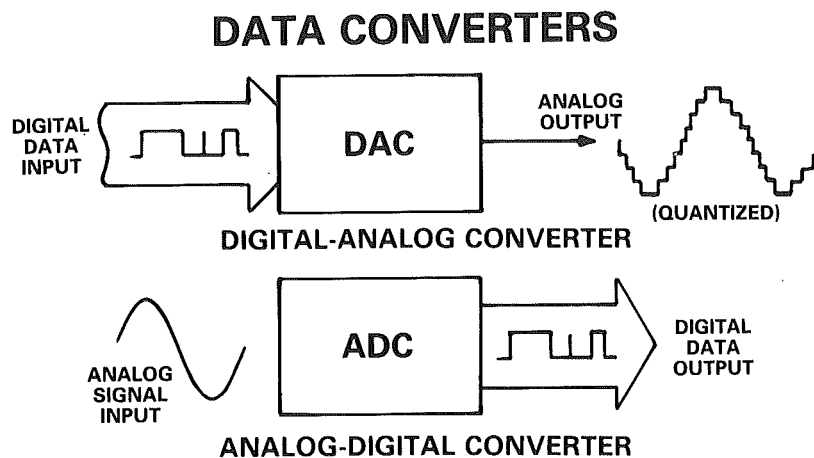


BASIC PRINCIPLES OF DATA CONVERTERS

BASIC PRINCIPLES OF DATA CONVERTERS

A Digital-to-Analog Converter, or DAC as it is generally abbreviated, is a device which multiplies a digital input and reference and outputs an analog quantity. Its converse is the Analog-to-Digital Converter, or ADC, which has an analog signal as its input and a digital representation of that input as its output.

Such definitions are rather imprecise—so let us consider each part of each definition in detail, starting with a DAC. A DAC has a digital input, which is to say that data is presented to it in the form of a numerical quantity, expressed in one of the standard ways in which numbers may be represented electronically. Different DACs will have different logic interfaces (for example TTL, ECL, CMOS, or some more specialized one), different data formats (simple binary, complementary binary, BCD, or some more complex and specialized format), and different input arrangements (data may be applied as a parallel word, a serial word, a number of nibbles to make a complete word, or whatever) but the essential feature from the point of view of definition is that the control input to a DAC is digital.



This section covers the basics of Data Converters, that is to say Analog-to-Digital and Digital-to-Analog Converters (ADCs and DACs). It will cover the basic functions, architectures by which these functions are accomplished, the specifications which apply to data converters, and the means by which they are measured.

DEFINITIONS

It is helpful to define what ADCs and DACs are in general terms and then to make the definitions more rigorous.

DAC CONTROL INPUT

May Be Any Logic Type: TTL, ECL, CMOS, $\pm 4\text{mA}$ etc.
May Be Any Format: Binary, Complementary Binary, Twos Complement, Ones Complement, BCD, Gray Code, or What Have You?
May Be Any Structure: Parallel, Serial or Word-Serial.

BUT IT MUST BE DIGITAL!
(There may be an analog reference input as well).

Many DACs have another input, the reference input, which is analog and not digital, but this need not be considered in the basic definition of a DAC although it will be dealt with in some detail later on.

The output of a DAC is analog. Engineers unfamiliar with data converters usually assume that the output of a DAC is a voltage but in fact it may be any analog quantity—the usual outputs are a voltage or a current, or an attenuation.

But although the output is analog it has one special characteristic which is unusual in analog signals—it is quantized. Since the input to a DAC is digital there can be only a finite number of input states—there can therefore be only a finite number of output states.

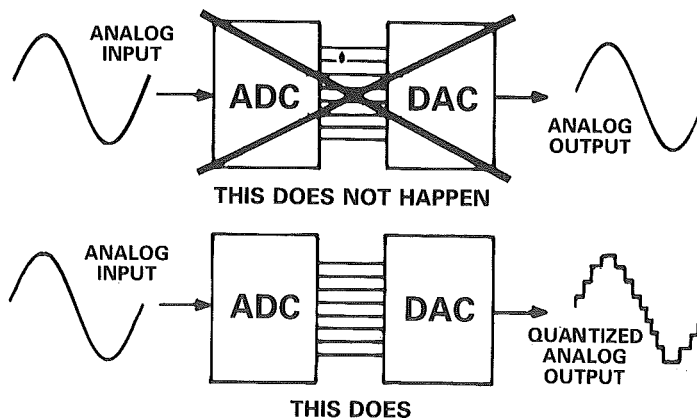
DAC OUTPUT

May Be Any Analog Variable: Usually Voltage, Current or Resistance (Attenuation)
But May Be Anything (Light, Water Flow, Phase Shift, Etc.)
May NOT Have Any Value: Since a DAC Has Digital Input It Has Only a Finite Number of Input States—It has, Therefore, Only a Finite Number of Output States.

DAC Inputs and Outputs are Quantized

Many people assume, without thinking, that if an ideal ADC and DAC are connected together and a smooth analog signal is applied to the ADC input it will reappear at the DAC output. This is not the case—the signal that appears at the DAC output is an approximation made up of discrete steps. The size of these steps depends on the resolution and conversion rate of the system. The difference between the original input and this quantized output is sometimes expressed in the form of noise on the input signal, known as “Quantization Noise”. In an N-bit system the quantization noise has a value of $-6N$ dB with respect to full scale.

QUANTIZATION NOISE



FOR AN N-BIT CONVERTER QUANTIZATION NOISE IS $-6N$ dB ON FULL SCALE

This mistake lies behind many problems in the design of systems involving data conversion and it is very important that engineers who use DACs and ADCs are constantly aware which signals are quantized and which are not.

We can thus define a Digital-to-Analog Converter as a device which converts a digital input to an output which is an analog parameter and we must note that the output, like the input, is quantized.

DIGITAL-TO-ANALOG CONVERTER FUNCTION

A Digital-to-Analog Converter (DAC) Converts Digital Code to an Analog Parameter.

Any Parameter may be a DAC Output but Common Ones Are:

- **Voltage**
- **Current**
- **Resistance (Attenuation)**

The Analog Output is Quantized

It is not necessary to spend so much time in considering the definition of an Analog-to-Digital Converter (ADC) since most of the considerations are similar. Most ADCs take an appreciable time to perform a conversion and so the digital interfacing is usually more complex. This is because, in addition to the data output, there may be a number of other digital ports such as “start conversion” and address inputs and “busy” (i.e., conversion in progress) outputs. Such complexity does not affect the definition of function.

The function of an ADC is to output, in digital form, an approximation to the value of its analog input. The output is only an approximation since the analog input may have any value within the defined input range and yet the digital output may only take one of a discrete number of values. There is therefore a difference between the actual output and the input of up to half an LSB which is known as the quantization uncertainty

ANALOG-TO-DIGITAL CONVERTER FUNCTION

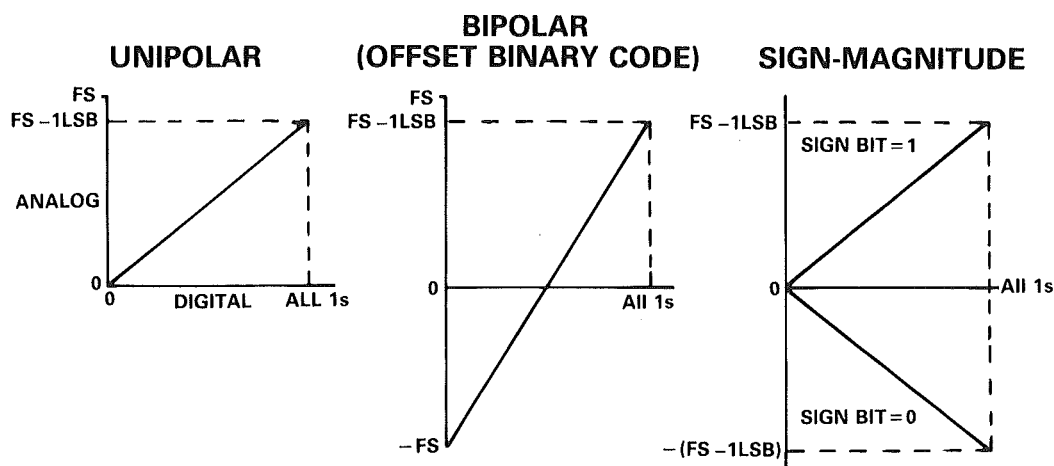
An Analog-to-Digital Converter (ADC) produces a digital output corresponding to the value of signal applied to its analog input.

Since the output is digital it can only take one of a finite number of discrete values and therefore may not exactly correspond to the value of the analog input. The difference is known as the QUANTIZATION UNCERTAINTY.

In addition to the basic ADC or DAC which we have defined real ADCs and DACs frequently contain many extra analog and digital functions. Since these are not essential to the ADC or DAC function there is no real point in discussing them at this stage in the seminar.

ADCs and DACs may be unipolar or bipolar. A unipolar device has an output with constant polarity while a bipolar device has an output which may be of either polarity. There are two types of bipolar device, the standard “bipolar” ADC or DAC and the “signed”. In the standard “bipolar” a digital zero signal (all 0’s) corresponds to analog full-scale negative, an all 1’s to full scale (minus 1 LSB) positive, and (MSB = 1 and all 0’s) to zero. In the “signed” ADC or DAC the analog signal goes from zero to full scale (minus 1 LSB) as the digital goes from all 0’s to all 1’s and an extra bit corresponds to polarity. It is obvious that all three have the same basic structure—the unipolar is the simplest case, the standard bipolar is a unipolar with a negative offset of 1 MSB, and a signed one is a unipolar one plus a sign bit.

DATA CONVERTER TYPES



We are assuming here that the code involved is binary—converters are sometimes built using Binary-Coded Decimal (BCD) Code or Gray Code but the majority are binary. (BCD converters are useful where digital displays are involved and Gray Code converters are invaluable where their property of changing only one bit at a time prevents errors during transitions from one code to the next). But there are several different binary codes which may be used with bipolar converters.

COMMONLY USED BIPOLAR CODES

Number	Decimal Fraction		Sign + Magnitude	Twos Complement	Offset Binary	Ones Complement
	Positive Reference	Negative Reference				
+7	+7/8	-7/8	0 1 1 1	0 1 1 1	1 1 1 1	0 1 1 1
+6	+6/8	-6/8	0 1 1 0	0 1 1 0	1 1 1 0	0 1 1 0
+5	+5/8	-5/8	0 1 0 1	0 1 0 1	1 1 0 1	0 1 0 1
+4	+4/8	-4/8	0 1 0 0	0 1 0 0	1 1 0 0	0 1 0 0
+3	+3/8	-3/8	0 0 1 1	0 0 1 1	1 0 1 1	0 0 1 1
+2	+2/8	-2/8	0 0 1 0	0 0 1 0	1 0 1 0	0 0 1 0
+1	+1/8	-1/8	0 0 0 1	0 0 0 1	1 0 0 1	0 0 0 1
0	0+	0-	0 0 0 0	0 0 0 0	1 0 0 0	0 0 0 0
0	0-	0+	1 0 0 0	(0 0 0 0)	(1 0 0 0)	1 1 1 1
-1	-1/8	+1/8	1 0 0 1	1 1 1 1	0 1 1 1	1 1 1 0
-2	-2/8	+2/8	1 0 1 0	1 1 1 0	0 1 1 0	1 1 1 0
-3	-3/8	+3/8	1 0 1 1	1 1 0 1	0 1 0 1	1 1 0 0
-4	-4/8	+4/8	1 1 0 0	1 1 0 0	0 1 0 0	1 0 1 1
-5	-5/8	+5/8	1 1 0 1	1 0 1 1	0 0 1 1	1 0 1 0
-6	-6/8	+6/8	1 1 1 0	1 0 1 0	0 0 1 0	1 0 0 1
-7	-7/8	+7/8	1 1 1 1	1 0 0 1	0 0 0 1	1 0 0 0
-8	-8/8	+8/8		1 0 0 0	0 0 0 0	

The simplest code, offset binary, simply increases from all 0's to all 1's as the analog quantity goes from full scale negative to 1 LSB below full scale positive. Complementary binary is exactly the same code but with negative logic (0's become 1's and 1's become 0's).

Two's complement is the same as offset binary, except that the MSB is inverted (i.e., it is 1 below zero and 0 from zero on up), and one's complement is a bipolar code formed by inverting all the bits of unipolar binary code for the corresponding negative values—it has the same problem as the signed (sign + magnitude) converter in that there are two codes for zero.

STATIC SPECIFICATIONS

Data converter specifications are frequently confusing but are essentially simple—the confusion most often arises from inconsistent use of units. The error in a converter may be specified in least significant bits (LSB), millivolts (or microamps in a current operated device), percentage of full scale, or parts per million (ppm) of full scale. Quantization noise is expressed in decibels (dB) below full scale. Different manufacturers, or even one manufacturer on different occasions, will express the same parameter in any of these units and it is essential that any user of data converters be familiar with the conversion. The table summarizes it very simply and should, if possible, be memorized—if the whole table is too long the 10-bit row is easily remembered (10V/10 bits/1 LSB $\approx$ 10mV $\approx$ 1/10 percent) and the others may easily be calculated from it by multiplying or dividing by powers of 2.

DATA CONVERTER ERROR TERMINOLOGY (10 VOLT FULL SCALE)

RESOLUTION (BITS)	mV	% FS	ppm FS	dB FS
8	39.1	0.391	3906	- 48
10	9.77	0.098	977	- 60
12	2.44	0.024	244	- 72
14	0.61	0.006	61	- 84
16	0.153	0.0015	15	- 96
18	0.038	0.00038	3.8	- 108
20	0.0095	0.00010	1.0	- 120

A SIMPLE MNEMONIC IS TO REMEMBER THAT AT 10V FS AND 10-BITS

1 LSB $\approx$ 10mV $\approx$ 1/10 % FS ($\approx$ 1000ppm $\approx$ -60dB).

ONE CAN THEN WORK IN POWERS OF 2 TO CALCULATE THE OTHERS.

D/A CONVERTER ERRORS FOUR BASIC CONVERTER ERRORS (STATIC, 25°C)

1. Offset
The error which adds to all codes equally. (User trimmable)
- 1a. Zero
The error when the digital input code calls for zero output.
2. Gain (Scale Factor)
An error in the slope of the transfer characteristic. It affects all codes by the same percentage. (User trimmable)
3. Nonlinearity (Integral Nonlinearity [INL] or Relative Accuracy)
Integral nonlinearity is a measure of the deviation of a converter's transfer characteristic from a straight line. (Not user trimmable)
4. Differential Nonlinearity [DNL]
The measure of the difference between the ideal (1LSB) and the actual change in analog output for 1LSB increase in digital code. (Not user trimmable)

There are four basic dc specifications for data converters: gain, offset, linearity and differential linearity. Of these, gain, offset and differential linearity may vary sufficiently with variation of temperature that their temperature coefficients must also be specified. The linearity of a converter does, or course, vary with temperature but rarely enough for it to be necessary to specify its temperature coefficient.

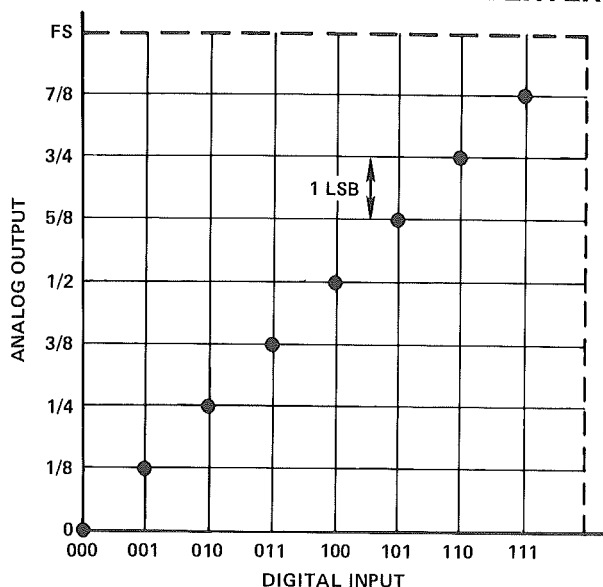
TEMPERATURE COEFFICIENTS

1. Offset TC
A shift due to temperature which affects all readings equally and is expressed as $\mu\text{V}/^\circ\text{C}$ or ppm FS/ $^\circ\text{C}$.
- 1a. Zero TC
Zero shift due to temperature. In a bipolar DAC the zero TC has contributions from both the offset and gain TCs.
2. Gain TC
Change of the slope of the transfer characteristic with change of temperature. It is expressed as ppm of reading/ $^\circ\text{C}$. (Reference may or may not be included in this specification)
3. Differential Nonlinearity TC
Shows the relative change in bit-weights with temperature and is a measure of when a converter can be expected to go nonmonotonic. It is expressed a ppm FS/ $^\circ\text{C}$.

All the static specifications (and their temperature coefficients) may be defined by considering the graph of the transfer characteristics of the converter. Because the definitions that follow have been written to be equally true for both ADCs and DACs they are slightly more complex and pedantic than the definitions for one or the other but this generality emphasises that essential similarity of converter specifications. We should, however, consider the transfer characteristics of DACs and ADCs separately before considering their static specifications so that the differences as well as the similarities are quite clear.

The transfer characteristic of an ideal unipolar 3-bit DAC is shown in the diagram. It consists of eight points which lie on a straight line. All 0's code gives zero output and all 1's gives 1 LSB below full scale.

CONVERSION RELATIONSHIP FOR AN IDEAL 3-BIT D/A CONVERTER

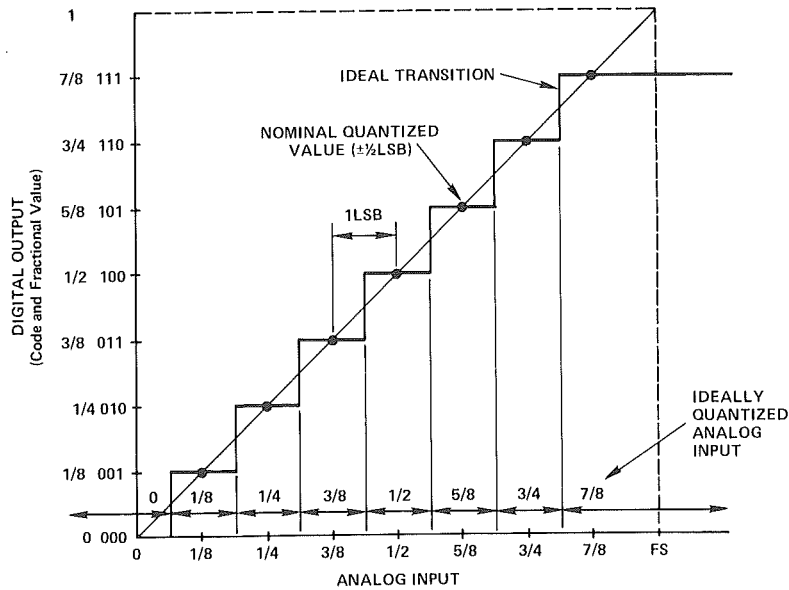


The output for all 1's is 1 LSB below full scale and not full scale because a DAC can be considered as a digitally-controlled potentiometer which produces an analog output which is a normalized fraction of its reference input—this reference is “full scale” and to produce full-scale output from an N-bit DAC would require a digital input of 1 in the (N + 1)th bit and then all 0's. Since a N-bit DAC does not have an (N + 1)th bit the largest possible output is when the input is all 1's and the output is then 1 LSB below the reference or “full scale”.

As we keep emphasizing, a DAC has both its inputs and its outputs quantized and it is misleading to think of its characteristic as a line—it is a series of points. Nevertheless it can be quite useful to consider the line through these points when discussing the definitions of DAC specifications.

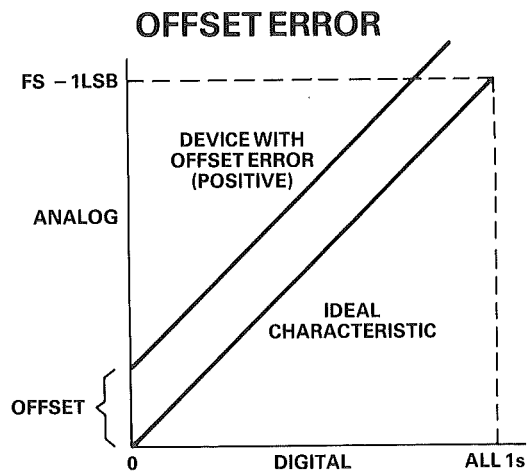
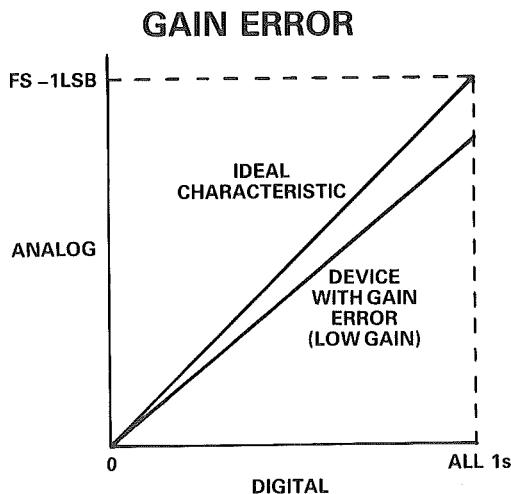
The transfer characteristic of an ideal 3-bit ADC is also shown. In this case the analog input may have any value within its range but the digital output is still quantized. Out of range inputs normally produce all 0's or all 1's outputs, depending on whether they are underrange or overrange respectively, but some types of ADC do not behave in such a convenient way and these types may have output(s) indicating out of range inputs. If the analog input is increased the first change in an ideal ADC's output occurs for an input of 0.5 LSB and subsequent changes occur 1 LSB apart until the last takes place 1.5 LSB below full scale. The resulting diagram is a series of steps, 1 LSB wide, and the line which we consider when discussing specifications is the line through the midpoints of these steps.

CONVERSION RELATIONSHIP FOR AN IDEAL A/D CONVERTER



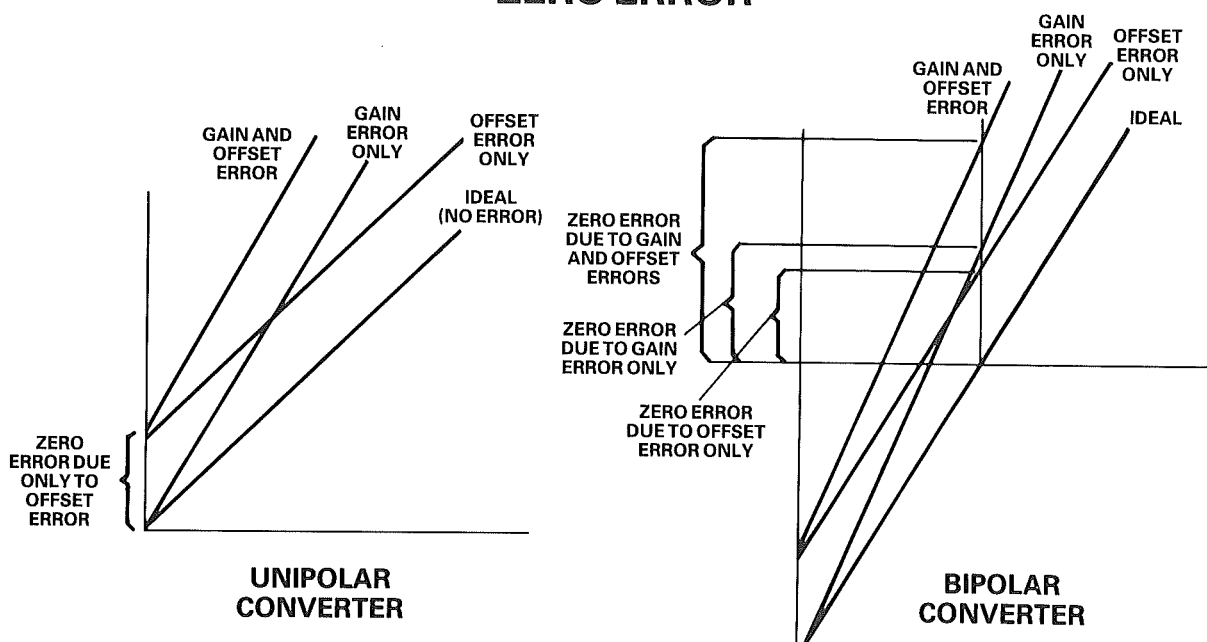
The first specification to consider is the gain. This is defined as the analog scale factor governing the relationship between signal amplitude at the analog and digital ports (when we define digital amplitude we refer, of course, to the amplitude represented by the digital code, not to logic levels). More simply the gain is the gradient of the transfer characteristic. Ideally a change of K LSB in analog signal should correspond to a change of K LSB of digital code—if the gradient is steeper or shallower than this there is said to be a gain error. Normally the gain error is measured by noting the analog change for a digital change of all 0's to all 1's—it should be $(2^n - 1)$ LSB. Gain error applies (multiplicatively) to all codes equally and may therefore be adjusted by the converter user who can usually increase or decrease the gain by altering resistance somewhere in the converter.

The next specification that we shall consider, offset, also applies to all codes equally. The offset error adds to all codes and, again, may generally be user trimmed by adding an equal and opposite error so that the two cancel—the way that this is done depends on the converter architecture.



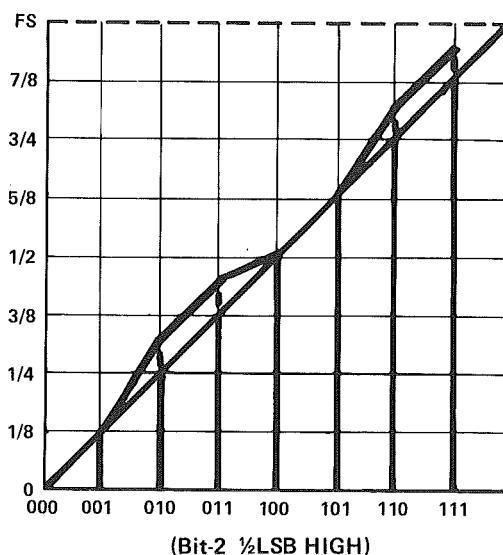
There is frequently confusion (even among data sheet writers) about the difference between offset and zero errors. The zero error is defined as the error in analog value when the digital code calls for analog zero. In unipolar devices the offset error and the zero error are identical but in bipolar devices the zero error contains contributions from both offset and gain error. For this reason the two terms should not be considered interchangeable.

ZERO ERROR



Both gain errors and offset errors, applying equally to all codes, may be trimmed by the converter user. The other two dc errors do not and may not. The specifications are linearity and differential linearity and the corresponding errors are known as (integral) nonlinearity and differential nonlinearity (abbreviated to INL and DNL) respectively.

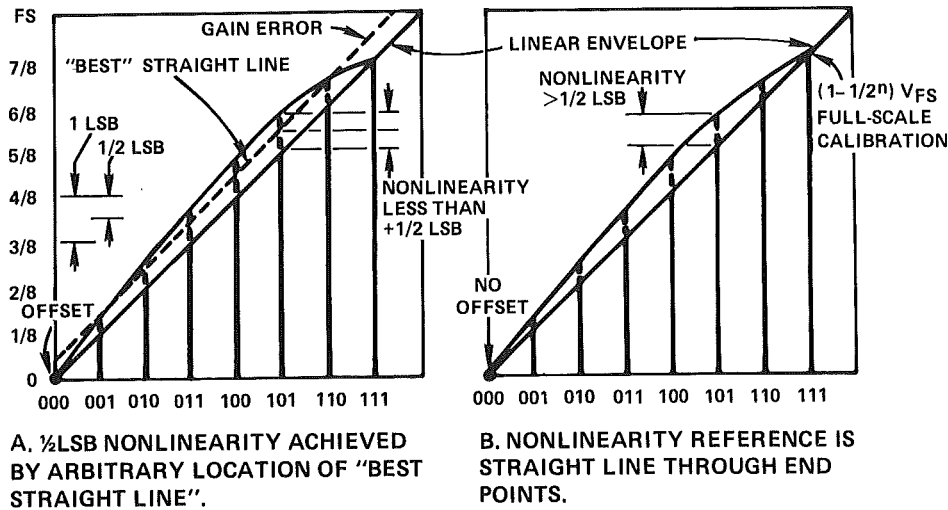
LINEARITY ERROR



The INL is the deviation of the transfer characteristic of the converter from a straight line. At Analog Devices INL is defined by the "end-point" method—that is to say the INL is defined as the maximum difference between the actual plot of the device response and the straight line drawn from the all 0's point to the all 1's point. It is not unknown for some manufacturers to define INL by the "best straight line" method—here the INL is defined as the maximum difference between the actual plot of the device and the best straight line through its points. Such a definition has the advantage of giving a better "paper" specification but the disadvantage

of being virtually unuseable as a specification for system error analysis (a system is not expected to work to a best straight line—it is expected to work to the ideal straight line, the one through the end points). For most practical purposes the end-point figure is twice as large as the best straight line one (although it is possible to invent circumstances where it is not).

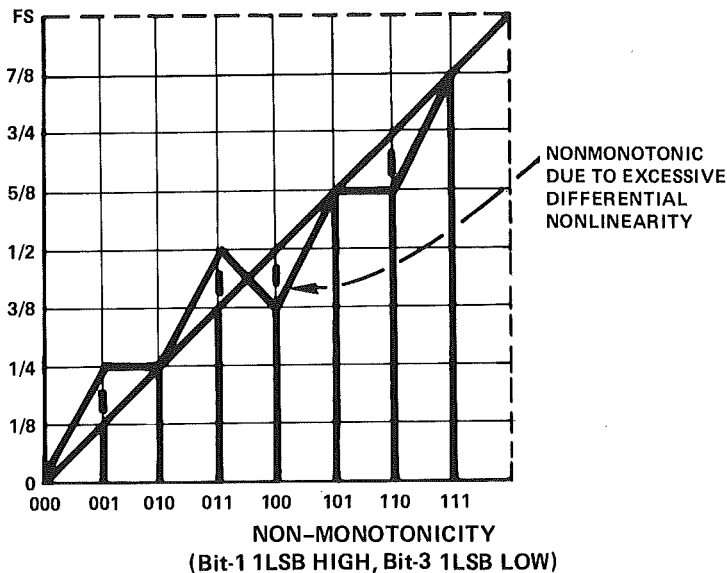
COMPARISON OF LINEARITY CRITERIA



The differential nonlinearity (DNL) of a converter describes the variation in analog value between pairs of adjacent digital codes. In an ideal converter there is no variation, a digital code change of 1 LSB always corresponds to an analog change of 1 LSB, and the DNL is zero—in practice this rarely occurs. Although the basic principle is the same for both ADCs and DACs the visible effects on the plots of their characteristics differ sufficiently that it is worthwhile looking at the two cases separately.

In an ideal DAC an increase of digital code by 1 LSB causes the output to increase by 1 LSB. If, at any particular code transition, the analog change is not 1 LSB then the device is suffering from DNL at that transition.

DIFFERENTIAL NONLINEARITY



DNL always refers to the change of analog output with +1 LSB change of code—even if the analog output is correct at a particular code there may be still be DNL in the transition to that code. If the DNL is negative and has a magnitude of over 1 LSB then an increase of digital code will cause a decrease in analog output. A DAC which behaves in this way is said to be nonmonotonic. A mathematical function is said to be monotonic between certain limits if it has no maxima or minima between those limits. A DAC is said to be monotonic if it has no local maxima or minima in its transfer characteristic—that is to say that the slope of its transfer characteristic has the same sign over its whole range.

Monotonicity is critical for many DAC applications—if a DAC in a servo system of any sort is nonmonotonic the feedback becomes positive rather than negative in the nonmonotonic region and the system may oscillate or latch. For this reason most DAC specifications contain guarantees of monotonicity to certain levels or resolution. However, if the DNL is guaranteed to be less than 1 LSB then the DAC will be monotonic, even if this is not explicitly guaranteed.

MONOTONICITY

Monotonicity requires that a converter NEVER give decreasing analog output for increasing digital code. Nonmonotonicity is the result of excess differential nonlinearity ($>1\text{LSB}$).

Monotonicity is essential for many control applications.

A converter which has DNL specified $\leq 0.5\text{LSB}$ is better characterized than one which is merely "monotonic" – but most manufacturers would specify the monotonicity as well, even though the specification is redundant.

Sometimes a data sheet will guarantee monotonicity at a particular temperature only. It is often possible to predict whether the device will be monotonic at other temperatures. In the example given two DACs are compared, both with initial accuracy of 60ppm but one having a DNL TC of 1ppm/°C and the other having 2ppm/°C. At +125°C the first has a total error of $60 + 100 \times 1 = 160\text{ppm}$ which is less than 240ppm so the device is monotonic to 12 bits, the second has a possible total error of $60 + 100 \times 2 = 260\text{ppm}$ which is more than 240ppm so this device MAY not be 12-bit monotonic (if it were to be guaranteed monotonic at 125 degrees the manufacturer would be guaranteeing that at least some of the errors would cancel—this might be a reasonable guarantee). The same computation and considerations may be applied to an ADC to predict missing codes (see below).

MONOTONICITY

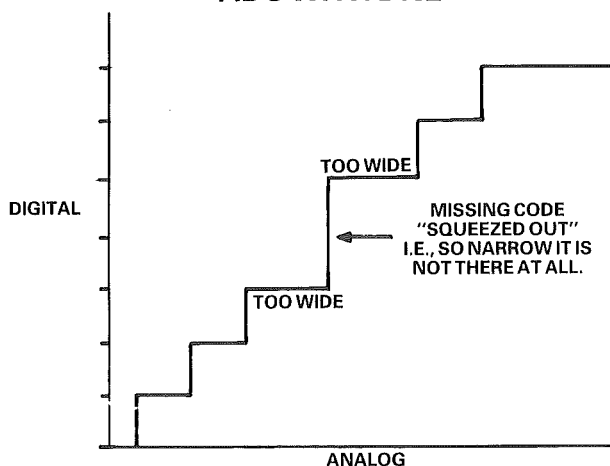
- Factors Contributing:**
1. Initial Accuracy
 2. TC of DNL
 3. Temperature Change

Examples (12-bit DAC – $1\text{LSB} = 244\text{ppm FS}$):

Initial Accuracy:	0.25LSB (60ppm)	0.25LSB (60ppm)
DNL TC:	1ppm/°C	2ppm/°C
Temperature Range	–55°C to +125°C	–55°C to +125°C
Max Possible Temperature Change:	+100°C	+100°C
Max Possible DNL Change:	100ppm	200ppm
New Max Possible Error:	$60 + 100 = 160$	$60 + 200 = 260$
Monotonic (Error $< 1\text{LSB}$)?	YES	NO

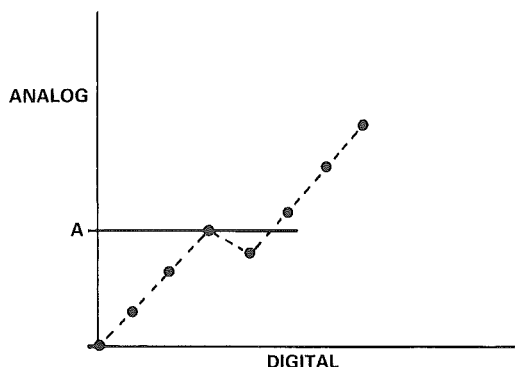
In an ideal ADC each code is exactly 1 LSB wide. The effect of DNL on an ADC is to make particular codes wider or narrower than this—as in the case of a DAC DNL is generally specified in terms of LSBs. If the DNL of an ADC is less than -1LSB for any reason the code width will become negative, that particular code will not occur, and the converter will suffer from a missing code. Missing code(s) is the ADC equivalent of nonmonotonicity and is as serious in many applications.

ADC WITH DNL



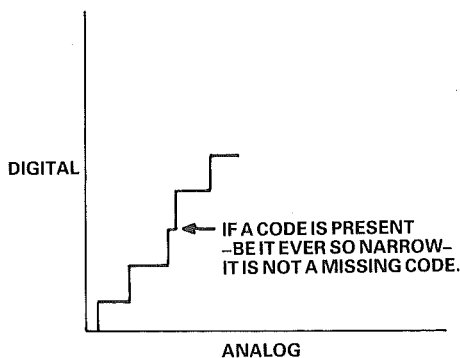
The normal cause of missing codes in an ADC containing a DAC is that the DAC is nonmonotonic. ADCs using DACs search through various DAC codes until they find one with an analog output which matches the analog input to the ADC (the exact search algorithm used depends on the type of ADC). If the DAC is nonmonotonic the algorithm may fail and certain codes will not occur. Consider the 3-bit DAC in the diagram—if it is used as a component in an ADC any input large enough to give an output of 3 LSB is also large enough to give an output of 4 LSB (since the DAC output for 3 is greater than the DAC output for 4 and the search algorithm will look at 4 before it looks at 3). Thus the ADC will never give an output of 3—so 3 is a missing code.

NONMONOTONIC DAC MAKES AN ADC WITH MISSING CODES



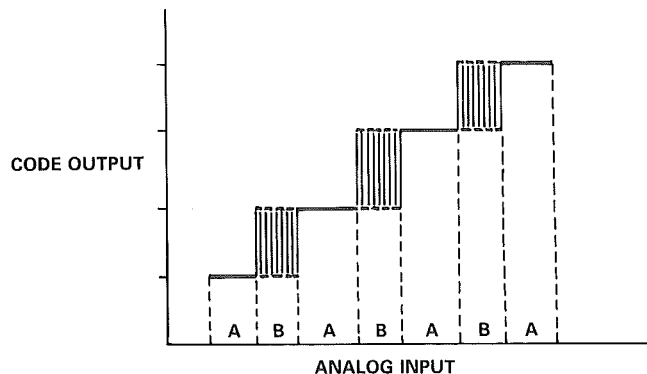
There is some dispute over the exact definition of “missing code” in real ADC specifications. In an ideal, noise-free, ADC if the DNL is never less than -1 LSB then every code, be it ever so narrow, will be present. But real ADCs have noise which results in a transfer characteristic which is blurred at code transitions.

MISSING CODES



SIMPLISTIC DEFINITION OF MISSING CODE

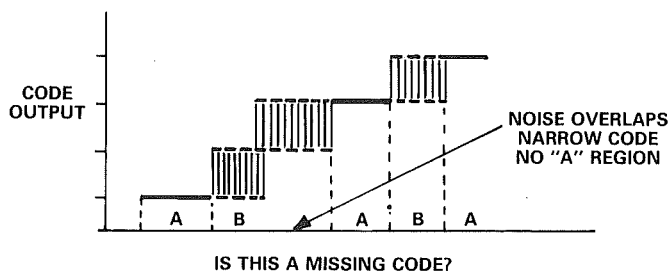
MISSING CODES



EFFECT OF A NOISY CONVERTER

If the analog input lies in one of the regions A in such a noisy ADC the output will always be the same code, but if it lies in one of the regions B it is evident that converter noise may cause the output to be one of the two adjacent codes. If such a noisy converter also has poor DNL it may be that some of the A regions may be squeezed out.

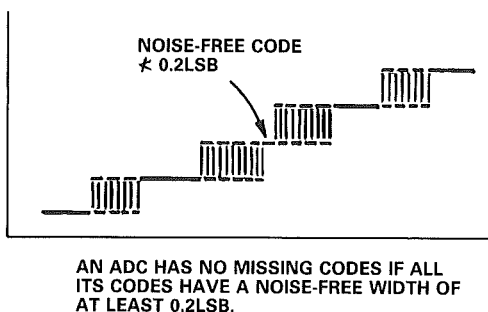
MISSING CODES



EFFECT OF NOISE AND DNL COMBINED

In this case the code which is always noisy will sometimes be missing—but not always. Should it be considered a missing code or not? This point has caused considerable discussion among converter manufacturers and users and opinions range from the low-cost manufacturer's "If a code is present sometimes then it can't be missing."—which certainly keeps his yields up to the purist's "If a converter is noisy then I'm not interested anyway."—There is consensus that a code should be noise-free for at least 0.2 LSB of its range in order not to be considered missing—this is a demanding specification at high resolution since it calls for DNL to be $<0.8\text{LSB}$ in a noise-free converter and for the noise to be less than $(0.4\text{LSB} - \text{DNL}/2)$ in a noisy one.

MISSING CODES



CONSERVATIVE MISSING CODE DEFINITION

In addition to the four major static specifications (and the temperature coefficients of three of them) there are a number of other static specifications which are normally defined on converter data sheets. The majority, such as input impedances, logic levels, power consumption, etc., are self explanatory and need not be considered here, but two, power supply sensitivity and absolute accuracy, should be mentioned further.

In an ideal converter, the performance is unaffected by variations in the power supply (in fact the diagram of the ideal converter does not even show a power supply—the ideal converter, like the perfect black body, the reversible heat engine, and the "little old lady who only used it to go to church on Sundays", is a useful theoretical concept which does not exist in reality). In fact many (but by no means all) converters will only work within quite narrow ranges of power supply and even within these limits variations in supply may affect gain, offset and linearity. The greatest effect is generally upon gain—the result of imperfections in the reference and buffer—and such converters normally have power supply sensitivity (variation of gain vs. variation of supply voltage) as one of their specifications.

The absolute accuracy of a converter is the difference between the actual and expected output for any given input. It is the sum of the gain, offset and linearity errors at that input, and the figure may be calculated for a particular input (remembering that gain errors are proportional to input and, if end-point linearity is considered, linearity errors are greatest away from the end-points) or for a worst case over the whole transfer characteristic. As we mentioned earlier, ADCs, but not DACs, have another error, the quantization uncertainty, which is

intrinsic to all ADCs and adds 0.5 LSB uncertainty even to a perfect one. Absolute accuracy and its variation with temperature are not often specified on a data sheet but may be calculated by the user, from the static specifications which are given, for the particular circumstances under which the converter is to be used.

Such "error budget" calculations can be frightening—each error, in itself, is small but the total, over temperature, frequently causes system specifications to be rethought or much more accurate converters to be specified.

DYNAMIC SPECIFICATIONS

In addition to static specifications, converters have noise specifications and speed related dynamic specifications. Noise is defined in the usual way and relates to the analog input of ADCs and the analog output of DACs.

CONVERTER NOISE

In addition to noise feedthrough from the digital to the analog parts of converters the analog sections themselves are affected by noise.

This occurs as voltage and current noise—with similar spectral characteristics to noise in other analog circuitry—at the input of ADCs and the output of DACs.

The speed related dynamic specifications differ greatly between ADCs and DACs. A detailed study of the dynamic performance of high speed converters is contained in a seminar prepared by the Computer Labs division of Analog Devices and will not be given here. This seminar will merely consider the speed related specifications of normal low-speed converters.

The speed specifications of ADCs are of two type—analog and digital. The digital specifications relate to the switching speeds and propagation delays in the output logic circuitry and will be discussed in the "Microprocessor Interface" section of this seminar. The analog specifications are the conversion time and the maximum input signal bandwidth.

ANALOG-DIGITAL CONVERTERS MAJOR DYNAMIC SPECIFICATION IS CONVERSION TIME

**IT MAY BE BETWEEN NANOSECONDS AND SECONDS
DEPENDING ON THE TYPE OF ADC.**

Considered as a black box an ADC has an analog input and a digital output. It is rarely possible, however, to assume that the digital output reflects the state of the analog input a fixed period before—most conversion techniques are more complex than that. We shall discuss different conversion techniques and their timings in detail in the section on "ADC Architecture"—at this point it is sufficient to say that most ADCs require a digital "start conversion" command and the digital output data becomes available some time later. This delay is called the "conversion time". Different ADCs may have conversion times which range from nanoseconds to seconds.

Some ADCs latch their analog input during conversion. Such an analog latch is called a "Sample-and-Hold Amplifier" or "Track-and-Hold Amplifier" variously abbreviated to SHA, SAH, or THC—it is intrinsic to some ADC architectures and is sometimes added to other types of ADC to improve their performance. The most important SHA specification is usually its acquisition time. SHAs are discussed in detail in the "Data Acquisition Subsystems" section of this seminar.

If the analog input is not latched it must not change by more than 0.5 LSB during a conversion unless the converter is of a type (such as a VFC or an integrating ADC) which integrates and averages the input signal during the conversion period. This severely limits the input bandwidth of an ADC (the bandwidth of an integrating ADC is limited by its own integration). Consider an n -bit ADC which is digitizing a sine wave whose peak-peak amplitude is the full-scale range of the ADC. During the conversion period the input may not change by more than 0.5 LSB. At its maximum rate of change a signal of F Hz changes 0.5 LSB in $1/(2^{n+1} \times \pi \times F)$ seconds. This gives us an equation relating n , $F(\max)$ (the maximum input frequency which may be digitized without a SHA), and T_c (the conversion time).

ADC WITHOUT A SHA

$$F(\max) = \frac{1}{2^{(n+1)} \cdot \pi \cdot T_C} \text{ Hz}$$

Solving for $n=16$ and $T_C = \mu\text{s}$ gives a maximum input frequency to a $10\mu\text{s}$ 16-bit ADC ($10\mu\text{s}$ is very fast for a 16-bit ADC) of 0.24Hz!

For an AD670 (8-bit $10\mu\text{s}$) the figure is 124Hz and for an AD7572 (12-bit $5\mu\text{s}$) it is 8Hz.

This equation leads us to the startling conclusion that even such a fast ADC as the AD7572 $T_C = 5\mu\text{s}$) has an input bandwidth, when used without a SHA, of less than 8Hz.

The two important DAC specifications are settling time and glitch—maximum update rate is also important but unless the DAC logic is unusually slow it is normally limited by the settling time and need not be considered separately. Multiplying DACs should also specify reference bandwidth, both -3dB and 1 LSB feedthrough.

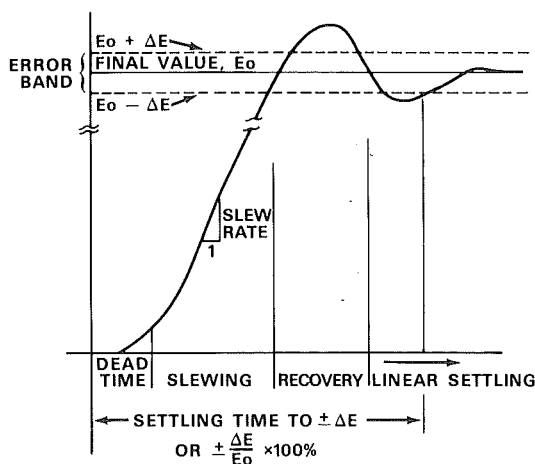
Settling time specifications are very confusing. Manufacturers seem to delight in using different standards for different devices so that comparison is difficult, if not impossible. The reason is simple—specifications are chosen to show the device in the best possible light. If we are to make sense of a settling time specification we should understand what is meant by settling time.

The settling time of a DAC is the time taken for the output to move to, and remain within, a specified error band after the digital input has changed. It is important to note the “and remain” part of this definition—in many DACs the output will ring in and out of the error band for a few cycles before finally settling within it—the specification relates to the final entry to the error band, not the first one. The settling time may be subdivided into dead time, slewing, recovery, and linear settling time. The dead time, known as “propagation delay”, is the time between the digital change and the analog output starting to move, the slewing time is the time during which the analog output is changing as fast as it can (i.e., at its maximum slew rate), the recovery time is the time during which the converter recovers from its fast slewing phase, and the linear settling time is the time during which it settles to its final state.

In general we may consider only the total settling time and not bother with subdividing it, but if the slewing time occupies the majority of the total settling time then the settling time will be proportional to the step size and different analyses may be necessary for different step sizes.

If the device is critically damped there is no recovery time and the output passes directly from slewing to linear settling. Some DACs are hard to damp in this way since the impedance of their internal current output DAC varies with code and the current-voltage converter cannot be optimized for a fixed source impedance.

SETTLING TIME MEASUREMENT

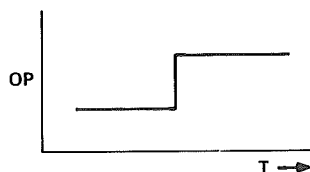


THE SETTLING TIME OF A DAC IS THE TIME BETWEEN THE LOGIC EDGE WHICH INITIATES A CODE CHANGE AND THE ANALOG OUTPUT MOVING, AND REMAINING, WITHIN THE SPECIFIED OUTPUT ERROR BAND.

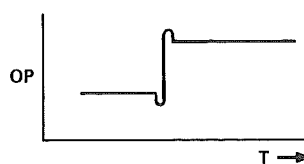
When considering settling time specifications it is also important to realize that in some types of DAC different transitions have different rates (for example positive going will be faster than negative going if pull-up is with a transistor and pull-down with a resistor), and the settling in a voltage output DAC will be slower than in a current output DAC of the same general type since capacitance charging takes time.

If we change the digital input to a DAC we expect the analog output to change from one value to another. In practice the output during switching may go well outside the region between the start and finish values—this effect is called a “glitch” and has two separate causes.

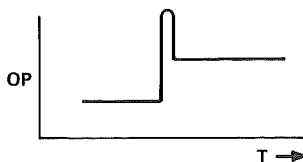
GLITCHES



IDEAL DAC TRANSITION



“DOUBLET” GLITCH CAUSED BY CHARGE COUPLING



GLITCH CAUSED BY LOGIC SKEW

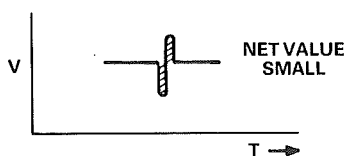
If stray capacity couples the digital circuitry to the analog circuitry (as is inevitable to some extent) then when the digital circuitry switches a transient will appear in the analog circuitry. Such capacity will be due both to stray capacity on the DAC chip and to package and circuit-board capacity. In many types of DAC (but not all!) this type of glitch is fairly small (and can be minimized by minimizing stray capacities external to the DAC itself), does not vary much with code, and frequently consists of a “doublet” pulse having equal positive and negative excursions. This type of glitch is thus generally fairly easy to filter from the DAC output.

If positive-going and negative-going transitions in the DAC logic have different speeds then during a code change the faster operation will be complete first. This means that there will be a short period when some of the switches in the DAC have changed and some have not—this causes the DAC output to go to an intermediate state which may be well outside the region between its start and finish. Glitches of this type are frequently much larger than the previous type, are not “doublets”, and vary greatly with code. They are much harder to filter.

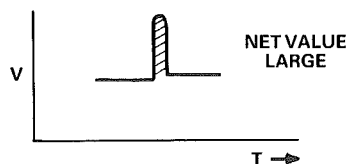
Glitches may be removed by filtering the DAC output (at the cost of increased settling time), by placing a SHA in the DAC output which is set to hold during DAC transitions (this is a good method but is limited by glitches in the SHA itself), or by designing the DAC for minimum glitch impulse.

Although the specifications of glitches often speaks of “glitch energy” or “glitch charge” these terms are often misnomers—while it is quite reasonable to evaluate the energy (in picojoules) or the charge (in picocoulombs) associated with a glitch, the figure generally specified by DAC manufacturers is the impulse—the net product of volts and time in the glitch pulse.

GLITCHES



NET VALUE SMALL



NET VALUE LARGE

GLITCH IMPULSE

OFTEN (INCORRECTLY) CALLED GLITCH ENERGY

Glitch is normally specified for the worst case (i.e., the transition having most glitch) and the variation with code may also be specified. Fixed reference DACs have their glitch specified for normal operation but multiplying DACs are often specified with a zero reference input, a "doublet" type of glitch but will give unfairly good results for glitches caused by logic skew. (Multiplying DACs are discussed in more detail later but for the benefit of those unfamiliar with the term it should be understood that all DACs have references—in the general case the reference has a fixed nominal value but a multiplying DAC is designed to use a reference which may be varied widely, so that in essence the output of a multiplying DAC is the product of the digital code and the analog reference input. For this reason the dynamic specifications of a multiplying DAC should include the bandwidth of the reference input for operation within specification.)

GLITCH

Glitch is measured under normal operating conditions.

In the case of a multiplying DAC the reference may be set to zero to measure "doublet" glitch, and to its maximum value to measure combined "doublet" and skew-related glitch.

CONVERTER TESTING

The methods and apparatus used to test converters are influenced by a number of factors relating to converter applications, the nature and speed of tests to be performed, and the skill of the test personnel involved. The relative importance of various converter performance specifications is dependent on each particular application, and the converter user is naturally interested in testing those parameters which significantly influence his system performance. Two typical applications illustrate how the application of the converter influences the importance of various performance parameters.

Differential linearity, fast settling time and a small switching transient (glitch) are generally of concern when DACs are used as CRT vector generators, since display quality depends critically on these parameters. Small errors in absolute calibration or zero are rarely of consequence, since they cause only small shifts of display size and position, which can be easily corrected by the operator.

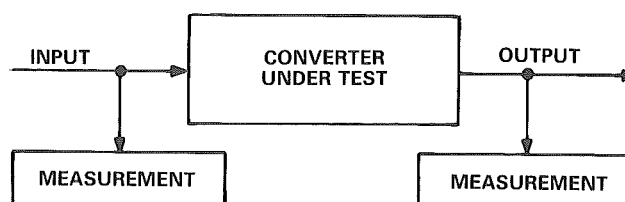
In contrast, a DAC used as a programmable stimulus generator in an automatic test system requires good absolute calibration and zero-stability but does not need fast dynamic response, transient-free switching, or exceptional differential linearity.

The configuration and degree of automation of converter test circuitry is influenced by the purpose of the test (e.g., engineering performance evaluation, incoming inspection, functional checks, etc.), the versatility, measurement speed, data-reduction, and display capability of the test equipment, and the skill required for its operation. Simple test fixtures designed to test a few converter parameters can be made easily and inexpensively but must usually be operated by relatively skilled people, and test data obtained from them must usually be processed before it is of use.

Versatile automatic testers are necessarily complex and costly. An automatic tester generally performs tests quickly, however, and can be operated by less-skilled people. High-resolution converters have large numbers of data points which must be examined to extract performance information. A 12-bit DAC, for example, has 2^{12} , or 4096, possible states. Fortunately, by knowing the type of converter errors or deviations from ideal performance that are commonly encountered, it is possible to devise tests which permit useful data to be gained from the investigation of far fewer than the 2^n states associated with an n-bit converter.

The general principle of testing a converter is to compare its input and output and see whether they agree and, if not, by how much.

GENERALIZED CONVERTER TEST



**WHILE THE DIGITAL READING IS GENERALLY TRIVIAL
THE ANALOG MEASUREMENT MAY BE VERY DEMANDING**

To measure the accuracy of a 12-bit converter to $\pm 1/4$ LSB by such methods a voltmeter with relative accuracy (linearity) to $\pm 0.001\%$ is required. Such an instrument costs several thousand dollars since increased accuracy costs much more than additional digits. To make such a measurement the user would either need to automate the test sequence or memorize the exact voltage for each of 2^n codes. It is easier to use a precision DAC as a reference and work from there by comparing analog and digital signals.

USING A REFERENCE DAC

An 8-bit measurement requires a 10-bit DAC.

An 10-bit measurement requires a 12-bit DAC.

An 12-bit measurement requires a 14-bit DAC.

An 14-bit measurement requires a 16-bit DAC.

An 16-bit measurement requires a 18-bit DAC.

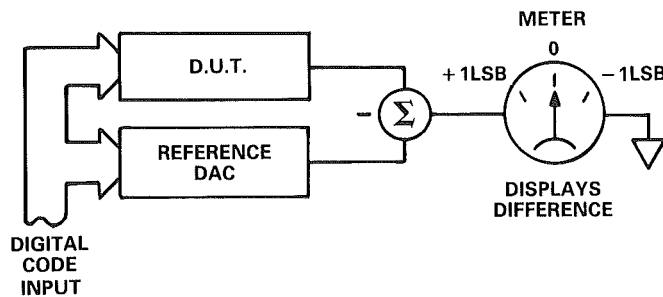
An 18-bit measurement requires a 20-bit DAC.

An 20-bit measurement requires the National Bureau of Standards.

DAC TEST CIRCUITRY

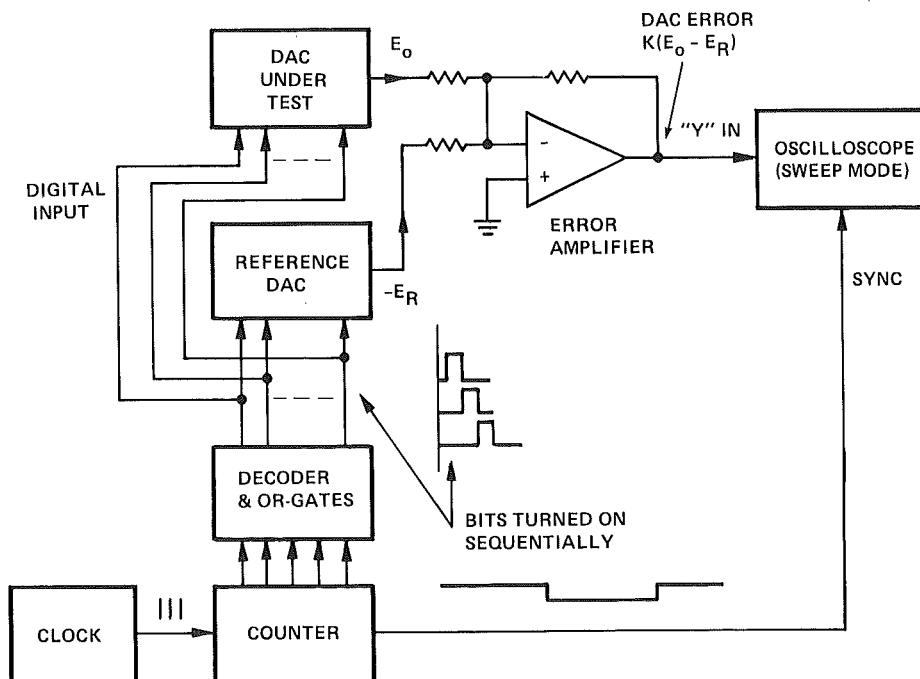
To test a DAC using a reference DAC uses the general principle shown in the diagram—the same code is applied to both DACs and the difference in their outputs, rather than their absolute values, is displayed.

DAC ERROR TESTING



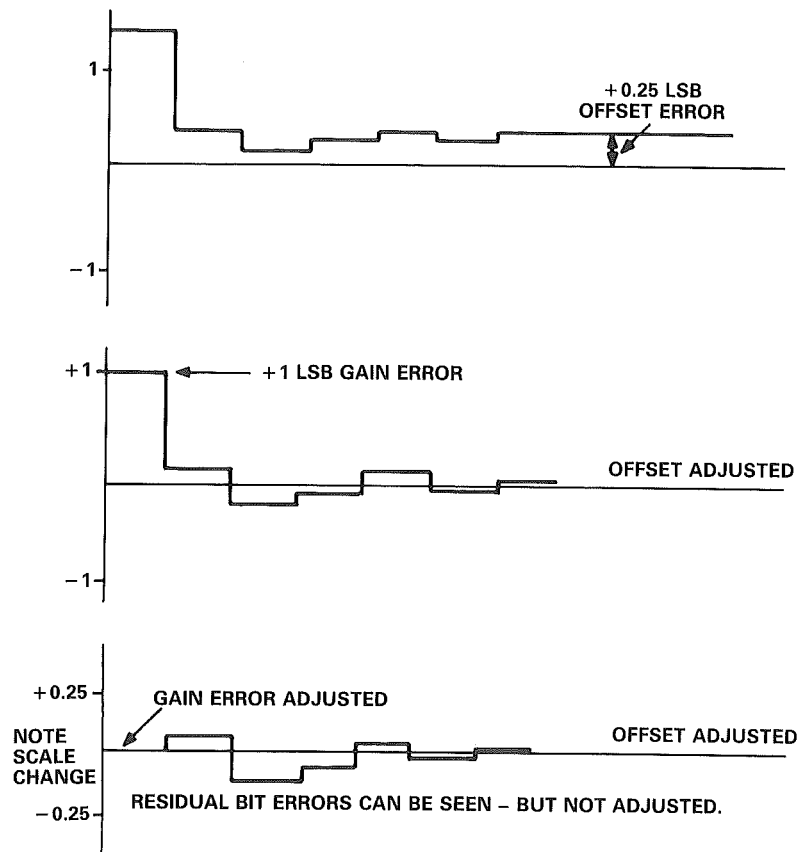
This is a very slow method and it may be sped up considerably if the code entry is automatic and only a few codes are used. In the "bit-scan" mode the same comparison is made between a DUT and a reference DAC but the difference output is applied to the vertical input of an oscilloscope. Only $n+2$ codes are applied to an n -bit DAC—these are all 1's, each bit set 1 in turn, starting with the MSB, while the rest are 0, and all 0's. The oscilloscope is in normal sweep mode and is triggered by the leading edge of the all 1's code—its time-base is adjusted so that the sweep displays all $n+2$ outputs.

DAC DYNAMIC TEST – BIT SCAN MODE



The oscilloscope displays the error in the DUT at each code in turn. With a code of all 0's it displays the offset error. With a code of all 1's it displays the sum of the gain and offset errors.

DAC BIT – SCAN TEST – DISPLAYS



As mentioned earlier the gain and offset errors of a DAC may be trimmed by the user and we may use this bit-scan display to do it. Order is important—the offset should be adjusted first (at the all 0's code) until it is zero. The error at the all 1's is now just a gain error and it should also be adjusted to zero—exactly as shown in the diagram.

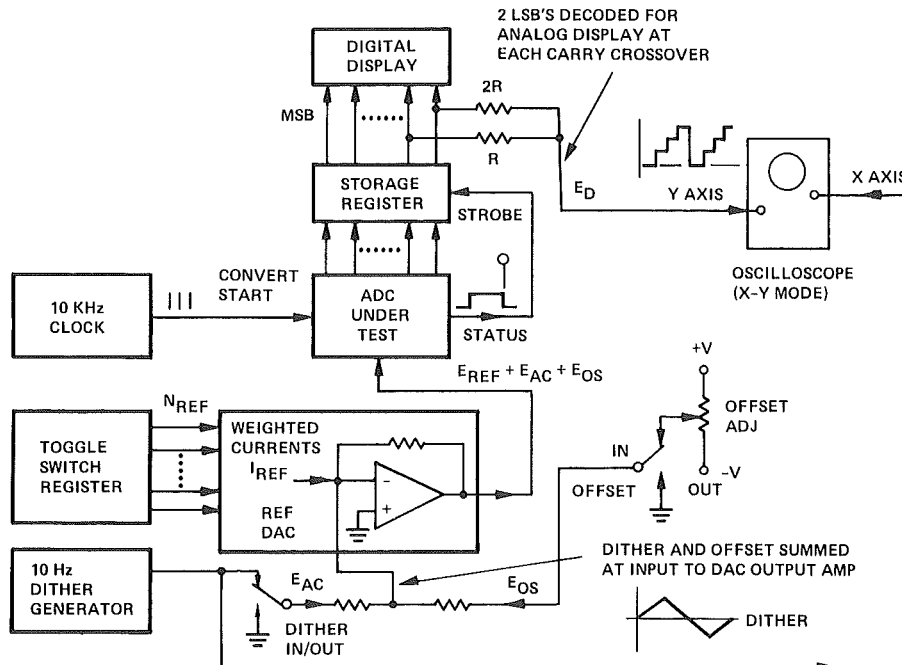
The remaining display shows the individual bit errors. Many DAC architectures define individual bits independently—this test is an excellent way of demonstrating bit errors in such DACs (DAC manufacturers use the same test to perform individual bit trims but these are a manufacturing step and cannot be performed by the user—all that he can do is to see how good or bad his DAC is).

The test may be used with DACs whose bits interact and is a convenient way of trimming offset and gain but it will not be so useful in assessing the linearity of such devices and more complex tests may be necessary to do so.

ADC TEST CIRCUITRY

An obvious simple ADC test is slowly to increase the analog input and observe the bit transitions as they occur, deriving the offset, gain, and linearity errors from them. It is a very slow and boring test to perform and a faster and equally efficient test is the "dynamic cross-plot" which, like the DAC bit-scan test, allows trimming as well as measurement of gain and offset errors.

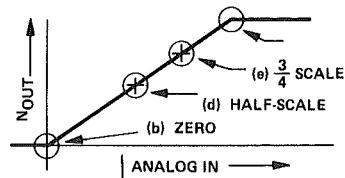
ADC DYNAMIC CROSS-PLOT TEST



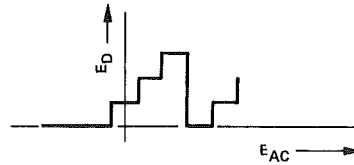
This test uses a reference DAC as well—the test set-up is shown in the diagram. It looks complex but may, in fact, be built in most laboratories in under an hour. A low frequency, low amplitude waveform is summed with the DAC output and applied to the ADC input—the waveform is also applied to the horizontal input of an oscilloscope working in the X-Y mode. This low frequency waveform is known as a "dither" waveform and may be sinusoidal or triangular but not discontinuous (sawtooth or square). Its period should be more than 1000 times the conversion time of the ADC so that input level changes during conversion are insignificant, and its amplitude should be 10-12 LSB. This low dither frequency limits the technique's usefulness in measurements on converters with conversion times much in excess of 1ms but is otherwise unimportant.

This dither waveform superimposed on the DAC output causes the ADC to perform a series of conversions 5 or 6 LSB above and below the DAC output. The two LSB of the ADC output are applied to a 2-bit DAC (a fancy name for 2 resistors!) via a latch (or directly if conversion is fast and the pauses between conversion are long enough) and thus to the oscilloscope screen. As the dither waveform rises and falls the ADC cycles through 10 to 12 adjacent codes and they appear on the oscilloscope as a series of 4-step staircases. By observing these staircases it is possible to measure, and adjust, the ADC's performance.

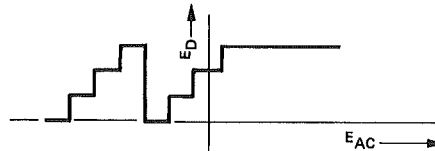
ADC DYNAMIC CROSS-PLOT WAVEFORMS



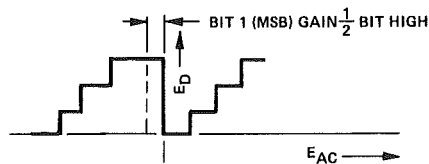
A. WAVEFORM LOCATION ON TRANSFER CURVE



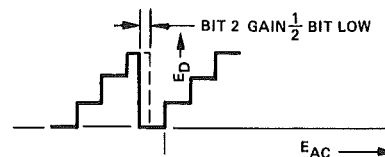
B. ZERO CALIBRATION



C. FULL-SCALE CALIBRATION



D. HALF-SCALE CARRY, BIT-1 GAIN HIGH



E. 3/4 SCALE CARRY-BIT 2 GAIN LOW

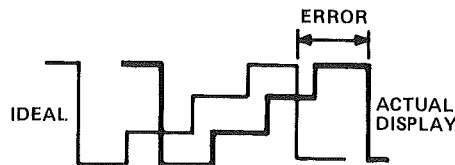
First of all the position of the undeflected oscilloscope beam is noted and the dither waveform turned on. The reference DAC is set up with 1 in the LSB and all 0's and the bottom step of the cross-plot waveform is adjusted to center on the Y-axis, as in B of the diagram, by adjusting the ADC offset. The reference DAC is then set to 0 in the LSB and all 1's, as in C of the diagram, and the ADC gain is adjusted so that the pen ultimate step is centered on the Y-axis.

Offset and gain are trimmed by this arrangement and other ADC measurements may now be made (in fact offset and gain should theoretically be trimmed with the reference DAC at all 0's and all 1's respectively and the codes used to introduce some interaction but they are used for two reasons—the interaction is insignificant, and if all 0's or all 1's were to be used the “step” involved would extend to minus infinity or plus infinity respectively and it would not be possible to “center” it).

Once the ADC under test is trimmed the cross-plot may be used to investigate DNL, INL, and noise. The DAC is set to the major bit transitions and the code widths in the staircase observed. If they are of very different widths the device has poor DNL; if one or more codes are not present the device has missing codes.

If the DAC is set to a particular code and the display shows clearly that, that code is not centered on the Y-axis then the device has an INL error of the amount that the code is displaced. If INL error is suspected of being so large that the “wrong” staircase is on screen then the codes should be increased in steps which are sufficiently small to see the correct staircase “walking away” across the screen.

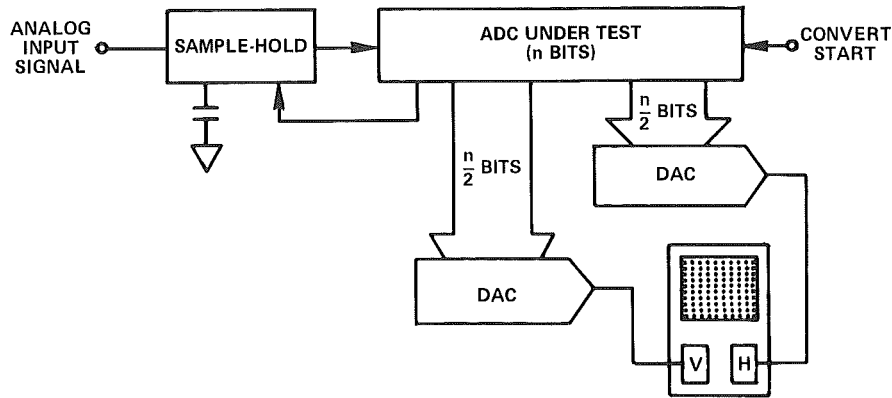
INTEGRAL NONLINEARITY IN AN ADC CAUSES THE DYNAMIC CROSS-PLOT DISPLAY TO BE DISPLACED FROM ITS IDEAL POSITION



Noise in the ADC manifests itself as fuzzy edges to the cross-steps. It may be measured in LSB or mV.

A simple test for missing codes under the traditional definition uses two $\frac{n}{2}$ -bit DACs to test an n-bit ADC. The ADC input is slowly increased at a rate of less than 0.1 LSB per conversion (so that at least 10 conversions occur for each LSB), the ADC outputs are fed half to each DAC (which output goes to which input, on which DAC, is unimportant), and the DAC outputs go to the X and Y inputs of a storage oscilloscope in the X-Y mode. Each code is a unique point in a matrix on the oscilloscope screen—it is easy to spot a missing point in a matrix, and if a point is missing a code is missing! (Uniform point brightness indicates good DNL, and wide variation of brightness indicates poor DNL).

ADC MISSING CODE TEST



This test is very simple to perform and interpret and gives a very quick and efficient screen for ADCs with missing codes.

DATA CONVERTER ARCHITECTURES

Data converters are relatively complex structures and may contain some or all of the following subsystems:

SUBSYSTEMS USED TO BUILD DATA CONVERTERS

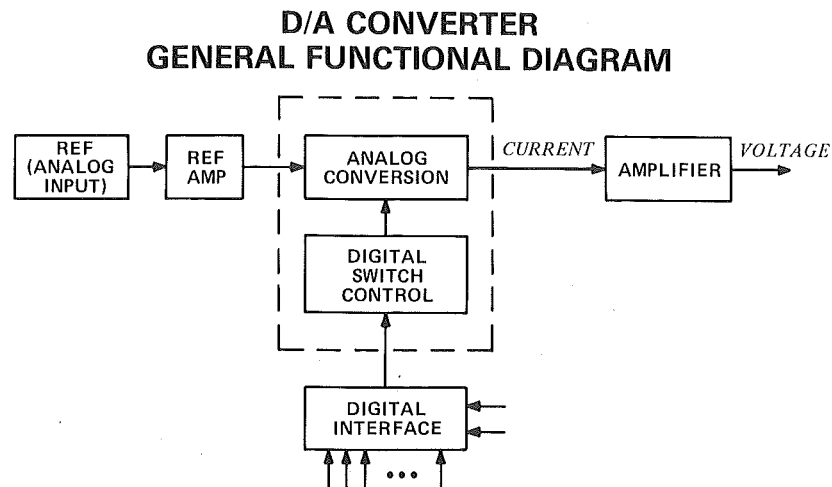
- Amplifiers and Comparators
- Voltage References
- Precision Resistors
- Current Sources
- Logic Circuitry
- Switches

THESE MAY BE MANUFACTURED WITH MANY DIFFERENT TECHNOLOGIES.

These may be made using many different structures and technologies. Many of the technologies used by Analog Devices to manufacture data converters are described in the section of the seminar devoted to process technology but it is not necessary to understand the technology to understand the architectures of different types of data converter—all that is necessary is to realize that few monolithic technologies are equally suitable for the manufacture of all these subsystems. Furthermore in real life, as opposed to theoretical designs, few subsystems are perfect and converter architectures will be chosen to minimize the effects of imperfections in the technologies used.

DAC ARCHITECTURES

A generalized DAC architecture is shown in the diagram.

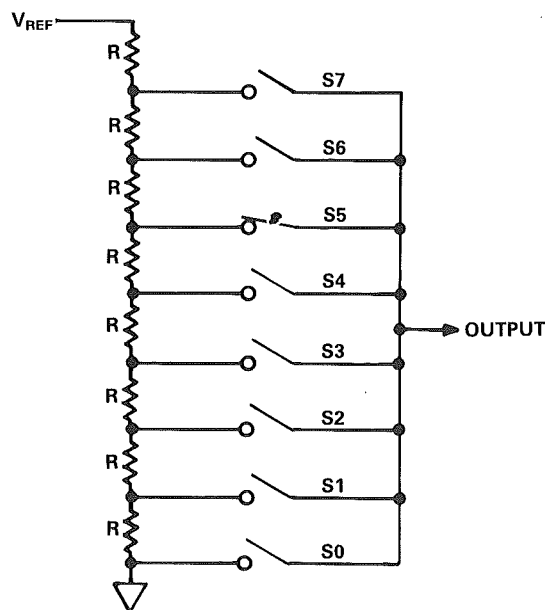


It consists of a core—which is essential to the DAC function—and a number of other subsystems which are convenient but not necessary. The core contains the basic data conversion and digital switch control which normally consists of resistors (or precision current sources) and switches which are generally transistors or MOS devices.

Other desirable subsystems include amplifiers, references, and digital subsystems of varying complexity. Which optional subsystems are included in a DAC depends on the function for which it is designed and the process which is used to manufacture it. This process is determined by the performance required of the core converter and we shall consider the various types of core before we consider their surrounding subsystems.

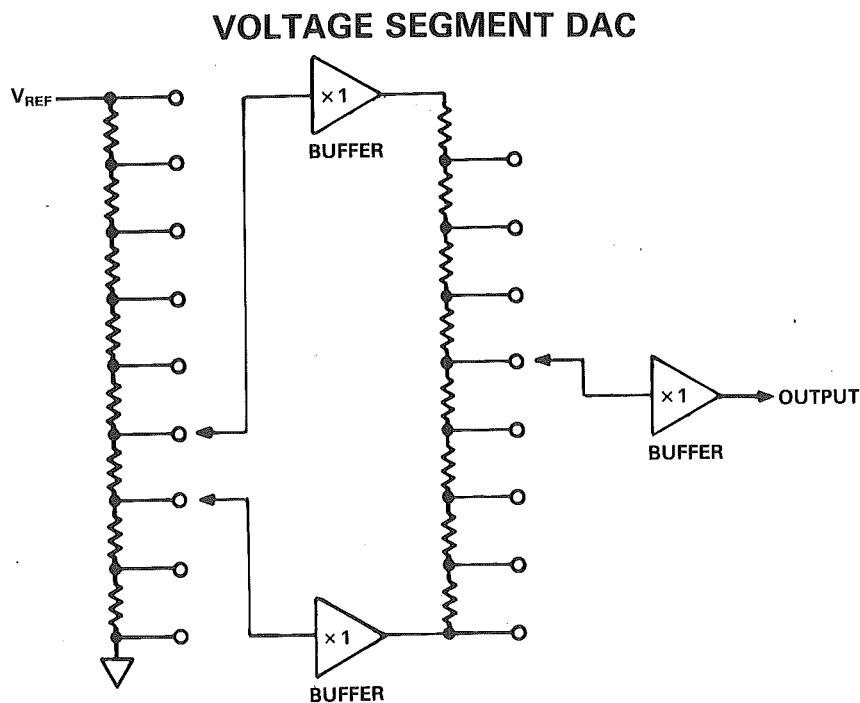
DAC CORES

The simplest DAC uses a Kelvin Divider—a simple chain of resistors with a switched tap.



Only one switch is closed at any time and the output voltage is the voltage at the point on the divider chain to which the output is switched. Such a DAC is very simple in concept, is monotonic even if the resistors are not well-matched, and (since all the voltages are always present) has potentially quite high speed, but does have certain drawbacks. First of all it has a high, and code variable, output impedance (although this problem can be minimized by the use of a buffer amplifier). Secondly an n -bit converter requires 2^n resistors which can result in a large structure, thirdly the switches used must operate with either side biased anywhere between ground and V_{REF} , and finally the logic requires extensive decoding.

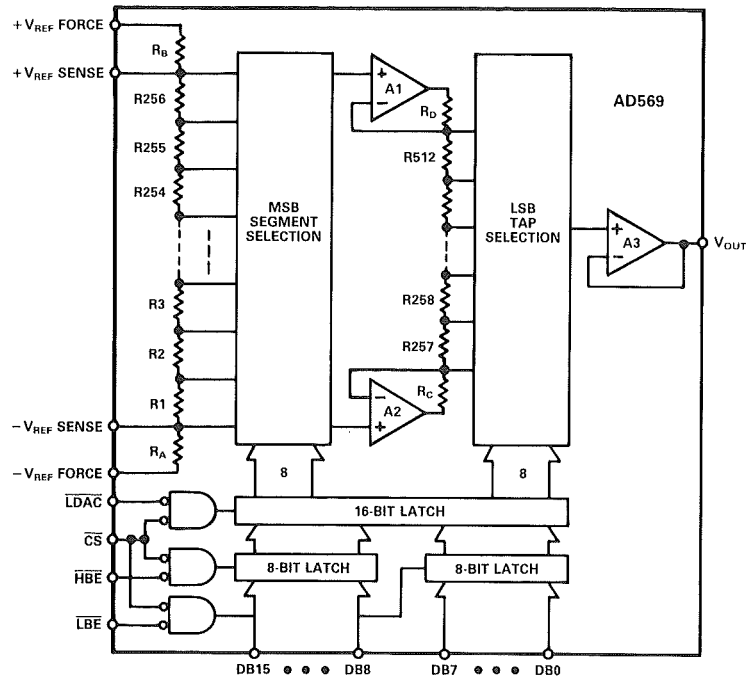
These disadvantages are such that the basic structure is very rarely used in real commercial DACs but extensions of the technique are quite useful since they retain the intrinsic monotonicity of the system while overcoming some of its limitations. The most common extension is the voltage segment DAC where there are two dividers.



The reference voltage is connected across the first divider and the second divider is connected between adjacent taps on the first divider—the particular pair being chosen by the more significant bits of input code. It is evident that if the second divider is connected directly to the first it will load it and reduce the voltage across the particular resistor it is connected to but this difficult can easily be overcome by putting unity gain buffers between the two if the whole DAC is to be monotonic either the buffer offsets must be well under an LSB or the switching scheme must be more complex and one buffer be connected only to odd taps and the other to even taps.

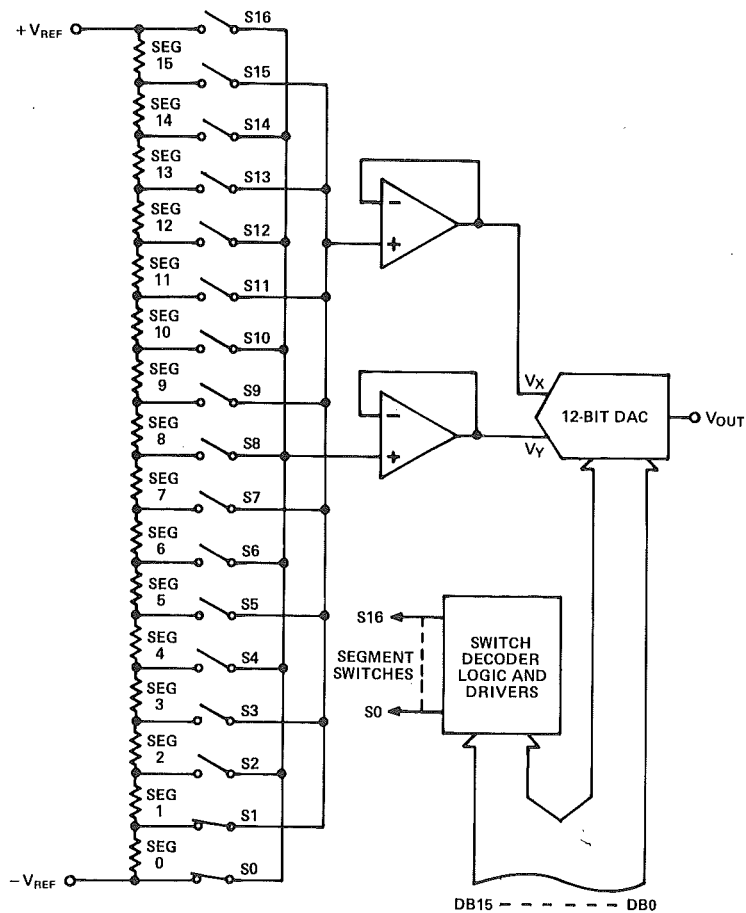
The result of this scheme is that the reference input is divided in turn by the first and second resistor chain and so an n -bit DAC may be made with two chains of $2^{(n/2)}$ resistors. Furthermore the DAC is intrinsically monotonic even if the resistors are grossly mismatched (gross resistor mismatches cause gross integral linearity errors and can even cause fairly major differential linearity errors but never of such a sense as to cause non-monotonicity). An example of such a DAC is the AD569 which uses BIMOS technology to make a 16-bit voltage segmented DAC.

AD569 FUNCTIONAL BLOCK DIAGRAM



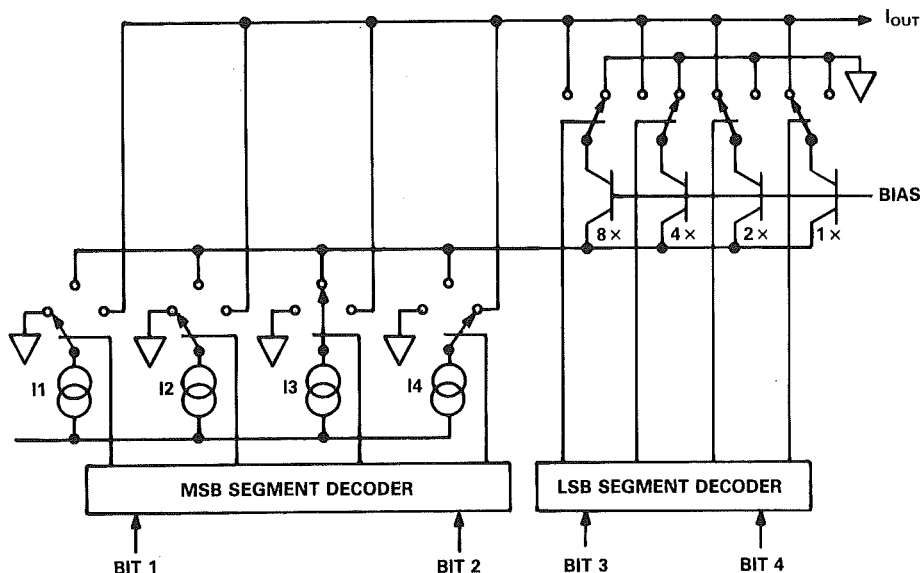
The AD569 uses untrimmed resistors (it is quite expensive to trim over 500 resistors per chip) and has INL of 13-bits while being monotonic to 16-bits—in the type of closed-loop application for which it is intended an INL of 13-bits is more than adequate.

The second half of a segmented DAC need not be another simple divider chain but may be some other type of DAC.



The principle of the segmented DAC may also be applied to current segmentation where instead of a resistor chain dividing a reference voltage a number of switches route currents.

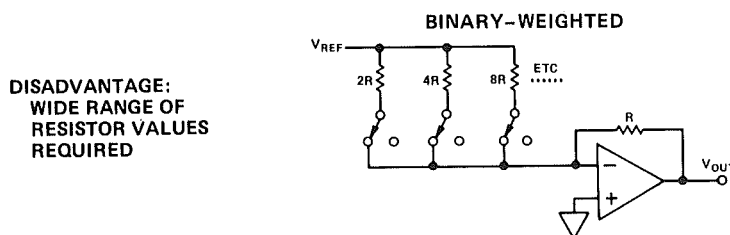
CURRENT SEGMENT 4-BIT DAC



In the DAC illustrated four switches route equal currents to ground, to the output, or to the output via a second 2-bit DAC. This second DAC divides its input current into four equal parts and routes each to the output or to ground. It is evident that such a system is the current equivalent of the voltage segmented DAC described above and is inherently monotonic: as the input code increases from zero at first I_1 , I_2 , I_3 , and I_4 all flow to ground, then increasing amounts of I_1 flow to the output while the rest flow to ground, then all I_1 flows to the output and increasing amounts of I_2 with it while the remainder flow to ground, and so on. Since any current connected to the output by increasing code is never disconnected while the code continues to increase the output is nonmonotonic.

Despite the simplicity of simple voltage- or current-divider structures the usual text-book illustration of a DAC does not use them. This is because the simple binary-weighted DAC has a major conceptual advantage—if driven by a simple binary number it does not require any decoding logic—each logic input drives a switch.

CONTINUOUS BINARY-WEIGHTED NETWORK

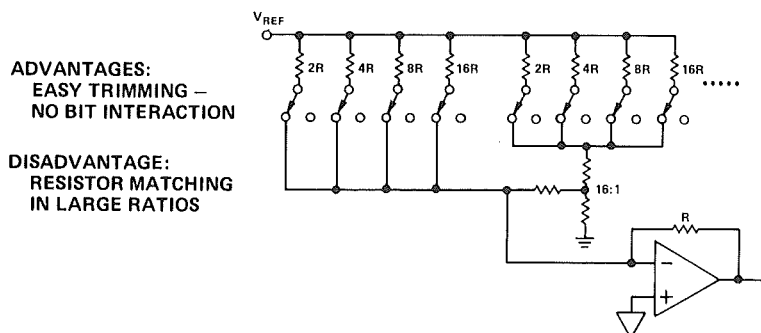


A series of resistors in the ratio $2R$, $4R$, $8R$, $16R$, etc., are connected to a reference voltage and switched to a current summing point at ground potential. The switches are controlled respectively by the MSB, next MSB, etc., and it is clear that twice as much current flows in each resistor as in its successor. Thus the total current at the summing point is set by the reference and the binary control inputs to the switches and we have a current output DAC. If a voltage output is needed an I-V converter formed by an operational amplifier and a resistor (of value R) is all that is necessary to provide it.

The binary weighted DAC is conceptually very simple and is therefore widely used for teaching purposes but it suffers from a serious drawback in real life: its range of resistors. An n -bit voltage output DAC of this type requires a resistor range of $2^n:1$ and in practical integrated circuits it is very difficult to make accurately matched resistors with a range of more than 30:1—giving a maximum resolution of only 5 bits.

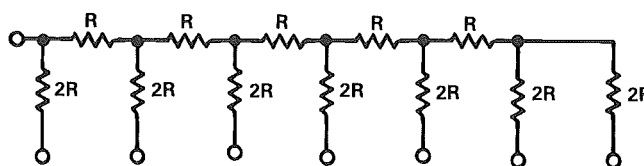
This limitation may be overcome by building two or more such binary weighted 4-bit DACs and attenuating their outputs accordingly. This technique is known as the successively weighted binary quad and has been used in a number of early DACs.

SUCCESSIVELY-WEIGHTED BINARY QUADS



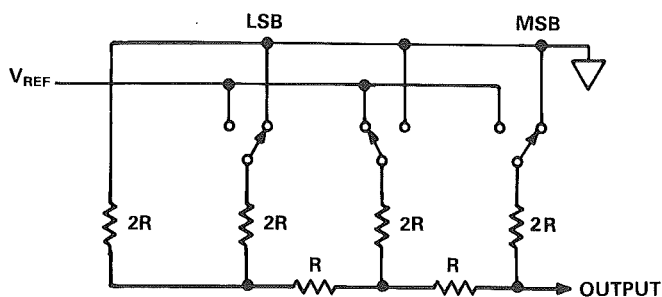
Its disadvantage is still the range of resistance necessary. Although it is quite possible to manufacture precision resistors having a ratio of 16:1 whose resistances match and track over temperature, it is not economical of chip (or thin-film substrate) area. For this reason most binary weighted DACs have been superseded by DACs built with ladder networks.

A SIX BIT R-2R LADDER NETWORK



A ladder network is an array of resistors, organized as shown in the diagram and having only two different values of resistors— R and $2R$. It is simple in concept and if the resistors match to 1 part in 2^n will give n -bit monotonicity but unlike the voltage dividing DACs described above, where the switches involved had only to switch voltages to high-impedance buffers, the ladder network is a current operated device and the switches used with it must be of sufficiently high performance as not to degrade the DAC performance. It may be used in two different configurations, each having its own particular benefits and disadvantages.

STANDARD 3-BIT DAC USING R-2R LADDER NETWORK

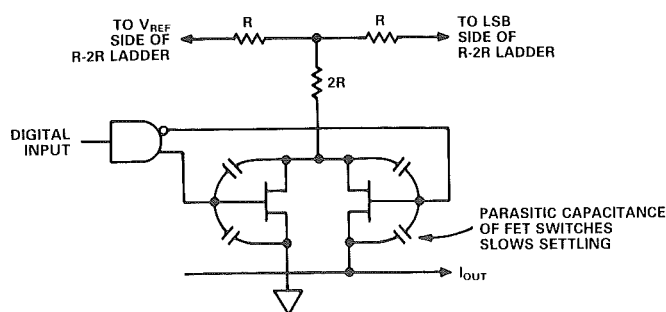


(OUTPUT IS A VOLTAGE BUT AS Z_{OUT} IS FIXED
[$=R$] THE DAC MAY ALSO BE USED AS A
CURRENT OUTPUT DAC INTO A LOW IMPEDANCE).

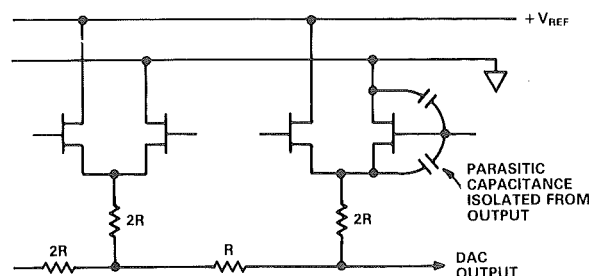
In the most common configuration, the "Standard R-2R DAC", the output is taken from the end of the network and each of the arms is switched between ground and a reference voltage. If the reference source impedance is assumed to be negligible it is evident that the impedance at the output terminal is R and does not change with code. It is slightly less evident but may quite simply be calculated from first principles that the output voltage is the product of the reference voltage and the normalized binary number applied to the switches—alternatively if the output is connected to a point at ground potential the output current will be equal to the reference voltage times the normalized binary code divided by R .

We thus see that the standard configuration offers voltage or current output and a constant output impedance. The manufacturing resistor trim algorithm is simple and also, since any stray capacity between the digital control lines and the switches goes to low impedance reference or ground lines, any charge injected into the DAC during digital transitions flows quickly to ground and so the glitch impulse is low and the settling time is fast. Nevertheless the configuration does have its drawbacks.

DAC ARCHITECTURE AFFECTS GLITCH



"NORMAL" CMOS DAC STRUCTURE

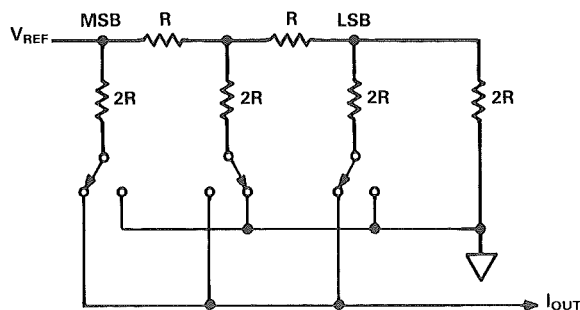


"BACKWARDS" DAC STRUCTURE OF AD7572 FACILITATES FAST SETTLING

The input impedance at the reference input depends on the digital code applied to the switches. This should not be a major problem but it does necessitate low impedance references and connections and prevents trimming of the DAC gain by means of a small trimmer in series with the reference input (gain must either be trimmed by trimming the value of the reference voltage or by altering gain elsewhere in the system). The other problem is a technological one but can be quite serious: as the switches change state the end of the resistance arms go from ground potential to V_{REF} . This can have two possible effects—modulation of switch resistance and modulation of the ladder resistance itself. If the ladder is made with diffused resistors the resistance value may depend on the bias between the resistor and the diffusion well in which it is sited and changes in bias will cause changes of resistance and so affect the DAC accuracy.

Modulation of switch resistance is more serious. There is no difficulty in replacing diffused resistors with thin film ones which do not have the resistance modulation problem but there is no substitute for semiconductor switches (DACs have been built with relays but this paper describes reality, not ivory towers). Again, this problem does not affect the performance of available DACs, it affects the type of DAC which is available. Many older MOS and CMOS processes suffered from switch channel modulation sufficiently to limit the specifications, or the range of available reference voltages, of DACs using this architecture. Modern processes can be made more tolerant of switch channel bias and standard DAC performance can be improved, but even so the possible reference voltage with this architecture is much lower than with an inverted R-2R DAC and the range of available products is smaller.

INVERTED 3-BIT DAC USING R-2R LADDER NETWORK



(OUTPUT IS A CURRENT AND OUTPUT IMPEDANCE VARIES WITH CODE.)

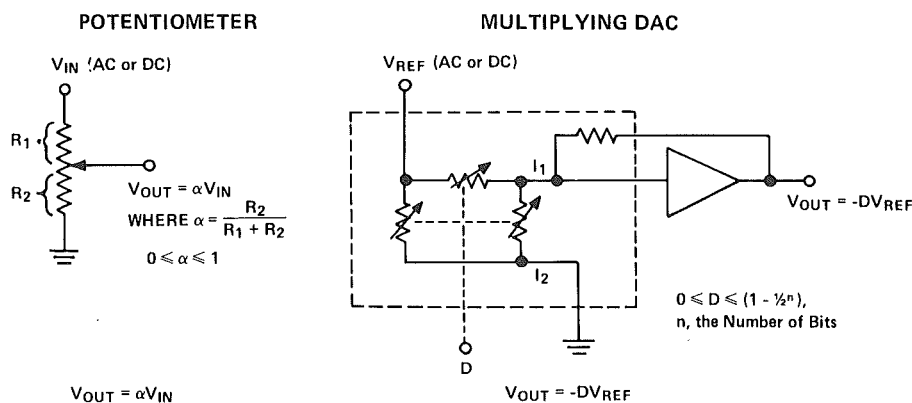
In the “Inverted R-2R DAC” the connections for the reference and output are reversed from the previous case. This means that the reference input impedance is code-invariant and so the gain of this DAC may be trimmed with a low value potentiometer in series with the reference input. The output impedance, though, now varies with code and the output can only be a current flowing to a virtual ground point. This code variable output impedance make frequency compensation of an output I-V converter more difficult since it is driven by a varying source impedance. Furthermore the capacity of the output line is much larger (and code-variable) because of the parasitic capacity of the FET switches, which degrades the high-frequency performance of such a DAC.

The greater charge injection in the output line during code changes, as a consequence of charge being coupled from the switches into the output line rather than into low impedance ground and reference lines, also reduces performance by increasing glitch energy. On the other hand the switches are always very near ground potential, whatever the value of reference, so switch resistance modulation is not a problem, and the reference may in many cases exceed the switch supply. Indeed if the switches are MOS devices, which can conduct in either direction, the reference may have either polarity and may even be ac.

All DACs have outputs which are the product of their reference and the normalized digital code applied to them but DACs which will function correctly when their reference is varied over a wide range (which must include zero but need not necessarily be bipolar) are known as “multiplying DACs” or MDACs. CMOS DACs with inverted R-2R ladder networks are particularly good MDACs since their reference may be bipolar and, if they are operated in the bipolar mode, they are capable of four quadrant multiplication. The basic MDAC may be thought of as a digitally-controlled potentiometer.

MULTIPLYING D/A CONVERTERS

A multiplying DAC is nothing more than a digitally controlled potentiometer (attenuator).

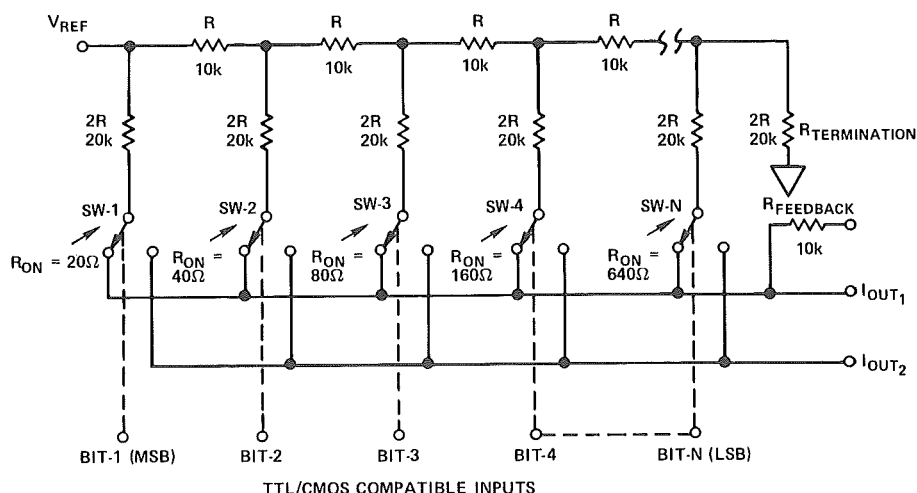


The output of such a multiplying DAC is the product of two variables, one digital (the input code) and one analog (the reference). They may therefore be used as computational devices, digitally-controlled attenuators, and digitally programmable resistors. There are even special MDACs, called “LOGDACs”, which contain very high resolution MDACs and decoding circuitry arranged so that each LSB change in the digital input produces a fixed percentage (decibel) change in the attenuation from reference input to output, unlike conventional MDACs where each LSB change produces an equal output increment. The difference may be considered equivalent to the difference between linear and logarithmic potentiometers.

It is important to realize that although MDACs are widely used as digitally controlled potentiometers and hybrid analog/digital computing devices they are widely used, with external fixed references, in normal DAC applications.

The final drawback of the inverted R-2R DAC is apparent only to the manufacturer—the trimming algorithm is considerably more complex than that of the standard part. With the use of computers during automated laser trim operations this is not a very serious problem. Both standard and inverted ladder networks may quite readily be trimmed for 14-bit INL and DNL over temperature. Above 14-bits trimming ladder networks becomes more demanding and although it is possible to manufacture 14-bit ladder networks which retain 16-bit performance over their full operating temperature range it is more usual to use mixed techniques—generally segmented plus ladder—at higher resolutions.

AD754X FAMILY LADDER NETWORK SWITCHES ARE LARGER TOWARDS THE MSB



Each switch in a ladder network carries twice the current of its successor. Therefore it is customary to scale the switches, making the ones nearer the MSB end of the DAC larger, although scaling rarely continues for more than five or six bits.

AUXILIARY DAC SYSTEMS

In addition to the cores which we have discussed in some detail DACs may contain a number of different subsystems. They include amplifiers, references, and more or less complex digital systems.

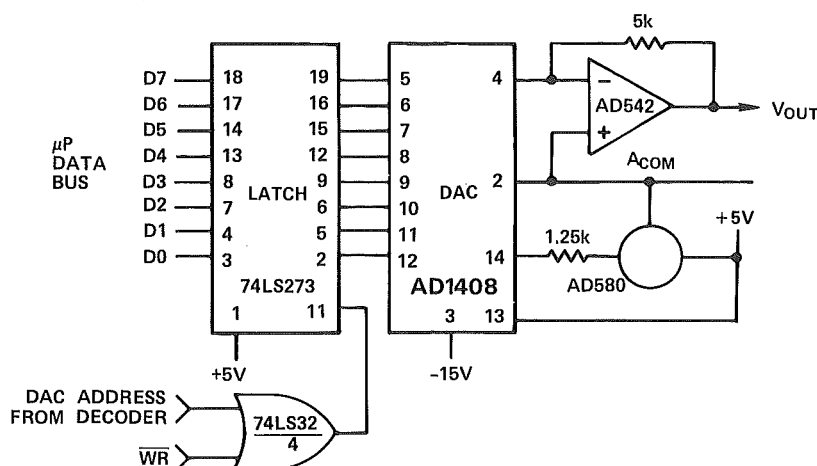
Amplifiers include buffer amplifiers for references, buffer amplifiers for segmented DACs (as described above), and operational amplifiers which function as I-V converters to convert current output DACs to voltage output DACs. In general the output amplifiers of DACs are not capable of driving large output currents—this is not because the design of such amplifiers is particularly difficult but because the increased dissipation due to a large output current will cause temperature gradients on the chip which will cause temperature related mismatch of the resistors in the DAC and thus degrade its accuracy. Where large output currents are required it is generally better to use a separate power output stage (which may often be a single power transistor within the DAC's output feedback loop).

All DACs require a reference—a source of voltage or current which defines the DAC output scaling. MDACs, as explained above, have an output which is the product of the reference input and the digital code but all other types of DAC require a fixed reference (possibly with provision for slight variation in order to provide gain trim). This fixed reference is frequently provided within the DAC itself.

Whether or not a DAC contains an internal reference depends on two things. If a DAC is required to function in a potentiometric manner together with other circuitry in a system it cannot use an internal reference. On the other hand many DACs would be improved by an internal reference but do not have one because the particular integrated circuit technology used to manufacture them is incapable of making a reference of adequate performance—this is discussed in more detail in the section of the seminar dealing with processes.

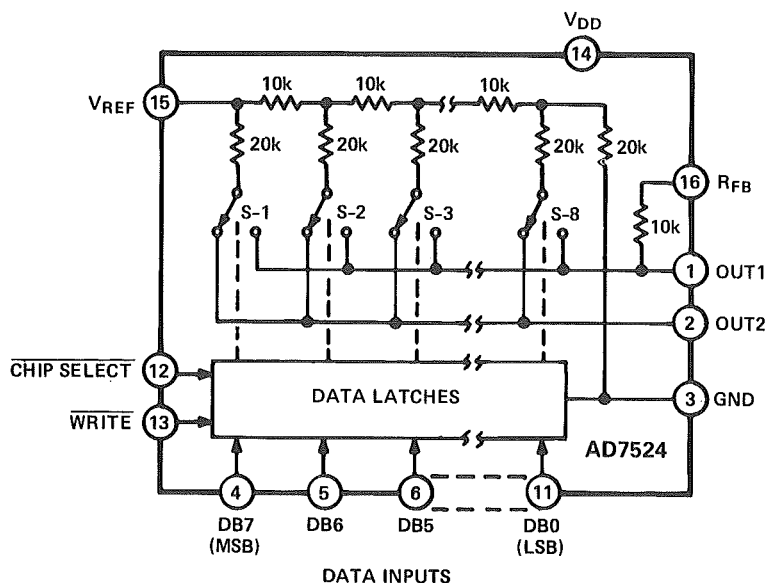
Finally all DACs contain digital circuitry. At its simplest this is merely a number of level-shifters to allow standard logic (TTL, CMOS, ECL, etc) to drive the DAC switches but today few DACs are so simple—most contain latches and many contain additional circuitry to enable them to interface to microprocessors with minimal additional componentry. Such circuitry has evolved slowly.

μP INTERFACE TO EARLY 8-BIT DAC



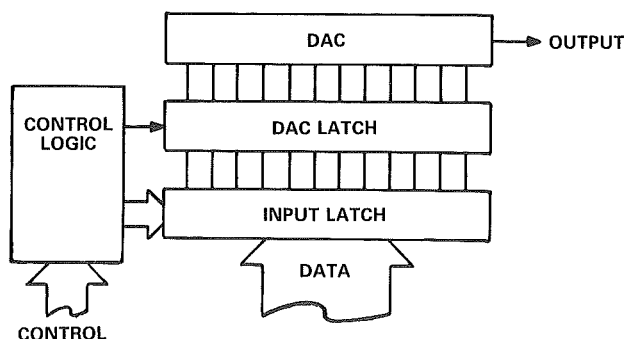
The earliest monolithic DACs contained a DAC core with TTL or CMOS interface levels and required that data be applied continuously to the inputs. Practical applications of such DACs, therefore, required the use of external latches to hold valid data and DAC manufacturers quite quickly included such latches in their products.

EARLY CMOS DAC WITH INTEGRATED LATCH



While this worked well for n -bit DACs working from M -bit data buses where $M \geq n$, extra circuitry was still necessary where high definition DACs were to be loaded from narrow buses.

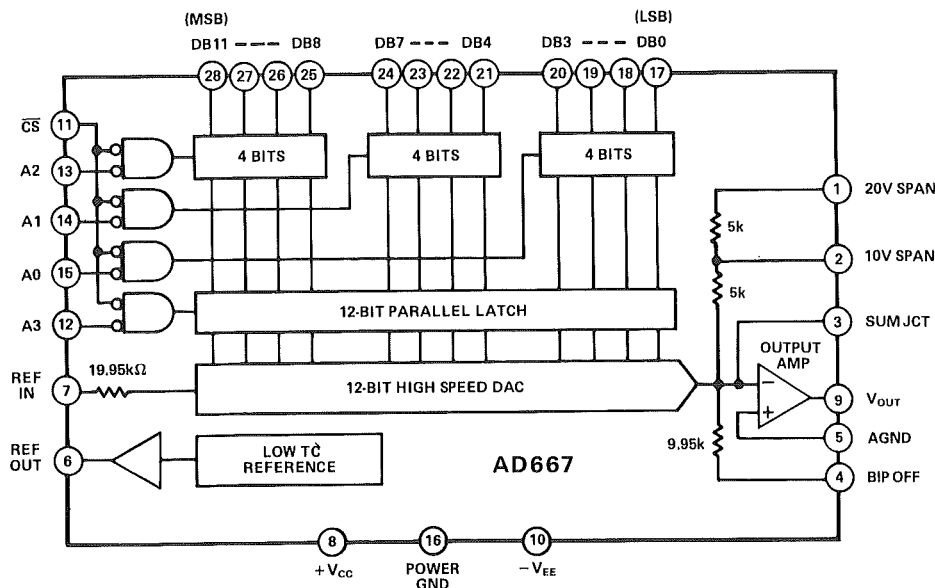
A DOUBLE-BUFFERED DAC



This led to the introduction of the “double-buffered” DAC which contains two banks of latches—the first rank of “input” latches which acquire data from external sources and the second rank of “output” latches

which take their data from the input latches and present it to the DAC. The important feature of the double-buffered DAC is that the input latches may be loaded piecemeal without affecting the DAC output—only when the second rank latches are loaded (which need not occur until sometime after the data in the input latches is complete) does the DAC output change. The structure of the input latches of DACs have been made to load from 1-bit, 4-bit, or 8-bit data buses, and there are DACs containing shift registers, FIFOs, and RAM. The important principle is that there is an intermediate store where data is held prior to being presented to the DAC to update the output.

AD667 FUNCTIONAL BLOCK DIAGRAM



The AD667 is an excellent example of a DAC containing most of the auxiliary subsystems mentioned in this section. It is a 12-bit voltage output DAC containing its own reference, buffer, and output amplifier and having a double-buffered data structure which allows it to be loaded from microprocessors having 4-bit, 8-bit, 12-bit or 16-bit data buses.

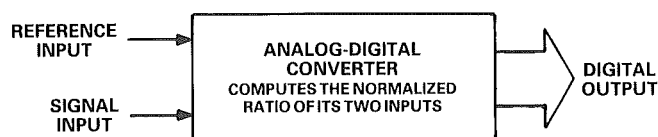
AD667 FEATURES

- Complete 12-Bit D/A Function
- Double-Buffered Latch
- On Chip Output Amplifier
- High Stability Buried Zener Reference
- Single Chip Construction
- Monotonicity Guaranteed Over Temperature
- Linearity Guaranteed Over Temperature: 1/2LSB max
- Settling Time: 4μs max to 0.01%
- Low Power: 300mW Including Reference
- TTL/5V CMOS Compatible Logic Inputs

ADC ARCHITECTURES

There are innumerable ways of obtaining a digital representation of an analog quantity, which is what an analog-digital converter does, but all of them involve comparing the input to the converter with a reference in some way that allows the computation of the ratio of the two quantities.

BASIC ANALOG-DIGITAL CONVERTER FUNCTION



This section of the seminar considers a number of the more common systems for performing this function. They are the flash converter, and its slower, but more economical and higher resolution, variant the half-flash

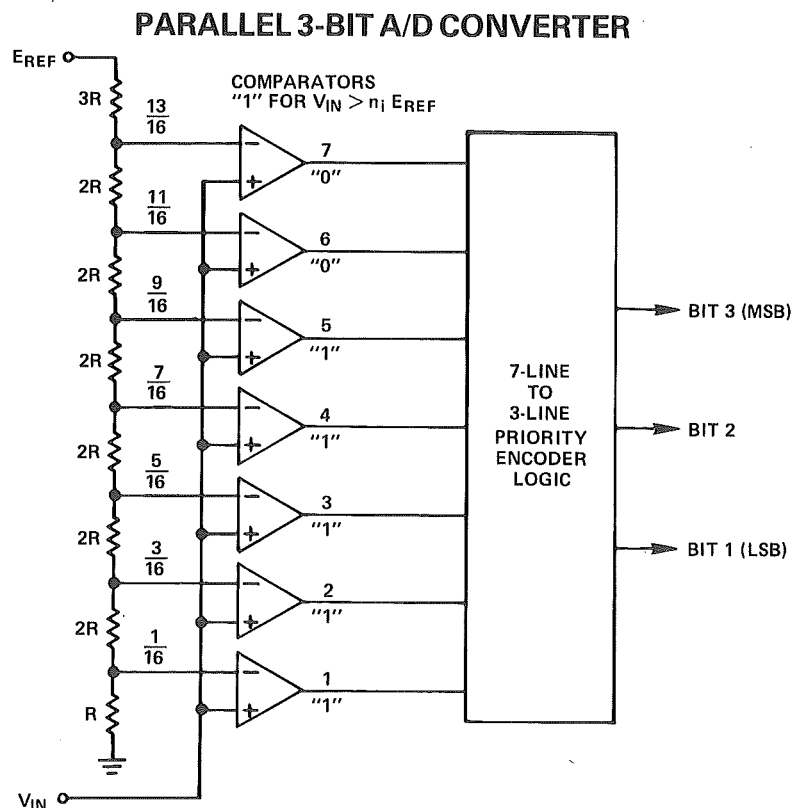
converter, the various types of integrating ADC and, closely related, the two types of voltage-to-frequency converter, and the two types of ADC incorporating DACs—the tracking and the successive approximation converters. Each type will be considered in turn and we shall also consider an ADC system, the floating-point ADC, which may be used to increase the dynamic range of an ADC.

ANALOG-DIGITAL CONVERTER TYPES

Flash and Half-Flash	• Fast • Power-Hungry • Lower Resolution
Integrating	• High Resolution • Inexpensive • Slow • Noise Immune
Voltage-Frequency Converters (VFCs)	• Fast Response But Slow Conversion • Serial • High Resolution • Noise Immune
Tracking (Counter/Comparator)	• Fast in Track • Slow in Multi-Channel Applications • Susceptible to Noise
Successive Approximation	• Quite Fast • Flexible and Versatile • Accurate
Floating-Point	• Uses Any Type of ADC • Wide Dynamic Range • Complex

FLASH & HALF-FLASH CONVERTERS

The flash or parallel ADC is the fastest variety of solid-state ADC. An n -bit flash converter consists of 2^n resistors connected in series with the reference voltage applied across the chain (all the resistors are equal except the top one, which is 1.5 times the value of the rest, and the bottom one, which is 0.5 times). In a unipolar flash converter the bottom of the chain is grounded but in a bipolar one the top and bottom of the chain are connected to positive and negative references respectively. A comparator is connected to each resistor junction, which requires 2^{N-1} comparators, and the signal input is connected to the other input of all of them. The comparator outputs are connected to the inputs of a logic circuit called a priority encoder.



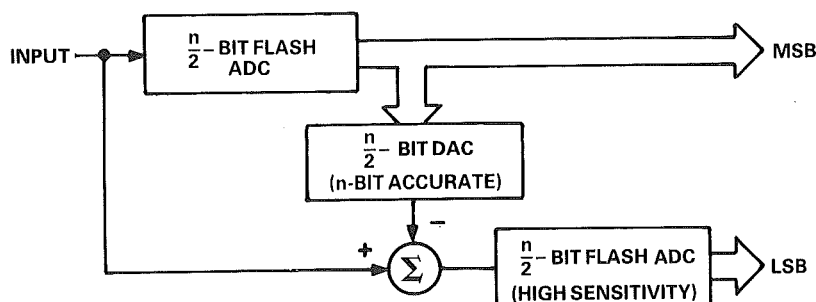
It is evident that the voltages on the comparator inputs, reading from the bottom of the chain, are 0.5 LSB, 1.5 LSB, 2.5 LSB, etc., above ground or the negative reference, until the highest is 1.5 LSB below the reference voltage, and that all the comparators whose reference inputs are lower than the signal input will have one output state while all the rest will have the other. The priority encoder has as its output the number of the highest comparator on the chain which has an input larger than its reference and this, of course, corresponds to the magnitude of the input voltage and therefore the system performs an analog-digital conversion. Since all the comparators are working simultaneously the time between a signal being present on the input and the corresponding digital code becoming available at the output is the sum of the comparator propagation delay and the priority encoder propagation delay—both may be very short and flash converters of 6-bit and 8-bit resolution are available with conversion times well under 5ns—corresponding to conversion rates of hundreds of megasamples/second. Some flash converters contain latches in either the comparators or the logic circuitry to lock the output state to prevent it from changing while a reading is taken.

The basic concept of a flash converter is simple and, providing the comparator offsets are all less than 0.5 LSB, it has no missing codes (although its INL will depend on the quality of its resistor matching—and a flash converter could be made deliberately nonlinear if such a feature was required for a particular application). But despite its speed and lack of missing codes the flash converter has three major disadvantages: its size, its input impedance, and its power consumption. An n -bit flash converter contains $2^n - 1$ comparators (if it has an extra comparator to indicate overrange, as many flash converters do, it contains 2^n). Thus an 8-bit one contains 255 comparators and a 10-bit one 1023. Such large numbers of comparators, which are precision analog circuits and not simple memory cells, occupy quite large chip areas, even with modern chip component densities, and converters of these resolutions are correspondingly expensive. Since the input is applied to every comparator the input impedance in a converter containing a large number of comparators will be the parallel combination of all their input impedances—which makes high demands on the buffer circuit or SHA driving the converter and may make the analog bandwidth substantially less than the Nyquist frequency implied by the conversion rate. Furthermore, in order to achieve adequate speed the comparators must consume relatively high currents and therefore high resolution, high speed, flash converters consume considerable amounts of power.

In many demanding applications users are prepared to pay these cost and power penalties but at present 12-bit resolution is impracticable and 16-bit is impossible.

For higher resolutions the “half-flash” or “subrange” converter offers a development of the technique which gives higher resolution and lower power and system complexity at the cost of longer conversion times and greater conceptual complexity. A half-flash converter is illustrated in the diagram. It consists of two $n/2$ -bit flash converters, a $n/2$ -bit DAC, and a subtractor circuit.

BASIC HALF-FLASH CONVERTER (FEED FORWARD ARCHITECTURE)

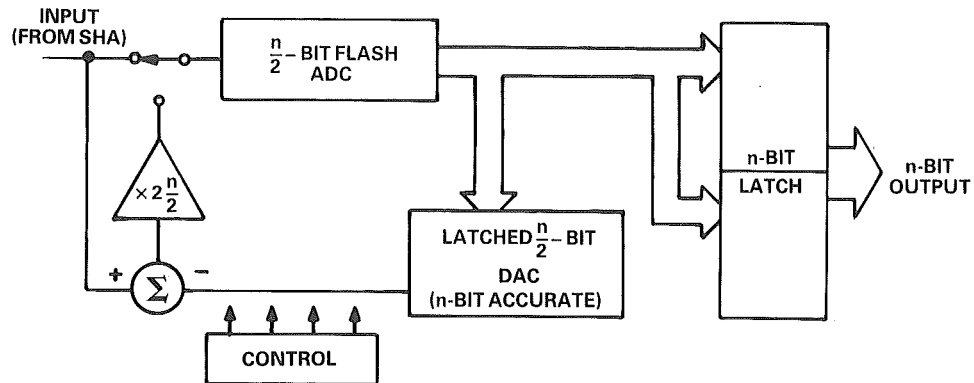


The input signal is applied to one of the $n/2$ -bit flash converters (which must be accurate to better than n -bits, despite its relatively low resolution) which provides as its output the $n/2$ most significant bits of the conversion. These $n/2$ bits of data go both to the output and to an $n/2$ -bit DAC (again n -bits accurate) which gives an analog output equal to the $n/2$ MSBs of the input, which are subtracted from it, leaving a signal containing only the $n/2$ LSBs which is applied to a second flash converter of $n/2$ -bits resolution. Either this second converter is $2^{(n/2)}$ times as sensitive as the first, or the signal is amplified by $2^{(n/2)}$ before being applied the converter which provides the LSBs for the output.

It is evident that although the MSBs of the conversion are available as quickly as with a normal flash converter the LSBs are additionally delayed by the propagation delays of the DAC, the subtractor, and the second flash ADC (and the amplifier if one is used). The technique is therefore somewhere between two and three times slower than a simple flash converter but where a simple N-bit converter contains 2^n comparators this one contains only $2^{(n/2+1)}$ comparators plus, of course, a DAC and a subtractor. A half-flash converter is therefore considerably simpler, uses fewer components, and is less power-hungry than a full flash converter.

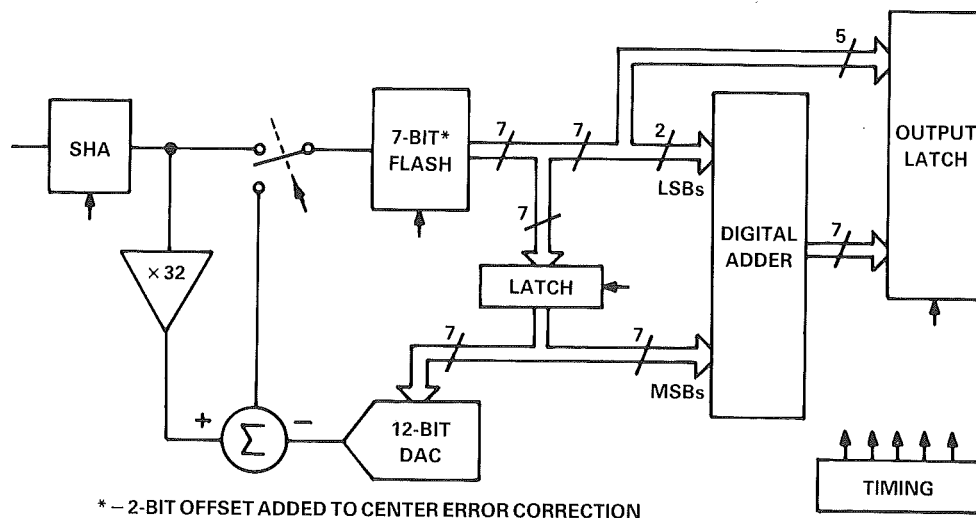
The feedback subrange ADC uses even fewer components, since it contains only one $n/2$ -bit flash ADC, but is still more complex in theory. The MSBs from the first conversion are latched to the DAC, and the DAC output subtracted from the original signal and the result amplified as before but in this converter the same $N/2$ -bit ADC is used again for the second conversion. This requires some extra signal switching, timing, and data latches.

FEEDBACK HALF-FLASH ADC



Even better performance can be achieved by using this basic technique with an extra bit of resolution in the flash converter and using a digital adder for error correction. The AD375 and HAS1201 are two 1 μ s 12-bit ADCs using this architecture.

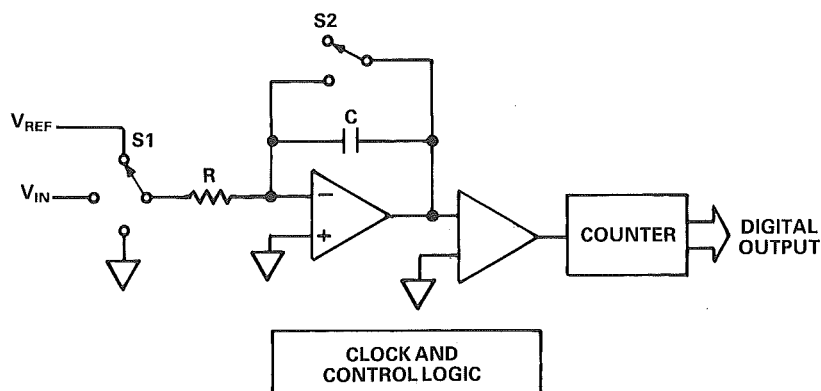
AD375 – 1 μ s 12-BIT FEEDBACK SUBRANGE ADC WITH DIGITAL ERROR CORRECTION



INTEGRATING ADCs

Integrating analog-digital converters are slow. They are, however, inexpensive, simple in concept, relatively immune to noise, and can be made to consume very little power. They are the form of ADC generally used in digital voltmeters and all applications where slow transducers (thermocouples, etc.) drive a display.

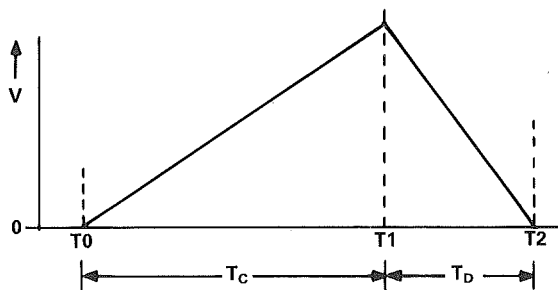
BASIC INTEGRATING ADC



The basic system consists of an integrator (an operational amplifier and a capacitor C), a comparator, an up/down counter, and a clock and control logic.

The integrator is held discharged. At the start of a conversion it charges from the unknown input voltage, through the resistor R, for a fixed time determined by the clock and the counter. It is then discharged by applying a known reference voltage to the resistor and the time that it takes to discharge is read by the counter. Since the integrating capacitor, the resistor, and the clock frequency are unchanged during the charge and discharge cycles it is evident that the ratio of the charge and discharge times is equal to the ratio of the reference and unknown input voltages. The conversion accuracy is unaffected by the absolute values of R, C, or F and any noise on the input signal is integrated for the whole signal sampling period.

OPERATION OF INTEGRATING ADC



THE INTEGRATOR CHARGES FOR FIXED TIME T_C WITH V_{IN} . THE DISCHARGE TIME WITH A FIXED REFERENCE ($= V_{REF}$) INPUT IS THEN MEASURED (T_D).

$$\text{RATIO } \frac{|V_{IN}|}{|V_{REF}|} = \frac{T_D}{T_C}$$

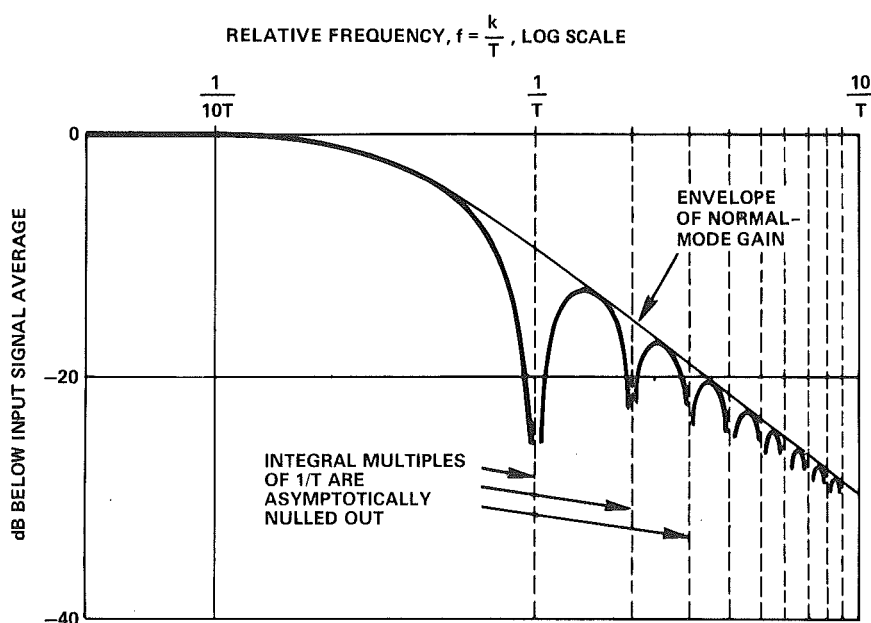
Leakage and offset in the integrator and hysteresis and offset in the comparator do affect the accuracy, though, and most practical integrating ADCs perform a conversion with the integrator input grounded to compute the errors due to these effects and then subtract these errors from the result of the actual conversion. Such complexity requires more logic and switching circuitry, although no more analog components, and so integrating ADCs are usually made with an IC process which is suitable for high density low power logic—CMOS. The use of CMOS, however, sets limits on the noise and accuracy possible from the integrator and comparator, and prevents the integration of very accurate references on the chip. The most accurate integrating converters, therefore, use external operational amplifiers and references and are capable of better than 18-bit accuracy, but even with CMOS amplifiers and references built on a CMOS chip 12- to 13-bit performance is possible.

The nature of an integrating ADC is such that its DNL is very good and it does not suffer from missing codes—but its INL and gain accuracy may be far worse than its resolution, leading to the well-known “digital voltmeter effect” where naive engineers infer that a reading is accurate to 2.5 digits when quite simple cross-checks will show that it is only good to 5% but with 2.5 digit resolution.

The basic integrating ADC is unipolar. Where bipolar operation is required it is generally achieved by an extra comparator switching an inverting/noninverting buffer prior to the unipolar ADC, giving a signed rather than true bipolar response. This is, in fact, the most convenient form of data for DVMs, so true bipolar outputs are rarely available from integrating ADCs.

The sampling period, T , of an integrating ADC is fixed. If the frequency of any ripple on the input to the converter is an integral multiple of $1/T$ a whole number of cycles will occur during each integration and the net contribution to the charge in the integrator due to ripple will be zero. Thus an integrating ADC has near infinite rejection of input frequencies n/T where n is an integer, and with suitable choice of T (say 16.667ms in the USA and wherever else line frequency is 60Hz and 20ms in Europe and wherever else it is 50Hz) line ripple on the signal input will be disregarded. This is another reason why integrating DVMs are popular—but it is also a strong argument against Europeans buying DVMs in the USA, where they are less expensive, for use in Europe, where the line frequency is different.*

NORMAL MODE RESPONSE OF DUAL-RAMP ADC

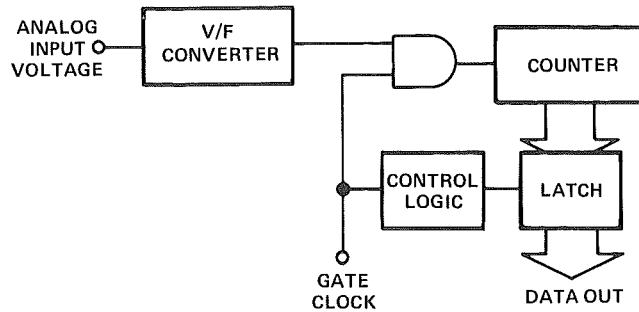


VOLTAGE-TO-FREQUENCY CONVERTERS

An ADC technique closely related to the integrating ADC and having several of its advantages and disadvantages uses a voltage-to-frequency converter (VFC). A VFC is an oscillator whose output frequency is proportional to an input voltage. Simple VFCs may be made with 555 IC timers or even with neon lamps but their nonlinearity is very bad—purpose-built VFCs, on the other hand, are available with nonlinearity of 0.002% or better. A VFC is not, by itself, an ADC but it forms one with the addition of a counter and a timed gating circuit.

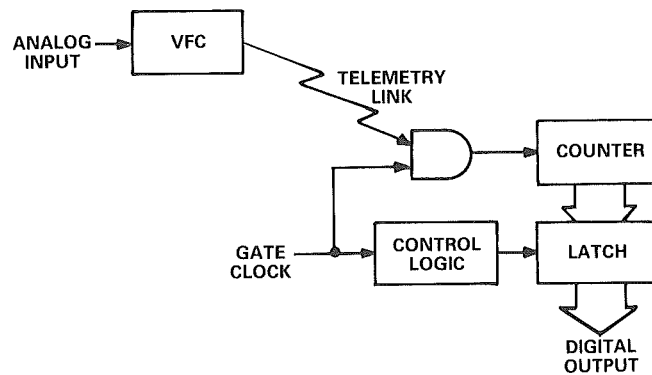
*If the sampling period is 100ms, of course, the DVM will reject line frequencies of 50Hz, 60Hz and 400Hz.

VFC USED FOR ANALOG-TO-DIGITAL CONVERSION



Like the integrating ADC, the gated VFC has inherently good DNL (no missing codes), signal integration for noise rejection, and high resolution. In addition it is particularly well-suited for telemetry applications since only a serial data channel is needed between the VFC and the counter/gate circuitry.

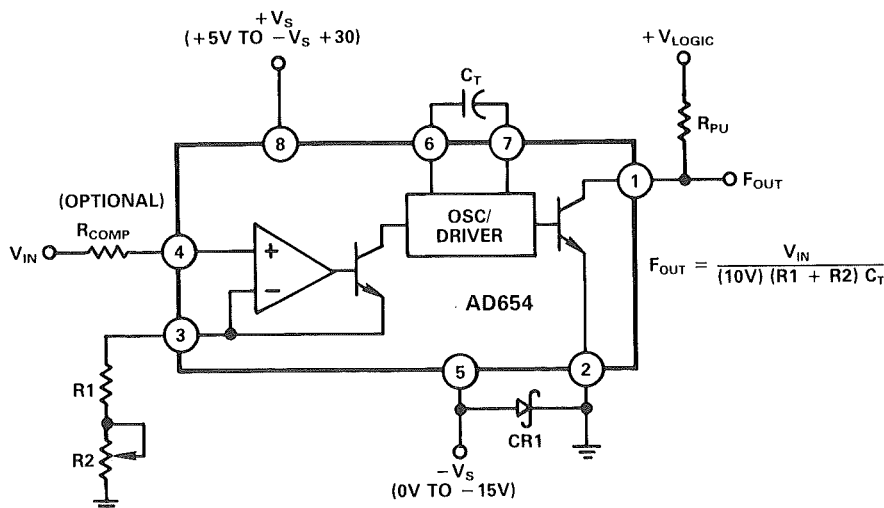
A VFC IS IDEALLY SUITED TO TELEMETRY APPLICATIONS



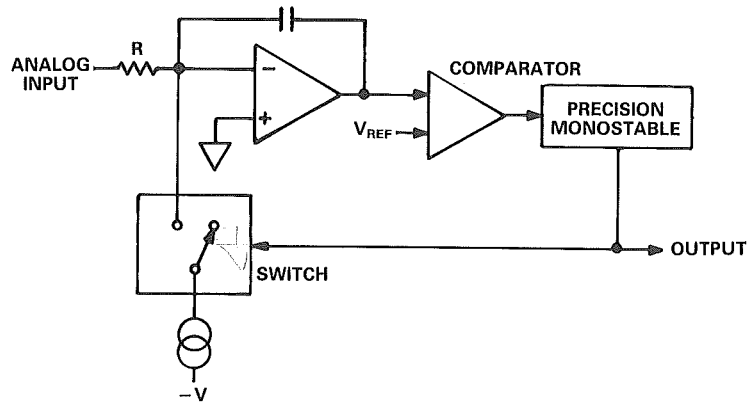
Most VFCs are used in unipolar mode (what is a negative frequency?) but if a bipolar ADC is required to be built from a VFC it is possible to inject an offset so that the VFC input can be positive or negative. This is analogous to a frequency modulated carrier.

The simplest form of VFC is an astable multivibrator whose timing current is set by the applied input voltage. The AD537 and AD654 are examples of such VFCs.

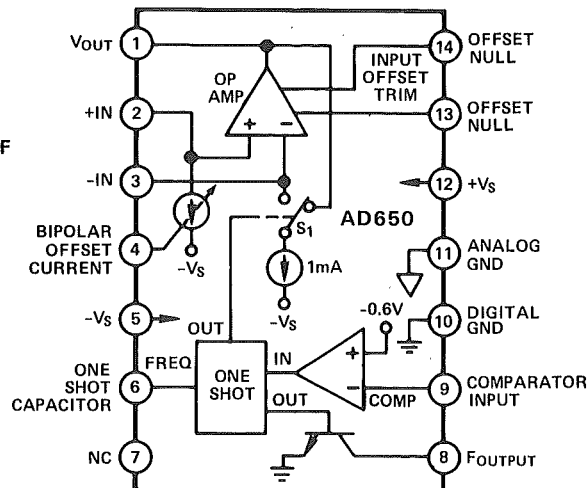
STANDARD V-F CONNECTION FOR POSITIVE INPUT VOLTAGES



This architecture is similar to that of an integrating ADC in that the input is applied to an integrator but instead of integrating for a fixed time the input to the comparator is set at a reference voltage and when the integrator output passes the reference the comparator triggers a monostable multivibrator and fixed charge (a fixed current switched to the integrator input for a fixed time) removed from the integrator.



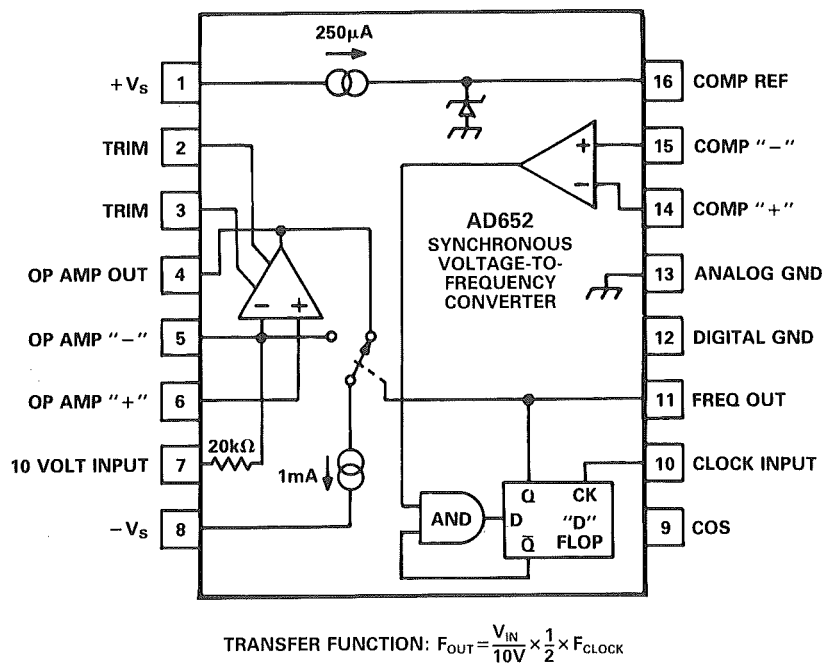
The AD650 contains a circuit intended to minimize switching nonlinearities. Instead of being connected to the integrator only during a pulse from the monostable the precision current source in the AD650 is always connected to the integrator. While the integrator is charging the current source is connected to the integrator output as a load, when the monostable fires the current is diverted to the integrator input where it discharges the integrator. However, the net current flowing in the integrator output is unchanged since the precision current source is still connected to the output, but now via the integration capacitor. This small change in architecture removes the majority of nonlinearities due to switching transients and causes the AD650 to be more than five times more linear than earlier attempts at monolithic VFCs using the same type of architecture.



V/F Conversion to 1MHz
Unipolar, Bipolar, or Differential V/F
Very High Linearity
 0.002% typ at 10kHz
 0.005% typ at 100kHz
 0.1% max at 1MHz
Input Offset Trimmable to Zero
CMOS or TTL Compatible
Reliable Monolithic Construction
V/F or F/V Conversion

Nevertheless the gain and linearity of this type of VFC is still dependent on the stability and dielectric absorption respectively of the monostable timing capacitor. These can be made very good but at temperature extremes they are still likely to be the first factors to affect the performance of the VFC. There is no way of avoiding the problem with any type of conventional VFC architecture. A synchronous VFC, however, is unaffected by this particular difficulty.

AD652 SYNCHRONOUS V-F CONVERTER

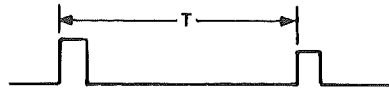


The AD652, a typical synchronous VFC, shows the difference between the two architectures. The AD651 is virtually identical to the AD652 except that its monostable multivibrator has been replaced by a D-type flip-flop driven by an external clock. In a synchronous VFC the integrator charges towards the threshold of the comparator just as in the asynchronous case but now the discharge does not take place as soon as the integrator passes the threshold but on the first clock pulse after it has done so, moreover the discharge pulse is no longer timed by a monostable but lasts for exactly one cycle of the external clock.

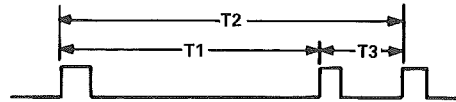
This arrangement removes all reliance on capacitor timing and makes the VFC gain depend only on the stability of the clock frequency and of the current source, yielding a further improvement in linearity and gain stability.

Synchronous VFCs are also particularly convenient for use in microprocessor systems since the VFC clock can be derived from the microprocessor clock and the VFC output read synchronously. They are useless, though, in any application where any amount of spectral purity is required in the VFC output (such as in phase-locked loop circuits) since the effect of clock-synchronism is to introduce massive phase jitter in the VFC output (the effect is equally visible on an oscilloscope or a spectrum analyzer and frequently misleads people inexperienced with synchronous VFCs into believing that the device is faulty).

DIFFERENCE BETWEEN OSCILLOSCOPE DISPLAYS OF NORMAL VFC AND SYNCHRONOUS VFC



NORMAL VFC.
T VARIES SMOOTHLY AS
ANALOG INPUT IS ALTERED.

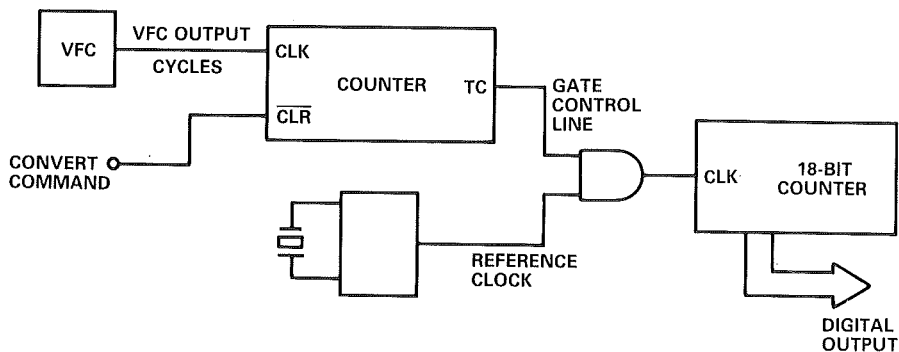


SYNCHRONOUS VFC.
TIMING IS QUANTIZED AND THE RATIO
OF CYCLES OF LENGTH T_1 AND OF
LENGTH T_2 VARIES AS ANALOG INPUT IS
ALTERED ($T_3 = 1$ CLOCK CYCLE).
 $T_1 = N T_3$ $T_2 = (N + 1) T_3$

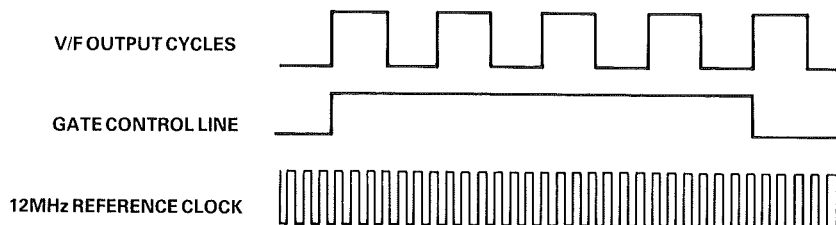
The resolution of a VFC depends on the sampling interval—even though changes in a synchronous VFC are quantized and those in a normal VFC are not the difference is not evident since it is the sampling time and not the quantized/nonquantized nature of the output signal which affects the resolution.

There is one way to beat this speed/resolution dilemma—to time the intervals between pulses of a nonsynchronous VFC. The AD1170 is an 18-bit smart ADC which uses this technique—counting cycles of a 12MHz reference clock for a preset number of cycles of its VFC. The AD1170 is called a “smart” ADC because it contains an 8051 microcomputer and the user can program whatever tradeoff between conversion time and resolution that he requires.

GATED CLOCK ADC



TIMING FOR GATED CLOCK ADC

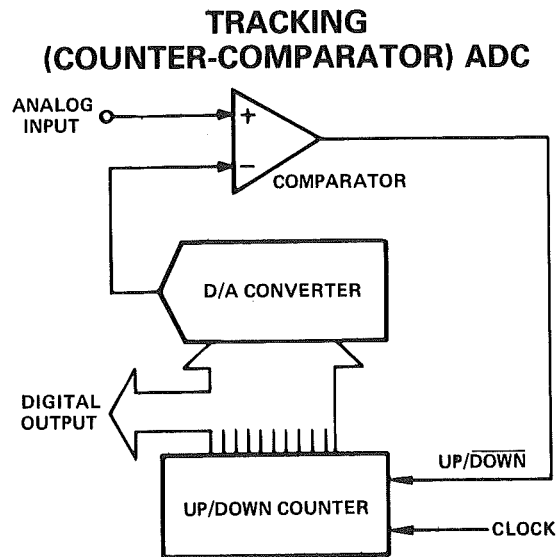


$C = \text{NUMBER OF V/F CYCLES DURING } T_{(INT)}$

$N = \text{NUMBER OF REFERENCE CLOCKS DURING } T_{(INT)}$

TRACKING ADCs

The two remaining types of ADC both compare the output of a DAC with the analog input signal to determine its amplitude. A tracking ADC controls the DAC with an up/down counter—if the analog input is larger than the DAC output the comparator causes the counter to count up until the two signals are equal, if the analog input is smaller the comparator causes the counter to count down.



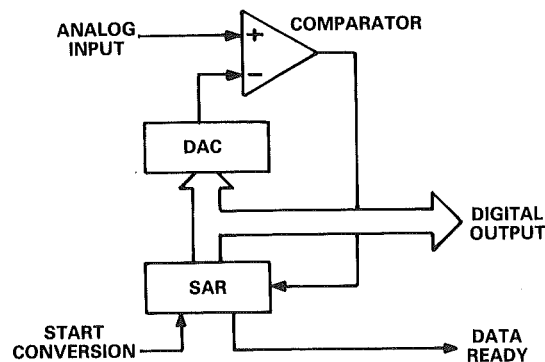
If a single comparator is used the counter will change direction every clock cycle once the DAC output has reached the analog input and the output will cycle above and below the actual value. If two comparators are used, with $+0.5$ LSB and -0.5 LSB offset respectively in the comparator enabling down counting and the one enabling up counting, then the converter will come to rest when the DAC output is within 0.5 LSB of the analog input.

If the analog input changes no faster than 1 LSB per clock cycle then the converter will track the input signal as it moves—on the other hand if the input steps from zero to full-scale or vice versa then the output will take $2^n - 1$ clock cycles to follow it. The tracking converter is thus quite fast when working with a slowly-varying input and very slow in adjusting to a step. For this reason they are mostly used in single-channel low speed applications and are rarely manufactured as ICs.

SUCCESSIVE APPROXIMATION ADCs

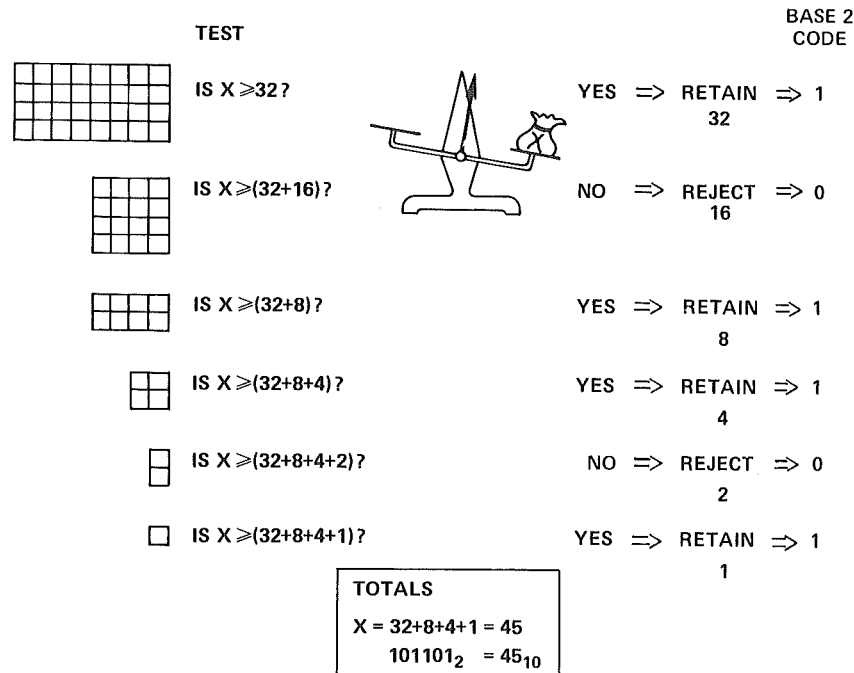
If the counter in a tracking ADC is replaced with a logic function called a successive approximation register (SAR) it produces a much more powerful type of ADC—the successive approximation ADC.

SUCCESSIVE APPROXIMATION ADC



An SAR has an N-bit parallel output which is connected to the DAC data input. On receipt of a convert command the SAR sets all its bits to logic 0 except the MSB which it sets to logic 1. After a delay during which the DAC and comparator settle the comparator output indicates whether or not the DAC output is greater or less than the analog input. If it is greater the SAR resets the MSB to logic 0, if it is not it leaves it at logic 1. The next bit is now set to logic 1 and the cycle repeated—again if the DAC output is greater than the analog input the bit is reset to logic 0, otherwise not. The procedure is repeated with each bit in turn, from the MSB to the LSB and when it is complete the SAR output, and hence the DAC output, is within 0.5 LSB of the value of the analog input.

SUCCESSIVE APPROXIMATION ADC ANALOGY



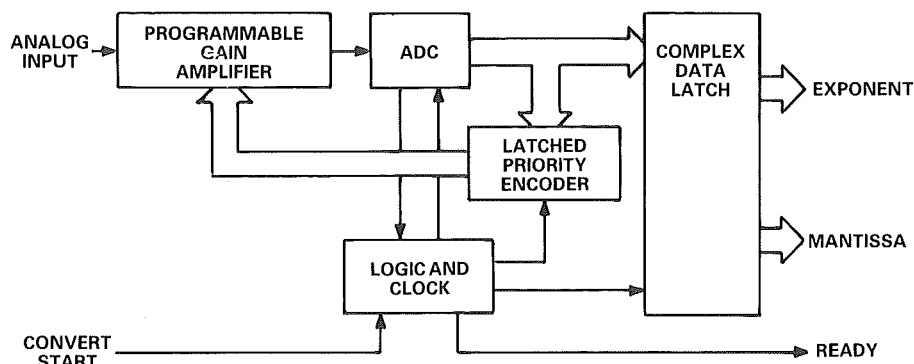
The concept of the successive approximation converter is directly analogous to that of a beam balance with a set of precision weights in successive binary ratio. One performs the measurement by placing the unknown weight on one side of the beam, then testing each known weight against the unknown, starting with the largest and working down. On each test if the weight tips the scale it is discarded, if it does not it is retained. The procedure continues to the smallest weight—if the balance is true and the weights all exact the unknown can be determined with a resolution equal to the smallest weight. (If a weight equal to half the smallest weight is added to the unknown side of the balance—analogue to half an LSB offset in the ADC—it can be shown that the measurement is always accurate to one half the smallest weight).

Since tracking and successive approximation ADCs use DACs they may be made unipolar, bipolar, or signed with little difficulty. Many such ADCs are designed for unipolar and bipolar operation with logic or switch selection of the bipolar offset.

Unlike the tracking converter the successive approximation converter does not deliver an output continuously—it performs discrete conversions. Normally ADCs start a conversion when they receive a logic signal telling them to do so but sometimes they convert continuously, retaining the result of the previous conversion until the new one is ready.

An n-bit SAR is barely more complex than an n-bit counter and easily implemented in most logic families. Therefore to make an N-bit successive approximation ADC we require an n-bit DAC, an n-bit SAR, and a comparator of adequate performance to handle 0.5 LSB differences at n bits. While it is perfectly possible to construct one's own ADC using from the individual components it is rarely economic to do so (it is sometimes economic to attach a DAC and a comparator to a microprocessor which is otherwise underutilized and use it as an SAR but as the price of ADCs drops this becomes less and less beneficial). Successive approximation ADCs are available in monolithic form up to 12-bit and in hybrid form at higher resolution (up to 16-bit). The choice of process used to manufacture a successive approximation ADC will depend on the exact functions required in the converter.

FLOATING-POINT ADC



A floating-point ADC is not a particular type of converter architecture but is a system which allows any converter architecture to give greater dynamic range. A basic floating-point ADC consists of an ADC preceded by a programmable gain amplifier (PGA) and some logic. A floating-point conversion consists of two conversions: the first takes place with the PGA at minimum gain and the second takes place after the PGA has been set in the light of the result of the first conversion. The output then consists of an exponent, which is derived from the result of the first conversion, and a mantissa, which is the result of the second. This floating-point architecture considerably increases the dynamic range of an ADC but does not increase its resolution.

AUXILIARY ADC SYSTEMS

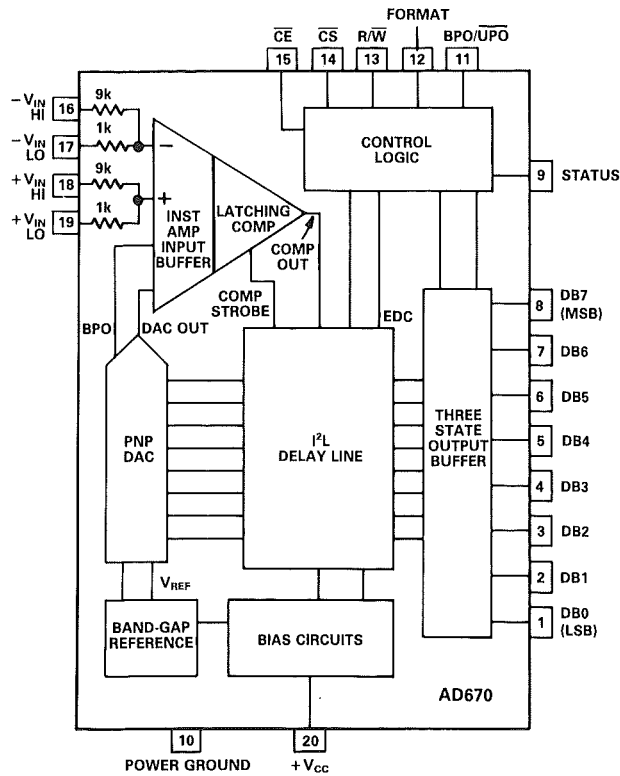
As with DACs, when complete ADCs first became available in hybrid or monolithic form they were so welcome that users were quite prepared to surround them with a considerable amount of additional circuitry. Now that circuitry is gradually being built into the ADCs themselves.

Such circuitry includes references, amplifiers and SHAs, and logic.

As with DACs all ADCs require references and many now contain band-gap buried-zener voltage references, according to the accuracy and stability required, on their chips or within their hybrid packages. Since CMOS processes do not make good references those ADCs which are fabricated in CMOS because of their logic requirements frequently do not contain references, but as second- and third-generation CMOS processes with good analog components (such as Analog Devices' LCCMOS) become more widely used more and more new ADCs do contain their own references.

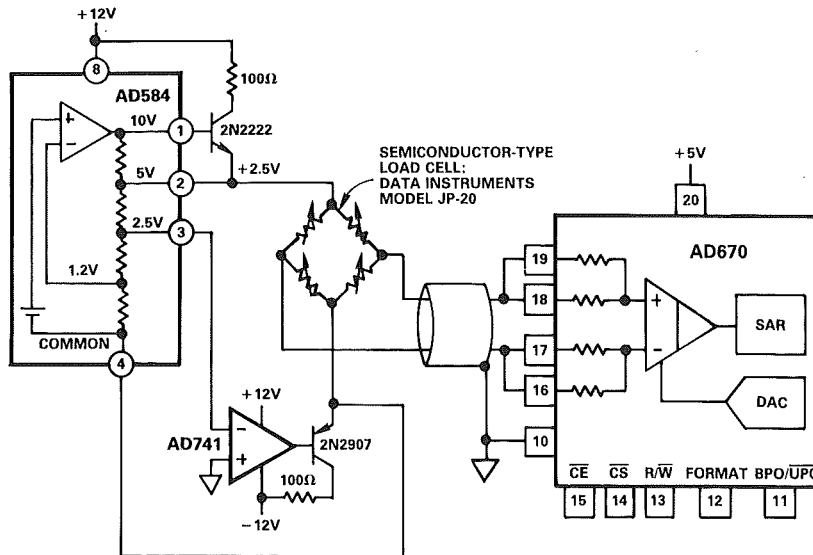
The input impedance of many ADCs is quite low, and indeed the nature of the comparator in some successive approximation ADCs causes their input impedance to be modulated at the SAR clock rate. Frequently ADCs must be driven with differential signals with common-mode voltages superimposed on them. Furthermore, as discussed above and in the section of the seminar dealing with SHAs, the input of successive approximation ADCs must not move by more than 0.5 LSB during conversion. We thus see that ideally all ADCs should contain buffer amplifiers (which may also be SHAs in the case of successive approximation ADCs) to isolate them from high source impedances, common-mode voltages, and wideband input signals. This is slowly happening but since including an amplifier on a chip increases its size and power consumption and may be impracticable with some IC processes.

AD670 BLOCK DIAGRAM

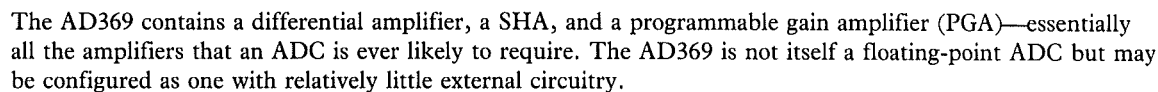


The AD670 is an excellent example of a monolithic ADC containing a differential amplifier. The amplifier has high input impedance, high common-mode rejection, and high gain and allows the AD670, in many applications, to be connected directly to a bridge sensor.

AD670 LOAD CELL INTERFACE



AD7575 FUNCTIONAL DIAGRAM



All ADCs contain logic in order to function at all. Today many contain many types of extra logic circuitry to make them more convenient to use. This generally consists of various types of output latches and buffers to ease the interfacing of ADCs to many types of microprocessor and display's. Many integration ADC's contain drivers for LCD and LED displays so that a DVM may consist of little more than a single IC and a display, and other types of ADC are frequently organized so that they may interface with several different widths of data bus. The conversion triggering circuitry is also becoming more sophisticated so that various automatic modes exist where reading data retriggers the converter, or the converter runs continuously but is locked out of the data buffer when it is being read, or other, more complex, schemes.

There are too many conflicting ADC requirements for there ever to be a single ADC for all applications, as data converter technology becomes continuously more sophisticated the possibilities for integrating extra functions within a monolithic ADC will lead to devices which can perform all the tasks which today are performed by a number of different types.