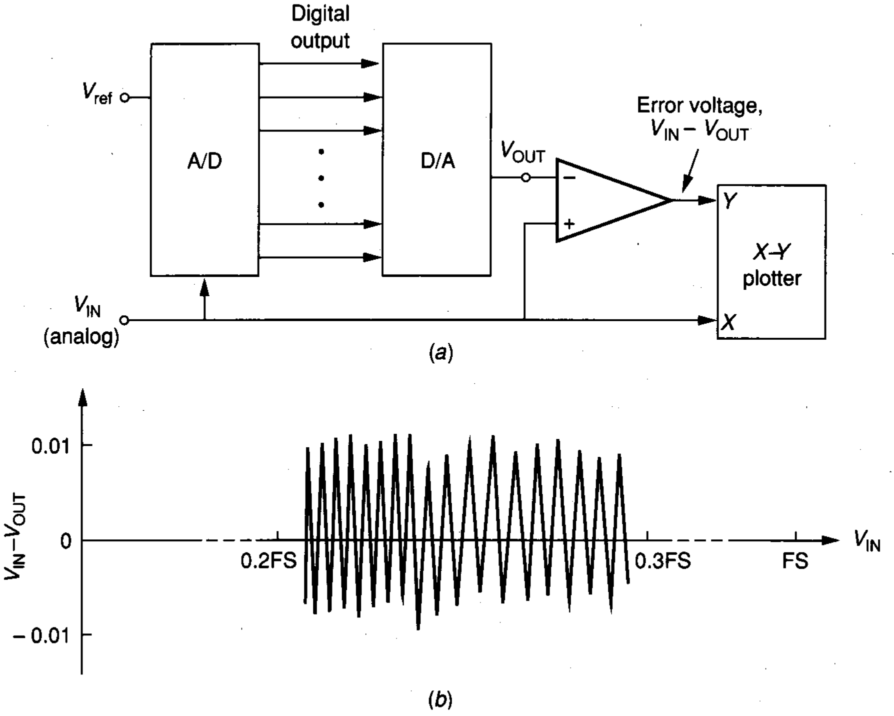

FIGURE 8.3-5
(*a*) A method of testing A/D and D/A converters, (*b*) Typical output for the error voltage of *a*.

is used to perform the tests on the converter. One simple means of testing both the D/A and A/D converters is shown in Fig. 8.3-5*a*. In this configuration, the output of an A/D converter is connected to the input of the D/A converter, resulting in an error voltage that can be plotted on an *X*-*Y* recorder as a function of the amplitude of the analog input signal. A typical portion of the error voltage for a 12-bit converter is shown in Fig. 8.3-5*b*. In this test either the A/D or D/A converter will be the device under test (DUT). The other converter must have higher performance and resolution than the DUT in order to be able to measure its performance without error. The order of the A/D and D/A converters in Fig. 8.3-5*a* can be interchanged so that the input and output are digital words that can be compared.

Several other approaches to testing the performance of an A/D converter will be briefly described. A high-quality sinusoid whose peak-to-peak amplitude is equal to the full scale value of the analog input can be used to test the A/D converter. Two such tests are called the *histogram test* and the *Fast Fourier Transform (FFT) test*. In the histogram test, a sinusoid of FS peak-to-peak amplitude is applied to the input of the A/D converter. The frequency is selected to avoid coherence with the sample rate of the A/D converter. A large number of samples of the digital output code are taken and expressed in the form of a histogram. A perfect A/D converter would produce a cusp-shaped probability density function given by

$$p(V) = \left[\pi\left(A^2 - V^2\right)^{1/2}\right]^{-1} \tag{8.3-2}$$

where A is the peak amplitude of the sine wave (FS/2) and $p(V)$ is the probability of an occurrence at a voltage V. Nonlinear behavior will show up as spikes rising above the ideal probability density function, and missing codes will be gaps in the probability density function. The frequency of the sinusoid can be increased to determine the upper frequency limits for the A/D converter performance. The histogram test can also detect offset and gain errors. Offset errors cause the histogram to be asymmetrical about the A/D converter output code corresponding to midscale. Gain errors can be determined from the width of the histogram.

The FFT test can also be used to evaluate the A/D converter performance. First, a spectrally pure sinusoid of amplitude 0.5 FS is applied to the input of the A/D converter. A 2^N point record sampled at the maximum sampling rate is taken and converted to the frequency domain using an FFT algorithm. The harmonics of the input sinusoid caused by the nonlinearity of the A/D converter are aliased into the baseband spectrum and can be used to identify the nonlinear performance of the A/D converter. The frequency of the input sine wave must be selected so that harmonics aliased into the baseband do not coincide with the fundamental. If the ratio of the fundamental to the highest amplitude harmonic on an N-bit A/D converter is greater than $6N$ dB, the error contribution of the nonlinearity is insignificant compared with the quantization noise. Unfortunately, even very low levels of harmonics may be sufficient to create encoding errors by causing a voltage near a threshold level to go to the next quantization level.

The above tests are typical of those applied to complex A/D converters. Further details and other types of testing procedures can be found in the references.[6–8] We next consider the various architectures for realizing A/D converters in MOS and/or bipolar technologies.

8.3.1 Serial A/D Converters

The serial A/D converter is similar to the serial D/A converter in that it performs serial operations until the conversion is complete. We shall examine two architectures, called the *single-slope* and the *dual-slope*. Figure 8.3-6 gives the block diagram of a single-slope serial A/D converter. This type of converter consists of a ramp generator, an interval counter, a comparator, an AND gate, and a counter that generates the output digital word. At the beginning of a conversion cycle, the analog input is sampled and held and applied to the positive terminal of the comparator. The counters are reset, and a clock is applied to both the time interval counter and the AND gate. On the first clock pulse, the ramp generator begins to integrate the reference voltage, V_{ref}. If V_{IN}^* is greater than the initial output of the ramp generator, then the output of the ramp generator, which is applied to the negative terminal of the comparator, begins to rise. Because V_{IN}^* is greater than the output of the ramp generator, the output of the comparator is high and each clock pulse applied to the AND gate causes the counter at the output to

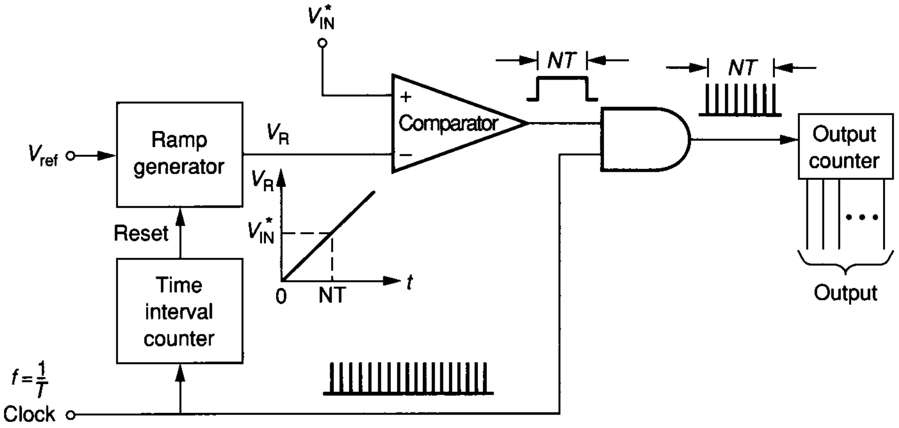

FIGURE 8.3-6
Block diagram of a single-slope serial A/D converter.

count. Finally, when the output of the ramp generator is equal to V_{IN}^*, the output of the comparator goes low, and the output counter is now inhibited. The binary number representing the state of the output counter can now be converted to the desired digital word format.

The single-slope A/D converter can have many different implementations. For example, the interval counter can be replaced by logic to detect the state of the comparator output and reset the ramp generator when its output has exceeded V_{IN}^*. The serial A/D converter has the advantage of simplicity of operation. Disadvantages of the single-slope A/D converter are that it is subject to error in the ramp generator and it is unipolar. Another disadvantage of the single-slope A/D converter is that a long conversion time ($2^N T$ in the worst case) is required if the input voltage is near the value of V_{ref}.

A block diagram of a dual-slope A/D converter is shown in Fig. 8.3-7. The basic advantage of this architecture is the elimination of the dependence of the

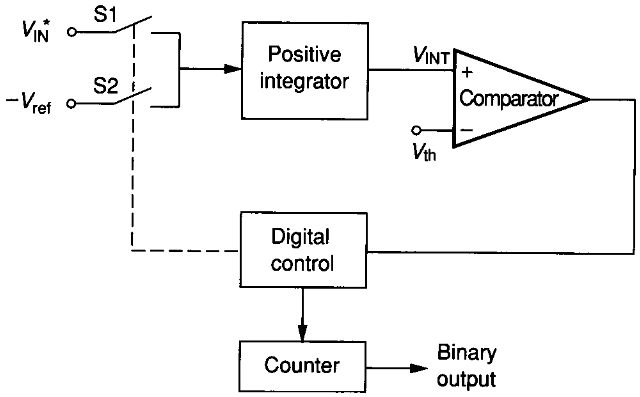

FIGURE 8.3-7
Block diagram of a dual-slope A/D converter.

conversion process on the linearity and accuracy of the ramp generator. Initially, V_{INT} is zero, and the input is sampled and held. In this scheme, it is necessary for V_{IN}^* to be positive. The conversion process begins by resetting the positive integrator until the output of the integrator is equal to the threshold, V_{th}, of the comparator. Next, S1 is closed, and V_{IN}^* is integrated for N_{ref} number of clock cycles. Figure 8.3-8 illustrates the conversion process. It is seen that the slope of the voltage at V_{INT} is proportional to the amplitude of V_{IN}^*. The voltage, $V_{\text{int}}(t)$, during this time is given as

$$V_{\text{int}}(t) = K\int_0^{N_{\text{ref}}T} V_{\text{IN}}^* dt + V_{\text{int}}(0) = KN_{\text{ref}}TV_{\text{IN}}^* + V_{\text{th}} \qquad (8.3\text{-}3)$$

where T is the clock period. At the end of N_{ref} counts, the carry output of the counter is applied to switch S2 and causes $-V_{\text{ref}}$ to be applied to the integrator. Now the integrator integrates negatively with a constant slope because V_{ref} is constant. When $V_{\text{int}}(t)$ becomes less than the value of V_{th}, the counter is stopped, and its binary count can be converted into the digital word. This is demonstrated by considering $V_{\text{int}}(t)$ during the time designated as t_2 in Fig. 8.3-8. This voltage is given as

$$V_{\text{int}}(t) = V_{\text{int}}(0) + K\int_0^{N_{\text{out}}T} (-V_{\text{ref}})dt \qquad (8.3\text{-}4)$$

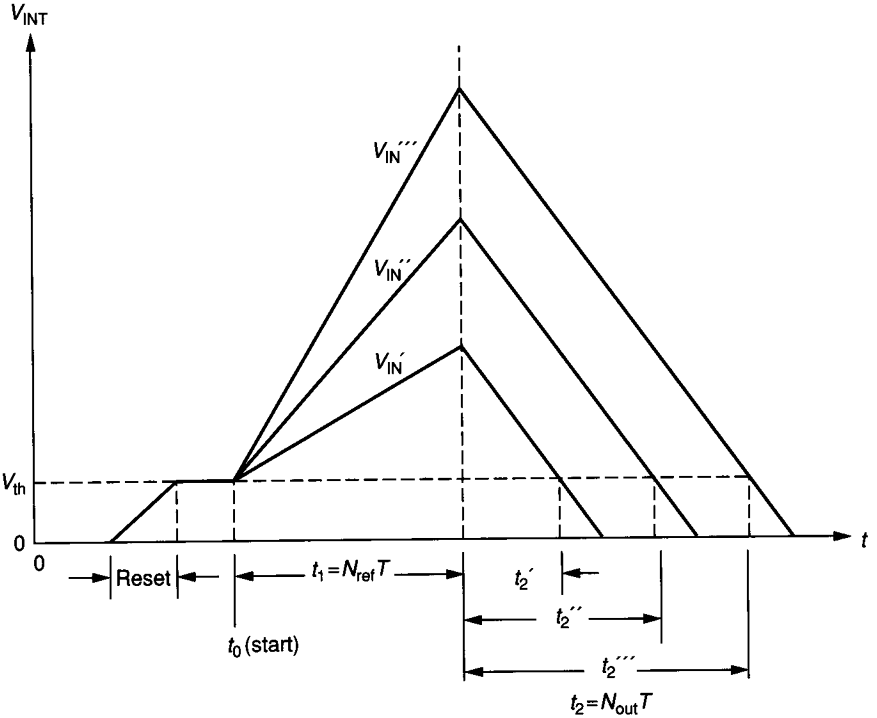

FIGURE 8.3-8
Waveforms illustrative of the dual-slope A/D converter of Fig. 8.3-7.

However, when $t = N_{\text{out}}T$, then Eq. 8.3-4 becomes

$$V_{\text{int}}(N_{\text{out}}T) = (KN_{\text{ref}}TV^*_{\text{IN}} + V_{\text{th}}) - KV_{\text{ref}}N_{\text{out}}T \tag{8.3-5}$$

Because $V_{\text{int}}(N_{\text{out}}T) = V_{\text{th}}$, then Eq. 8.3-5 can be solved for N_{out}, giving

$$N_{\text{out}} = N_{\text{ref}}\left(\frac{V^*_{\text{IN}}}{V_{\text{ref}}}\right) \tag{8.3-6}$$

It is seen that N_{out} will be some fraction of N_{ref}, where that fraction corresponds to the ratio of V^*_{IN} to V_{ref}.

The output of the serial dual-slope D/A converter (N_{out}) is not a function of the threshold of the comparator, the slope of the integrator, or the clock rate. Therefore, it is a very accurate method of conversion. The only disadvantage is that it takes a worst-case time of $2(2^N)T$ for a conversion, where N is the number of bits of the A/D converter. The positive integrator of this scheme can be replaced by a switched capacitor integrator, which will be discussed in Sec. 8.5.

The preceding two examples are representative of the architecture and resulting performance of serial A/D converters. Other forms of serial conversion exist in the literature.[9,10] The serial A/D converter can be expected to be slow but to provide a high resolution. Typical values for serial A/D converters are conversion frequencies of less than 100 Hz and greater than 12 bits of resolution.

8.3.2 Successive Approximation A/D Converters

A second category of A/D converters is called *successive approximation A/D converters*. This class of A/D converters converts an analog input into an N-bit digital word in N clock cycles. Consequently, the conversion time is less than for the serial converters without much increase in the complexity of the circuit. We will examine successive approximation converters that use a combination of voltage-scaling and charge-scaling D/A converters, serial D/A converters, and algorithmic D/A converters.

Figure 8.3-9 illustrates the architecture of a successive approximation A/D converter. This converter consists of a comparator, a D/A converter, and digital control logic. The function of the digital control logic is to determine the value of each bit in a sequential manner based upon the output of the comparator. To illustrate the conversion process, assume that the converter is unipolar (only positive analog signals can be applied). The conversion cycle begins by sampling the analog input signal to be converted. Next, the digital control circuit assumes that the MSB is 1 and all other bits are 0. This digital word is applied to the D/A converter, which generates an analog signal of $0.5V_{\text{ref}}$, which is compared to the sampled analog input, V^*_{IN}. If the comparator output is high, then the digital control logic makes the MSB 1. If the comparator output is low, the digital control logic makes the MSB 0. This completes the first step in the approximation sequence. At this point, the value of the MSB is known. The approximation process continues by once more applying a digital word to the D/A converter, with the MSB having its proven value, the second bit guessed at 1, and all

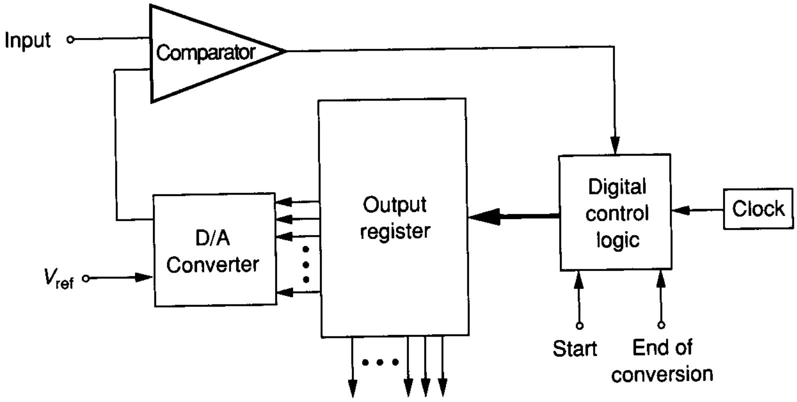

FIGURE 8.3-9
Example of a successive approximation A/D converter architecture.

remaining bits having a value of 0. Again, the sampled input is compared to the output of the D/A converter with this digital word applied. If the output of the comparator is high, the second bit is proven to be 1. If the output of the comparator is low, the second bit is 0. The process continues in this manner until all bits of the digital word have been decided by the successive approximation process.

Figure 8.3-10 shows how the successive approximation sequence works in converting the analog output of the D/A converter to the sampled analog input.

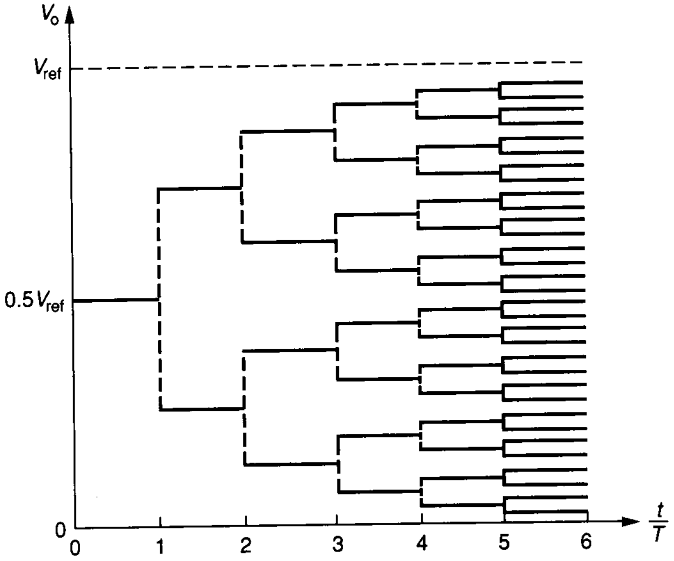

FIGURE 8.3-10
The successive approximation process.

It is seen that the number of cycles required for the conversion to an N-bit word is N. It is also observed that as N becomes large, the requirement of the comparator to distinguish between almost identical signals must increase. Bipolar A/D conversion can be achieved by using a sign bit to choose either $-V_{\text{ref}}$ or V_{ref}.

The digital control logic of Fig. 8.3-9 is often called a *successive approximation register* (SAR). An example of a 5-bit successive approximation A/D converter using a SAR is shown in Fig. 8.3-11. This SAR has the advantage of compatibility with a bit-slice approach, which makes it attractive for integrated circuit implementation. The bit-slice consists of a shift register (SR), an AND gate (G), a SR flip-flops (FF) with direct reset capability, and an analog switch (AS).

Figure 8.3-12 shows an example of a successive approximation A/D converter that uses the voltage-scaling and charge-scaling D/A converter of Fig. 8.2-18. The extra components, in addition to the D/A converter, include a comparator and a SAR. From the concepts of Sec. 6.6, we know that the comparator should have a gain greater than $(V_L 2^{M+K}/V_{\text{ref}})$, where V_L is the minimum output swing of the comparator required by the logic circuit it drives. For example, if $M + K = 12$ and $V_L = V_{\text{ref}}$, then the comparator must have a voltage gain of at least 4096.

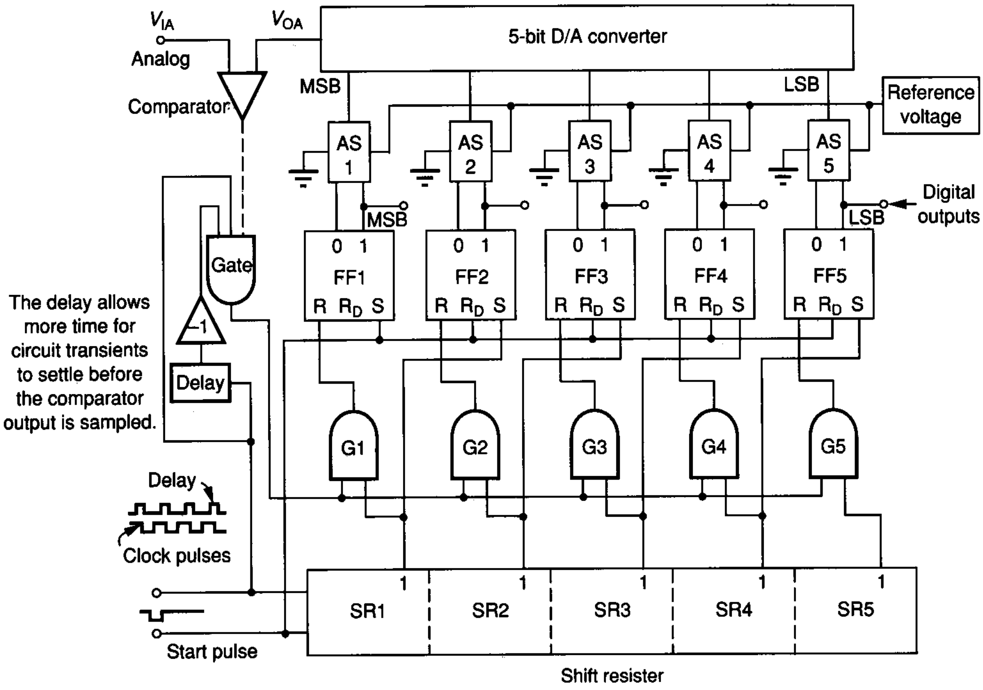

FIGURE 8.3-11
Five-bit successive approximation A/D converter with shift register control.

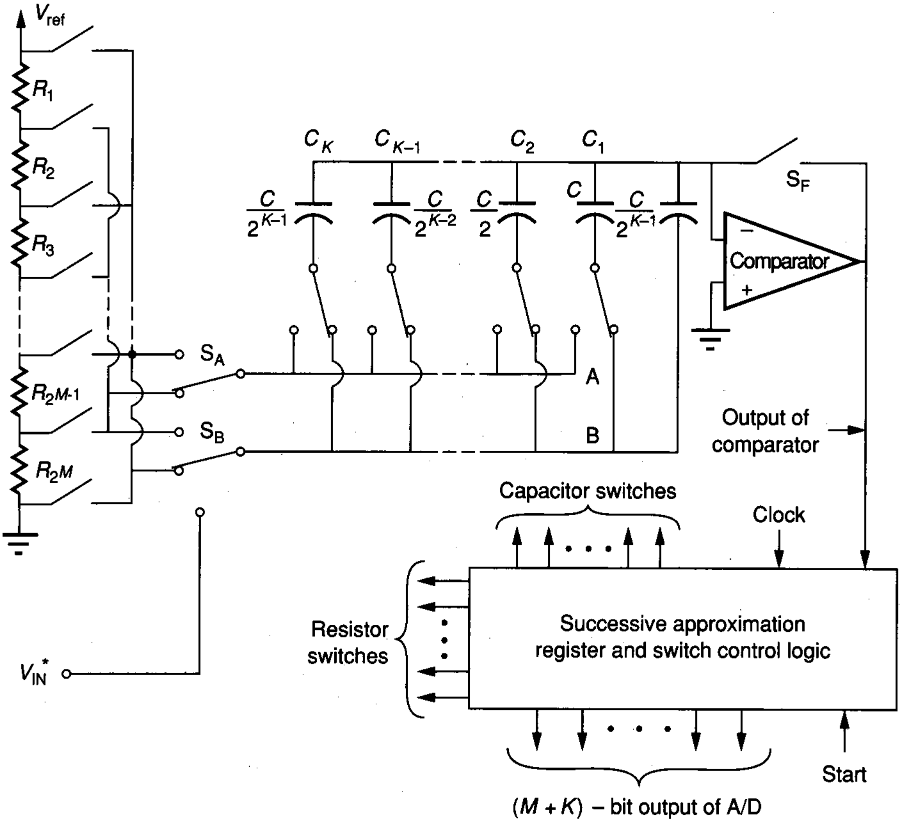

FIGURE 8.3-12
A voltage-scaling, charge-scaling, successive approximation A/D converter.

The conversion operates as follows. With S_F closed, the bottom plates of C_1 through C_K are connected through switch S_B to V^*_{IN}. The voltage stored on the capacitor array at the end of the sampling period is actually V^*_{IN} minus the threshold voltage of the comparator, which removes the threshold as a source of offset error. Because the comparator has unity feedback, it must be compensated in order to remain stable in this step. After switch S_F is opened, a successive approximation search among the resistor string taps is performed to find the segment in which the stored sample lies. Next, buses A and B are switched to the ends of the resistor defining this segment. Finally, the capacitor bottom plates are switched in a successive approximation sequence until the comparator input voltage converges back to the threshold voltage. The sequence of comparator outputs is a digital code corresponding to the unknown analog input signal. The A/D converter in Fig. 8.3-12 is capable of 12-bit monotonic conversion with a differential linearity of less than $\pm\frac{1}{2}$ LSB and a conversion time of 50 ms.[11]

A successive approximation A/D converter using the serial D/A converter of Fig. 8.2-21 is shown in Fig. 8.3-13. This converter works by converting the

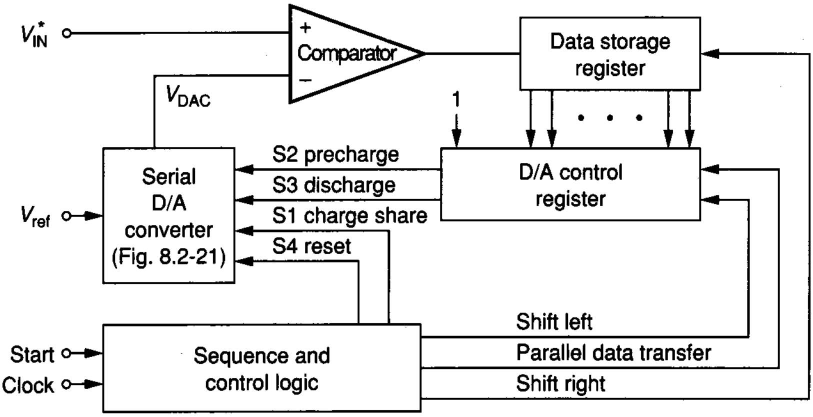

FIGURE 8.3-13
A serial A/D converter using the serial D/A converter of Fig. 8.2-21.

MSB, a_1, first. (The ith bit is denoted as d_i for D/A conversion and ith LSB is denoted as a_i for A/D conversion.) The control logic takes a very simple form because the D/A input string at any given point in the conversion is just the previously encoded word taken LSB first. For example, consider the point during the A/D conversion where the first K MSBs have been decided. To decide the $(K + 1)$th MSB, a $(K + 1)$-bit word is formed in the D/A control register by adding a 1 as the LSB to the K-bit word already encoded in the data storage register. A $(K + 1)$-bit D/A conversion then establishes the value of a_{N-K} by comparison with the unknown voltage V_{IN}^*. The bit is then stored in the data storage register, and the next serial D/A conversion is initiated. The conversion sequence is shown in detail in Table 8.3-1. Altogether, $N(N + 1)$ clock cycles are required for an N-bit A/D converter using the configuration of Fig. 8.3-13.

TABLE 8.3-1
Conversion sequence for the serial D/A converter of Fig. 8.3-13.

D/A conversion number	D/A input word						Comparator output	Number of charging steps
	d_N	d_{N-1}	d_{N-2}	$\cdots$	d_2	d_1		
1	1	—	—		—	—	a_1	2
2	1	a_1	—		—	—	a_2	4
3	1	a_2	a_1		—	—	a_3	6
·	·	·	·		·	·	·	·
·	·	·	·		·	·	·	·
·	·	·	·		·	·	·	·
N -1	1	a_{N-2}	a_{N-3}		a_3	a_2	a_{N-1}	$2(N - 1)$
N	1	a_{N-1}	a_{N-2}	$\cdots$	a_2	a_1	a_N	$2N$

Total number of charging steps $= N(N + 1)$

An algorithmic A/D converter patterned after the algorithmic D/A converter of the preceding section is shown in Fig. 8.3-14. This N-bit A/D converter consists of N stages and N comparators for determining the signs of the N outputs. Each stage takes its input, multiplies it by 2, and adds or subtracts the reference voltage depending on the sign of the previous output. The comparator outputs form an N-bit digital representation of the bipolar analog input to the first stage. The operation of the algorithmic A/D converter can be demonstrated by the following example.

Example 8.3-1. Illustration of the operation of the algorithmic A/D converter. Assume that the sampled analog input to a 4-bit algorithmic A/D converter is 1.5 V. If V_{ref} is equal to 5 V, then the conversion proceeds as follows. Since $V_{\text{IN}}^* = 1.5$ V is positive, the output of the comparator of stage 1 is high, which corresponds to a digital 1. Stage 1 then multiplies this value by 2 to get 3 V and subtracts V_{ref}, obtaining an output of -2 V. Stage 2 input sees a negative value, which causes the comparator of this stage to be low, equivalent to a digital 0. Stage 2 then multiplies -2 V by 2 and adds the 5 V reference to output a value of 1 V. Because the output of stage 2 is positive, the comparator of stage 3 is high. The 1 V signal is then multiplied by 2 and 5 V is subtracted, giving a stage 3 output of -3 V. The conversion ends when the comparator of the fourth stage goes low because of the negative input voltage from stage 3.

The digital output word is 1010 for this example. To determine whether this is correct, we use the following formula.

$$V_{\text{ANALOG}} = V_{\text{ref}}(b_1 2^{-1} + b_2 2^{-2} + b_3 2^{-3} + \cdots + b_N 2^{-N})$$

where b_i is +1 if the ith bit is 1 and -1 if the ith bit is 0. In this example, we see that

$$V_{\text{ANALOG}} = 5\left(\frac{1}{2} - \frac{1}{4} + \frac{1}{8} - \frac{1}{16}\right) = 5(0.3125) = 1.5625$$

It is seen that the value of V_{ANALOG} will eventually converge to the value 1.5.

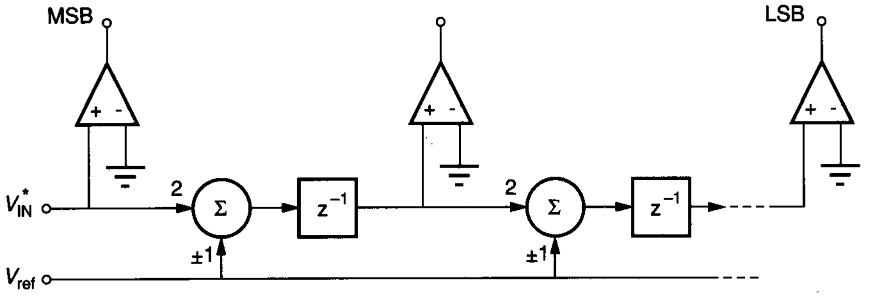

FIGURE 8.3-14
Pipeline implementation of the algorithmic A/D converter.

The algorithmic A/D converter of Fig. 8.3-14 has the disadvantage that the time to convert a sample is N clock cycles, although one complete conversion can be obtained at each clock cycle. The algorithmic A/D converter is considered to be ratio-independent because the performance does not depend on the ratio accuracy of a capacitor or resistor array. The multiplication by 2 of the first stage must be accurate to within 1 LSB, which is a distinct disadvantage of the pipeline configuration of the algorithmic A/D converter.

The iterative reduction of Fig. 8.2-23 resulting in Fig. 8.2-24 can be applied to the A/D converter of Fig. 8.3-14 in a manner similar to what was done for the algorithmic pipeline D/A converter. The analog output of the ith stage can be expressed as

$$V_{Oi} = [2V_{O,i-1} - b_i V_{\text{ref}}]z^{-1} \tag{8.3-7}$$

where b_i is +1 if the ith bit is 1 and −1 if the ith bit is 0. This equation can be implemented with the circuit in Fig. 8.3-15*a*. The next step is to incorporate the ability to sample the analog input voltage at the start of the conversion. This step is shown in Fig. 8.3-15*b*. In this implementation, $-V_{\text{ref}}$ has been

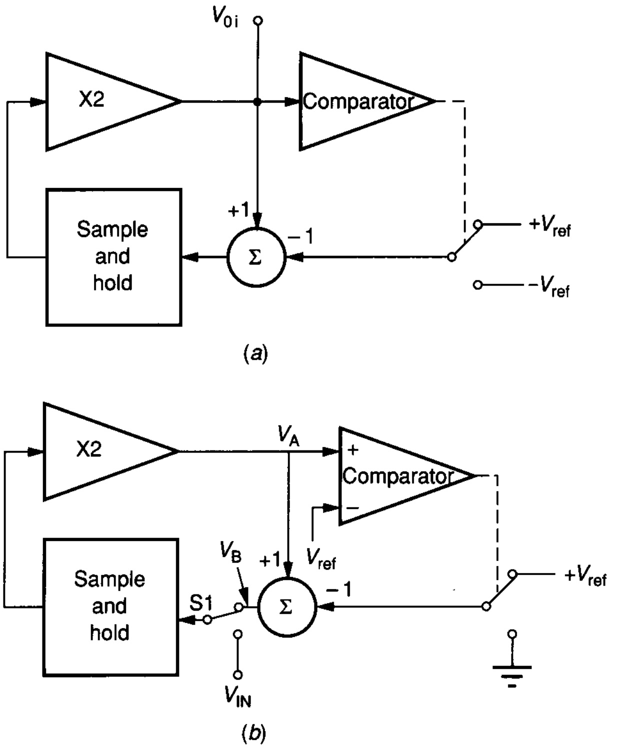

FIGURE 8.3-15
(*a*) Implementation of Eq. 8.3-7, (*b*) Implementation of the iterative algorithmic A/D converter.

replaced with ground for simplicity. The iterative version of the algorithmic A/D converter consists of a sample-and-hold circuit, a gain-of-2 amplifier, a comparator, and a reference subtraction circuit.

The operation of the converter involves first sampling the input signal by connecting switch S1 to V_{IN}. V_{IN}^* is then applied to the gain-of-2 amplifier. To extract the digital information from the input signal, the resultant signal, denoted as V_A, is compared to the reference voltage. If V_A is larger than V_{ref}, the corresponding bit is set to 1 and the reference voltage is then subtracted from V_A. If V_A is less than V_{ref}, the corresponding bit is set to 0 and V_A is unchanged. The resultant signal, denoted by V_B, is then transferred by means of switch S1 back into the analog loop for another iteration. This process occurs until the desired number of bits have been obtained, whereupon a new sampled value of the input signal will be processed. The digital word is processed in a serial manner with the MSB first. An example illustrates the process.

Example 8.3-2. Conversion process of an iterative algorithmic A/D converter. The iterative algorithmic A/D converter of Fig. 8.3-15*b* is to be used to convert an analog signal of $0.8V_{ref}$. Figure 8.3-16 shows the waveforms for V_A and V_B during the process. T is the time for one iteration cycle. In the first iteration, the analog input of $0.8V_{ref}$ is applied by switch S1 and results in a value for V_A of $1.6V_{ref}$, which corresponds to a V_B value of $0.6V_{ref}$ and a MSB of 1. During the next iteration, V_B is multiplied by 2 to give a V_A of $1.2V_{ref}$. Thus, the next bit is also 1, and V_B is $0.2V_{ref}$. V_A during the third iteration is $0.4V_{ref}$, resulting in the assignment of 0 for the next bit and a value of $0.4V_{ref}$ for V_B. The fourth iteration gives V_A as $0.8V_{ref}$, which gives $V_B = 0.8V_{ref}$ and the fourth bit as 0. The fifth iteration gives $V_A = 1.6V_{ref}$, $V_B = 0.6V_{ref}$, and the fifth bit as 1. This procedure continues as long as desired. The digital word after the fifth iteration is 11001 and is equivalent to an analog voltage of $0.78125V_{ref}$.

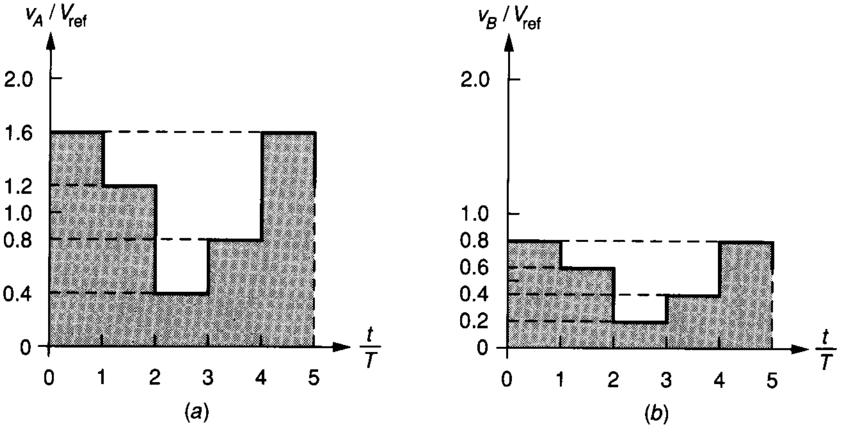

FIGURE 8.3-16
Waveforms for Example 8.3-2 and Fig. 8.3-15*b*: (*a*) V_A, (*b*) V_B.

The iterative algorithmic A/D converter can be constructed from very little precision hardware. Its implementation in a monolithic technology can therefore be area-efficient. A distinct advantage over the pipeline configuration is that all of the gain-of-2 amplifiers are identical because only one is used in an iterative manner. Sources of error for this A/D converter include low operational amplifier gain, finite input offset voltage in the operational amplifier, charge injection from the MOS switches, and capacitance voltage dependence. An integrated 12-bit A/D converter using the approach of Fig. 8.3-15*b* has exhibited an experimental performance having a differential linearity and integral linearity of 0.019% (0.8LSB) and 0.034% (1.5LSB), respectively, for a sample rate of 4 kHz. These values increased to 0.022% (0.9LSB) and 0.081% (3.2LSB) for a sample rate of 8 kHz.

The successive approximation A/D architecture is a very general one, as has been shown. It can make use of any of the D/A converters we have illustrated. If serial D/A converters are used, the conversion time is increased and the area required is decreased. In general, successive approximation A/D converters can have conversion times that fall within the range of 10^4 to 10^5 conversions/second. They are capable of 8 to 12 bits of untrimmed accuracy. The number of bits can be increased if trimming is permitted.

8.3.3 Parallel A/D Converters

In many applications, it is necessary to have a smaller conversion time than is possible with the previously defined A/D converter architectures. This has led to the development of high-speed A/D converters that use parallel techniques to achieve short conversion times. The ultimate conversion speed is one clock cycle, which typically consists of a set-up and a convert phase. Some of the high-speed architectures trade off speed with area and require more than one clock cycle but less than the N clock cycles required for the successive approximation A/D architecture. Another method of improving the speed of the converter is to increase the speed of the individual components. Typically, the comparator sample time, T_{sample} (see Eq. 8.3-1), is the limiting factor for the speed. In this presentation, we shall consider the parallel, the time-interleaved, the two-step, and the ripple approaches to implementing a high-speed A/D converter.

Figure 8.3-1 is a general block diagram of a high-speed A/D converter known as the *parallel* or *flash* A/D converter. An example of how this converter works is illustrated in Fig. 8.3-17. Figure 8.3-17 shows a 3-bit parallel A/D converter. V_{ref} is divided into eight values, as indicated in the figure. Each of these values is applied to the positive terminal of a comparator. The outputs of the comparators are taken to a digital encoding network, which determines the digital output word from the comparator outputs. For example, if V_{IN}^* is $0.6V_{\text{ref}}$, then the top three comparator outputs are 1s and the bottom four are 0s. The digital encoding network would identify 100 as the corresponding digital word. Many versions of this basic concept exist. For example, the voltage at the taps may be in multiples of $V_{\text{ref}}/16$ with $V_{\text{ref}}/8$ voltage differences between the taps. Also, the resistor string can be connected between $+V_{\text{ref}}$ and $-V_{\text{ref}}$ to achieve bipolar conversion (positive and negative analog output voltages).

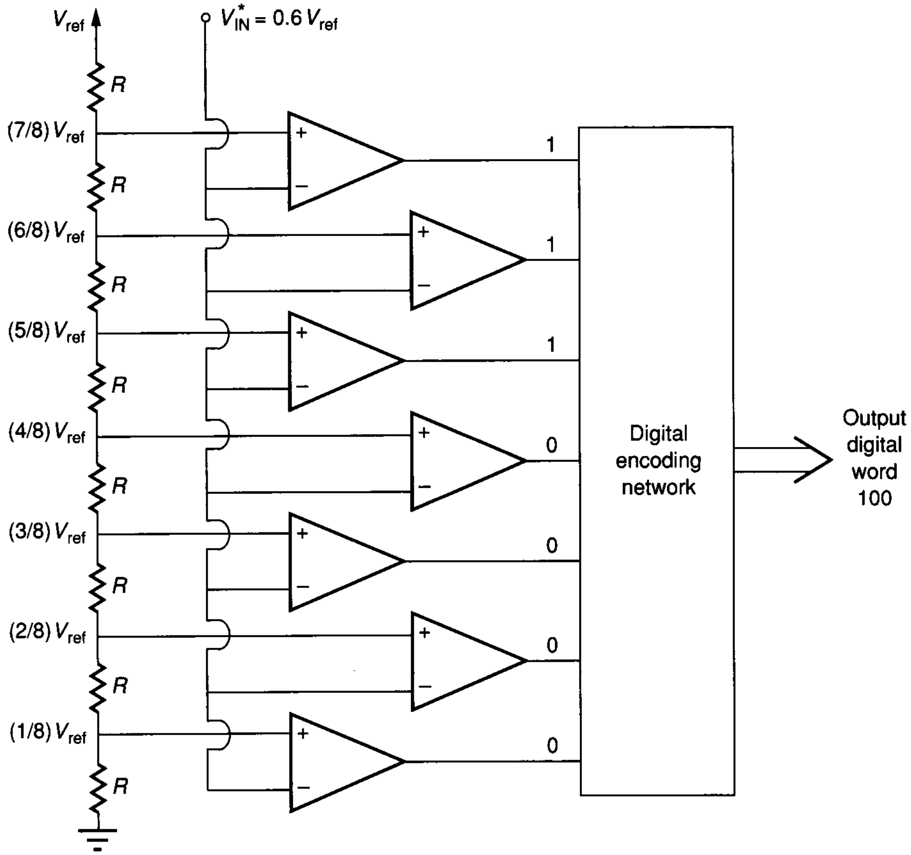

FIGURE 8.3-17
A 3-bit parallel A/D converter.

The parallel A/D converter of Fig. 8.3-17 converts the analog signal to a digital word in one clock cycle, which has two phase periods. During the first phase period, the analog input voltage is sampled and applied to the comparator inputs. During the second phase period, the digital encoding network determines the correct output digital word and stores it in a register/buffer. Thus, the conversion is limited by how fast this sequence of events can occur. Typical clock frequencies can be as high as 20 MHz for CMOS and 100 MHz for BJT technologies. This gives a theoretical conversion time of 50 ns and 10 ns, respectively. The sample-and-hold time may be larger than these values and could prevent these conversion times from being realized. Another problem is that as N increases, the number of comparators required is $2^N - 1$. For N greater than 8, too much area is required. Other methods we shall discuss give almost the same conversion times with much more efficient utilization of chip area.

One method of achieving small system conversion times is to use slower A/D converters in parallel, which is called *time-interleaving* and is shown in Fig.

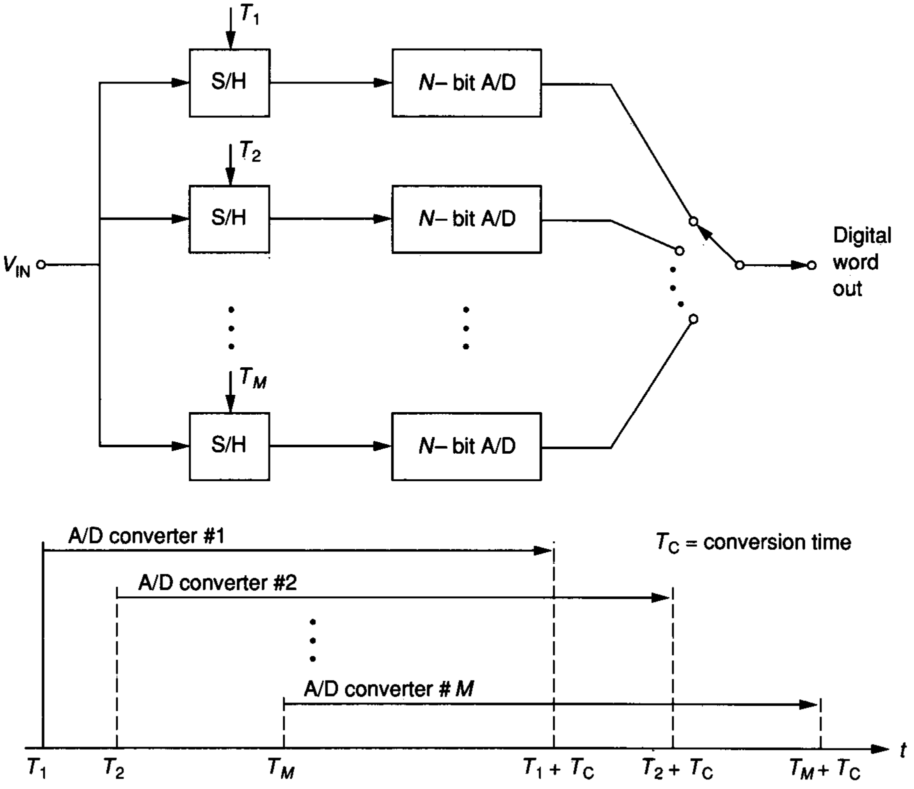

FIGURE 8.3-18
A time-interleaved A/D converter array.

8.3-18. Here M successive approximation A/D converters are used in parallel to complete the N-bit conversion of one analog signal per clock cycle. The sample-and-hold circuits consecutively sample and apply the input analog signal to their respective A/D converters. N clock cycles later, the A/D converter provides a digital word output. If $M = N$, then a digital word is output at every cycle. If one examines the chip area for an N bit A/D converter using the parallel A/D converter architecture ($M = 1$) compared with the time-interleaved architecture for $M =$ N, the minimum area will occur for a value of M between 1 and N.

Combining the parallel approach with a series approach results in an A/D converter architecture with high speed and reasonable area. This approach is often called a pipeline A/D converter, particularly if the number of series stages is greater than two. Figure 8.3-19 shows a $2M$-bit A/D converter using two M-bit parallel A/D converters. The method first converts the M MSBs and then converts the M LSBs. Consequently, only $2^{M+1} - 2$ comparators are required to convert a $2M$-bit digital word. For the configuration of Fig. 8.3-19, the analog input is applied to the left-hand string of $2^M - 1$ comparators during the first clock phase, and the M MSBs are encoded during the second clock phase. During the

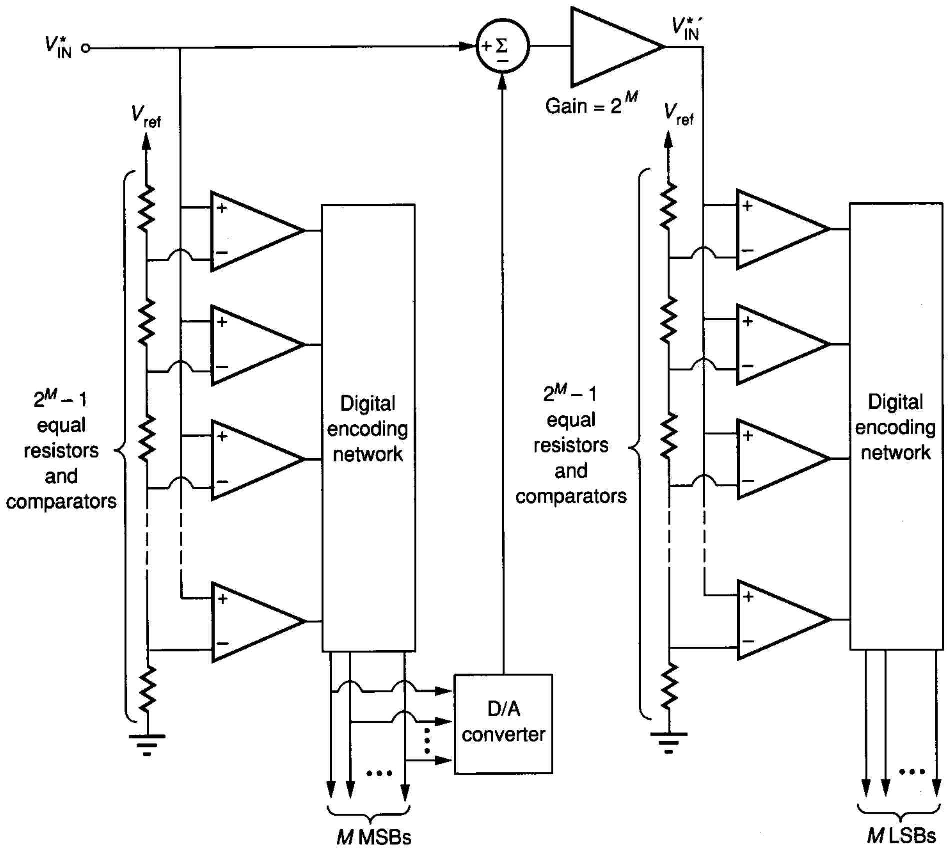

FIGURE 8.3-19
A 2M-bit parallel-series A/D converter configuration.

third clock phase, the M MSBs are converted to an analog equivalent, which is subtracted from V_{ref}, multiplied by a gain of 2^M, and applied to the right-hand string of $2^M - 1$ comparators. Finally, during the fourth clock phase, the M LSBs are encoded. Thus, if the clock has two phases, then in two clock cycles a 2M-bit digital word will be converted.

It is possible to make this conversion in three phases if necessary because the second and third phases can be combined into a single phase. Figure 8.3-20 shows the microphotograph of a parallel-series 8-bit A/D converter using the architecture of Fig. 8.3-19 implemented with MOS technology. The conversion time for this converter is in the range of 2 μs and is limited by the sampling time of the comparators and the settling times for the logic encoding network.

Up to this point analog-to-digital conversion techniques compatible with BJT and CMOS technology have been presented. The major categories include serial, successive approximation, and parallel A/D converters and are compared in Table 8.3-2 along with high performance A/D converters discussed next.

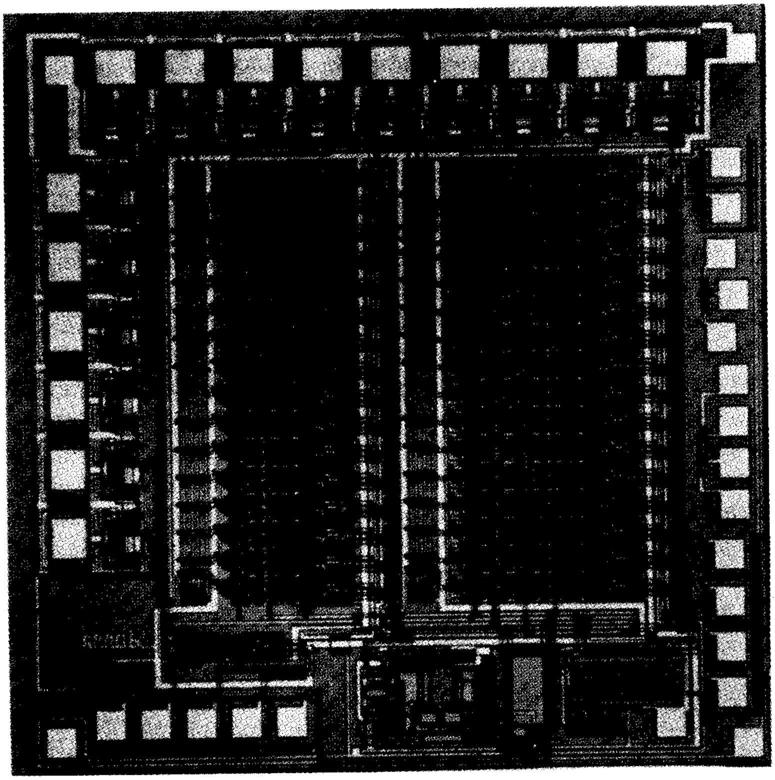

FIGURE 8.3-20
Microphotograph of the implementation of Fig. 8.3-19 for $M = 4$.

TABLE 8.3-2
Comparison of the performance of the various types of A/D converters

A/D converter type	Performance characteristics
Serial	1–100 conversions/second; 12–14-bit accuracy; requires no element-matching; a stable reference voltage is necessary
Successive approximation	10,000–100,000 conversions/second; 8–10 bits of untrimmed accuracy; 12–14 bits monotonicity; 12–14 bits trimmed accuracy
Parallel	$10^6 - 2 \times 10^7$ conversions/second; 7–8 bits of accuracy; requires large area
High performance	8000–10^6 conversion/second; 12–18 bits of accuracy

8.3.4 High-Performance A/D Converters

Recent advances in A/D converters have resulted in converters with significant improvement in performance over those considered in the previous categories. The performance improvement is typically in the areas of resolution or speed. Converters whose resolution is approaching 18 bits or sample rates of several hundred megasamples per second are now feasible. This discussion will examine three A/D converters that offer improved performance. These A/D converters include selfcalibrating A/D converters, pipeline A/D converters, and oversampled A/D converters.

A self-calibrating A/D converter is one that includes a calibration cycle to adjust its transfer characteristic to approach the ideal transfer characteristic. A block diagram of a self-calibrating A/D converter using the basic architecture of Fig. 8.3-12 is shown in Fig. 8.3-21. This circuit consists of an N-bit charge-scaling array called the main D/A converter, an M-bit voltage scaling array called the sub–D/A converter, and a voltage-scaling array called a calibration D/A converter. The calibration D/A converter must have several more bits of resolution than the sub–D/A converter. Digital control circuits govern capacitor switching during the

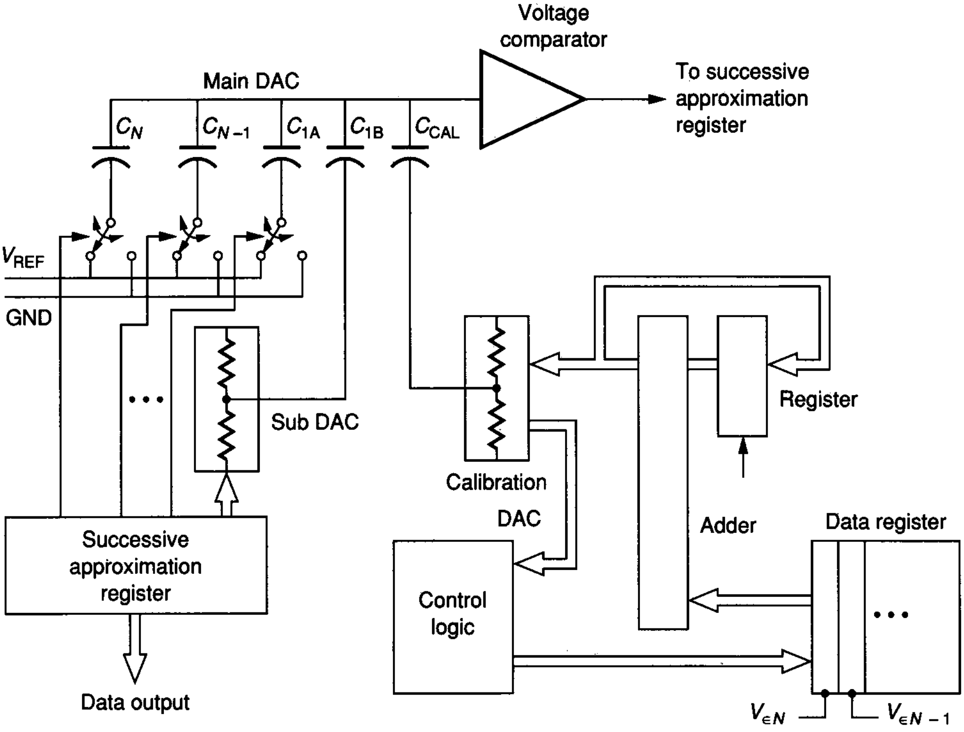

FIGURE 8.3-21
Block diagram of a self-calibrating A/D converter.

calibration cycle and store the nonlinearity correction terms in data registers. The ratio errors of the sub–D/A converter and overall quantization errors accumulate during digital computation of error voltages. To overcome these errors, at least one bit of additional resolution is needed during the calibration cycle. This extra bit is used to achieve final linearity within 1 LSB of an ideal straight line or within 0.5 LSB of an ideal staircase converter response. In practice, two extra bits are used in order to have a margin of safety. Typical performance of a self-calibrating A/D converter is 15–16 bits of resolution, an offset error of less than 1 LSB, a dynamic range of over 90 dB, a nonlinearity of less than 0.25 LSB, and a conversion time of less than 50 μs.

A pipeline converter is different from the A/D converter of Fig. 8.3-19 in that more stages are used with fewer bits per stage. This allows the designer to make tradeoffs between area and conversion time. A typical pipeline converter might have four stages with 3–4 bits per stage. A 2-bit implementation of a pipeline stage is shown in Fig. 8.3-22*a*. The input is sampled and held and applied to the comparators C_0 through C_2, where the 2-bit digital output is encoded. The digital output controls the switches on the resistor string to obtain an analog output equivalent to the 2-bit digital word. This analog voltage is subtracted from the original input and multiplied by 4, which results in the residue voltage passed on to the next stage. Figure 8.3-22*b* shows the plot of the amplified residue voltage versus the analog input voltage of the A/D converter of Fig. 8.3-22*a*.

The errors that can occur in the pipeline converter are shown in Table 8.3-3 along with their effects and possible solutions. Auto-zeroing, trimming, and calibration have been discussed previously. Digital error correcting techniques solve the problem of where one stage makes a conversion that the following stage cannot resolve. For example, consider two 2-bit cascaded stages forming a 4-bit pipeline converter. Assume that $V_{in}^* = (3/8)V_{ref}$, which corresponds to a digital output code of $D = 0110$. Because $(3/8)V_{ref}$ is greater than $(2/8)V_{ref}$, the first stage gives a digital output of $D_1 = 10$. Converting D_1 to an analog voltage gives $(4/8)V_{ref}$, which is subtracted from $(3/8)V_{ref}$ to give $-(1/8)V_{ref}$. Multiplying by 4 gives an amplified residue of $-(4/8)V_{ref}$ applied to the second stage. The second stage will output a digital code of $D_2 = 00$. When D_1 and D_2 are combined, the digital output code is 1000, which is in error. This error could be removed by increasing the number of bits by one at the second stage and using this extra bit to achieve the proper output. Unfortunately, the extra bit does not extend the resolution of the A/D converter. An alternative correction technique is to introduce an intentional offset to all the comparators so that the direction of the A/D decision error is always known. For example, if the intentional offsets are negative when the A/D subconverter makes its decision, then the residue voltage is always negative. The sign of the residue can be used to temporarily correct (decrement) the digital output code without using an extra bit.

One of the sources of nonlinearity in the pipeline A/D converter is the reference scaling nonlinearity and is illustrated in Fig. 8.3-23. When the peak

This is all messed up!
Specify bin breaks, emphasize considering error impact, don't pick sample point at bin edge.

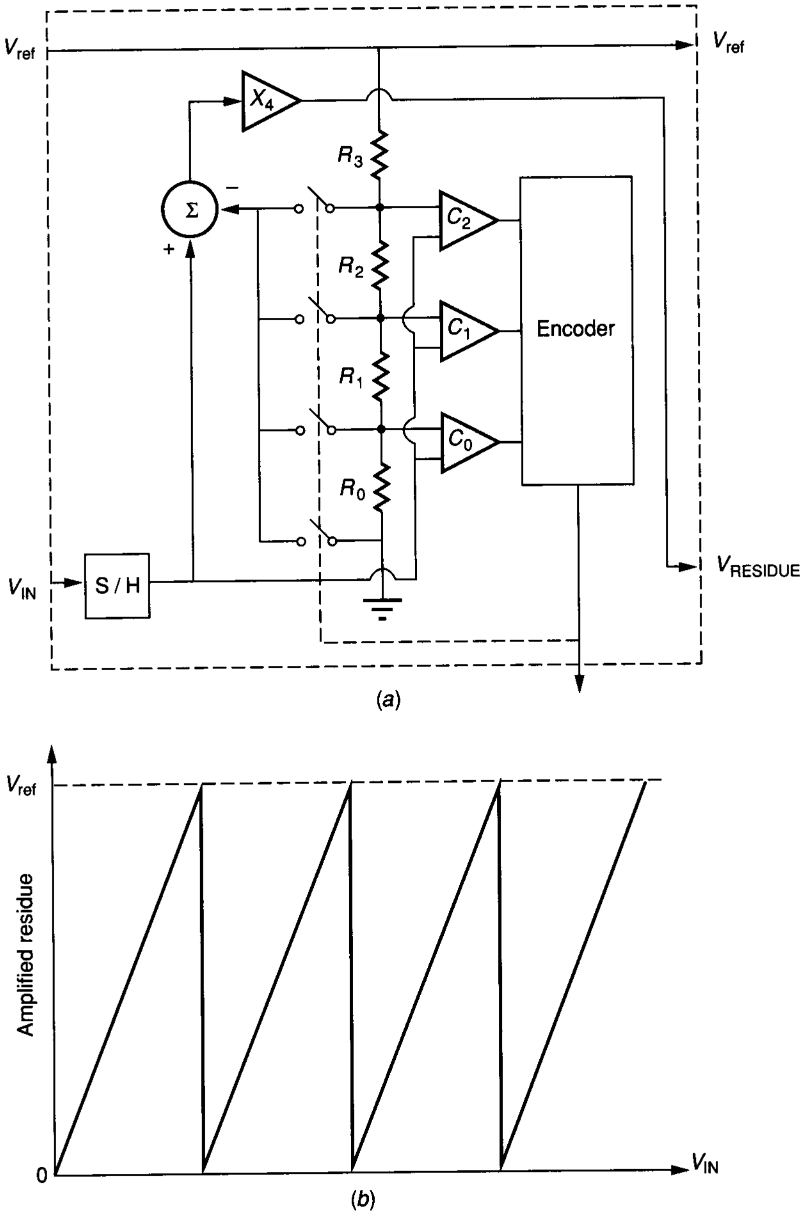

FIGURE 8.3-22
(*a*) Two-bit implementation of a pipeline stage, (*b*) Ideal plot of the amplifier residue versus the analog input.

TABLE 8.3-3
Static error analysis for a pipeline A/D converter

Element	Error(s)	Effect(s)	Solution(s)
S/H	Offset	Offset Nonlinearity	Auto-zeroing Digital error correction
A/D converter	Offset	Offset Nonlinearity	Auto-zeroing Digital error correction
	Nonlinearity	Nonlinearity	Digital error correction
D/A converter	Nonlinearity	Nonlinearity	Trim/calibrate
Interstage amplifier	Offset Gain error	Nonlinearity Nonlinearity	Auto-zeroing Trim/calibrate

of the amplified residual voltage is not equal to V_{ref}, plus and minus differential nonlinearity errors will be produced. For converters up to 10 bits, these errors can be sufficiently reduced by adjusting the residue so that it is always equal to the reference. Feedforward reference correction techniques have been used to extend this error minimization approach to 13 bits.[12]

Pipeline A/D converters offer a flexible approach to achieving high performance. A subranging technique has been used to further enhance the per-

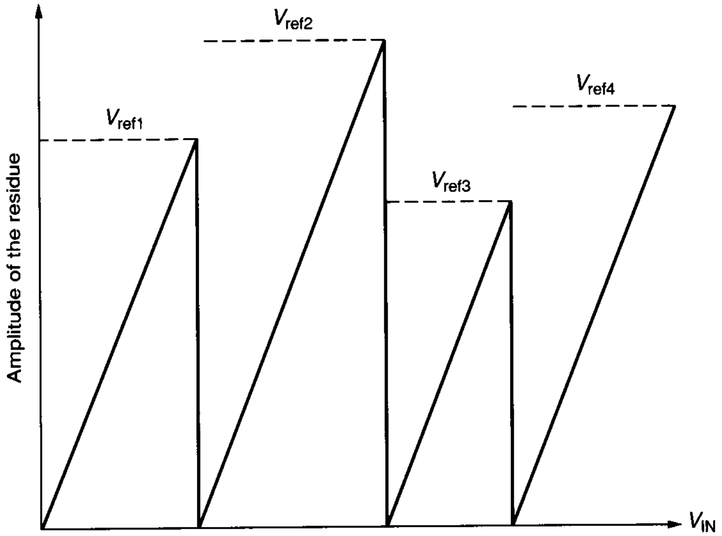

FIGURE 8.3-23
Illustration of feedforward reference correction.

formance of the pipeline A/D converter.[13] In the subranging A/D converter, the conversion is done in several cycles. In each successive cycle, the gain of the residue amplifier is increased until the full scale of the amplified residue voltage is reached. In this manner, errors can be corrected and removed on each successive cycle.

The performance of the pipeline A/D converters depends on the technology used. CMOS converters typically have 13–14 bits of resolution, 5–10 μs conversion times, and 10–30 mW of power dissipation. Bipolar pipeline converters typically have 10 bits of resolution, 200 ns conversion times, and 100–500 mW of power dissipation. The summary of this section will compare these converters with others, including those using BiCMOS technology.

Oversampled A/D converters offer a means of exchanging resolution in time for resolution in amplitude in order to circumvent the need for complex precision analog circuits. A basic oversampled A/D converter architecture is shown in Fig. 8.3-24. This architecture includes a clocked feedback loop, which produces a coarse estimate that oscillates about the true value of the input, and a digital filter, which averages this coarse estimate to obtain a finer approximation. This approximation is accurate to more bits at a lower (decimated) sampling rate. Although trading resolution in time for resolution in amplitude is a simple concept, the key to usefulness of oversampling systems based on the architecture of Fig. 8.3-24 is that the amplitude resolution is enhanced faster than the time resolution is reduced.

The feedback loop with the integrating filter $H(z)$ in Fig. 8.3-24 forces the quantization error in the N-bit estimate to have a high-frequency spectrum. When the low-pass digital filter subsequently reduces the bandwidth of the spectrum, it excludes a more-than-linear proportion of the quantization noise. Typical theoretical signal-to-noise ratio enhancement from the initial N-bit estimate is

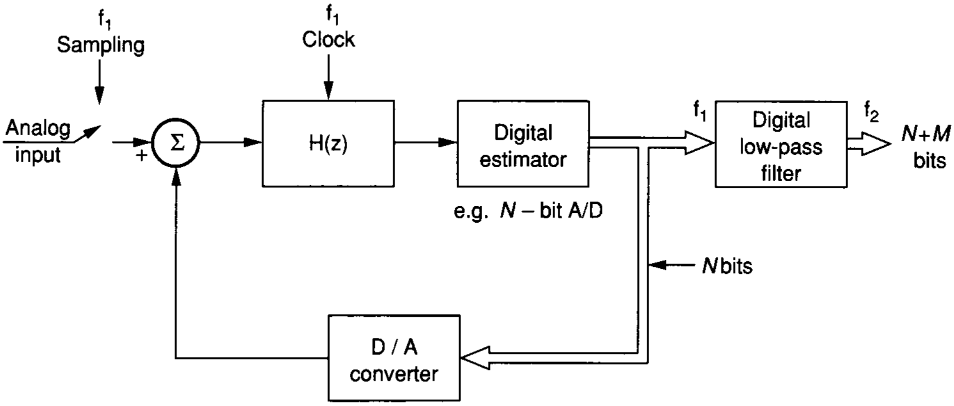

FIGURE 8.3-24
A basic oversampled A/D converter block diagram; $H(z)$ is an integrating filter.

$$\Delta S/N = 9L - 5.2\,\text{dB} \tag{8.3-8}$$

where L is the number of octaves of oversampling when $H(z)$ is first-order. If $H(z)$ is second-order, then the estimate is

$$\Delta S/N = 15L - 13\,\text{dB} \tag{8.3-9}$$

If we assume an oversampling factor of 128, the S/N enhancements for a first-order and second-order oversampling converter are 57.8 dB and 92 dB, respectively.

Of special interest is the case where $N = 1$. In this case, the coarse estimate and the D/A converter have a 1-bit resolution. When $N = 1$, the circuit of Fig. 8.2-24 is called a *delta-sigma modulator*. One advantage of delta-sigma A/D, oversampling converters is that because they obtain resolution by linearly interpolating between two states, they do not require an array of precision binary scaling elements for the D/A converter. This means that nonlinearity errors will not exist. Figure 8.3-25 shows a practical delta-sigma oversampling A/D converter with both one and two integrators in the loop [$H(z)$ first- and second-order].

The output of the delta-sigma modulator is applied to a low-pass digital filter whose function is to operate on the 1-bit signal to remove frequencies and quantization noise above the desired signal bandwidth. The output of the filter is a multibit digital representation at a lower (decimated) sampling rate. Basic delta-sigma modulator systems generate severe noise components for certain input values. In order to remove this noise, a square wave dither frequency within the stopband of the decimating filter randomizes the delta-sigma noise. The square wave dither frequency does not appear at the output of the filter or prevent arbitrary dc inputs from being passed through the filter.

The output of the delta-sigma modulator can be downsampled by a finite-impulse response digital low-pass filter. One possible architecture of the digital

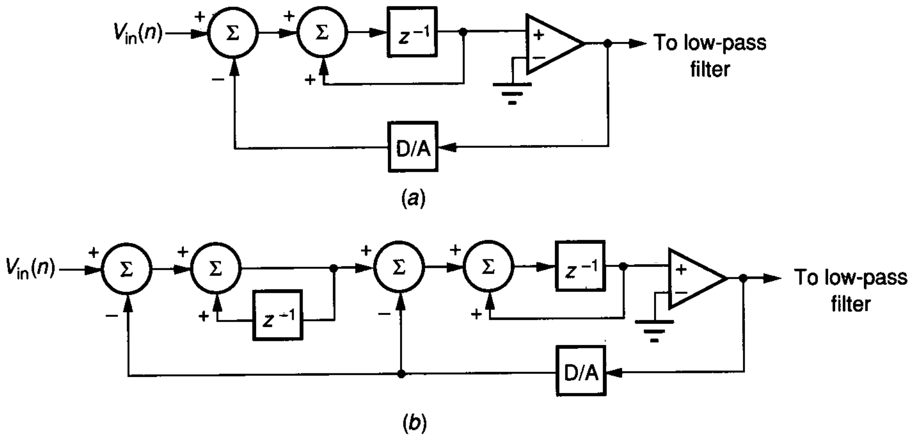

FIGURE 8.3-25
Typical delta-sigma modulators using (*a*) one and (*b*) two integrators in *z*-domain notation.

filter is shown in Fig. 8.3-26. This architecture uses a 256:1 decimation factor and a 1-bit digital input signal. A 1024-point impulse response, of which the symmetric half is stored in a read-only memory, is distributed to four accumulators. No explicit multiplications are necessary because of the 1-bit input signal. The impulse-response coefficients were designed to satisfy the dual objectives of quantization-noise removal and anti-alias filtering. Because the finite-impulse response low-pass filters are relatively insensitive to coefficient roundoff, 6-bit coefficients are adequate to define the impulse response.

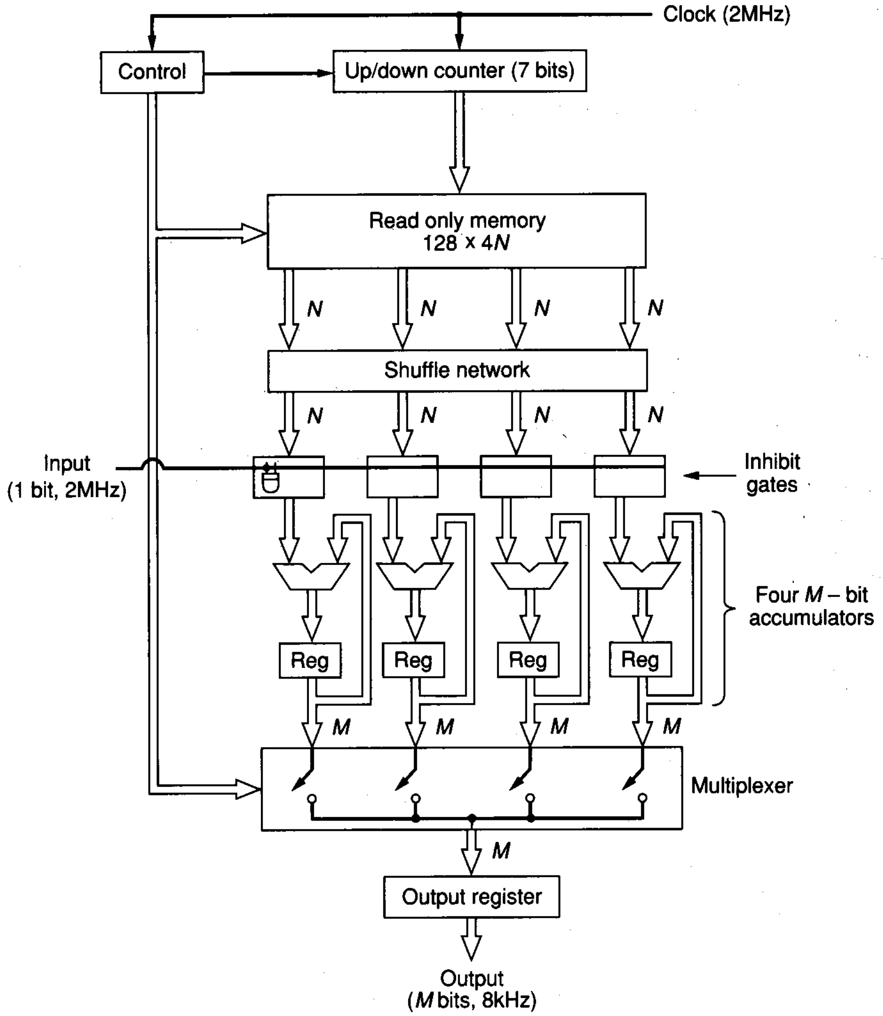

FIGURE 8.3-26
Architecture of a 1024-point finite-impulse response low-pass digital filter with values of $N = 6$ and $M = 15$.

Typical performance for a CMOS delta-sigma oversampled A/D converter is 14–16 bits at a sampling rate of 1–20 kHz, and a power dissipation of 10–30 mW. Dynamic ranges of up to 96 dB are possible, with noise being the limiting factor. The oversampling A/D converter is very suitable to applications such as digital audio.

8.3.5 Summary

A/D conversion techniques compatible with bipolar and CMOS technologies have been presented in this section. The major categories included serial, successive approximation, and parallel A/D converters. Table 8.3-2 provides a summary of these A/D converters. High-performance A/D converters offer improved accuracy (resolution) or speed (conversion time). Examples of high-performance A/D converters include self-calibration methods, pipeline converters with error correcting methods, and delta-sigma oversampled A/D converters. Some of the high-speed A/D converters are beginning to offer capabilities in the video range of frequencies.

The performance of A/D converters is steadily increasing with improvements in technology. Figure 8.3-27 shows the performance of monolithic A/D converters as of 1988.[14] This graph plots converter sampling rate on the horizontal axis and

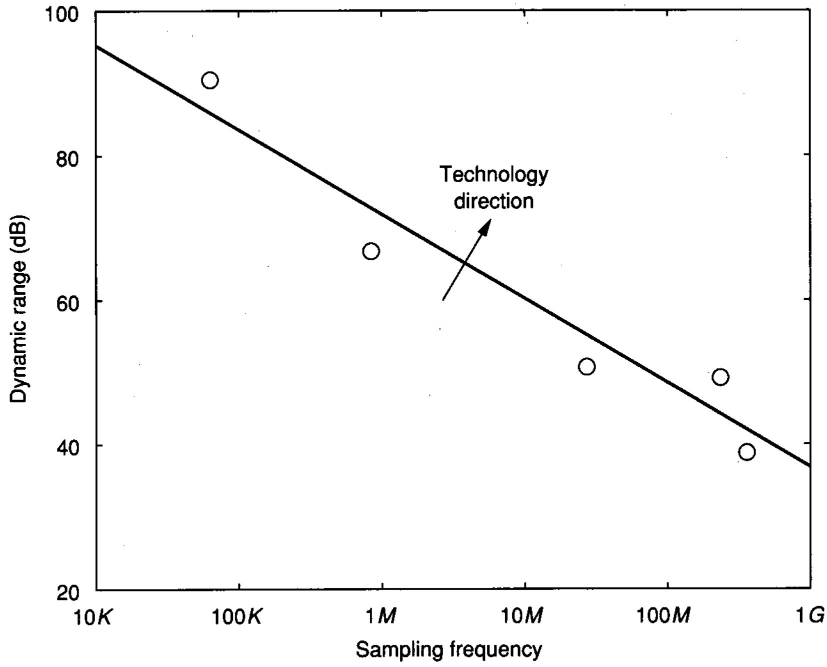

FIGURE 8.3-27
1988 monolithic A/D converter performance in terms of dynamic range versus sampling frequency.

converter dynamic range (signal-to-noise plus distortion) on the vertical axis. Each point on the graph represents the performance of the best monolithic converters. These points fall close to the line drawn on the figure. If such a figure were drawn for each year back to the mid-1970s when monolithic A/D converters first appeared, the movement of the line would be just over 2 dB per year in the vertical direction. The slope of the line is −0.6, supporting the conjecture that it is more difficult to be accurate than fast.

Flash or parallel architectures dominate at and below the 60 dB level. Self-calibrated, successive approximation architectures dominate above the 60 dB line. Self-calibrated delta-sigma oversampled A/D converters will probably provide increasing dynamic ranges.

Figure 8.3-28 is similar to Fig. 8.3-27 except that the various types of converters and types of technology are identified.[15–25] This graph shows how combined technologies such as BiMOS can be used to achieve performance beyond that obtainable with single technologies.

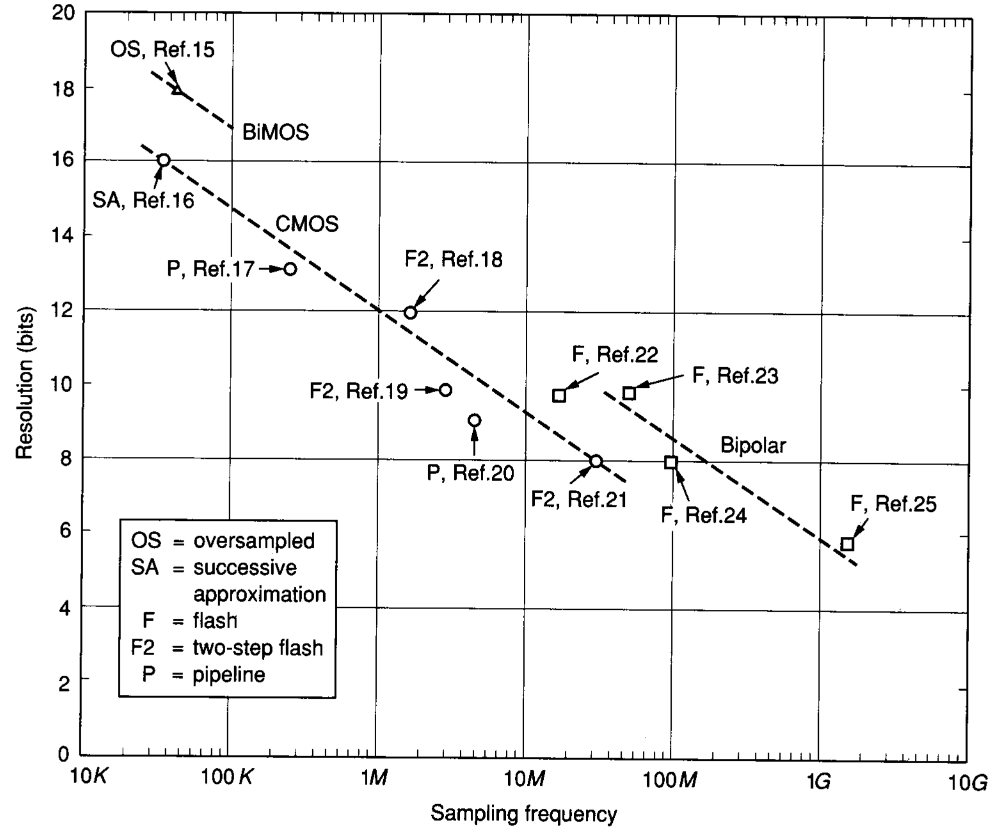

FIGURE 8.3-28

Comparison of high-performance, monolithic A/D converters in terms of resolution versus sampling frequency. Circles, squares, and triangles are CMOS, bipolar, and BiMOS technologies, respectively. The architecture and reference is indicated in the figure (Courtesy of J. P. Hwang).

8.4 CONTINUOUS-TIME FILTERS

One of the largest applications of analog signal processing circuits is in the area of frequency-domain filters. However, until the recent development of switched capacitor circuit techniques,[26] integrated circuit technology was not practical for the implementation of analog filters. The primary reasons were that RC time constants could not be defined with the accuracy necessary for filters and the area requirements for audio filters were unacceptably large. With switched capacitor circuit techniques, the time constants are proportional to capacitor ratios. In integrated circuit technology, it is possible to achieve capacitor ratios sufficiently accurate for filter requirements. In order to design switched capacitor filters, it is necessary to understand the design of continuous-time filters. The objective of this section is to prepare the reader for the following section, which concerns switched capacitor filter design.

For simplicity, we will initially consider linear filters that are classified into four different groups according to the type of gain characteristic. We will assume that the desired gain of the filter is unity in the *passband*, where frequencies are transmitted, and zero in the *stopband*, where frequencies are rejected. Furthermore, we shall assume that under ideal conditions, the passband and stopband are adjacent to each other and that ω_T is the frequency where the transition is made from one band to the other.

The magnitudes of the frequency-dependent gain of the four types of filters that will be considered are shown in Fig. 8.4-1. Figure 8.4-1*a* shows the frequency response of a *low-pass filter*, which has the ideal magnitude response of

$$|H_{LP}(j\omega)| = \begin{cases} 1 & 0 \le \omega \le \omega_T \\ 0 & \omega_T < \omega < \infty \end{cases} \tag{8.4-1}$$

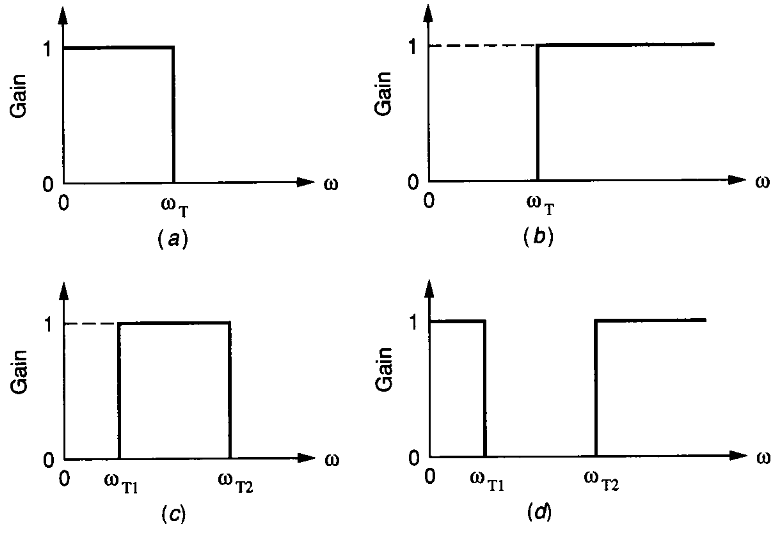

FIGURE 8.4-1
Types of filters: (*a*) Low-pass, (*b*) High-pass, (*c*) Bandpass, (*d*) Band-elimination.

Figure 8.4-1*b* illustrates the magnitude frequency response of a *high-pass filter*, described mathematically as

$$|H_{\mathrm{HP}}(j\omega)| = \begin{cases} 0 & 0 \le \omega < \omega_{\mathrm{T}} \\ 1 & \omega_{\mathrm{T}} \le \omega < \infty \end{cases} \tag{8.4-2}$$

The magnitude frequency response of a *bandpass filter* is shown in Fig. 8.4-1*c*. The ideal frequency response of this type of filter is

$$|H_{\mathrm{BP}}(j\omega)| = \begin{cases} 0 & 0 \le \omega < \omega_{\mathrm{T1}} \\ 1 & \omega_{\mathrm{T1}} \le \omega \le \omega_{\mathrm{T2}} \\ 0 & \omega_{\mathrm{T2}} < \omega < \infty \end{cases} \tag{8.4-3}$$

Finally, the frequency response of a *band-elimination filter* is shown in Fig. 8.4-1*d*. The ideal magnitude frequency response of this filter is given mathematically by

$$|H_{\mathrm{BE}}(j\omega)| = \begin{cases} 1 & 0 \le \omega < \omega_{\mathrm{T1}} \\ 0 & \omega_{\mathrm{T1}} \le \omega \le \omega_{\mathrm{T2}} \\ 1 & \omega_{\mathrm{T2}} < \omega < \infty \end{cases} \tag{8.4-4}$$

The phase shift of each of the filters of Fig. 8.4-1 can be expressed as

$$\operatorname{Arg} H(j\omega) = -\omega T_{\mathrm{d}}, \qquad 0 \le \omega \le \infty \tag{8.4-5}$$

where T_{d} is the time delay of the filter. Because T_{d} is equal to the negative of the derivative of the phase shift, Eq. 8.4-5 defines a system that has a constant time delay of T_{d} seconds for each frequency transmitted through the filter.

In practice, the ideal filter characteristics of Fig. 8.4-1 cannot be realized. It is not possible to have discontinuous changes in transmission as a function of frequency or to have zero transmission over a region or band of frequencies. Consequently, we can only approximate these ideal filter characteristics. Let us consider how we can modify the ideal filter specifications of Fig. 8.4-1 so that they can be realized in practice. Our efforts will be focused on the low-pass filter. The resulting considerations can be easily extended to the other three types of filters.

8.4.1 Low-Pass Filters

Consider the low-pass filter frequency response of Fig. 8.4-2. The frequency range has been divided into three parts. From zero to the passband frequency, ω_{PB}, the gain must be between G_{PB} and unity. From the stopband frequency, ω_{SB}, to infinity, the gain must be between G_{SB} and zero. The frequency range between ω_{PB} and ω_{SB} is called the *transition region*. The gain of the filter is unspecified in this region, although a good approximation would be expected to follow closely a straight line drawn from point A to point B. Another way of interpreting Fig. 8.4-2 is to say that the filter response must fall in the shaded regions. The reasons for the transition region and a nonzero value of G_{SB} are clear from the concepts of circuit theory. However, it may not be clear why G_{PB} cannot be unity. The answer is that there is a tradeoff between how close G_{PB} is to unity and the width of the transition region. We shall see that permitting G_{PB} to be less than unity allows the value of $\omega_{\mathrm{SB}}/\omega_{\mathrm{PB}}$ to approach unity and, thus, a smaller transition region.

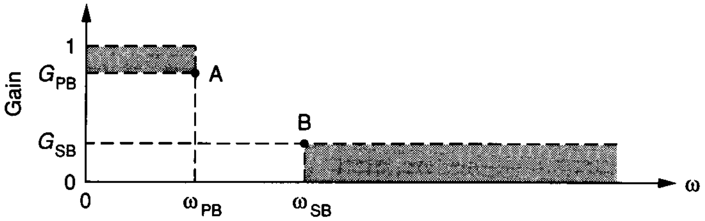

FIGURE 8.4-2
Practical specifications for a low-pass filter.

All transfer functions of active and/or passive filters with a finite number of lumped components are rational functions in s. With this observation, the filter design problem is generally partitioned into two parts: approximation and synthesis. In the approximation problem, a rational function (with real coefficients), often termed an approximating function, is derived to meet the filter requirements. The synthesis problem concerns the design of a circuit that has the approximating function as its transfer function.

There are several types of filter approximation functions that satisfy the requirements of Fig. 8.4-2. These filter types can all be expressed by the rational polynomial transfer function given as

$$H(s) = \frac{b_0 + b_1 s + b_2 s^2 + \cdots + b_{n-1}s^{n-1} + b_n s^n}{a_0 + a_1 s + a_2 s^2 + \cdots + a_{n-1}s^{n-1} + a_n s^n} \tag{8.4-6}$$

The order of this transfer function is n. The types of filter approximations are generally classified by some aspect of their frequency response. For example, the *Butterworth filter approximation* has a magnitude response that is maximally flat in the passband and monotonically decreasing in the transition and passband regions. For the Butterworth low-pass filter approximation, $b_0 = 1$ and $b_i = 0$ for all $i \geq 1$ in Eq. 8.4-6. An example of some Butterworth filter approximations for various values of n is given in Fig. 8.4-3. It is customary to normalize the passband frequency ω_{PB} to 1 rps when using the approximation. The normalized ω_{PB} is transformed to the desired transition frequency during the synthesis phase of the

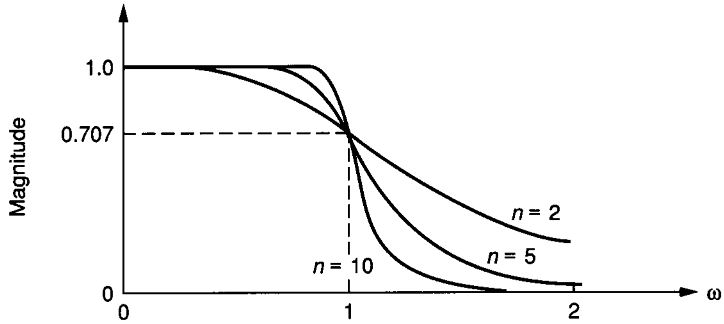

FIGURE 8.4-3
Magnitude characteristics of a Butterworth approximation for orders of $n = 2, 5$, and 10.

problem. Figure 8.4-3 shows only the magnitude characteristics of the Butterworth approximation. All approximating functions are characterized by both a magnitude and phase response although many filter specifications are based solely on the magnitude response. The phase shift of the Butterworth filter approximation may be found in other sources.[27]

Several other filter approximations are often used in filter design. One of these is called the *Chebyshev filter approximation* and is illustrated in Fig. 8.4-4 for various values of n. It is seen that the magnitude of the Chebyshev filter approximation ripples in the passband and is monotonically decreasing in the transition and stopband regions. The advantage of the Chebyshev filter approximation is that for a given n, the slope of the magnitude response in the vicinity of ω_{PB} is steeper than that of a Butterworth filter approximation, resulting in a smaller transition region if G_{PB} and G_{SB} are identical.

A third popular filter approximation is called the *elliptic filter approximation* and is shown in Fig. 8.4-5. This approximation ripples both in the stopband and the passband, but is monotonically decreasing in the transition region. The elliptic filter approximation has the narrowest transition region of any type of filter characteristics considered here, given the same values of n, G_{PB}, and G_{SB}. Other filter approximations have been tabulated and feature other frequency characteristics such as linear phase shift.

Fortunately, there is considerable tabulated information available on standard filter approximations.[27–34] The objective of the filter designer in using this tabulated information is to obtain a polynomial such as Eq. 8.4-6, its roots, or a passive RLC ladder network. Any of these three form the starting point for the design using one of many methods to arrive at an active filter realization of the filter approximation. Switched capacitor filter design can start from the active filter realization, the approximating function, or from the original information used to characterize the filter requirements.

To use the tabulated information, it is necessary to determine the order n from the filter specification. Because n determines the number of components and complexity of a filter, the minimum value of n is generally selected. Although the filter specification can be given as shown in Fig. 8.4-2, it is more customary to

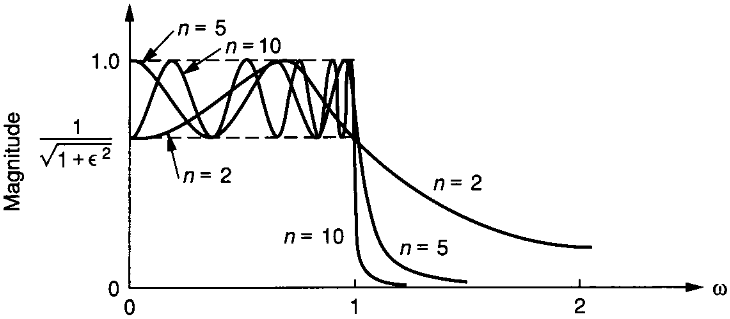

FIGURE 8.4-4
Magnitude characteristics of a Chebyshev approximation for an arbitrary ϵ and for orders n = 2, 5, and 10.