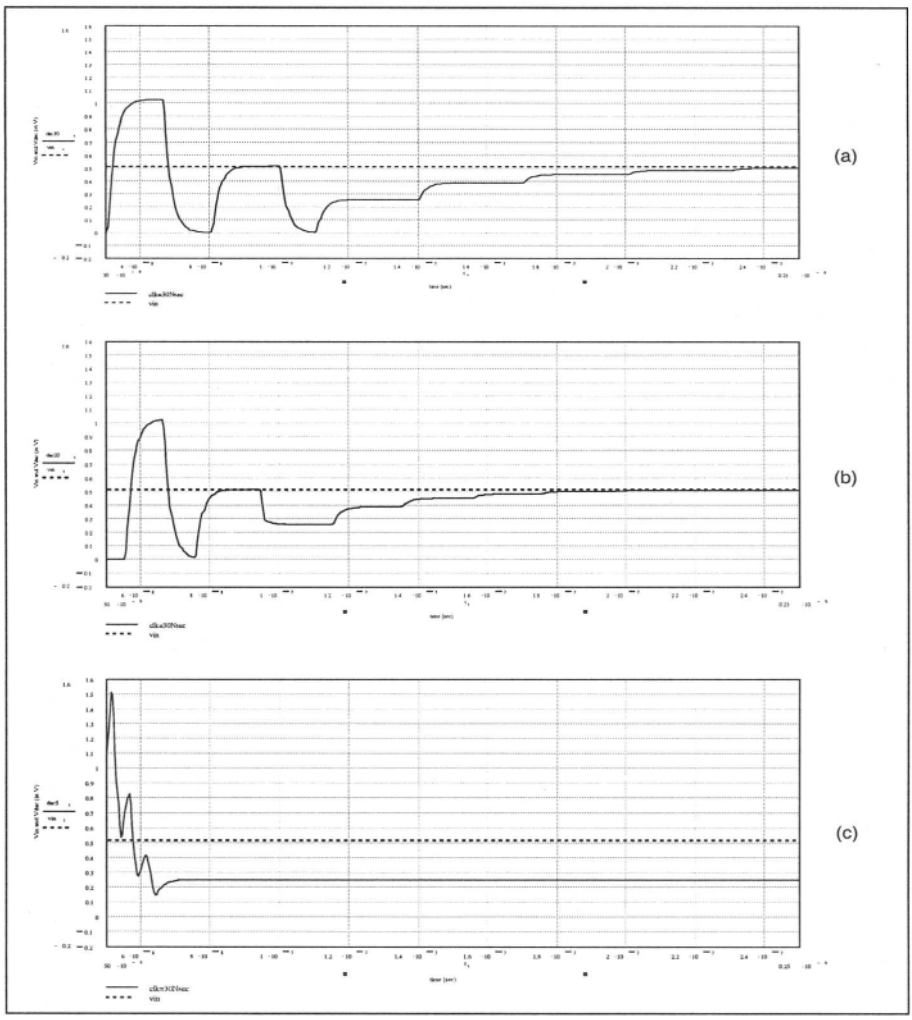


Figure 5-8. Clock speed effects on SAR A/D accuracy: **(a) clk=30 nsec, (b) clk=20 nsec, (c)clk =5 nsec**

Since there are two dominant time constants in the loop, the equivalent time constant is the RSS (Root Square of the Sum of the squares) combination of the two (in this case $\sqrt{2^2 + 1^2} = \sqrt{5}\,n$ sec = 2.236 nsec). Therefore the required amount of time for 8-bit settling is 2.236·5.545= 12.4 nsec.

Figure 5-8 illustrates the effects of clock period on the converter accuracy for our 8-bit converter SAR A/D. The figure shows the same 8-bit SAR A/D with a constant analog input voltage of 0.512V. Using three different clock speeds, we observe the equivalent A/D output code by monitoring the D/A voltage at the end of the conversion. The clock periods are 30nsec, 20nsec and 5nsec. In the figure, we notice that with a clock period of 30 nsec the final code is equivalent to 0.506V. At the 20 nsec the accuracy is maintained at 0.506V. When the clock speed reaches 5nsec the accuracy is lost, and the final value is 0.249V. As explained above at a clock period of 5 nsec the loop speed of 12.4 nsec was exceeded causing the converter to fail its settling requirements.

1.1.4 Dynamic A/D performance

Now we proceed to investigate the dynamic performance of the SAR A/D and its dynamic limitation as a function of input frequency. We know from previous discussions that the comparator used in the A/D loop has a very wide bandwidth (100 MHz). The question is how does the input signal frequency affect the converter accuracy?

For this investigation, we examine the dynamic behavior assuming a linear device (no ILE errors for the D/A). Using a clock period of 20 nsec, we guarantee that the loop is capable of settling (see previous example). With a sample frequency of 2.5 MSPS (conversion start command) and 128 time samples we use an input frequency of 19.53125 kHz (2.5 MHz / 128). The FFT analysis performed on the reconstructed input signal results in almost 8 effective bits at f_{BEAT}. Gradually we raise the input frequency up to 2.48 MHz or just about the sample rate. At this frequency, the dynamic performance becomes 7.36 ENOB or 46.1 dB. (*More accurately at a frequency of 2.48046875 MHz we are at the beat frequency below the sample rate (2.5MHz -19.53125 kHz)).*

This test obviously exceeds the Nyquist criteria of two samples / input cycle so why is the dynamic performance still this good? The answer is simple: with a sinusoidal input signal, the maximum rate of change occurs when the sinewave crosses the zero line. We can calculate the maximum change in voltage between two adjacent samples of the form:

$$V_1 = 1.024 \cdot \sin\left(2 \cdot \pi \cdot 2.048046875 \cdot 10^6 \cdot t_1\right) \qquad 5.1$$

and

$$V_2 = 1.024 \cdot \sin\left(2 \cdot \pi \cdot 2.048046875 \cdot 10^6 \cdot t_2\right) \qquad 5.2$$

where t_1 and t_2 are the two instances in time that the input signal is sampled and the input signal amplitude is 1.024V. The maximum rate of change is found to be 0.049068 V. This is due to the beat frequency. By increasing the frequency of the input signal we observe that the dynamic performance gets reduced. At an input frequency of approximately 50.37 MHz the SINAD is about 37 dB or 6.86 ENOB. This stands to reason since we reached approximately half the small signal bandwidth. This is illustrated in figure 5-9.

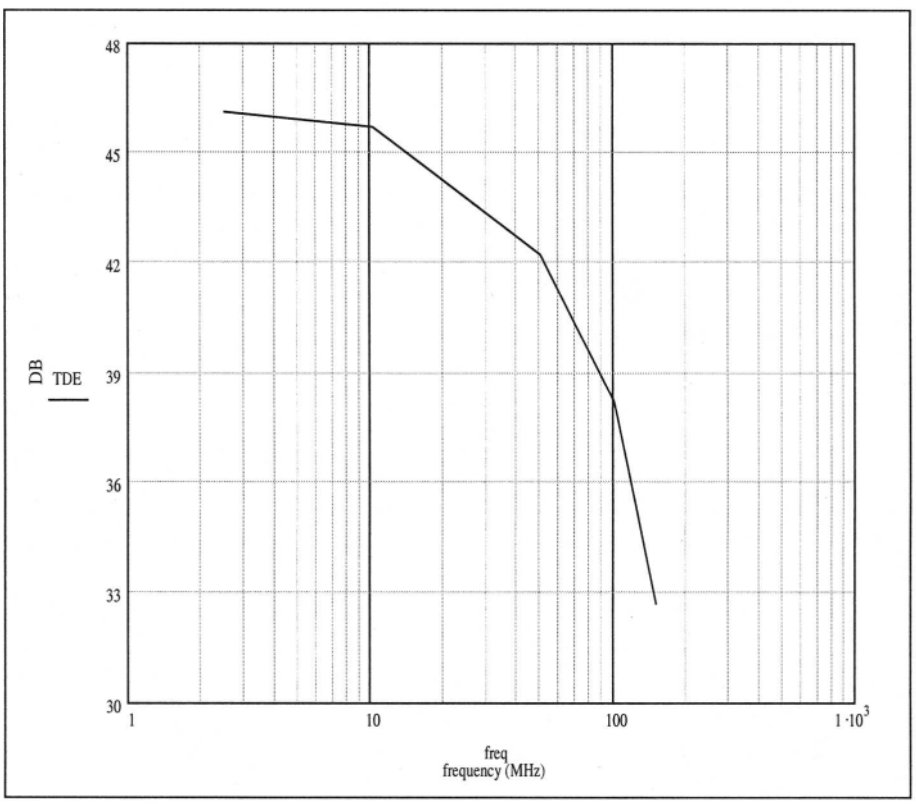


Figure 5-9. Dynamic behavior of SAR A/D

In the figure, we observe a very gentle reduction in performance such that at an input of over 150 MHz the device is still better than 5 effective bits. The important thing to remember for this converter however is that in order to obtain this sort of performance we need to use a track and hold amplifier in

front of the A/D. This way the analog signal converted by the A/D is held constant for the duration of the entire conversion cycle (in our case 8 clock cycles · **25 nsec = 200 nsec**).

1.2 Time interleaved converters

Given the speed limitation of the SAR A/D the obvious question is how can we improve the conversion time in a system? One way of improving the conversion time while using slow converters is to use multiple converters in parallel. This method allows one converter to perform its conversion while other converters in the array acquire the signal and get ready to start their own conversion process.

This technique is relevant not only to SAR A/Ds but to any converter, that has long conversion time and needs to be used in a high speed system. This parallel method is called for obvious reasons a parallel pipeline or time interleaved approach. In an interleaved converter, the conversion cycle is performed as illustrated in figure 5 - 10.

As seen in the figure, track & hold #1 captures the analog input first. Next, A/D #1 begins its conversion based on the THA #1 acquired analog voltage. At the next sampling instance, THA # 2 begins its acquisition of the input while A/D # 1 carries on the conversion process it started. When THA # 2 completes its analog input acquisition, A/D # 2 is ready to start its own conversion and THA #3 is ready to acquire its sample.
The sampling sequence continues until the last A/D (A/D # 4 in our example) is ready to begin its conversion while in parallel THA # 1 restarts the sampling cycle.

Figure 5-11 (a) illustrates a block diagram for this parallel pipeline scheme. In the figure, the output switch is synchronized and delayed from the input switch by the amount of time required for the individual A/D to complete its conversion.

Figure 5-11 (b) is the spice schematic used to simulate various error effects in time interleaved converters. In figure 5 – 11 (b) we see the track and hold amplifiers modeled with switches S1 through S4 and hold capacitors C1 through C4. The switches S5 through S8 are switched "ON" at the end of acquisition time of the respective channel so when the outputs are interleaved they arrive at the output node – ADA – at the proper instance.

As we emphasized numerous times, the SAR A/D technique requires that the analog input is held constant throughout the entire conversion cycle. If this condition is not met, the residue from one bit trial to the next varies compromising the accuracy of the converter in midstream.

An additional requirement is that all the channels in an interleaved A/D must match perfectly to one another. If this matching requirement is overlooked the combined converter becomes inaccurate.

In the next few examples we examine how various channel mismatches affect the overall converter accuracy.

In each of the cases, we assume that one and only one channel has an error relative to the other channels and that this is the only error in the converter. With this procedure, we have a means of evaluating error budgets and their associated signature for the combined converter.

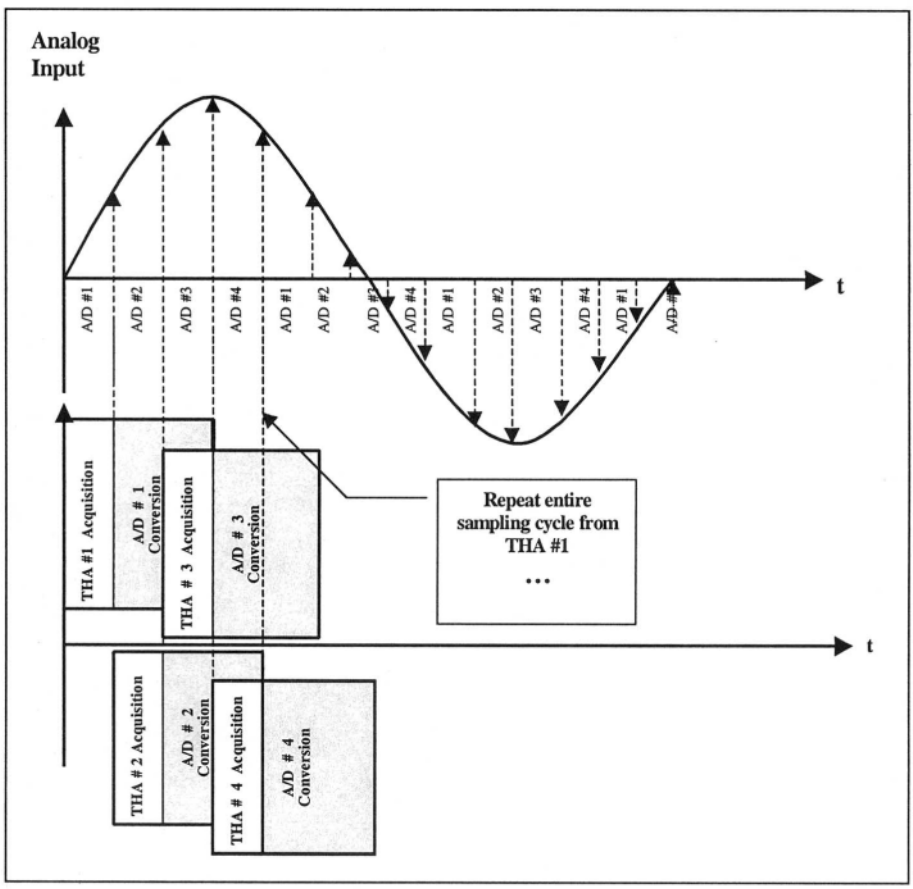


Figure 5-10. The cyclical nature of the time interleaved converter

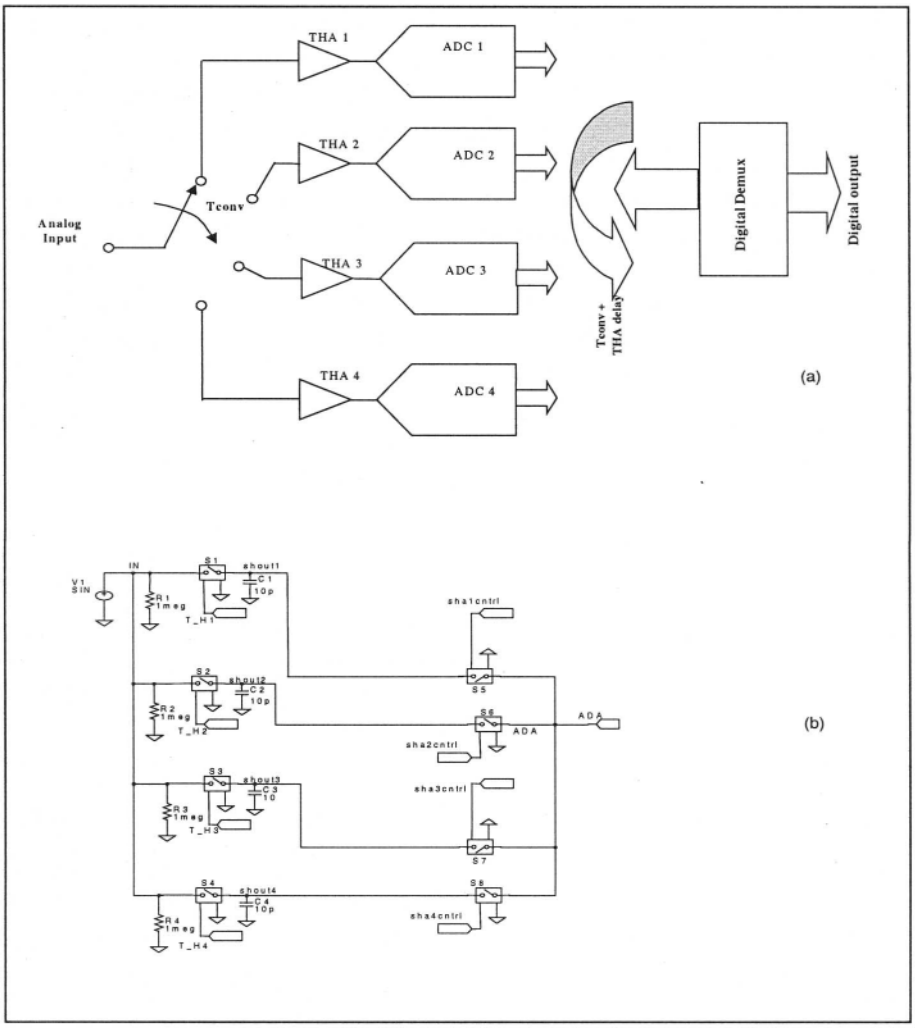


Figure 5-11. Block diagram - the digital output of each A/D is demuxed at the end of its conversion

Our examples assume also that each converter has infinite resolution (no quantization error). This assumption does not compromise the results and the generality of the simulations and allows us to observe the effects of channel mismatches. In our analysis, we consider an interleaved A/D with four parallel channels.

1.2.1 Case 1: offset error in channel # 1

The first case considers offset errors in the first channel of the interleaved converter relative to the other three channels that have no error.

If channel # 1 has an offset error of 10 mV relative to the remaining three channels the combined A/D has an error component in its reconstructed signal. Consider what happens if the analog input of the interleaved converter is a DC voltage. In this case, the first channel has an output equal to its offset voltage while the other three channels have outputs equal to the input. If we monitor the reconstructed output of the interleaved A/D then for one fourth of the time we notice an offset while for the remaining three-quarters of the time the output is equal to the input. Similarly, if the input signal is a sinewave then the reconstructed output has a frequency spectrum with a spur at multiple frequencies of the sampling signal divided by the number of parallel channels:

$$F_{SPUR} = i \cdot \frac{Fsample}{M} \qquad 5.3$$

In the equation, M represents the number of parallel channels employed in the interleaved A/D and the spurs occur at multiples of this frequency (where i is an integer). The spectrum of the output for this case is illustrated in figure 5 – 12 (b).

1.2.2 Case 2: gain error in channel # 1

Next, we examine the case where channel # 1 has a gain error relative to the other three channels. Considering a sinewave input signal we can intuitively see an amplitude modulation occurring for the channel with the gain error relative to the other channels. As expected for an AM modulated signal we anticipate seeing spurs of the original input at frequencies of:

$$F_{SPUR} = i \cdot \frac{Fsample}{M} \pm Fin \qquad 5.4$$

Again M represents the number of parallel channels in the array and i is an integer. In other words, the erroneous frequency components are seen at: $(F_{sample}/M) \pm Fin$, $(2 \cdot F_{sample}/M) \pm Fin$, $(3 \cdot F_{sample}/M) \pm Fin$ and so on... The spectrum of the output for this case is illustrated in figure 5 – 12 (c).

1.2.3 Case 3: systematic sample instant error in channel # 1

This error in the sampling instance of one channel relative to the other three results in an inaccuracy. Normally, when an A/D samples an input signal we assume mathematically that all samples are equidistant in time. If however one of the interleaved A/D channels samples at a different time than expected, the signal has a different amplitude than anticipated. Again, if we deal with a sinewave input this timing error causes a spur of the original input signal. In an A/D, this error is caused by a delay in sampling signal from one channel to the next. This is a similar to the case we analyzed in the flash A/D when one column had a delay in the sampling instance relative to the other columns. Again, the harmonic content of the reconstructed output looks like a combination of gain error but without the carrier F_{sample}/M. The spectrum of the output for this case is illustrated in figure 5 – 12 (d).

In all cases examined, we use 128 time data points resulting in 64 harmonic bins. Since the test performed is a beat frequency test, the input frequency is 10 MHz/ 128 = 78.125 kHz. The combined A/D has four channels and is sampled at a frequency of 10 MSPS. Each track & hold has an acquisition time of 10 psec (= 1 Ω · 10 pF) so the acquisition time is almost instantaneous and causes no error due to incomplete acquisition. The compound converter has a total conversion rate of 2.5 MSPS (4 channels each sampled at 2.5 MSPS lead to a combined sampling rate of 10 MSPS). Figure 5 - 12 (a) illustrates the case where no errors exist between channels. The SINAD of the A/D is 70.97 dB (or 11.48 ENOB).

Next case – figure 5 – 12 (b) shows the case where channel # 1 has an offset error of 10mV. Here we observe as predicted above two major spurs: one in bin 32 with an amplitude of –46.1dB (=2.5 MSPS) and one in bin 64 with an amplitude of –46 dB (= 5MSPS). The SINAD is 43.03 dB or (or 6.85 ENOB).
Graph 5 – 12 (c) represents a gain error of 1% on channel # 1. The harmonic content is summarized in table 5-1:

Table 5-1. Harmonic content of time interleaved converters

Harmonic bin	Equivalent Frequency	Relative Frequency	Level (dB)
31	2.421875 MHz	Fs/4 - Fin	-52.00
32	2.500000 MHz	Fs/4	-45.86
33	2.578125 MHz	Fs/4 + Fin	-52.16
63	4.921875 MHz	2*Fs/4 - Fin	-52.00
64	5.000000 MHz	2*Fs/4	-45.79

The last case analyzed – graph 5 – 12 (d) represents a timing error of 3 nsec. The harmonics reside now in bins 31 with amplitude of –67.47 dB, bin 33

with **–68.55 dB** and bin 63 with **–68.56 dB.** The SINAD is 62.64 dB (equivalent to 10.11 ENOB).

The cases just studied underscore the importance of matching the channels in an interleaved A/D.

The examples illustrate how a device with perfectly matched channels performs with an accuracy of almost 11.5 effective bits can deteriorate to an accuracy of less than 7 bits if the gain of a single channel is mismatched by 1%. Of course, multiple channel errors (gain, offset and delay) will worsen the overall A/D performance.

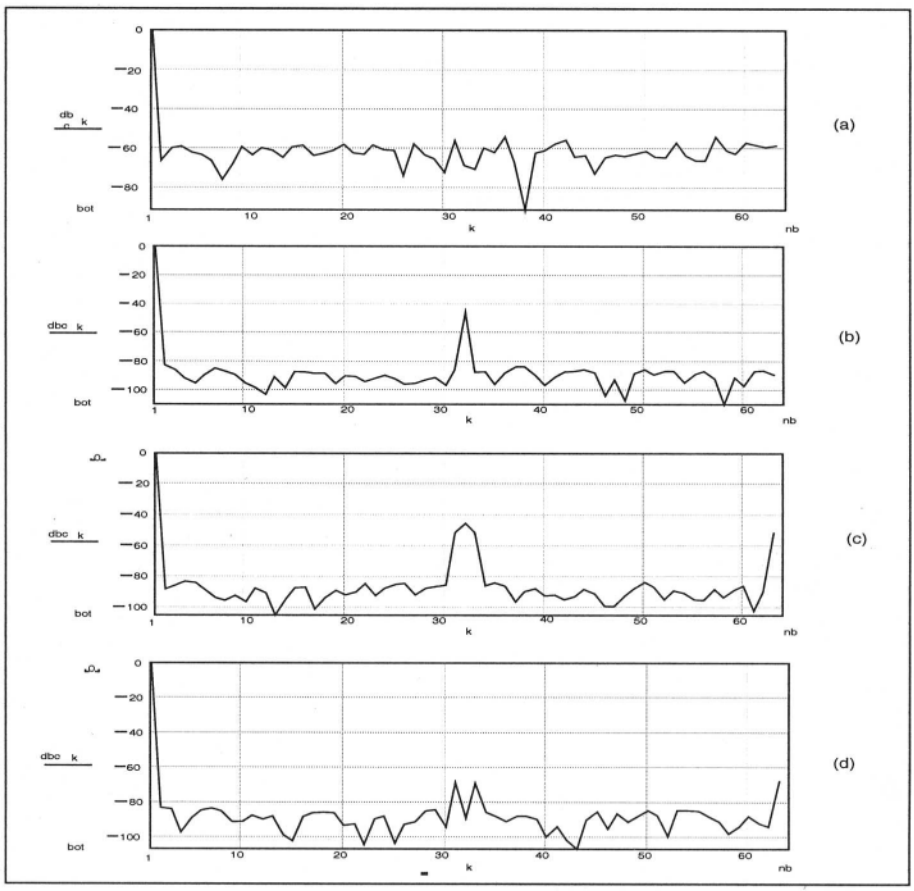


Figure 5-12. Dynamic performance (TDE) as a function of errors in one of the four channels. (a) no error between channels, (b) offset error in 1st channel, (c) gain error in the 1st channel and (d) timing error in the first channel

SUMMARY

- In a SAR A/D the D/A nonlinearities are less apparent than in a comparable flash converter since the search algorithm tends to compensate for previous decisions and the reconstructed output has vastly different harmonic components.

- In a SAR A/D an error in linearity at the LSB, contribute to a higher noise floor than an equal linearity error for the MSB.

- The clock utilized for the SAR A/D has a critical speed that can be used for a given converter accuracy. It has to accommodate both the trial and decision stages. To decide on the fastest clock cycle we need to know the speed limitations of each of the components in the loop. The RSS combination of the comparator time constant and the D/A time constant are the major contributors to this limit (in the absence of slew rate limitations).

- The THA is of crucial importance in the use of a SAR A/D. This device maintains the analog input constant in front of the comparator so the comparator can react to a consistent residue from one bit decision to the next.

- In order to increase conversion rate one can employ time interleaved A/Ds. While using this approach it is of paramount importance to have good matching between adjacent channels. Thus, it is customary to use a shared voltage reference for all the channels as a means of obtaining the same gain.

PROBLEMS

8. Use the comparator model of chapter 2 to obtain an open loop gain of 70dB, **BW=50** MHz and slew rate **SR= +/- 2 V/nsec.** Use the new comparator model in an 8-bit SAR converter assuming no D/A errors.
   (a) Find the highest sample rate possible to maintain the 8-bit accuracy for the converter.
   (b) Obtain a graph of TDE as a function of input frequency (as shown in figure 5 – 9) using the sample rate found above

9. Introduce an error of 1 LSB in second significant bit of the D/A. Perform FFT on 128 time data points at the following frequencies:
   (a) at the lowest beat frequency
   (b) at a frequency corresponding to –6dB point in TDE relative to problem 1.

10. Using the type of THA shown in the chapter construct an interleaved A/D converter with 8 parallel SAR A/D. Add a current source of **10 μA** from holding capacitor in THA #1 to ground. This will cause a voltage droop of 10mV in 10 nsec.
   (a) find the spurious frequencies caused by this phenomena relative to the "no droop" case
   (b) add equal **10 μA** currents on all holding caps so all channels are matched. How does the addition of droop influence the TDE?

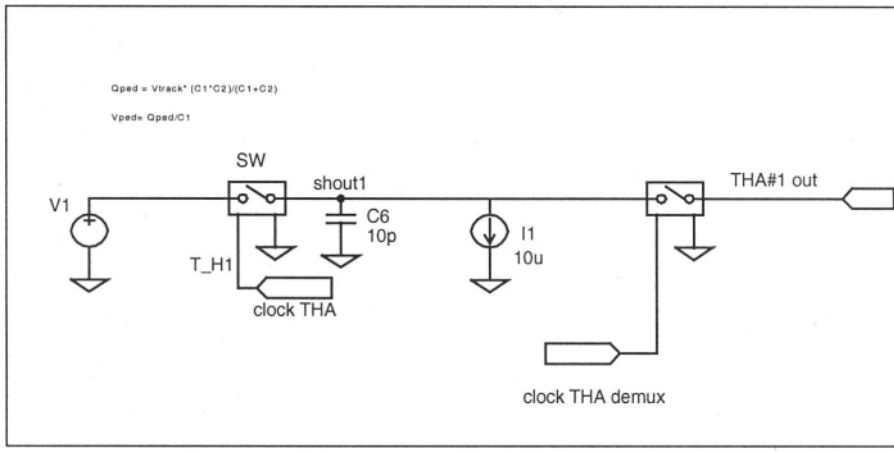


Figure 5-13. THA model for problem 3

Chapter 6

Folding A/D converters

1. INTRODUCTION

The folding concept is relatively new compared with the other converters examined so far. In 1975 A. Arbel and R. Kurz presented this new concept as a technique for obtaining high-speed A/D converters (45). Philips Research Laboratories in Eindhoven (46, 47, 48, 49, 50) and others (52,53) have further developed the method.

The primary effort concentrated on reducing the number of comparators in flash architectures in order to reduce circuit complexity. The expansion of the folding concept led to other advantages in comparison with the flash converter:

- reduced active area,
- lower power dissipation and
- reduced input capacitance resulting in easier drive requirements at the A/D input.

All these improvements are achieved while maintaining conversion speed comparable to the flash converter.

A block diagram of a folding A/D is shown in figure 6-1.

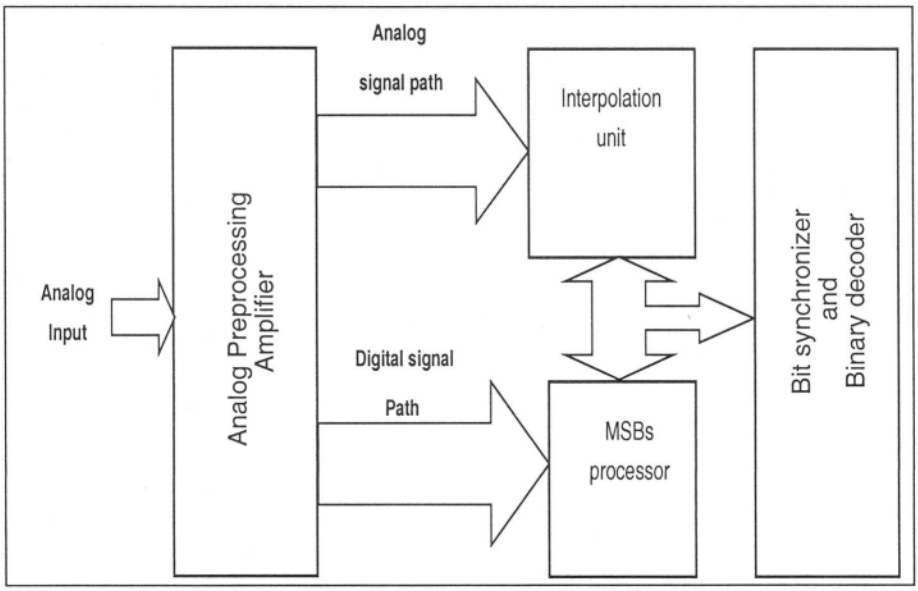


Figure 6-1. Folding A/D block diagram

The reduction in converter size is achieved by preceding the actual A/D with an analog-preprocessing unit as illustrated in figure 6-1. This device is in essence a continuous *nonlinear* analog encoder that reshapes the analog input voltage into a mathematical function capable of detecting zero crossings corresponding to A/D code transitions. The preprocessing front-end is followed by an interpolation unit which uses previously found zero crossings of the MSBs in order to detect the lower order bits. The conversion process is performed by detecting the MSBs with a flash converter working in parallel with the interpolation unit. Finally the synchronization and decoding unit synchronizes the MSBs and the LSBs prior to presenting the digital code to the output drivers.

1.1 The analog preprocessor

To understand the operation of the folding A/D, we start by investigating the role of the preprocessing nonlinear folding front-end. The idea behind the preprocessing unit is to manipulate the input voltage so that several zero crossings are obtained at its output. This allows a single comparator to be used for the detection of several zero-crossings across the input range, hence the number of comparators is reduced compared to a flash A/D.

It is important to understand the preprocessing concept in the context of an INPUT-OUTPUT transfer curve rather than in the time domain (most people have a tendency to think of sinewaves in the time domain only).

The preprocessing concept is outlined in figure 6-2.

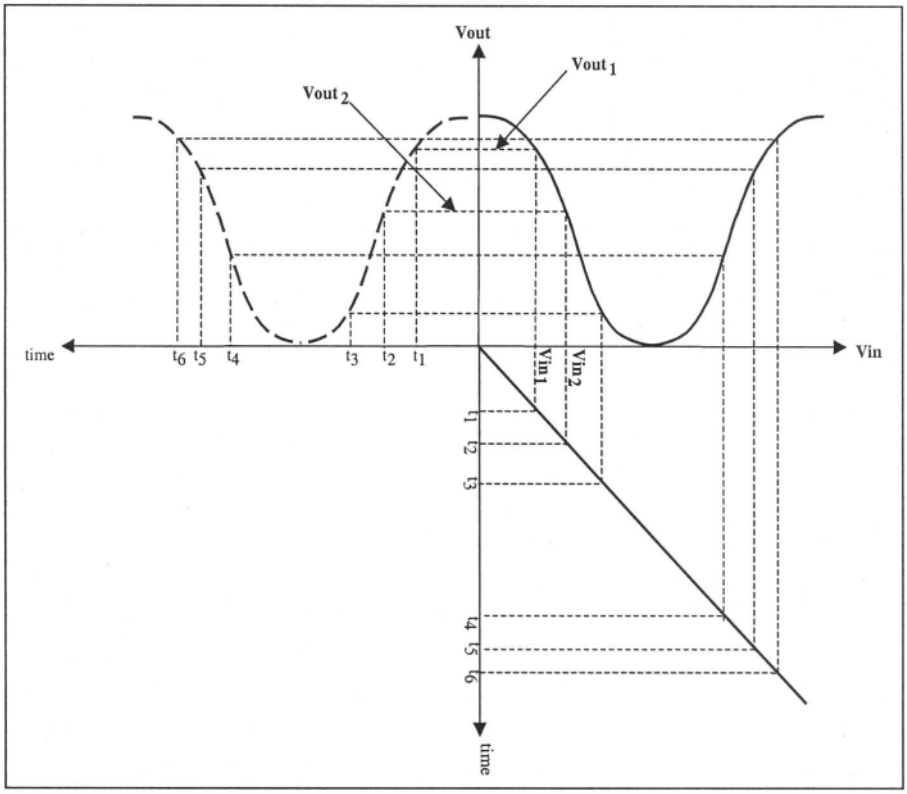


Figure 6-2. Input / Output transfer curve of the folding amplifier - Time domain linear ramp (lower right) and associated output time domain waveform (upper left)

Figure 6-2 contains the following elements:

- a transfer curve showing the relation between input and output (voltage in – voltage out)
- an input ramp in the time domain and
- the corresponding output waveform resulting from the input ramp (also in the time domain).

The information presented in figure 6-2 is somewhat unusual so an explanation is in order.

In our example the input-output relationship of the preprocessing unit is nonlinear and resembles a cosine function. This is presented in the upper right hand corner of the figure. In the lower right-hand corner of the figure the input ramp is shown with its associated time increments t1, t2...
The time axis is directed toward the bottom of the page for the input voltage such that $t_1 < t_2 < t_3$ and so on.

In the left-hand corner of the figure we observe the corresponding output voltage of the preprocessing device. The time scale for the output voltage increases toward the left side of the figure (again $t_1 < t_2 < t_3$).

Assume for now that the input voltage to this device is a linear ramp as shown in figure 6 – 2. At time t_1 the input ramp reaches the level Vin_1 resulting in an output voltage $Vout_1$. Next at t_2 the input reaches the level Vin_2 with the corresponding $Vout_2$ at the output etc. The distance between time t_2 and t_1 towards the bottom of the page in the lower right-hand graph (input) is equal to the distance between t_2 and t_1 in the left-hand graph (output).

As a result of the nonlinear characteristic of this transfer curve a **linear ramp at the input** of the preprocessing amplifier is mapped into an **output "cosine" voltage.** The fact that our preprocessing unit intentionally distorts the analog input is a counter-intuitive development (the relevance of the output voltage being a cosine function will be explained shortly).

Practical preprocessing amplifiers are based on the hyperbolic-tangent relation of input-output transfer relation of a differential bipolar pair (or the equivalent square root relation for a differential CMOS pair).

In figure 6 – 3 (a) we illustrate the transfer curve of the hyperbolic-tangent:

$$I_{out} = I_{tail} \cdot \tanh\left(\frac{Vin}{0.052}\right) \qquad 6.1$$

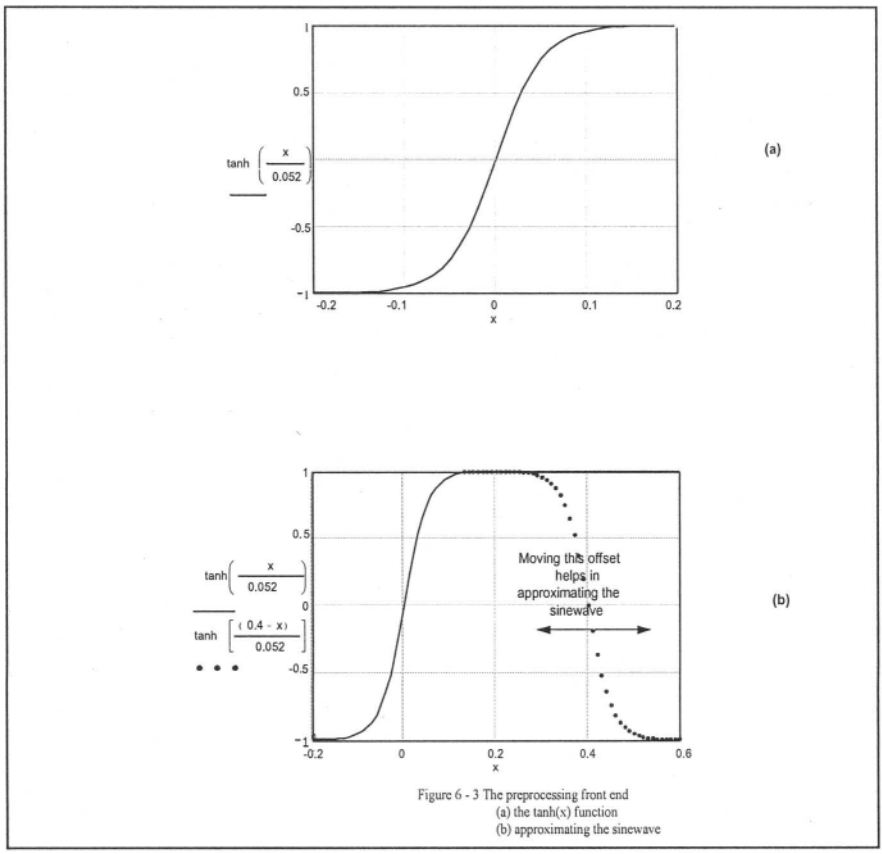


Figure 6-3. The preprocessing amplifier transfer curve - (a) tanh (x/0.052) and (b) approximating a sinewave

As the analog input voltage V_{in} moves from –0.2V to +0.2V the output current I_{out} moves from $-I_{tail}$ to $+I_{tail}$. If a similar transfer curve is offset relative to the first and subtracted from the curve in figure 6 – 3 (a), a graph as shown in figure 6 – 3 (b) is obtained. In other words by simply crosscoupling two sets of differential pairs and carefully choosing the reference levels a transfer function approximating the cosine function can be achieved. The approximation of a cosine function is of course maintained for a limited range of voltages. This idea can be extended such that multiple cosine cycles are obtained along the full-scale range, as we will illustrate by example. The concept is shown in a bipolar implementation in figure 6 -4.

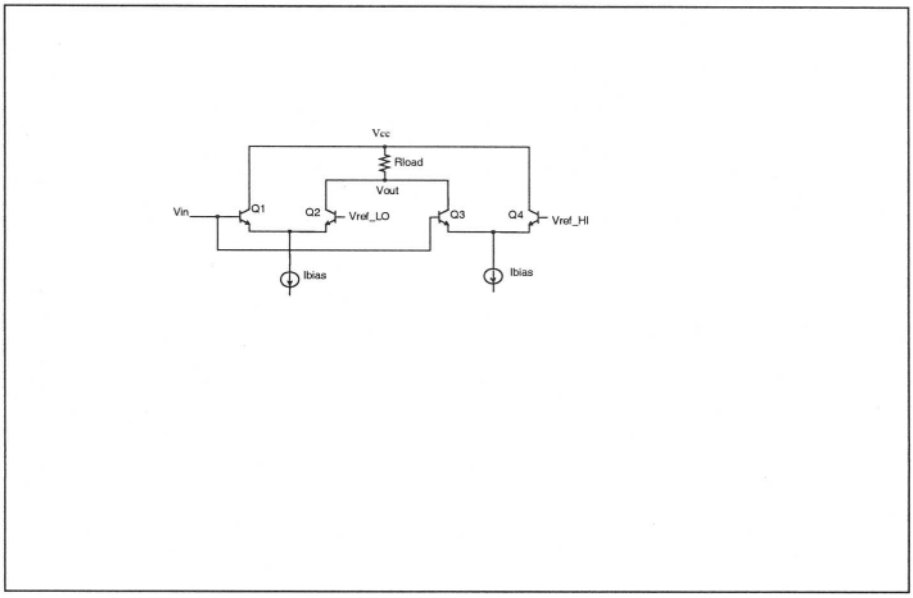


Figure 6-4. Coupling differential pair for preprocessing a folding A/D

In this chapter this implementation is abbreviated CDP (Coupled Differential Pair – see also ref. 1, 51).

The implementation of figure 6 - 4 uses four bipolar transistors connected in a wired-OR fashion. The two differential pairs – Q1 through Q4 have cross-coupled outputs such that Q2 (connected to the low reference at its input) has its output connected to Q3 collector of the second pair (rather than Q4 which is connected to the high reference at its input). As a result of the cross coupling the hyperbolic tangent of each pair produces a "bell" shaped function. The particular shape obtained can be adjusted with several degrees of freedom: voltage differences between adjacent references, amount of emitter degeneration, and gain of the stage (ratio of load resistor to degeneration resistor).

By properly choosing the reference voltages and the amount of emitter degeneration a transfer curve approaching the "cosine" relation can be obtained. Figure 6 - 5 shows the changes in this input-output relation as a function of reference voltages.

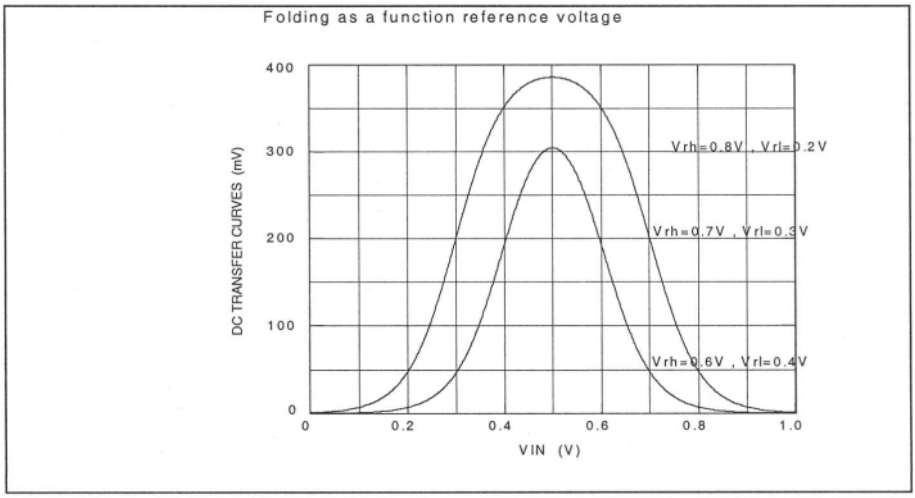


Figure 6-5. The folding transfer curve as a function of reference

The Spice model for the CDP is illustrated in figure 6 - 6.

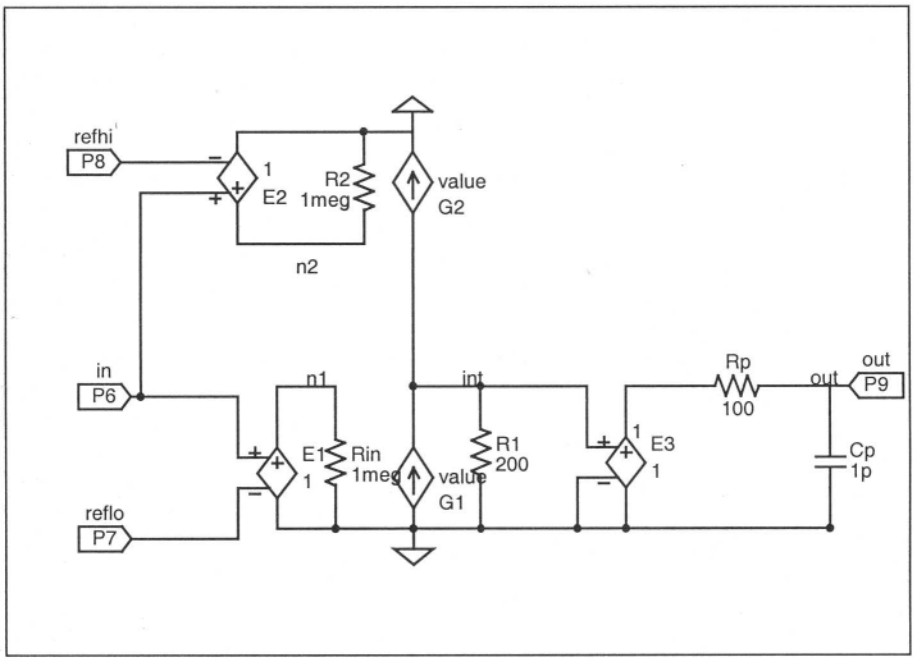


Figure 6-6. Spice model of the CDP - Spice schematic

As in previous chapters an effort is made to simplify the model of the preprocessing folding device. The intent is to shorten the simulation time by minimizing the number of nodes in the SPICE model without sacrificing the accuracy in its physical behavior. By using the G1, G2 voltage controlled current sources we achieve the large signal dependency of hyperbolic tangent expected in a bipolar differential pair:

$$I_{out} = I_{tail} \cdot \tanh\left(\frac{Vin}{2 \cdot V_T}\right) \qquad 6.2$$

where V_T is the thermal voltage (= **26 mV** at room temperature). If the differential pair has degeneration resistors in their emitters than the denominator is increased from the **$2 \cdot V_T$ (=52 mV)** to a larger number.
Our model uses **I_{tail} = 1mA** resulting in the familiar transconductance relation above and adopts an amount of degeneration of approximately 50 mV:

$$I_{out} = I_{tail} \cdot \tanh\left(\frac{Vin}{0.10}\right) \qquad 6.3$$

The components Rp and Cp in the model control the gain bandwidth product of the pair. In our example the CDP has a gain-bandwidth product of approximately 9.4 GHz with **Rp=100** ohm and **Cp=0.3pF.** This is shown in figure 6 – 6 (b) as the small signal response of the CDP.
The references used in our model for the CDP are **reflo=0.4V** and **refhi=0.6V.** This voltage difference of 0.2V in the CDP approximates the values used in the folding A/D we will analyze later in the chapter for static and dynamic performance.
Considering the equation **$I_{tail} \cdot \tanh (Vin / a)$ -** where *a* represents the amount of degeneration including $2V_T$ - and expanding it to a MacLaurin series with V_{in} as a variable we obtain:

$$I_{tail} \cdot \tanh\left(\frac{Vin}{a}\right) \approx \frac{1}{a} \cdot Vin - \frac{1}{3 \cdot a^3} \cdot V^3 + \frac{2}{15 \cdot a^5} \cdot Vin^5 \qquad 6.4$$

The transconductance equation for a differential CMOS transistor pair is (see ref.54):

$$I_{out} = I_{SS} \cdot \sqrt{\frac{\beta \cdot Vin^2}{I_{SS}} - \frac{\beta^2 \cdot V^4 in}{4 \cdot I_{SS}^2}} \qquad 6.5$$

In the equation the parameters Iss and β are specific to a given process and device geometry and V_{in} is the differential input voltage to the pair. Again we can expand the equation into series with V_{in} as the variable:

$$I_{SS}\cdot\sqrt{\frac{\beta\cdot V_{in}^{2}}{I_{SS}}-\frac{\beta^{2}\cdot V_{in}^{4}}{4\cdot I_{SS}^{2}}}\approx I_{SS}\sqrt{\frac{\beta}{I_{SS}}}\cdot V_{in}-\frac{1}{8}\cdot\sqrt{\frac{\beta}{I_{SS}}}\cdot\beta\cdot V_{in}^{3}-\frac{1}{128\cdot I_{SS}}\cdot\sqrt{\frac{\beta}{I_{SS}}}\cdot\beta^{2}\cdot V_{in}^{5} \quad 6.6$$

The two equations are very similar up to the fifth order and each has constant coefficients for a given process. This implies that using the hyperbolic tangent with a larger degeneration for the CMOS case will result in a reasonable approximation for a CMOS CDP. This methodology is comparable to what was described in chapter 2 for the CMOS comparator model. The similarity between the two transconductances was also illustrated in chapter 2.

With the CDP model of figure 6 – 6, we can proceed to investigate the behavior of the folding stage and the advantage it offers over the flash converter. As showed in figure 6 - 2 the preprocessing unit (or folding amplifier) mapped a linear transfer characteristic into a sinusoidal waveform. This manipulation of a linear input ramp to a sinusoidal shape gives rise to the multiple zero crossings at the CDP output. This method allows us to reduce the number of comparators for detecting several threshold crossings and substitute them with a single comparator.

By using a sinusoidal shape in the transfer function we get the benefit that additional zero crossings can be obtained by simple manipulations of trigonometric functions (53).

Two preprocessing amplifiers may be assembled by using two references such that a cosine and a sine waveform are obtained. The references of the cosine preprocessor have the same span as the references of the sine preprocessor. However both end-scale references of the sine CDP are offset by half the voltage span of adjacent zeros of the cosine CDP. This offset creates the needed phase difference between the sine and cosine phases. Next by performing multiplication, addition, and subtraction we obtain the additional zero crossings. The idea is based on the following trigonometric relationship:

$$\cos(2\cdot\alpha)=\cos^{2}(\alpha)-\sin^{2}(\alpha) \quad 6.7$$

and

$$\sin(2\cdot\alpha)=2\cdot\sin(\alpha)\cdot\cos(\alpha) \quad 6.8$$

Assuming that the **sin (α)** and **cos (α)** are available at the output of the preprocessing front end, we can obtain the **sin (2· α), sin (4· α)** and the associated **cos (2· α), cos (4· α)** etc. by using the above identities. Thus less significant bits can be obtained from the more significant bits by doubling the number of zero crossings as illustrated in figure 6 - 7.

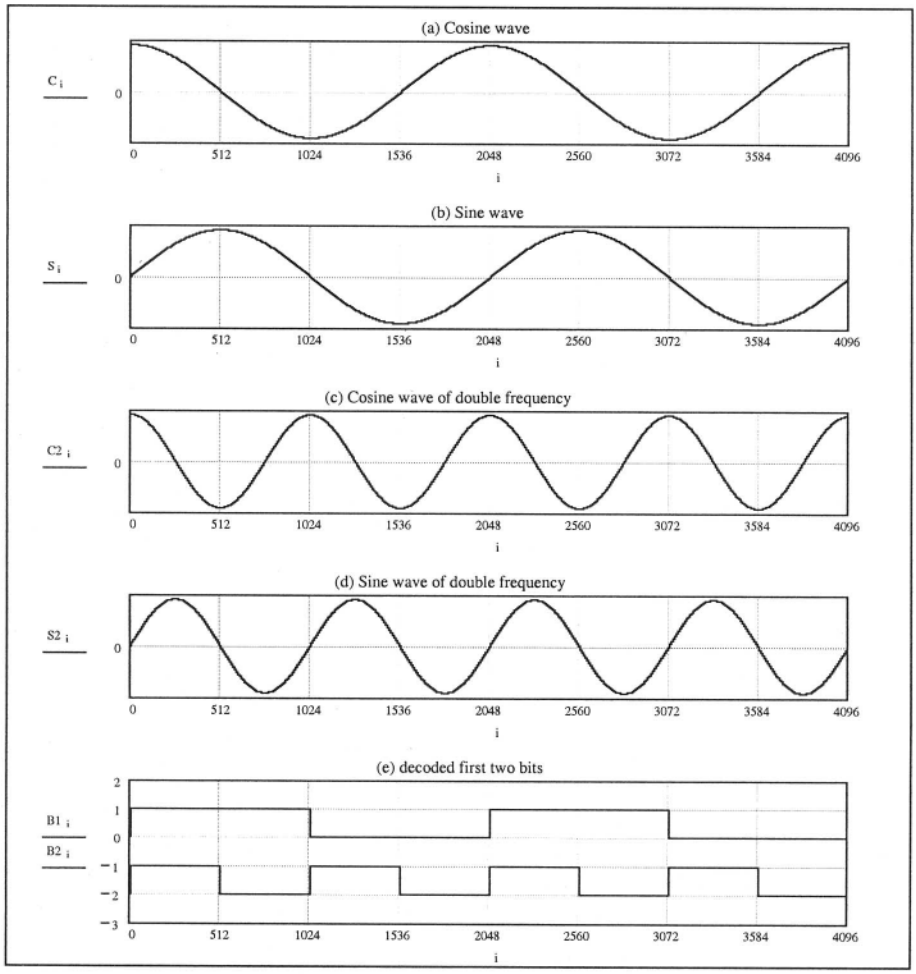


Figure 6-7. Trigonometric interpolation

In figure 6 – 7 we observe the cosine waveform and the associated sinewave on the top two panels. The following two panels are the cosine and sine waves of the doubled frequency – C2 and S2 - obtained by the relationships above. By detecting the zero crossings of the sinewave – S - in panel two we obtain the first bit – B1 in the last panel. Similarly by detecting

the zero crossings of the sinewave with doubled frequency – S2 – we obtain the second bit – B2.

The method just described in performing the interpolation is outlined in reference 53.

Another approach in obtaining additional zero crossings is to create multiple phases between the sine and cosine voltages and detect the zero crossings of these new phases. The method is described in references 1 and 51. References 51 and 52 are in-depth error analysis of this interpolation method and will not be discussed in this work (they are however a very good description of folding A/D operation and its error budgets).

An interesting matter in the folding A/D arises from the behavior of the CDPs at the input extremes. At those points the differential pair transfer curve ceases to approximates the sinusoidal behavior. Reaching this extreme, the current becomes constant and the sinusoid becomes distorted. This can be observed in figure 6 - 8.

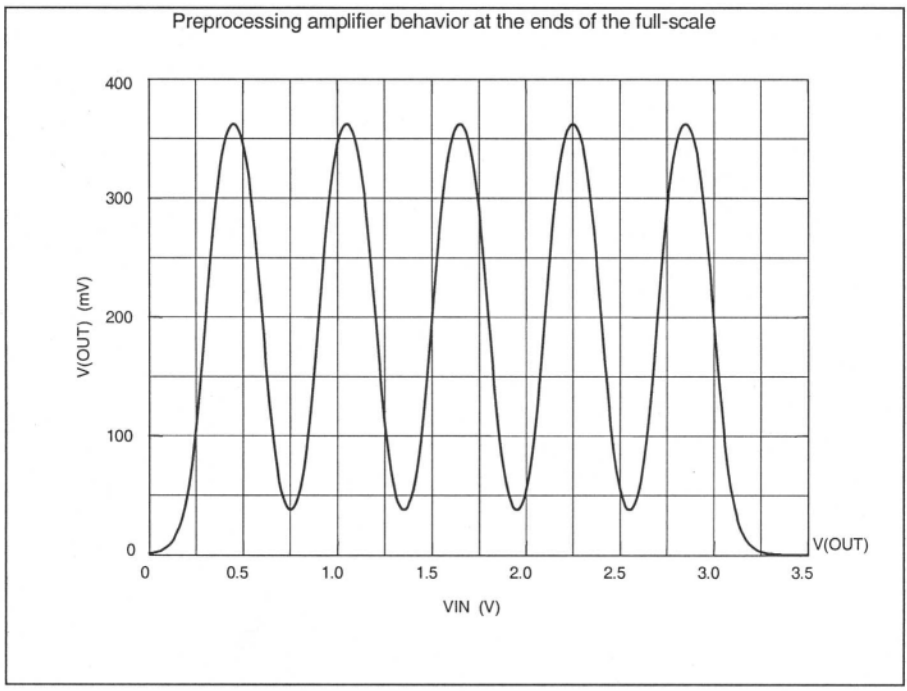


Figure 6-8. Preprocessing amplifier end-point behavior

As outlined in the figure, at the edges of the full-scale, the CDPs reach a plateau and the approximation to a sinewave ceases. To avoid the distortion we can limit the active input to the sinusoidal region and add dummy stages around the full scale. The dummy stages have the role of continuing the

sinusoidal behavior of the midrange CDPs. These stages have reference voltages that reside outside the full-scale range of the A/D. Therefore the nonlinear regions are never reached by the input. These dummy CDPs extend the sinusoid beyond the active input excursion and minimize the error in the **-FS** to **+FS** range (53).

1.2 Folding preprocessing unit – an example

With the SPICE models developed so far, we are ready to examine an 8-bit folding A/D and observe its static and dynamic behavior.
The first block of the converter we consider, is the nonlinear analog encoder – the folder – illustrated in figure 6-9.

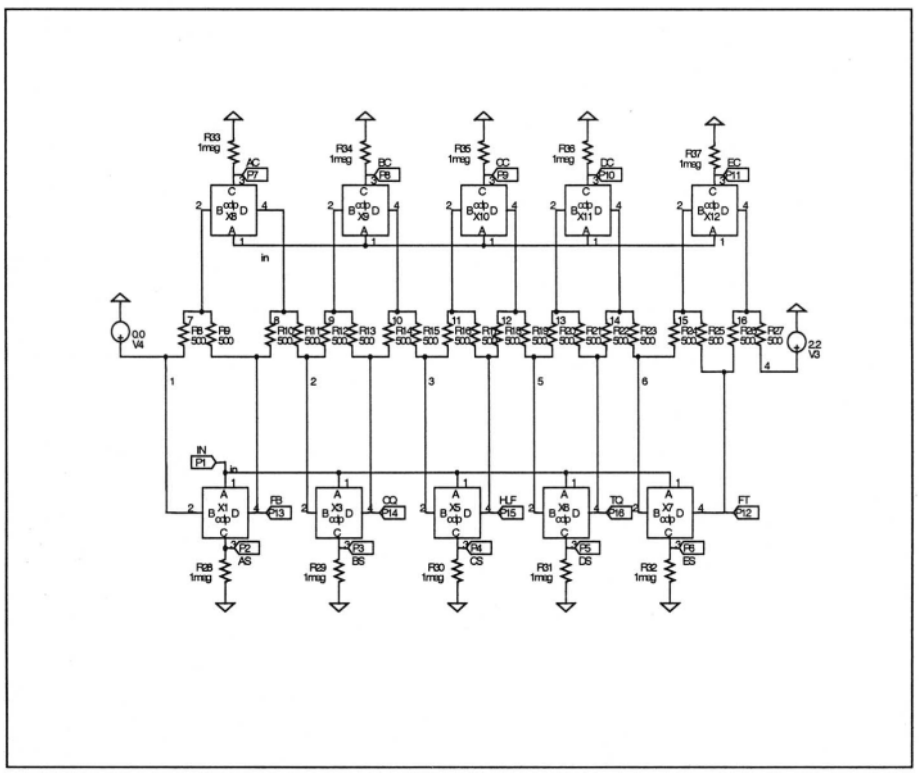


Figure 6-9. Preprocessing front-end with two phases - SIN and COS

Using the CDP simplified model in figure 6 - 6 we can investigate the preprocessing front-end operation. In figure 6 – 9 we observe a ladder comprised of equal resistors R_8 through R_{27}. The voltage references at the ladder extremes are $V_3 = 2.2V$ and $V_4 = 0V$ resulting in CDP reference levels

situated 0.22V apart from each other **(=2.2V / 10).** The five CDP blocks X_1, X_3, X_5, X_6 and X_7 produce the sine phase when outputs P_2 through P_6 are connected to a resistive load. X_8 through X_{12} produce the cosine phase when their outputs P_7 thorough P_{11} are connected in parallel to an equal resistive load. The two folded signals – SIN and COS - resulting from an input linear ramp are depicted in figure 6 - 10.

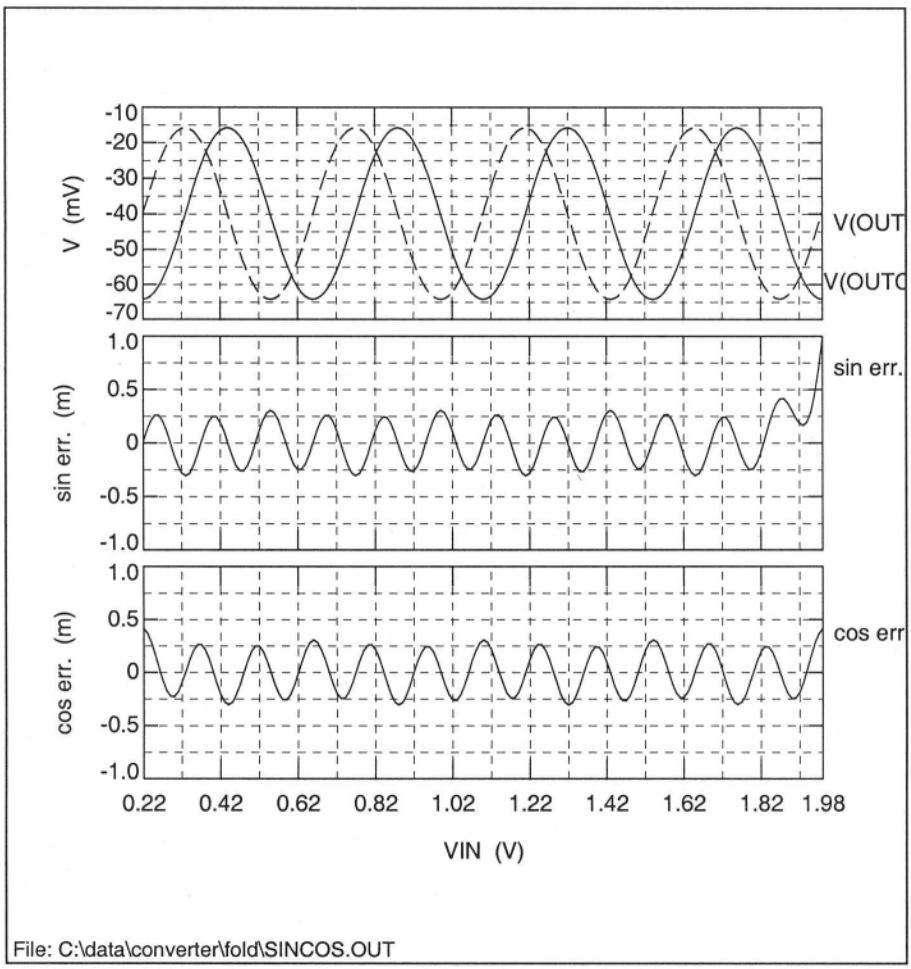


Figure 6-10. SIN and COS signal generation at the output of preprocessing unit and associated deviation from the perfect sine and cosine waveform

Figure 6 – 10 illustrates that the sinewaves have four cycles during the analog ramp transition from 0.22V to 1.98V. As outlined above we restrict the input range for the converter so the sinusoidal approximation is

maintained as long as the extremes of the full-scale range are not reached (0V and 2V).

Figure 6 - 10 also shows the error of the sinusoidal approximation. To compare the accuracy of the SIN and COS phases to an actual SIN/COS generator, we use the behavioral model available in SPICE to generate the two signals:

$$V(tst_S) = 0.04 + 0.02456 \cdot \sin\left(\frac{V_{in} \cdot 9 \cdot \pi}{1.98}\right) \qquad 6.9$$

and

$$V(tst_C) = 0.04 + 0.02456 \cdot \cos\left(\frac{V_{in} \cdot 9 \cdot \pi}{1.98}\right) \qquad 6.10$$

These two test signals represent the sine and cosine functions of a scaled version of the input signal. As the input voltage is gradually increased from 0V the test voltages complete approximately four and a half cycles (since a sinewave requires $2 \cdot \pi$ to complete one cycle, $9 \cdot \pi$ represent approximately 4.5 cycles). As shown in figure 6 - 10 a close match to the SIN and COS phases is obtained. In our example the analog input is scaled for a full-scale-range of 1.76 V **(=1.98V – 0.22V)** resulting in an LSB of 6.875 mV **(=1.76/2^8).** The SIN output phase deviation from an actual sinewave shown in figure 6 - 10 is less than 0.3 mV (or less than one twentieth of an LSB for the converter).

Several points should be made regarding the circuit implementation of our example. The transfer curve of the CDP in figure 6 – 6 is obtained with the hyperbolic tangent with an equivalent degeneration of 50 ohms. The active input range for each CDP is set to be 220 mV. This voltage span between adjacent inputs of each CDP results in currents of **± 1mA** at the CDP output. At node **int** in figure 6 – 6 this results in a voltage swing of **± 200 mV (=1mA· tanh(Vin/0.1)).** With **E3=1** (and assuming a low enough frequency such that C_P and R_P do not limit the response) this results in a **±200 mA** current swing at the CDP' output.

Of course this is quite a large current even for a high speed A/D, but the choice is made to illustrate a point. Since the physical swing expected from such a circuit is of the order of a hundred milivolts we use a load of 0.2 ohms in our model. Since each two adjacent CDPs are active concurrently for an input voltage it means that using a 0.2 ohm load results in an 80 mV output swing (0.2 · 400 mA).

The point of the exercise is as follows: in creating the model a value of 0.2 ohm is not realizable (or even sensible) for an IC. However, the actual physical behavior of the folder is not affected by the scaling. Changing the value of E3 and/or G1 would result in a more practical value for the load resistor. (The reader is encouraged to verify the accuracy of this statement.)

Another important characteristic of the folding A/D can be observed in figure 6 -11. This is the frequency multiplication of the input signal distinctive to the folding operation.

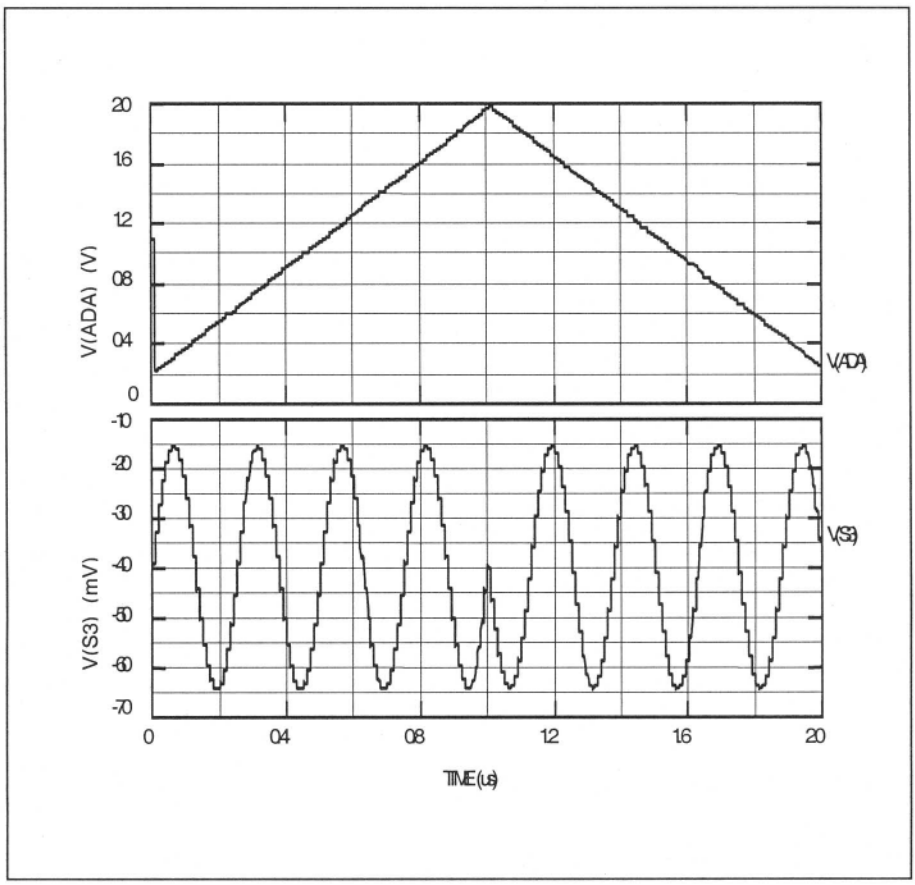


Figure 6-11. S3 signal obtained for one cycle of input ramp

In our example the folder creates a sinewave with four cycles across the full-scale range for an analog input ramp. If the input is a triangular waveform then for each leg of the input signal from **–FS** to **+ FS** we obtain four cycles followed by four corresponding cycles on the way down from the

+FS to –FS. In the case of a sinusoidal input the output of the preprocessing folder has therefore eight distinctive cycles at its output for similar reasons. This means that the input frequency to the preprocessing unit results in a frequency multiplication by a factor of eight (in our example).

Next, each interpolator behaves in similar fashion by doubling the input signal frequency at its output - as illustrated in figure 6 - 7. Thus, a frequency multiplication by a factor of two for each subsequent interpolator produces considerable demand on the process (in terms of speed requirements).

For example if the input frequency of our folding A/D is 10 MHz and the folding preprocessor has four cycles, followed by six interpolators it means that the output frequency of the last interpolator will be approximately 5 GHz. The total frequency multiplication becomes therefore: 10 MHz $\cdot$ 8 $\cdot$ 2^6 = 5.12 GHz. With a bipolar process with an f_τ = 15 GHz the remaining gain available at that frequency will be $\beta \cong 3$ (=15GHz / 5 GHz)!

This result shows that the folding operation creates a significant demand for bandwidth. This is not a very strict limitation in the final folding design since adding a sample and hold amplifier in the converter chain alleviates the bandwidth requirement for subsequent stages. This solution comes at the cost of increased power requirements (added by the THA) and adds pipeline delays, which are acceptable in most applications.

1.3 The linear interpolator

We take now a brief look at the interpolators. The easiest implementation of the interpolator is a four-quadrant multiplier as shown in figure 6 - 12. The two inputs for the circuit are the pair A and A_{bar} and B and B_{bar}. The resistors R_1 and R_2 linearize the behavior of Q_1 and Q_2 while R_3 and R_4 linearize the transfer characteristics of Q_3 and Q_4. The only difference between the two inputs is the level shifting required for the pair Q_3 and Q_4 to prevent the pair from saturating.

Using the four-quadrant multiplier of figure 6 – 12 allows us to implement the trigonometric identities just mentioned. The trigonometric interpolation technique outlined in our example is preferred over the resistive interpolation of (51,52) since it is only searching for zero crossings. Given that the four-quadrant multiplier detects a zero for any input equal to zero, it results that even if the sine or cosine approximations are not exactly correct the interpolation will be acceptable as long as the zero crossings are accurate. This is obviously an easier requirement than the resistive multiphase interpolation given that in this particular implementation each phase needs to be matched to the converter accuracy at the zero crossing.

Using only two phases – SIN and COS – the acceptable errors can be larger outside the zero crossing.

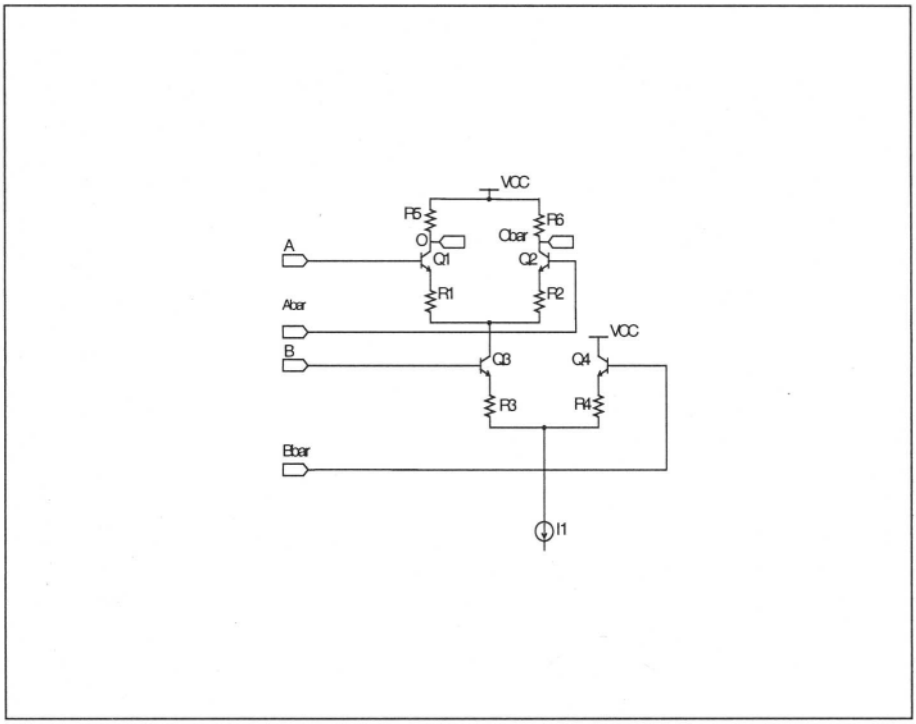


Figure 6-12. NPN implementation of the interpolator

On the other hand, the multi-phase interpolation has the benefit of using resistors, which are inherently linear components and provide good matching when good careful layout rules are observed. However loading effects can prove rather stringent with this approach. From these considerations, the trigonometric interpolation is the preferred approach and it will be the method employed in our folding A/D example.

Finally, to complete the description of the blocks used in developing our Spice model of the folding A/D we illustrate the comparator model and its associated Bode plot in figure 6 -13.

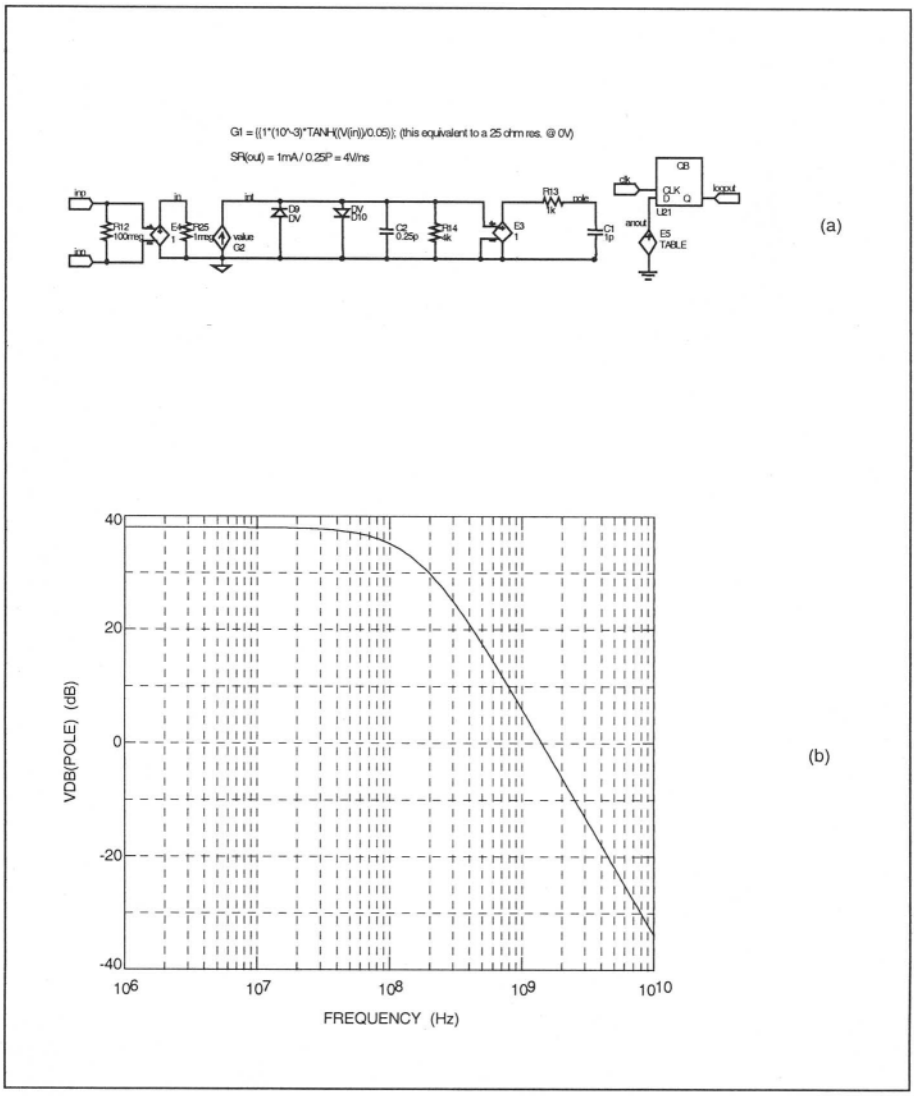


Figure 6-13. Folding A/D comparator model - (a) Spice model and (b) AC transfer curve

As in previous chapters we model the comparator using the now familiar hyperbolic tangent function, assuming a gain-bandwidth product of 4.5 GHz and slew-rate limitation of 4V/nsec.

1.4 Example of a folding 8-bit A/D

The SPICE circuit schematic used to simulate our 8-bit folding A/D is shown in figure 6 - 14.

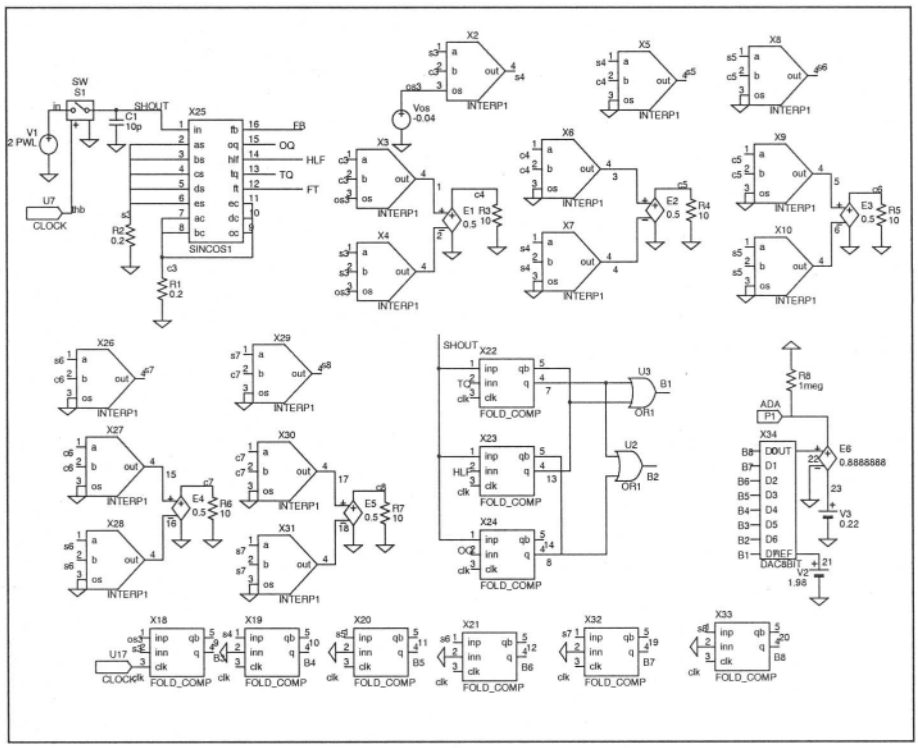


Figure 6-14. 8-bit folding A/D Spice model

As explained in the previous section our example illustrates the folding A/D operation using the folding preprocessor SINCOS with the 0.2Ω resistive loads R_2 and R_3 necessary to restrict the output voltage swing to 80 mV. The resistance value is not relevant since it only implies a scaling factor while the bandwidth and slew rate limitations still represent an accurate physical behavior for the circuit.

A sample and hold amplifier is used at the input of the converter to guarantee that the input signal is held constant during the conversion process. Lacking the sample and hold results in an erroneous conversion: the coarse converter (flash converter at the front end) is mismatched in time with the folding converter. In other words, the residue left after the flash conversion takes place changes after the first part of the conversion. This

means that subsequent stages have newer data, which is unsynchronized to the first part of the conversion. As outlined in the previous chapter this is one of the most important aspects in any subranging A/D. In our example the holding capacitor C1 and the switch S1 simulate the THA function.

The next function implemented in figure 6 -14 is the flash converter needed to generate the first two most significant bits (B1 and B2). Without consideration of over/under range for the flash converter, only three comparators are required (X22 – X24). The decoding of the flash is accomplished with the help of U_2 and U_3 OR gates. In our example these gates have insignificant delays **(=1psec)** as their delays are inconsequential to the understanding of the folding A/D operation.

The interpolators X2 through X10 and X26 through X31 are used to obtain the sinewave S4 through S8 and the cosine phase C4 through C8. These blocks implement the trigonometric approximation described in the previous section.

Finally for each bit derived with the help of an interpolator, only one comparator is required following each subsequent SIN phase. Therefore for the 6 LSBs we use six comparators (XI8, X19, X20, X21, X32, X33). The total number of comparators used for the entire 8-bit converter is therefore limited to nine, compared with 255 required in a flash A/D with similar accuracy.

The error of the INPUT-OUTPUT transfer curve of the 8-bit folding converter is shown in figure 6 -15 after the digital output of the converter is reconstructed with the help of X34 – an 8-bit D/A. The error is the difference between the analog input and the reconstructed output voltage. As expected the reconstructed output is limited to a swing of 0.22V to 1.98V and can be monitored on the node ADA.

The graph in figure 6 - 15 is obtained with an input voltage ramp covering the full-scale range in 1µsec. Although the quantization error is riding on top of the transfer curve, the cyclical nature of the folder can be easily observed.

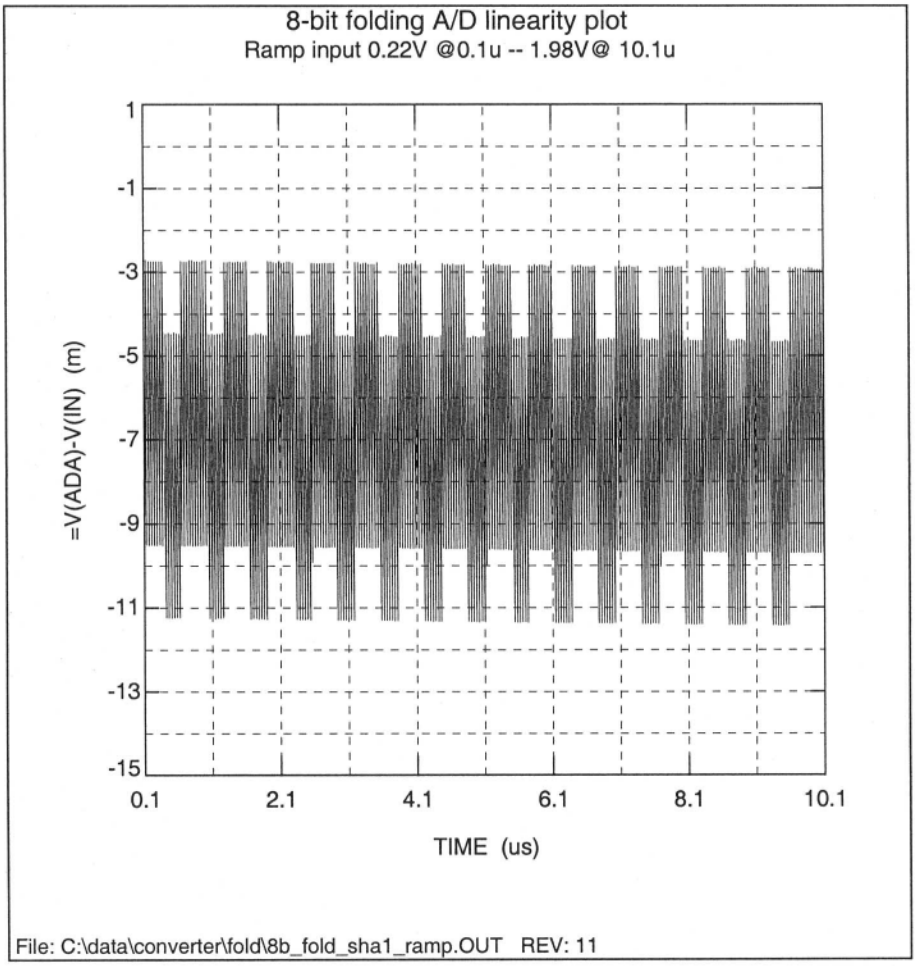


Figure 6-15. The 8-bit folding A/D linearity error plot **(1 LSB = 6.875 mV)**

1.4.1 Folding A/D converter dynamic performance

Using the Spice model for the converter in figure 6 -14 we evaluate now its dynamic behavior. As in previous chapters we perform a beat frequency test with a sampling rate of 100 MSPS. Using a time data set of 128 points we obtain a beat frequency **fin=781.25 kHz.** As in previous chapters we evaluate the A/D performance by performing several simulations with increased input frequency while keeping the sampling rate constant at 100 MSPS. The corresponding dynamic performance is shown in figure 6 - 16.

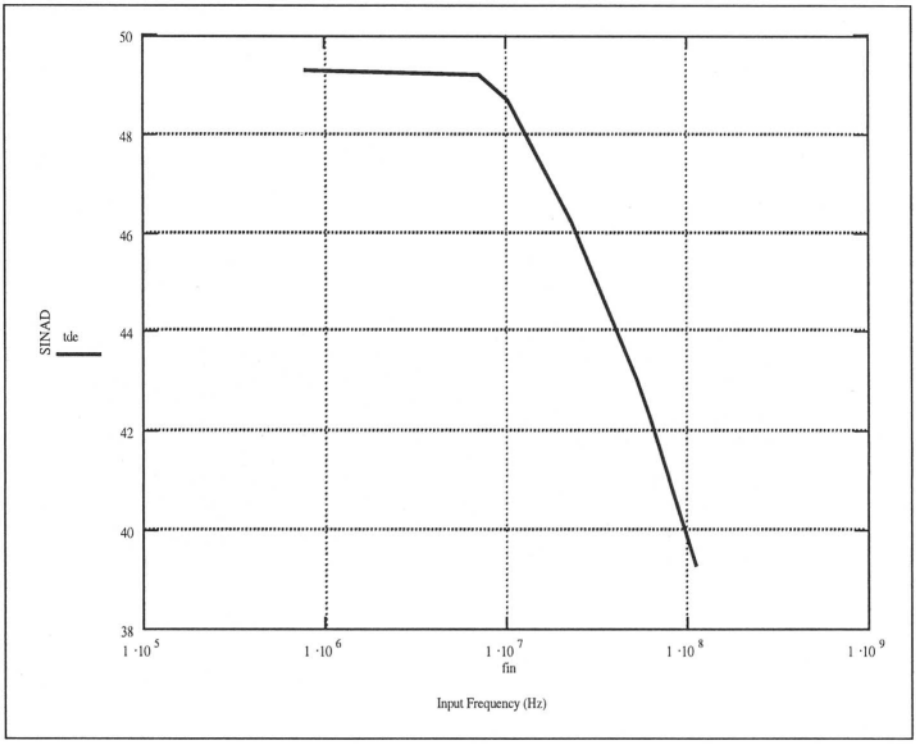


Figure 6-16. Dynamic performance of the 8-bit folding A/D of figure 6-14

The performance suggested by the graph is quite satisfactory. While not as linear as a flash, the converter has a reasonably constant dynamic performance up to 100 MHz. In fact the change from DC to 50 MHz is only 6 dB – a reduction of one effective bit in performance. Given the frequency multiplication of the folding scheme this is quite remarkable and not an intuitive result. The figure suggests however that beyond the 50 MHz barrier the converter starts loosing dynamic performance quite rapidly. The reason is obviously due to the frequency multiplication.

1.4.2 Folding A/D static errors and associated dynamic behavior

As in previous chapters, in the next sections we introduce static errors in the transfer curve of the converter. Since we already analyzed the behavior of the flash converter with threshold error in various comparators we analyze other errors characteristic to the folder.

First we examine the behavior of the folder with an offset error in one of the CDPs. The linearity plot for the 1 LSB error in X5 CDP is shown in figure 6 - 17.

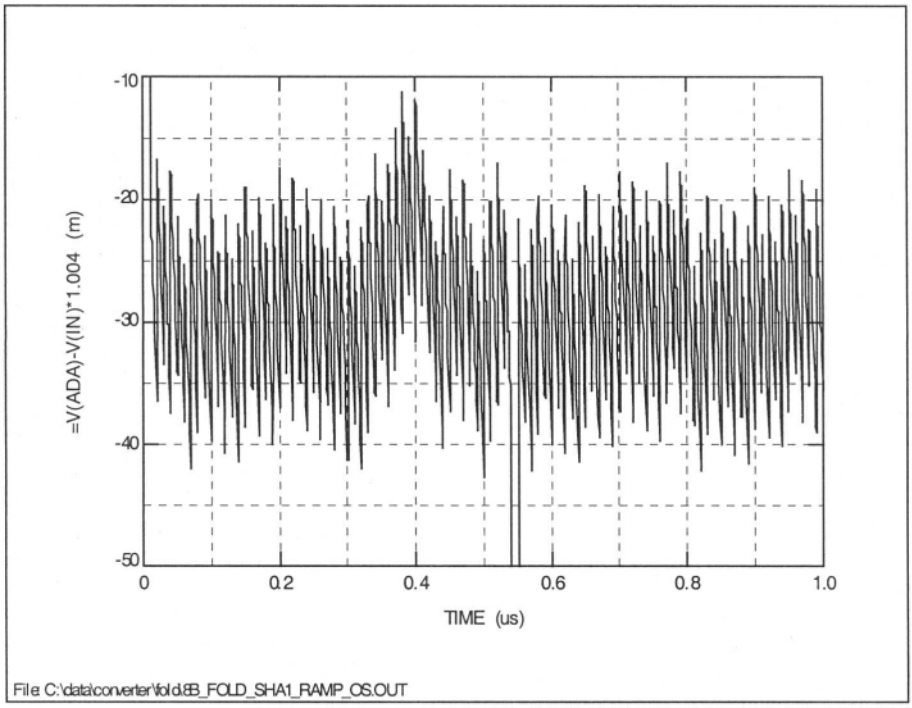


Figure 6-17. 8-bit folding A/D - Linearity error plot with CDP having 1 LSB of error (1 LSB = 6.875 mV)

The figure is obtained using a slow input ramp. In the figure we notice that the nonlinearity is restricted to a small region of the full-scale range – the area where the CDP error occurs. In principle the "bulge" seen in the second cycle of the folder is happening in a very narrow section of the transfer curve. Given the confined area the expected harmonic content should result in relatively high order harmonics. An FFT analysis verifies that the harmonics limiting the dynamic performance are the tenth and the twelfth. With this offset error in the middle CDP of the SIN curve (X5 in figure 6 - 9) we obtain a SINAD = 47.37 dB at low input frequencies. This is a reduction of approximately 1.7 dB in dynamic performance at an input frequency of 781.25 kHz (with no errors the original performance was 49.09 dB). Increasing the error to 4 LSBs (or 27.5 mV) in X5 farther reduces the dynamic performance to 40.34 dB (or approximately 9 dB).

Next, we examine the harmonics caused by the same (1 LSB) error in the comparator making the LSB decision – X33 in figure 6 - 14. With this error

we obtain a **SINAD = 43.66 dB** at 781.25 kHz which is a reduction of approximately 6 dB relative to the nominal case. Now the harmonics limiting the dynamic performance are third, fifth and seventh harmonic and are due to a much higher frequency components due to the frequency multiplication.

These examples illustrate that the folding A/D is typically characterized by very smooth linearity transitions and therefore the harmonics produced are low order tones. Given however the sinusoidal approximations of the trigonometric function in the front end, many harmonics are produced. This latest comment is a direct result of a mathematical fact: the Bessel function is produced by SIN[sin(Vin)]. The Bessel function is known to have quite an extended frequency spectrum.

SUMMARY

- The folding A/D converter presented in this chapter has substantially the same dynamic performance as a flash converter. Its power dissipation and input capacitance are significantly lower however than the flash, resulting in a converter that is easier to use in a system.

- One of the important benefits of the architecture is the limited number of comparators required for the converter. This results in impressive dynamic performance while keeping the amount of silicon real estate to a minimum.

- The architecture presented relies on the four-quadrant multiplier as the simplest means of obtaining zero crossings from the preceding zero crossings. While other methods were mentioned the author feels that the four quadrant trigonometric method is more advantageous and more error forgiving, resulting in a more robust converter.

- A Spice model has been suggested for the various blocks of a folding A/D converter that enables the user to predict various error contributions to the overall dynamic behavior. This model constitutes a means of evaluating the relevant parameters affecting the folding A/D dynamic performance without compromising the physical behavior of the circuit.

- An important parameter affecting the folding A/D accuracy is the optimization of the reference spacing for the CDP block. This is accomplished with the following degrees of freedom:
  - gain of the CDP
  - amount of degeneration in the differential pairs for improved linearity

- Although it seems that the above parameters are independent, there are limitations associated with the gain-bandwidth product of the block as well as the overall approximation to a sinewave needed for the SINCOS block.

- The accuracy in rendering the sinusoidal signal in the CDP depends to a large extent on the reference levels. To achieve the best accuracy in producing the sinusoidal function two dummy stages need to be

added at the full-scale range extremes. These stages improve the sinusoidal approximation in the area of interest for the converter's input.

- A similar approach as in the CDP is used on the interpolation cell. Again similar degrees of freedom are available as outlined for the CDP. For a practical and robust folding A/D front end the gain of those blocks should be limited to a gain of no more than two or three so no significant loss occur for the bandwidth due to frequency multiplication.

PROBLEMS

1. The CDP model in the chapter uses the equation

$$I_{out} = I_{tail} \cdot \tanh\left(\frac{Vin}{0.10}\right)$$

Change the amount of degeneration from 0.1 V to 0.2 V in the equation. This is equivalent to higher degree of local feedback in the differential pair. Therefore better linearity can be observed. Sweep the analog input voltage to the CDP and optimize the reference voltages at its input to obtain the most favorable references for the SINCOS block.

2. With the CDP block and reference voltages in problem 1 find the error of the SINCOS block from a perfect sinewave.

3. As seen in the chapter and the previous two problems the resulting converter full scale will most likely have odd-valued end points (in the example 0.22 to 1.98 V!). To resolve this issue, scaling of the analog input voltage can be done in an amplifier preceding the SINCOS block. Alternately the same result can be accomplished in the track and hold amplifier. Perform the scaling required to obtain a converter with full-scale range of 2.048 V which result in an **LSB=8mV**.

4. Change the comparator model in figure 6 – 13 to obtain a slew-rate limit of 2V/nsec for both positive and negative directions. Observe the resulting TDE for the device presented in figure 8-14.

Chapter 7

Pipelined A/D converter

1. INTRODUCTION

This chapter describes the behavior of a class of converters known as serial pipeline A/Ds.

The topology was developed with the intent of achieving high conversion rate with moderate hardware complexity by splitting the conversion task into smaller operations. At any time during the conversion process, the first stage operates on the most recent sample of the input while the following stage operates on the residue from the previous sample. This is analogous to an assembly line where each station carries out a different activity resulting in reduced overall assembly time. The conversion time of a pipeline A/D is governed by the slowest operation along the signal path (similar to the time the unit spends in the assembly line).

The serial pipeline A/D converter has similarities with the interleaved A/D described in chapter 4 but its operation is different in that the subranging blocks perform their operations in series along the converter chain rather than in parallel.

A two-stage pipeline converter is illustrated in figure 7-1. Although figure 7-1 seems similar to the subranging converter of figure 4-1 there is a major difference between the two figures. In figure 7-1 the residue obtained after the coarse conversion and the subtracting amplifier is transferred to a second track-and-hold amplifier. THA#2 increases the overall speed of the

converter by allowing the coarse A/D to start a new conversion as soon as THA#2 sampled the residue voltage. While the coarse converter output "goes through the pipe" the fine converter completes its previous conversion.

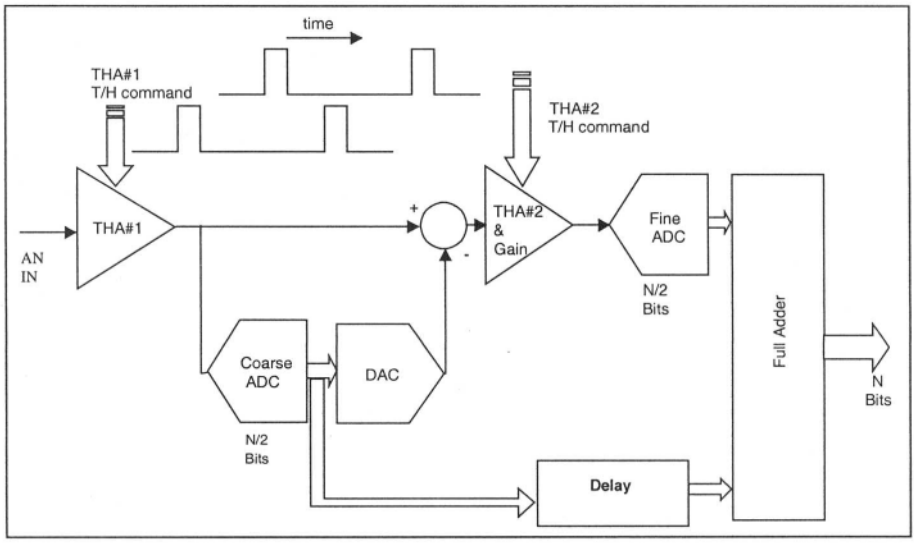


Figure 7-1. Pipeline A/D converter - general topology

THA#1 in figure 7 – 1 samples the analog input, followed by a coarse A/D converter. The coarse converter obtains the first N/2 bits and its output is converted back into an analog (quantized) signal using a D/A. The D/A required for the operation has only N/2 bits of resolution but N bits accuracy. The quantized signal is then subtracted from the THA#1 output and passed along to THA#2. After THA#2 completes its acquisition, THA#1 begins to acquire a new input signal, and the residue is converted simultaneously by the fine A/D, completing the conversion process with the next N/2 bits. The digital code from the coarse converter is delayed by the delay unit for a period equivalent to the time required for the residue signal to be processed by the difference amplifier, THA#2 and the fine A/D. After the time alignment, the coarse and fine digital codes are combined in the synchronization & adder unit to be presented to the output bus.
The process just described is in essence a mechanism of segmenting the full-scale range of the input signal with a coarse approximation and then refining the code with the help of the fine converter as illustrated in figure 7-2.

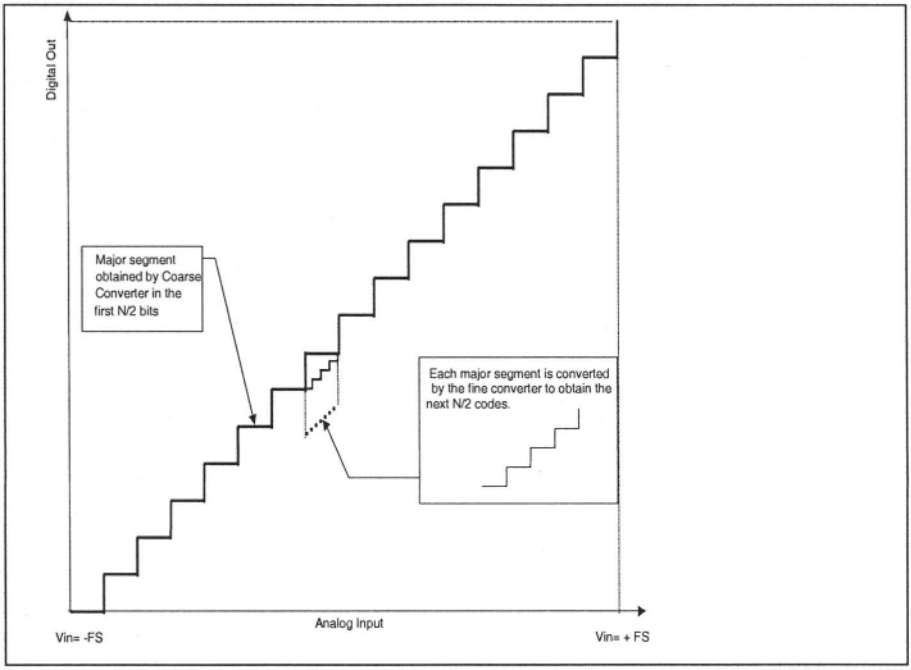


Figure 7-2. The coarse and fine converters have to fit their segments along the transfer curve

Only **30 (= $(2^4-1)\cdot 2$)** comparators are needed when implementing an 8-bit converter using this topology, whereas a flash architecture requires 255 **($=2^N-1$)** comparators for the same functionality. This reduces the complexity of the overall converter resulting in reduced device area as well as reducing the power dissipation.

When employing multiple A/D in the pipeline it is expected that mismatches exist between the various components. Therefore a correction algorithm must be used to mitigate the issue of segment matching. The correction method modifies the composite output code at the end of the pipe so the overall converter code can attain the expected accuracy.
Figure 7-3 illustrates the concept of segments matching for several cases:

- In 7-3 (a) the segments fit perfectly resulting in a perfect line with no deviation from one region to the next.
- In figure 7-3 (b) the even and odd regions do not overlap perfectly resulting in jumps from one region to the next. These abrupt steps cause non-uniform code widths associated with harmonics in the frequency domain.
- A correction algorithm is usually employed to alleviate the problem as outlined in figure 7-3 (c): the end of each segment is overlapped with the beginning of the next one. In doing so an area of redundancy is obtained

such that the coarse and fine converters are forced to match each other by forcing the more significant bits to align to the lower rank decision. The fine converter MSB decision is usually added to the LSB decision of the coarse converter, and the combined result is the output code of the A/D (this will be shown with the help of an example).

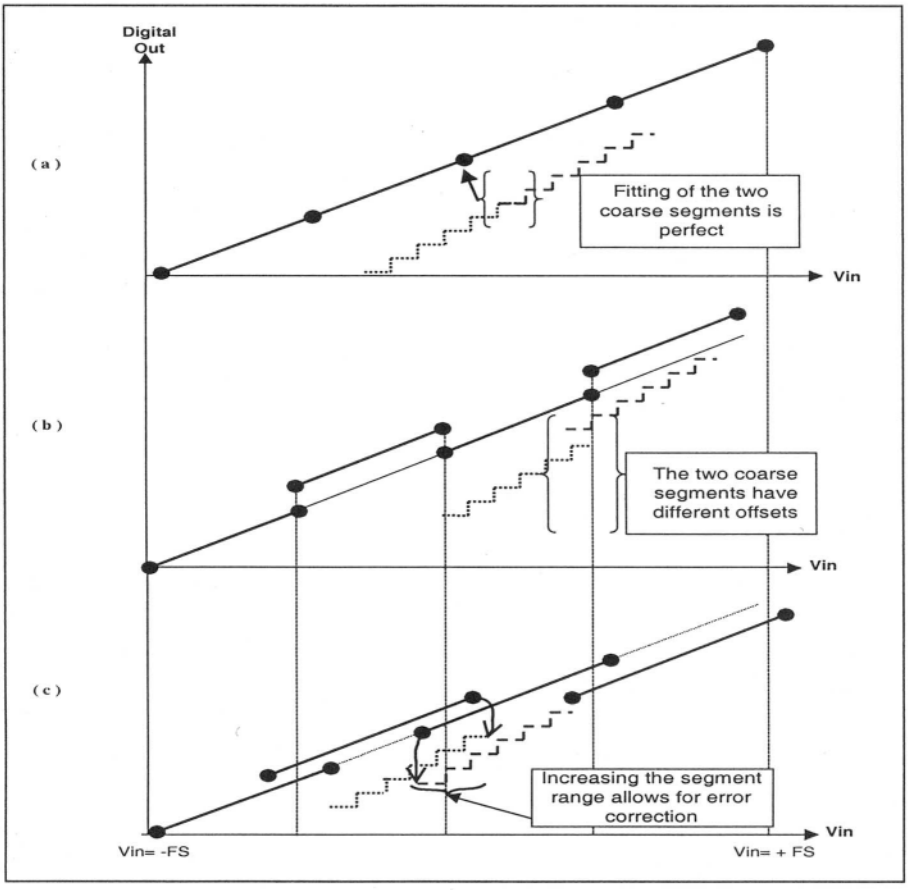


Figure 7-3. Coarse segments have errors at boundary. Adding correction range can alleviate the matching

Next, we take a look at the "signatures" of various error sources in the overall behavior of the composite A/D. Some common errors are illustrated in figure 7-4 in the time domain.

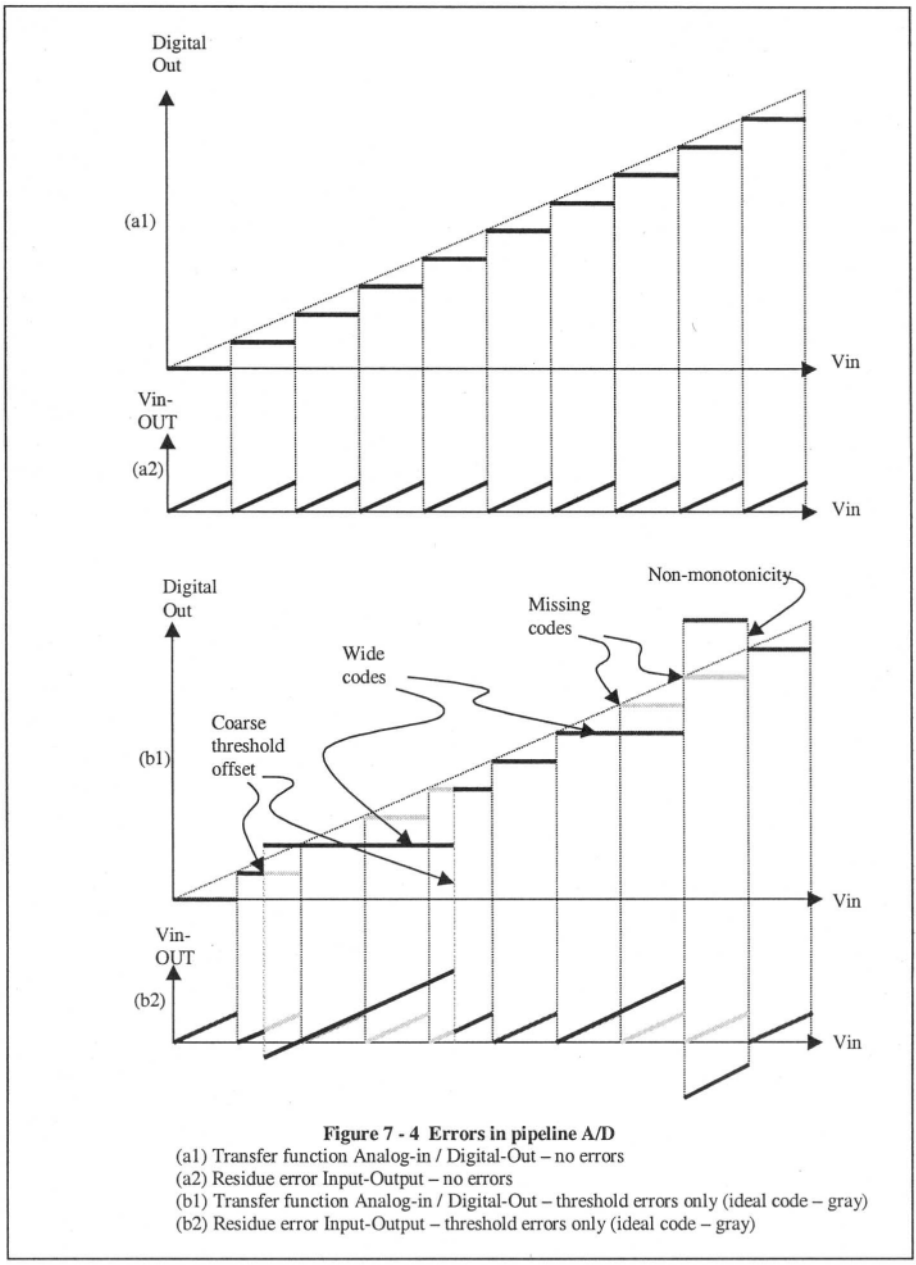


Figure 7 - 4 Errors in pipeline A/D
(a1) Transfer function Analog-in / Digital-Out – no errors
(a2) Residue error Input-Output – no errors
(b1) Transfer function Analog-in / Digital-Out – threshold errors only (ideal code – gray)
(b2) Residue error Input-Output – threshold errors only (ideal code – gray)

Figure 7-4. Errors in pipeline A/D

Figure 7-4 (a1) illustrates the transfer function of a perfect A/D; in figure 7-4 (a2) we see the difference between the analog input and the

reconstructed output (the result of a D/A output whose input is the digital code of the A/D). Figure 7-4 (b1) shows a similar transfer curve as (a1) but for a converter that has linearity errors. In figure 7-4 (b2) we see the residue of the reconstructed code for the converter in (b1).

When the coarse A/D has its threshold shifted from the ideal case by an offset or gain error, a code of the wrong width results. When a code width becomes zero the result is a missing code. Furthermore, when the residue becomes too negative the result is a non-monotonic behavior of the A/D. The errors shown in figure 7-4 can be corrected by employing digital correction schemes (as we will show shortly). To be able to correct the output code, the D/A as well as the residue amplifier must be linear to the overall converter accuracy (N bits). If this requirement is met, the residue consisting of the difference between the input and the coarse A/D output is accurate and no information is lost in spite of coarse converter inaccuracy (offset, gain or nonlinearity). Let us elaborate on the last point: reinspecting figure 7 – 2 we find that the fitting of the coarse and fine segments makes a very subtle assumption - that is that *the coarse quantizer in the chain is accurate to its expected number of bits (= N/2).* If this is not the case then the residue voltage can overrange the fine converter resulting in the wrong output code (as it will be illustrated by example). The error in residue can be the results of a static (offset, gain or linearity error) or dynamic (incomplete settling) mismatch in fitting the coarse to the fine segment. The mechanism defining how the pipeline converter performs dynamically is controlled by the behavior of the residue voltage or stated differently by the matching of the fine/coarse segments.

When the pipeline algorithm is employed, the accuracy required for the various quantizer blocks in the chain is not trivial. This subject is a major topic considered in reference 53. Without going into elaborate details, we review now its conclusions.

1.1 Component accuracy requirements in a pipeline A/D

From our discussions in chapter 1, we realize that the quantization process gives rise to spurious tones. As the quantization step is reduced (increased converter accuracy), the spurious components are decreased. Therefore, increasing the first quantizer resolution by one bit, results in halving the error amplitude and doubling the number of quantization steps across the full-scale range. The consequence is a 6 dB smaller error spread across twice the bandwidth with resulting spurs that are reduced by 9 dB (as explained also in chapter 1). Another way of looking at this is as follows: the increase in the resolution of the first converter reduces the coarse to fine discontinuity step size in the transfer function. Diminishing the number of

discontinuities of segment fitting results in a lower energy at the mismatch of adjacent sections and therefore enhances the converter dynamic performance. Since the residue constitutes a potential error source, it stands to reason that increasing the number of bits in the first quantizer results in improved SFDR for the overall converter. A side benefit of the increase in the coarse A/D accuracy is that it also reduces the speed and accuracy demands on the residue amplifier.

The matter of residue D/A and error amplifier has been stated repeatedly in terms of correction algorithm effectiveness. Fundamentally the D/A errors can be corrected only through trim or calibration of the D/A (see references 57, 59, 60, 61, 62, 63, 64, 65, 66) and this process needs to take place prior to the A/D conversion process.

Without limiting the generality of our examples in this chapter, we assume that the D/A converter is accurate to N bit (presenting therefore the principle of error correction after the calibration process is complete).

2. ERRORS IN PIPELINE A/D

To understand what happens in a pipeline A/D without error correction our first example examines an 8-bit A/D employing two 4-bit converters. The converter considered is shown in figure 7 – 5. The figure shows a coarse 4-bit flash converter - X1 as the first quantizer in the pipeline. The device employs a reference of 2.56V for its ladder resulting in an LSB of 160 mV **(=2.56/2^4).** Its digital output code is converted back into an analog voltage with the help of a D/A – X9 and subtracted from the analog input using the residue amplifier modeled as a voltage controlled-voltage source E_{dac}.

The D/A converter – X9 - is accurate to the overall accuracy of the final converter as suggested above. In other words, although the DAC has 4 bit of resolution it is accurate to the 8-bit level. Next, the residue voltage resulting from the subtraction is gained up by a factor of 16 with the help of $E_{residue}$. Prior to the amplification however, the residue signal is being held with the help of THA#2. As usual we model the THA in figure 7 – 5 with the help of switch **(S_2)** and hold capacitor **(C_2).** By amplifying the error signal (SHA1 – DAC coarse), the accuracy requirement of the fine converter is relaxed by a factor equal to the interstage gain (in our example 16x). This means that any error in the fine A/D when referred to the input of the composite A/D is divided by the gain of this amplifier.

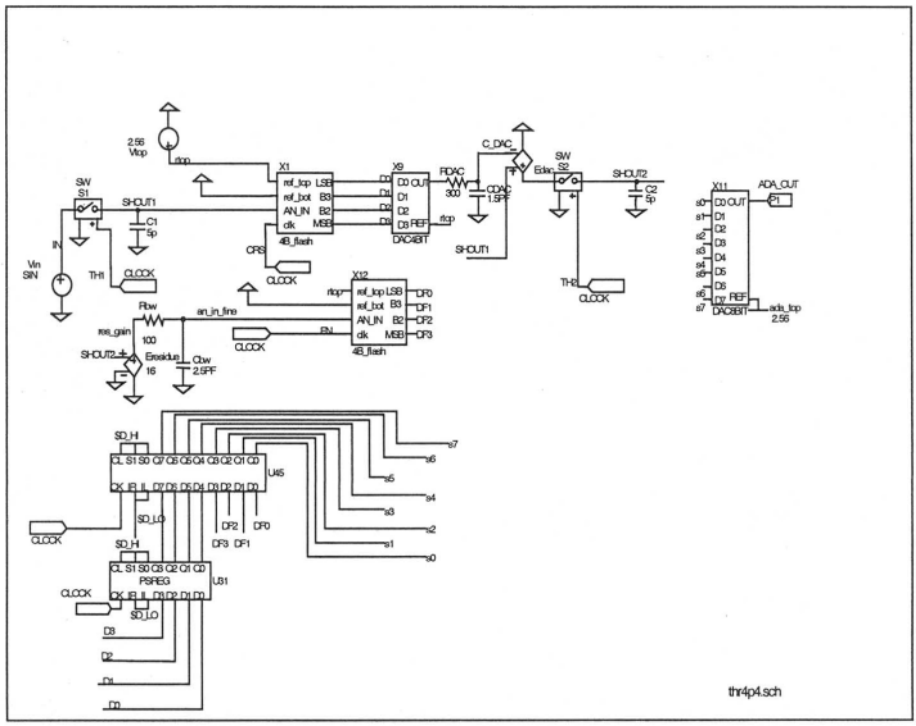


Figure 7-5. Spice schematic for an 8-bit converter obtained from two 4-bit converters (no error correction)

For this reason the fine converter – X12 - uses the same references (0 to 2.56V) in figure 7 – 5 as the coarse A/D. The result is that the fine LSB has a magnitude of 160 mV (referenced to its own input). Given the amplification of the residue amplifier, the fine LSB magnitude referenced to the composite A/D input is **10 mV (=160mV/16).**

The dynamic limitation of the converter in figure 7 – 5 is modeled by limiting the bandwidth of the THAs, D/A and residue amplifier with simple R·C time-constants ($\mathbf{R_{DAC}}\cdot\mathbf{C_{DAC}}$ for the D/A and $\mathbf{R_{SW}}\cdot\mathbf{C_{1,2}}$ for the THAs). We will elaborate on the dynamic limitation later in the chapter. Finally, to accommodate the delays associated with the propagation delays of the coarse A/D, DAC and error amplifier the digital code is aligned with latches U31 and U45.

The timing diagram for the converter is illustrated in figure 7 – 6. First THA#1 acquires the input signal in the interval 1-4 nsec. As soon as THA#1 completes the acquisition, the coarse converter starts its conversion, which lasts for the next 0.5 nsec. At the end of coarse conversion U31 latches the coarse digital code at time 4.5 nsec. At the same time the second THA starts

its acquisition of the residue which is completed at 9 nsec (allowing for the D/A and error amplifier to settle). At this time the fine conversion is ready to begin. Its digital code gets latched at 9.5 nsec with the help of U45 in figure 7 – 5. The entire cycle gets repeated every 10 nsec.

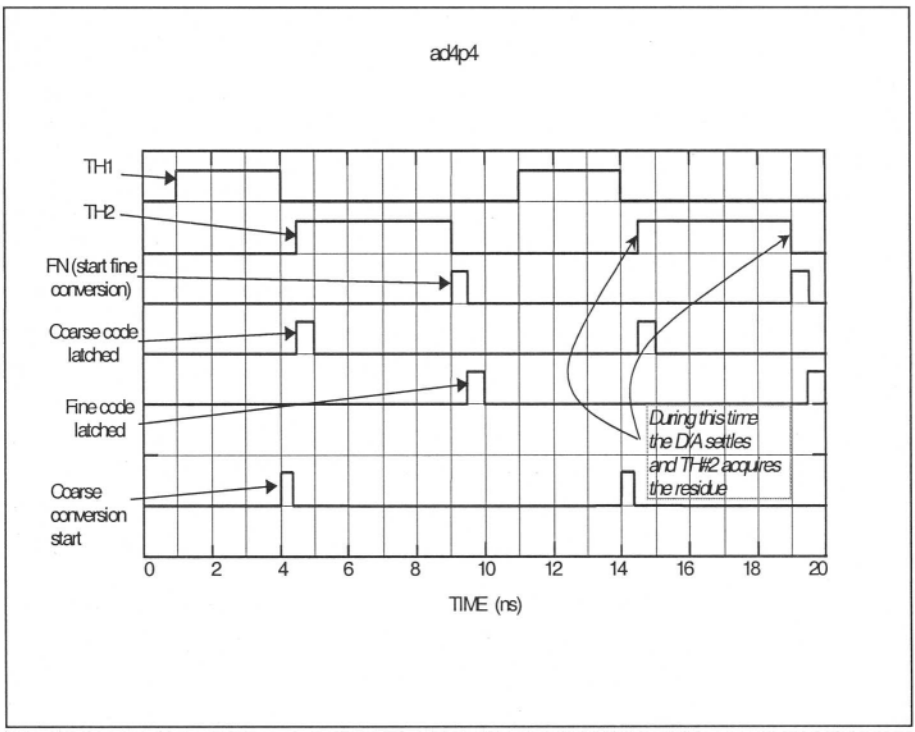


Figure 7-6. Timing diagram for the converter in figure 7-5

With the pipeline converter model of figure 7 - 5, we can proceed to test the dynamic performance of the composite converter while none of its components cause errors. Using a 512-point FFT and a sampling rate of 100 MSPS, we test the converter's performance at an analog input frequency of 195.3125 **kHz (= 100 MHz/ 512).** The spectrum is illustrated in figure 7 - 7 (a) and the SINAD is calculated to be 48.34 dB **(ENOB=7.73 bit).** This is a reasonable performance due to the expected round-off error and for the accuracy of the components used.

2.1 Errors in a practical pipeline A/D

As in previous chapters, in the following sections we introduce the kind of errors we would expect in a practical pipelined converter. To understand

the impact of each error to the overall performance, we simulate the behavior of each error mechanism separately.

First, we introduce a coarse converter **offset error**. This is accomplished by adding two equal voltage sources in series with ref_top and ref_bottom of X1 in figure 7 - 5. By making these error voltages equal to 80 mV we shift the threshold of each comparator of the coarse A/D by 0.5 LSBs. Using the same conditions as before we obtain a SINAD of 29.7 dB or **ENOB=4.635** bit. This performance is a far cry from an 8-bit converter and is due to the fact that the fine A/D is overranged for approximately half the time due to the coarse converter offset. Stated differently: as soon as the input signal to the fine converter exceeds its input range the fine converter produces a constant output. As a result, the coarse and fine segments do not match and the discontinuities produce frequency spurs. The frequency spectrum for this case is illustrated in figure 7 – 7 (b).

Next, we examine the effects of **gain error** in the coarse converter. This effect is modeled by using a top reference of **2.64V (=2.56V + 0.08V)** and keeping the bottom reference at zero. Now each of the comparators in the coarse converter has its threshold increased by 3.125% **(=2.64/2.56).** As a result each comparator reference has its threshold shifted upward by 3.125% from the original location until the topmost comparator has its decision moved by 0.5 LSBs. Now the fine A/D will spend less time being overranged, as was the case for the offset error. The dynamic performance measured now is **SINAD=35.17 dB** or an **ENOB=5.543 bits.** The frequency spectrum is illustrated in figure 7 – 7 (c).

Although the peak amplitudes of the offset and gain errors are the same in our examples, the resulting harmonic content is vastly different as figure 7 – 7 reflects. Not only is the total dynamic error (TDE) different for these cases, but also the spectrum shows much higher spurs for the offset error than for the gain error. Although it is difficult to predict intuitively the harmonic locations caused by the converter' errors we will attempt to explain the simulation results. Table 7.1 summarizes some of the FFT highest spurs in each simulation and the highest tones for each case are emphasized (gray).

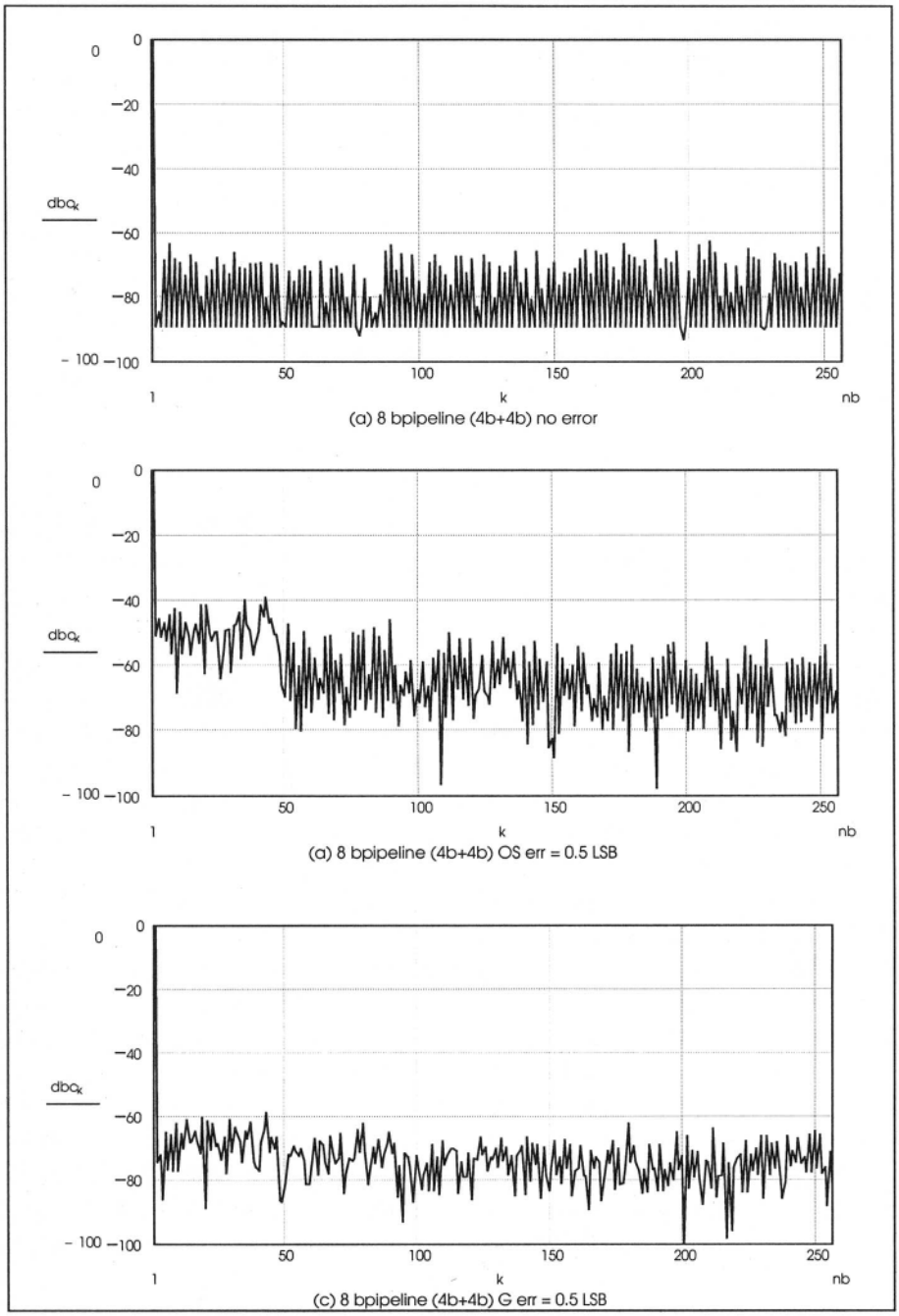


Figure 7-7. Dynamic performance of 8-bit pipeline A/D - (a) No error (b) 0.5 LSB offset error (c) 0.5 LSB gain error

Table 7-1. Major frequency spurs for 8-bit pipeline converter without error correction

8 bit pipeline with **no error correction** (highest bin shaded in each column)			
Harmonic bin	No error	0.5 LSB Offset error	0.5 LSB Gain error
19	-79.09	-48.26	-53.12
21	-72.51	-75.54	-52.78
25	-66.44	-46.32	-50.90
27	-70.37	-68.58	-52.50
29	-72.88	-47.19	-70.62
31	-66.18	-54.88	-49.29
33	-70.75	-45.98	-66.90
37	-72.23	-45.75	-57.39
41	-67.70	-55.04	-48.53
43	-77.80	-49.58	-49.10
47	-69.59	-52.00	-53.91

For the offset error, we expect to have an abrupt change in codes at the transitions from one coarse segment to the next. This discontinuity occurs sixteen times on the positive slope and an equal number of times on the negative slope. For this reason we would expect harmonics, which are multiples of 32. Since the fundamental resides in bin 0, the thirty-first bin will have this harmonic. Given the small number of frequency bins and the low frequency resolution we find that bins 29 and 33 have similar size harmonics and are in close proximity to bin 32.

For the gain error case, we would expect the harmonic content to be lower than for the offset case. The main reason is that in the case of offset error the deviation from the ideal case is uniform and equal to 0.5 LSBs at the transition from coarse to fine A/D. In the gain error case however, the deviation is gradual from 0 LSBs to 0.5 LSB (at the full scale of the coarse converter). The average value of the error is therefore twice as big in the offset case as in the gain error case. Here again, as in the previous example there are 32 discontinuity points on the sinewave at the transition between the coarse and fine segments. This is reflected in the high spur in bin 31, which is in line with our prediction.

In summary our examples prove that the pipeline topology has stringent demands on the sub-blocks used for the composite converter. If these devices do not meet the overall converter accuracy then the mismatches can cause significant spurs in the frequency domain.

2.2 Error correction – an example

The mismatch of the quantizers and their relative accuracy examined above point to the need of correcting the coarse decisions further in the pipeline. With this in mind, we proceed now to examine a pipeline A/D with

digital error correction. The error correction is accomplished by employing one bit of overlap between the LSB of the coarse A/D and the MSB bit of the fine A/D. The model for this converter is outlined in figure 7 – 8.

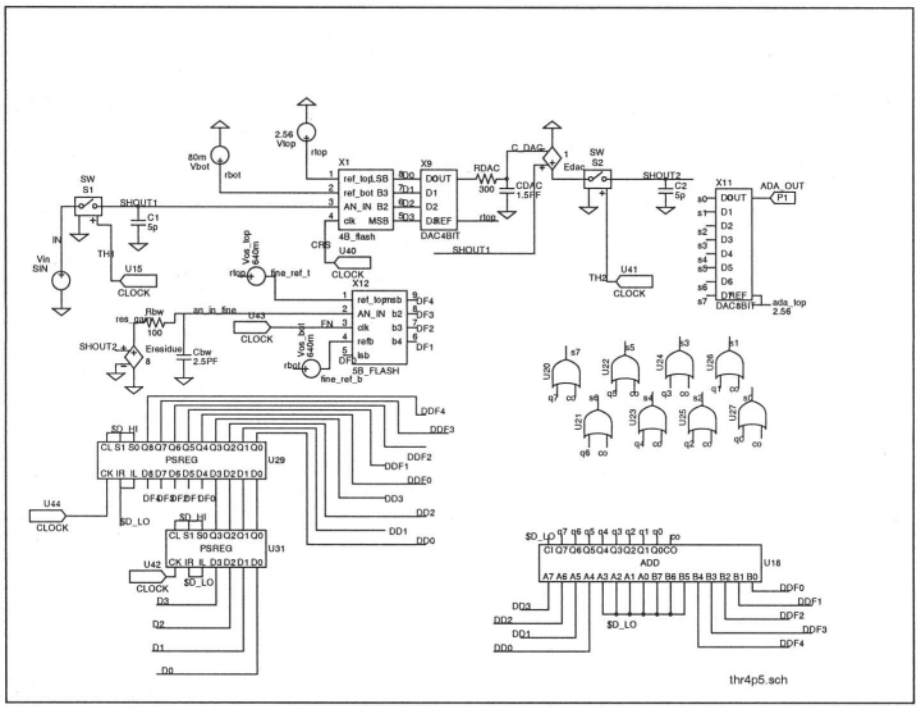


Figure 7-8. An 8-bit serial pipeline A/D with error correction (one bit overlap)

The error correction converter differs from the non-corrected version in several ways. Now, for the same 8-bit composite converter we use a 5-bit fine A/D for the second rank and the error amplifier has a gain of 8x. With the reduced amplifier gain, only half of the fine A/D range is used, while the other half is utilized for error correction. In order to accommodate both positive and negative errors, the references to the fine A/D are offset by one quarter of its FSR (=0.64V in our example). In our model this is accomplished by shifting the fine A/D references by -0.64V (this is equivalent to shifting the analog input by +0.64V). Now if the coarse quantizer makes an error in its decision we can correct its output code with the help of the fine converter provided that the D/A and the error amplifier do not contribute errors.

After lining up in time the coarse and fine digital codes with the help of latches U29 and U31 the fine code is added to the coarse code and the LSB

of the coarse is overlapped with the MSB of the fine converter. This is accomplished with the full-adder U18.
To avoid a roll over of the output code (one LSB count over the + full-scale causing an output to roll to the origin - 0.... 00) the OR gates U20 through U27 are added.

As we reiterated so many times this method corrects for errors of the coarse A/D but not for D/A or residue amplifier errors. The concept is illustrated graphically in figure 7 – 9.
On the left side of figure 7 – 9 we see the coarse A/D codes with their associated threshold levels. After the 8x amplification and addition of ¼ scale (0.64V) of the fine A/D we see that the coarse LSB span is projected into the active range of the fine A/D (0 to 1.28 V). By reducing the active area of the fine A/D we allow the residue voltage to extend above 1.28V and below 0V but require it to stay above –0.64V and below 1.92V. The allowable correction range is shown in figure 7 – 9 on the right side. At the top of the figure we can see the reference voltages required for each of the A/D to obtain the error correction.
Now the converter can deal with errors up to one bit in magnitude (+ and – 0.5 LSB for a range of 1 LSB total). The "cost" paid for achieving this feature is in the increased resolution requirement of the fine converter with the associated power and area expansion.

Let us return briefly to the requirement of D/A converter accuracy, which was reiterated through the chapter. This can be understood by examining figure 7 – 10, where we see the output voltage of the D/A after the coarse A/D reconstruction. The figure shows two cases of gain error in the signal path: a coarse A/D error (figure 7 – 10 a) followed by a reconstructing D/A gain error (figure 7 – 10 b).

Assume for example that the coarse A/D has a gain error of 10% (gain=l.l). This means that each of the comparator thresholds is shifted up by 10%. If this error does not overrange the fine A/D, it will result in a fine A/D output code which when added to the coarse A/D will essentially correct the threshold errors of the coarse A/D. On the other hand, if the coarse A/D has its thresholds at the nominal place but the D/A has a gain error of 10%, the residue amplifier will have the same input as in the previous case. Again, if we assume that the fine converter is not overranged, its code will try to correct the residue and will move the overall A/D code erroneously (remember that the coarse A/D had the correct code and the error in residue was due to the D/A gain error). Looking at figure 7 – 10 we observe that the saw-tooth residue voltage in front of the residue amplifier is a good indicator of whether the error is the result of gain error in the coarse A/D or the reconstructing D/A.

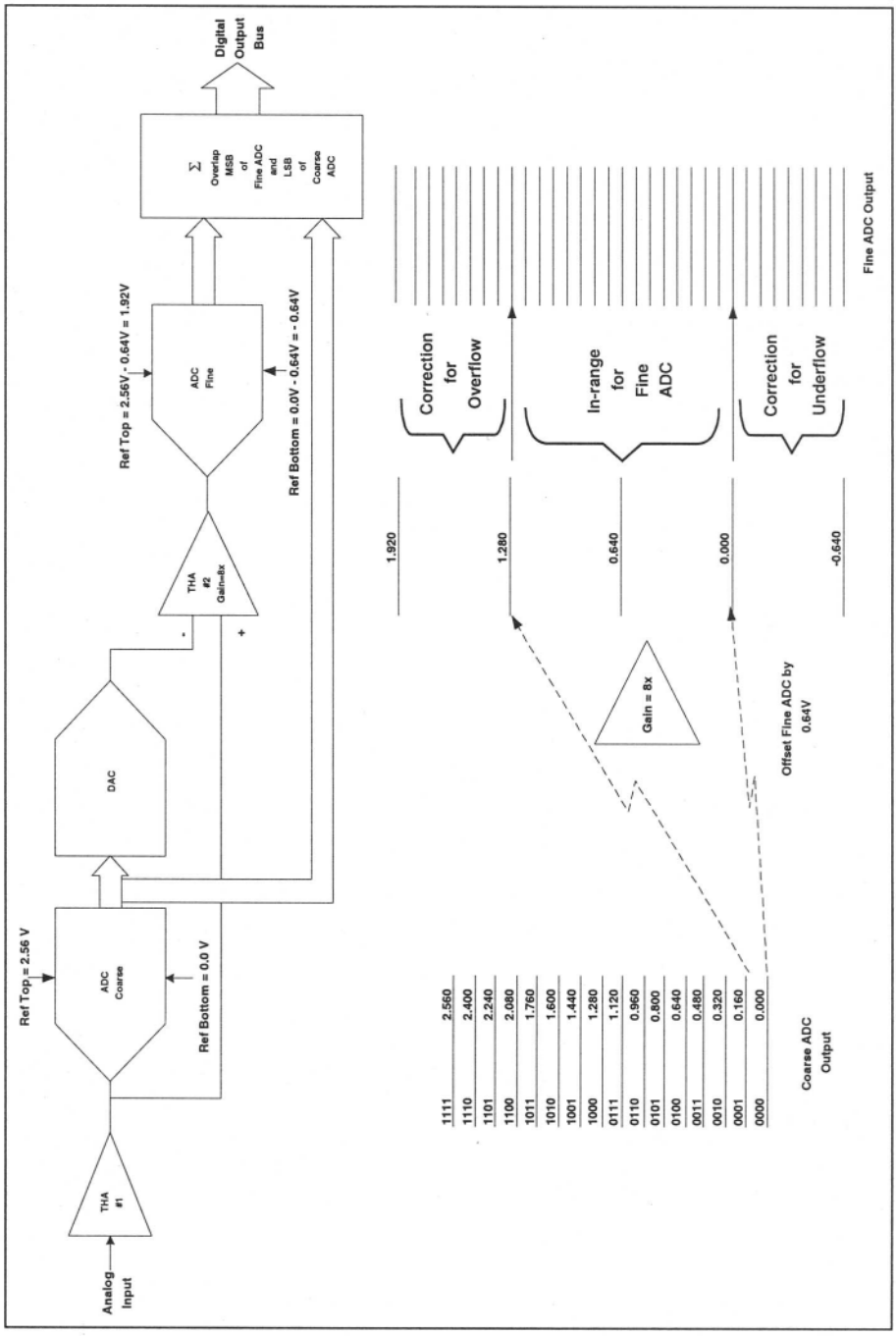


Figure 7-9. Pipeline A/D and the concept of error correction with one bit of overlap

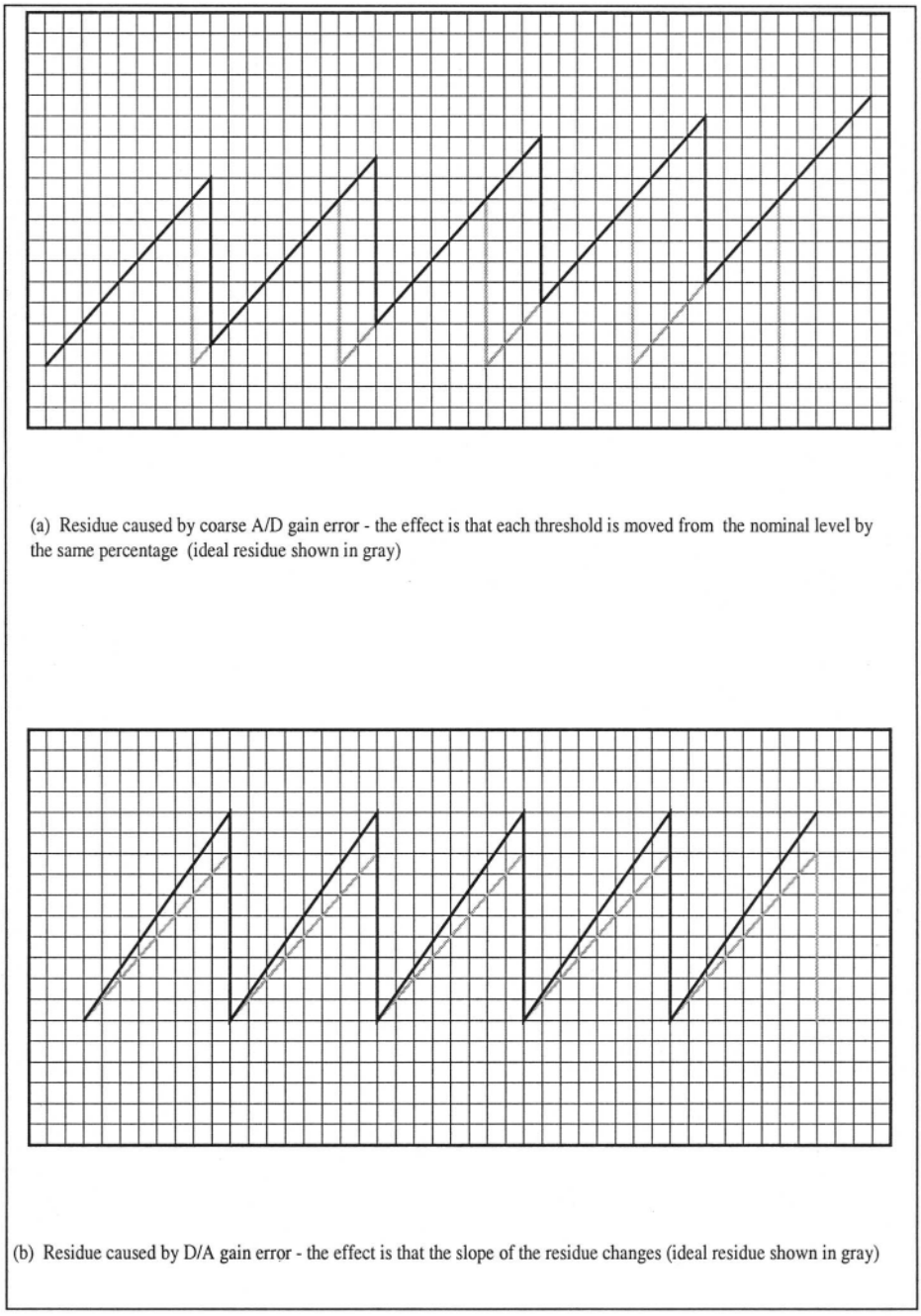


Figure 7-10. Gain errors in pipeline A/D - (a) if caused by the coarse A/D - can be corrected by the digital error correction and (b) if caused by the reconstructing D/A - result in overall linearity error of the composite A/D

The sharp portion of the saw-tooth occurs at equal distance corresponding to the ideal threshold locations if the gain error belongs to the D/A. If however the saw-tooth has increasing length on its x-axis, then the gain error belongs to the coarse A/D. Phrased differently, if the residue voltage is a saw tooth with the same height and equal slopes of the tooth then the error originates in the D/A. If however the residue is tilted in the overall amplitude then the error originates in the A/D. This is the underlying principle used in error detection and correction algorithms.

Now, lets take a brief look at the way the dynamic limitation for various components of the circuit are modeled in figure 7 - 8. The main reasons restricting the conversion time are the acquisition time of the two THAs employed, D/A settling time and bandwidth limitation of the residue amplifier. These effects are modeled in a similar manner for both the corrected and uncorrected converter: each of the THAs has a 5 pF holding cap and a switch resistance of 100 Ω resulting in a bandwidth of approximately 320 MHz ($=1/[2\cdot\pi\cdot5\cdot10^{-12}\cdot100]$). The D/A bandwidth is modeled with the help of a single pole roll-off - R_{DAC} = 300 Ω and C_{DAC} = 1.5 pF resulting in a bandwidth of 350 MHz ($=1/2\cdot\pi\cdot1.5\cdot10^{-12}\cdot300$)or a time constant τ = 0.45 nsec.

- The residue amplifier has its pole at τ = 0.25 nsec. This is accomplished with a resistor R_{BW} = 100 Ω and a capacitor C_{BW} = 2.5 pF.
- Finally both THAs have time constants τ = 0.5 nsec resulting from the switch resistance of 100Ω and holding caps of 5 pF.

The time required by THA#2 to acquire its output voltage is:

$$\tau_{eff} = \sqrt{\tau_{DAC}^2 + \tau_{THA}^2} = \sqrt{0.45^2 + 0.5^2} = 0.6727 n\text{sec} \qquad 7.1$$

Since the residue needs to be accurate to 8-bit level we need to allow it to settle for a time equal to:

$$\tau_{eff} \cdot \ln(2^8) = \tau_{eff} \cdot \ln(256) \approx 3.73 n\text{sec} \qquad 7.2$$

This is within the allotted time for the THA#2 to acquire the residue voltage (4.5 nsec in our implementation) and is reasonable for a 100 MSPS converter.

The residue voltage is next amplified by a factor of 8x with a time constant of 250 psec (an equivalent bandwidth of 640 MHz). The input to the fine A/D needs only be accurate now to 4-bit accuracy and requires an amount of time of:

$$0.25 \cdot \ln(2^4) \approx 0.693 n\text{sec} \qquad 7.3$$

Let us examine the dynamic performance of the error corrected A/D of figure 7 – 8. To be consistent both converters (corrected and uncorrected) are tested under the same conditions (f_s = 100 MSPS and f_{in} = 195.3125 kHz). With no errors in any of the blocks we measure a SINAD = 48.91 dB or ENOB=7.83. The spectrum is outlined in figure 7 – 11.

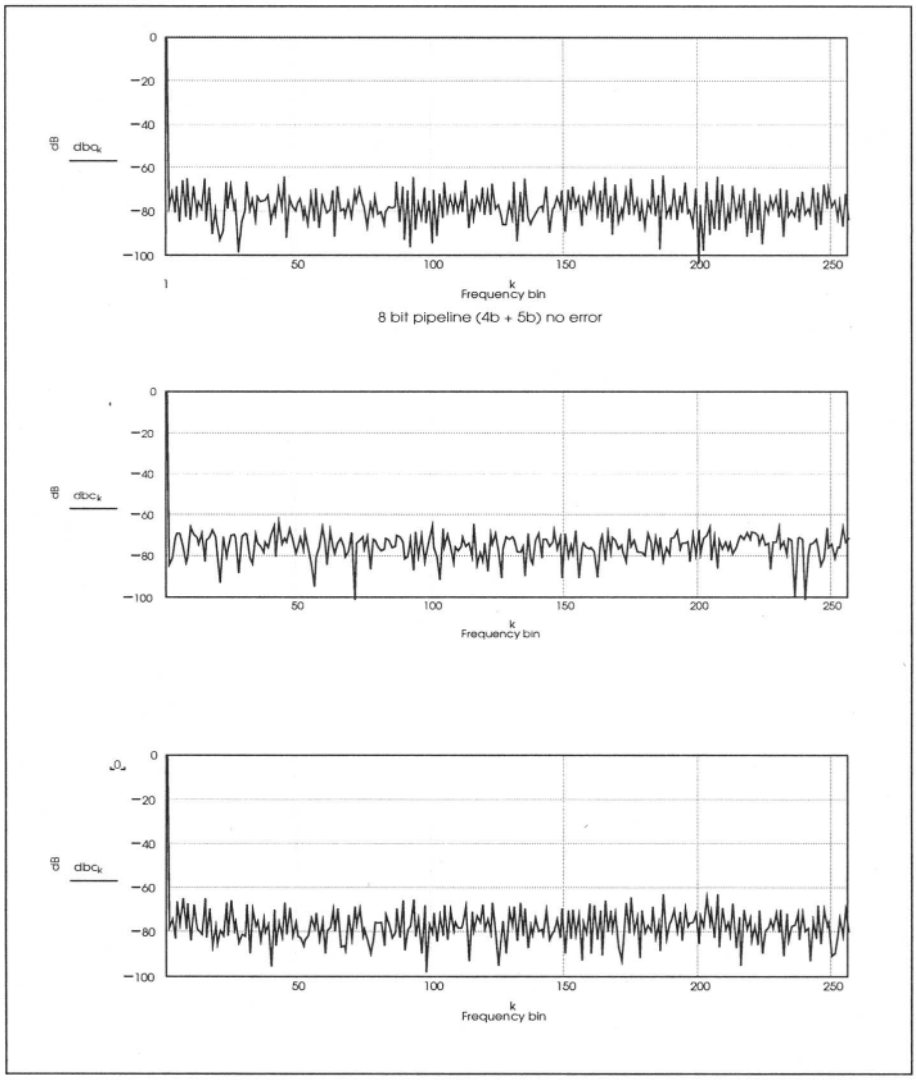


Figure 7-11. FFT plot of pipeline converter with 1-bit of overlap

Generally speaking the low frequency spectrum is an indication of the converter ability to digitize the signal when no major transients occur and it's by and large a hint of static errors. The frequency spectrum for this case is shown in figure 7-11 (a).

Next we consider the case where the coarse A/D has an offset error of 0.5 LSB (=80 mV=160mV· 0.5). The difference between the input (THA#1 output) and the reconstructed D/A is now 80 mV and is strictly caused by the change in threshold of the coarse A/D. Again this is modeled with both the top and the bottom references of the coarse A/D shifted with two equal voltage sources of 80 mV (ref_top=2.64 and ref_bottom=80mV). At the output of the residue amplifier, this is equivalent to 640 mV (=80mV· 8). This is now within the active range of the fine A/D and therefore by digitally adding the MSB of the fine A/D with the coarse A/D LSB we get a correction of the output composite code. The same argument can be made on gain or linearity error of the coarse converter or any combination of errors in the coarse A/D that do not overrange the residue amplifier or the fine A/D (a combined error of no more than one coarse LSB).
With an offset error of 0.5 LSBs in the coarse converter we obtain now a SINAD=48.297 dB or ENOB=7.724. This is significant improvement over the non-corrected case. The result is only 0.7 dB different relative to the same converter without any errors and the minor difference is due mainly to round-off error.

Next we examine the case of 0.5 LSBs of gain error in the coarse A/D. In this case the residue will resemble figure 7 – 10 (a) and each of its comparators threshold will be shifted by:

$$3.125\% = \frac{2.56V + 0.08V}{2.56V}$$

The resulting SINAD for this case is 48.95 dB (ENOB=7.83). Again we notice a big improvement over the uncorrected case.
The main harmonics for each case are summarized in table 7-2 below.

Table 7-2. Major frequency spurs for the 8-bit pipeline converter with error correction

8 bit pipeline **with error correction** (highest bin shaded in each column)			
Harmonic bin	No error	0.5 LSB Offset error	0.5 LSB Gain error
15	-65.23	-83.14	-65.93
25	-68.41	-70.16	-65.78
41	-77.88	-65.47	-70.16
43	-69.87	-63.55	-75.52
45	-64.14	-71.65	-66.90

The two pipelined converters examined prove conclusively that error correction can bring major improvements in dynamic performance. Although the errors considered in both examples were similar in value the correction algorithm reduced the harmonics to acceptable values for an 8-bit converter. We also confirmed that although each of the quantizers had an accuracy of 4 bit the overall converter corrected the output code with the assumption that the reconstructed residue did not overrange the fine converter.

2.3 Residue D/A error

Finally we examine the effects of gain error in the reconstructing D/A – X9 of figure 7 – 8 – or the voltage controlled-voltage source – Edac. This is the final proof of the strict accuracy requirements of the residue D/A. Assuming a gain error of 3.125% in the Edac we reexamine the dynamic performance under similar gain error as investigated for the coarse A/D. Performing the FFT test at the same frequency as before results in a SINAD = 32.825 dB. The major harmonics are summarized in table 7-3:

Table 7-3. 8-bit pipeline with residue D/A error

8 bit pipeline with D/A gain error	
Harmonic	dB
2^{nd}	-39.60
3^{rd}	-43.48
4^{th}	-47.42
5^{th}	-48.48
6^{th}	-48.37

This example illustrates the importance of D/A accuracy in a pipelined converter. The reconstructed output for this case is illustrated in figure 7 – 12 (a) in the time domain and 7 – 12 (b) in the frequency domain (FFT). With a relatively small gain error the time domain shows clearly how the error becomes worse on the positive peak of the analog input, resulting in a large second harmonic. In fact the first six harmonics are the main limitation in the converter's dynamic performance.

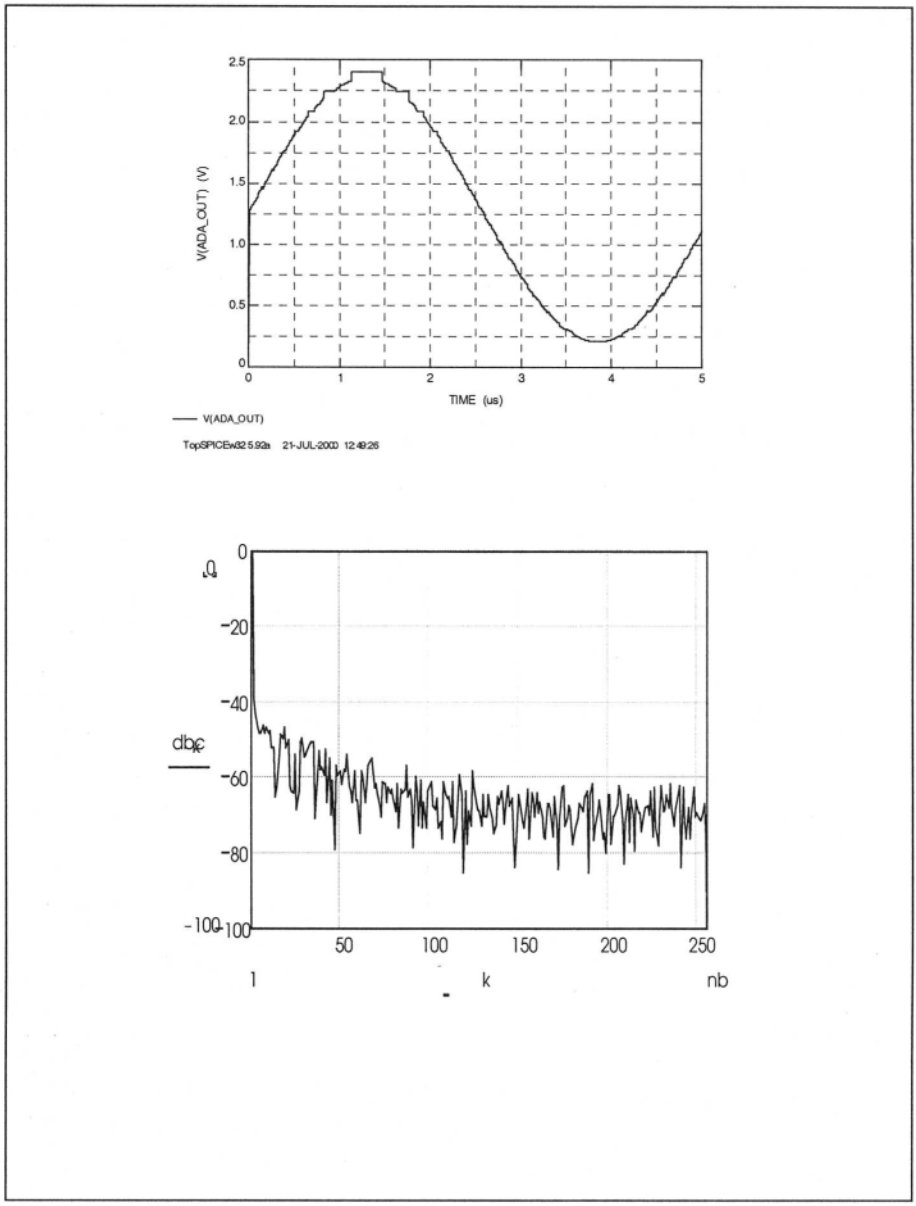


Figure 7-12. Pipeline A/D with 1-bit of overlap correction

SUMMARY

- The pipeline A/D converter improves the conversion throughput by employing additional THAs and processing the residue concurrently with the coarser stages.

- Using quantizers with limited accuracy, a more accurate converter can be obtained by amplifying the residue and overlapping the digital codes at the full adder's input.

- D/A and residue amplifier errors need to be either trimmed or calibrated in order to obtain the correct residue for the finer stages.

- An error correction algorithm using an overlap correction bit between the coarse A/D LSB and the fine A/D MSB can be employed to correct for threshold errors of the coarse converter. This can only be accomplished as long as the D/A and residue amplifier are linear and do not add their own errors.

- The digital error correction can be used to correct for threshold errors of the coarse A/D alone. If other errors exist between the coarse A/D and the error amplifier the result is that the fine converter will override the coarse decision erroneously resulting in the wrong pipelined code (se figure 7 – 10).

- Increasing the resolution of the coarse converter in a two-step pipeline converter improves the SFDR of the overall converter. More specifically an increase of one bit in resolution increases the SFDR by 9 dB. This guarantees less periodic errors in the overall converter (see chapter 1).

- The dynamic behavior of an offset error is very different than the dynamic behavior for a gain error in the coarse A/D. For similar peak error values, the offset error will generate higher spurs than the gain error.

PROBLEMS

1. Using the model of figure 7 – 5 find the timing signature caused by the following components:
   (a) Change the D/A gain by increasing Vrtop voltage by 0.5 coarse LSB (=80 mV).
   (b) Add 0.5 LSB (=80 mV) of offset to the D/A by adding a voltage reference in series with C_DAC node.
   (c) Change residue gain by changing the Edac value by 0.5 coarse LSB (=1+ (2.56+0.16)/2.56 = 1.03125)
   (d) Change residue amplifier gain Eresidue as in (c).
   (e) Change fine A/D offset (ref_top and ref_bottom of X12) as in 80mV.
   (f) Change fine A/D gain (ref_top of X12) to have an equivalent 0.5 coarse LSBs (1.03125).

2. Repeat problem 1 above using the error corrected converter of figure 7 – 8 and find whether the correction algorithm can correct these errors.

3. The error correction algorithm described in this chapter can improve the dynamic performance by correcting higher errors in the coarse A/D by increasing the amount of overlapped bits. What is the required gain of the residue amplifier if the expected coarse A/D error is 3LSBs.

4. Repeat problem 3 above when the coarse A/D error is 4 LSBs and show that using two overlapped bits results in error correction of up to 4 LSBs.

Chapter 8

Serial Pipeline A/D with 1.5-bit / stage

1. INTRODUCTION

In contrast to the topologies discussed so far in this book, the method described in this chapter does not rely on high accuracy comparators.

The 1.5-bit/stage architecture, is a unique implementation of the serial pipelined A/D converter. Each of the pipeline stages detects a single bit while using an error correction range of one-half-bit per stage. The technique has been described and used for many years (56, 57, 61, 67) as a means of obtaining high accuracy, high-speed converters without complex CMOS comparators, which are known to have high offsets. In recent years, with the advent of sub-micron CMOS geometry, the method gained increased acceptance and has expanded the resolution of this class of A/D.

The most common implementation of this topology is realized in CMOS technology. The method attractiveness stems from its simplicity and the fact that high-resolution converters can be implemented by repeated utilization of a simple stage. Given the compactness of the topology, high conversion speed can be achieved requiring relatively small die area.

The serial pipeline 1.5-bit converter is very similar in concept with the SAR A/D.

The scheme evolved from the SAR topology as follows: in the SAR A/D the first decision – MSB is based on a comparison between the analog input and half the reference voltage. Next bit (B2) makes a comparison between

the input and a voltage equal in magnitude to the first comparison plus one quarter of the reference and so on.. .(see chapter 5).
The same result can be achieved by doubling the residue and comparing it to a known voltage level so all stages in the pipe use the same reference. This concept is illustrated in figure 8 – 1.

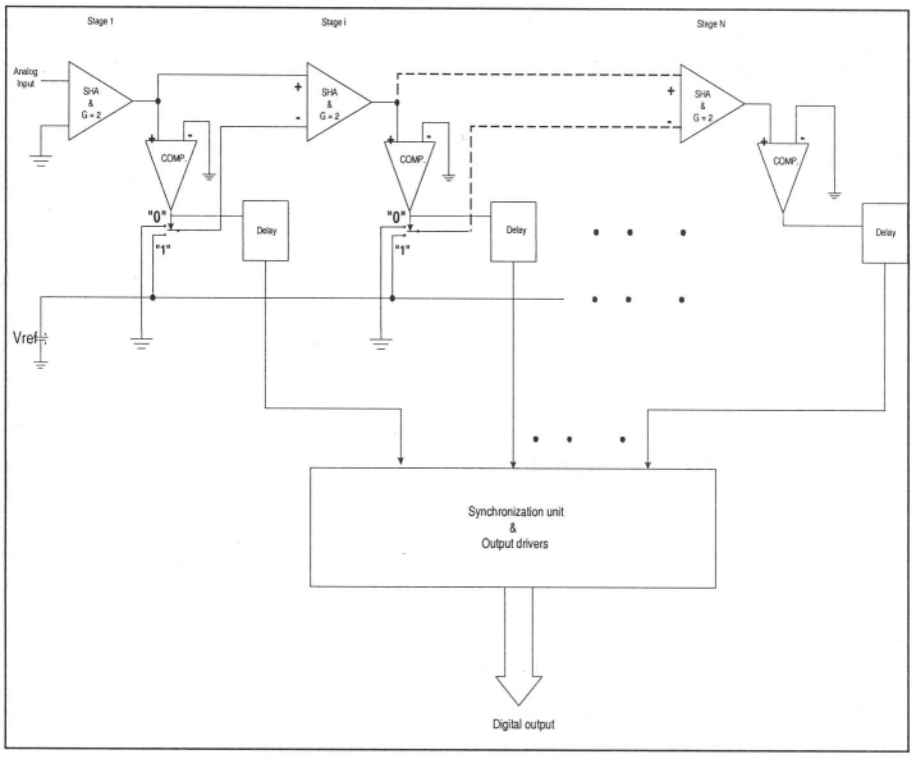


Figure 8-1. One bit / stage pipeline converter

The figure outlines the converter operation as follows: in the first phase, the SHA#1 samples the input signal followed by a comparison phase in the comparator/latch. In the next phase, the signal is multiplied by two, and depending on the comparison of comparator#1 the reference level or zero is subtracted and forwarded to the input of SHA#2 input. The algorithm continues in a similar fashion for each subsequent stage. After N samples the final residue reaches the last stage and the output of all comparators is available at the output drivers. Since each SHA in the chain has a delay of one-half cycle, it takes N/2 cycles until the conversion is completed. As explained in the previous chapter, any error such as gain, offset or nonlinearity in the present stage may cause the following stage to overrange and therefore compromise the converter's accuracy. Therefore an error

correction mechanism is required to mitigate the problem. One way of achieving the correction is to increase the active range of each stage allowing for a modification of its digital code. In the original SAR at the end of each bit comparison, we either added half of the previous reference to the accumulator or zero and once a bit was set no correction occurred.

An improvement to the SAR algorithm can be achieved by adding a correction scheme whereby each bit decision is allowed to have a certain amount of error, which can be corrected at the end of the conversion cycle. By introducing two reference voltages in front of the decision circuit (comparator) a comparison window is established, allowing the input voltage to be evaluated against these levels. The improved algorithm is accomplished as follows:

- *add half* the reference level if the voltage is below the negative reference voltage,
- *add zero* if the input to the stage is above the negative reference and below the positive reference or
- *subtract half* the reference if the input to the stage is above the positive reference

Using this method, results in a total of three decision levels for each stage instead of the two levels used in the SAR. The increased circuit complexity allows for the addition of a redundant decision level and provides for half-bit of error correction in each stage. This technique was first described by S. Lewis and P. Gray in reference 57 and subsequently expended by Lewis et al (ref. 67 and others - 56,58,61). The concept is illustrated in figure 8-2.

As shown in figure 8-2, the input signal is first acquired by the SHA. During the next phase, the held signal is compared with the help of a window comparator to see whether it resides between the two reference levels. The result of the comparison sets the switch position such that next stage's input is equal to twice the output of the previous stage shifted by a positive or negative offset as explained above.
Figure 8-2 illustrates also the flow chart for the algorithm used in the implementation of the 1.5 bit/stage converter as outlined in reference 58.
The converter in figure 8 – 2 can be realized using single-ended or differential circuit implementations.

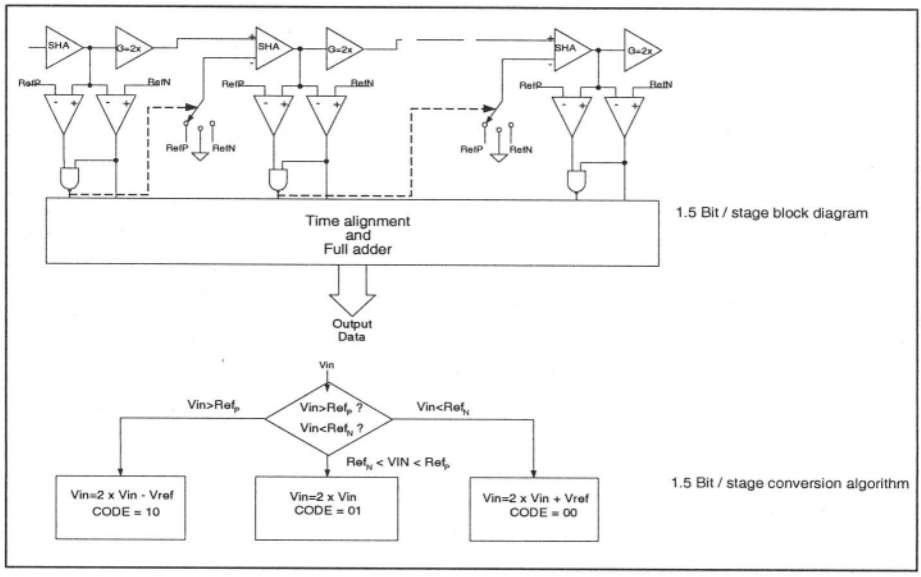


Figure 8-2. A pipeline converter with 1.5-bit / stage

An example of a possible implementation of this converter is shown in figure 8 - 3.

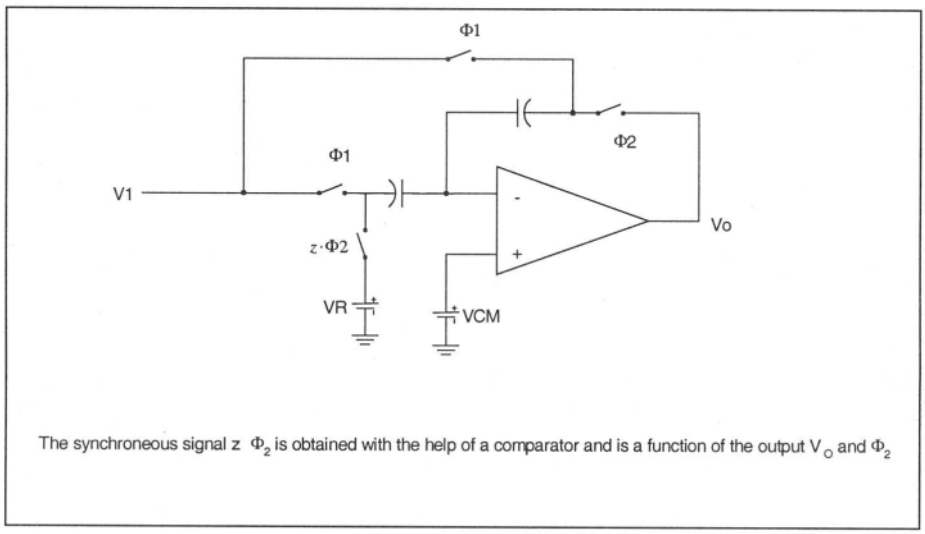


Figure 8-3. Single-ended implementation of the MDAC

Our example portrays a single-ended realization of the circuit using an op-amp, four switches and two capacitors. During the first phase (**Φ1**), the input voltage V1 is sampled on the bottom plate of capacitors Cs and Cf. During the second phase (**Φ2**), the charges stored on the two capacitors get redistributed and the output voltage gets offset with the help of the reference voltage V_R. The switch above V_R is controlled by a signal called **z·Φ2**.This control signal is the result of the comparison of the output voltage V_O synchronized with **Φ2**.

To understand the technique let us write the charge equations for the circuit. We simplify the task by assuming that the operational amplifier has an open-loop gain A. We also assume in the analysis that the switches do not contribute charge-injection errors to the circuit. The charge equations for each phase (**Φ1** and **Φ2**) are:
during phase **Φ1**:

$$Q(\Phi 1) = V_{CM} \cdot (C_S + C_F) - V_1 \cdot (C_S + C_F) \qquad 8.1$$

during phase **Φ2**:

$$Q(\Phi 2) = \left(V_{CM} - \frac{V_O}{A} \right) \cdot (C_S + C_F) - V_R \cdot C_S - V_O \cdot C_F \qquad 8.2$$

Since the charge on the capacitor is preserved **Q(Φ1) = Q(Φ2)**.
Rewriting the equations and extracting **Vo** we obtain the following transfer function for the circuit:

$$V_O = V_1 \cdot \frac{A \cdot (C_S + C_F)}{C_S + C_F \cdot (1 + A)} - V_R \cdot \frac{A \cdot C_S}{C_S + C_F \cdot (1 + A)} \qquad 8.3$$

For a better understanding of the transfer function we assume for a moment that the open loop gain of the amplifier A is infinite. Taking the limit of the equation above with $A \rightarrow \infty$ we obtain:

$$V_O = V_1 \cdot \frac{C_S + C_F}{C_F} - V_R \cdot \frac{C_S}{C_F} \qquad 8.4$$

If the reference voltage V_R is dependent on the output voltage V_O such that it can be positive, zero or negative then the equation represents the implementation of the algorithm in figure 8-2.

If the capacitors C_s and C_f are equal in value then the circuit provides the needed gain of two with the associated input dependent reference offset. The version of the circuit described is frequently implemented in a differential manner. Writing the equation for the differential case is a little more involved, but since the circuit is linear the task can be simplified by the use of superposition method: write the equation for each half of the differential paths by inspection and add the resulting equations.

Some of the benefits achieved by using the differential approach are:

- even harmonics are reduced (when the differential paths are symmetric)
- improved common-mode rejection
- improved power supply rejection
- cancellation of 1/f noise and offset when the differential stage is implemented as a correlated-double sampler (CDS).

Since the differential circuit offers much better performance, we examine next the differential realization for the SHA and gain of two block. The spice model which is an extension of the single ended diagram of figure 8 - 3 is illustrated in figure 8 – 4. As in figure 8 – 3, we use a linear switch and a simplified linear op-amp model. This block is the core of the converter and is usually called the multiplying DAC or MDAC. It implements the following functions:

- SHA
- subtraction and
- D/A.

As in the previous chapters however we utilize a spice macro-model to observe the converter's behavior rather than performing a mathematical analysis. Using SPICE as a tool rather than performing the mathematical analysis has the benefit that the circuit does not have to be linear. In addition the gain reduction with increased input frequency can be modeled in the equations above where **A=A(f)**.

In the implementation illustrated in figure 8 – 4 the three functions, share a capacitor array of equal values.

In practice the MDAC is implemented with the use of transfer gates (an NMOS in parallel with a PMOS device) rather than the analog switches of our model. In the SPICE implementation we use linear switches with "ON" resistance of **100 Ω** and threshold of 2 V. The amplifier is modeled with the help of a voltage controlled voltage source E6; E1 and E2 represent an open

loop gain of 60 dB **(x=2,000 x 0.5).** The bandwidth is limited with the help of R_{BW} and C_{BW} resulting in a –3dB corner frequency of approximately 16 MHz. The differential output is simulated with the help of E1 and E2 with an attenuation of 0.5. The result is therefore a gain-bandwidth product of 16 GHz **(=1000·16 MHz).** While this is a large gain-bandwidth for a CMOS device, it can be achieved using a **0.25μ** technology at the gain of 60 dB.

Ro and Rob **(=500 Ω)** represent the open loop output resistance of the amplifier.

The comparators are implemented as voltage controlled-voltage sources with a constant gain of 50x (E8, E9, E11 and E12). No effort is made to include bandwidth limitation in the comparator behavior.

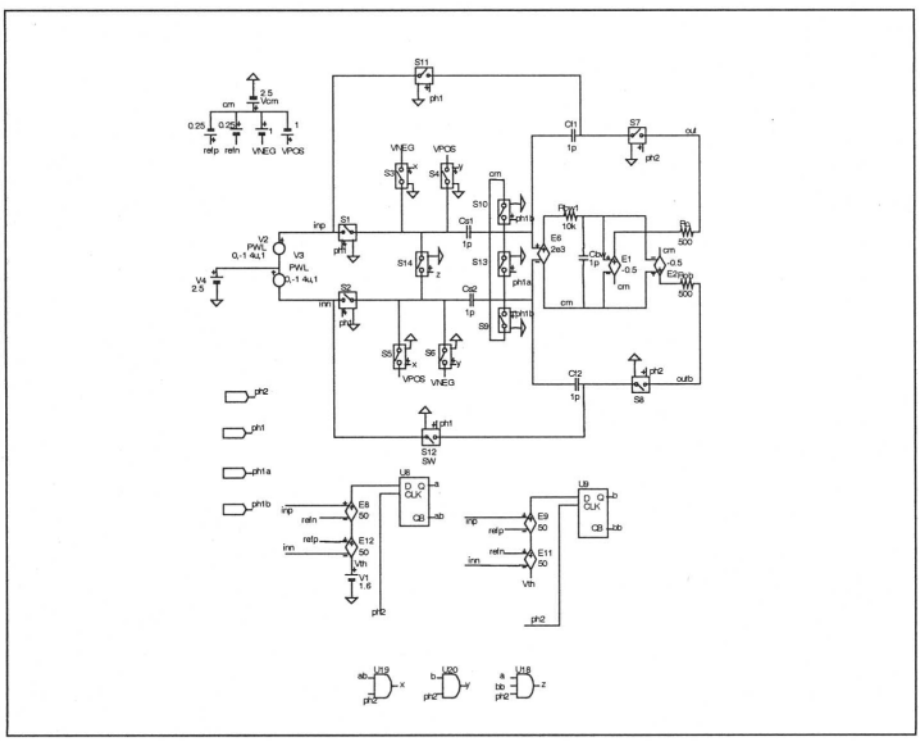


Figure 8-4. MDAC Spice schematic

Although the comparator model includes four inputs, the practical implementation can be fairly straightforward (it can be designed with the use of four transistors arranged as two differential pairs as implemented in reference 56).

Finally, the comparators outputs use the edge triggered DLATCH U8 and U9. In a physical implementation the comparators use regenerative positive feedback, thus allowing for a relatively low DC gain.

The comparator can essentially be single ended and use only one set of reference voltages and one phase of the input signal (the positive reference **refp** and the input signal **inp**). Such an implementation however would lack the ability to reject the common-mode transients of the negative input **inn** and negative reference **refn**.
The entire MDAC model, while fairly simplistic is a relatively good approximation of a single-pole roll-off implementation. While some of the practical limitations of the amplifier and comparators such as slew-rate limitation, common mode rejection and power supply rejections are idealized this model can give us a good understanding of the MDAC behavior.

As figure 8 – 4 shows the device operates in two main phases similarly to the single ended version described in figure 8 – 3. The signals used in the model are described briefly (given the new features added in the differential model).

The analog input voltage is sampled during the first phase. During the second phase the sampling capacitor Cs gets connected to **+Vref,** GNDor– Vref depending on the result of the comparator decision. This determines the digital output of the MDAC and whether the reference voltages are added or subtracted from the residue. The detection of the range is performed by the AND gates U18, U19 and U20 and their outputs x, y and z control the position of switches S3, S4, S5 and S6 (see figure 8-4 again).
The transfer function of the MDAC is illustrated in figure 8-5.

Figure 8 – 5 shows the three possible output states for the MDAC: 00, 01 and 10 and the reference levels at which the transitions occur when the reference voltages are ± 1V and the full-scale range of the device is ± 2V. The state 11 is reserved for the correction range as will be shown shortly. The associated timing diagram for the MDAC is illustrated in figure 8 – 6. The two clock phases **Φ1** and **Φ2** have the purpose of establishing the sampling and holding instances. They are essentially two nonoverlapping signals (for a simple realization scheme see reference 68). The control signals **Φ1A** and **Φ1B** are intended to reduce the sample to hold transition. While **Φ1, Φ1A** and **Φ1B** are high, the op-amp inputs are connected to each other and to the common-mode voltage (CM) and the SHA inputs are connected to the sampling caps C_{s1} and C_{s2} and to the feedback capacitors C_{f1} and C_{f2}.

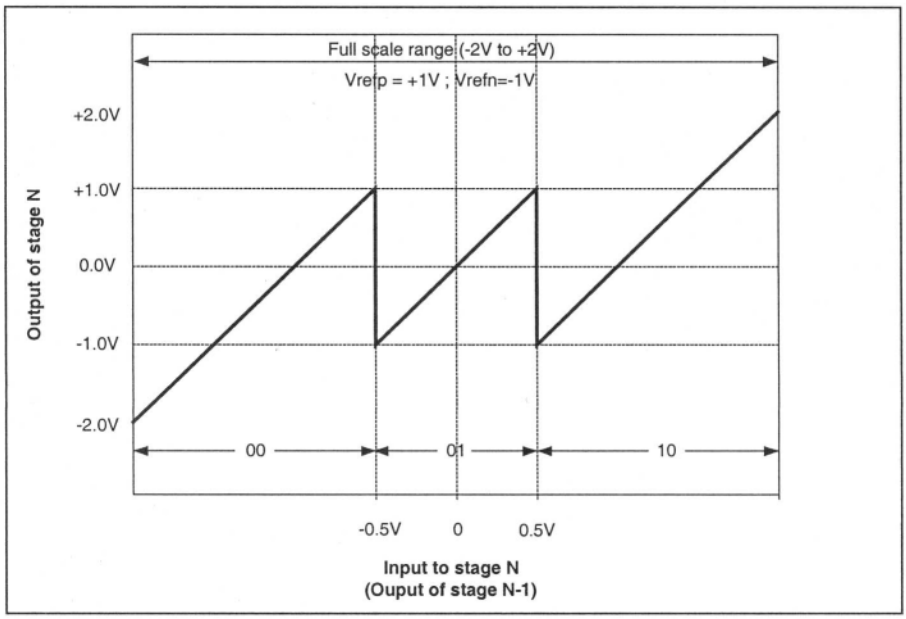


Figure 8-5. Transfer curve of the MDAC

When **Φ1B** goes low, the amplifier's inputs get disconnected from the common-mode voltage CM but remain connected to each other until **Φ1A** goes low which cause the inputs to be sampled on the capacitors.

During the phase **Φ2,** the capacitors C_{f1} and C_{f2} get connected to the amplifier's outputs and the sampling capacitors C_{s1} and C_{s2} get connected to each other and the references VPOS or VNEG depending on the digital signals X, Y or Z. Assuming perfect matching of C_{s1}, C_{s2}, C_{f1} and C_{f2}, at the end of the second phase the output of the MDAC is equal to twice the input voltage plus or minus the reference voltage as shown in figure 8–5.

Using the MDAC in figure 8–4 and the associated timing diagram we obtain three output codes for the digital outputs y and z. Assuming that y is the MSB and z is the LSB of the pair we obtain the codes 00, 01, and 10. The code 11 does not occur allowing for a relatively large correction range. Only two comparators are used to obtain the two thresholds with their three associated decision levels. Each comparator is offset by **+/- 0.25·Vref** thus making the block able to tolerate offset errors of 0.25.Vref without performance degradation as it will be shown by an example.

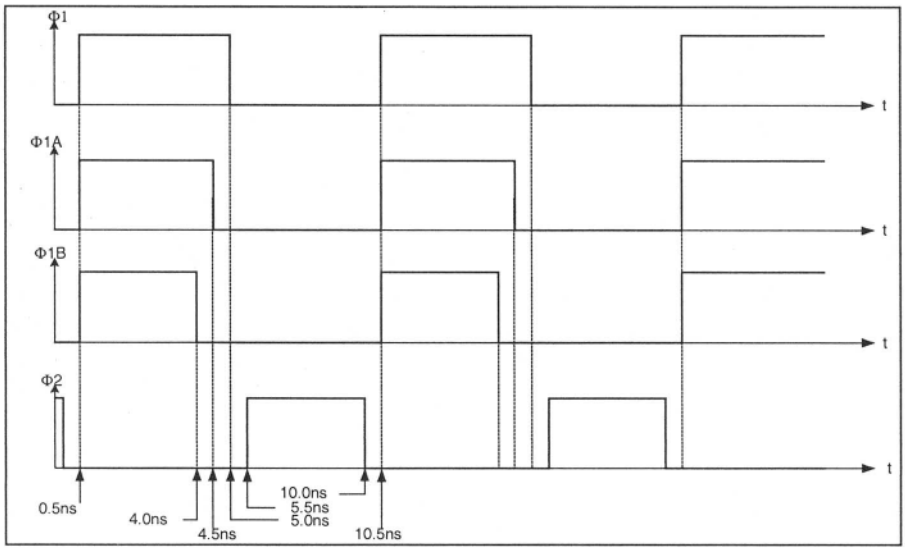


Figure 8-6. Timing diagram for the MDAC (non-overlapping phases)

1.1 An 8-bit serial pipeline A/D example

Using the MDAC model of figure 8-4 we describe next an implementation of a serial pipeline A/D converter with 1.5 bit/stage. As in previous chapters we consider an 8-bit converter only in order to keep the simulation times reasonably short. Figure 8-7 illustrates our 8-bit A/D. In figure 8-7, devices X1 – X7 represent the MDACs. X1 detects the most significant bit while X7 detects the less significant bit. The LSB of the converter is detected with a single comparator (modeled with the help of E18, E19 and latches U95 and U55). Since the LSB does not require error correction, this is the way in which most practical converters are implemented.

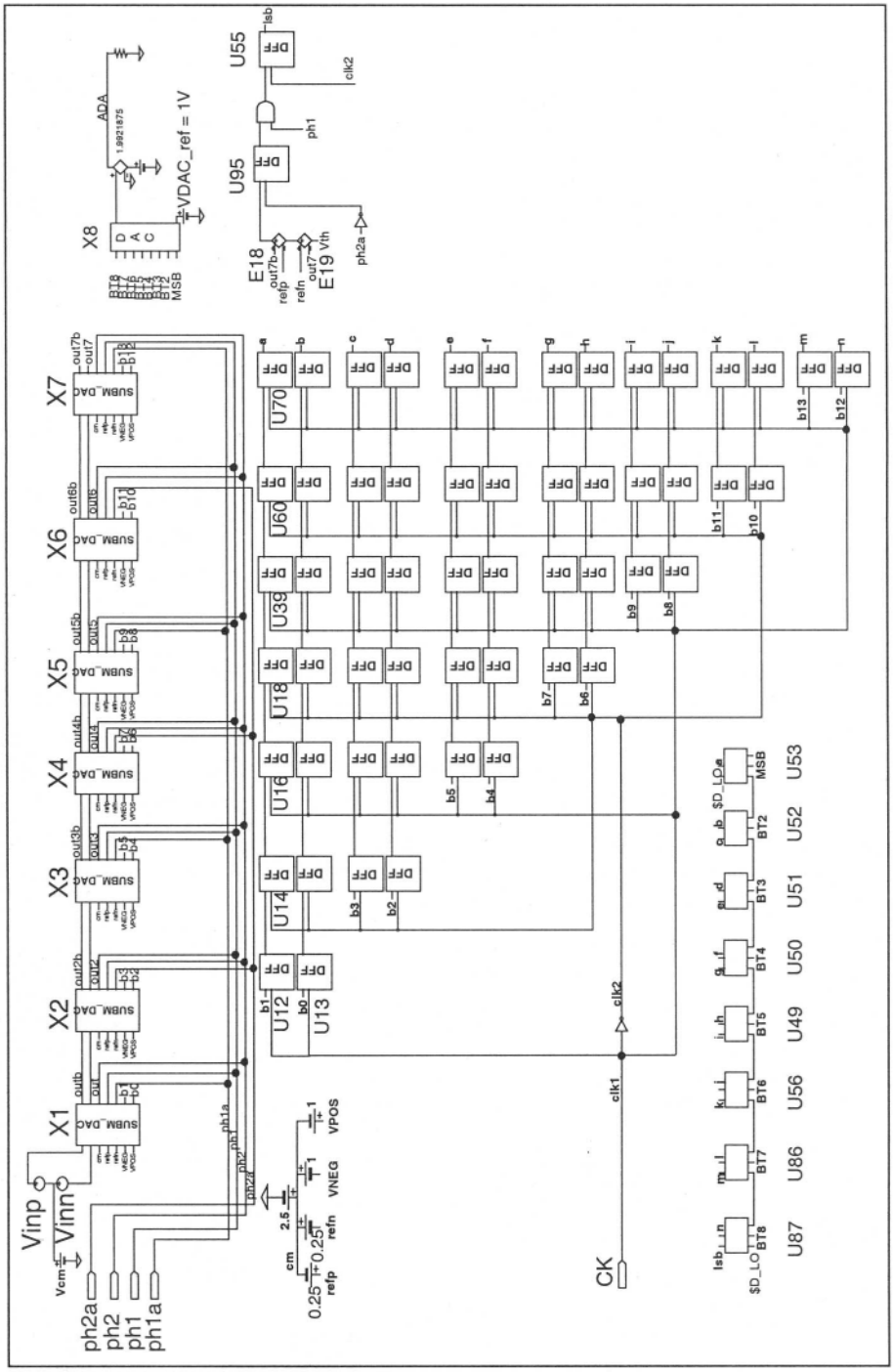


Figure 8-7. Serial pipeline A/D with 1.5-bit / stage - Spice schematic

Alternately, at the cost of power increase we can obtain higher resolution by making the last block in the chain a 2-bit flash A/D (3 comparators and associated logic). As seen in figure 8 – 7 each stage samples the previous stage' output out of phase with its immediate neighbors. The digital outputs of the MSB stage ("b0" and "b1") are latched with the help of U12 and U13. These output levels are delayed in time with the help of U14, U16, U18, U39, U60 and U70 such that the error correction can take place only after each stage propagated its result to the end of the chain.
Subsequent bits are detected in a similar fashion to the MSB. Since each bit occurs one clock cycle later than the previous bit, it requires one less delay and therefore one less latch. Hence the diagonal expansion in the diagram.

The error correction is implemented with the help of U53, U52, U51, U50, U49, U56, U86 and U87, which are full-adder cells (two digital inputs, a carry-in, sum and carry-out). The error correction forces the output code to be corrected from the LSB and adding one more bit per stage until reaching the MSB (as shown in the previous chapter).
In our example we connect the digital output code of the A/D to the input of a perfectly linear 8-bit D/A converter (X8) for reconstruction purposes.
The input signal to the converter is fully differential and is offset by a common mode voltage of 2.5V. Although our implementation uses the same common mode voltages for both the input and the output of the MDAC (as usually done in practical converters) this is not an absolute necessity.
The reference voltages V_{NEG} and V_{POS} have magnitudes of 1V and the associated comparator offsets V_n and V_p have magnitudes of 0.25V $(=V_R/4)$. This, results is a full-scale range of 4V for the converter (+/- 2V). The lack of orderly sequence in the component identifiers in figure 8-7 stems from us trying to maintain the same device identities assigned by Topspice™ in the files provided to the reader in appendix A.

As in previous chapters, we examine first the converter's dynamic performance without any errors in any of its subcircuits. The SINAD of our 8-bit converter without errors is illustrated in figure 8-8 as a function of input signal frequency. The dynamic performance is tested using a sampling frequency of 100 MSPS and 256 time points resulting in an input beat frequency of 390.625kHz.

Two parameters in our model (figure 8–4) affect the settling speed with which the capacitors acquire the charge:

- limited bandwidth of the amplifier – modeled by the E6 and R_{BW} and C_{BW} in the MDAC and
- output impedance of the amplifier – modeled by El, E2, R_O and R_{OB}.

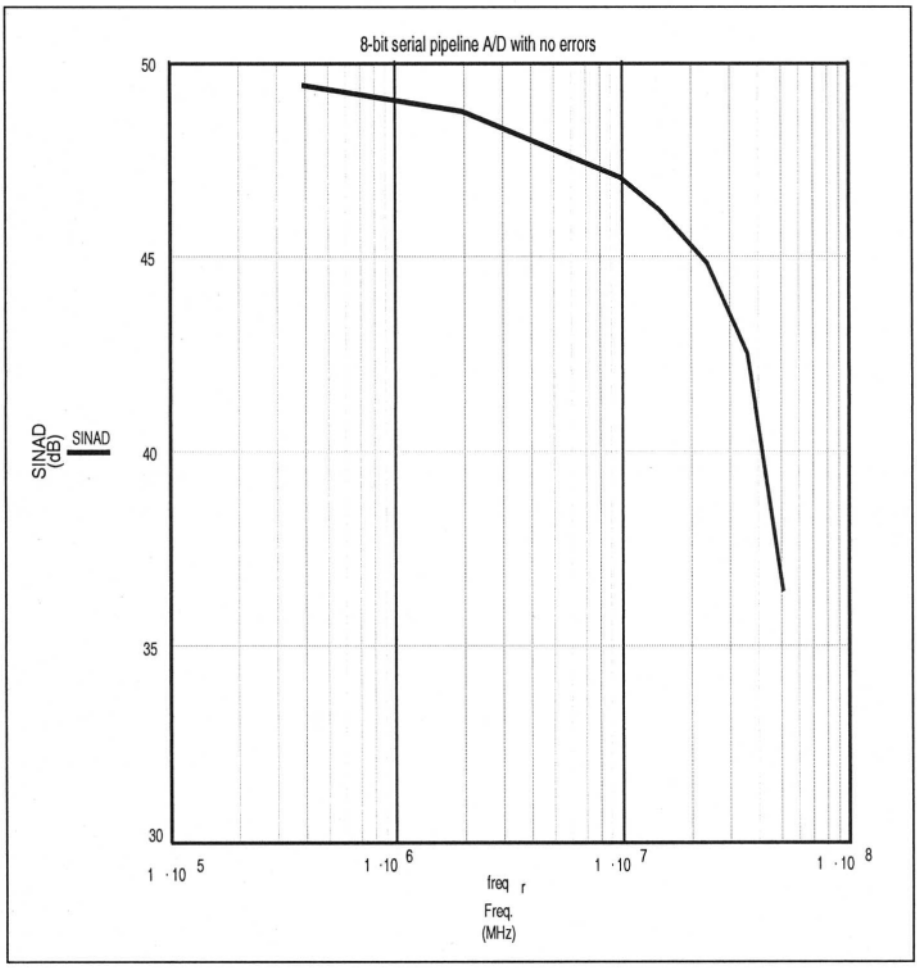


Figure 8-8. Dynamic performance of 8-bit serial pipeline A/D with no error

These parameters restricts the acquisition of high frequency input signals in the amplifier and its drive ability as follows. Assuming that the amplifier has an open-loop resistance R, we calculate its output impedance as:

$$Z(\omega) = \frac{R}{A(\omega)} \qquad 8.5$$

Since the amplifier gain $A(\omega)$ gets reduced with a slope of –20 dB/decade beginning at the pole frequency ω_P:

$$A(\omega) = \frac{A_0}{1 + j \cdot \frac{\omega}{\omega_p}} \qquad 8.6$$

it results that the output impedance of the amplifier increases from a value of R/A_0 at low frequency and it behaves therefore as an inductor:

$$Z(\omega) = \frac{R}{A_0} \cdot \left(1 + j \cdot \frac{\omega}{\omega_P}\right) \qquad 8.7$$

This behavior affects the amplifier speed because it exhibits the settling of a voltage across a capacitor driven from voltage source in series with an "inductor" $(=Z(\omega))$.

There are other things influencing the converter's accuracy in actual implementation:

- accuracy of the capacitor ratios
- the voltage coefficient of the capacitors (related to the capacitors ratios)
- accuracy of V_{POS} and V_{NEG}
- transfer gates (switches) linearity

With very careful layout techniques, a high degree of symmetry can be achieved in the MDAC. This symmetry causes a reduction in even harmonics resulting therefore in a differential output with only odd harmonic content.

1.1.1 Capacitance mismatch in the MDAC

In practice the circuit errors are not always symmetric. Our first example will examine the effects of these capacitors mismatch. Figure 8–9 illustrates the case in which the MDAC capacitors are mismatched so that only one of the four in the array deviates from the nominal value.

Figure 8–9 reflects the behavior of the 8-bit converter when the MSB MDAC when the sampling cap C_{s1} has a 25% error compared to the other three capacitors (top). The remaining MDACs are perfectly matched. While a 25% seems a large error between adjacent capacitors, this is just to illustrate the importance capacitor matching in the MDAC and is not a reflection of a serious issue in the I.C. layout.

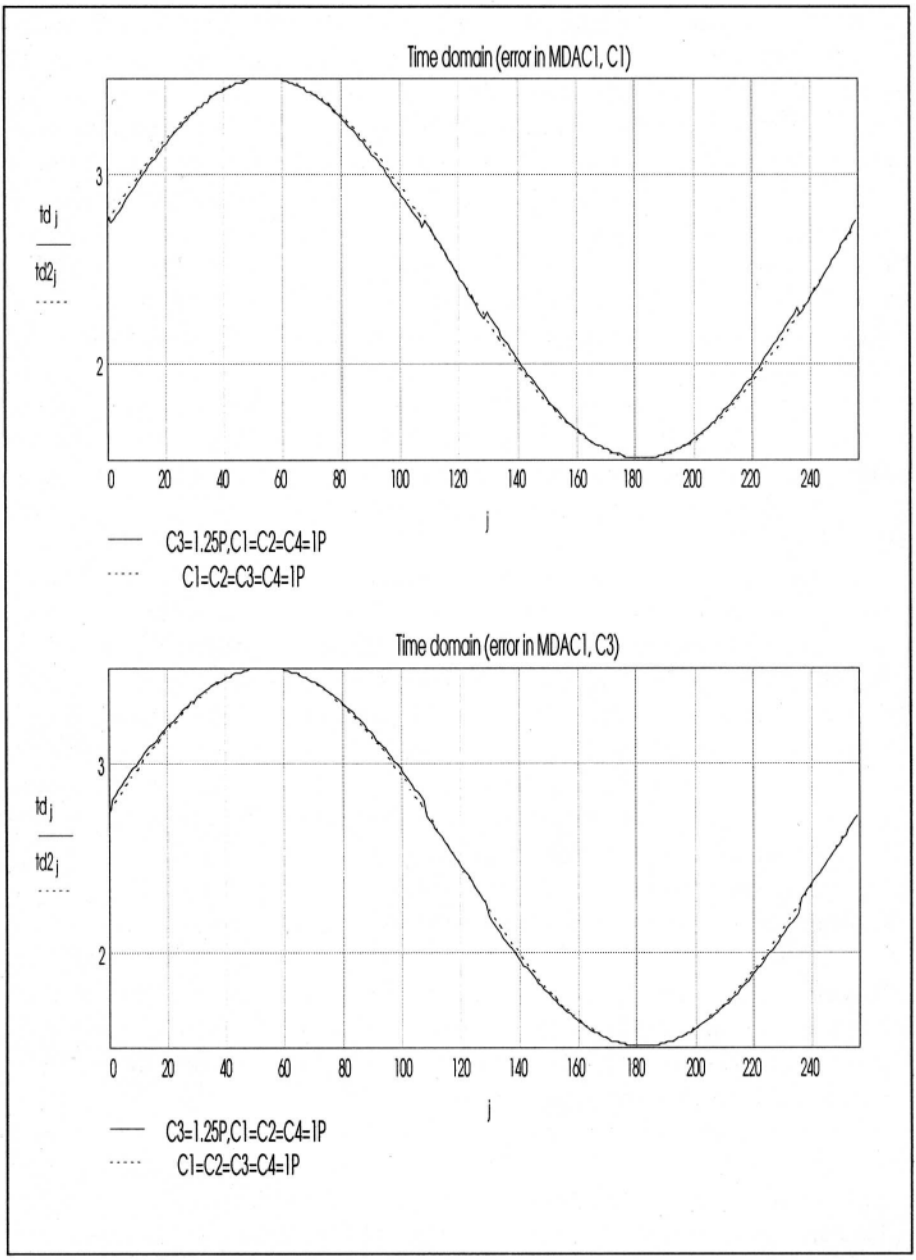


Figure 8-9. Time domain reconstruction of 8-bit serial pipeline A/D with 1.5 bit / stage - (a) Capacitance mismatch in MDAC#1 - C1=1.25pF, C2=C3=C4=1pF; (b) Capacitance mismatch in MDAC#1 - C3=1.25pF, C1=C2=C4=1pF

The case is presented here so the reader can observe the signature caused by capacitance mismatch error in this converter. The associated graph on the bottom of the figure represents a 25% error of the feedback capacitor C_{f1}. The time domain error shown in figure 8 – 9 represents an asymmetric error in each case resulting in significant degradation in the dynamic performance. The error is significant enough as to make the time domain reconstruction noticeably distorted. The SINAD for each case of figure 8 – 9 are: SINAD(with C1 error)= 32.684 and SINAD(with C3 error)=33.474 respectively. Since the errors just discussed are fairly large it stands to reason that the converter's performance is pretty poor (approximately 5.2 effective bits). This asymmetrical error in capacitance not only creates a gain error but in a practical converter causes also the input common-mode voltage to be different than the output common-mode in the MDAC. Our model has infinite common-mode rejection and therefore the settling of the common-mode transient is not observed in the A/D performance.

A more practical and interesting case to discuss then the blatant mismatch above is when the capacitance mismatch in the MDAC is more subtle. For example let us simulate the dynamic performance of the 8-bit A/D when C1=1 pF versus the case when C1=1.008 pF. This is a minute difference, but it represents an error of one part in 125 or the equivalent of 2 LSBs (1LSB represents an error of 1part in 256 for an 8-bit converter). If the MDAC in question is the first in the chain – X1 we notice mostly a gain error for the entire converter and the SINAD= 49.065 dB compared with the fully matched capacitor case of 49.138 dB.

If however, we put the same 2 LSB error in second MDAC (X2) then we should expect a more noticeable error in the dynamic performance of the converter (due to the DAC error of the MDAC as explained in the previous chapter). A 2 LSB error in X2 is obtained with a capacitor error that is twice the error in X1, and the capacitor value required for this case is C1=1.016pF. The reason is that the first MDAC has a gain of 2 and therefore the X2 error referred to the input of the ADC is divided by 2.

If we simulate this error under similar conditions as above, we obtain a SINAD(C1=1.016pF in X2)=48.34 dB – a reduction of almost 1dB compared to the matched case! This reduction in SINAD reminds us that DAC gain errors of a pipeline A/D can become very significant in the overall converter performance.

This example illustrates also how critical is the layout of such a converter. The reason is as follows: to obtain the high gain bandwidth product our example used a capacitor array with unit capacitors of 1 pF. This guaranteed a fast settling time and relatively low noise (1pF has an integrated RMS noise of 64uV [=k·T/C]). However, any parasitic wiring capacitance can

cause a mismatch in the array as explained above. If the connection of each capacitor unit to the amplifier or the switches is done with unequal conductor lengths the result is an unbalanced array. This mismatch causes reduced dynamic performance as shown by the example.

This is a good illustration of what is commonly referred to as the Pelgrom's rule. The rule states that the amount of mismatch is inversely proportional to the device area. Thus, to obtain a high resolution converter it is imperative to use as large a capacitor area as possible and this of course contradicts with the need for a small capacitance required to achieve the fast acquisition time.

The example just analyzed shows the effects of initial mismatch in the MDAC. Another practical issue arises from the voltage coefficient of the capacitors in "real life". As the accuracy of the converter increases, the voltage coefficient of the capacitors becomes more relevant. This demand becomes even more difficult when the linear and quadratic terms of the coefficient become large. The reason is that in cases like that there are no schemes capable of balancing the differential signal for even and odd distortions. In these instances one should remember that the mismatch could result from the switch capacitance in combination with the routing signal that controls the switch. While the differential signal driving the amplifier is symmetric around the common mode voltage, the control signal driving the switches is not. Therefore, the charge injection seen by the holding capacitors is unbalanced as well. This concept was discussed in chapter 4. All these issues need to be kept in mind in the development of the MDAC.

1.1.2 Errors created by input signal coupling into the references – V_{POS}, V_{NEG}

Our next example examines the effects of input signal coupling into one of the references. This is a practical problem that results from having the reference signal routed across the chip to various MDACs and passing along the way next to the input signal. We examine this case by adding a 10mV voltage source with equal phase and frequency as the input signal in series to the voltage source V_{POS}. Using a frequency of 390.625kHz we measure a SINAD = 47.235 dB or the equivalent of ENOB = 7.55 bits. Even though the culprit is a signal with less than one LSB in magnitude (1LSB = 4V/256 ≅ 16 mV) the reduction in SINAD is fairly drastic. The reason for the extreme behavior is a consequence of the residue signal being amplified as it goes through the signal path. This leads to large errors as the residue reaches the last stage in the pipe after a total gain of 128 in the previous stages. Not only is the SINAD reduced, but also the harmonic content is vastly different than previous examples. In the ideal case no even harmonics were present; now

the even harmonics become quite high due to the asymmetry of the signals. Second harmonic for this case is -54.7 dB with third harmonic of -59.9 dB and fourth of -61.9 dB.

1.1.3 Errors created by input signal coupling into refp, refn

Another case worth mentioning is the case when the input signal is coupled into the references of the window comparator **(refp** or **refn).** This is simulated with the same frequency signal and the same amplitude above (the $\mathbf{V_{POS}}$ case). This time however we connect the parasitic signal in series with the **refp** reference. Now the dynamic performance is SINAD = 48.96 dB. This is almost the same as the ideal case (49.138 dB)! The obvious question is why is one reference **($\mathbf{V_{POS}}$)** more sensitive to noise than the other (refp)?

The reason is that the reference **refp** is not amplified through the signal path. In addition, we know that the error correction can correct comparator offset errors in excess of 0.25· $\mathbf{V_R}$. Looking at the frequency spectrum of the reconstructed signal we notice again that the odd harmonics are very low compared with the $\mathbf{V_{POS}}$ case. This is illustrated in figure 8 – 10. Notice the lack of even harmonics when the coupling is into refp.

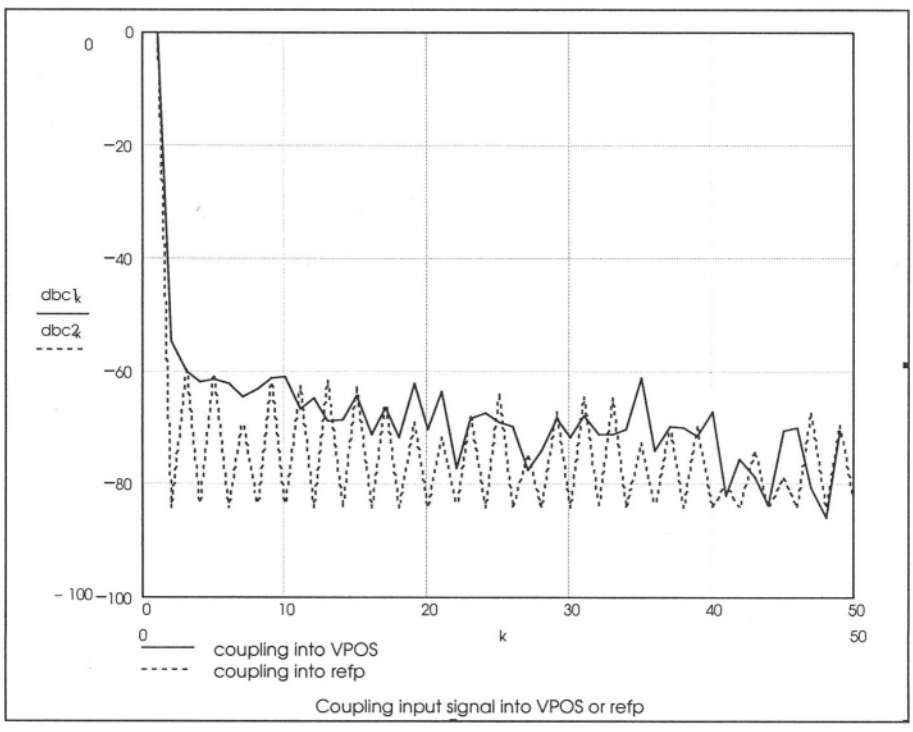


Figure 8-10. Coupling the input signal into Vpos (solid) and REFp (dotted)

SUMMARY

- Utilizing the SHA, DAC and error amplifier in one building block as the MDAC is one of the most compact and beneficial circuit implementations for a 1.5 bit/stage converter.

- Using the MDAC in a 1.5 bit/stage converter results in a very efficient digital error correction as shown in figure 8–7

- As any serial pipeline converter, the 1.5 bit/stage requires very accurate interstage gain and DAC function. The comparators used for the window comparator can be fairly inaccurate given the redundant state used for error correction.

- Using the MDAC as the main element in a 1.5 bit/stage A/D allows for the implementation of a serial pipeline converter with good dynamic performance at high sampling rates. This is accomplished with relatively low power consumption and at the expense of having pipeline delays.

- The MDAC outlined in the chapter can be modified to implement a CDS function, which can help reduce the 1/f noise contribution.

- It is imperative to match the capacitor units used in the MDAC array not only in capacitance area itself but also total conductor routing in order to reduce the mismatch thus maintaining good dynamic performance.

- Relatively large errors in comparator offsets can be tolerated when using the 1.5 bit/stage with digital error correction methods outlined. The digital error correction outlined in figure 8–7 constitutes a very robust correction algorithm.

- The most sensitive stages to error are the first stages since farther stages in the pipeline have their error attenuated by previous stage gain.

- Asymmetric behavior in the signal path can be caused by unequal coupling of the input signal to the references V_{POS} and V_{NEG}. This may lead to high content of even harmonics. However relatively large

coupling of the input signal to the reference voltages refp, refn in the window comparator do not affect SINAD.

PROBLEMS

1. Using the spice circuit outlined in the MDAC in figure 8–4, plot the transfer curve caused by altering the values of C1 and C2 to 1.25 pF while maintaining the values of C3=C4=1pF.

2. Using the spice circuit outlined in figure 8–7 and the MDAC outlined in figure 8–4, find the time domain signature caused by altering the values of C1 and C2 to 1.25 pF while maintaining the values of C3=C4=1pF in X1 at an input frequency fin= 390.625kHz.

3. For problem 8.2 find the SINAD at an input frequency fin= 390.625kHz.

4. How would this error affect the dynamic performance of the 8 bit A/D if this MDAC is used in X2 while X1 is the nominal circuit in figure 8–7.

5. Observe the SINAD of the circuit in figure 8-7 when the reference V_{NEG} has a non-zero impedance connection to X2 only and a capacitance C_1 of 5pF as shown in figure P8.5

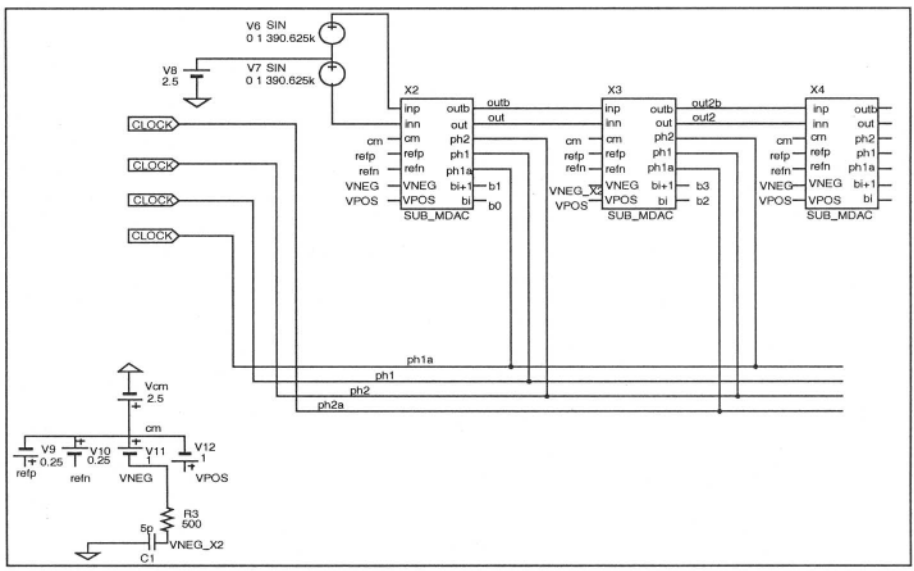


Figure 8-11. P 8.5

6. Repeat problem 4 when the input frequency is changed from 390.625 kHz to 35.54 MHz and plot the SINAD as a function of input frequency.

7. Repeat problem 1, using the nonlinear capacitor model developed in chapter 4 (use equation 4.6 with parameters C_0= 1 pF, C_1=10 PPM/V and C_2=10 PPM/V^2) and verify the transfer curve linearity.

8. For the case in problem 7 find the SINAD of the 8-bit A/D of figure 8 – 7 at an input frequency fin= 390.625kHz

Bibliography

1. Rudy van de Plassche, "Integrated Analog to Digital and Digital to Analog Converters", Kluwer Academic Publishers, 1994
2. J.E. Solomon, "The Monolithic Op Amp: A Tutorial Study", IEEE Journal of Solid State Circuits, vol.SC9, pp.314-332, December 1974
3. Harry J.M. Veendrick , "The behavior of flip-flops used as synchronizers and prediction of their failure rate", IEEE Journal of Solid State Circuits, vol.SC-15, pp. 169-176, April 1980
4. P.R.Gray, R.G. Mayer , "Analysis and design of analog integrated circuits", 2nd edition – John Wiley and Sons
5. I.Getreu, A. Hadiwidjaja, J.Brinch, "An integrated circuit comparator macromodel", IEEE Journal of Solid State Circuits, vol.SC-11, pp. 826-833, December 1976
6. A. Barna, "Propagation delay in current mode switching circuits", IEEE Journal of Solid State Circuits, pp. 123-124, April 1976
7. A. Moscovici, "Understanding comparator dynamic response", Electronic Design, pp. 77-84, August 7, 1995
8. T.O. Anderson, "Code conversion for parallel A/D converters", Computer Design, July 1973
9. Analog Devices, "High speed design seminar", 1989/1990
10. R.C. Hicks, "A survey of A/D converters for RADAR applications", Technical Report TR-10, May 1991
11. J. Doernberg, H-S. Lee and D.A. Hodges, "Full speed testing of A/D converters", U.C. Berkeley, May 1984
12. M. LaBouff and S. Sockolov, "Test flash A/D converters to unearth hidden specs", Electronic Design pp. 119-124, June 25, 1987

13. W. Kester, "Flash ADCs provide the basis for high speed conversion", EDN, pp. 101-110, January 4, 1990
14. W. Kester, "DSP test techniques keep flash ADCs in check", EDN, pp. 133-142, January 18, 1990
15. W. Kester, "Measure flash ADC performance for trouble-free operation", EDN, pp.103-114, February 1,1990
16. R.K. Ushani, "Subranging ADCs operate at high speed with high resolution", EDN, pp. 155-162, April 11, 1991
17. R.K. Ushani, "Dynamic specifications describe performance of subranging ADCs", Datel application note AN-5, Reprinted from EDN 1991
18. M.J. Demler, "Time domain techniques enhance testing of high speed ADCs", EDN, pp. 115-120, March 30, 1992
19. IEEE Standard for performance measurements of A/D and D/A converters for PCM Television Video Circuits – IEEE October 15, 1984
20. M. Koen, "Comparing ADC architectures is a designer's best bet", Electronic Design, pp.119-122, January 22,1987
21. A.G.F. Dingwall, V. Zazzu, "An 8 MHz CMOS subranging 8 bit A/D converter", IEEE Journal of Solid State Circuits, vol. SC-20, pp. 1138-1143, December 1985
22. R.K. Ushani, "Subranging ADCs operate at high speed with high resolution", Datel Application note AN-5, Reprinted from EDN 1991
23. F. Goodenough, "Multistep, monolithic 10 and 12 bit ADCs digitize 5 to 20 MHz sine-waves at Nyquist. 12 bit IC ADCs sample at up to 20 MHz", Electronic Design, pp. 47-59, October 24,1991
24. C.S.G. Conroy, D.W.Cline and P.R.Gray , "An 8 bit 85 MS/s parallel pipeline A/D converter in 1µm CMOS", IEEE Journal of Solid State Circuits. Special Issue on Solid State Circuits, pp. 447–454, April 1993
25. W.C. Black and D. A. Hodges, "Time interleaved converter arrays", IEEE Journal of Solid State Circuits, vol.SC-15, pp. 1022-1029, December 1980
26. K. Poulton, J. Corcoran and T. Hornak, "A 1 GHz 6 bit ADC system", IEEE Journal of Solid State Circuits, vol.SC-22, pp.962-970, December 1987
27. P.Louzon, "Decipher high sample rate ADC specs", Electronic Design, pp. 91-100, March 20,1995
28. R.E. Leonard, "Data converters: Getting to know dynamic specs", Datel Application note, AN-3
29. R.E. Leonard, "Understanding data converters' frequency domain specifications", Datel Application note, AN-4
30. J.D. Giacomini, "High performance ADCs require dynamic testing", Electronic Design, pp. 55- 65, August 6, 1992

31. Dynamic testing of A/D converters – H.P. Product note 5180A-2
32. B.E. Peetz, A.S. Muto, J.M. Nell, "Measuring waveform recorder performance", pp21-29, H.P. Journal, November 1982
33. W.T. Colleran, "A 12 bit, 100 MS/s A/D converter using interpolation, and Analog encoding", PhD thesis, UCLA, December 1993
34. M. Inoue, H. Sadamatsu, A. Matsuzawa, A. Kanda, T.Takemoto, "A monolithic 8 bit A/D converter with 120 MHz conversion rate", International Solid State Circuits Conference, pp.296-297, IEEE, February 1984
35. T. Takemoto, M. Inoue, H. Hadamatsu, A. Matsuzawa and K. Tsuji, "A Fully parallel 10 bit A/D converter with video speed", IEEE Journal of Solid State Circuits, vol. SC-17, pp. 1133-1138, December 1982
36. A. Matsuzawa, M. Kagawa, M. Kanoh, K. Tatehara, T. Yamaoka, and K. Shimizu, "A 10 bit 30 MHz two step parallel BiCMOS ADC with internal S/H", ISSCC Digest of Technical Papers, pp. 162-163, February 1990
37. D.R.Martin and D.J.Secor, "High Speed analog to digital converters in communication systems: Terminology, architecture, theory and performance", TRW Electronic Systems Group, 1981
38. D. Dooley, "Data conversion integrated circuits",IEEE Press, 1980
39. B. Porat, "A course in digital signal processing", John Wiley & Sons, 1997
40. R.E. Leonard, "Picking the right Sample and Hold amp for various data acquisition needs", Datel Application note, AN-2
41. A.Matsuzawa, H.Yamada, "Parallel type A/D converter", US Patent 4,963,874
42. A.Papoulis, "Signal analysis", McGraw Hill, 1977
43. R.W.Ramirez, "FFT fundamentals and concepts", Prentice Hall, 1997
44. T.L.Tewksbury and H.S.Lee, "Characterization, modeling and minimization of transient threshold voltage shifts in MOSFETS", 1993 IEEE Custom Integrated Circuits Conference, pp. 14.4.1-14.4.3
45. A. Arbel and R. Kurz, "Fast ADC", IEEE Transactions on Nuclear Science, vol. NS-22, pp. 446-451, February 1975
46. R.E.J. van de Grift and R. van de Plassche, "A monolithic 8-bit video AD Converter", IEEE Journal of Solid State Circuits, vol. SC-19, pp. 374-378, – June 1984
47. R.E.J. van de Grift, I.Rutten and M. van de Veen, "An 8-bit video ADC incorporating folding and interpolation techniques", IEEE Journal of Solid State Circuits, vol. SC-22, pp. 944-953, December 1987
48. R. van de Plassche and P. Baltus, "An 8-bit 100 MHz full Nyquist A/D Converter", IEEE Journal of Solid State Circuits, vol. SC-23, pp. 1334-1344, December 1988

49. J. Van Valburg and R. van de Plassche, "An 8-bit 650 MHz folding ADC", IEEE Journal of Solid State Circuits, vol. SC-27, pp. 1662-1666, December 1992
50. A. Venes, B. Nauta and R. van de Plassche, "Low power folding A/D converters", Proceedings of the Workshop Advances in Analog Circuit Design, Villach, Austria 1995.
51. G.S. Ostrem, "High speed A/D converters for mixed signal ASIC applications", Norwegian University of Science and Technology, PhD thesis, December 1996
52. O. Moldsvor and G.S. Ostrem, "An 8-bit, 200 MSPS folding and interpolating ADC", Analog Integrated Circuits and Signal Processing, 15, pp. 37-47, 1998
53. Hui Pan, "A 3.3V 12 bit, 50 MS/s CMOS A/D converter in 0.6 µm CMOS with 80 dB SFDR", UCLA, PhD thesis, December 1998
54. Phillip Alien and Douglas R. Holdberg, "CMOS Analog Circuit Design", Saunders College Publishers, 1987
55. P.W. Tuinenga, "SPICE – A guide to circuit simulation & analysis using Pspice", Prentice Hall 1988
56. David Cline, "Noise, speed, and power tradeoffs in pipelined A/D converters", PhD Thesis, University of California, Berkeley, November, 1995.
57. Stephen H. Lewis and Paul R. Gray, "A pipeline 5 MSPS 9-bit Analog to Digital Converter", ISSCC, pp. 210-211, IEEE, February 1987
58. Bernard Ginetti, Paul G.A.Jespers and Andre Vandemeulebroecke, "A CMOS 13-b Cyclic RSD A/D converter", IEEE Journal of Solid State Circuits, vol. SC-27, pp. 957-965, July 1992
59. Michael K. Mayes and Sing W. Chin, "A multistep A/D converter family with efficient architecture", IEEE Journal of Solid State Circuits, December 1989
60. Madhav P.V.Kolluri, "A 12-b 500-ns subranging ADC", IEEE Journal of Solid State Circuits, vol. SC-24, pp. 1498-1506, December 1989
61. T.B. Cho and Paul R. Gray, "A 10 b, 20 MSPS, 35 mW pipeline A/D converter", Proc. IEEE Custom Integrated Circuits Conference, pp. 23.2.1-23.2.4, May 1994
62. ADC Kazuya Sone, Yoshido Nishida and Naotoshi Nakadai, "A 10-b 100 MSPS pipelined subranging BiCMOS", IEEE Journal of Solid State Circuits, December 1993
63. Andrew N. Karanicolas, Hae-Seung Lee and Katilal L. Bacrania, "A 15-b 1 MSPS digitally self-calibrated pipeline ADC", IEEE Journal of Solid State Circuits, December 1993
64. Sehat Sutarja and Paul R. Gray, "A 250 KSPS Pipelined, Analog to Digital converter, ISSCC, pp. 228-229, IEEE, February 1988

65. Bang-Sup Song, Michael F. Tompsett and Kadaba R. Lakshikumar, "A 12-b 1 MSPS capacitor error-averaging pipelined A/D converter" , IEEE Journal of Solid State Circuits, vol. SC-23, pp1324-1333, December 1988
66. Seung-Hoon Lee and Bang-Sup Song, "Digital-domain calibration of multistep A/D converter", IEEE Journal of Solid State Circuits, vol. SC-27, pp. 1679-1688, December 1992
67. Stephen Lewis, H. Scott Fetterman, George F. Gross, R. Ramachandran and T.R. Vishwanathan, "A 10-b 20 MSPS Analog to Digital converter", IEEE Journal of Solid State Circuits, vol. SC-27, pp. 351-358, March 1992
68. R. Jacob Baker, Harry W. Li, David E. Boyce, "CMOS circuit design, layout, and simulation", IEEE Press series on Microelectronic Systems, 1998
69. Ron Kielkowski, "Inside SPICE – overcoming the obstacles of circuit simulations", McGraw-Hill, 1994
70. H. Kimura, A. Matsuzawa, T. Nakamura, S. Sawada, "A 10-bit 300 MHz Interpolated – Parallel A/D Converter", IEEE Journal of Solid State Circuits, vol.SC-28, pp. 438-446, April 1993
71. Peter Jean-Woo Lim, "Performance limits of circuits for A/D conversion", Ph.D. dissertation Stanford University – March 1991

APPENDIX A

Appendix A includes all of the files used for the simulations in the book. These files are provided for the reader that may be using different software than the author. The files are categorized by chapters. Each Spice file contains the appropriate path for the library needed if the circuit is hierarchical.

MATHCAD AND SPICE FILES

CHAPTER 1

TDE (SINAD)

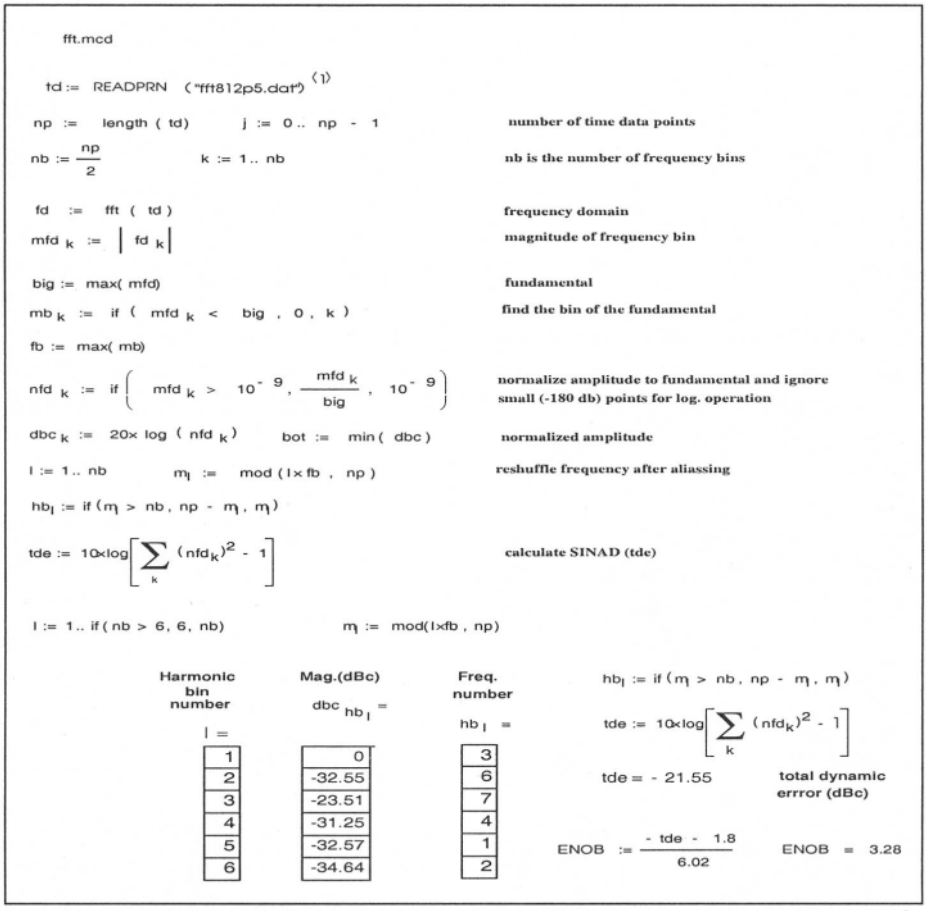


Figure 13. Mathcad - FFT file

ALIASING

ALIAS.mcd

$fb := 437.5$ input frequency $Fs := 1000$

$np := 16$ number of time data points $j := 0..np - 1$

$nb := \frac{np}{2}$ $k := 1..nb$ nb is the number of frequency bins

$l := 1..nb$ reshuffle frequency after aliassing

$f_l := fb \cdot l$ $mm_l := \mathrm{mod}(f_l, Fs)$ $al_l := \mathrm{if}\left(mm_l > \frac{Fs}{2}, Fs - mm_l, mm_l\right)$

$f_l =$	$al_l =$	$l =$
437.5	437.5	1
875	125	2
$1.313 \cdot 10^3$	312.5	3
$1.75 \cdot 10^3$	250	4
$2.188 \cdot 10^3$	187.5	5
$2.625 \cdot 10^3$	375	6
$3.063 \cdot 10^3$	62.5	7
$3.5 \cdot 10^3$	500	8

In this example read the results as follows:
fund. @ 437.5
H2 @ 875 aliases @125
H3 @ 1312.5 aliases @ 312.5
H4 @ 1750 aliases @ 250 etc...

Figure 14. Mathcad - aliasing calculation

CHAPTER 2

COMP5

```
R11 inp inn 100k
R19 0 inp 100MEG
R20 inn 0 100MEG
Vinp inp 0 PWL
+ 0,-0.5
+ 5E-10,-0.5
+ 1E-09,0.5
Vinn inn 0 0.48v
G1 0 int value {5*(10^-3)*TANH((V(in))/0.06)}
```

```
E2 1 0 int 0 10
Rp 1 out 0.75k
Cp out 0 1p
CSR int 0 0.5p
D8 2 int DV
D9 int 3 DV
V2 3 0 0
V3 0 2 0
Rg int 0 2.4k
E3 in 0 inp inn 1
R24 in 0 1meg
.END
```

COMP7_S

```
.SUBCKT comp7_s inp inn out outb
.model DV D
R11 inp inn 100k
R19 0 inp 100MEG
R20 inn 0 100MEG
inp 0 PWL
+ 0,-0.5
+ 2E-09,-0.5
+ 3E-09,0.5
inn 0 0.495v
G2 0 int value {5*(10^-3)*TANH((V(in))/0.06)}
E2 1 0 int 0 1
Rp 1 pole 0.75k
Cp pole 0 1p
CSR int 0 0.4p
D8 2 int DV
D9 int 3 DV
V2 3 0 0
V3 0 2 0
Rg int 0 24k
E3 in 0 inp inn 1
R24 in 0 1meg
D10 Vth out D
D11 Vth outb D
Gout out outb pole 0 1m
Routb outb Vss 10k
Rout Vss out 10k
* Power rail voltage sources
Vss Vss 0 -5
Vth Vth 0 2
.ENDS comp7_s
```

COMP8A_S

```
.SUBCKT comp8a_s  inp inn outb out
.model DV D
R11 inp inn 100k
R19 0 inp 100MEG
R20 inn 0 100MEG
G2 0 int value {5*(10^-3)*TANH((V(in))/0.06)}
E2 1 0 int 0 1
Rp 1 pole 0.75k
Cp pole 0 1p
CSR int 0 0.4p
D8 2 int DV
D9 int 3 DV
V2 3 0 0
V3 0 2 0
Rg int 0 24k
E3 in 0 inp inn 1
R24 in 0 1meg
D10 outb Vth D
Dl l out Vth D
Gout outb out pole 0 1m
Routb out Vss 0.8k
Rout Vss outb 0.8k
* Power rail voltage sources
Vss Vss 0-1.3
VthVth00
.ENDS comp8a_s
```

CLAT1

```
.SUBCKT c_lat1  inp inn clk out outb
* comp with latch
.model DV D
.MODEL ECLFF UEFF (TPCLKLH=1p TPCLKHL=1p)
R11 inp inn 100meg
G1 0 int value {5*(10^-3)*TANH((V(in))/0.06)}
E2 1 0 int 0 10
Rp 1 pole 0.75k
Cp pole 0 1p
CSR int 0 0.5p
D8 0 int DV
D9 int 0 DV
Rg int 0 2.4k
E3 in 0 inp inn 1
R24 in 0 1meg
D11 2 Vth D
Gout 0 2 pole 0 1m
```

```
Routb 2 Vss 0.8k
U1 DFF out outb $D_HI $D_HI clk 2 ECLFF
* Power rail voltage sources
Vss Vss 0 -1.3
Vth Vth 0 0
.ENDS c_lat1
```

CLAT2

```
.SUBCKT c_lat2  inp inn clk out outb
.model DV D
.MODEL ECLFF UEFF (TPCLKLH=1p TPCLKHL=1p)
R11 inp1 inn 100meg
G2 0 int value {5*(10^-3)*TANH((V(in))/0.06)}
E2 1 0 int 0 10
Rp 1 pole 0.75k
Cp pole 0 1p
CSR int 0 0.5p
D8 0 int DV
D9 int 0 DV
Rg int 0 2.4k
E3 in 0 inp1 inn 1
R24 in 0 1meg
D11 login Vth D
Gout 0 login pole 0 1m
Routb login Vss 0.8k
U1 DFF out outb $D_HI $D_HI clk login ECLFF
I1 0 int 4m
Vos inp inp1 0.065
* Power rail voltage sources
Vss Vss 0 -1.3
Vth Vth 0 0
.ENDS c_lat2
```

FASTCOMP

```
fastcomp
.model q npn (IS=3e-15 BF=100 IKF=25m CJE=0.1P TF=8p JC=0.25P)
.op method=gear
Q1 1 inp 2 Q
Q2 3 inn 2 Q
R1 VDD 1 200
R2 VDD 3 200
Ct1 200.5p
I1 2 0 5ma
```

```
Q3 4 1 5 Q
Q4 6 3 5 Q
R3 VDD 4 200
R4 VDD 6 200
Ct2 5 0 0.5p
I2 5 0 5ma
V1  inp 0 0 AC 1 PWL
+ 0,-0.5
+ IE-09,-0.5
+ 2E-09,0.5
V2 inn 0 0
Q5 outb 4 7 Q
Q6 out 6 7 Q
R5 VDD outb 500
R6 VDD out 500
Ct3 7 0 0.2p
I3 7 0 2ma
C1 VDD 1 0.35p
C2 VDD 3 0.35p
C3 VDD 4 0.35p
C4 VDD 6 0.35p
C5  VDD outb 0.8p
C6 VDD out 0.8p
* Power rail voltage sources
VDD VDD 0 5
.END
```

CHAPTER 3

3BFLASH

```
3bflash
.model  ATOD ATOD (VLMAX=-1.8 VHMIN=-0.8)
.model DV D
.model D D
.inc  C:\data\converter\flash\comp8a_s.lib
X1 IN 1 2 3 COMP8A_S
X2 IN 4 5 6 COMP8A_S
X3 IN 7 8 9 COMP8A_S
X4 IN 10 11  12 COMP8A_S
X5 IN 13  14 15 COMP8A_S
X6 IN 16 17  18 COMP8A_S
X7 IN 19 20 21  COMP8A_S
R1 VRT 1  100
R2 1 4 100
```

```
R3 4 7 100
R4 7 10 100
R5 10 13 100
R6 13 16 100
R7 16 19 100
R8 19 VRB 100
U1 CLOCK 22 clock TPER=8n (0,-1.8) (4n ,-0.8)
U2 OR(4) LSB 23 24 25 26 TPCLKLH=1e-12,TPCLKHL=1e-12
U3 OR(4) B2 23 27 25 28 TPCLKLH=1e-12,TPCLKHL=1e-12
U4 OR(4) MSB 23 27 24 29 TPCLKLH=1e-12,TPCLKHL=1e-12
Vin IN 0 PWL
+ 0,-1
+ 1E-06,1
U5 DFF 23 27 $D_HI $D_HI 22 3 TPCLKLH=1e-12,TPCLKHL=1e-12 IS=0 ECLFF
U6 DFF 27 24 $D_HI $D_HI 22 6 TPCLKLH=1e-12,TPCLKHL=1e-12 IS=0 ECLFF
U7 DFF 24 29 $D_HI $D_HI 22 9 TPCLKLH=1e-12,TPCLKHL=1e-12 IS=0 ECLFF
U8 DFF 29 25 $D_HI $D_HI 22 12 TPCLKLH=1e-12,TPCLKHL=1e-12 IS=0 ECLFF
U9 DFF 25 28 $D_HI $D_HI 22 15 TPCLKLH=1e-12,TPCLKHL=1e-12 IS=0 ECLFF
U10 DFF 28 26 $D_HI $D_HI 22 18 TPCLKLH=1e-12,TPCLKHL=1e-12 IS=0 ECLFF
U11 DFF 26 30 $D_HI $D_HI 22 21 TPCLKLH=1e-12,TPCLKHL=1e-12 IS=0 ECLFF
* Power rail voltage sources
VRT VRT 0 +1V
VRB VRB 0 -1V
.INC 3BFLASH.CMD
```

16COMPS

```
.SUBCKT 16comps RT RT_1 AIN clk RB "D" A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11
A12 A13 A14 A15 "U"
*            1  2   3  4  5  6  7   8  9 10 11  12 13 14 15  16  17  18  19  20  21  22
* 16comps
.model OR UGATE (TPLH=1p TPHL=1p)
.model ATOD ATOD (VLMAX=-1.295 VHMIN=-1.305)
.inc C:\data\topspice\Circuits\c_lat1.lib
R8 1 2 5
R7 2 3 5
R6 3 4 5
R5 4 5 5
R1 RT RT_1 5
R2 RT_1 6 5
R3 6 7 5
R4 7 5 5
R16 8 9 5
R15 9 10 5
R14 10 11 5
R13 11 12 5
R9 1 13 5
```

```
R10 13 145
R11 14 155
R12 15 12 5
X1 AIN RT_1 clk "U" A15 C_LAT1
X2 AIN 6 clk A15 A14 C_LAT1
X3 AIN 7 clk A14 A13 C_LAT1
X4 AIN 5 clk A13 A12 C_LAT1
X5 AIN 4 clk A12 A11 C_LAT1
X6 AIN 3 clk A11 A10 C_LAT1
X7 AIN 2 clk A10 A9 C_LAT1
X8 AIN clk A9 A8 C_LAT1
X9 AIN 13 clk A8 A7 C_LAT1
X10 AIN 14 clk A7 A6 C_LAT1
X11 AIN 15 clk A6 A5 C_LAT1
X12 AIN 12 clk A5 A4 C_LAT1
X13 AIN 11 clk A4 A3 C_LAT1
X14 AIN 10 clk A3 A2 C_LAT1
R17 RB 8 5
X15 AIN 9 clk A2 A1 C_LAT1
X16 AIN 8 clk A1 "D" C_LAT1
.ENDS 16comps
```

6BADA

```
6bada
.model OR UGATE (TPLH=lp TPHL=lp)
.inc C:\data\converter\spicc_models\flash\16comps.lib
.OPTIONS ACCT ITL1=500 ITL2=200 ITL4=40 ITL5=0 LIMPTS=10000
.model ATOD ATOD (VLMAX=-1.295 VHMIN=-1.305)
.model DTOA DTOA (TSW0=lp TSWl=lp TSWX=lp TSWZ=lp)
X3 RT RT 1 2 ref1 t48 t49 t50 t51 t52 t53 t54 t55 t56 t57 t58 t59 t60 t61 t62
+ t63 t64 16COMPS
X4 ref1 ref1 3 2 ref2 t32 t33 t34 t35 t36 t37 t38 t39 t40 t41 t42 t43 t44 t45
+ t46 t47 t48 16COMPS
X5 ref2 ref2 4 2 ref3 t16 t17 t18 t19 t20 t21 t22 t23 t24 t25 t26 t27 t28 t29
+ t30 t31 t32 16COMPS
X6 ref3 ref3 5 2 RB 6 t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 t11 t12 t13 t14 t15 t16
+ 16COMPS
Vin 1 0 PWL
+ 0,0
+ 1.6E-07, 1.6
U1 OR(32) 1sb t63 t61 t59 t57 t55 t53 t51 t49 t47 t45 t43 t41 t39 t37 t35 t33
+ t31 t29 t27 t25 t23 t21 t19 t17 t15 t13 t11 t9 t7 t5 t3 t1
U2 OR(32) b5 t63 t62 t59 t58 t55 t54 t51 t50 t47 t46 t43 t42 t39 t38 t35 t34
+ t31 t30 t27 t26 t23 t22 t19 t18 t15 t14 t11 t10 t7 t6 t3 t2
U3 OR(32) b4 t63 t62 t61 t60 t55 t54 t53 t52 t47 t46 t45 t44 t39 t38 t37 t36
+ t31 t30 t29 t28 t23 t22 t21 t20 t15 t14 t13 t12 t7 t6 t5 t4
U4 OR(32) b3 t63 t62 t61 t60 t59 t58 t57 t56 t47 t46 t45 t44 t43 t42 t41 t40
```

```
+ t31 t30 t29 t28 t27 t26 t25 t24 t15 t14 t13 t12 t11 t10 t9 t8
U5 OR(32) b2 t63 t62 t61 t60 t59 t58 t57 t56 t55 t54 t53 t52 t51 t50 t49 t48
+  t31 t30 t29 t28 t27 t26 t25 t24 t23 t22 t21 t20 t19 t18 t17 t16
U6 OR(32) msb t63 t62 t61 t60 t59 t58 t57 t56 t55 t54 t53 t52 t51 t50 t49 t48
+ t47 t46 t45 t44 t43 t42 t41 t40 t39 t38 t37 t36 t35 t34 t33 t32
U7 CLOCK 2 clock TPER=1n (0,-1.8) (0.5n ,-0.8)
Vref 7 0  1.6V
X7 $D_LO $D_LO lsb b5  b4  b3 b2 msb 7 ADA_OUT DAC8BIT PARAMS:
TDELAY=lp
R1 1 3 1k
R2 3 4 1k
R3 4 5 1k
C1 3 0 0.7p
C2 4 0 0.7p
C3 5 0 0.7p
* Power rail voltage sources
VRT RT 0+1.6V
VRB RB 0 0V
.INC 6bada_DLY1.CMD
.END
```

6BADA_DLY1

```
6bada_dlyl
.model  OR UGATE (TPLH=lp TPHL=lp)
.inc C:\data\converter\spice_models\flash\l6comps.lib
.OPTIONS     ACCT ITL1=500  ITL2=200  ITL4=40     ITL5=0  LIMPTS=10000
.model  ATOD ATOD (VLMAX=-1.295 VHMIN=-1.305)
.model  DTOA  DTOA(TSWO=lp TSW1=lpTSWX=lp TSWZ=lp)
X3 RT RT 1 2 ref1 t48 t49 t50 t51 t52 t53 t54 t55 t56 t57 t58 t59 t60 t61 t62
+ t63 t64 16COMPS
X4 ref1 ref1 3 2 ref2 t32 t33 t34 t35 t36 t37 t38 t39 t40 t41 t42 t43 t44 t45
+ t46 t47 t48 16COMPS
X5 ref2 ref2 4 2 ref3 t16 t17 t18 t19 t20 t21 t22 t23 t24 t25 t26 t27 t28 t29
+ t30 t31 t32  16COMPS
X6 ref3 ref3 5 2 RB 6 t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 t11 t12 t13 t14 t15 t16
+ 16COMPS
Vin 1 0 PWL
+ 0,0
+ 1.6E-07, 1.6
U1 OR(32) lsb t63 t61 t59 t57 t55 t53 t51 t49 t47 t45 t43 t41 t39 t37 t35 t33
+ t31 t29 t27 t25 t23 t21 t19 t17 t15 t13 t11 t9 t7 t5 t3 t1
U2 OR(32) b5 t63 t62 t59 t58 t55 t54 t51 t50 t47 t46 t43 t42 t39 t38 t35 t34
+ t31 t30 t27 t26 t23 t22 t19 t18 t15 t14 t11 t10 t7 t6 t3 t2
U3 OR(32) b4 t63 t62 t61 t60 t55 t54 t53 t52 t47 t46 t45 t44 t39 t38 t37 t36
+ t31 t30 t29 t28 t23 t22 t21 t20 t15 t14 t13 t12 t7 t6 t5 t4
U4 OR(32) b3 t63 t62 t61 t60 t59 t58 t57 t56 t47 t46 t45 t44 t43 t42 t41 t40
+ t31 t30 t29 t28 t27 t26 t25 t24 t15 t14 t13 t12 t11 t10 t9 t8
```

```
U5 OR(32) b2 t63 t62 t61 t60 t59 t58 t57 t56 t55 t54 t53 t52 t51 t50 t49 t48
+ t31 t30 t29 t28 t27 t26 t25 t24 t23 t22 t21 t20 t19 t18 t17 t16
U6 OR(32) msb t63 t62 t61 t60 t59 t58 t57 t56 t55 t54 t53 t52 t51 t50 t49 t48
+ t47 t46 t45 t44 t43 t42 t41 t40 t39 t38 t37 t36 t35 t34 t33 t32
U7 CLOCK 2 clock TPER=ln (0,-1.8) (0.5n ,-0.8)
Vref 7 0 1.6V
X7 $D_LO $D_LO Isb b5 M b3 b2 msb 7 ADA_OUT DAC8BIT PARAMS:
TDELAY=lp
R1 1 3 1k
R2 3 4 1k
R3 4 5 1k
C1 3 0 0.7p
C2 4 0 0.7p
C3 5 0 0.7p
* Power rail voltage sources
VRT RT 0 + 1.6V
VRB RB 0 0V
.END
```

CHAPTER 4

NLR

```
.INC c:\data\converter\sha\vcr.lib
X1 B A CONT A VCR PARAMS: RNOM=1
Vin A 002PWL
+ 0,0
+ 1.3E-07,5
Vcont CONT 0 5
RL B 0 1u
.END
```

VCR

```
* Voltage controlled resistor model
.SUBCKT VCR 1 2 3 4
+ PARAMS: RNOM=1 ;nominal R at V(3,4)=0V
RIN 3 4 lG ;to avoid open node
* Generate voltage proportional to resistance vs controlling voltage (3,4)
* the normalized R values (R/RNOM) can not be zero.
ER vr 0 POLY(l) (3,4) 48.55 37.33 -27 7.8 -0.773
RE vr 0 1 g ;dummy load
* Voltage dependent resistor
GR 1 2 VALUE = {V(l,2)/RNOM/(V(vr)+1u)} ;1u added to avoid divide by zero
```

```
.ENDS VCR
```

DROOP

```
.MODEL SW VSWITCH (VON=2 VOFF=1.95 RON=1 ROFF=1000G)
.OPTION OPTS NUMDGT=8 FILOUT METHOD=GEAR
V1 IN 0 SIN 2 1.95 1.07421875meg
R1 IN 0 1meg
S1 IN shout 1 T_H1 0 SW
C1 shout1 0 10p IC=2.48
Vth T_H1 0 PULSE 0 5 0 0.1n 0.1n 1n 40n
R3 shout1 0 1g
Idroop shout1 0 1n
.END
```

JITTER

```
.MODEL SW VSWITCH (VON=2 VOFF=1.995 RON=1 ROFF=1000G)
.INC c:\data\converter\sha\vcr.lib
.OPTION OPTS NUMDGT=8 FILOUT METHOD=GEAR
V1 IN 0 PWL
+ 0,0
+ 2.5E-10,1
R1 IN 0 1 meg
S1 IN shout1 T_H N SW
C1 shoutl 0 l0p IC=2.48
Vth T_H 0 PWL
+ 0,4
+ 2.5E-10,0
R3 shout1 0 1g
R4 IN 0 1 meg
S2 IN shout2 T_H 0 SW
C2 shout2 0 l0p IC=2.48
R6 shout2 0 1g
Vnoise N 0 FILE v1.dat
.END
```

NONLINEAR CAP

```
.MODEL SW VSWITCH (VON=2 VOFF=1.95 RON=1 ROFF=1000G)
.INC c:\data\convertertsha\vcr.lib
.OPTION OPTS NUMDGT=8 FILOUT METHOD=GEAR
V1 IN 0 SIN 2 1.95 4.98046875meg
```

```
R1 IN 0 1meg
S1 IN shout1 T_Hl  0  SW
C1 shout1 0 poly 10p l0.000lp l0.0000lp
Vth T_H1 0 PULSE 0 5 0 0.5n 0.5n 20n 40n
R3 shout10 lg
R4 IN 0   1meg
S2 IN shout2 T_H1 0 SW
C2 shout2 0 l0p IC=2.48
R6 shout2 0 1g
.END
```

SWITCH RESISTANCE MODULATION

```
.MODEL SW VSWITCH (VON=2 VOFF=1.95 RON=1 ROFF=1000G)
.INC c:\data\converter\sha\vcr.lib
V1 IN 0 SIN 2 1.95 0.4892578125meg
R1 IN 0 1meg
S1 1 shout1 T_1 0 SW
C1 shoutl 0 l0p IC=2.48
Vth T_H1 0 PULSE 0 5 0 0.5n 0.5n 20n 40n
X2 1 IN T_H1 IN VCR PARAMS: RNOM=1
R2 1 0 1g
R3 shoutl 0 1g
R4 IN 0 1meg
S2 2 shout2 T_H1 0 SW
C2 shout2 0 l0p IC=2.48
R5 2 0 1g
R6 shout2 0 1g
R7 IN 2 65
.END
```

CHAPTER 5

COMP1SAR

```
.SUBCKT comp1sar  inp inn logout
* complsar
*

R12 inp inn l00meg
G2 0 int value{1*(10^-3)*TANH((V(in))/0.25)}
E3 1 0 int 010
R13 1 pole 1k
C1 pole 0 1p
C2 int 0 0.25p
```

```
D9 0 int DV
D10 int 0 DV
R14 int 0 4.5k
E4 in 0 inp inn 1
R25 in 0 1 meg
E5 anout 0 TABLE {V(pole)*10} -2,0V 2,4V
U20 BUF logout anout
.ENDS comp1sar
```

MACRO_SAR

```
.SUBCKT macro_sar  ADA IN CLK IN1
* 8 BIT ADC SAR a
.inc C:\data\converter\sar\comp1sar.lib
.MODEL SREGD USREG TPLH=2n TPHL=2n
.MODEL  NANDD UGATE (TPLH=2n TPHL=2n)
.MODEL  BUFD UGATE (TPLH=2n TPHL=2n)
.MODEL  INVD UGATE (TPLH=2n TPHL=2n)
U10 DFF MSB 1 T7 D7 $D_HI $D_HI  TPCLKLH=2n  TPCLKHL=2n
U11  DFF B6 2 T6 D6 $D_HI $D_HI  TPCLKLH=2n  TPCLKHL=2n
U12 DFF B5 3 T5 D5 $D_HI $D_HI  TPCLKLH=2n  TPCLKHL=2n
U13 DFF B4 4 T4 D4 $D_HI $D_HI  TPCLKLH=2n  TPCLKHL=2n
U14 DFF B3 5 T3 D3 $D_HI $D_HI  TPCLKLH=2n  TPCLKHL=2n
U16 DFF B2 6 T2 D2 $D_HI $D_HI  TPCLKLH=2n  TPCLKHL=2n
U17 DFF B1 7 T1 D1 $D_HI $D_HI  TPCLKLH=2n  TPCLKHL=2n
U18 DFF B0 8 T0 D0 $D_HI $D_HI  TPCLKLH=2n  TPCLKHL=2n
X1 B0 B1 B2 B3 B4 B5 B6 MSB ref DAC DAC8BIT
Vref ref 0 2.048
R1 IN 0 1meg
R2 DAC ADA 100
C1 ADA 0 20p
X2 ADA IN CMP comp1sar
U19 SREG(10) SBGN S7 S6 S5 S4 S3 S2 S1 S0 S_1 CLRCMD CLK 9 SREGD
U24 NAND T7 S7 CLKD
U25 NAND T6 S6 CLKD
U26 NAND T5 S5 CLKD
U27 NAND T4 S4 CLKD
U28 NAND T3 S3 CLKD
U29 NAND T2 S2 CLKD
U30 NAND T1 S1 CLKD
U31 NAND T0 S0 CLKD
U32 BUF CLKD CLK BUFD
U33 INV CLKBD CLK INVD
U61 AND D7 CLRCMD 10
U62 AND D6 CLRCMD 11
U63 AND D5 CLRCMD 12
U64 AND D4 CLRCMD 13
U65 AND D3 CLRCMD 14
```

```
U66 AND D2 CLRCMD 15
U67 AND D1 CLRCMD 16
U68 AND D0 CLRCMD 17
U69 NAND(3) 10 CMP CLKBD S7
U70 NAND(3) 11 CMP CLKBD S6
U71 NAND(3) 12 CMP CLKBD S5
U72 NAND(3) 13 CMP CLKBD S4
U73 NAND(3) 14 CMP CLKBD S3
U74 NAND(3) 15 CMP CLKBD S2
U75 NAND(3) 16 CMP CLKBD S1
U76 NAND(3) 17 CMP CLKBD S0
U77 NAND 18 S_1 CLKD
U78 DELAY CLRCMD 18
U79 AND 19 CLKBD IN1
U80 DELAY 9 19
.ENDS macro_sar
```

DAC8NL

```
.SUBCKT DAC8NL_LSB D0 D1 D2 D3 D4 D5 D6 D7 REF OUT PARAMS:
TDELAY=1ns
* 8-bit DAC (digital-to-analog converter).
* The digital inputs D0 thru D7 MUST be logic signals - they must be
* connected to U digital device outputs. If you want to apply analog voltages
* insert U BUF elements between the signals and inputs.
V1 1 0 1V
O0 DTOA A0 0 1 D0 DACMODDA
R0 A0 0 1G
O1 DTOA A1 0 1 D1 DACMODDA
R1 A1 0 1G
O2 DTOA A2 0 1 D2 DACMODDA
R2 A 20 1G
03 DTOA A3 0 1 D3 DACMODDA
R3 A3 0 1G
O4 DTOA A4 0 1 D4 DACMODDA
R4 A4 0 1G
O5 DTOA A5 0 1 D5 DACMODDA
R5 A5 0 1G
O6 DTOA A6 0 1 D6 DACMODDA
R6 A6 0 1G
O7 DTOA A7 0 1 D7 DACMODDA
R7 A7 0 1G
E1 OUT 0 VALUE
+ {(V(A7)*12.8+V(A6)*6.4+V(A5)*3.2+V(A4)*1.6+V(A3)*.8+V(A2)*.4+V(A1)*.2+
+ V(A0)*0.2)/25.6*V(REF)}
ROUT OUT 0 1G
.MODEL DACMODDA DTOA (RLO0=200 RHI0=10MEG RLO1=10MEG RHI1=200
+ RLOX=10MEG RHIX=10MEG RLOZ=10MEG RHIZ=10MEG
```

```
+ TSWO={TDELAY} TSW1={TDELAY} TSWX=(TDELAY} TSWZ={TDELAY})
.ENDS DAC8NL_LSB
```

CHAPTER 6

8B_FOLD

```
.inc C:\data\converter\fold\sincos.lib
.inc C:\data\converter\fold\interp1.lib
.inc C:\data\CONVERTER\FOLD\CDP.LIB
.inc  C:\data\converter\fold\fold_comp.lib
.options acct lvltim=1  numdgt=6 chgtol=1e-16 method=gear
.model  ATOD ATOD (VLMAX=1.998 VHMIN=2.002)
.model  OR1  UGATE (TPLH=lp TPHL=lp)
V1 in 0 0.22 PWL
+ 0,0.22
+ 2.5E-06,1.98
R1 c3 0 10
R2 s3 0 10
X2 s3 c3 os3 s4 INTERP1
Vos os3 0 -0.03925
X3 c3 c3 os3  1 INTERP1
X4 s3 s3 os3 2 INTERP1
El c4 0 1 2 0.5
R3 c4 0 10
X5 s4 c4 0 s5 INTERP1
X6 c4 c4 0 3 INTERP1
X7 s4 s4 0 4 INTERP1
E2 c5 0 3 4 0.5
R4 c5 0 10
X8 s5 c5 0 s6 INTERP1
X9 c5 c5 0 5 INTERP1
X10 s5 s5 0 6 INTERP1
E3 c6 0 5 6 0.5
R5 c6 0 10
U1 CLOCK clk clock TPER=10n 0 0 9n 1
U2 OR B2 7 8 OR1
X18 os3 s3 clk B3 9 FOLD_COMP
X19 s4 0 clk B4 10 FOLD_COMP
X20 s5 0 clk B5 11 FOLD_COMP
X21 s6 0 clk B6 12 FOLD_COMP
X22 in TQ clk1 7 13 FOLD_COMP
X23 in HLF clk1 13 8 FOLD_COMP
X24 in OQ clk1 8 14 FOLD_COMP
U3 OR B1 7 13 OR1
```

```
X25 in s3 s3 s3 s3 s3 c3 c3 c3 c3 c3 FT TQ HLF OQ FB SINCOS
X26 s6 c6 0 s7 INTERP1
X27 c6 c6 0 15 INTERP1
X28 s6 s6 0 16 INTERP1
E4 c7 0 15 16 0.5
R6 c7 0 10
X29 s7 c7 0 s8 INTERP1
X30 c7 c7 0 17 INTERP1
X31 s7 s7 0 18 INTERP1
E5 c8 0 17 18 0.5
R7 c8 0 10
X32 s7 0 clk B7 19 FOLD_COMP
X33 s8 0 clk B8 20 FOLD_COMP
X34 B8 B7 B6 B5 B4 B3 B2 B1 21 22 DAC8BIT
V2 21 0 1.98
E6 ADA 23 22 0 0.8888888
R8 0 ADA 1meg
V3 23 0 0.22
U4 BUF 24 clk
U5 BUF 25 24
U6 BUF clk1 25
.END
```

SINCOS

```
.inc C:\data\CONVERTER\FOLD\CDP.LIB
Vin in 0 0.22
Rloads 0 outs 0.2
X1 in 1 outs 2 cdp
X3 in 3 outs 4 cdp
X5 in 5 outs 6 cdp
V3 7 0 2.2
G1 tsts 0 value {0.04+0.02456*sin(v(in)*9*3.1415/1.98)}
R7 tsts 0 1
X6 in 8 outs 9 cdp
X7 in 10 outs 11 cdp
R8 12 1 500
R9 12 2 500
R10 13 2 500
R11 13 3 500
R12 14 3 500
R13 14 4 500
R14 15 4 500
R15 15 5 500
R16 16 5 500
R17 16 6 500
R18 17 6 500
R19 17 8 500
```

```
R20 18 8 500
R21 18 9 500
R22 19 9 500
R23 19 10 500
R24 20 10 500
R25 20 11 500
X8 in 12 outc 13 cdp
X9 in 14 outc 15 cdp
X10 in 16 outc 17 cdp
X11 in 18 outc 19 cdp
Rloadc outc 0 0.2
R26 21 11 500
R27 21 7 500
X12 in 20 outc 21 cdp
V4 1 0 0.0
G2 tstc 0 value {0.04-0.02456*cos(v(in)*9*3.1415/1.98)}
R28 tstc 0 1
.END
```

CDP

```
cdp
Vin vin 0 0.5 AC 1
Vinlo reflo 0 0.4
G1 0 int value {1*(10^-3)*TANH((V(n1))/0.1)}
E3 1 0 int 0 1
Rp 1 2 100
Cp 2 0 0.3p
E1 n1 0 vin reflo 1
Rin n1 0 1meg
E2 n2 0 vin refhi 1
Vinhi refhi 0 0.6
G2 int 0 value {1*(10^-3)*TANH((V(n2))/0.1)}
R1 int 0 200
R2 0 n2 1meg
Gout 0 out 2 0 1
R3 out 0 400
.END
```

INTERP1

```
.SUBCKT interp1 A B OS AB
* interp1
* cdp
G1 0 n4 value {1*(10^-3)*TANH((V(n1))/0.1)}
Rp 1 AB 100
```

```
Cp AB 0 lp
E1 n1 0 A OS 1
Rin n1 0 1meg
R1 n4 0 200
G2 0 n3 value {V(n4)/200*TANH(V(n2)/0.1)}
E2 n2 0 B os 1
R2 n2 0 1meg
R3 n3 0 200
E3 1 0 n3 0 4
.ENDS interp1
```

FOLD_COMP

```
.SUBCKT fold_comp  inp inn CLK Q  QB
*  fold_comp
R12 inp inn 100meg
G2 0 int value {1*(10^-3)*TANH((V(in))/0.05)}
E3 1 0 int 0 1
R13 1 pole 1k
C1 pole 0 1p
C2 int 0 0.25p
D9 0 int DV
D10 int 0 DV
E4 in 0 inp inn 1
R25 in 0 1 meg
E5 anout 0 TABLE {V(pole)*10} -2m, 0V 2m,4V
U21 DFF q QB $D_HI $D_HI CLK anout
Rsr int 0 4k
.ENDS fold_comp
```

CHAPTER 7

16COMPS2

```
.SUBCKT 16comps2  RT RT_1 AIN clk RB "D" A1 A2 A3 A4 A5 A6 A7 A8 A9 A10
A11 A12 A13 A14 A15 "U"
*              1  2   3   4   5  6  7   8  9 10 11  12 13 14 15  16  17  18  19  20  21  22
* 16comps2
* 4 bit flash
.model  OR UGATE (TPLH=1p TPHL=1p)
.model  ATOD ATOD (VLMAX=-1.295 VHMIN=-1.305)
.inc C:\data\converter\pipeline\c_lat2.1ib
R8 1 2 5
R7 2 3 5
```

```
R6 3 4 5
R5 4 5 5
R1 RT RT_1 5
R2 RT_1 6 5
R3 6 7 5
R4 7 5 5
R16 8 9 5
R15 9 10 5
R14 10 11 5
R13 11 12 5
R9 1 13 5
R10 13 14 5
R11 14 15 5
R12 15 12 5
X1 AIN RT_1 clk "U" A15 C_LAT2
X2 AIN 6 clk A15 A14 C_LAT2
X3 AIN 7 clk A14 A13 C_LAT2
X4 AIN 5 clk A13 A12 C_LAT2
X5 AIN 4 clk A12 A11 C_LAT2
X6 AIN 3 clk A11 A10 C_LAT2
X7 AIN 2 clk A10 A9 C_LAT2
X8 AIN 1 clk A9 A8 C_LAT2
X9 AIN 13 clk A8 A7 C_LAT2
X10 AIN 14 clk A7 A6 C_LAT2
X11 AIN 15 clk A6 A5 C_LAT2
X12 AIN 12 clk A5 A4 C_LAT2
X13 AIN 11 clk A4 A3 C_LAT2
X14 AIN 10 clk A3 A2 C_LAT2
R17 RB 8 5
X15 AIN 9 clk A2 A1 C_LAT2
X16 AIN 8 clk A1 "D" C_LAT2
.ENDS 16comps2
```

THR4P4

```
.inc C:\data\converter\pipeline\4B_flash.lib
.model OR UGATE(TPLH=1p TPHL=1p)
.model ATOD ATOD (VLMAX=-1.295 VHMIN=-1.305)
.OPTION OPTS METHOD=GEAR
.model DTOA DTOA (TSW0=1p TSW1=1p TSWX=1p TSZ=1p)
.MODEL lat1 UGFF (TPCLKLH=0.5n TPCLKHL=0.5n)
.MODEL SW VSWITCH (VON=2 VOFF=1.5 RON=100 ROFF=1000G)
.inc C:\data\converter\pipeline\5B_FLASH.lib
.model NOR UGATE (TPLH=1p TPHL=1p)
.model PSREG USREG (TPLH=1p TPHL=1p TPLOAD=1p)
.model ADD UALU (TPLH=1p TPHL=1p)
X1 1 2 SHOUT1 CRS D3 D2 D1 D0 4B_flash
Vtop rtop 0 2.56
```

```
Vin IN 0 SIN 1.28 1.25 195.3125k
X9 D0 D1 D2 D3 rtop 3 DAC4BIT PARAMS: TDELAY=1p
Eresidue res_gain 0 SHOUT2 0 16
Rbw res_gain an_in_fine 100
Cbw an_in_fine 0 2.5pF
X11 s0 s1 s2 s3 s4 s5 s6 s7 ada_top ADA_OUT DAC8BIT PARAMS: TDELAY=1p
S1 SHOUT1 IN TH1 0 SW
S2 SHOUT2 4 TH2 0 SW
C1 SHOUTl 0 5p
C2 SHOUT2 0 5p
U15 CLOCK TH1 TPER=10n (0,0) (1n,1) (4n,0)
RDAC 3 C_DAC 300
CDAC C_DAC 0 1.5pF
Edac 4 0 SHOUT1 C_DAC 1
U31 PSREG 5 6 7 8 $D_HI $D_HI $D_HI 9 $D_LO $D_LO D3 D2 D1 D0 PSREG
U40 CLOCK CRS TPER=10n (0,0) (4n,1) (4.5n,0)
U41 CLOCK TH2 TPER=10n (0,0) (4.5n,l) (9n,0)
U42 CLOCK 9 TPER=10n (0,0) (4.5n,l) (5n,0)
U43 CLOCK FN TPER=10n (0,0) (9n,l) (9.5n,0)
U44 CLOCK 10 TPER=10n (0,0) (9.5n,l) (10n,0)
X12 rtop 0 an_in_fine FN DF3 DF2 DF1 DF0 4B_flash
U45 PSREG(8) s7 s6 s5 s4 s3 s2 s1 s0 $D_HI $D_HI $D_HI 10 $D_LO $D_LO 5 6 7 8
+ DF3 DF2 DF1 DF0
Vos 1 rtop 80m
V1 2 0 80m
* Power rail voltage sources
Vada_top ada_top 0 2.56
.END
```

THR4P5

```
.inc C:\data\converter\pipeline\4B_flash.lib
.model OR UGATE (TPLH=1p TPHL=1p)
.model ATOD ATOD (VLMAX=-1.295 VHMIN=-1.305)
.OPTION OPTS NUMDGT=10 FILOUT METHOD=GEAR
.model DTOA DTOA (TSW0=1p TSW1=1p TSWX=1p TSZ=1p)
.MODEL lat1 UGFF (TPCLKLH=0.5n TPCLKHL=0.5n)
.MODEL SW VSWITCH (VON=2 VOFF=1.5 RON=100 ROFF=1000G)
.inc C:\data\converter\pipeline\5B_FLASH.lib
.model NOR UGATE (TPLH=1p TPHL=1p)
.model PSREG USREG (TPLH=1p TPHL=1p TPLOAD=1p)
.model ADD UALU (TPLH=1p TPHL=1p)
X1 1 rbot SHOUT1 CRS D3 D2 D1 D0 4B_flash
Vtop rtop 0 2.56
Vbot rbot 0 0
Vin IN 0 SIN 1.28 1.15 195.3125k
X9 D0 D1 D2 D3 rtop 2 DAC4BIT PARAMS: TDELAY=1p
Eresidue res_gain 0 SHOUT2 0 8
```

```
Rbw res_gain an_in_fine 100
Cbw an_in_fine 0 2.5pF
X11 s0 s1 s2 s3 s4 s5 s6 s7 ada_top ADA_OUT DAC8BIT PARAMS: TDELAY=1p
S1 SHOUT1 IN TH1 0 SW
S2 SHOUT2 3 TH2 0 SW
C1 SHOUT1 0 5p
C2 SHOUT2 0 5p
U15 CLOCK TH1  TPER=10n (0,0)  (1n,1) (4n,0)
U18 ADD(8) q7 q6 q5 q4 q3 q2 q1 q0 co DD3 DD2 DD1 DD0 $D_LO $D_LO $D_LO
$D_LO
+ $D_LO $D_LO $D_LO DDF4 DDF3 DDF2 DDF1 DDF0 $D_LO ADD
X12 fine_ref_t an_in_fine FN fine_ref_b DF0 DF1  DF2 DF3 DF4 5B_FLASH
Vos_top rtop fine_ref_t 640m
Vos_bot rbot fine_ref_b 640m
U20 OR s7 q7 co
U21 OR s6 q6 co
U22 OR s5 q5 co
U23 OR s4 q4 co
U24 OR s3 q3 co
U25 OR s2 q2 co
U26 OR s1 q1 co
U27 OR s0 q0 co
RDAC 2 C_DAC 300
CDAC C_DAC 0 1.5pF
Edac 3 0 SHOUT1 C_DAC 1
U29 PSREG(9) DDF4 DDF3 DDF2 DDF1 DDF0 DD3 DD2 DD1 DD0 $D_HI $D_HI
$D_HI 4
+ $D_LO $D_LO DF4 DF3 DF2 DF1 DF0 5 6 7 8 PSREG
U31 PSREG 5 6 7 8 $D_HI $D_HI $D_HI 9 $D_LO $D_LO D3 D2 D1 D0 PSREG
U40 CLOCK CRS  TPER=10n (0,0)  (4n,1) (4.5n,0)
U41 CLOCK TH2  TPER=10n (0,0)  (4.5n,1) (9n,0)
U42 CLOCK 9  TPER=10n (0,0)  (4.5n,1) (5n,0)
U43 CLOCK FN  TPER=10n (0,0)  (9n,1) (9.5n,0)
U44 CLOCK 4  TPER=10n (0,0)  (9.5n,1) (10n,0)
Vos 1 rtop 0
* Power rail voltage sources
Vada_top ada_top 0 2.56
.END
```

4B_FLASH

```
.SUBCKT4B_flash  Ref_top Ref_bot AN_IN clk  MSB B2 B3 LSB
.model  OR UGATE (TPLH=1pTPHL=1p)
.inc C:\data\converter\pipeline\16comps2.lib
.options acct lvltim=1 numdgt=6 chgtol=1e-16
.model ATOD ATOD (VLMAX=-1.295 VHMIN=-1.305)
.model  DTOA  DTOA (TSW0=1p TSW1=1p TSWX=1p TSWZ=1p)
X4 Ref_top Ref_top AN_IN clk Ref_bot 1 t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 t11 t12
```

```
+ t13 t14 t15 t16 16COMPS2
U9 OR(8) 2 t15 t14 t13 t12 t11 t10 t9 t8
U10 OR(8) 3 t15 t14 t13 t12 t7 t6 t5 t4
U11 OR(8) 4 t15 t14 t11 t10 t7 t6 t3 t2
U12 OR(8) 5 t15 t13 t11 t9 t7 t5 t3 t1
U14 OR MSB t16 2
U15 OR B2 t16 3
U16 OR B3 t16 4
U17 OR LSB t16 5
.ENDS 4B_flash
```

5B_FLASH

```
.SUBCKT 5B_FLASH ref_top AN_IN clk refb lsb b4 b3 b2 msb
*5bada
.model  OR UGATE (TPLH=1p TPHL=1p)
.inc C:\data\converter\pipeline\16comps2.lib
.options acct 1vltim=1 numdgt=6 chgto1=1e-16
.model  ATOD ATOD (VLMAX=-1.295 VHMIN=-1.305)
.model  DTOA  DTOA (TSW0=1p TSW1=1p TSWX=1p TSWZ=1p)
X5 ref_top ref_top AN_IN clk ref3 t16 t17 t18 t19 t20 t21 t22 t23 t24 t25 t26
+ t27 t28 t29 t30 t31 t32 16COMPS2
X6 ref3 ref3 AN_IN clk refb 1 t1 t2 t3 t4 t5 t6 t7 t8 t9 t10  t11 t12 t13 t14
+ t15 t16 16COMPS2
U15 OR(17) b3 t32 t31 t30 t29 t28 t23 t22 t21 t20 t15 t14 t13 t12 t7 t6 t5 t4
+ OR
U16 OR(17) b2 t32 t31 t30 t29 t28 t27 t26 t25 t24 t15 t14 t13 t12 t11 t10 t9
+ t8 OR
U17 OR(17) msb t32 t31 t30 t29 t28 t27 t26 t25 t24 t23 t22 t21 t20 t19 t18 t17
+ t16 OR
U18 OR(17) lsb t32 t31 t29 t27 t25 t23 t21 t19 t17 t15 t13 t11 t9 t7 t5 t3 t1
+ OR
U19 OR(17) b4 t32 t31 t30 t27 t26 t23 t22 t19 t18 t15 t14 t11 t10 t7 t6 t3 t2
+ OR
.ENDS 5B_FLASH
```

CHAPTER 8

1P5B8

```
.MODEL SW VSWITCH (VON=2 VOFF=1.995 RON=100 ROFF=1000G)
.OPTION OPTS NUMDGT=7 FILOUT METHOD=GEAR  LVLTIM=1 PIVREL=1e-6
.include  c:\data\converter\spice_models\1p5\sub_mdac.lib
.model dff1  ueff (tpclkhl=1p tpclklh=1p)
```

```
.model inv2 ugate (tphl=1p tplh=1p)
.model add1 ualu (tph1=1p tplh=1p)
.model and1 ugate (tph1=1p tplh=1p)
Vp refp cm 0.25
Vn cm refn 0.25
VPOS cm VNEG 1
VNEG VPOS cm 1
Vcm cm 0 2.5
U2 CLOCK ph1 CLOCK TPER=10n (0,0) (0.5n ,1) (5n,0)
U3 CLOCK ph2 CLOCK TPER=10n (0,0) (5.5n ,1) (10n,0)
U7 CLOCK ph1a CLOCK TPER=10n (0,0) (0.5n ,1) (4n,0)
U11 CLOCK ph2a CLOCK TPER=10n (0,0) (5.5n ,1) (9n,0)
X1 1 2 cm refp refn VNEG VPOS b0 bl ph1a ph1 ph2 out outb SUB_MDAC
Vinp 1 0 PWL
+ 0,1.988
+ 4E-06,3.012
Vinn 2 0 PWL
+ 0,3.012
+ 4E-06,1.988
X2 outb out cm refp refn VNEG VPOS b2 b3 ph2a ph2 ph1 out2 out2b SUB_MDAC
X3 out2b out2 cm refp refn VNEG VPOS b4 b5 ph1a ph1 ph2 out3 out3b SUB_MDAC
X4 out3b out3 cm refp refn VNEG VPOS b6 b7 ph2a ph2 ph1 out4 out4b SUB_MDAC
X5 out4b out4 cm refp refn VNEG VPOS b8 b9 ph1a ph1 ph2 out5 out5b SUB_MDAC
U12 DFF 3 4 $D_HI $D_HI clk1 b1 dff1
U13 DFF 5 6 $D_HI $D_HI clk1 b0 dff1
U14 DFF 7 8 $D_HI $D_HI clk2 3 dff1
U15 DFF 9 10 $D_HI $D_HI clk2 5 dff1
U16 DFF 11 12 $D_HI $D_HI clk1 7 dff1
U17 DFF 13 14 $D_HI $D_HI clk1 9 dff1
U18 DFF 15 16 $D_HI $D_HI clk2 11 dff1
U19 DFF 17 18 $D_HI $D_HI clk2 13 dff1
U20 DFF 19 20 $D_HI $D_HI clk2 b3 dff1
U21 DFF 21 22 $D_HI $D_HI clk2 b2 dff1
U22 DFF 23 24 $D_HI $D_HI clk1 19 dff1
U23 DFF 25 26 $D_HI $D_HI clk1 21 dff1
U24 DFF 27 28 $D_HI $D_HI clk2 23 dff1
U25 DFF 29 30 $D_HI $D_HI clk2 25 dff1
U26 DFF 31 32 $D_HI $D_HI clk1 b5 dff1
U27 DFF 33 34 $D_HI $D_HI clk1 b4 dff1
U28 DFF 35 36 $D_HI $D_HI clk2 31 dff1
U29 DFF 37 38 $D_HI $D_HI clk2 33 dff1
U30 DFF 39 40 $D_HI $D_HI clk2 b7 dff1
U31 DFF 41 42 $D_HI $D_HI clk2 b6 dff1
U39 DFF 43 44 $D_HI $D_HI clk1 15 dff1
U40 DFF 45 46 $D_HI $D_HI clk1 17 dff1
U41 DFF 47 48 $D_HI $D_HI clk1 27 dff1
U42 DFF 49 50 $D_HI $D_HI clk1 29 dff1
U43 DFF 51 52 $D_HI $D_HI clk1 35 dff1
U44 DFF 53 54 $D_HI $D_HI clk1 37 dff1
U45 DFF 55 56 $D_HI $D_HI clk1 39 dff1
```

```
U46 DFF 57 58 $D_HI $D_HI clk1 41 dff1
U47 DFF 59 60 $D_HI $D_HI clk1 b9 dff1
U48 DFF 61 62 $D_HI $D_HI clk1 b8 dff1
U49 ADD(1) BT5 63 i h 64 add1
U50 ADD(1) BT4 65 g f 63 add1
U51 ADD(1) BT3 66 e d 65 add1
U52 ADD(1) BT2 67 c b 66 add1
U53 ADD(1) MSB 68 $D_LO a 67 add1
E10 69 70 out7b cm 1e4
V5 70 0 1.6
U54 DFF 71 72 $D_HI $D_HI clk1 69 dff1
U55 DFF lsb 73 $D_HI $D_HI clk2 74 dff1
U56 ADD(1) BT6 64 k j 75 add1
U57 AND 74 ph1 71 and1
U58 CLOCK clk1 TPER=10n (0,1) (3n ,0) (8n,1)
U59 INV clk2 clk1 inv2
X6 out5b out5 cm refp refn VNEG VPOS bl0 bl1 ph2a ph2 ph1 out6 out6b SUB_MDAC
X7 out6b out6 cm refp refn VNEG VPOS b12 b13 ph1a ph1 ph2 out7 out7b SUB_MDAC
U60 DFF 76 77 $D_HI $D_HI clk2 43 dff1
U61 DFF 78 79 $D_HI $D_HI clk2 45 dff1
U62 DFF 80 81 $D_HI $D_HI clk2 47 dff1
U63 DFF 82 83 $D_HI $D_HI clk2 49 dff1
U64 DFF 84 85 $D_HI $D_HI clk2 51 dff1
U65 DFF 86 87 $D_HI $D_HI clk2 53 dff1
U66 DFF 88 89 $D_HI $D_HI clk2 55 dff1
U67 DFF 90 91 $D_HI $D_HI clk2 57 dff1
U68 DFF 92 93 $D_HI $D_HI clk2 59 dff1
U69 DFF 94 95 $D_HI $D_HI clk2 61 dff1
U70 DFF a 96 $D_HI $D_HI clk1 76 dff1
U71 DFF b 97 $D_HI $D_HI clk1 78 dff1
U72 DFF c 98 $D_HI $D_HI clk1 80 dff1
U73 DFF d 99 $D_HI $D_HI clk1 82 dff1
U74 DFF e 100 $D_HI $D_HI clk1 84 dff1
U75 DFF f 101 $D_HI $D_HI clk1 86 dff1
U76 DFF g102 $D_HI $D_HI clk1 88 dff1
U77 DFF h 103 $D_HI $D_HI clk1 90 dff1
U78 DFF i 104 $D_HI $D_HI clk1 92 dff1
U79 DFF j 105 $D_HI $D_HI clk1 94 dff1
U80 DFF 106 107 $D_HI $D_HI clk2 b11 dff1
U81 DFF 108 109 $D_HI $D_HI clk2 b10 dff1
U82 DFF k 110 $D_HI $D_HI clk1 106 dff1
U83 DFF l 111 $D_HI $D_HI clk1 108 dff1
U84 DFF m 112 $D_HI $D_HI clk1 b13 dff1
U85 DFF n 113 $D_HI $D_HI clk1 b12 dff1
U86 ADD(1) BT7 75 m 1 114 add1
U87 ADD(1) BT8 114 lsb n $D_LO add1
.END
```

SUBMDAC

```
.SUBCKT sub_mdac  inp inn cm refp refn VNEG VPOS bi bi+1 ph1a ph1  ph2 out outb
.model inv1 ugate (tphl=1p tplh=1p)
.model and1 ugate (tphl=1p tplh=1p)
.model dff1  ueff (tpclkhl=1p tpclklh=1p)
U18 AND(3) bi a bb ph2 and1
U19 AND x ab ph2 and1
U20 AND bi+1  b ph2 and1
S2 inp 1 ph1 0 SW
S3 inn 2 ph1 0 SW
S4 VNEG 1  x 0 SW
S5 VPOS 1 bi+1 0 SW
S6 VPOS 2 x  0  SW
S7 VNEG 2 bi+1  0  SW
S8 outb 3  ph2 0 SW
S9 out 4 ph2 0 SW
S10 5 cm ph1a 0 SW
S11 6 cm ph1a 0 SW
C1 1 6 1p
C2 2 5  1p
E2 7 cm 8 cm -0.5
C3 6 3  1p
C4 5 4  1p
S12 3 inp ph1  0  SW
S13 4 inn ph1  0  SW
E3 9 cm cm 8 -0.5
R1  10 8 10k
C5 8 cm  1p
E7  10 cm 6 5 2e3
S14 5 6  ph1a 0 SW
S15 2  1 bi 0 SW
E9 11  12 inp refn 50
V4 Vth 0 1.6
E10  13  14  inp  refp  50
E12 14  Vth refn inn 50
E13 12  Vth  refp inn 50
R2  7 outb 500
R3 9 out 500
U27 INV 15 ph1a inv1
U28 DFF a ab $D_HI $D_HI  15  11 dff1
U29 DFF b bb $D_HI $D_HI  15  13 dff1
.ENDS sub_mdac
```

APPENDIX B

SPICE OPTIONS

In his book Inside SPICE – overcoming the obstacles of circuit simulations, the author Ron Kielkowski states:

> ...One of the most important commands in the SPICE input file is the .OPTIONS statement....

...Understanding the .OPTIONS statement parameters is crucial to producing fast, accurate, convergent SPICE simulations.

The book also suggests a set of options values that the author found to be reliable for most circuits, including the converters of this book. The suggested options are:

.OPTIONS ACCT ITL1=500 ITL2=200 ITL4=40 ITL5=0 LIMPTS=10000

In some cases the options are different than suggested above due to the fact that some circuits are more complex. In those cases the user is encouraged to use the options of the individual files contained in the examples. An example of such a case is the serial 1.5 bit/stage pipeline A/D. In this case we are dealing with a switch cap application that does not include a DC path from many nodes to the ground. This is a particular difficult case for SPICE to deal with. For this particular simulation several ambiguous but necessary options were used:

LVLTIM is the parameter defining the time step control algorithm during the transient simulation. In our example we set LVLTIM =1 which selects the iteration-count time step control algorithm, (the Topspice default is 2 which sets for the local truncation error algorithm).

PIVREL is one of the parameters related to the numerical pivoting algorithm. This option defines the ratio between the largest entry in the conductance array of the circuit matrix. In our example PIVREL = 1E-6 (compared with the Topspice default of 1E-3).

TRTOL and CHGTOL are the least understood SPICE parameters. TRTOL was a parameter added in SPICE to adjust the step size. In our example we used TRTOL=1 and CHGTOL=1e-16 while the Topspice defaults are TRTOL= 7 and CHGTOL=1e-14.

All these options just mentioned were found by trial and error given the difficulty in achieving convergence.

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

Z