Chapter
29

Data Converter Architectures

Applications such as wireless communications and digital audio and video have created the need for cost-effective data converters that will achieve higher speed and resolution. The needs required by digital signal processors continually challenge analog designers to improve and develop new ADC and DAC architectures. There are many different types of architectures, each with unique characteristics and different limitations. This chapter presents a basic overview of the more popular data converter architectures and discusses the advantages and disadvantages of each along with their limitations.

Now that we have defined the operating characteristics of ADCs in Ch. 28, a more detailed examination of the basic architectures will be discussed using a top-down approach. Because many of the converters use op-amps, comparators, and resistor and capacitor arrays, the top-down approach will allow a broader discussion of the key component limitations in later sections.

29.1 DAC Architectures

A wide variety of DAC architectures exist, ranging from very simple to complex. Each, of course, has its own merits. Some use voltage division, whereas others employ current steering and even charge scaling to map the digital value into an analog quantity.

29.1.1 Digital Input Code

In many cases, the digital signal is not provided in binary code but is any one of a number of codes: binary, BCD, thermometer code, Gray code, sign-magnitude, two's complement, offset binary, and so on. (See Fig. 29.1 for a comparison of some of the more commonly used digital input codes.) For example, it may be desirable to allow only one bit to change value when changing from one code to the next. If that is the case, a Gray code will suffice. The thermometer code is used quite frequently and can also be seen in Fig. 29.1. Notice that it requires $2^N - 1$ bits to represent an N-bit word. The choice of code depends on the application, and the reader should be aware that many types of codes are available.

Decimal	Binary	Thermometer	Gray	Two's Complement
0	000	0000000	000	000
1	001	0000001	001	111
2	010	0000011	011	110
3	011	0000111	010	101
4	100	0001111	110	100
5	101	0011111	111	011
6	110	0111111	101	010
7	111	1111111	100	001

Figure 29.1 Comparison of digital input codes.

29.1.2 Resistor String

The most basic DAC is seen in Fig. 29.2a. Comprised of a simple resistor string of 2^N identical resistors and switches, the analog output is simply the voltage division of the resistors at the selected tap. Note that a N:2^N decoder is required to provide the 2^N signals controlling the switches. This architecture typically results in good accuracy, provided that no output current is required and that the values of the resistors are within the specified error tolerance of the converter. One big advantage of a resistor string is that the output is always guaranteed to be monotonic.

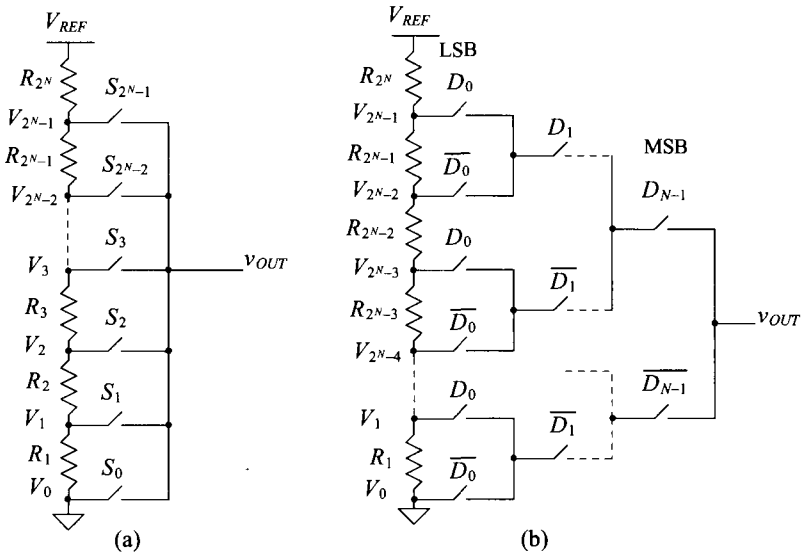

Figure 29.2 (a) A simple resistor-string DAC and (b) the use of a binary switch array to lower the output capacitance.

One problem with this converter is that the converter output is always connected to $2^N - 1$ switches that are off and one switch that is on. For larger resolutions, a large parasitic capacitance appears at the output node, resulting in slower conversion speeds. A better alternative for the resistor-string DAC is seen in Fig. 29.2b. Here, a binary switch array ensures that the output is connected to at most N switches that are on and N switches that are off, thus increasing conversion speed. The input to this switch array is a binary word since the decoding is inherent in the binary tree arrangement of the switches.

Another problem with the resistor-string DAC is the balance between area and power dissipation. An integrated version of this converter leads to a large chip area for higher bit resolutions because of the large number of passive components needed. Active resistors such as the n-well resistor can be used for low-resolution applications. However, as the resolution increases, the relative accuracy of the resistors becomes an important factor. Although the value of R could always be made small to minimize the chip area required, power dissipation would then become the critical issue as current flows through the resistor string at all times.

Example 29.1

Design a 3-bit resistor-string ladder using a binary switch array. Assume that V_{REF} = 5 V and that the maximum power dissipation of the converter is to be 5 mW (not including the power required by the digital logic). Determine the value of the analog voltage for each of the possible digital input codes.

The power dissipation will determine the current flowing through the resistor string by

$$I_{MAX} = \frac{5 \times 10^{-3}\ \text{W}}{5\ \text{V}} = 1\ \text{mA}$$

Since a 3-bit converter will have eight resistors, the value of R is

$$R = \frac{1}{8} \cdot \frac{5\ \text{V}}{1\ \text{mA}} = 625\ \Omega$$

The converter can be seen in Fig. 29.3. Examine the switch array if the input code is $D_2D_1D_0$ = 100 or 4_{10}. Since D_2 is high, the top switch will be closed and the lower switch, $\overline{D_2}$, will be open. In the row corresponding to D_1, since $D_1 = 0$, both of the switches marked $\overline{D_1}$ will be closed and the other two will be open. The LSB controls the largest number of switches; therefore, since D_0 is low, all of the $\overline{D_0}$ switches will be closed and all of the D_0 switches will be open. There should be only one path connecting a single tap on the resistor string to the output. This is bolded, with the resistor string tapped in the middle of the string. Therefore, v_{OUT} = ½ V_{REF} = 2.5 V. The remaining outputs can be seen in Fig. 29.4. ■

Mismatch Errors Related to the Resistor-String DAC

The accuracy of the resistor string is obviously related to matching between the resistors, which ultimately determines the INL and DNL for the entire DAC. Suppose that the i-th resistor, R_i, has a mismatch error associated with it so that

$$R_i = R + \Delta R_i \tag{29.1}$$

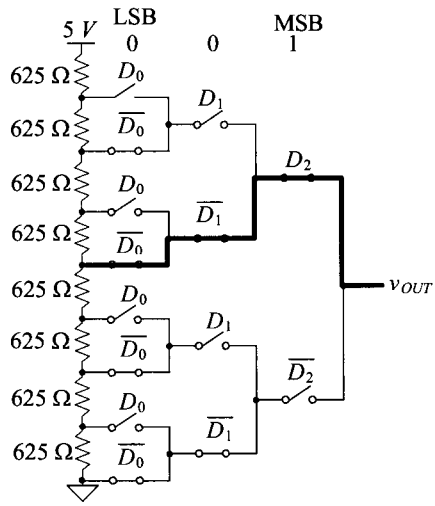

Figure 29.3 A 3-bit resistor-string DAC used in Ex. 29.1

where R is the ideal value of the resistor and ΔR_i is the mismatch error. Also suppose that the mismatches were symmetrical about the string so that the sum of all the mismatch terms were zero, or

$$\sum_{i=1}^{2^N} \Delta R_i = 0 \tag{29.2}$$

The value of the voltage at the tap associated with the i-th resistor should ideally be

$$V_{i,ideal} = \frac{(i)V_{REF}}{2^N}, \text{ for } i = 0, 1, 2, ..., 2^N - 1 \tag{29.3}$$

$D_2 D_1 D_0$	v_{OUT}
000	0
001	0.625
010	1.25
011	1.875
100	2.5
101	3.125
110	3.75
111	4.375

Figure 29.4 Output voltages generated from the 3-bit DAC in Ex. 29.1.

However, including the mismatch, the actual value of the i-th voltage is the sum of all of the resistances up to and including resistor i, divided by the sum of all of the resistances in the string. This can be represented by

$$V_i = V_{REF} \cdot \frac{\sum_{k=1}^{i} R_k}{\sum_{k=1}^{2^N} R_k} = V_{REF} \cdot \frac{\sum_{k=1}^{i} R + \Delta R_k}{2^N R} \tag{29.4}$$

The denominator does not include any mismatch error since it was assumed that the mismatches sum to zero as defined in Eq. (29.2). Notice that there is no resistor, R_0, corresponding to V_0 (see Fig. 29.2), and it is assumed that V_0 is ground. Equation (29.4) can be rewritten as

$$V_i = \frac{V_{REF}}{2^N R}\left[(i)R + \sum_{k=1}^{i} \Delta R_k\right] = \frac{(i)V_{REF}}{2^N} + \frac{V_{REF}}{2^N R}\sum_{k=1}^{i} \Delta R_k \tag{29.5}$$

or finally, the value of the voltage at the i-th tap is

$$V_i = V_{i,ideal} + \frac{V_{REF}}{2^N} \cdot \sum_{k=1}^{i} \frac{\Delta R_k}{R} \tag{29.6}$$

Equation (29.6) is not of much importance by itself, but it can be used to help determine the nonlinearity errors.

Integral Nonlinearity of the Resistor-String DAC

Integral nonlinearity (INL) is defined as the difference between the actual and ideal switching points, or

$$INL = V_i - V_{i,ideal} \tag{29.7}$$

and plugging in Eqs. (29.6) and (29.3) into (29.7) yields,

$$INL = \frac{V_{REF}}{2^N} \cdot \sum_{k=1}^{i} \frac{\Delta R_k}{R} \tag{29.8}$$

Equation (29.8) is a general expression for the INL for a given resistor, R_i, and requires that the mismatch of all the resistances used in the summation are known. However, this equation does not illuminate how to determine the *worst-case* or maximum INL for a resistor string.

Intuitively, one would think that the worst-case INL would occur at the top of the resistor string (i=2^N) with all of the ΔR_k's at their maximum values. However, the previous derivation was performed with the assumption that the mismatches summed to zero. With this restriction, the maximum INL occurs at the midpoint of the string where $i = 2^{N-1}$, corresponding to the case where the MSB was a one and all other bits were zero. Another condition that will ensure a worst-case scenario is to consider the lower half resistors at their maximum positive mismatch value and the upper half resistors at their maximum negative mismatch value, or vice versa.

If the resistors on a string were known to have 2% matching, then ΔR_k would be constrained to the bounds of

$$-0.02R \le \Delta R_k \le 0.02R \tag{29.9}$$

and the worst-case INL (again, % matching = 0.02) using Eq. (29.8) would be,

$$|INL|_{max} = \frac{V_{REF}}{2^N} \cdot \sum_{k=1}^{2^{N-1}} \frac{\Delta R_k}{R} = \frac{V_{REF}}{2^N} \cdot \frac{2^{N-1} \cdot \Delta R_k}{R} = \frac{1}{2} LSB \cdot 2^N \cdot (\%\ \text{matching}) = 0.01 V_{REF} \tag{29.10}$$

which for $INL < 0.5\ LSB$ requires $1/2^N >$ (% matching). For 2% matching, the maximum number of bits, N, is then 5! For better than 0.2% matching $N = 9$ bits.

Because the worst-case analysis was performed, the maximum INL occurs at the middle of the string. We can improve this specification on paper by using the "best-fit" approach to measuring INL. In this case, the reference line is simply shifted up slightly (refer to Ch. 28) so that it no longer passes through the end points, but instead minimizes the INL.

Example 29.2

Determine the effective number of bits for a resistor-string DAC, which is assumed to be limited by the INL. The resistors are passive poly resistors with a known relative matching of 1%, and $V_{REF} = 5$ V.

Using Eq. (29.10), the maximum INL will be

$$|INL|_{max} = 0.005 \cdot V_{REF} = 0.025 \text{ V}$$

Since we know that this maximum INL should be equal to ½ LSB in the worst case,

$$\frac{1}{2} LSB = \frac{5}{2^{N+1}} = 0.025 \text{ V}$$

and solving for N yields

$$N = \log_2\left(\frac{5}{0.025}\right) - 1 = 6.64 \text{ bits}$$

This means that the resolution for a DAC containing a resistor string matched to within 1% will be, at most 6 bits. ■

Differential Nonlinearity of the Worst-Case Resistor-String DAC

Resistor-string matching is not as critical when determining the DNL. Remembering that the definition of DNL is simply the actual height of the stair-step in the DAC transfer curve minus the ideal step height, we can write this in terms of the voltages at the taps of adjacent resistors on the string. Using Eq. (29.5), we can express this as,

$$|V_i - V_{i-1}| = \left| \left[\frac{(i)V_{REF}}{2^N} + \frac{V_{REF}}{2^N} \cdot \sum_{k=1}^{i} \frac{\Delta R_k}{R} \right] - \left[\frac{(i-1)V_{REF}}{2^N} + \frac{V_{REF}}{2^N} \cdot \sum_{k=1}^{i-1} \frac{\Delta R_k}{R} \right] \right|$$

which can be simplified to

$$|V_i - V_{i-1}| = \left| \frac{V_{REF}}{2^N} \left(1 + \frac{\Delta R_i}{R}\right) \right| \tag{29.11}$$

The DNL can then be determined by subtracting the ideal step height from Eq. (29.11),

$$DNL_i = \left| \frac{V_{REF}}{2^N} \left(1 + \frac{\Delta R_i}{R}\right) - \frac{V_{REF}}{2^N} \right| = \left| \frac{V_{REF}}{2^N} \cdot \frac{\Delta R_i}{R} \right| \tag{29.12}$$

and the maximum DNL will occur at the value of i for which ΔR is at its maximum value. If it is assumed once again that the resistors are matched to within 2 percent, the worst-case DNL will be

$$DNL_{max} = \left| 0.02 \cdot \frac{V_{REF}}{2^N} \right| = 0.02\ LSB \tag{29.13}$$

which is well below the ½ LSB limit. The INL is obviously the limiting factor in determining the resolution of a resistor-string DAC as its maximum value is 2^N times larger than the DNL.

29.1.3 *R-2R* Ladder Networks

Another DAC architecture that incorporates fewer resistors is called the *R-2R* ladder network. This configuration consists of a network of resistors alternating in value of R and $2R$. Figure 29.5 illustrates an N-bit *R-2R* ladder. Starting at the right end of the network, notice that the resistance looking to the right of any node to ground is $2R$. The digital input determines whether each resistor is switched to ground (noninverting input) or to the inverting input of the op-amp. Each node voltage is related to V_{REF}, by a binary-weighted relationship caused by the voltage division of the ladder network. The total current flowing from V_{REF} is constant, since the potential at the bottom of each switched resistor is always zero volts (either ground or virtual ground). Therefore, the node voltages remain constant for any value of the digital input.

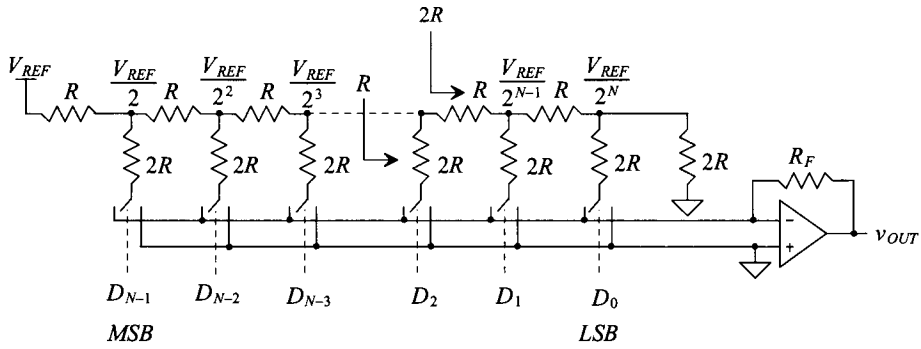

Figure 29.5 An *R-2R* digital-to-analog converter.

The output voltage, v_{OUT}, depends on currents flowing through the feedback resistor, R_F, such that

$$v_{OUT} = -i_{TOT} \cdot R_F \tag{29.14}$$

where i_{TOT} is the sum of the currents selected by the digital input by

$$i_{TOT} = \sum_{k=0}^{N-1} D_k \cdot \frac{V_{REF}}{2^{N-k}} \cdot \frac{1}{2R} \tag{29.15}$$

where D_k is the k-th bit of the input word with a value that is either a 1 or a 0.

This architecture, like the resistor-string architecture, requires matching to within the resolution of the converter. Therefore, the switch resistance must be negligible, or a small voltage drop will occur across each switch, resulting in an error. One way to eliminate this problem is to add dummy switches. Assume that the resistance of each switch connected to the $2R$ resistors is ΔR, as seen in Fig. 29.6. Dummy switches with one-half the resistance of the real switches are "hard-wired" so that they are always on and placed in series with each of the horizontal resistors. The total resistance of any horizontal branch, R', is

$$R' = R + \frac{\Delta R}{2} \tag{29.16}$$

The resistance of any vertical branch is $2R + \Delta R$, which is twice the value of the horizontal branch. Therefore, a $R' - 2R'$ relationship is maintained. Of course, a dummy switch equal to the switch size of a $2R$ switch will have to be placed in series with the terminating resistor as well.

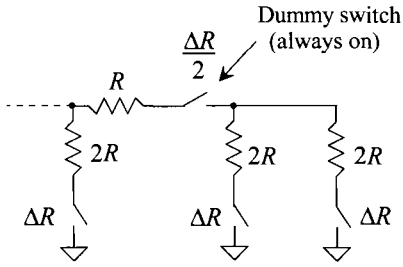

Figure 29.6 Use of dummy switches to offset switch resistance.

Example 29.3

Design a 3-bit DAC using an R-$2R$ architecture with R = 1 kΩ, R_F = 2 kΩ, and V_{REF} = 5 V. Assume that the resistances of the switches are negligible. Determine the value of i_{TOT} for each digital input and the corresponding output voltage, v_{OUT}.

Figure 29.7 shows the 3-bit DAC for a digital input of 001. The voltages at each node in the resistor network are labeled. For each switch, if the digital input bit is a zero, then the resistor is attached to the ground. If the bit is a one, then the resistor is attached to the virtual ground of the inverting input and current flows to the output of the op-amp. Therefore, for $D_2D_1D_0$ = 000, all of the switches are connected to ground, no current flows through the feedback resistor, and the output voltage, v_{OUT}, is zero.

When $D_2D_1D_0$ = 001, the rightmost resistor is switched to the op-amp inverting input and the other two resistors remain attached to ground. Therefore, the total current flowing through the feedback resistor is simply the current through the rightmost resistor, which is defined by Eq. (29.15) as

$$\frac{V_{REF}}{8} \cdot \frac{1}{2000} = 0.3126 \text{ mA}$$

and the output voltage, by Eq. (29.14) becomes,

$$v_{OUT} = -(0.3126 \text{ mA})(2000\ \Omega) = -0.625 \text{ V}$$

which is to be expected. The other values for the output voltage can be calculated using Eqs. (29.14) and (29.15) and are seen in Fig. 29.8. ■

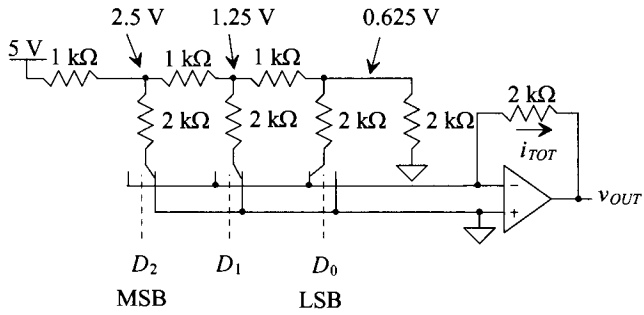

Figure 29.7 A 3-bit *R*-2*R*digital-to-analog converter used in Ex. 29.3.

$D_2D_1D_0$	i_{TOT} (mA)	v_{OUT} (V)
000	0	0
001	0.3125	– 0.625
010	0.625	– 1.25
011	0.625 + 0.3125 = 0.9375	– 1.875
100	1.25	– 2.5
101	1.25 + 0.3125 = 1.5625	– 3.125
110	1.25 + 0.625 = 1.875	– 3.75
111	1.25 + 0.625 + 0.3125 = 2.1875	– 4.375

Figure 29.8 Output voltages generated from the 3-bit DAC in Example 29.3.

29.1.4 Current Steering

In the previous section, a voltage was converted into a current, which then generated a voltage at the output. Another DAC method uses current throughout the conversion. Known as *current steering*, this type of DAC requires precision current sources that are summed in various fashions.

Figure 29.9 illustrates a generic current-steering DAC. This configuration requires a set of current sources, each having a unit value of current, I. Since there are no current sources generating i_{OUT} when all the digital inputs are zero, the MSB, D_{2^N-2}, is offset by two index positions instead of one. For example, for a 3-bit converter, seven current sources will be needed, labeled from D_0 to D_6. The binary signal controls whether or not the current sources are connected to either i_{OUT} or some other summing node (in this case ground). The output current, i_{OUT}, has the range of

$$0 \le i_{OUT} \le (2^N - 1) \cdot I \tag{29.17}$$

and can be any integer multiple of I in between. An interesting issue to note is the format of the digital code required to drive the switches. Since there are $2^N - 1$ current sources, the digital input will be in the form of a *thermometer code*. This code will be all ones from the LSB up to the value of the k-th bit, D_k, and all zeros above it. The point at which the input code changes from all ones to all zeros "floats" up or down and resembles the action of a thermometer, hence the name. Typically, a thermometer encoder is used to convert binary input data into a thermometer code.

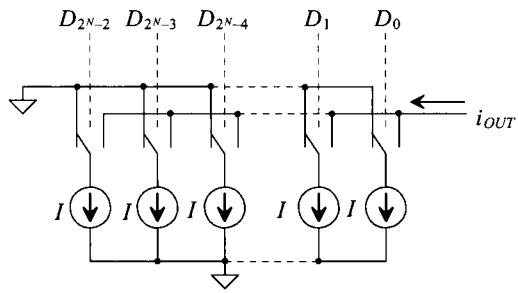

Figure 29.9 A generic current-steering DAC.

Another current-steering architecture is seen in Fig. 29.10. This architecture uses binary-weighted current sources, thus requiring only N current sources of various sizes versus $2^N - 1$ sources in the previous example. Since the current sources are binary weighted, the input code can be a simple binary number with no thermometer encoder needed.

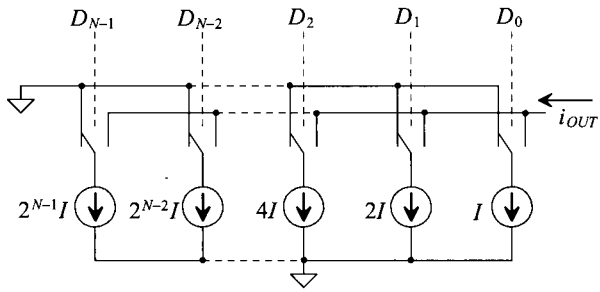

Figure 29.10 A current-steering DAC using binary-weighted current sources.

One advantage of the current-steering DACs is the high-current drive inherent in the system. Since no output buffers are necessary to drive resistive loads, these DACs are typically used in high-speed applications. Traditionally, high-speed current-steering DACs have been fabricated using bipolar technology. However, the ability to generate matched current mirrors makes CMOS an enticing alternative. Of course, the precision needed to generate high resolutions depends on how well the current sources can be matched or the degree to which they can be made binary weighted. For example, if a 13-bit DAC was designed using these architectures, there would have to be 8,191 current sources resident on the chip, not an insignificant amount. For the binary-weighted sources, only 13 current sources would be needed. Yet the size of the largest current source would have to be 4,096 or 2^{N-1} times larger than the smallest. Even if the unit current, I, was chosen to be 5 μA, the largest current source would be 20.48 mA!

Another problem associated with this architecture is the error due to the switching. Since the current sources are in parallel, if one of the current sources is switched off and another is switched on, a "glitch" could occur in the output if the timing was such that both of them were on or both were off for an instant. While this may not seem significant, if the converter is switching from 0111111 to 10000000, the output will spike toward ground and then back to the correct value if all the switches turn off for an instant. If the DAC is driving a resistive load and the output current is converted to a voltage, a substantial voltage spike will occur at the output.

Example 29.4

Construct a table showing the thermometer code necessary to generate the output shown in Fig. 29.11a for a 3-bit current-steering DAC using unit current sources.

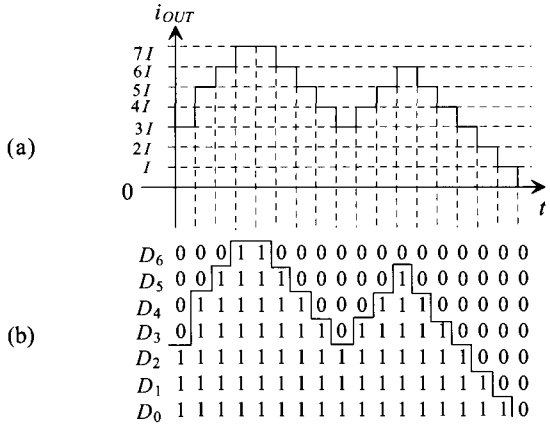

Figure 29.11 (a) Output of a 3-bit current-steering DAC and (b) the thermometer code input.

The thermometer code can be seen in Fig. 29.11b. When the code is all zeros, the output is 0 volts. Therefore, only 7 bits are needed to represent the 2^N or 8 states of a 3-bit DAC. Note how the interface between all ones and all zeros actually resembles the output signal itself. ■

Mismatch Errors Related to Current-Steering DACs

Analysis of the mismatch associated with the current sources is similar to the resistor string analysis. It is assumed that each current source in Fig. 29.9 is

$$I_k = I + \Delta I_k \text{ for } k = 1, 2, 3, ..., 2^N - 1 \tag{29.18}$$

where I is the ideal value of the current and ΔI_k is the error due to mismatch. If it is again assumed that the ΔI_K terms sum to zero and that one-half of the current sources contain the maximum positive mismatch, ΔI_{max}, and the other half contains the maximum negative mismatch, $-\Delta I_{max}$, (or vice versa), then the worst-case condition will occur at midscale with the actual output current being

$$I_{out} = \sum_{k=1}^{2^{N-1}} (I + \Delta I_k) = 2^{N-1} \cdot I + 2^{N-1} \cdot |\Delta I|_{max} = I_{out,ideal} + 2^{N-1} \cdot |\Delta I|_{max} \tag{29.19}$$

Since the INL is simply the actual output current minus the ideal, the worst-case INL is

$$|INL|_{max} = 2^{N-1} \cdot |\Delta I|_{max,INL} \tag{29.20}$$

The term, $|\Delta I|_{max,INL}$ represents the maximum current source mismatch error that will keep the INL less than ½ LSB. Each current source represents the value of 1 LSB; therefore, ½ LSB is equal to 0.5 I. Because the maximum INL should correspond to the ½ LSB, equating Eq. (29.20) to ½ I results in the value for $|\Delta I|_{max,INL}$,

$$|\Delta I|_{max,INL} = \frac{0.5I}{2^{N-1}} = \frac{I}{2^N} \tag{29.21}$$

Equation (29.21) illustrates the difficulty of using this architecture at high resolutions. If the value of I is set to be 5 μA, and the N is desired to be 12 bits, then

$$|\Delta I|_{max,INL} = \frac{5 \times 10^{-6}}{2^{12}} = 1.221 \text{ nA!} \tag{29.22}$$

which means that each of the 5 μA current sources must lie between the bounds of,

$$4.99878\ \mu A \le I_k \le 5.001221\ \mu A \tag{29.23}$$

to achieve a worst-case INL, which is within ½ LSB error.

The DNL is easily obtained since the step height in the transfer curve is equivalent to the value of the ideal current source, I. The maximum difference between any two adjacent values of output current will simply be the value of the single source, I_k, which contains the largest mismatch error for which the DNL will be less than ½ LSB, $|\Delta I|_{max,DNL}$:

$$I_{out(x)} - I_{out(x-1)} = I_k + |\Delta I|_{max,DNL} \tag{29.24}$$

Therefore, the DNL is simply

$$|DNL|_{max} = I_k + |\Delta I|_{max,DNL} - I_k = |\Delta I|_{max,DNL} \tag{29.25}$$

Equating the maximum DNL to the value of ½ LSB,

$$|\Delta I|_{max,DNL} = \tfrac{1}{2} LSB = \tfrac{1}{2} I \tag{29.26}$$

which is much easier to attain than the requirement for the INL.

For the binary-weighted current sources seen in Fig. 29.10, a slightly different analysis is needed to determine the requirements for INL and DNL. In this case, it will be assumed that the current source corresponding to the MSB (D_{N-1}) has a maximum positive mismatch error value and that the remainder of the bits (D_0 to D_{N-2}) contain a maximum negative mismatch error, so that the sum of all the errors equals zero. The INL is

$$|INL|_{max} = 2^{N-1}(I + |\Delta I|_{max,INL}) - 2^{N-1} \cdot I = 2^{N-1} \cdot |\Delta I|_{max,INL} \quad (29.27)$$

which is equivalent to the value of the current steering array in Fig. 29.9.

The DNL is slightly different because of the binary weighting of the current sources. One cannot add a single current source with each incremental increase in the digital input code. However, the worst-case condition for binary-weighted arrays tends to occur at midscale when the code transitions from 011111....111 to 100000....000. The worst-case DNL at this point is

$$DNL_{max} = \left[2^{N-1} \cdot (I + |\Delta I|_{max,DNL}) - \sum_{k=1}^{N-1} 2^{k-1} \cdot (I - |\Delta I|_{max,DNL}) \right] - I \quad (29.28)$$

which can be written as

$$DNL_{max} = 2^{N-1} \cdot (I + |\Delta I|_{max,DNL}) - (2^{N-1} - 1) \cdot (I - |\Delta I|_{max,DNL}) - I = (2^N - 1) \cdot |\Delta I|_{max,DNL} \quad (29.29)$$

and setting this value equal to ½ LSB and solving for ΔI_{max},

$$|\Delta I|_{max,DNL} = \frac{0.5I}{2^N - 1} = \frac{I}{2^{N+1} - 2} \quad (29.30)$$

Therefore, the DNL requirements for the binary-weighted current source array is more stringent than the INL requirements.

One interesting issue regarding the previous derivation is that the challenging accuracy requirements in Eq. (29.30) are placed only on the MSB current source. For each of the remaining binary-weighted sources, the DNL requirements become more relaxed. This is simply because the size of the MSB source is equivalent to all of the other sources combined, and so its value plays the most important role in the DAC's accuracy.

Example 29.5

Determine the tolerance of the MSB current source on a 10-bit binary-weighted current source array with a unit current source of 1 μA, which will result in a worst-case DNL that is less than ½ LSB.

Since Eq. (29.30) defines the maximum $|\Delta I|$ needed to keep the DNL less than ½ LSB, we must first use this equation,

$$|\Delta I|_{max,DNL} = \frac{1 \times 10^{-6}}{2^{11} - 2} = 0.4888 \text{ nA}$$

For a 10-bit DAC, the MSB current source will have a value that is 2^9 times larger than the unit current source, or 0.512 mA. Therefore, the range of values for which this array will have a DNL that is less than ½ LSB is

$$0.51199995 \text{ mA} \leq I_{MSB} \leq 0.5120004888 \text{ mA} \quad \blacksquare$$

29.1.5 Charge-Scaling DACs

A very popular DAC architecture used in CMOS technology is the charge-scaling DAC. Shown in Fig. 29.12a, a parallel array of binary-weighted capacitors, totaling $2^N C$, is connected to an op-amp. The value, C, is a unit capacitance of any value. After initially being discharged, the digital signal switches each capacitor to either V_{REF} or ground, causing the output voltage, v_{OUT}, to be a function of the voltage division between the capacitors.

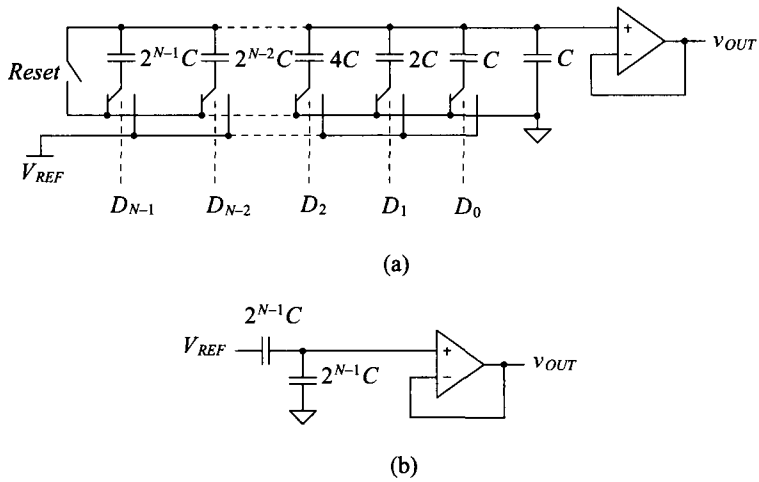

Figure 29.12 (a) A charge-scaling DAC, (b) the equivalent circuit with the MSB = 1, and all other bits set to zero.

The capacitor array totals $2^N C$. Therefore, if the MSB is high and the remaining bits are low, then a voltage divider occurs between the MSB capacitor and the rest of the array. The analog output voltage, v_{OUT}, becomes

$$v_{OUT} = V_{REF} \cdot \frac{2^{N-1}C}{(2^{N-1}+2^{N-2}+2^{N-3}+\ldots+4+2+1+1)C} = V_{REF} \cdot \frac{2^{N-1}C}{2^N C} = \frac{V_{REF}}{2} \tag{29.31}$$

which confirms the fact that the MSB changes the output of a DAC by ½ V_{REF}. Figure 29.12b shows the equivalent circuit under this condition. The ratio between v_{OUT} and V_{REF} due to each capacitor can be generalized to

$$v_{OUT} = \frac{2^k C}{2^N C} \cdot V_{REF} = 2^{k-N} \cdot V_{REF} \tag{29.32}$$

where it is assumed that the k-th bit, D_k, is one and all other bits are zero. Superposition can then be used to find the value of v_{OUT} for any digital input word by

$$v_{OUT} = \sum_{k=0}^{N-1} D_k 2^{k-N} \cdot V_{REF} \tag{29.33}$$

One limitation of this architecture as shown in Fig. 29.12a is the existence of a parasitic capacitance at the top plate of the capacitor array due to the op-amp. This will prohibit its use as a high-resolution data converter. A better implementation would include the use of a parasitic insensitive, switched-capacitor integrator (see Ch. 25) as the driving circuit. However, the capacitor array itself is the critical component of this data converter and is used in charge redistribution ADCs (Sec. 29.2.5).

The INL and DNL calculations for the binary-weighted capacitor array are identical to those for the binary-weighted current source array, except that the unit current source, I, and its corresponding error term, ΔI, are replaced by C and ΔC in Eqs. (29.27) – (29.30).

Example 29.6

Design a 3-bit charge-scaling DAC and find the value of the output voltage for $D_2D_1D_0 = 010$ and 101. Assume that $V_{REF} = 5$ V and $C = 0.5$ pF.

The 3-bit DAC can be seen in Fig. 29.13a. The equivalent circuits for the capacitor array can be seen in Fig. 29.13b and c. The value of the output voltage can be calculated by either using Eq. (29.32) or the equivalent circuits and performing the voltage division. For $D = 010$, the equivalent circuit in Fig. 29.13b yields

$$v_{OUT} = V_{REF} \cdot (\frac{1}{4}) = 1.25 \text{ V}$$

Using Eq. (29.33) to calculate v_{OUT} for $D = 101$ yields

$$v_{OUT} = \sum_{k=0}^{N-1} D_k 2^{k-N} \cdot V_{REF} = [1 \cdot (2^{-3}) + 0 \cdot (2^{-2}) + 1 \cdot (2^{-1})] \cdot 5 = \left(\frac{1}{8} + \frac{1}{2}\right) \cdot 5 = 3.125 \text{ V}$$

which is the result expected. ■

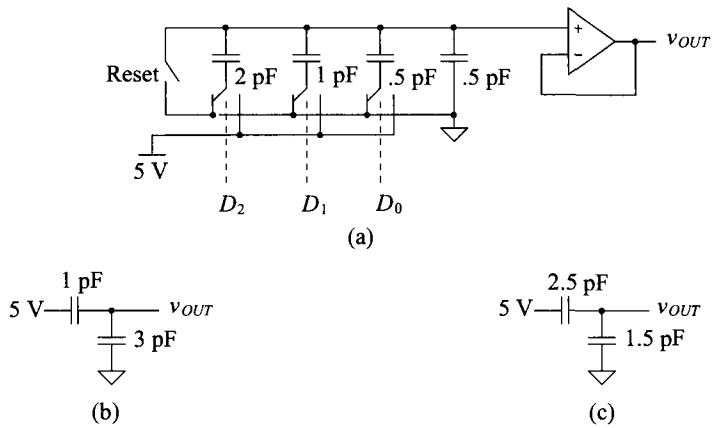

Figure 29.13 (a) A 3-bit charge-scaling DAC used in Ex. 29.6 and the equivalent circuits inputs equal to (b) 010 (c) 101.

Layout Considerations for a Binary-Weighted Capacitor Array

One problem with this converter is the need for precisely ratioed capacitors. As the number of bits increase, the ratio of the MSB capacitor to the LSB capacitor becomes more difficult to control. For example, Fig. 29.14a shows a 3-bit binary capacitor array using three capacitors. When the capacitor is fabricated, *undercutting* of the mask causes an error in the ratio of the capacitors, creating potentially large DNL and INL errors as N increases.

One solution to this problem is seen in Fig. 20.14b. Here, each capacitor in the array is constructed out of a unit capacitance. Undercutting then affects all of the capacitors in the same way, and the ratio between capacitors is maintained. Another problem that affects even this layout strategy is a nonuniform oxide growth. Gradients result in errors in the ratios of the capacitors. Figure 29.14c illustrates another layout strategy that overcomes this issue. The capacitors are laid out in a common-centroid scheme so that the first-order oxide errors average out to be the same for each capacitor.

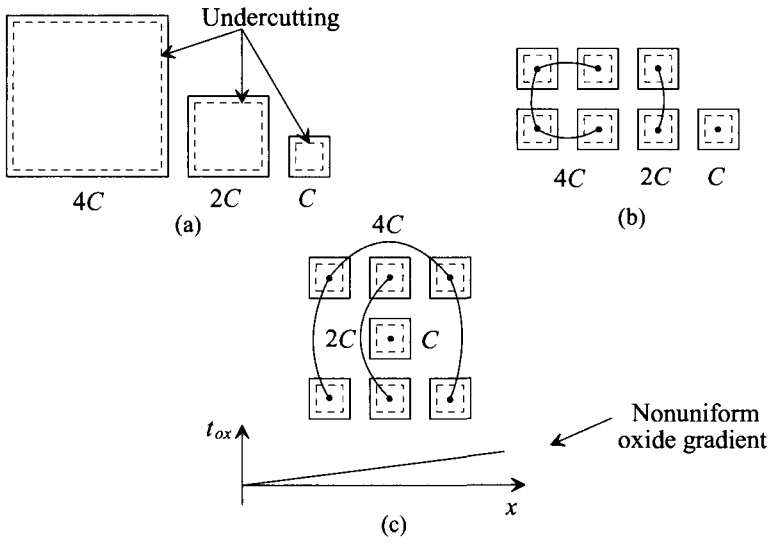

Figure 29.14 Layout of a binary-weighted capacitor array using (a) single capacitors (b) unit capacitors to minimize undercutting effect, and (c) common-centroid to minimize oxide gradients.

The Split Array

The charge-scaling architecture is very popular among CMOS designers because of its simplicity and relatively good accuracy. Although a linear capacitor is required using poly2, high resolutions in the 10- to 12-bit range can be achieved. Passive, double-poly capacitors have good matching accuracy as well. However, as the resolution increases, the size of the MSB capacitor becomes a major concern. For example if the unit capacitor, C,

were 0.5 pF, and a 16-bit DAC were to be designed, the MSB capacitor would need to be

$$C_{MSB} = 2^{N-1} \cdot 0.5 \text{ pF} = 16.384 \text{ nF} \tag{29.34}$$

If the capacitance between poly1 and poly2 is nominally 25 fF/μm^2, then the area required for this one capacitor is (roughly) 800 by 800 μm^2.

One method of reducing the size of the capacitors is to use a split array. A 6-bit example of the array is pictured in Fig. 29.15. This architecture is slightly different from the charge-scaling DAC pictured in Fig. 29.13 in that the output is taken off a different node and an additional attenuation capacitor is used to separate the array into a LSB array and a MSB array. Note that the LSB, D_0, now corresponds to the leftmost switch and that the MSB, D_5, corresponds to the rightmost switch. The value of the attenuation capacitor can be found by

$$C_{atten} = \frac{sum\ of\ the\ LSB\ array\ capacitors}{sum\ of\ the\ MSB\ array\ capacitors} \cdot C \tag{29.35}$$

where the sum of the MSB array equals the sum of LSB capacitor array minus C. The value of the attenuation capacitor should be such that the series combination of the attenuation capacitor and the LSB array, assuming all bits are zero, equals C.

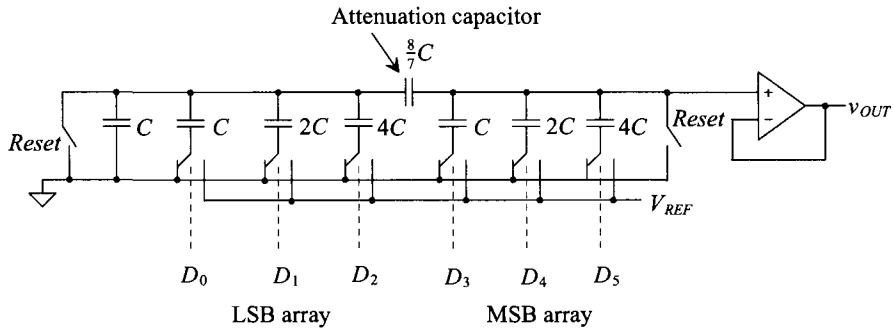

Figure 29.15 A charge-scaling DAC using a split array.

Example 29.7

Using the 6-bit charge-scaling DAC shown in Fig. 29.15, (a) show that the output voltage will be $\frac{1}{2} \cdot V_{REF}$ if (a) $D_5 D_4 D_3 D_2 D_1 D_0$ = 100000 and (b) the output will be $\frac{1}{64} \cdot V_{REF}$ if $D_5 D_4 D_3 D_2 D_1 D_0 = 000001$.

(a) If $D_5 = 1$ and the remaining bits are all zero, then the equivalent circuit for the DAC can be represented by Fig. 29.16a. The expression for the output voltage then becomes

$$v_{OUT} = \frac{4}{\left(\frac{8}{7} \text{ in series with } 8\right) + 3 + 4} \cdot V_{REF} = \frac{1}{2} \cdot V_{REF}$$

(b) For the second case, the equivalent circuit can be seen in Fig. 29.16b. The intermediate node voltage, V_A, is simply the voltage division between the C associated with D_0 and the remainder of the circuit, or

$$V_A = V_{REF} \cdot \frac{1}{\left(7 + \frac{\frac{8}{7}\cdot 7}{\frac{8}{7}+7}\right) + 1} = \frac{1}{8 + \frac{56}{57}} \cdot V_{REF} \tag{29.36}$$

The output voltage can be written as

$$v_{OUT} = V_A \cdot \frac{\frac{8}{7}}{\frac{8}{7}+7} = \frac{8}{57} \cdot V_A \tag{29.37}$$

Plugging Eq. (29.36) into Eq. (29.37) yields

$$v_{OUT} = V_{REF} \cdot \frac{8}{(8\cdot 57)+56} = \frac{V_{REF}}{64} \tag{29.38}$$

which is the desired result. ■

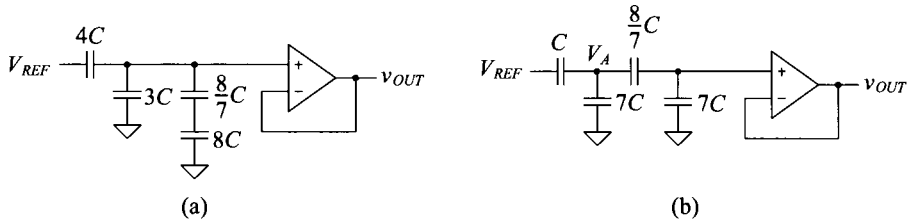

Figure 29.16 Equivalent circuits for Example 29.7.

29.1.6 Cyclic DAC

The cyclic DAC uses only a couple of simple components to perform the conversion. As seen in Fig. 29.17, a summer adds V_{REF} or ground to the feedback signal depending on the input bits. An amplifier with a gain of 0.5 feeds the output voltage back to the summer such that the output at the end of each cycle depends on the value of the output during the cycle before. Notice that the input bits must be read in a serial fashion. Therefore, the conversion is performed one bit at a time, resulting in N cycles required for each conversion. The voltage output at the end of the n-th cycle of the conversion can be written as

$$v_{OUT}(n) = \left(D_{n-1} \cdot V_{REF} + \frac{1}{2} \cdot v_A(n-1)\right) \cdot \frac{1}{2} \tag{29.39}$$

with a condition such that the output of the S/H is initially zero [$v_A(0) = 0$ V].

The accuracy of this converter is dependent on several factors. The gain of the 0.5 amplifier needs to be highly accurate (to within the accuracy of the DAC) and is usually generated with passive capacitors. Similarly, the summer and the sample-and-hold also need to be N-bit accurate. Limitations of the converter due to these fundamental building blocks will be discussed in more detail in Sec. 29.2.3. Since this converter uses a pseudo-"sampled-data" approach, implementing this architecture using switched capacitors is relatively easy.

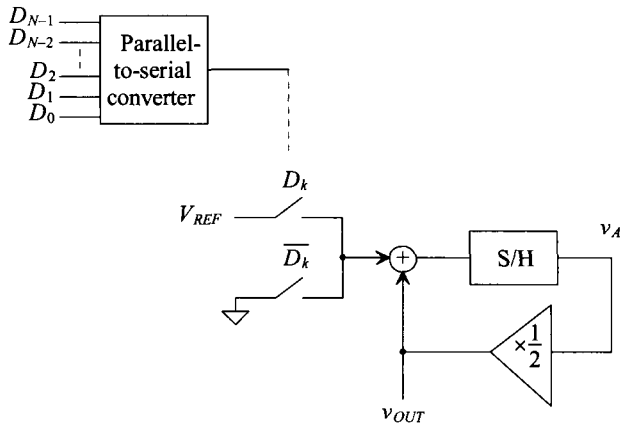

Figure 29.17 A cyclic digital-to-analog converter.

Example 29.8

Show the value of the output voltage at the end of each cycle for a 6-bit cyclic DAC with an input value of $D_5D_4D_3D_2D_1D_0 = 110101$. Assume that $V_{REF} = 5$ V.

We can predict the value of the output based on our previous experience with DACs. The digital input 110101 corresponds to 53_{10}. Therefore, the output voltage due to this input should be

$$v_{OUT} = \frac{53}{64} \cdot V_{REF} = 4.140625 \text{ V}$$

Now examine the cyclic converter in Fig. 29.17. By performing a 6-bit conversion and using Eq. (29.39), the outputs occurring at the end of each cycle can be seen in Fig. 29.18.

The output voltage at the end of the sixth cycle is precisely what was predicted. Note that had this been a 3-bit conversion, the output voltage at the end of cycle 3 would correspond to the value of the 3-bit DACs studied previously with an input of 101. ■

Cycle Number, n	D_{n-1}	$v_A(n\text{-}1)$	$v_{OUT}(n)$
1	1	0	½ (5 + 0) = 2.5 V
2	0	5	½ (0 + 2.5) = 1.25 V
3	1	2.5	½ (5 +1.25) = 3.125 V
4	0	6.25	½ (0 + 3.125) = 1.5625 V
5	1	3.125	½ (5 + 1.5625) = 3.28125 V
6	1	6.5625	½ (5 + 3.28125) = 4.140625 V

Figure 29.18 Output from the 6-bit cyclic DAC used in Ex. 29.8.

29.1.7 Pipeline DAC

The cyclic converter presented in the last section takes N clock cycles per N-bit conversion. Instead of recycling the output back to the input each time, we could extend the cyclic converter to N stages, where each stage performs one bit of the conversion. This extension of the cyclic converter is called a *pipeline* DAC and is seen in Fig. 29.19. Here, the signal is passed down the "pipeline," and as each stage works on one conversion, the previous stage can begin processing another. Therefore, an initial N clock cycle delay is experienced as the signal makes its way down the pipeline the very first time. After the N clock cycle delay, a conversion takes place at every clock cycle.

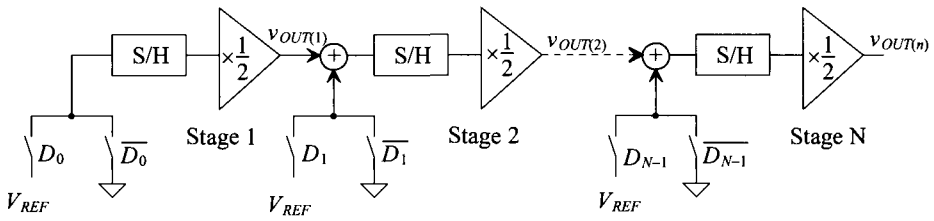

Figure 29.19 A pipeline digital-to-analog converter.

Besides the N clock cycle delay, this architecture can be very fast. However, the amplifier gains must be very accurate to produce high resolutions. Also, this architecture uses N times more circuitry than that of the cyclic, so there is a trade-off between speed and chip area. The output voltage of the n-th stage in the converter can be written as

$$v_{OUT(n)} = [D_{n-1} \cdot V_{REF} + v_{OUT(n-1)}] \cdot \frac{1}{2} \tag{29.40}$$

The operation of each stage in the pipeline can be summarized as follows: if the input bit is a 1, add V_{REF} to the output of the previous stage, divide by two, and pass the value to the next stage. If the input bit is a 0, simply divide the output of the previous stage by two and pass along the resulting value.

Example 29.9

Find the output voltage for a 3-bit pipeline DAC for three cases: $D_A = 001$, $D_B = 110$, and $D_C = 101$. Show that the conversion time to perform all three conversions is five clock cycles using the pipeline approach. Assume that $V_{REF} = 5$ V.

The first stage operates on the LSBs of each word; the second stage operates on the middle bits; and the last stage, the MSBs. Based on the pipeline strategy, once the LSB of the first input word is performed and passed on, the LSB of the second word, D_B, can begin its conversion. Similarly, once the LSB of the second stage is completed and passed on, the LSB of the third word, D_C, can begin. The conversion cycle for all three input words produces the output shown in Fig. 29.20. The items that are in bold are associated with the first input word, D_A, whereas the italicized numbers represent the values associated with D_B and the underlined items, D_C.

Clock Cycle	$v_{OUT(1)}$	$v_{OUT(2)}$	$v_{OUT(3)}$	D_0	D_1	D_2
1	**2.5**	0	0	**1**	0	0
2	*0*	**1.25**	0	*0*	**0**	0
3	2.5	*2.5*	**0.625**	1	*1*	**0**
4		1.25	*3.75*		0	*1*
5			3.125			1

Figure 29.20 Output from the 3-bit pipeline DAC used in Example 29.9.

The first output of the DAC is not valid until the end of the third clock cycle and should look familiar as the 3-bit DAC output for an input word of $D_2 D_1 D_0 = 001$. The following two clock cycles that produce outputs for $D_2 D_1 D_0$ equal 110 and 101, respectively. ■

29.2 ADC Architectures

A survey of the field of current A/D converter research reveals that a majority of effort has been directed to four different types of architectures: pipeline, flash-type, successive approximation, and oversampled ADCs. Each has benefits that are unique to that architecture and span the spectrum of high speed and resolution.

Since the ADC has a continuous, infinite-valued signal as its input, the important analog points on the transfer curve x-axis for an ADC are the ones that correspond to changes in the digital output word. These input transitions determine the amount of INL and DNL associated with the converter.

29.2.1 Flash

Flash or parallel converters have the highest speed of any type of ADC. As seen in Fig. 29.21, they use one comparator per quantization level ($2^N - 1$) and 2^N resistors (a resistor-string DAC). The reference voltage is divided into 2^N values, each of which is fed into a comparator. The input voltage is compared with each reference value and results in a thermometer code at the output of the comparators. A thermometer code exhibits all zeros for each resistor level if the value of v_{IN} is less than the value on the resistor string, and ones if v_{IN} is greater than or equal to voltage on the resistor string. A simple $2^N - 1{:}N$ digital thermometer decoder circuit converts the compared data into an N-bit digital word. The obvious advantage of this converter is the speed with which one conversion can take place. Each clock pulse generates an output digital word. The advantage of having high speed, however, is counterbalanced by the doubling of area with each bit of increased resolution. For example, an 8-bit converter requires 255 comparators, but a 9-bit ADC requires 511! Flash converters have traditionally been limited to 8-bit resolution with conversion speeds of 10–40 Ms/s using CMOS technology. The disadvantages of the Flash ADC are the area and power requirements of the $2^N - 1$ comparators. The speed is limited by the switching of the comparators and the digital logic.

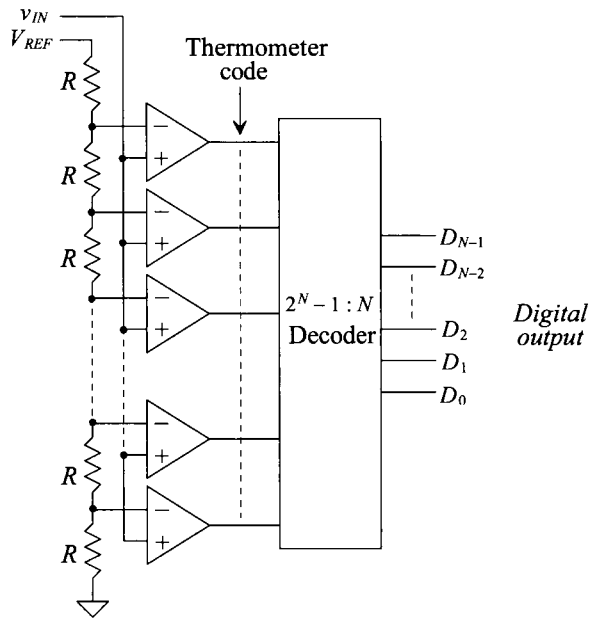

Figure 29.21 Block diagram of a Flash ADC.

Example 29.10

Design a 3-bit Flash converter, listing the values of the voltages at each resistor tap, and draw the transfer curve for $v_{IN} = 0$ to 5 V. Assume $V_{REF} = 5$ V. Construct a table listing the values of the thermometer code and the output of the decoder for $v_{IN} = 1.5$, 3.0, and 4.5 V.

The 3-bit converter can be seen in Fig. 29.22. As the values of all the resistors are equal, the voltage of each resistor tap, V_i, will be $V_i = V_{REF}\left(\frac{i}{8}\right)$ where i is the number of the resistor in the string for $i = 1$ to 7. Obviously, $V_1 = 0.625$ V, $V_2 = 1.25$ V, $V_3 = 1.875$ V, $V_4 = 2.5$ V, $V_5 = 3.125$, $V_6 = 3.75$ V, $V_7 = 4.375$ V. Therefore, when v_{IN} first becomes equal or greater than each of these values, a transition will occur in the transfer curve. The transfer curve can be seen in Fig. 29.23 and should look similar to those seen in Ch. 28. The quantization levels and their corresponding thermometer codes are summarized in Fig. 29.24.

The transfer curve of this ADC corresponds to the ADC with quantization error centered about +½ LSB, as discussed in Ch. 28 (Fig. 28.20). To shift the curve by ½ LSB so that the code transitions occur around the LSB values and the quantization error is centered around 0 LSB, the value of the last resistor in the string would have to be adjusted to $\frac{R}{2}$ and the value of the MSB resistor, closest to the reference voltage, would have to be made $1.5R$. Then the first code transition would occur at $v_{IN} = 0.3125$ V, and the last code transition would occur at $v_{IN} = 4.0625$, and so the transfer curve would exactly match that of Fig. 28.20.

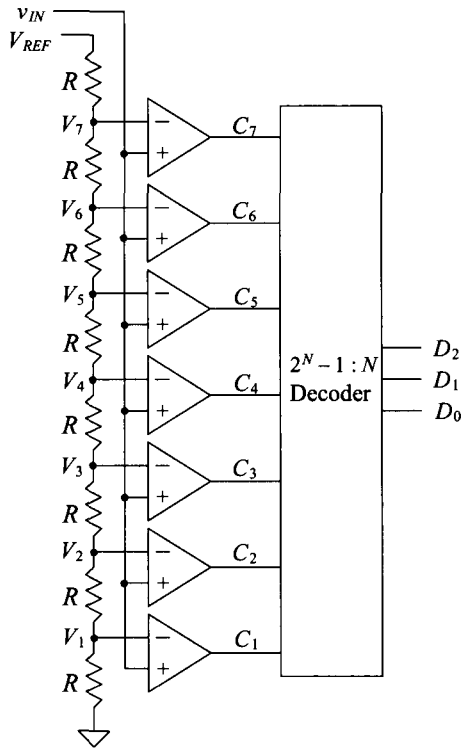

Figure 29.22 Three-bit Flash A/D converter to be used in Ex. 29.10.

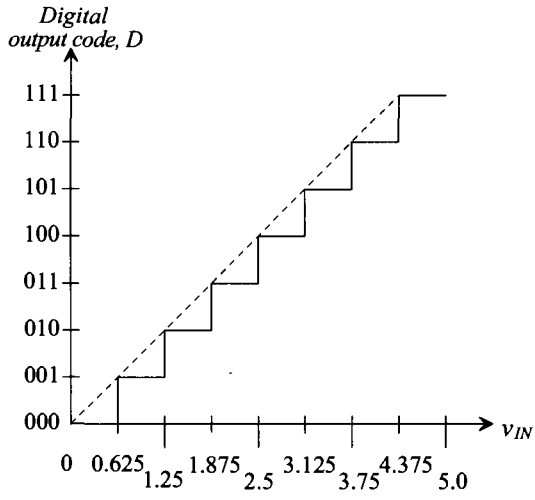

Figure 29.23 Transfer curve for the 3-bit Flash converter in Example 29.10.

v_{IN}	$C_7C_6C_5C_4C_3C_2C_1$	$D_2D_1D_0$
$0 \le v_{IN} < 0.625$ V	0000000	000
0.625 V $\le v_{IN} < 1.25$ V	0000001	001
1.25 V $\le v_{IN} < 1.875$ V	0000011	010
1.875 V $\le v_{IN} < 2.5$ V	0000111	011
2.5 V $\le v_{IN} < 3.125$ V	0001111	100
3.125 V $\le v_{IN} < 3.75$ V	0011111	101
3.75 V $\le v_{IN} < 4.375$ V	0111111	110
$4.375 \le v_{IN}$	1111111	111

Figure 29.24 Code transitions for the Flash ADC used in Ex. 29.10.

Based on Fig. 29.24, when $v_{IN} = 1.5$ V, only comparators C_1 and C_2 will have outputs of 1, since both V_1 and V_2 are less than 1.5 V. The remaining comparator outputs will be 0 since V_3 through V_8 will be greater than 1.5 V, thus generating the thermometer code, 0000011. The encoder must then convert this into a 3-bit digital word, resulting in 010. The same reasoning can be used to construct the data shown in Fig. 29.25. It should be obvious that if the polarity of the comparators were reversed, the thermometer code would be inverted. ■

v_{IN}	$C_7C_6C_5C_4C_3C_2C_1$	$D_2D_1D_0$
1.5	0000011	010
3.0	0001111	100
4.5	1111111	111

Figure 29.25 Output for the Flash ADC used in Ex. 29.10.

Accuracy Issues for the Flash ADC

Accuracy depends on the matching of the resistor string and the input offset voltage of the comparators. From our discussions earlier, we know that an ideal comparator should switch at the point at which the two inputs, v_+ and v_-, are the same potential. However, the offset voltage, V_{os}, prohibits this from occurring as the comparator output switches states as follows:

$$v_o = 1 \qquad \text{when } v_+ \ge v_- + V_{os} \tag{29.41}$$

$$v_o = 0 \qquad \text{when } v_+ < v_- + V_{os} \tag{29.42}$$

The resistor-string DAC was analyzed and presented in Sec. 29.1.2; the voltage on the i-th tap of the resistor string was found to be

$$V_i = V_{i,ideal} + \frac{V_{REF}}{2^N} \cdot \sum_{k=1}^{i} \frac{\Delta R_k}{R} \tag{29.43}$$

where $V_{i,ideal}$ is the voltage at the i-th tap if all the resistors had an ideal value of R. The term, ΔR_k, is the value of the resistance error (difference from ideal) due to the mismatch. Note that for the resistor-string DAC, the sum of the mismatch terms plays an important factor in the overall voltage at each tap.

The switching point for the i-th comparator, $V_{sw,i}$, then becomes

$$V_{sw,i} = V_i + V_{os,i} \tag{29.44}$$

where $V_{os,i}$ is the input referred offset voltage of the i-th comparator. The INL for the converter can then be described as

$$INL = V_{sw,i} - V_{sw,ideal} = V_{sw,i} - V_{i,ideal} \tag{29.45}$$

which becomes

$$INL = \frac{V_{REF}}{2^N} \cdot \sum_{k=1}^{i} \frac{\Delta R_k}{R} + V_{os,i} \tag{29.46}$$

The worst-case INL will occur at the middle of the string ($i = 2^{N-1}$), as described in Sec. 29.1.2 and Eq. (29.10). Including the offset voltage, the maximum INL will be

$$|INL|_{max} = \frac{V_{REF}}{2^N} \cdot \sum_{k=1}^{2^{N-1}} \frac{\Delta R_k}{R} + |V_{os,i}|_{max} = V_{REF} \cdot \frac{2^{N-1}}{2^N R} \cdot |\Delta R_k|_{max} + |V_{os,i}|_{max} \tag{29.47}$$

which can be rewritten as

$$|INL|_{max} = \frac{V_{REF}}{2} \cdot \left|\frac{\Delta R_k}{R}\right|_{max} + |V_{os,i}|_{max} \tag{29.48}$$

where it is assumed that the maximum positive mismatch occurs in all the resistors in the lower half of the string and the maximum negative mismatch occurs in the upper half (or vice versa) and that the comparator at the i-th tap contains the maximum offset voltage, $|V_{os,i}|_{max}$. Notice that the offset contributes directly to the maximum value for the INL. This explains another limitation to using Flash converters at high resolutions. The offset voltage alone can make the INL greater than ½ LSB.

Example 29.11

If a 10-bit Flash converter is designed, determine the maximum offset voltage of the comparators which will make the INL less than ½ LSB. Assume that the resistor string is perfectly matched and $V_{REF} = 5$ V.

Equation (29.48) requires that the offset voltage equal ½ LSB. Therefore,

$$|V_{os}|_{max} = \frac{5}{2^{11}} = 2.44 \text{ mV} \quad \blacksquare$$

The DNL calculation for the Flash converter is also attained using the analysis first presented in Sec. 29.1.2. Using the definition of DNL,

$$DNL = V_{sw,i} - V_{sw,i-1} - 1 \text{ LSB (in volts)} \tag{29.49}$$

Plugging in Eq. (29.44),

$$DNL = V_i + V_{os,i} - V_{i-1} - V_{os,i-1} - 1 \text{ LSB} \tag{29.50}$$

which can be written by using Eq. (29.6) as

$$DNL = V_{i,ideal} - V_{i-1,ideal} + \frac{V_{REF}}{2^N} \cdot \frac{\Delta R_i}{R} + V_{os,i} - V_{os,i-1} - 1\ LSB \qquad (29.51)$$

which becomes

$$DNL = \frac{V_{REF}}{2^N} \cdot \frac{\Delta R_i}{R} + V_{os,i} - V_{os,i-1} \qquad (29.52)$$

The maximum DNL will occur, assuming ΔR_i is at its maximum, $V_{os,i}$ is at its maximum positive value, and $V_{os,i-1}$ is at its maximum negative voltage. Thus,

$$|DNL|_{max} = \frac{V_{REF}}{2^N} \cdot \left|\frac{\Delta R_i}{R}\right|_{max} + 2|V_{os}|_{max} \qquad (29.53)$$

which assumes that the maximum offset voltage in the positive and negative directions are symmetrical. Therefore, both resistor-string matching and offset voltage affect the DNL of the converter.

29.2.2 The Two-Step Flash ADC

Another type of Flash converter is called the two-step Flash converter or the parallel, feed-forward ADC. The basic block diagram of a two-step converter is seen in Fig. 29.26. The converter is separated into two complete Flash ADCs with feed-forward circuitry. The first converter generates a rough estimate of the value of the input, and the second converter performs a fine conversion. The advantages of this architecture are that the number of comparators is greatly reduced from that of the Flash converter—from $2^N - 1$ comparators to $2(2^{N/2} - 1)$ comparators. For example, an 8-bit Flash converter requires 255 comparators, while the two-step Flash requires only 30. The trade-off is that the conversion process takes two steps instead of one, with the speed limited by the bandwidth and settling time required by the residue amplifier and the summer. The conversion process is as follows:

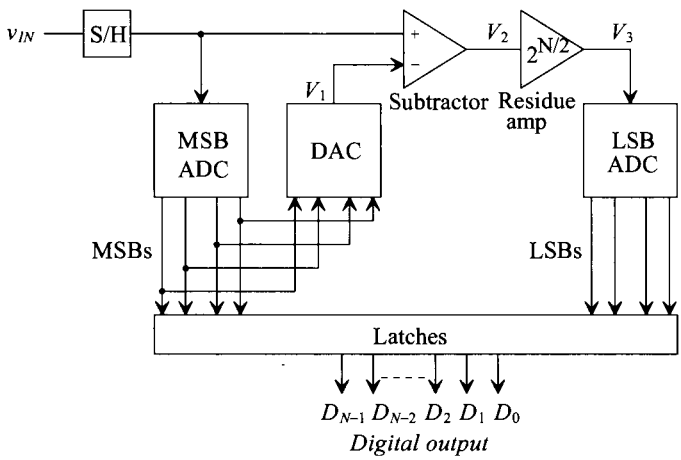

Figure 29.26 Block diagram of a two-step Flash ADC.

1. After the input is sampled, the most significant bits (MSBs) are converted by the first Flash ADC.
2. The result is then converted back to an analog voltage with the DAC and subtracted with the original input.
3. The result of the subtraction, known as the *residue*, is then multiplied by $2^{N/2}$ and input into the second ADC. The multiplication not only allows the two ADCs to be identical, but also increases the quantum level of the signal input into the second ADC.
4. The second ADC produces the least significant bits through a Flash conversion.

Some architectures use the same set of comparators in order to perform both steps. The multiplication mentioned in step 3 can be eliminated if the second converter is designed to handle very small input signals. The accuracy of the two-step ADC depends primarily on the linearity of the first ADC.

Figure 29.27 illustrates the two-step nature of the converter. A more intuitive approach can be explained with this picture. The first conversion identifies the segment in which the analog voltage resides. This is also known as a *coarse conversion* of the MSBs. The results of the coarse conversion are then multiplied by $2^{N/2}$ so that the segment within which V_{IN} resides will be scaled to the same reference as the first conversion. The second conversion is known as the *fine conversion* and will generate the final LSBs using the same Flash approach. One can see why the accuracy of the first converter is so important. If the input value is close to the boundary between two coarse segments and the first ADC is unable to choose the correct coarse segment, then the second conversion will be completely erroneous. The following example further illustrates the two-step algorithm.

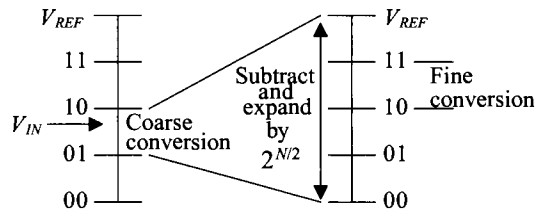

Figure 29.27 Coarse and fine conversions using a two-step ADC.

Example 29.12

Assume that the two-step ADC shown in Fig. 29.26 has four bits of resolution. Make a table listing the MSBs, V_1, V_2, V_3, and the LSBs for V_{IN} = 2, 4, 9, and 15 V assuming that V_{REF} = 16 V.

Since V_{REF} was conveniently made 16 V, each LSB will be 1 V. If V_{IN} = 2 V, the output of the first 2-bit Flash converter will be 00 since V_{REF} = 16 V and each resistor drops 4 V. The output of the 2-bit DAC, V_1, will therefore be 0, resulting in V_2 = 2 V. The multiplication of V_2 by the 4 results in V_3 = 8 V. Remember that

each 2-bit Flash converter resembles that of Fig. 29.21. The thermometer code from the second Flash converter will be 0011, which results in 10 as the LSBs. The other values can be calculated as seen in Fig. 29.28. ■

V_{IN}	D_3D_2 (MSBs)	V_1	V_2	V_3	D_1D_0 (LSBs)
2	00	0	2	8	10
4	01	4	0	0	00
9	10	8	1	4	01
15	11	12	3	12	11

Figure 29.28 Output for the Flash ADC used in Ex. 29.12.

Accuracy Issues Related to the Two-Step Flash Converters

As stated previously, the overall accuracy of the converter depends on the first ADC. The second Flash must have only the accuracy of a stand-alone Flash converter. This means that if an 8-bit, two-step Flash converter contains two 4-bit Flash converters, the second Flash needs only to have the resolution of a 4-bit Flash, which is not difficult to achieve. However, the first 4-bit Flash must have the accuracy of an 8-bit Flash, meaning that the worst-case INL and DNL for the first 4-bit Flash must be less than ±½ LSB for an 8-bit ADC. Thus, the resistor matching and comparators contained in the first ADC must possess the accuracy of the overall converter. Refer to Sec. 29.2.1 for derivations on INL and DNL for a Flash. The DAC must also be accurate to within the resolution of the ADC.

Accuracy Issues Related to the Operational Amplifiers

With the addition of the summer and the amplifier, other sources of accuracy errors are present in this converter. The summer and the amplifier must add and amplify the signal to within ±½ LSB of the ideal value. It is difficult to implement standard operational amplifiers within high-resolution data converters because of these accuracy requirements. The nonideal characteristics of the op-amp are well known and in many cases alone limit the accuracy of the data converter. In this case, the amplifier is required to multiply the residue signal by some factor of two. Although this may not seem difficult at first glance, a closer examination will reveal a dependency on the open-loop gain.

Suppose that the amplifier were being used in a 12-bit, two-step data converter. Remember that in order for a data converter to be N-bit accurate, the INL and DNL need to be kept below ±½ LSB and one-half of an LSB can be defined as

$$0.5 \text{ LSB} = \frac{V_{REF}}{2^{N+1}} \tag{29.54}$$

Since the output of the amplifier gets quantized to 6 bits, the amplifier would need to be 6-bit accurate to within ±½ LSB, resulting in an accuracy of

$$\text{Accuracy} = \frac{0.5\,LSB}{\text{Full scale range } (V_{REF})} = \frac{1}{2^{6+1}} = \frac{1}{128} = 0.0078 = 0.78\% \tag{29.55}$$

And suppose that a feedback amplifier with a gain of 64, or $2^{N/2}$, is used as the residue amplifier. The gain would need to be within the following range:

$$63.5 \text{ V/V} < A_{CL} < 64.5 \text{ V/V} \tag{29.56}$$

where A_{CL} is the closed-loop gain of the amplifier. Already, one can see the limitations of using operational amplifiers with feedback in high-accuracy applications. Designing an op-amp based amplifier with a high degree of gain accuracy can be difficult.

Generalizing this concept for an N-bit application requires knowledge of feedback theory discussed in Ch. 24. The closed-loop gain of the amplifier is expressed as

$$A_{CL} = \frac{v_o}{v_i} = \frac{A_{OL}}{1 + A_{OL}\beta} \tag{29.57}$$

where A_{OL} is the open-loop gain of the amplifier and β is the feedback factor. Also, from Ch. 24, it is known that as A_{OL} increases in value, the closed-loop gain, A_{CL}, approaches the value of $1/\beta$. Therefore, if it is assumed the closed-loop gain of the amplifier equals the ideal value of $1/\beta$ minus some maximum deviation from the ideal, ΔA, then,

$$A_{CL} = \frac{v_o}{v_i} = \frac{A_{OL}}{1 + A_{OL}\beta} = \frac{1}{\beta} - \Delta A \tag{29.58}$$

where $1/\beta$ is the desired value of the closed-loop gain (usually some factor of 2^N) and ΔA is the required accuracy ($\pm$½ LSB) of the gain (i.e., $(1/\beta) \cdot (1/2^{N+1})$). The right two terms of Eq. (29.58) can be solved for the open-loop gain of the amplifier,

$$|A_{OL}| = \frac{1}{\beta}(2^{N+1} - 1) \approx \frac{2^{N+1}}{\beta} \tag{29.59}$$

If the op-amp is used as a gain of 64 ($1/\beta$) and is required to amplify signals with 6-bit accuracy, then the open-loop gain of the amplifier must be at least $|A_{OL}| \geq 128 \cdot 64 = 8,192$ V/V. This is certainly an achievable specification. However, notice that for every bit increase in resolution, the open-loop gain requirement doubles. This is one reason two-step Flash converters are limited in resolution to approximately 12 bits (or less in a nanometer CMOS process).

The unity-gain frequency, f_{un} , required of an op-amp used in or with a data converter for a specific settling time t, (where $t < T_{clk}/2 = 1/2f_{clk}$) can be estimated assuming linear settling, and requiring the output of the op-amp be ½ LSB accurate, by

$$v_{out} = V_{outfinal}\left(1 - \frac{1}{2^{N+1}}\right) = V_{outfinal}(1 - e^{-t/\tau}) \text{ or } f_{un} \geq \frac{f_{clk} \cdot \ln 2^{N+1}}{\pi \cdot \beta} \tag{29.60}$$

This equation can be used to determine the minimum op-amp gain-bandwidth product (= f_{un}) needed to achieve a specific settling time provided the op-amp slew-rate doesn't come into play and the op-amp can be modeled as a first-order system.

Linearity of the amplifier is another aspect of amplifier performance that must be considered when designing ADCs. The amplifier must be able to linearly amplify the input signal over an input voltage range to within ½ LSB of the number bits that its output is quantized. If the amplifier is not designed correctly, nonlinearity is introduced as devices in the amplifier go into nonsaturation. Harmonic distortion occurs, resulting in an error within the ADC. Linearity is typically measured in terms of total harmonic distortion, or THD, (refer to Sec. 21.3.3). However, the transfer curve illustrates the

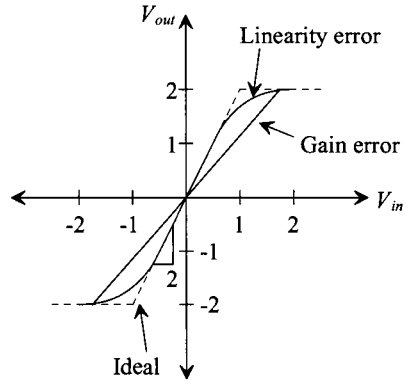

Figure 29.29 An op-amp transfer curve that distinguishes between gain error and linearity error.

limitation more effectively. Figure 29.29 shows a transfer curve of an op-amp with a gain of two. The ideal transfer curve is shown if the input range is known to be between –1 and 1 V. The actual transfer curve shows nonlinearity introduced at both ends of the input range. In order for the amplifier to be *N*-bit accurate, the slope of the actual transfer curve may not vary from the ideal by more than the accuracy required at the output of the amplifier. Note also in Fig. 29.29 the subtle difference between a gain error and nonlinearity. However, a gain error is much less harmful to an ADC's performance than harmonic distortion.

29.2.3 The Pipeline ADC

After examining the two-step ADC, one might wonder whether there is such a converter as a three-step or four-step ADC. In actuality, one could divide the number of conversions into many steps. The pipeline ADC is an *N*-step converter, with 1 bit being converted per stage. Able to achieve high resolution (10–13 bits) at relatively fast speeds, the pipeline ADC consists of *N* stages connected in series (Fig. 29.30). Each stage contains a 1-bit ADC (a comparator), a sample-and-hold, a summer, and a gain of two amplifier. Each stage of the converter performs the following operation:

1. After the input signal has been sampled, compare it to $\frac{V_{REF}}{2}$. The output of each comparator is the bit conversion for that stage.
2. If $v_{IN} > \frac{V_{REF}}{2}$ (comparator output is 1), $\frac{V_{REF}}{2}$ is subtracted from the held signal and pass the result to the amplifier. If $v_{IN} < \frac{V_{REF}}{2}$ (comparator output is 0), then pass the original input signal to the amplifier. The output of each stage in the converter is referred to as the *residue*.
3. Multiply the result of the summation by 2 and pass the result to the sample-and-hold of the next stage.

A main advantage of the pipeline converter is its high throughput. After an initial *latency* of *N* clock cycles, one conversion will be completed per clock cycle. While the residue of the first stage is being operated on by the second stage, the first stage is free to

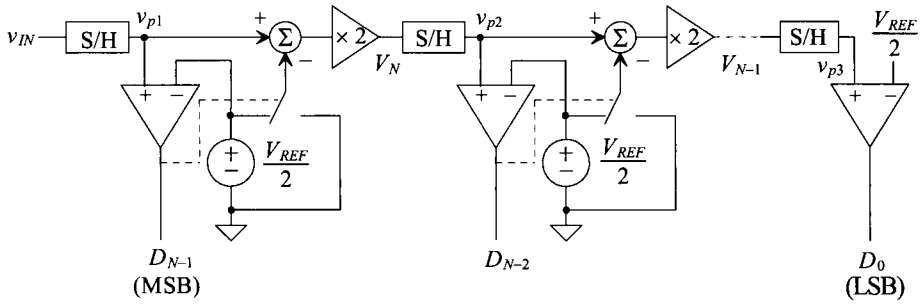

Figure 29.30 Block diagram of a pipeline ADC.

operate on the next samples. Each stage operates on the residue passed down from the previous stage, thereby allowing for fast conversions. The disadvantage is having the initial N clock cycle delay before the first digital output appears. The severity of this disadvantage depends, of course, on the application.

One interesting aspect of this converter is its dependency on the most significant stages for accuracy. A slight error in the first stage propagates through the converter and results in a much larger error at the end of the conversion. Each succeeding stage requires less accuracy than the one before, so special care must be taken when considering the first several stages.

Example 29.13

Assume that the pipeline converter shown in Fig. 29.30 is a 3-bit converter. Analyze the conversion process by making a table of the following variables: D_2, D_1, D_0, V_3, V_2, for $v_{IN} = 2$, 3, and 4.5 V. Assume that $V_{REF} = 5$ V, V_3 is the residue voltage out of the first stage, and V_2 is the residue voltage out of the second stage.

The output of the first comparator, $D_2 = 0$, since $v_{IN} < 2.5$ V. Since $D_2 = 0$, $V_3 = 2(2) = 4$ V. Passing this voltage down the pipeline, since $V_3 > 2.5$ V, $D_1 = 1$ and V_2 becomes

$$V_2 = \left(V_3 - \frac{V_{REF}}{2}\right) \times 2 = 3 \text{ V}$$

The LSB, $D_0 = 1$, since $V_2 > 2.5$ V, and the digital output corresponding to $V_{IN} = 2$ V, is $D_2D_1D_0 = 011$. The actual digital outputs are simply the comparator outputs, and the data can be completed as seen in Fig. 29.31. ■

v_{IN}	V_3 (V)	V_2 (V)	Digital Out ($D_2D_1D_0$)
2.0	4.0	3.0	011
3.0	1.0	2.0	100
4.5	4.0	3.0	111

Figure 29.31 Output for the pipeline ADC used in Ex. 29.13.

Accuracy Issues Related to the Pipeline Converter

The 1-bit per stage ADC can be analyzed by examining the switching point of each comparator for the ideal and nonideal case. Using Fig. 29.30, and assuming that all of the components are ideal, let $v_{IN,1}$ represent the value of the input voltage when the first comparator switches. This occurs when

$$v_{IN,1} = \frac{1}{2}V_{REF} \tag{29.61}$$

The positive input voltage on the second comparator, v_{p2}, can be written in terms of the previous stage, or

$$v_{p2} = [v_{IN} - \tfrac{1}{2} \cdot D_{N-1} \cdot V_{REF}] \cdot 2 \tag{29.62}$$

where D_{N-1} is the MSB output from the first comparator and is either a 1 or a 0. The second comparator switches when $v_{p2} = \frac{1}{2}V_{REF}$. The value of v_{IN} at this point, denoted as $v_{IN,2}$, is

$$v_{IN,2} = \tfrac{1}{2} \cdot D_{N-1} \cdot V_{REF} + \tfrac{1}{4}V_{REF} \tag{29.63}$$

Continuing on in a similar manner, we can write the value of the voltage on the positive input of the third comparator in terms of the previous two stages as

$$v_{p3} = \left[[v_{IN} - \frac{1}{2} \cdot D_{N-1} \cdot V_{REF}] \cdot 2 - [\frac{1}{2} \cdot D_{N-2} \cdot V_{REF}] \right] \cdot 2 \tag{29.64}$$

and the third comparator will switch when $v_{p3} = \frac{1}{2}V_{REF}$, which corresponds to the point at which v_{IN} becomes

$$v_{IN,3} = \tfrac{1}{2} \cdot D_{N-1} \cdot V_{REF} + \tfrac{1}{4} \cdot D_{N-2} \cdot V_{REF} + \tfrac{1}{8}V_{REF} \tag{29.65}$$

By now, a general trend can be recognized and the value of v_{IN} can be derived for the point at which the comparator of the N-th stage switches. This expression can be written as

$$v_{IN,N} = \tfrac{1}{2} \cdot D_{N-1} \cdot V_{REF} + \tfrac{1}{4} \cdot D_{N-2} \cdot V_{REF} + \tfrac{1}{8} \cdot D_{N-3} \cdot V_{REF} + \ldots\ldots + \frac{1}{2^{N-1}} \cdot D_1 \cdot V_{REF} + \frac{1}{2^N} \cdot V_{REF} \tag{29.66}$$

Notice that the preceding equation does not include D_0. This is because D_0 is the output of the N-th stage comparator.

Now that we have derived the switching points for the ideal case, the nonideal case can be considered. Only the major sources of error will be included in the analysis so as not to overwhelm the reader. These include the comparator offset voltage, $V_{COS,x}$, and the sample-and-hold offset voltage, $V_{SOS,x}$. The variable, x, represents the number of the stage for which each of the errors is associated, and the "prime" notation will be used to distinguish between the ideal and nonideal case. The reader should also be aware that the offset voltages can be of either polarity. It will be assumed that all of the residue amplifiers have the same gain, denoted as A.

The positive input to the first nonideal comparator, v'_{p1}, will include the offset from the first sample-and-hold, such that

$$v'_{p1} = v_{IN} + V_{SOS,1} \tag{29.67}$$

Now the first comparator will not switch until the voltage on the positive input overcomes the comparator offset as well. This occurs when

$$v'_{p1} = \tfrac{1}{2}V_{REF} + V_{COS,1} \tag{29.68}$$

Thus, equating Eqs. (29.67) and (29.68) and solving for the value of the input voltage when the switching occurs for the first comparator yields

$$v'_{IN,1} = \tfrac{1}{2}V_{REF} + V_{COS,1} - V_{SOS,1} \tag{29.69}$$

The input to the second comparator, v'_{p2}, can be written as

$$v'_{p2} = [v_{IN} + V_{SOS,1} - \tfrac{1}{2} \cdot D_{N-1} \cdot V_{REF}] \cdot A + V_{SOS,2} \tag{29.70}$$

and the value of input voltage at the point which the second comparator switches occurs when

$$v'_{IN,2} = \frac{1}{2} \cdot D_{N-1} \cdot V_{REF} + \frac{1}{2}\frac{V_{REF}}{A} - V_{SOS,1} - \frac{1}{A}(V_{SOS,2} - V_{COS,2}) \tag{29.71}$$

Continuing in the same manner, we can write the value of the input voltage that causes the third comparator to switch as

$$v'_{IN,3} = \tfrac{1}{2} \cdot D_{N-1} \cdot V_{REF} + \tfrac{1}{2} \cdot D_{N-2} \cdot \frac{V_{REF}}{A} - V_{SOS,1} - \tfrac{1}{A}V_{SOS,2} - \tfrac{1}{A^2}V_{SOS,3} - \tfrac{1}{A^2}\left[V_{COS,3} - \tfrac{1}{2}V_{REF}\right] \tag{29.72}$$

which can be generalized to the N-th switching point as

$$v'_{IN,N} = \tfrac{1}{2} \cdot D_{N-1} \cdot V_{REF} + \tfrac{1}{2} \cdot D_{N-2} \cdot \frac{V_{REF}}{A} + \ldots + \tfrac{1}{2} \cdot D_1 \cdot \frac{V_{REF}}{A^{N-2}} + \tfrac{1}{2} \cdot \frac{V_{REF}}{A^{N-1}} + \frac{V_{COS,N}}{A^{N-1}} - \sum_{k=1}^{N} \frac{V_{SOS,k}}{A^{k-1}} \tag{29.73}$$

The INL can be calculated by subtracting switching point between the nonideal and ideal case. Therefore, the INL of the first stage is found by subtracting Eqs. (29.69) and (29.61).

$$INL_1 = v'_{IN,1} - v_{IN,1} = V_{COS,1} - V_{SOS,1} \tag{29.74}$$

The second stage INL is

$$INL_2 = v'_{IN,2} - v_{IN,2} = \frac{V_{REF}}{2}\left(\frac{1}{A} - \frac{1}{2}\right) - V_{SOS,1} - \frac{V_{SOS,2}}{A} + \frac{V_{COS,2}}{A} \tag{29.75}$$

and the INL for the N-th stage is

$$INL_N = \tfrac{1}{2} \cdot D_{N-2} \cdot V_{REF} \cdot \left(\frac{1}{A} - \frac{1}{2}\right) + \tfrac{1}{2} \cdot D_{N-3} \cdot V_{REF} \cdot \left(\frac{1}{A^2} - \frac{1}{4}\right) + \ldots$$
$$+\tfrac{1}{2} \cdot D_1 \cdot V_{REF} \cdot \left(\frac{1}{A^{N-2}} - \frac{1}{2^{N-2}}\right) + \tfrac{1}{2} \cdot V_{REF} \cdot \left(\frac{1}{A^{N-1}} - \frac{1}{2^{N-1}}\right) + \frac{V_{COS,N}}{A^{N-1}} - \sum_{k=1}^{N} \frac{V_{SOS,K}}{A^{k-1}} \tag{29.76}$$

Equations (29.74)–(29.76) are very important to understanding the limitations of the pipeline ADC. Notice the importance of the comparator and summer offsets in Eq. (29.74). The worst-case addition of the offsets must be less than ½ LSB to keep the ADC N-bit accurate. The second stage is more dependent on the gain of the residue amplifier as seen in Eq. (29.75). The gain error discussed in the previous section plays an important role in determining the overall accuracy of the converter. Now examine the effects of the offsets on the INL of the N-th stage. In Eq. (29.76), both the comparator and summer offsets of the N-th stage (when $k = N$) are divided by a large gain. Therefore, the latter stages in a pipeline ADC are not as critical to the accuracy as the first stages, and die area and power can be reduced by using less accurate designs for the least significant stages. The summation term in Eq. (29.76) also reveals that the summer offset of the first stage ($k = 1$) has a large effect on the N-th stage. However, this point is inconsequential since $V_{SOS,1}$ must be minimized to achieve N-bit accuracy for the first stage anyway. Typically, if the INL and DNL specifications can be made N-bit accurate in the first few stages, the latter stages will not adversely affect overall accuracy.

The DNL can be found by calculating the difference between the worst-case switching points and subtracting the ideal value for an LSB. As defined earlier, the worst case will occur at midscale when the output switches from 0111...111 to 1000...000 as v_{IN} increases. Thus, the DNL is

$$DNL_{max} = v'_{IN,1} - v'_{IN,N} - \frac{V_{REF}}{2^N} \tag{29.77}$$

where $v'_{IN,N}$ is calculated using Eq. (29.73) and assuming that D_{N-1} is a zero and that all of the other bits are ones. Plugging in Eqs. (29.69) and (29.73) into Eq. (29.77) yields

$$DNL_{max} = \frac{1}{2}V_{REF}\left(1-\sum_{k=1}^{N-1}\frac{1}{A^k}\right) + V_{COS,1} - \frac{V_{COS,N}}{A^{N-1}} + \sum_{k=2}^{N}\frac{V_{SOS,K}}{A^{k-1}} - \frac{V_{REF}}{2^N} \tag{29.78}$$

Again, the term that dominates this expression is the comparator offset associated with the first stage and the summer offset of the second stage. The entire expression in Eq. (29.78) must be less than ½ LSB for the ADC to have N-bit resolution.

29.2.4 Integrating ADCs

Another type of ADC performs the conversion by integrating the input signal and correlating the integration time with a digital counter. Known as single- and dual-slope ADCs, these types of converters are used in high-resolution applications but have relatively slow conversions. However, they are very inexpensive to produce and are commonly found in slow-speed, cost-conscious applications.

Single-Slope Architecture

Figure 29.32 illustrates the single-slope converter in block level form. A counter determines the number of clock pulses that are required before the integrated value of a reference voltage is equal to the sampled input signal. The number of clock pulses is proportional to the actual value of the input, and the output of the counter is the actual digital representation of the analog voltage.

Since the reference is a DC voltage, the output of the integrator should start at zero and linearly increase with a slope that depends on the gain of the integrator. Notice that the reference voltage is defined as negative so that the output of the inverting

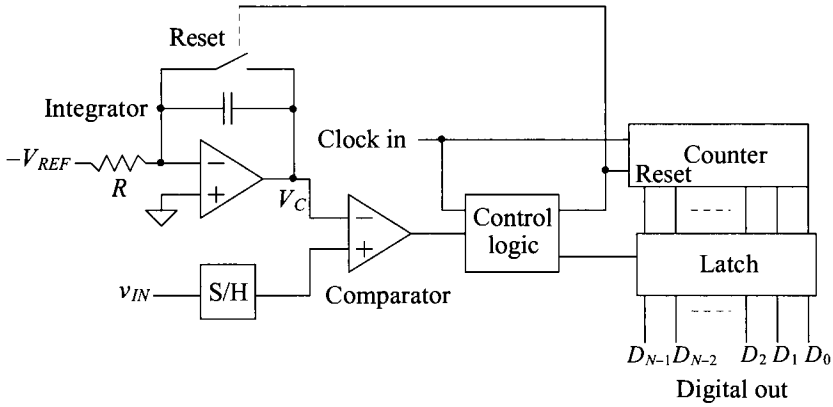

Figure 29.32 Block diagram of a single-slope ADC.

integrator is positive. At the time when the output of the integrator surpasses the value of the S/H output, the comparator switches states, thus triggering the control logic to latch the value of the counter. The control logic also resets the system for the next sample. Figure 29.33 illustrates the behavior of the integrator output and the clock.

Note that if the input voltage is very small, the conversion time is very short, as the counter has to increment only a few times before the comparator latches the data. However, if the input voltage is at its full-scale value, the counter must increment to its maximum value of 2^N clock cycles. Thus, the clock frequency must be many times faster than the bandwidth of the input signal. The conversion time, t_c, depends on the value of the input signal and can be described as

$$t_c = \frac{v_{IN}}{V_{REF}} \cdot 2^N \cdot T_{CLK} \tag{29.79}$$

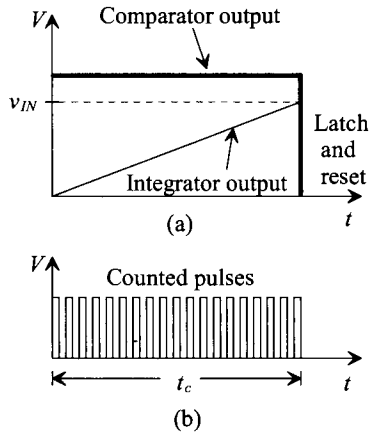

Figure 29.33 Single-slope ADC timing diagrams for (a) the comparator inputs and outputs and (b) the resulting counted pulses.

where T_{CLK} is the period of the clock. The sampling rate is inversely proportional to the conversion time and can be written as

$$f_{sample} = \frac{V_{REF}}{V_{IN} \cdot 2^N} \cdot f_{CLK} \tag{29.80}$$

Example 29.14

Determine the clock frequency needed to form an 8-bit, single-slope converter, if the analog signal bandwidth is 20 kHz.

Since the sampling rate required is 40 kHz, then the worst-case situation would occur for a full-scale input, in which event the integrator output would have to climb to its maximum value and the counter would increment 2^N times during the corresponding 25 μs period between samples. Therefore, the clock frequency would need to be 2^N times faster than the sampling rate or 10.24 MHz. ■

Accuracy Issues Related to the Single-Slope ADC

Obviously, many potential error sources abound in this architecture. At the end of the conversion, the voltage across the integrating capacitor, V_C, assuming no initial condition, will be

$$V_C = \frac{1}{C}\int_0^{t_c} \frac{V_{REF}}{R} dt = \frac{V_{REF} \cdot t_c}{RC} \tag{29.81}$$

where t_c is the conversion time. Plugging Eq. (29.79) into Eq. (29.81) yields

$$V_C = \frac{2^N \cdot T_{CLK} \cdot v_{IN}}{RC} = \frac{2^N \cdot v_{IN}}{f_{CLK} \cdot RC} \tag{29.82}$$

Equation (29.82) is a revealing one in that the final voltage on the integrator output depends not only on the value of the input voltage, which is to be expected, but also on the value of R, C, and f_{CLK}. Therefore, any nonideal effects affecting these values will have an influence on the accuracy of the integrator output from sample to sample. For example, if an integrated diffused-resistor is used, then the voltage coefficient of the resistor could limit the accuracy, since the resistor will be effectively nonlinear. Similarly, the capacitor may have charge leakage or aging effects associated with it. Also, any jitter in the clock will affect the overall accuracy. The integrator must have a linear slope to within the accuracy of the converter, which depends on the specifications of the op-amp (open- loop gain, settling time, offset, etc.) and must be considered accordingly.

Offset voltages on the comparator, the S/H, or the integrator result in additional or fewer clock pulses, depending on the polarity of the offset. A delay also exists from the time that the inputs to the comparator are equal and the time that the output of the counter is actually latched. The reference voltage must also stay constant to within the accuracy of the converter.

Dual-Slope Architecture

A slightly more sophisticated design known as the dual-slope, integrating ADC (Fig. 29.34) eliminates most of the problems encountered when using the single-slope converter. Here, two integrations are performed, one on the input signal and one on V_{REF}. The input voltage in this case is assumed to be negative, so that the output of the inverting

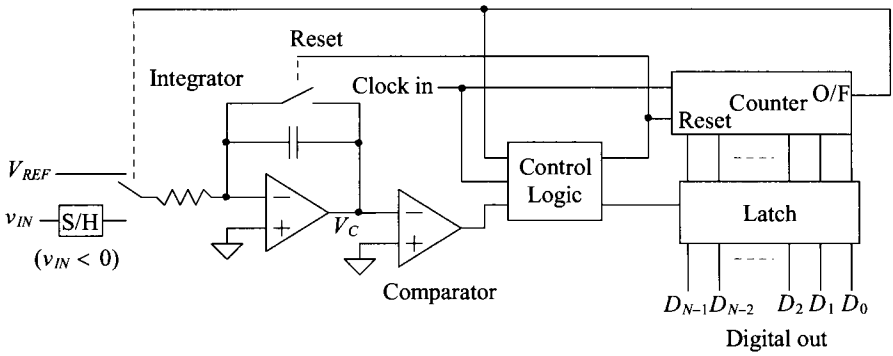

Figure 29.34 Block diagram of a dual-slope ADC.

integrator results in a positive slope during the first integration. Figure 29.35 illustrates the behavior for two separate samples. The first integration is of fixed length, dictated by the counter, in which the sample-and-held signal is integrated, resulting in the first slope. After the counter overflows and is reset, the reference voltage is connected to the input of the integrator. Since v_{IN} was negative and the reference voltage is positive, the inverting integrator output begins discharging back down to zero at a constant slope. A counter again measures the amount of time for the integrator to discharge, thus generating the digital output.

For Fig. 29.35, a 3-bit ADC is being used. Thus, the first integration period continues until the beginning of the eighth (2^3) clock pulse, which corresponds to the counter's overflow bit. Note that the integrator's output corresponding to V_B is twice the value of the output corresponding to V_A. Thus, it requires twice as many clock pulses for

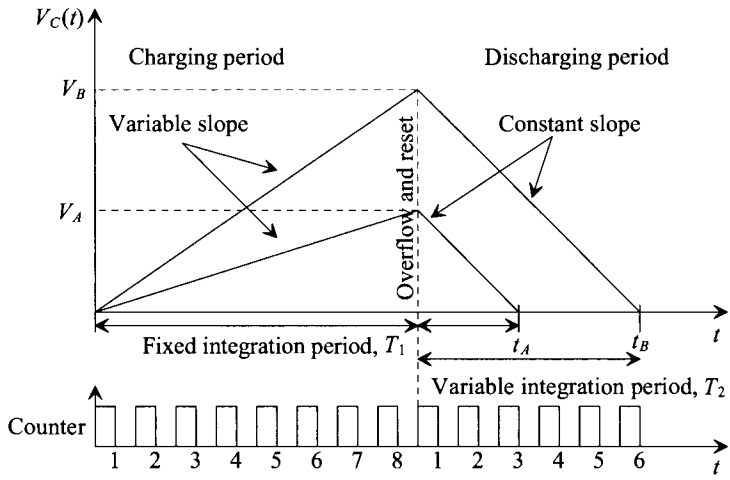

Figure 29.35 Integration periods and counter output for two separate samples of a 3-bit dual-slope ADC.

the integrator to discharge back to zero from V_B than from V_A. The output of the counter at t_A is three or 011, while the counter output at t_B is twice that value or six (110) and the quantization is complete.

Notice that the first slope varies according to the value of the input signal, while the second slope, dependent only on V_{REF}, is constant. Similarly, the time required to generate the first slope is constant, since it is limited by the size of the counter. However, the discharging period is variable and results in the digital representation of the input voltage.

Accuracy Issues Related to the Dual-Slope ADC

One may wonder how the dual-slope converter is an improvement over the single-slope architecture, since a significantly longer conversion time is required. The first integration period requires a full 2^N clock cycle and cannot be decreased, because the second integration might require the full 2^N clock cycles to discharge if the maximum value of v_{IN} is being converted. However, the dual slope is the preferred architecture because the same integrator and clock are used to produce both slopes. Therefore, any nonidealities will essentially be canceled. For example, assuming that the S/H is ideal, the gain of the integrator at the end of the first integration period, T_1, becomes

$$V_C = -\frac{1}{C}\int_0^{T_1} \frac{v_{IN}}{R} dt = \frac{|v_{IN}| \cdot T_1}{RC} \tag{29.83}$$

The output at the end of T_1 is positive since the input voltage is considered to be negative and the integrator is inverting. After the clock has been reset, the discharging commences, with the initial condition defined by the value of the integrator output at the end of the charging period, or

$$V_C = \frac{|v_{IN}| \cdot T_1}{RC} - \frac{1}{C}\int_0^{T_2} \frac{V_{REF}}{R} dt \tag{29.84}$$

Once the value of the integrator output, V_C, reaches zero volts, Eq. (29.84) becomes

$$V_C = \frac{|v_{IN}| \cdot T_1}{RC} - \frac{V_{REF} \cdot T_2}{RC} = 0 \tag{29.85}$$

or,

$$|v_{IN}| \cdot T_1 = V_{REF} \cdot T_2 \tag{29.86}$$

At the end of the conversion, the dependencies on R and C have canceled out. Since we also know that the counter increments 2^N times at time, T_1, and the counter increments D times at time, T_2, Eq. (29.86) can be rewritten as

$$\frac{D}{2^N} = \frac{|v_{IN}|}{V_{REF}} \tag{29.87}$$

where D is the counter output that is actually the digital representation of the input voltage. Thus, it can be written that the ratio of the input voltage and the reference voltage is proportional to the ratio of the binary value of the digital word, D, and 2^N. Therefore, since the same clock pulse is responsible for the charging and discharging times, any irregularities will also cancel out.

29.2.5 The Successive Approximation ADC

The successive approximation converter performs basically a binary search through all possible quantization levels before converging on the final digital answer. The block diagram is seen in Fig. 29.36. An N-bit register controls the timing of the conversion where N is the resolution of the ADC. V_{IN} is sampled and compared to the output of the DAC. The comparator output controls the direction of the binary search, and the output of the successive approximation register (SAR) is the actual digital conversion. The successive approximation algorithm is as follows.

1. A 1 is applied to the input of the shift register. For each bit converted, the 1 is shifted to the right 1-bit position. $B_{N-1} = 1$ and B_{N-2} through $B_0 = 0$.
2. The MSB of the SAR, D_{N-1}, is initially set to 1, while the remaining bits, D_{N-2} through D_0, are set to 0.
3. Since the SAR output controls the DAC and the SAR output is 100...0, the DAC output will be set to $\frac{V_{REF}}{2}$.
4. Next, v_{IN} is compared to $\frac{V_{REF}}{2}$. If $\frac{V_{REF}}{2}$ is greater than v_{IN}, then the comparator output is a 0 and the comparator resets D_{N-1} to 0. If $\frac{V_{REF}}{2}$ is less than v_{IN}, then the comparator output is a 1 and the D_{N-1} remains a 1. D_{N-1} is the actual MSB of the final digital output code.
5. The 1 applied to the shift register is then shifted by one position so that $B_{N-2} = 1$, while the remaining bits are all 0.
6. D_{N-2} is set to a 1, D_{N-3} through D_0 remain 0, while D_{N-1} remains the value from the MSB conversion. The output of the DAC will now either equal $\frac{V_{REF}}{4}$ (if $D_{N-1} = 0$) or $\frac{3V_{REF}}{4}$ (if $D_{N-1} = 1$).

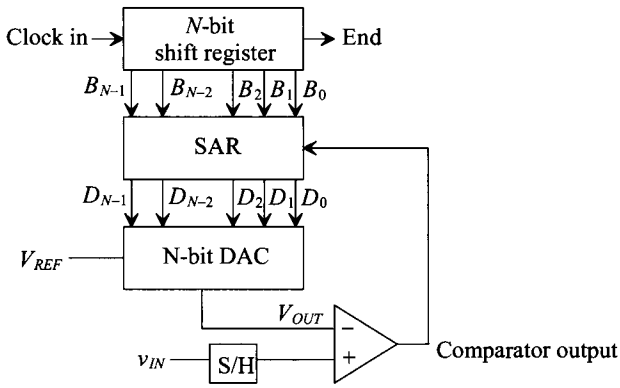

Figure 29.36 Block diagram of the successive approximation ADC.

7. Next, v_{IN} is compared to the output of the DAC. If the DAC output is greater than v_{IN} then the comparator output drives D_{N-2} to 0. If the DAC output is less than v_{IN} then D_{N-2} remains a 1.

8. The process repeats until the output of the DAC converges to the value of v_{IN} within the resolution of the converter.

Figure 29.37 shows an example of the binary search nature of the converter. The bolded line shows the path of the conversion for 101, corresponding to $\frac{5}{8}V_{REF}$. All possible quantization levels are represented in the binary tree. With each bit decided, the search space decreases by one-half until the correct answer is converged upon.

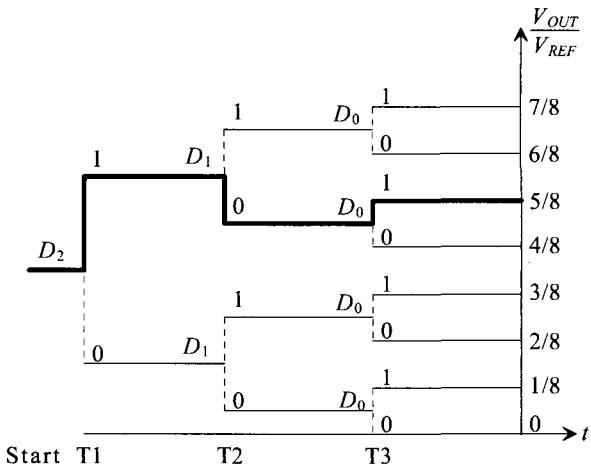

Figure 29.37 Binary search performed by a 3-bit successive approximation ADC for D=101.

Example 29.15

Perform the operation of a 3-bit successive approximation ADC similar to Fig. 29.36 with V_{REF} = 8. Make a table that consists of $D_2D_1D_0$, $B_2B_1B_0$, V_{OUT} (the output from the DAC) and the comparator output, which shows the binary search algorithm of the converter for v_{IN} = 5.5 V and 2.5 V.

We will designate $D_2.D_1.D_0.$ as the initial output of the SAR before the comparator makes its decision. The final value is designated as $D_2D_1D_0$. Notice that if the comparator is a 1, $D_2.D_1.D_0.$ differs from $D_2D_1D_0$, but if the comparator outputs a 0, then $D_2.D_1.D_0. = D_2D_1D_0$. The output of the shift register is designated as $B_2B_1B_0$.

Following the algorithm discussed previously, initially v_{IN} = 5.5 V and is compared with 4 V. Since the comparator output is 0, the MSB remains a 1. The next bit is examined, and the output of the DAC is now 6 V. Since $V_{OUT} > v_{IN}$, the comparator output is 1, which resets the current SAR bit, D_1, to a 0 at the end of period T2. Lastly, the LSB is examined, and v_{IN} is compared with 5 V. Since $v_{IN} > V_{OUT}$, the comparator output is a 0, and the current SAR bit, D_0, remains a 1. The

results can be examined in Fig. 29.38a. The final value for $D_2D_1D_0$ is 101, which is what is expected considering that 101 in binary is equivalent to 5_{10}. Figure 29.38b shows the data for the ADC using v_{IN} = 2.5 V. The final value for v_{IN} = 2.5 is 010, which again is what is expected for 3-bit resolution. ■

Step	v_{IN}	$B_2B_1B_0$	$D_2D_1D_0$	V_{OUT}	Comp Out	$D_2D_1D_0$
T1	5.5	100	100	$1/2\ V_{REF} = 4$ V	0	100
T2	5.5	010	110	$(1/2+1/4)V_{REF} = 6$ V	1	100
T3	5.5	001	101	$(1/2+1/8)V_{REF} = 5$ V	0	101

(a)

Step	v_{IN}	$B_2B_1B_0$	$D'_2D'_1D'_0$	V_{OUT}	Comp Out	$D_2D_1D_0$
T1	2.5	100	100	$1/2\ V_{REF} = 4$ V	1	000
T2	2.5	010	010	$1/4\ V_{REF} = 2$ V	0	010
T3	2.5	001	011	$(1/4+1/8)V_{REF} = 3$ V	1	010

(b)

Figure 29.38 Results from the 3-bit successive approximation ADC using (a) v_{IN} = 5.5 and (b) 2.5 V.

The successive approximation ADC is one of the most popular architectures used today. The simplicity of the design allows for both high speed and high resolution while maintaining relatively small area. The limit to the ADC's accuracy depends mainly on the accuracy of the DAC. If the DAC does not produce the correct analog voltage with which to compare the input voltage, the entire converter output will contain an error. Referring again to Fig. 29.37, we can see that if a wrong decision is made early, a massive error will result as the converter attempts to search for the correct quantization level in the wrong half of the binary tree.

The Charge-Redistribution Successive Approximation ADC

One of the most popular types of successive approximation architectures uses the binary-weighted capacitor array (analyzed in Sec. 29.1.5) as its DAC. Called a charge-redistribution, successive-approximation ADC, this converter samples the input signal and then performs the binary search based on the amount of charge on each of the DAC capacitors. Figure 29.39 shows an N-bit architecture. A comparator has replaced the unity gain buffer used in the DAC architecture. The binary-weighted capacitor array also samples the input voltage, so no external sample-and-hold is needed.

The conversion process begins by discharging the capacitor array, via the reset switch. Although this may appear to be an insignificant action, the converter is also performing automatic offset cancellation. Once the reset switch is closed, the comparator acts as a unity gain buffer. Thus, the capacitor array charges to the offset voltage of the comparator. This requires that the comparator is designed to be unity-gain stable, which means that internal compensation may have to be switched in during the reset period. Next, the input voltage, v_{IN}, is sampled onto the capacitor array. The reset switch is still closed, for the top plate of the capacitor array needs to be connected to virtual ground of the unity gain buffer. The equivalent circuit is seen in Fig. 29.40a. The reset switch is

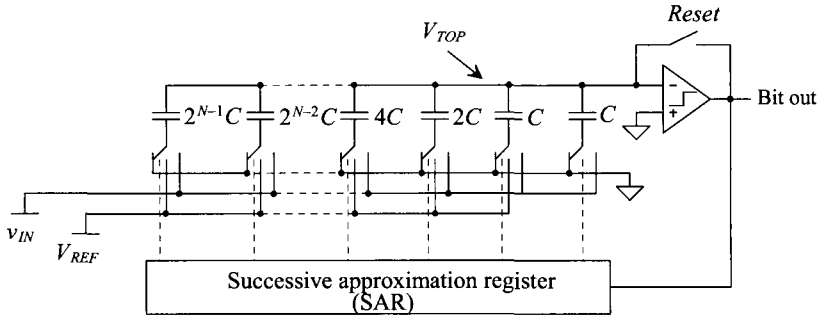

Figure 29.39 A charge redistribution ADC using a binary-weighted capacitor array DAC.

then opened, and the bottom plates of each capacitor in the array are switched to ground, so that the voltage appearing at the top plate of the array is now $V_{OS} - v_{IN}$ (Fig. 29.40b). The conversion process begins by switching the bottom plate of the MSB capacitor to V_{REF} (Fig. 29.40c). If the output of the comparator is high, the bottom plate of the MSB capacitor remains connected to V_{REF}. If the comparator output is low, the bottom plate of the MSB is connected back to ground. The output of the comparator is D_{N-1}. The voltage at the top of the capacitor array, V_{TOP}, is now

$$V_{TOP} = -v_{IN} + V_{OS} + D_{N-1} \cdot \frac{V_{REF}}{2} \tag{29.88}$$

The next largest capacitor is tested in the same manner as seen in Fig. 29.40d. The voltage at the top plate of the capacitor after the second capacitor is tested becomes

$$V_{TOP} = -v_{IN} + V_{OS} + D_{N-1} \cdot \frac{V_{REF}}{2} + D_{N-2} \cdot \frac{V_{REF}}{4} \tag{29.89}$$

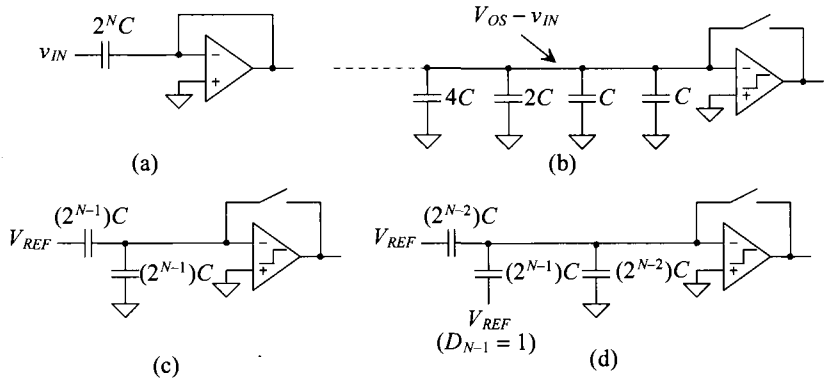

Figure 29.40 The charge redistribution process: (a) Sampling the input while autozeroing the offset, (b) the voltage at the top plate after sampling, (c) the equivalent circuit while converting the MSB, and (d) the equivalent circuit while converting the next largest capacitor with the MSB result equal to one.

The conversion process continues on with the remaining capacitors so that the voltage on the top plate of the array, V_{TOP}, converges to the value of the offset voltage, V_{OS} (within the resolution of the converter), or

$$V_{TOP} = -v_{IN} + V_{OS} + D_{N-1} \cdot \frac{V_{REF}}{2} + D_{N-2} \cdot \frac{V_{REF}}{4} + ... + D_1 \cdot \frac{V_{REF}}{2^{N-1}} + D_0 \cdot \frac{V_{REF}}{2^N} \approx V_{OS} \qquad (29.90)$$

Note that the initial charge stored on the capacitor array is now redistributed onto only those capacitors that have their bottom plates connected to V_{REF}.

Accuracy Issues Related to the Charge-Redistribution, Successive-Approximation ADC

Obviously, the limitation of this architecture is the capacitor matching. The mismatch is analyzed in the same manner as the binary-weighted current source array of Sec. 29.1.4. Thus, substituting the value of the unit capacitance, C, for the value of the unit current source, I, and using Eqs. (29.27) – (29.30),

$$|INL|_{max} = \frac{2^{N-1} \cdot V_{REF} \cdot (C + |\Delta C|_{max,INL})}{2^N \cdot C} - \frac{2^{N-1} \cdot V_{REF} \cdot C}{2^N \cdot C} = \frac{V_{REF}}{2} \cdot \frac{|\Delta C|_{max,INL}}{C} \qquad (29.91)$$

where the maximum ΔC that will result in an INL that is ½ LSB is

$$|INL|_{max} = \frac{V_{REF}}{2} \cdot \frac{|\Delta C|_{max,INL}}{C} = \frac{V_{REF}}{2^{N+1}} = \frac{1}{2} \text{ LSB} \rightarrow |\Delta C|_{max,INL} = \frac{C}{2^N} \qquad (29.92)$$

The DNL is defined by

$$DNL_{max} = \frac{(2^N - 1) \cdot V_{REF} |\Delta C|_{max,DNL}}{2^N \cdot C} \qquad (29.93)$$

with the maximum ΔC, which results in a DNL less than ½ LSB:

$$DNL_{max} = \frac{(2^N - 1) \cdot V_{REF} |\Delta C|_{max,DNL}}{2^N \cdot C} = \frac{V_{REF}}{2^{N+1}} = \frac{1}{2} \text{ LSB} \rightarrow |\Delta C|_{max,DNL} = \frac{C}{2^{N+1} - 2} \qquad (29.94)$$

29.2.6 The Oversampling ADC

ADCs can be separated into two categories depending on the rate of sampling. The first category samples the input at the Nyquist rate, or $f_N = 2F$ where F is the bandwidth of the signal and f_N is the sampling rate. The second type samples the signal at a rate much higher than the signal bandwidth. This type of converter is called an oversampling converter.

The oversampling ADC is able to achieve much higher resolution than the Nyquist rate converters. This is because digital signal processing techniques are used in place of complex and precise analog components. The accuracy of the converter does not depend on the component matching, precise sample-and-hold circuitry, or trimming, and only a small amount of analog circuitry is required. Switched-capacitor implementations are easily achieved, and, as a result of the high sampling rate, only simplistic anti-aliasing circuitry needs to be used. However, because of the amount of time required to sample the input signal, the throughput is considerably less than the Nyquist rate ADCs.

Differences in Nyquist Rate and Oversampling ADCs

The typical process used in analog-to-digital conversion is seen in Fig. 29.41a, while the block diagram for the oversampling ADC is seen in Fig. 29.41b. After filtering the signal to help minimize aliasing effects, the signal is sampled, quantized, and encoded or decoded using simple digital logic to provide the digital data in the proper format. When using oversampling ADCs, little if any, anti-alias filtering is needed, no dedicated S/H is required, the quantization is performed with a modulator, and the encoding usually takes the form of a digital filter.

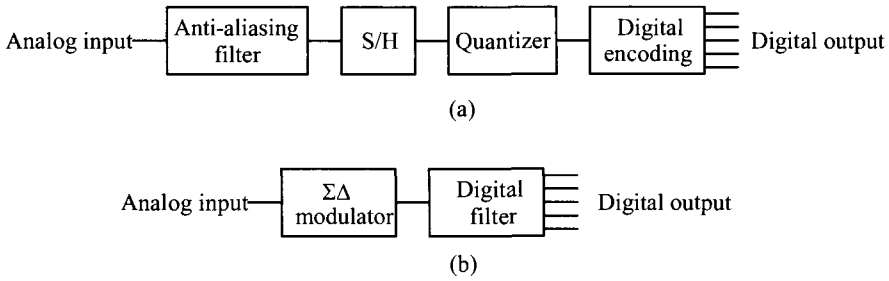

Figure 29.41 Typical block diagram for (a) Nyquist rate converters and (b) oversampling ADCs.

Since the oversampling converter samples the signal bandwidth at many times, aliasing is not a serious problem. A discussion of the frequency characteristics of aliasing was presented in Ch. 28. Figure 29.42a shows that when using Nyquist rate converters, a sampled signal in the frequency domain appears as a series of band-limited signals at multiples of the sampling frequency (see Fig. 28.26 for more details). As the sampling frequency decreases, the frequency spectra begin to overlap, and aliasing (Fig. 29.42b) occurs. Complex, "brickwall" filters are needed to correct the problem.

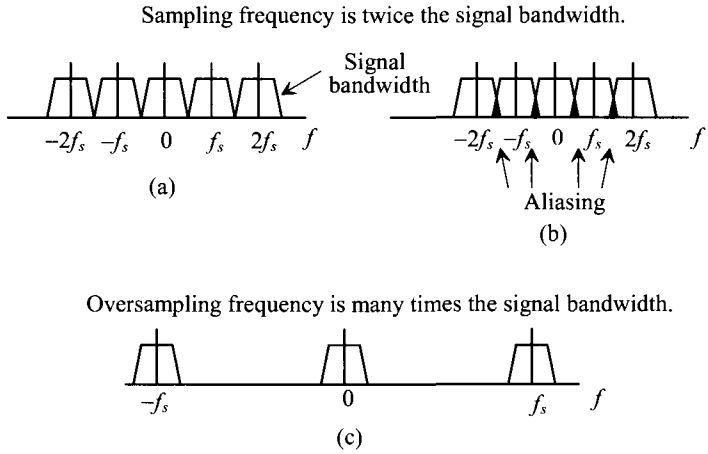

Figure 29.42 Frequency domain for (a) Nyquist rate converters, (b) the aliasing that occurs, and (c) an oversampling converter.

For oversampled ADCs, aliasing becomes much less of a factor. Since the sampling rate is much greater than the bandwidth of the signal, the frequency domain representation shows that the spectra are widely spaced, as seen in Fig. 29.42c. Therefore, overlapping of the spectra, and thus aliasing, will not occur, and only simple, first-order filters are required.

Oversampling converters typically employ switched-capacitor circuits and therefore do not need sample-and-hold circuits. The output of the modulator is a pulse-density modulated signal that represents the average of the input signal. The modulator constructs these pulses in real time, and so it is not necessary to hold the input value and perform the conversion.

As stated previously, the modulator actually provides the quantization in the form of a pulse-density modulated signal. Referred to as sigma-delta ($\Sigma\Delta$) or delta-sigma ($\Delta\Sigma$) modulation, the density of the pulses represents the average value of the signal over a specific period. Figure 29.43 illustrates the output of the modulator for the positive half of a sine wave input. Note that for the peak of the sine wave, most of the pulses are high. As the sine wave decreases in value, the pulses become distributed between high and low according to the sine wave value.

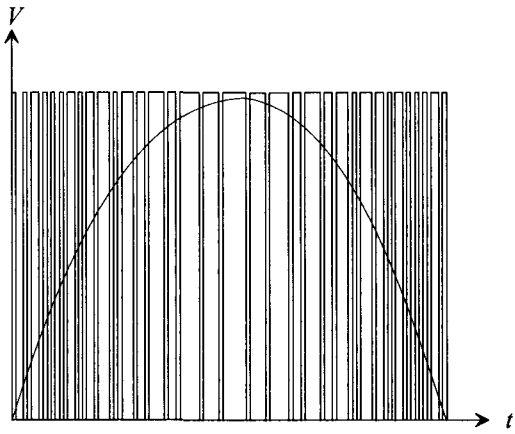

Figure 29.43 Pulse-density output from a sigma-delta modulator for a sine wave input.

If the frequency of the sine wave represented the highest frequency component of the input signal, a Nyquist rate converter would take only two samples. The oversampling converter, however, may take hundreds of samples over the same period to produce this pulse-density signal.

Digital signal processing is then used, which has two purposes: to filter any out-of-band quantization noise and to attenuate any spurious out-of-band signals. The output of the filter is then downsampled to the Nyquist rate so that the resulting output of the ADC is the digital data. This data represents the average value of the analog voltage over the oversampling period. The effective resolution of oversampling converters is determined by the values of signal-to-noise ratio and dynamic range obtained.

The First-Order ΣΔ Modulator

Now that the basic function of the ΣΔ modulator has been described, it would be useful to examine its inner workings and determine why ΣΔ modulation is so beneficial for generating high-resolution data. A basic first-order ΣΔ modulator can be seen in Fig. 29.44. Here, an integrator and a 1-bit ADC are in the forward path, and a 1-bit DAC is in the feedback path of a single-feedback loop system. The variables labeled are in terms of time, T, which is the inverse of the sampling frequency and k, which is an integer. The 1-bit ADC is simply a comparator that converts an analog signal into either a high or a low. The 1-bit DAC uses the comparator output to determine if $+V_{REF}$ or $-V_{REF}$ is summed with the input.

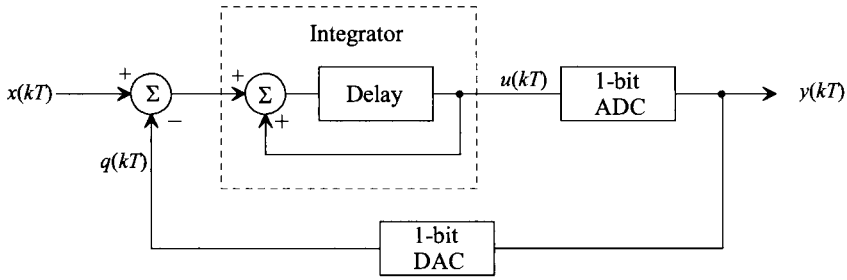

Figure 29.44 A first-order sigma-delta modulator.

While the benefits of ΣΔ modulation are not obvious, a simple derivation of the output, $y(kT)$, illuminates its distinct advantages. The output of the integrator, $u(kT)$, can be described as

$$u(kT) = x(kT - T) - q(kT - T) + u(kT - T) \tag{29.95}$$

where, $x(kT - T) - q(kT - T)$ is equal to the integrator's previous input, and $u(kT - T)$ is its previous output. The quantization error for the 1-bit ADC, as discussed in Ch. 28, is again defined as the difference between its output and input such that

$$Q_e(kT) = y(kT) - u(kT) \tag{29.96}$$

Plugging Eq. (29.95) into Eq. (29.96), the output response, $y(kT)$ is

$$y(kT) = Q_e(kT) + x(kT - T) - q(kT - T) + u(kT - T) \tag{29.97}$$

An ideal 1-bit DAC has the following characteristic: if the input, $y(kT) = 0$, the output, $q(kT) = -V_{REF}$, and if $y(kT) = 1$, then $q(kT) = V_{REF}$. In reality, a 1-bit DAC consists of a couple of switches connecting V_{REF} or $-V_{REF}$ to a common node, so it is not difficult to assume that the DAC is ideal. Therefore,

$$y(kT) = q(kT) \tag{29.98}$$

Utilizing Eq. (29.96) and Eq. (29.97), we find that Eq. (29.98) becomes

$$y(kT) = x(kT - T) + Q_e(kT) - Q_e(kT - T) \tag{29.99}$$

Therefore, the output of the modulator consists of a quantized value of the input signal delayed by one sample period, plus a differencing of the quantization error between the present and previous values. Thus, the real power of ΣΔ modulation is that the quantization noise, Q_e, cancels itself out to the first order.

A frequency domain example further illuminates this important fact. Suppose that the first-order modulator can be modeled in the s domain, as seen in Fig. 29.45, with an ideal integrator represented with transfer function of $\frac{1}{s}$, the 1-bit ADC modeled as a simple error source, $Q_e(s)$, and again the DAC considered to be ideal, such that $y(s)$ is equal to $q(s)$. It is also assumed that the bandwidth of the input signal is much less than the bandwidth of the modulator. Therefore, using simple feedback theory, $v_{OUT}(s)$ becomes

$$v_{OUT}(s) = Q_e(s) + \frac{1}{s} \cdot [v_{IN}(s) - v_{OUT}(s)] \tag{29.100}$$

and solving for v_{OUT} yields,

$$v_{OUT}(s) = Q_e(s) \cdot \frac{s}{s+1} + v_{IN}(s)\frac{1}{s+1} \tag{29.101}$$

Note that the transfer function from v_{IN} to v_{OUT} follows that of a low-pass filter and that the transfer function of the quantization noise follows that of a high-pass filter. Plotted together in Fig. 29.46, it is seen that in the region where the signal is of interest, the noise has a small value while the signal has a high gain, and that at higher frequencies, beyond the bandwidth of the signal, the noise increases. The modulator has essentially pushed the power of the noise out of the bandwidth of the signal. This high-pass characteristic is known as *noise shaping* and is a powerful concept used within oversampling ADCs. Low-pass filtering is then performed by the digital filter in order to remove all of the out-of-band quantization noise, which then permits the signal to be downsampled to yield the final high-resolution output.

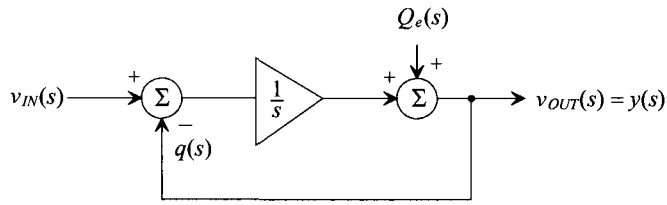

Figure 29.45 A frequency domain model for the first-order sigma-delta modulator.

As the ΣΔ modulator is generating the pulse-density modulated output, it is interesting to examine the mechanics occurring in the loop, which result in an average of the input. An actual ΣΔ modulator might resemble Fig. 29.47. A switched-capacitor integrator provides the summing as well as the delay needed. The 1-bit ADC is a simple comparator, and the 1-bit DAC is simply two voltage-controlled switches that select either V_{REF} or $-V_{REF}$ to be summed with the input. A latched comparator provides the necessary loop delay. Notice that the variables are voltage representations of the variables

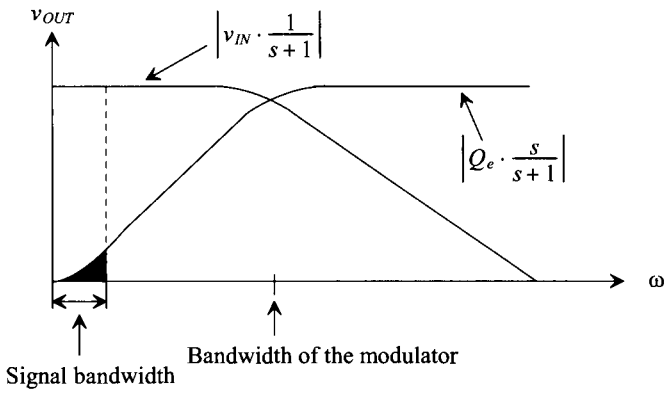

Figure 29.46 Frequency response of the first-order sigma-delta modulator.

used in Fig. 29.44. Remember that the function of the integrator is to accumulate differences between the input signal and the output of the DAC. If it is assumed that the input, $v_x(kT)$, is a positive DC voltage, then the output of the integrator should increase. However, the feedback mechanism is such that the 1-bit ADC (the comparator) has a low output if the integrator output, $v_u(kT)$, is positive. Thus, V_{REF} appears at the output of the DAC and is subtracted from the input, and the integrator output is driven back toward zero. The opposite occurs when $v_u(kT)$ is negative such that the integrator output is always driven toward zero by the feedback mechanism. An example will illustrate the operation of the modulator in more detail.

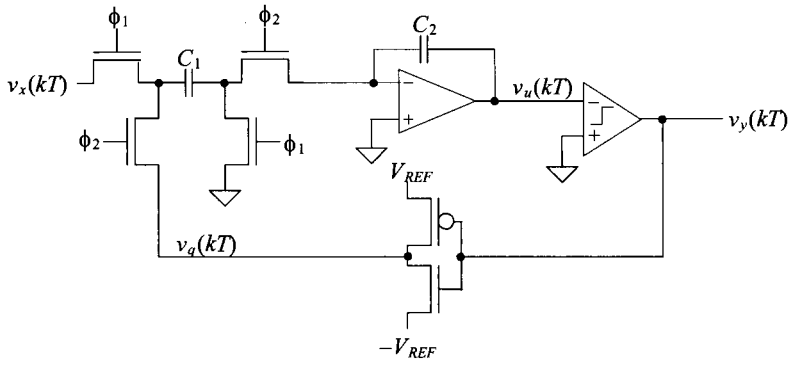

Figure 29.47 Implementation of a first-order sigma-delta modulator using a switched capacitor integrator.

Example 29.16

Using a general first-order ΣΔ modulator, assume that the input to the modulator, $v_x(kT)$ is a positive DC voltage of 0.4 V. Show the values of each variable around the ΣΔ modulator loop and prove that the overall average output of the DAC

approaches 0.4 V after 10 cycles. Assume that the DAC output is ±1 V, and that the integrator output has a unity gain with an initial output voltage of 0.1 V, and that the comparator output is either ±1 V.

The present integrator output will be equal to the sum of the previous integrator output and the previous integrator input. Therefore, Eq. (29.95) becomes

$$v_u(kT) = v_u(kT - T) + v_a(kT - T) \tag{29.102}$$

where

$$v_a(kT) = v_x(kT) - v_q(kT) \tag{29.103}$$

and the quantizing error, $Q_e(kT)$, is defined by Eqs. (29.96) and (29.98) as

$$Q_e(kT) = v_q(kT) - v_u(kT) \tag{29.104}$$

The initial conditions define the values of the variable for $k = 0$. The output of the integrator is given to be 0.1 V. Thus, the ADC output is low, the DAC output is V_{REF}, and the output of the summer, $v_a(0)$, is $0.4 - V_{REF} = -0.6$ V.

The output for $k = 1$ begins again with the integrator output. Using Eq. (29.102), $v_u(kT)$ becomes

$$v_u(T) = 0.1 + (-0.6) = -0.5 \text{ V}$$

Since the output of the integrator is negative, the output of the comparator is positive and $-V_{REF}$ is subtracted from 0.4 to arrive at the value for $v_a(T)$.

Continuing in the same manner and using the previous equations, we note the voltages for each cycle in Fig. 29.48. After 10 cycles through the modulator, the average value of $v_q(kT)$ becomes,

$$\overline{v_q(kT)} = \frac{7-3}{10} = 0.4\ V$$

k	$v_a(kT)$	$v_u(kT)$	$v_q(kT) = v_y(kT)$	$Q_e(kT)$	$\overline{v_q(kT)}$
0	–0.6	0.1	1.0	0.9	1.0
1	1.4	–0.5	–1.0	–0.5	0
2	–0.6	0.9	1.0	0.1	0.333
3	–0.6	0.3	1.0	0.7	0.50
4	1.4	–0.3	–1.0	–0.7	0.20
5	–0.6	1.1	1.0	–0.1	0.333
6	–0.6	0.5	1.0	0.5	0.429
7	1.4	–0.1	–1.0	–0.9	0.25
8	–0.6	1.3	1.0	–0.3	0.333
9	–0.6	0.7	1.0	0.3	0.40

Figure 29.48 Data from the first-order ΣΔ modulator.

Notice that the behavior of $\overline{v_q(kT)}$ swings around the desired value 0.4 V. If we were to continue computing values, as k increases, the amount that $\overline{v_q(kT)}$ differs from 0.4 V would decrease. Ideally, we could make the deviation of $\overline{v_q(kT)}$ as small as desired by allowing the modulator to take as many samples as necessary to meet that accuracy. ■

It is interesting to examine the effects of using a nonideal comparator. Suppose the integrator's output was smaller than the offset voltage of the comparator. A wrong decision would be made, causing $v_y(kT)$ to be the opposite of the desired value. However, as k increases, this error is averaged out, and the modulator still converges on the correct answer. Therefore, the comparator does not have to be very accurate in its ability to distinguish between two voltages, in contrast to Nyquist rate comparators.

The Higher Order ΣΔ Modulators

Higher order ΣΔ modulators exist which provide a greater amount of noise shaping. A second-order ΣΔ modulator can be seen in Fig. 29.49. A derivation of the second-order transfer function would reveal that the output contained a delayed version of the input plus a second-order differencing of the quantization noise, Q_e (see Problem 29.38). A third-order modulator would contain third-order differencing of the quantization and can be constructed by adding another integrator similar to integrator A into the system.

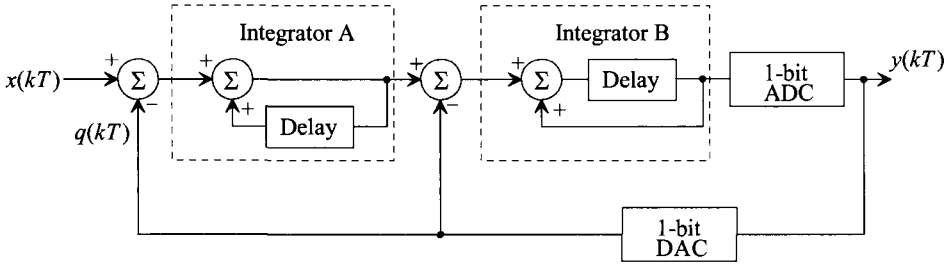

Figure 29.49 A second-order, sigma-delta modulator.

Figure 29.50 shows the noise-shaping functions of a first-, second-, and third-order modulator. The cross-hatched area under each of the curves represents the noise that remains in the signal bandwidth and is a magnified version of the blackened area of Fig. 29.46. As the order increases, notice that more of the noise is pushed out into the higher frequencies, thus decreasing the noise in the signal bandwidth. It should be reiterated that ΣΔ modulators do not attenuate noise at all. In fact, they add quantization noise that is very large at high frequencies. But because almost all of the noise is out of the signal bandwidth, it can easily be filtered, leaving only a small portion within the signal bandwidth. This point is important because the ΣΔ modulator should not be construed as a filtering circuit.

The resolution also increases as the order of the ΣΔ modulator and the oversampling ratio increases, as seen in Fig. 29.51. Using a first-order modulator, one can expect an increase in dynamic range of 9 dB with every doubling of the oversampling

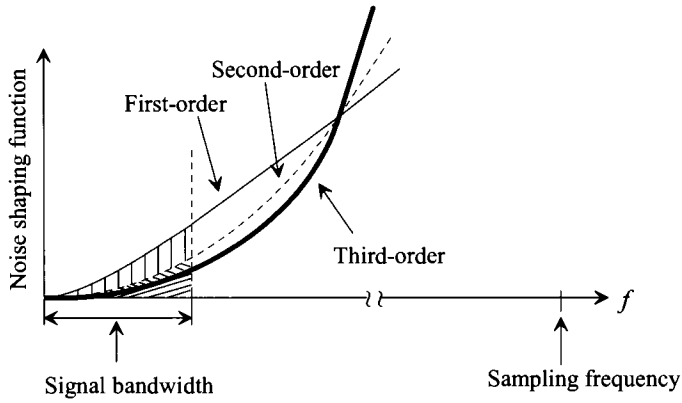

Figure 29.50 Noise shaping comparison of a first-, second- and third-order modulator.

ratio. This correlates to an approximate increase in resolution of 1.5 bits according to Eq. (28.28). The higher-order modulators have even greater gains in resolution as a 2.5-bit increase is attained with each doubling of the oversampling ratio using a second-order modulator, while the third-order modulator increases 3.5 bits.

One could essentially construct a high-order ΣΔ modulator with many integrators. However, as with any system employing feedback, stability becomes a critical issue. The same holds true for the high-order ΣΔ modulators. Several other topologies have been developed which can implement modulators in a cascaded fashion and are guaranteed to be stable. However, considerable matching requirements need to be overcome.

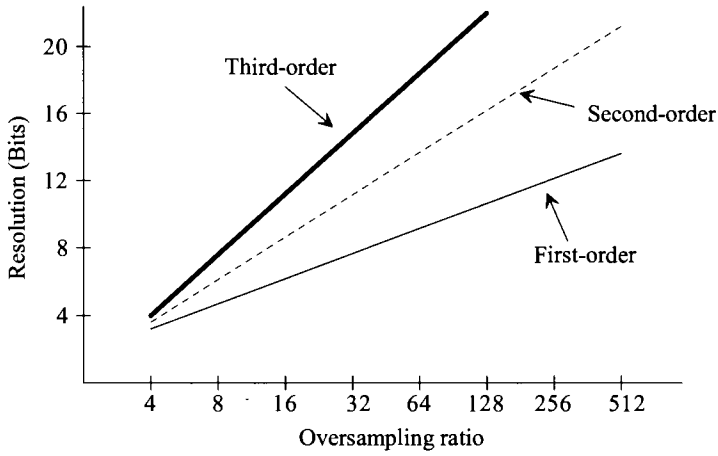

Figure 29.51 Comparison of first-, second-, and third-order modulators versus oversampling ratio and resolution.

ADDITIONAL READING

[1] R. J. Baker, *CMOS: Mixed-Signal Circuit Design, Second Edition*, Wiley-IEEE Press, 2009.

[2] R. Schreier and G. C. Temes, *Understanding Delta-Sigma Data Converters*, Wiley-IEEE Press, 2005. ISBN 978-0471465850

[3] R. J. van de Plassche, *CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters*, Second Edition, Springer, 2003.

[4] J. W. Bruce, "Nyquist-rate digital to analog converter architectures," *IEEE Potentials*, vol. 20, no. 3, pp. 24–28, August 2001. Good overview of digital to analog converters.

[5] M. Gustavsson, J. J. Wikner, and N. Tan, *CMOS Data Converters for Communications*, Springer, 2000. ISBN 978-0792377801

[6] J. W. Bruce, "Meeting the analog world challenge: Nyquist rate analog to digital converter architectures," *IEEE Potentials,* vol. 17, no. 5, pp. 36–39, January 1999. Very good introduction to ADCs.

[7] S. R. Norsworthy, R. Schreier, and G. C. Temes (eds.), *Delta-Sigma Data Converters: Theory, Design, and Simulation*, Wiley-IEEE Press, 1997. ISBN 978-0780310452

[8] M. Ismail and T. Fiez, *Analog VLSI Signal and Information Processing*, McGraw-Hill, 1994.

[9] M. J. M. Pelgrom, et. al, "25-Ms/s 8-bit CMOS A/D Converter for Embedded Application," *IEEE Journal of Solid-State Circuits*, vol. 29, no. 8, pp. 879–886, August 1994.

[10] D. Choi, et. al, "Analog Front-End Signal Processor for a 64 Mbits/s PRML Hard-Disk Drive Channel," *IEEE Journal of Solid-State Circuits*, vol. 29, no. 12, pp. 1596–1605, December 1994.

[11] M. Yotsuyanagi, T. Etoh, and K. Hirata, "A 10 Bit 50 MHz Pipelined CMOS A/D Converter with S/H," *IEEE Journal of Solid State Circuits*, vol. 28, no. 3, pp. 292–300, March 1993.

[12] P. Vorenkamp and J. P. M. Verdaasdonk, "A 10 b 50 Ms/s Pipelined ADC," *IEEE ISSCC Digest of Technical Papers*, pp. 34–35, February 1992.

[13] B. Razavi and B. A. Wooley, "A 12-b, 5-MSample/s Two-Step CMOS A/D Converter," *IEEE Journal of Solid State Circuits*, vol. 27, no. 12, pp. 1667–1678, December 1992.

[14] N. Shiwaku, "A Rail-to-Rail Video-band Full Nyquist 8-bit A/D Converter," *Proceedings of the 1991 Custom Integrated Circuits Conference.*

[15] B. P. Brandt, *Oversampled Analog-to-Digital Conversion,* Integrated Circuits Laboratory, Technical Report No. ICL91-009, Stanford University, 1991.

[16] R. L. Geiger, P. E. Allen, and N. R. Strader, *VLSI - Design Techniques for Analog and Digital Circuits*, McGraw-Hill Publishing Co., 1990.

[17] B. S. Song, S. H. Lee, and M. F. Tompsett, "A 10-bit 15 MHz CMOS Recycling Two-Step A/D Converter," *IEEE Journal of Solid State Circuits*, vol. 25, no. 12, pp. 1328–1338, December 1990.

[18] T. Shimizu, et al., "A 10-bit, 20 MHz Two-Step Parallel A/D Converter with Internal S/H," *IEEE Journal of Solid State Circuits*, vol. 24, no. 1, pp. 13–20, February 1989.

[19] J. Dornberg, P. R. Gray, and D. A. Hodges, "A 10-bit, 5-Msample/s CMOS Two-Step Flash ADC," *IEEE Journal of Solid State Circuits*, vol. 24, no. 2, pp. 241–249, April 1989.

[20] B. E. Boser, "Design and Implementation of Oversampled Analog-to-Digital Converters," Ph.D. Dissertation, Stanford University, 1988.

[21] S. Sutarja and P. R. Gray, "A Pipelined 13-bit, 250-ks/s, 5-V Analog-to-Digital Converter," *IEEE Journal of Solid State Circuits*, vol. 23, no. 6, pp. 1316–1323, December 1988.

[22] K. Uchimura et al, "Oversampling A-to-D and D-to-A Converters with Multistage Noise Shaping Modulators," *IEEE Transactions on Acoustics, Speech and Signal Processing*, pp. 1899–1905, December 1988.

[23] B. S. Song, M. F. Tompsett, and K. R. Lakshmikumar, "A 12-bit, 1-MSample/s Capacitor Error-Averaging Pipelined A/D Converter," *IEEE Journal of Solid State Circuits*, vol. 23, no. 6, pp. 1324–1333, December 1988.

[24] Y. Matsuya, K. Uchimura, et al, "A 16-bit Oversampling A/D Conversion Technology Using Triple Integration Noise Shaping," *IEEE Journal of Solid State Circuits*, vol. 22, no. 6, pp. 921–929, December 1987.

[25] S. H. Lewis and P. R. Gray, "A Pipelined 5-Msample/s 9 bit Analog-to-Digital Converter," *IEEE Journal of Solid State Circuits*, vol. 22, no. 6, pp. 954–961, December 1987.

[26] K. Bacrania, "A 12 Bit Successive-Approximation ADC with Digital Error Correction," *IEEE Journal of Solid State Circuits*, vol. 21, no. 6, pp. 1016–1025, December 1986.

[27] J. Shyu, G. C. Temes, and F. Krummenacher, "Random Errors in MOS Capacitors and Current Sources," *IEEE Journal of Solid State Circuits*, vol. 16, no. 6, pp. 948–955, December 1984.

[28] P. R. Gray, D. A. Hodges, and R. W. Brodersen (eds.), *Analog MOS Integrated Circuits*, Wiley-IEEE, 1980. ISBN 0-471-08964-8

[29] J. L. McCreary and P. R. Gray, "All-MOS Charge Redistribution Analog-to-Digital Conversion Techniques - Part I," *IEEE Journal of Solid State Circuits*, vol. 10, no. 6, pp. 371–379, December 1975.

[30] R. E. Suarez, P. R. Gray, and D. A. Hodges, "All-MOS Charge Redistribution Analog-to-Digital Conversion Techniques - Part II," *IEEE Journal of Solid State Circuits*, vol. 10, no. 6, pp. 379–385, December 1975.

PROBLEMS

29.1 A 3-bit, resistor-string DAC similar to the one shown in Fig. 29.2a was designed with a desired resistor of 500 Ω. After fabrication, mismatch caused the actual value of the resistors to be

$R_1 = 500, R_2 = 480, R_3 = 470, R_4 = 520, R_5 = 510, R_6 = 490, R_7 = 530, R_8 = 500$

Determine the maximum INL and DNL for the DAC assuming $V_{REF} = 5$ V.

29.2 An 8-bit resistor string DAC similar to the one shown in Fig. 29.2b was fabricated with a nominal resistor value of 1 kΩ. If the process was able to provide matching of resistors to within 1%, find the effective resolution of the converter. What is the maximum INL and DNL of the converter? Assume that $V_{REF} = 5$ V.

29.3 Compare the digital input codes necessary to generate all eight output values for a 3-bit resistor string DAC similar to those shown in Fig. 29.2a and b. Design a digital circuit that will allow a 3-bit binary digital input code to be used for the DAC in Fig. 29.2a. Discuss the advantages and disadvantages of both architectures.

29.4 Plot the transfer curve of a 3-bit *R*-2*R* DAC if all *R*s = 1.1 kΩ and 2*R*s = 2 kΩ. What is the maximum INL and DNL for the converter? Assume all of the switches to be ideal and $V_{REF} = 5$ V.

29.5 Suppose that a 3-bit *R*-2*R* DAC contained resistors that were perfectly matched and that $R = 1$ kΩ and $V_{REF} = 5$ V. Determine the maximum switch resistance that can be tolerated for which the converter will still have 3-bit resolution. What are the values of INL and DNL?

29.6 The circuit illustrated in Fig. 29.5 is known as a current-mode *R*-2*R* DAC, since the output voltage is defined by the current through R_F. Shown in Fig. 29.52 is an *N*-bit voltage-mode *R*-2*R* DAC. Design a 3-bit voltage mode DAC and determine the output voltage for each of the eight input codes. Label each node voltage for each input. Assume that $R = 1$ kΩ and that $R_2 = R_1 = 10$ kΩ and $V_{REF} = 5$ V.

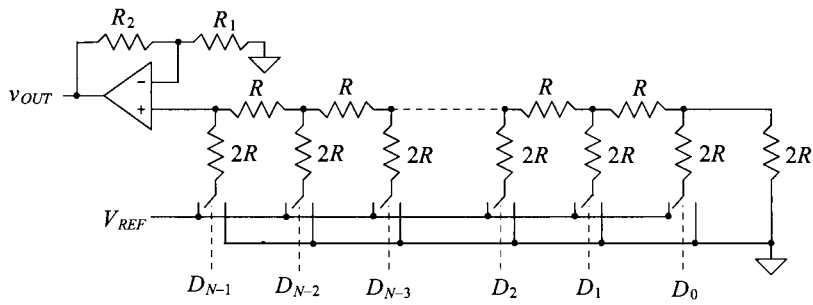

Figure 29.52 DAC used in Problem 29.6.

29.7 Design a 3-bit, current-steering DAC using the generic current-steering DAC shown in Fig. 29.9. Assume that each current source, I, is 5 μA, and find the total output current for each input code.

29.8 A certain process is able to fabricate matched current sources to within 0.05%. Determine the maximum resolution that a current-steering (nonbinary-weighted) DAC can attain using this process.

29.9 Design an 8-bit current-steering DAC using binary-weighted current sources. Assume that the smallest current source will have a value of 1 μA. What is the range of values that the current source corresponding to the MSB can have while maintaining an INL of ½ LSB? Repeat for a DNL less than or equal to ½ LSB.

29.10 Prove that the 3-bit charge-scaling DAC used in Ex. 29.6 has the same output voltage increments as the R-$2R$ DAC in Ex. 29.3 for $V_{REF} = 5$ V and $C = 0.5$ pF.

29.11 Determine the output of the 6-bit, charge-scaling DAC used in Ex. 29.7 for each of the following inputs: $D = 000010$, 000100, 001000, and 010000.

29.12 Design a 4-bit, charge-scaling DAC using a split array. Assume that $V_{REF} = 5$ V and that $C = 0.5$ pF. Draw the equivalent circuit for each of the following input words and determine the value of the output voltage: $D = 0001$, 0010, 0100, 1000. Assuming the capacitor associated with the MSB had a mismatch of 4 percent, calculate the INL and DNL.

29.13 For the cyclic converter shown in Fig. 29.17, determine the gain error for a 3-bit conversion if the feedback amplifier had a gain of 0.45 V/V. Assume that $V_{REF} = 5$ V.

29.14 Repeat Problem 29.13 assuming that the output of the summer was always 0.2 V greater than the ideal and that the amplifier in the feedback path had a perfect gain of 0.5 V/V.

29.15 Repeat Problem 29.13 assuming that the output of the summer was always 0.2 V greater than ideal and that the amplifier in the feedback path had a gain of 0.45 V/V.

29.16 Design a 3-bit pipeline DAC using $V_{REF} = 5$ V. (a) Determine the maximum and minimum gain values for the first-stage amplifier for the DAC to have less than ±½ LSBs of DNL assuming that the rest of the circuit is ideal. (b) Repeat for the second-stage amplifier. (c) Repeat for the last-stage amplifier.

29.17 Using the same DAC designed in Problem 29.16, (a) determine the overall error (offset, DNL, and INL) for the DAC if the S/H amplifier in the first stage produces an offset at its output of 0.25 V. Assume that all of the remaining components are ideal. (b) Repeat for the second-stage S/H. (c) Repeat for the last-stage S/H.

29.18 Design a 3-bit Flash ADC with its quantization error centered about zero LSBs. Determine the worst-case DNL and INL if resistor matching is known to be 5%. Assume that $V_{REF} = 5$ V.

29.19 Using the ADC designed in Problem 29.18, determine the maximum offset that can be tolerated if all of the comparators have the same magnitude of offset, but with different polarities, to attain a DNL of less than or equal to ±½ LSB.

29.20 A 4-bit Flash ADC converter has a resistor string with mismatch as shown in Table 29.1. Determine the DNL and INL of the converter. How many bits of resolution does this converter possess? V_{REF} = 5 V.

Resistor	Mismatch (%)
1	2
2	1.5
3	0
4	–1
5	–0.5
6	1
7	1.5
8	2
9	2.5
10	1
11	–0.5
12	–1.5
13	–2
14	0
15	1
16	1

Table 29.1 Mismatch in resistors used in Problem 29.20

29.21 Determine the open-loop gain required for the residue amplifier of a two-step ADC necessary to keep the converter to within ½ LSB of accuracy with resolutions of (a) 4 bits, (b) 8 bits, and (c) 10 bits.

29.22 Assume that a 4-bit, two-step Flash ADC uses two separate Flash converters for the MSB and LSB ADCs. Assuming that all other components are ideal, show that the first Flash converter needs to be more accurate than the second converter. Assume that V_{REF} = 5 V.

29.23 Repeat Ex. 29.12 for V_{IN} = 3, 5, 7.5, 14.75 V.

29.24 Repeat Ex. 29.13 for V_{IN} = 1, 4, 6, 7 V and V_{REF} = 8 V.

29.25 Assume that an 8-bit pipeline ADC was fabricated and that all the amplifiers had a gain of 2.1 V/V instead of 2 V/V. If V_{IN} = 3 V and V_{REF} = 5 V, what would be the resulting digital output if the remaining components were considered to be ideal? What are the DNL and INL for this converter?

29.26 Show that the first-stage accuracy is the most critical for a 3-bit, 1-bit per stage pipeline ADC by generating a transfer curve and determining DNL and INL for the ADC for two cases: (1) The gain of the first-stage residue amplifier set equal to 2.2 V/V and (2) the second-stage residue amplifier set equal to 2.2 V/V. For each case, assume that the remaining components are ideal. Assume that the V_{REF} = 5 V.

29.27 An 8-bit single-slope ADC with a 5 V reference is used to convert a slow-moving analog signal. What is the maximum conversion time assuming that the clock frequency is 1 MHz? What is the maximum frequency of the analog signal? What is the maximum value of the analog signal which can be converted?

29.28 An 8-bit single slope ADC with a 5 V reference uses a clock frequency of 1 MHz. Assuming that all of the other components are ideal, what is the limitation on the value of RC? What is the tolerance of the clock frequency which will ensure less than 0.5 LSB of INL?

29.29 An 8-bit dual slope ADC with a 5 V reference is used to convert the same analog signal in Problem 29.27. What is the maximum conversion time assuming that the clock frequency is 1 MHz? What is the minimum conversion time that can be attained? If the analog signal is 2.5 V, what will be the total conversion time?

29.30 Discuss the advantages and disadvantages of using a dual-slope versus a single slope ADC architecture.

29.31 Repeat Ex. 29.15 for a 4-bit successive approximation ADC using V_{REF} = 5 V for v_{IN} = 1, 3, and full-scale.

29.32 Assume that v_{IN} = 2.49 V for the ADC used in Problem 29.31 and that the comparator, because of its offset, makes the wrong decision for the MSB conversion. What will be the final digital output? Repeat for v_{IN} = 0.3025, assuming that the comparator makes the wrong decision on the LSB.

29.33 Design a 3-bit, charge-redistribution ADC similar to that shown in Fig. 29.39 and determine the voltage on the top plate of the capacitor array throughout the conversion process for v_{IN} = 2, 3, and 4 V, assuming that V_{REF} = 5 V. Assume that all components are ideal. Draw the equivalent circuit for each bit decision.

29.34 Determine the maximum INL and maximum DNL of the ADC designed in Problem 29.33 assuming that the capacitor array matching is 1%. Assume that the remaining components are ideal and that the unit capacitance, C, is 1 pF.

29.35 Show that the charge redistribution ADC used in Problems 29.32 and 29.33 is immune to comparator offset by assuming an initial offset voltage of 0.3 and determining the conversion for v_{IN} = 2 V.

29.36 Discuss the differences between Nyquist rate ADCs and oversampling ADCs.

29.37 Write a simple computer program or use a math program to perform the analysis shown in Ex. 29.16. Run the program for k = 200 clock cycles and show that the average value of $v_q(kT)$ converges to the correct answer. How many clock cycles will it take to obtain an average value if $v_q(kT)$ stays within 8-bit accuracy of the ideal value of 0.4 V? 12-bit accuracy? 16-bit accuracy?

29.38 Prove that the output of the second-order ΣΔ modulator shown in Fig. 29.49 is,

$$y(kT) = x(kT - T) + Q_e(kT) - 2Q_e(kT - T) + Q_e(kT - 2T)$$

29.39 Assume that a first order ΣΔ ADC used on a satellite in a low earth orbit experiences radiation in which an energetic particle causes a noise spike resulting in the comparator making the wrong decision on the 10th clock period. Using the program written in Problem 29.37, determine the number of clock cycles required before the average value of $v_q(kT)$ is within 12-bit accuracy of the ideal value of 0.4 V. How many extra clock cycles were required for this case versus the ideal conversion used in Prob. 37?