A 16-bit, 100 kS/s Self-Calibrating Cyclic

Analog-to-Digital Converter

by

Michael J. Guidry

Submitted to the Department of Electrical Engineering and Computer Science

in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering and Computer Science

at the Massachusetts Institute of Technology

January 31, 2000

Author____

Department of Electrical Engineering and Computer Science
January 31, 2000

Certified by____

Hae-Seung Lee
Thesis Supervisor

Accepted by____

Arthur C. Smith
Chairman, Department Committee on Graduate Theses

A 16-bit, 100 kS/s Self-Calibrating Cyclic Analog-to-Digital Converter

by

Michael J. Guidry

Submitted to the
Department of Electrical Engineering and Computer Science

January 31, 2000

In Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

Self-calibrating A/D architectures minimize the sometimes expensive process of in-factory calibration, thus allowing manufacturers to reduce chip cost. This thesis presents an A/D converter built using one such architecture, the reference-refreshing architecture. This converter shows under simulation 100 kHz performance at 16 bits, with INL of 1.2 LSBs, SNR of 89 dB, and power consumption of 170 mW.

Thesis Supervisor: Harry Lee
Title: Professor of Electrical Engineering

Table of Contents

List of Figures

CHAPTER 1

INTRODUCTION

Advances in silicon processing technology have dramatically reduced the cost of producing silicon integrated circuits. In high-performance analog circuitry, the cost of actually producing a circuit has become quite small, compared with the cost of trimming and testing of such a circuit before being sold to the consumer. To continue the trend of being able to produce higher performance circuits at lower cost, it is necessary to investigate circuit techniques which require less external modification and verification than those presently in use.

One category of high-performance analog circuits which is coming under the above constraints is the 16-bit analog-to-digital converter. The majority of circuits of this type are designed around an algorithm called the successive approximation algorithm. While this algorithm is quite effective, these circuits must be carefully calibrated and tested before being released, which is relatively expensive and eliminates much of the cost advantage made in reduced silicon cost. This research is concerned with the design of a 16-bit A/D using a different algorithm, the reference-refreshing cyclic algorithm. This algorithm uses a self-calibrating technique to ensure desired performance, and thus does not require the calibration and testing necessary of the successive approximation converter.

CHAPTER 2

BACKGROUND

This chapter gives some background information on analog-to-digital converters, and is divided into four sections. The first section discusses the very basics of analog-to-digital conversion. The second section discusses some of the parameters used to characterize an analog-to-digital converter. The third section discusses the division of analog-to-digital converters into roughly three categories, based on the trade-off between converter resolution and conversion rate. The final section is a survey of the most popular architectures used in the category of interest in this research, namely medium speed, medium accuracy converters.

2.1 Fundamentals of Analog-to-Digital Converters

An N-bit analog-to-digital converter (A/D) is a device that samples an analog input at some point in time, and then quantizes this sampled input into an N-bit digital word. An A/D can be unipolar, in which case its input range would be from zero volts to some voltage V_{fs} (called the full scale voltage), or it can be bipolar, in which case its input range would be between two voltages of equal magnitude and opposite sign $-V_{fs}/2$ and $V_{fs}/2$. In the unipolar case, the output bits b_0, b_1,...b_{N-1}, will be binarily weighted such that the relationship between the bits and the input voltage will be:

$$V_{IN} = \frac{V_{fs}}{2^N}(b_{N-1}2^{N-1} + b_{N-2}2^{N-2} + \ldots b_0 2^0) + \varepsilon$$

In the bipolar case, the relationship will be:

$$V = \frac{V_{fs}}{2^N}(b_{N-1}2^{N-1} + b_{N-2}2^{N-2} + \ldots b_0 2^0) - \frac{V_{fs}}{2} + \varepsilon$$

The quantity ε represents the difference between the actual input and the digital representation of the input. In an ideal A/D, this quantity will be less than the smallest resolvable input voltage (which has magnitude equal to $V_{fs}/2^N$). This smallest resolvable voltage is sometimes referred to as 1 LSB, since a change in the input by this voltage will cause a change in the output code by one least significant bit (LSB). This unit is used to describe very small voltages, such as those associated with DNL and INL (discussed in the next section).

2.2 A/D Performance Measures

Since no A/D operates perfectly, several measures of A/D performance have been developed. Some of these measures are: signal-to-noise ratio, differential nonlinearity, integral nonlinearity, gain and offset error, and total harmonic distortion. These measures are described below.

2.2.1 Signal-to-noise ratio

The ideal transfer function of an N-bit analog-to-digital converter is shown in figure 2-1. It shows that every output code corresponds to a range of input voltages. This means that given an output code, the voltage which produced this code can only be known to 1 part in 2^N; thus, even in an ideal N-bit A/D, information is lost in the conversion process. Since this error occurs when the analog voltage has been quantized to a digital code, the error is referred to as quantization error.

From a signal processing standpoint, this error can be modeled as a noise source. For simplification, it is assumed that for every digital code, the possible analog value producing that code is a uniformly distributed random variable throughout the corresponding input range. If the input range is equal to Δ, then the variance of this random variable will be equal to $\Delta^2/12$. If it is then

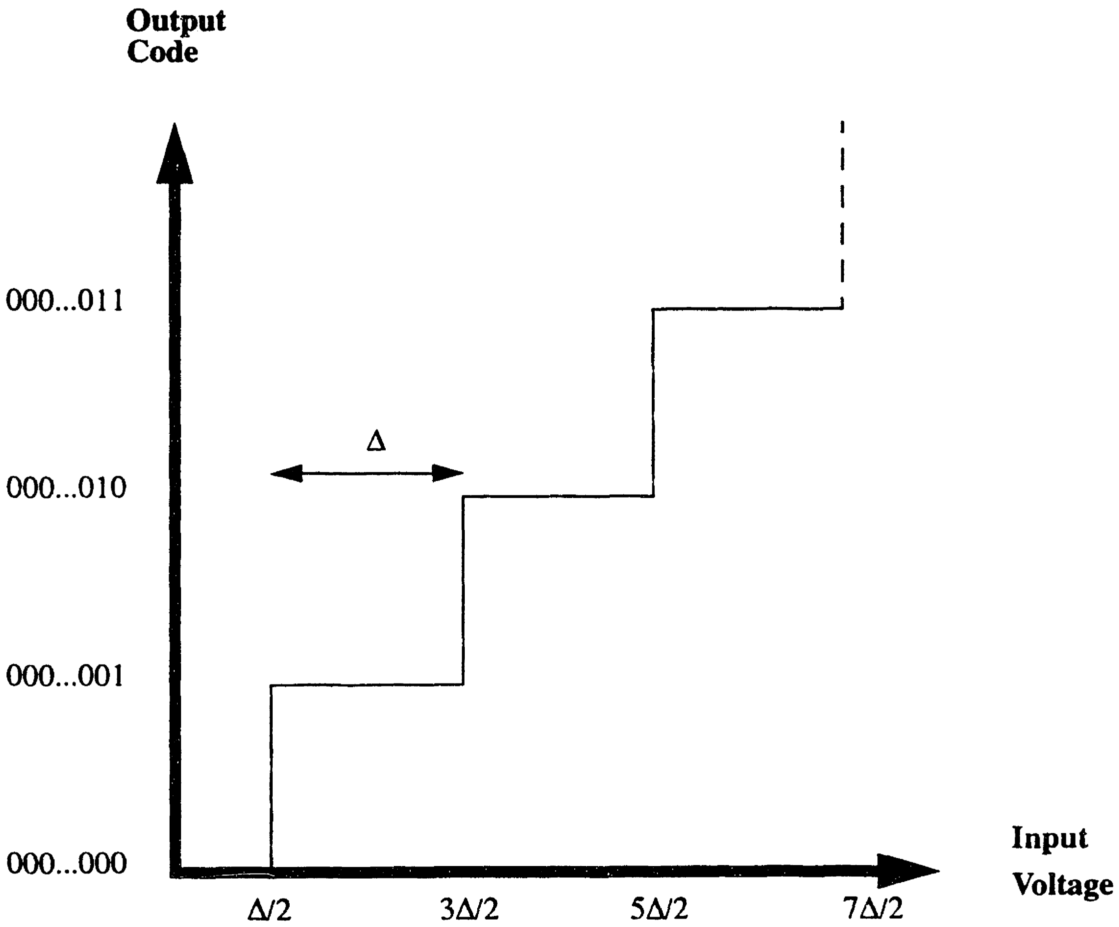


**Figure 2-1:
Ideal N-bit A/D Transfer Function**

$$\Delta = \frac{\text{input range}}{2^N}$$

assumed that any sample is independent from any other sample, the quantization errors will form a white-noise sequence. This sequence will have a noise power equal to the variance $\Delta^2/12$.

Signal-to-noise ratio (SNR) is defined to be

$$SNR = 10\log\left(\frac{\text{input power}}{\text{noise power}}\right)$$

For an ideal N-bit converter, the SNR for a full-scale input will be

$$SNR = 10\log\left(\frac{\frac{{V_{fs}}^2}{8}}{\frac{\Delta^2}{12}}\right)$$

$$= 10\log\left(\frac{\frac{V_{fs}^2}{8}}{\frac{V_{fs}^2}{12 \cdot 2^{2N}}}\right)$$

$$= 10\log(1.5 \cdot 2^{2N})$$

which simplifies to SNR = 6.02N + 1.8.

In the case of an actual A/D, there will be other noise sources due to nonidealities in such components as the comparator, sample-and-hold, switches, etc. In this case, the amount of noise in the converter can be measured at the output of the converter. Since any noise source is indistinguishable from any other at the output, the total noise can be treated as a quantization noise. This allows the above equation to be used to determine the effective resolution of the converter (the actual resolution if the total noise really were just a quantization noise). The effective number of bits (ENOB) of a converter is defined to be

$$ENOB = \frac{SNR - 1.8}{6.02}$$

which is just the inverse of equation above. For instance, a nominal 16-bit A/D which has an SNR of 83 dB for a full-scale input will have an ENOB of 13.5 bits.

2.2.2 Differential Nonlinearity and Missing Codes

In the ideal A/D, the size of the input range corresponding to an output code is the same for every output code. Any deviation from this in an actual A/D is referred to as differential nonlinearity (DNL). DNL is usually measured in LSBs from the ideal range. Whereas DNL exists for each output code, in practice the term DNL is used to refer to the worse-case value of DNL for an A/D.

In the case of very large DNL, it is possible that there will be no input range corresponding to a particular output code. Since this code can never be produced for any input, it is called a missing code. Missing codes can cause problems for any converter used in a feedback loop, so it is important that they be avoided.

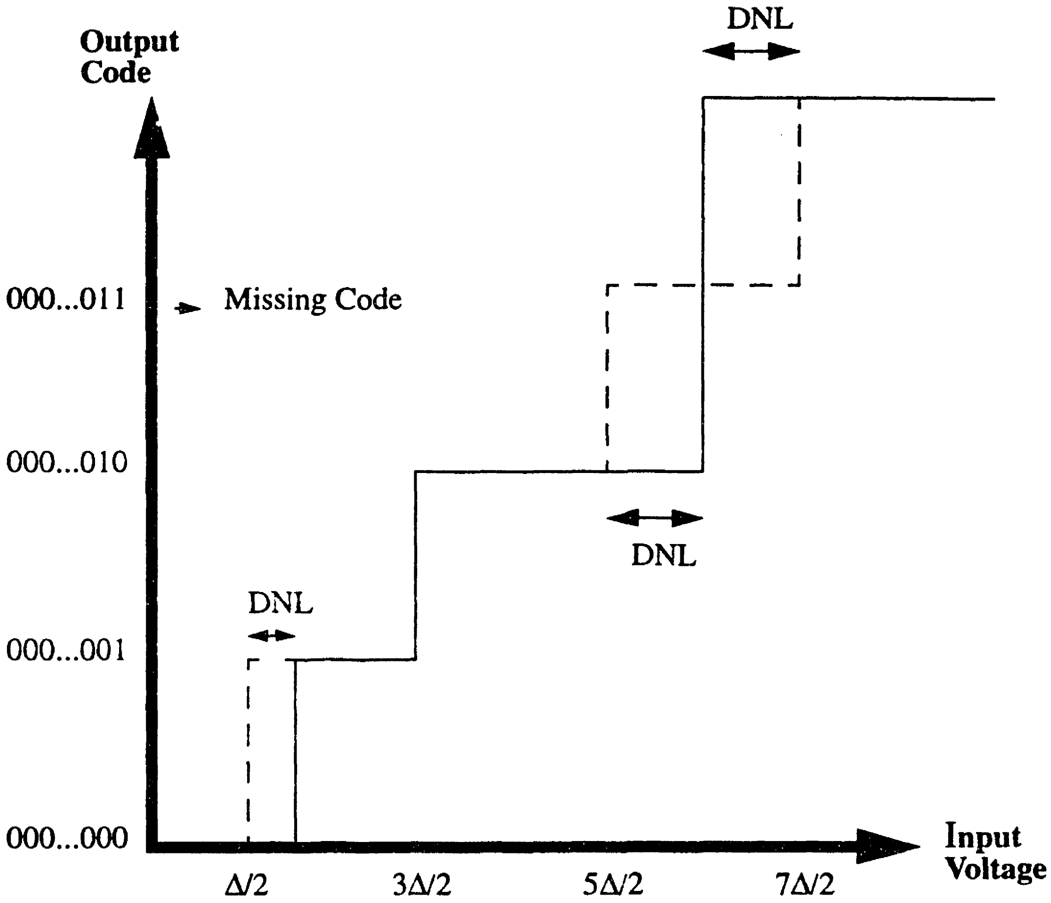


Figure 2-2: DNL and Missing Codes in an A/D
Dashed line is ideal transfer curve
Solid line is actual transfer curve

2.2.3 Integral Nonlinearity

If one were to draw a straight line connecting the endpoints of the transfer curve of an ideal A/D, the transfer curve would intersect the line at every transition of the input, halfway between output codes (see figure 2-3). Any deviation from straight-line behavior in an actual A/D is called integral nonlinearity (INL). As with DNL, INL occurs for every possible digital code, but the term INL is often used only to describe the worse case value of INL.

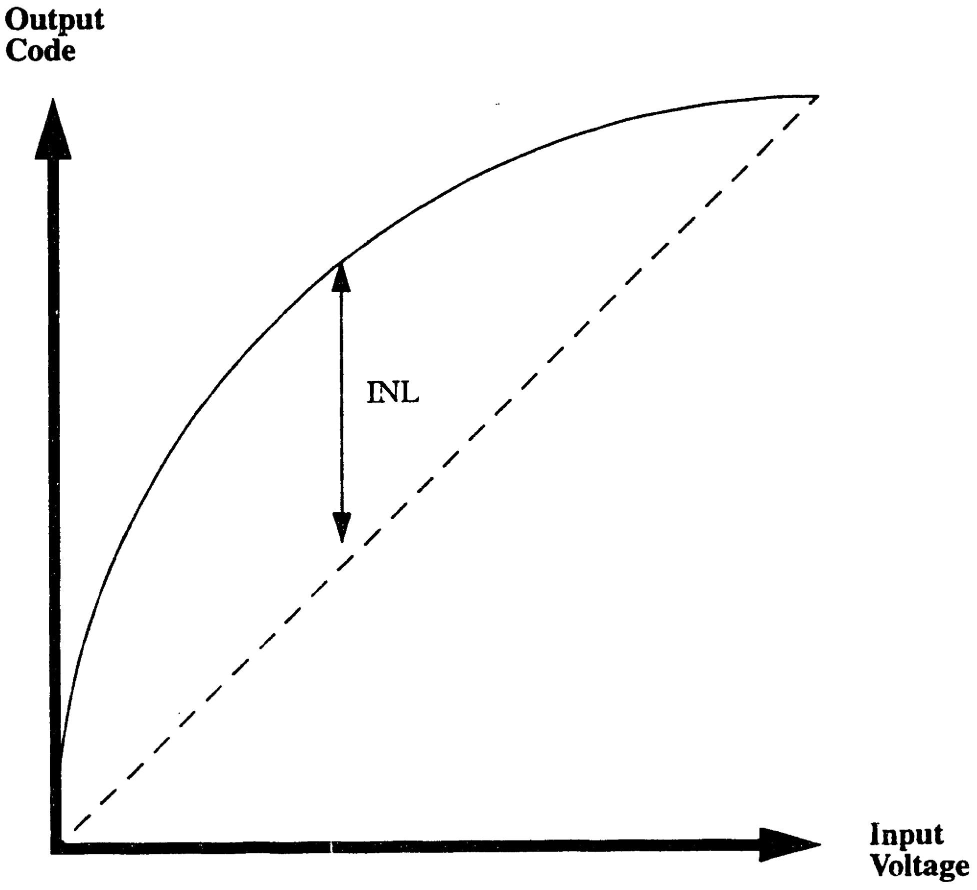


Figure 2-3: INL in an A/D
Dashed line is ideal transfer curve
Solid line is actual transfer curve

(scale is max zoom out, so staircase shape not shown)

2.2.4 Gain error & offset

Offset in an A/D is present when a zero input does not correspond to an all-zero output code. This error is usually easy to correct for, by subtracting the code which results from a zero input from all other codes. Gain error occurs when the straight-line approximation as described in the previous section does not have the ideal slope.

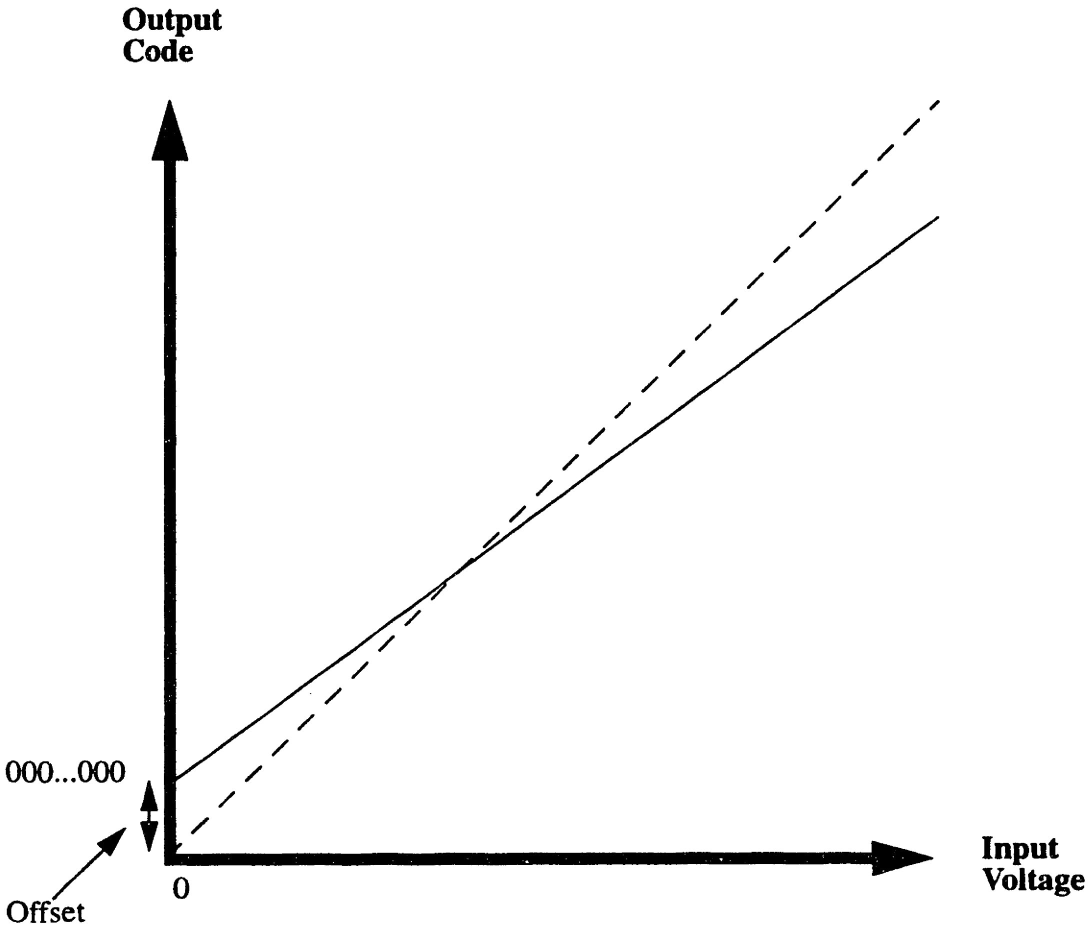


Figure 2-4

Offset and Gain Error in an A/D
dashed line is ideal transfer curve
solid line is actual transfer curve

2.2.5 Total Harmonic Distortion

In section 2.2.1, it was assumed that the error resulting from one conversion was independent of the error in any other conversion. If the input to the converter is a waveform of some sort, then this is not necessarily the case. A special problem occurs when the waveform happens to be a sinusoidal waveform. In this case, one would like the output to only contain frequency components at the frequency of the input sinusoid. If there are any nonlinearities in the converter, this will cause frequency components to be placed at harmonics of the input frequency. This effect is called harmonic distortion. Harmonic distortion is sometimes treated as a noise source, so that it can be grouped with the uncorrelated noise in the circuit, with the resulting performance measure becoming signal-to-(noise plus distortion) ratio.

Figure 2-5 shows an FFT plot of a converter's output when there is harmonic distortion. Whereas only one frequency component is desired, there are other components at multiples of the fundamental.

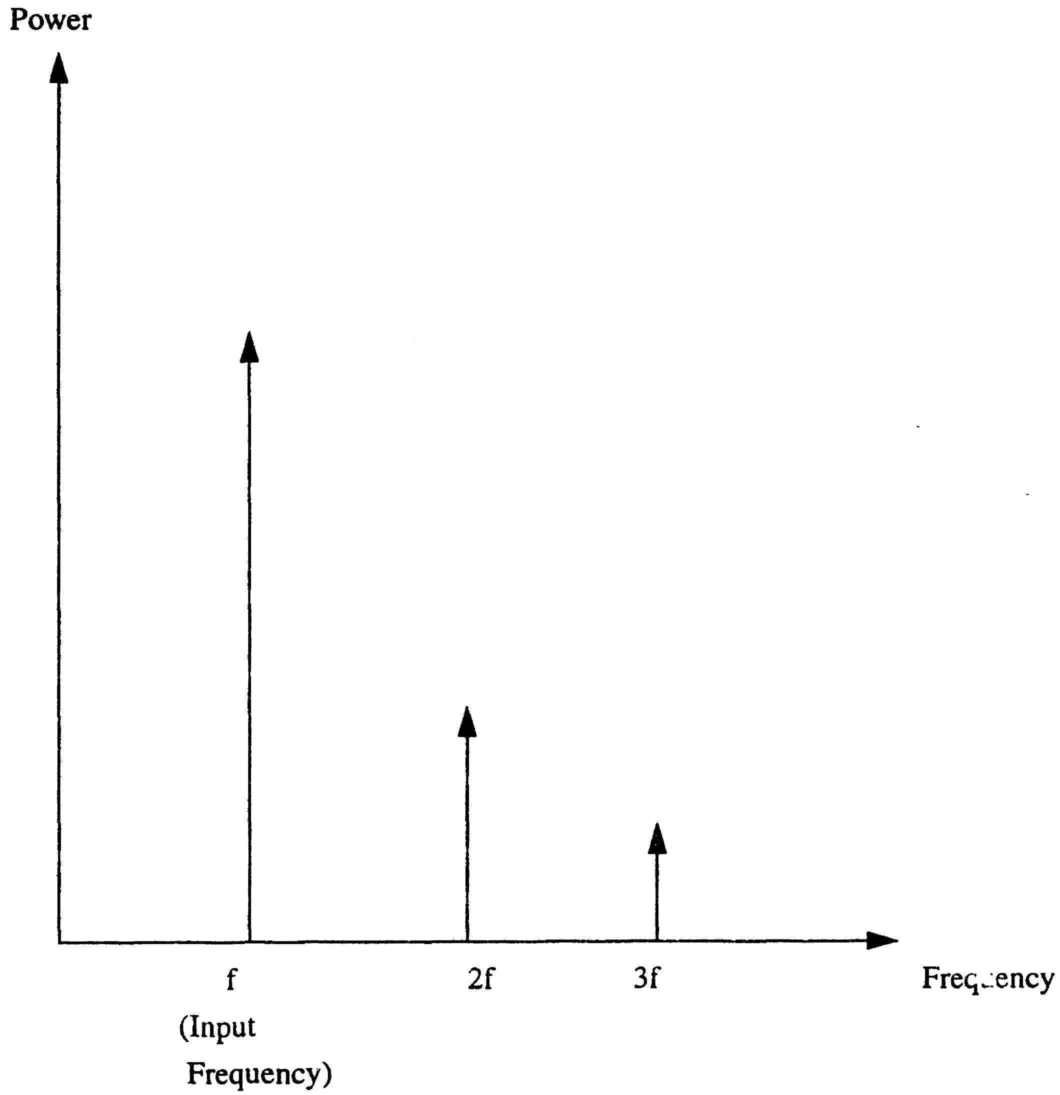


Figure 2-5
Total Harmonic Distortion

Shown is the example when the input to an A/D is a pure sinusoid of frequency f. Harmonic distortion creates energy at frequencies that are multiples of the input frequency.

2.3 TRADEOFF BETWEEN SPEED AND RESOLUTION

In any practical application of an A/D, there will be a trade-off between converter resolution and conversion speed. This trade-off has led to the division of A/D's into roughly three categories: high speed, low resolution converters; low speed, high resolution converters; and medium speed, medium resolution converters. Each category has architectures which are most practical for that category:

Table 1: Analog-to-Digital Converter Architectures

Category	Architecture
High Speed, Low Resolution	Flash Pipelined
Low Speed, High Resolution	Integrating, Σ–Δ
Medium Speed, Medium Resolution	Successive Approximation Cyclic

This research is concerned with the final category, medium speed, medium resolution converters. There are generally two types of converters used for this application, discussed in the next section.

2.4 MEDIUM SPEED, MEDIUM RESOLUTION A/Ds

Two types of A/Ds used as medium speed, medium resolution converters are: successive approximation and cyclic converters.

2.4.1 Successive Approximation Converter

The successive approximation converter (figure 2-6) is presently the most popular architecture used in the category of interest. The successive approximation converter algorithm works as follows: first, the successive approximation register (SAR) "guesses" the end result by setting the most significant bit (MSB) to "1" and leaving all less significant bits "0". This guess is then used as the input to a digital-to-analog converter (DAC). The output of the DAC is connected to the negative input of a comparator. The positive input of the comparator is connected to the input voltage. This causes the output of the comparator to be the result of the comparison of the input voltage and the analog version of the "guessed" code. If the output of the comparator is a "1"

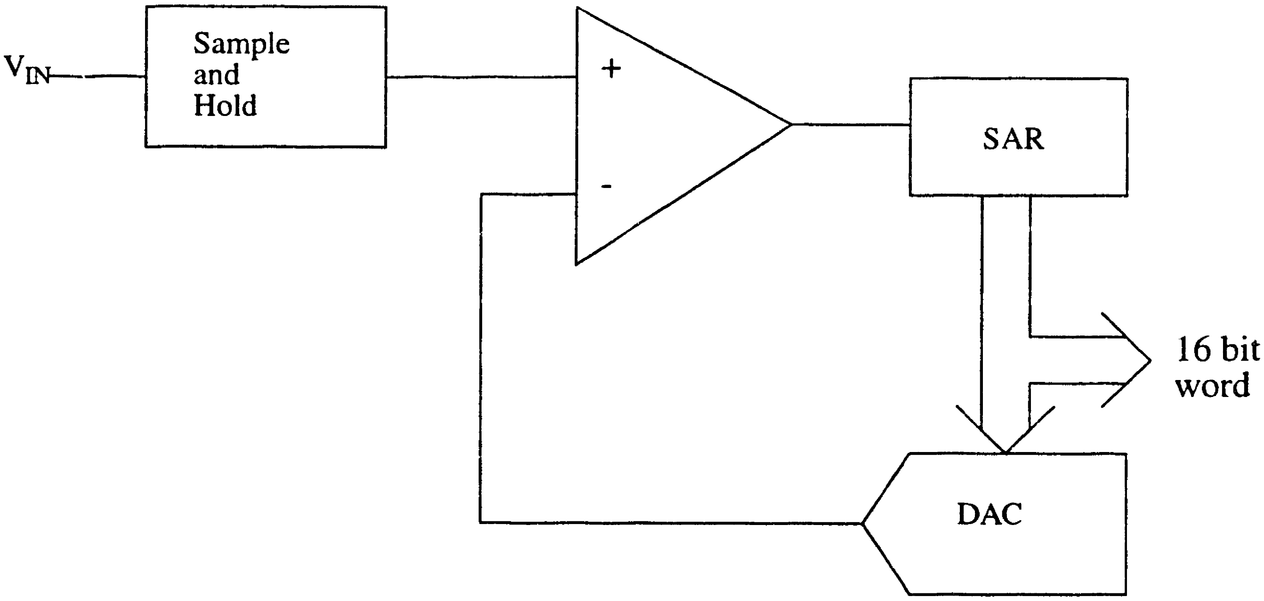


Figure 2-6: Successive Approximation A/D Converter (from Johns and Martin, p.493)

(meaning the "guessed" voltage is less than the input voltage), the MSB is kept at “1”. If the output of the comparator is "0", the MSB is then changed to a "0". Next, the SAR will make a new "guess" by changing the next most significant bit to a "1", and the algorithm repeats itself until all bits have been determined.

This algorithm has several aspects which make it attractive for use in medium speed, medium resolution converters. First of all, it requires a modest amount of circuitry; thus, it does not require much die area. This is in contrast to ultrafast flash converters which require 2^N comparators to perform an N-bit conversion. Secondly, this algorithm only requires N clock cycles for an N-bit conversion, unlike many ultra high resolution converters which require 2^N clock cycles to perform a conversion. Finally, many calibration techniques have been developed which allow practical implementation of SAR converters at 16-bit resolution.

As SAR converters have increased in accuracy, they have also increased in cost. This is due to two reasons. The first reason is that most of the calibration techniques which allow higher performance must be physically performed on the chip before it is packaged. This requires special machines and manpower to be used during fabrication, and so adds to the cost of fabrication. The second reason is that to guarantee performance, manufacturers must test the chip before it can be delivered to customers. This also requires machinery and manpower, and will also add to the cost of fabrication. These two costs are now a significant component in the overall cost of the chip.

In order to reduce the overall cost of a converter, it would be necessary to reduce the cost incurred in test and calibration. It would be desirable to implement an architecture which may require less testing and calibration than the SAR converter.

2.4.2 Cyclic Converter

One architecture which can achieve similar performance to the SAR converter is the cyclic converter (figure 2-7). The cyclic converter works as follows: first, the MSB decision is made by comparing the input voltage to the middle value of the input range. If the input is higher than this midscale value, the MSB will be a "1", else the decision is a "0". Next, the converter produces the analog voltage equal to two times the input voltage. From this voltage, a reference voltage is then either added (in the case the comparator decision was a "0") or subtracted (in the case the comparator decision was a "1"). The resulting voltage is called the "residue". At the end of a cycle, when the residue voltage has been computed, the residue is then sampled on the input stage of the converter, thus becoming the input voltage for the next cycle of the conversion. This cyclic behavior continues until all bit decisions have been made.

This architecture has many of the same advantages of a SAR converter. This converter can achieve reasonably high speed with relatively few components, and to increase resolution one only need run the converter for extra clock cycles.

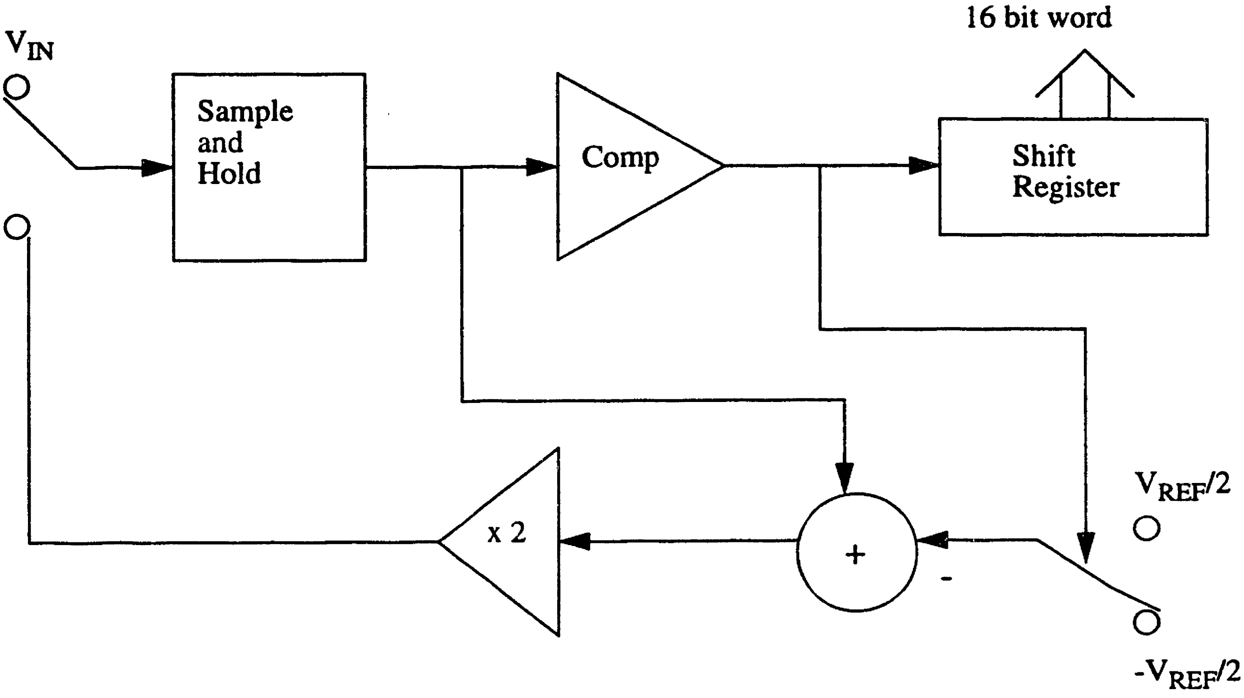


Figure 2-7: Cyclic Converter (from Johns and Martin, p.505)

The next chapter describes a method of self-calibration for cyclic converters called the reference-refreshing technique. Since it is a self-calibration technique, it provides a cheaper way of calibration over external calibration.

CHAPTER 3

REFERENCE-REFRESHING CYCLIC A/D

One method for self-calibrating a cyclic A/D is the reference-refreshing method. This method, and the circuit which implements it, is described in this chapter.

3.1 Reference-refreshing technique

The most difficult feature to implement in the cyclic converter is the precise multiply-by-two computation. There have been many proposed ratio-independent techniques which make it easier to implement this computation. One of these techniques is called the reference-refreshing technique. In a normal cyclic converter, the residue is always computed using a fixed reference. If it is assumed there is an error ε in the multiply-by-two computation, such that the residue is actually multiplied by $K = 2(1+\varepsilon)$, the residue voltages will evolve like this: (B_N being the bit decision made on the nth cycle):

$$\begin{aligned} V_{residue} &= K\{\ldots K[K(V_{in} - B_1 V_{ref}) - B_2 V_{ref}] + \ldots - B_N V_{ref}\} \\ &= K^{N-1}\left[V_{in} - B_1 V_{ref} - B_2 \frac{V_{ref}}{K} - B_3 \frac{V_{ref}}{K^2} - \ldots - B_N \frac{V_{ref}}{K^{N-1}}\right] \end{aligned}$$

From this it can be seen that the relative weights of the bits will be K and not 2. This results in an imprecise conversion (see figure 3-1).

In the reference-refreshing technique, the cyclic loop is modified such that the loop stores not only the residue, but also the reference voltage. In the cyclic process, the reference will go through the same circuitry as the residue voltage; thus, the same error in the multiply-by-two computation will be introduced into the reference voltage. The reference voltage of the nth cycle can be written as $V_{ref}^{[n]} = V_{ref}(1+\varepsilon)^n$, and the residue becomes:

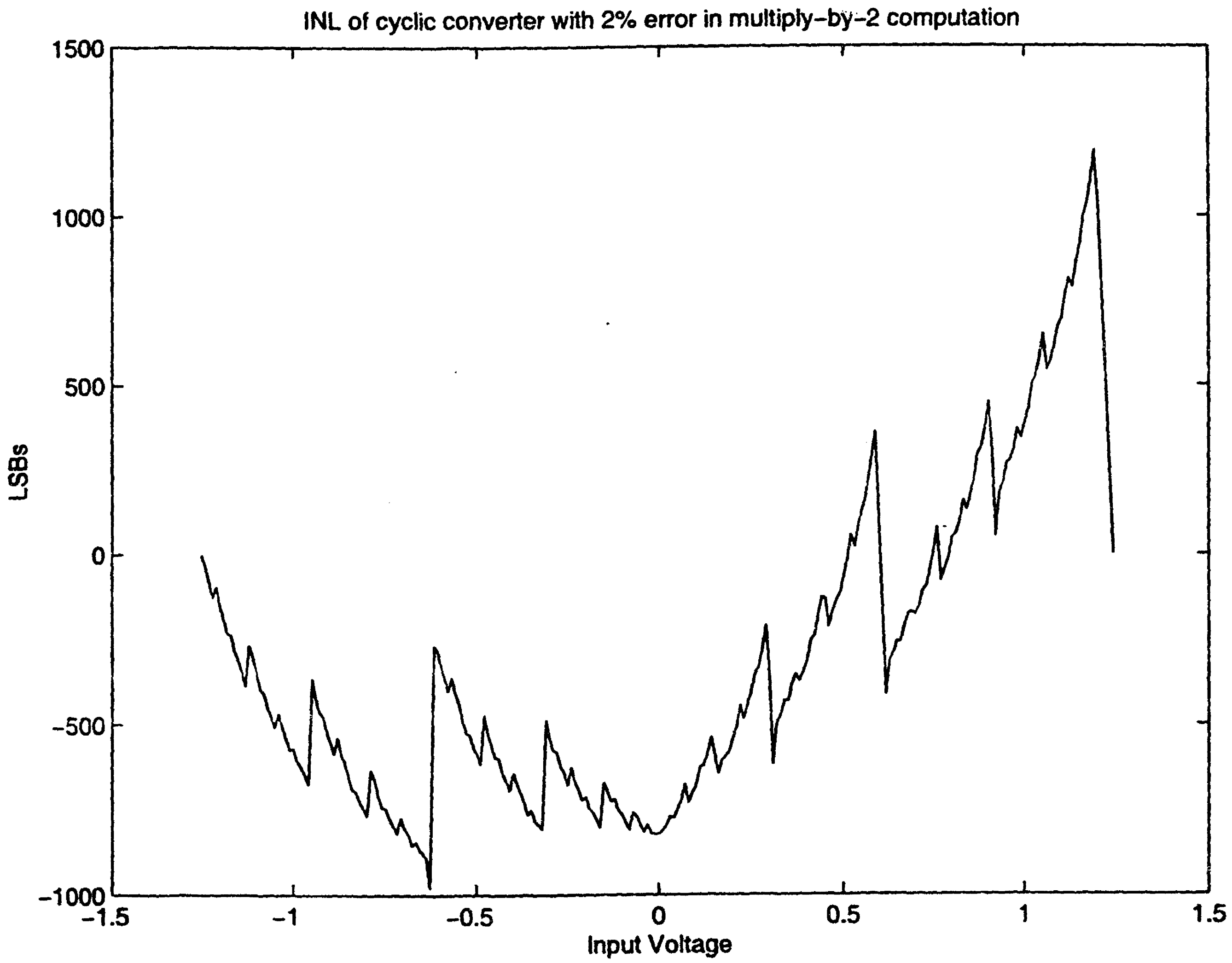


Figure 3-1: A/D Transfer function with error in multiply-by-2 computation

$$\begin{aligned} V_{residue} &= K\{K[K(V_{in} - B_1 V_{ref}^{[1]}) - B_2 V_{ref}^{[2]}] - B_2 V_{ref}^{[2]}\} \\ &= K^N\left[V_{in} - B_1 V_{ref}^{[1]} - B_1 \frac{V_{ref}^{[1]}}{K} - B_2 \frac{V_{ref}^{[2]}}{K^2} - \ldots - B_N \frac{{V_{ref}}^{[N]}}{K^{N-1}}\right] \\ &= K^N\left[V_{in} - B_1 V_{ref} - B_2 \frac{V_{ref}}{2} - B_3 \frac{V_{ref}}{2^2} - \ldots - B_N \frac{V_{ref}}{2^{N-1}}\right] \end{aligned}$$

The bit ratio is now two, and a precise conversion will result. It is important to note that this error cancellation depends on the error itself being a linear error, which it will never quite be. Therefore, it is important in the design of this converter that elements be designed such that the errors introduced will only be linear errors.

3.2 Reference refreshing cyclic A/D

The schematic of the reference-refreshing cyclic converter is shown in figure 3-2. It is composed of a comparator, two amplifiers, offset capacitors CO1, CO2, CO3, and sample/hold capacitors C1, C2, C3, C4 and C5.

3.3 Operation

The operation of the reference-refreshing converter is as follows:

Phase 1: The offset cancellation capacitors CO1, CO2, CO3 store the offset voltages of the amplifiers and comparator.

Phase 2: The input voltage is sampled on the input capacitor C3, and the reference is sampled on to C1.

Phase 3: The input voltage is transferred to capacitor C4, and the reference is transferred to C2. Also, on this phase the first bit decision is made.

Phase 4: The input is now sampled on to capacitor C1, and the reference is sampled on to C3.

Phase 5: The input is transferred to capacitor C2, and the reference is sampled on to C4.

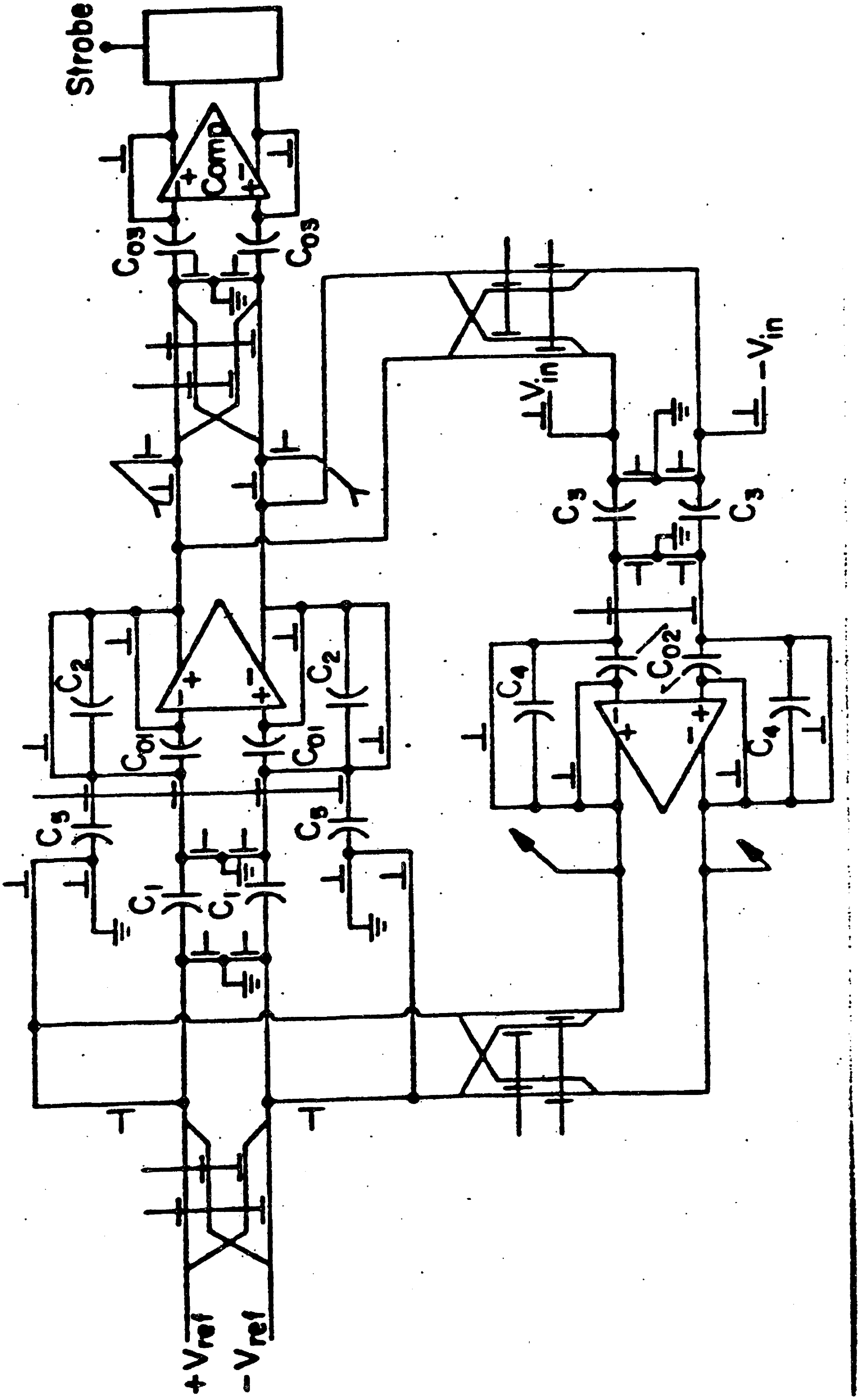


Figure 3-2: Reference-Refreshing Converter (from Shih, page 52)

Phase 6: If the first bit decision was a "1", the reference must be subtracted from the input voltage to determine the residue. This is done by sampling the reference on to C5 while connecting C5 to C2. Whether or not this is done, the next bit decision is now made.

Phase 7: The residue on C2 is now sampled on to C3, and the reference is sampled on to C1.

Phase 8: The residue on C3 is transferred to C4, and C3 is sampled with the opposite polarity of the residue. Doing this while C3 is connected to C4 has the effect of adding the residue to itself on C4, and thus doubling the residue.

Phase 9: The reference is transferred on to C2.

After this, phases 4 through 9 repeat until all bit decisions have been made. Each cycle, both the residue and reference pass through the loop composed of C1, C2, C3, and C4. This has the effect, described in the first section of this chapter, of cancelling out errors to the first order.

3.4 Advantages

The reference-refreshing converter is essentially a self-calibrating converter. By cycling the signal and the reference through the same loop, the same (linear) errors will be introduced into both signal and reference, and as discussed before, it will cause the output bits to be weighted by precisely two. This is without any external calibration, thus eliminating a potentially costly process.

3.5 Design Considerations

There are a few special considerations in the design of a reference-refreshing A/D. These considerations include noise, linearity, and speed.

3.5.1 Noise

At the 16-bit level, noise becomes a very important design constraint. For a full-scale range of 5 V, 1 LSB has a magnitude less than 80 uV. It is important then to keep all noise sources well below this level, so that they do not corrupt the signal. For the amplifiers and comparator, this is a straightforward constraint: the input-referred noise of these devices should be kept well below 1 LSB.

Noise also plays a part in choosing the size of the capacitors. A circuit composed of a resistor R and capacitor C will have due to thermal noise from the resistor a noise variance equal to kT/C, regardless of the size of the resistor. In this converter, which is a sampled-capacitor network, this sampling noise will be introduced several times even in one cycle. Shih [5] computed that the total noise variance of one cycle will be equal to $4\sqrt{2}\sqrt{(kT)/C}$. Since the residue is multiplied by two each cycle before going to the next cycle, the input-referred noise variance is reduced by a factor of $4^{(n-1)}$ on the nth cycle. The resulting input-referred noise deviation for a 16-bit converter will therefore be $6.6\sqrt{(kT)/C}$.

This noise gain of the converter is a large disadvantage. It requires capacitors much larger than necessary in other 16-bit converter. For instance, in the case of a 5 V input range, 1 LSB for 16-bits is 76 uV. To have input-referred noise less than 1/4 LSB, as is often used in other converters, requires 500 pF capacitors. This is quite large, and would be impractical to implement. This circuit uses 100 pF capacitors, which would have an input-referred noise of 42.6 uV, or a little over 1/2 LSB.

3.5.2 Linearity

As discussed in part I, the reference-refreshing converter works well in eliminating first-order errors. It will also cancel some high order error, as discussed in [5], but not as well as the first order error. It is therefore important to design the components of this converter such that the errors introduced are mostly linear errors. In the case of capacitor mismatch, the errors will only be linear; however, the amplifiers are nonlinear, and can introduce unwanted error components. Therefore, the amplifiers must be designed to keep this constraint in mind.

3.5.3 Speed

Having large capacitors requires large Gm amplifiers to drive them, as the unity-gain bandwidth of an amplifier is equal to Gm/C. It is also necessary to have a high slew rate, in the case that the amplifier is driven out of its linear region in transient conditions. These requirements are achieved through large input devices, and high bias currents.

The next chapter discusses the implementations of the amplifiers and comparator.

CHAPTER 4

IMPLEMENTATION

This chapter describes the design of the components so that the circuit satisfies the design goals. The circuits were simulated with Analog Devices' proprietary ADICE program.

4.1 Amplifiers

As discussed in the previous chapter, there were several considerations to be made when designing the amplifiers in this circuit. While actual gain was not an issue, as gain error is calibrated out, nonlinearities are a concern, as the converter will not cancel these out perfectly. The amplifiers need to be fast to drive the large capacitors and still settle quickly. Since the amplifiers are used in unity-gain configurations, they need to be unity-gain stable and have enough phase margin.

The amplifier topology used is shown in figure 4-1. It is a single-stage, folded cascode amplifier. The input and output branches are designed to run at 2.5 mA each. Biasing the input stage at high current makes it easier to meet the high-g_m spec with reasonable size devices. Biasing the output stage at high current is necessary to meet the slew-rate requirement. The cascode transistors are connected with level-shifted common-source amplifiers to provide output resistance enhancement, as described in [6]. This topology allows for a wide output swing, and increases the DC open loop gain of the amplifier. While the gain itself is not important, having high open-loop gain has the advantage of diminishing the effects of nonlinearities when the amplifier is connected in a unity-gain configuration. Any nonlinearities in the open-loop gain will be reduced by a factor of the DC open-loop gain when the amplifier is connected in a unity-gain configuration.

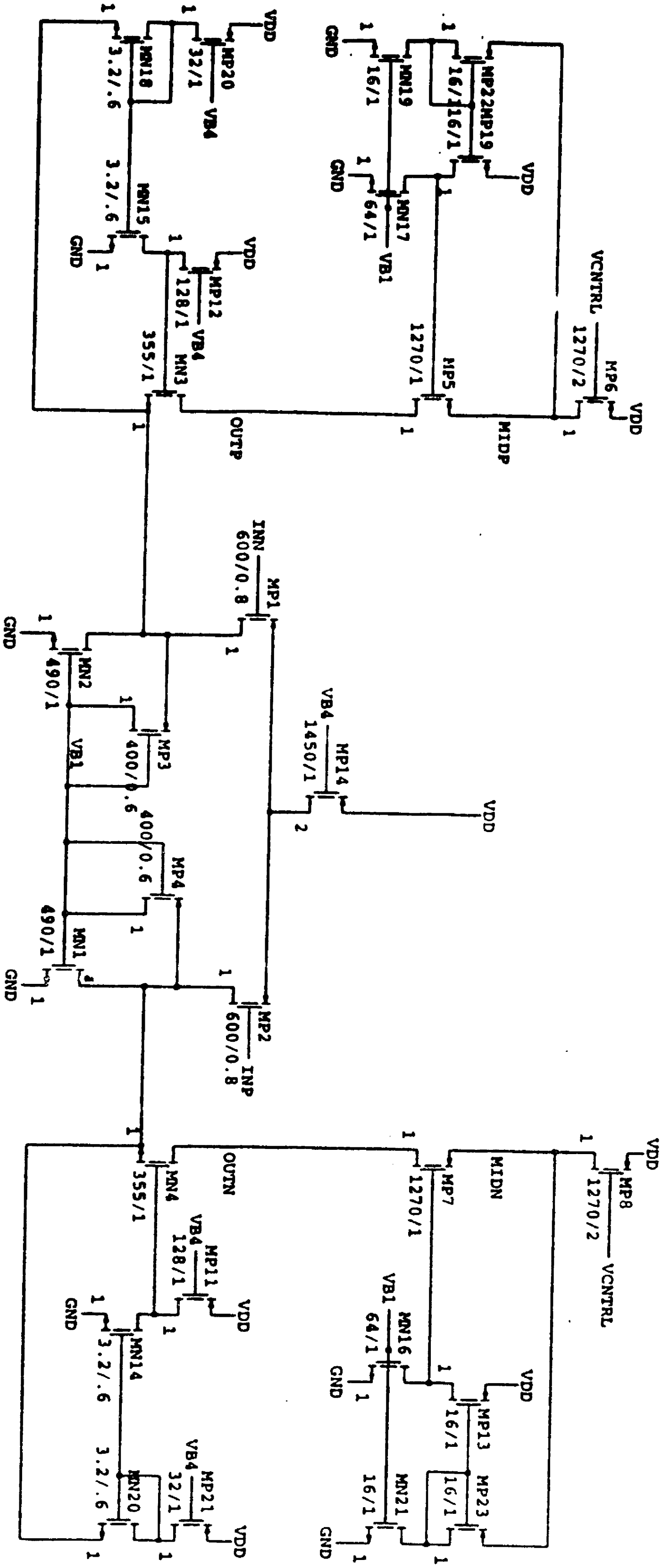


Figure 4-1: Amplifier Gain Circuit

The common-mode feedback circuit is shown in figure 4-2. There are two differential pairs MNC1-2 and MNC3-4. When the output common-mode is equal to the desired value, the current flowing through the current mirror MPC1-4 will be balanced, and the variable current sources MPC5-6 will maintain their state. If there is any imbalance (output common-mode not equal to the desired value), the differential pair will act as a g_m source, sourcing or sinking current into the gates of the variable current sources, which in turn drive the output branch of the gain circuit until the output common-mode reaches the appropriate value. It is important to note that stability is a concern in this type of circuit, as there are two high-impedance nodes in the common-mode loop, the output nodes of the gain circuit and the input node of the variable current sources. The current sources were kept small, and the output resistance of the g_m stage was also kept small, to push the equivalent pole to higher frequencies, and to ensure common-mode loop stability.

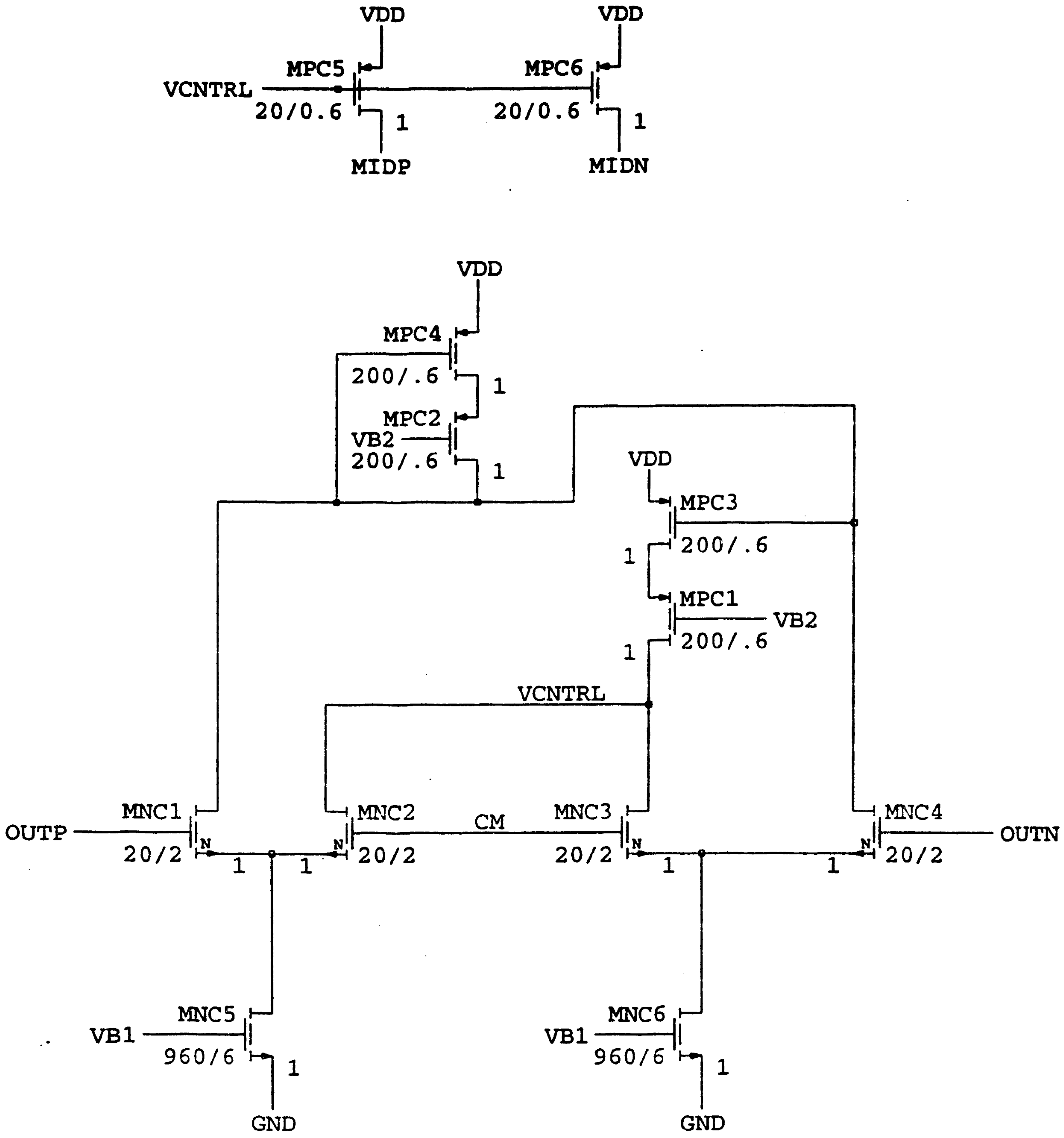


Figure 4-2: Amplifier Common Mode Feedback Circuit

The biasing circuit is shown in figure 4-3. This circuit uses complementary current mirrors to create a positive feedback loop, which settles to desired bias voltages for Vb1-4. This circuit is stable as long as the aspect ratio of MNB1 is smaller than that of MNB2.

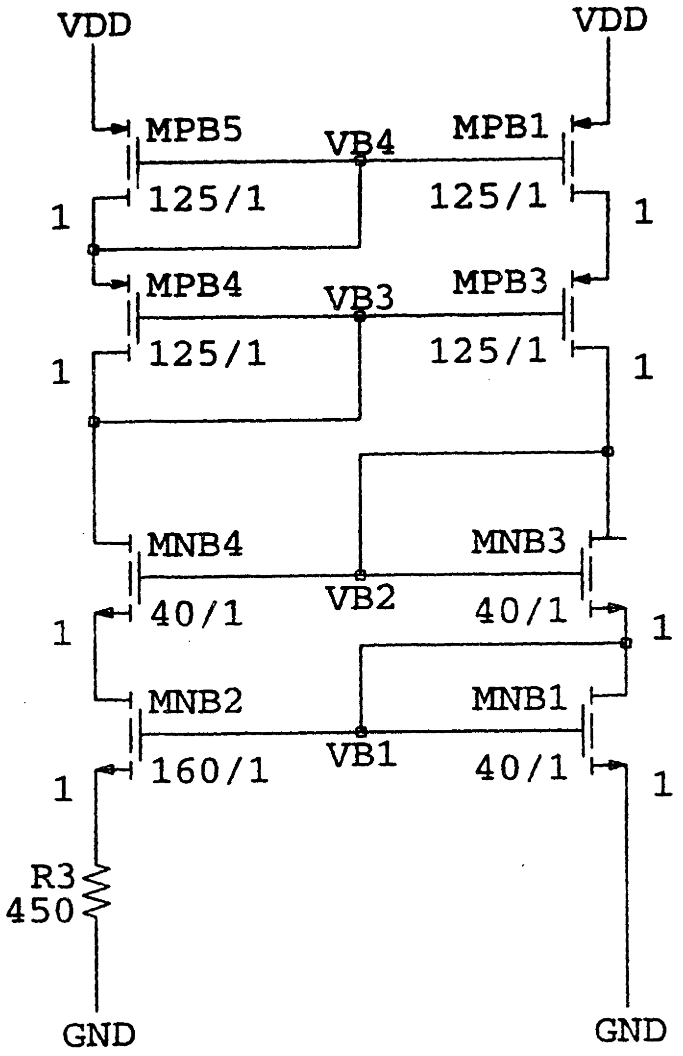


Figure 4-3: Amplifier Biasing Circuit

Figures 4-4 and 4-5 show the simulated response of the amplifier. Figure 4-4 shows the frequency response of the amplifier, whereas figure 4-5 shows the differential- and common-mode step responses.

Table 2: Simulated Amplifier Characteristics

Open-loop gain	14,000
Phase margin	52 degrees
Slew rate	83 V/us
16 bit settling time	177 ns
Power	55 mW

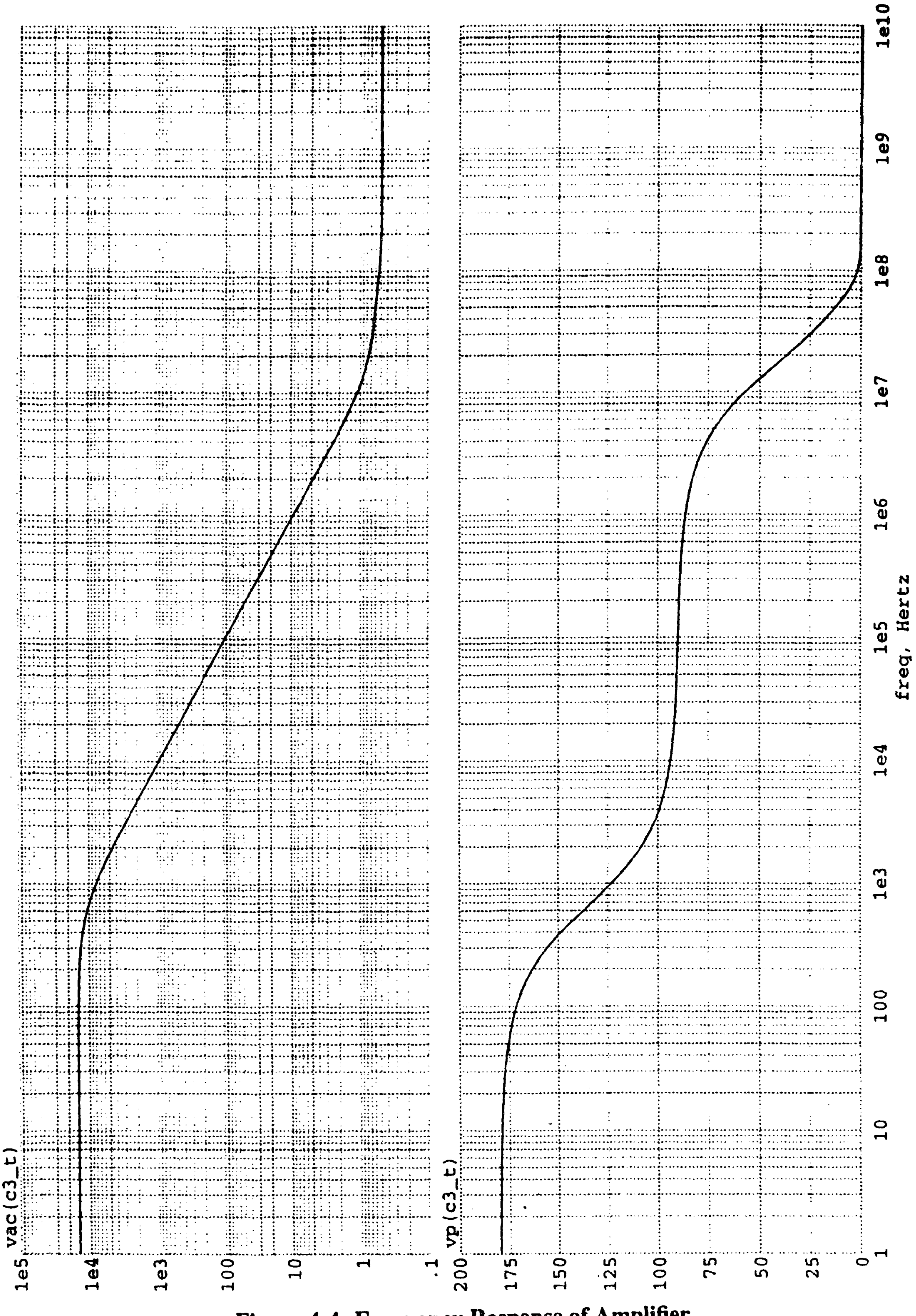


Figure 4-4: Frequency Response of Amplifier

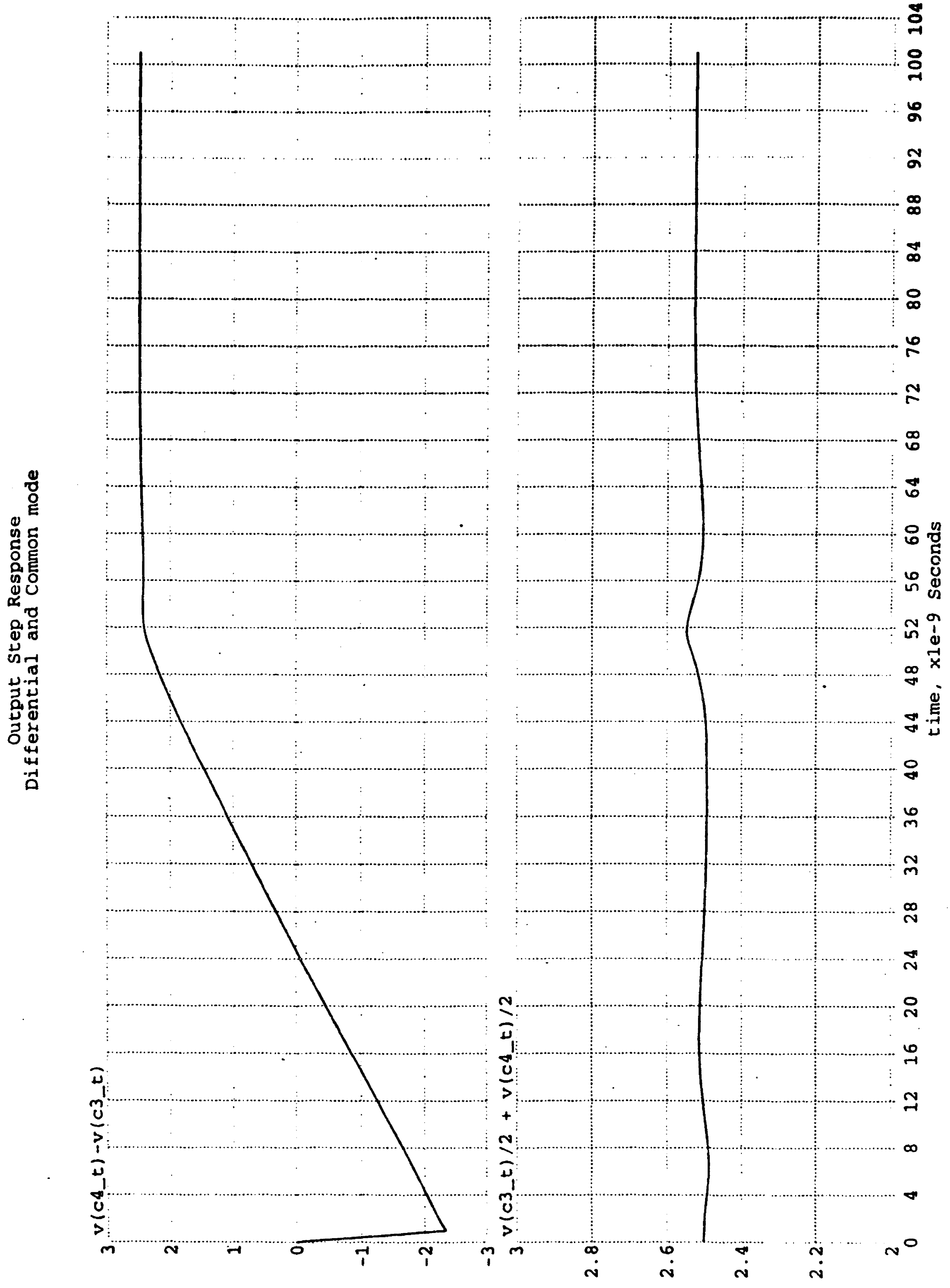


Figure 4-5: Step Response of Amplifier

4.2 Comparator

The comparator is a very important part of the converter design, as it can contribute significant noise to the transfer curve of the converter. The comparator's offset is also important, as it can create a dead region in the transfer curve if it is too large.

The comparator schematic used in this converter is shown in figure 4-6. It consists of an amplifier stage followed by a latch. The amplifier uses an identical topology to the other amplifiers in the circuit. The latch is a track-and-latch circuit [7]. When the LATCH signal is held low, the internal nodes of the circuit are at either VDD or GND. When the LATCH signal goes high, the latch becomes a positive feedback circuit, and the output of the circuit will quickly go to a '1' or '0' depending on the output of the amplifier.

The comparator noise can have a large effect on the overall SNR of the converter. For instance, if the comparator noise is equal to the sampling noise (42.6 uV), the SNR will be reduced by 3 dB. The noise introduced by a comparator depends on many things, including process factors. Since the circuit was not actually fabricated, noise from the comparator was not used in the final SNR calculation.

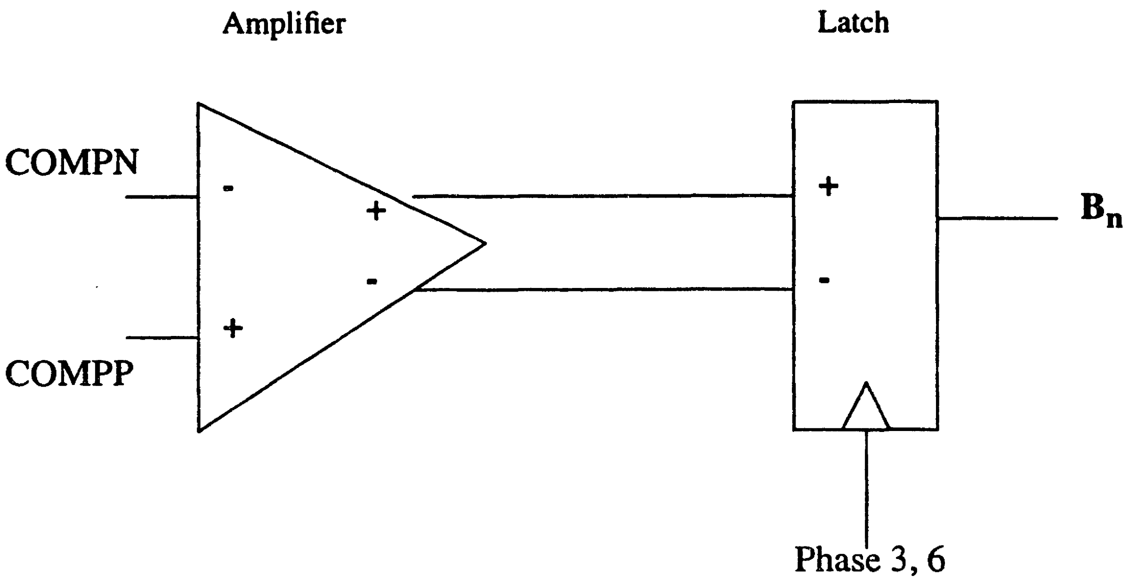


Figure 4-6: Comparator Schematic

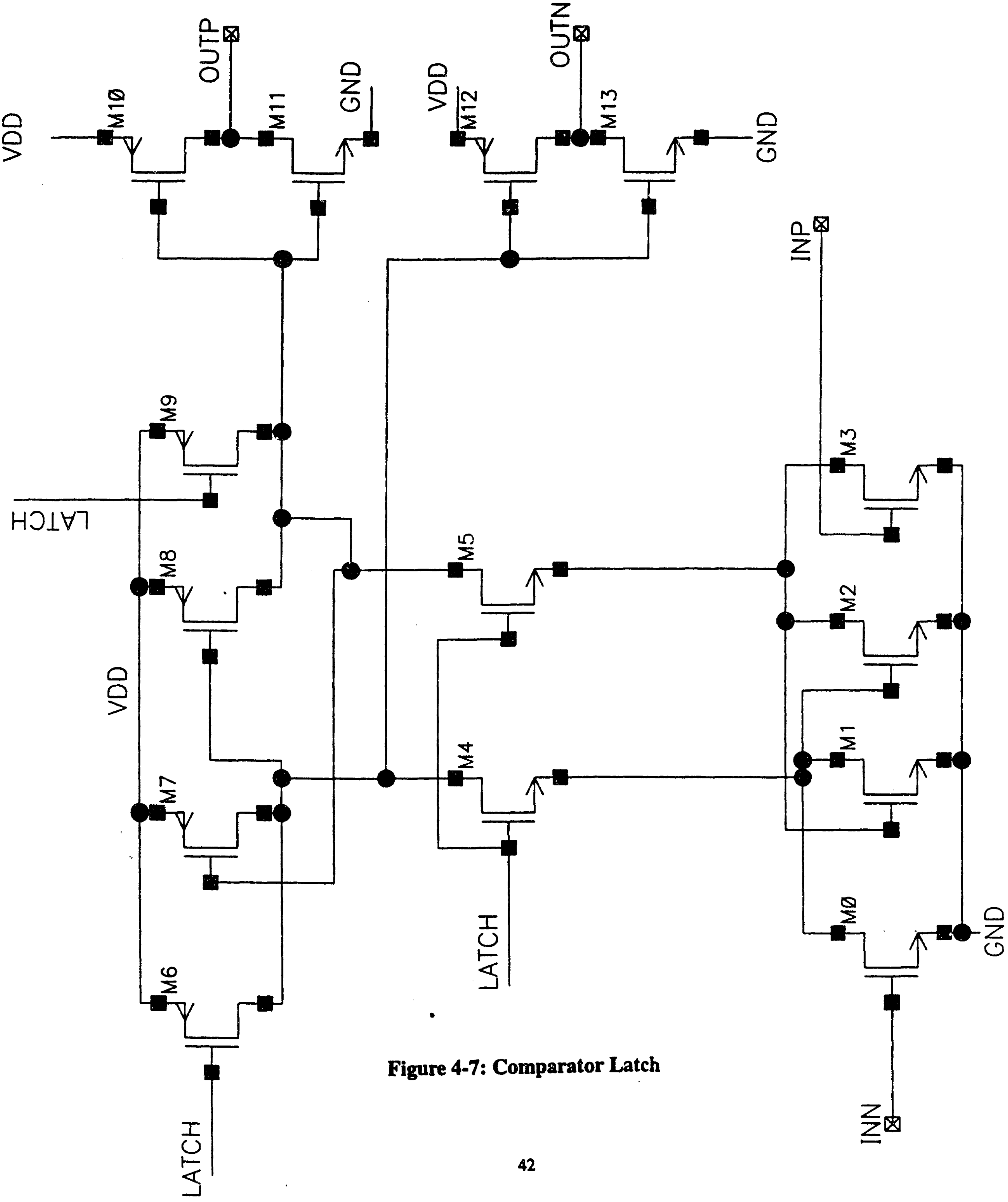


Figure 4-7: Comparator Latch

CHAPTER 5

RESULTS

Due to time constraints, it was not possible to build a physical implementation of this converter. Therefore, a mathematical model was developed in MATLAB to predict how the circuit would perform once constructed. This model used parameters found in ADICE simulations of the various components to simulate the converter performance. A listing of the MATLAB code used is found in the Appendix.

The basic procedure was to have ADICE determine the transfer curve (input vs. output) of the amplifier in unity-gain feedback, and to feed this information to MATLAB. MATLAB then performed a best-fit curve operation to determine the coefficients of the polynomial best resembling the ADICE transfer curve. This polynomial was used in the MATLAB simulation of the converter performance to represent the amplifier.

Figure 5-1 shows the simulated transfer curve. Figure 5-2 shows the INL curve, derived by subtracting the endpoint-to-endpoint straight line from the transfer curve of 5-1.

Table 3: Simulated Converter Specifications

Speed	100 kHz
Signal-to-noise ratio	89 dB
ENOB	14.5
INL	1.2 LSB
Power	165 mW

Figure 5-1: Simulated Transfer Curve

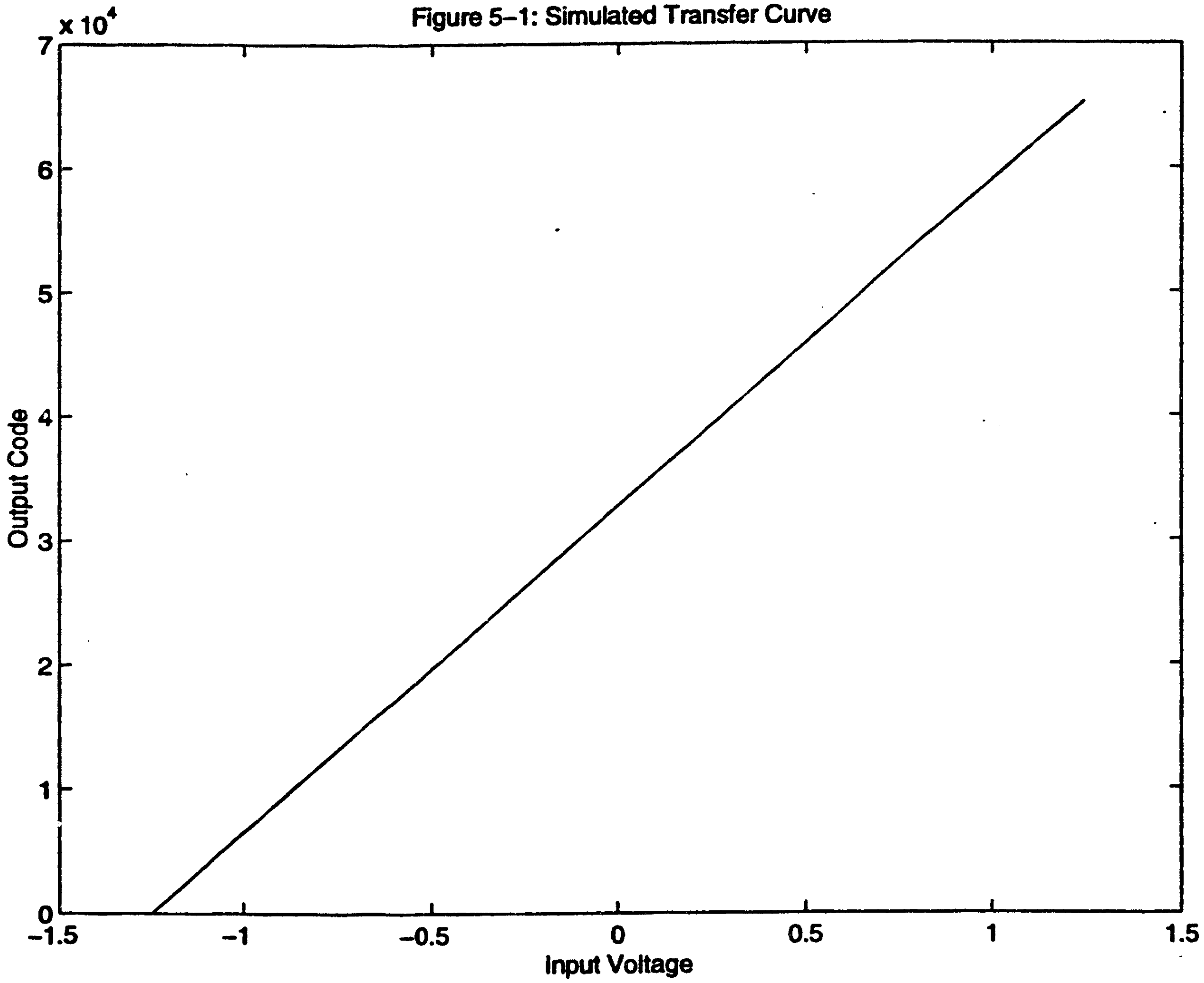
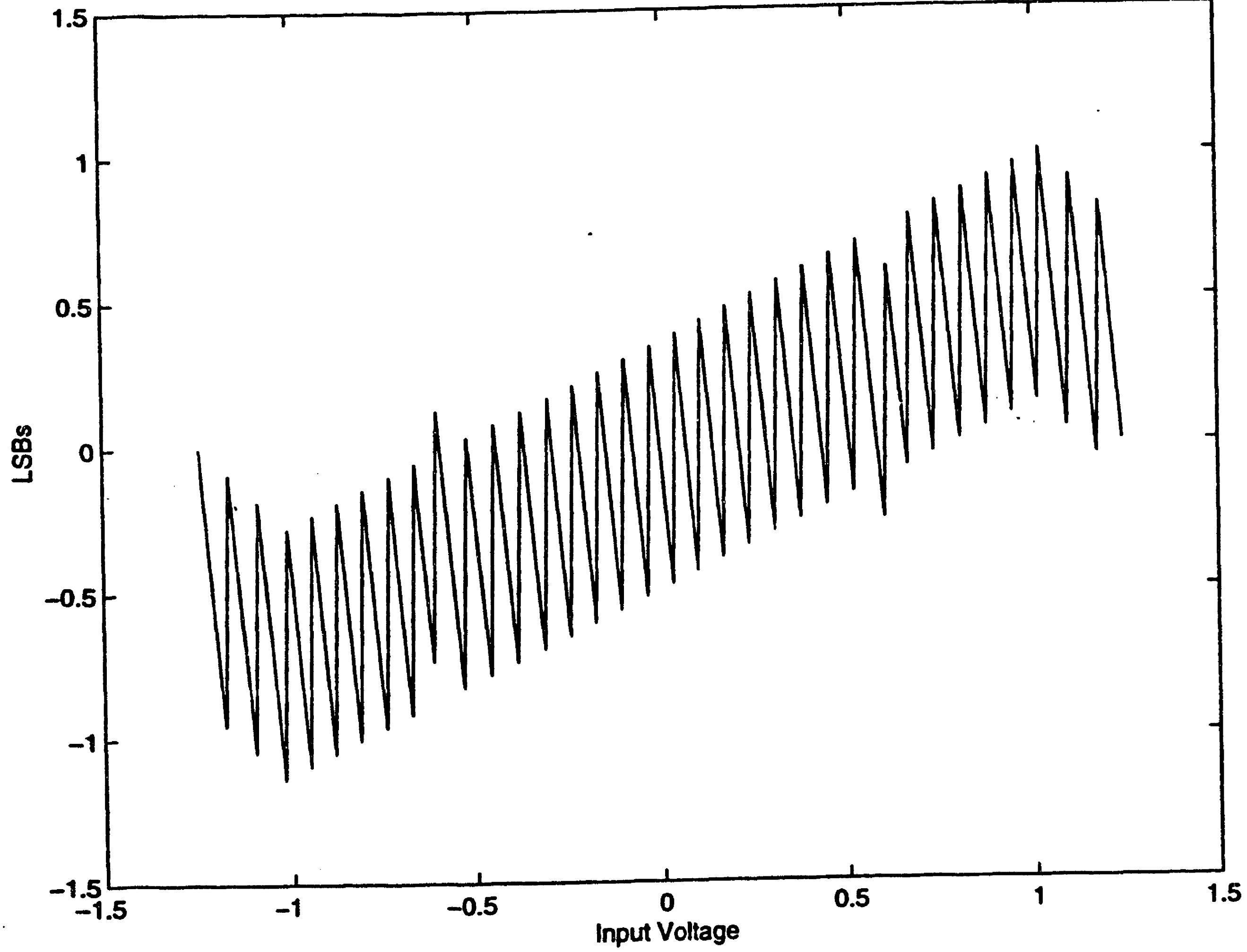


Figure 5–2 : Simulated INL



CHAPTER 6

CONCLUSIONS

Self-calibrating techniques are becoming important as manufacturers seek ways of reducing the overall costs of silicon integrated circuits. The reference-refreshing technique is an example of one of these techniques. The circuit designed here is a 16-bit reference-refreshing A/D which operates with parameters comparable to modern 16-bit A/Ds. However, at higher operating speeds, the power consumption of this design will also increase, eventually to impractical levels. Therefore, as 16-bit converters approach 1 MHz operating speeds and beyond, it is necessary to continue to investigate alternative self-calibrating techniques as a means to reducing circuit cost.

APPENDIX

This is a listing of the MATLAB code used to simulate the circuit and produce some of the graphs shown in this thesis. *nonlin.m* uses a listing of ADICE simulation data to generate polynomial coefficients that best represent the amplifier data. *refresh.m* uses these coefficients to simulate a conversion. *testconverter.m* generates a transfer curve of the converter.

nonlin.m

```
load data2;
x=data2(:,1);
y=data2(:,2);

j=1;
for j=1:10
        a=polyfit(x,y,j);
        if max(abs(y-polyval(a,x))) < 1e-9
                break;
end

end
```

refresh.m

```
function [vout] = refresh(vin,a)
N=16;
Vref=1.25;
Vhfs=Vref/2;
v=vin;
d=[];
if v>0
        d(1)=1;
        for i=2:N
                if v>Vhfs
                        Vhfs=polyval(a,polyval(a,Vhfs));
                        v=2*(polyval(a,polyval(a,v))-Vhfs);
                        d(i)=1;
                else
                        Vhfs=polyval(a,polyval(a,Vhfs));
                        v=2*polyval(a,polyval(a,v));
                        d(i)=0;
                end
        end
```

```
else
        d(1)=0;
        Vhfs=-Vhfs;
        for i=2:N
                if v<Vhfs
                        Vhfs=polyval(a,polyval(a,Vhfs));
                        v=2*(polyval(a,polyval(a,v))-Vhfs);
                        d(i)=0;
                else
                        Vhfs=polyval(a,polyval(a,Vhfs));
                        v=2*polyval(a,polyval(a,v));
                        d(i)=1;
                end
        end
end

vout=0;
for i=1:N
        vout=vout+(d(i)*2^(N-i));
end
```

testconverter.m

```
Vref=1.25;
LSB=Vref/(2^16);
x=[-Vref+LSB:0.01:Vref-LSB];
y=[];
INL=[];
[xx,yy]=size(x);
for j=1:yy
        y(j)=refresh(x(j),a);
end

figure(1)
plot(x,y)

m=(y(yy)-y(1))/(x(yy)-x(1));
b=y(1)-m*x(1);
hold on
plot(x,m*x+b,'r')
xlabel('Input Voltage');
ylabel('Output Code');
title('Figure 5-1: Simulated Transfer Curve');
hold off
```

```
INL=y-m*x-b;
figure(2)
plot(x,INL)
xlabel('Input Voltage');
ylabel('LSBs');
title('Figure 5-2 : Simulated INL');
```

BIBLIOGRAPHY

[1] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[2] D. A. Johns and K. Martin. *Analog Integrated Circuit Design*. John Wiley and Sons, Inc., New York, New York, 1997.

[3] S.-Y. Chin and C.-Y. Wu. "A CMOS Ratio-Independent and Gain-Insensitive Algorithmic Analog-to-Digital Converter", *IEEE Journal of Solid State Circuits*, Vol. 38, pp.1201-1207, August 1996.

[4] A. N. Karanicolas, H.-S. Lee, and K. L. Bacrania. "A 15b 1Ms/s Digitally Self-Calibrating Pipeline ADC", *IEEE Int. Solid-State Circuits Conf.*, pp. 60-61, San Francisco, February 1993.

[5] C.-C. Shih. "Precision Analog to Digital and Digital to Analog Conversion using Recirculating Algorithmic Architectures". University of California, Berkeley PhD Thesis, July 1985.

[6] A. Coban and P. Allen "A 1.75-V Rail-to-Rail CMOS Opamp", *Proceedings of the IEEE International Symposium on Circuits and Systems*, Vol. 5, pp. 5497-5500, London, June 1994.

[7] A. Yukawa. "A CMOS 8-bit High-Speed A/D Converter", *IEEE Journal of Solid State Circuits*, Vol. SC-20, no. 3, pp.775-779, June 1985.

THESIS PROCESSING SLIP

FIXED FIELD: ill. ____________ name ____________

index ____________ biblio ____________

► COPIES: [✓ Archives] Aero Dewey [✓ Eng] Hum

Lindgren Music Rotch Science

TITLE VARIES: ►□ ____________

NAME VARIES: ►☑ James

IMPRINT: (COPYRIGHT) ____________

► COLLATION: 50 l

► ADD: DEGREE: ________ ► DEPT.: ____________

SUPERVISORS: ____________

NOTES:

cat'r: date

page

► DEPT: E.E. ► F38

► YEAR: 2000 ► DEGREE: M.Eng.

► NAME: GUIDRY, Michael J.