Vietnam National University HCMC

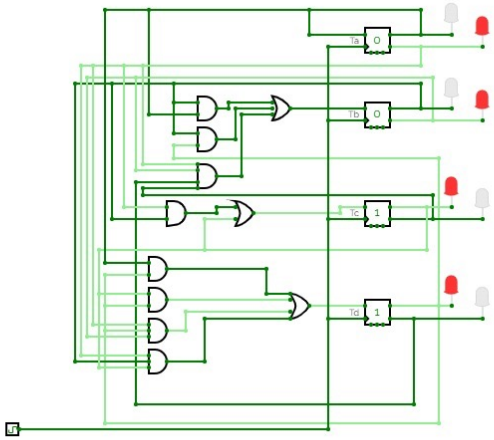
International University

School of Electrical Engineering

EE053IU

Digital Logic Design

Lecture 12: Signal Conversion and Processing



INSTRUCTOR: Dr. Vuong Quoc Bao

1. Analog-to-Digital Conversion

Sampling and Filtering

- An anti-aliasing filter and a sample-and-hold circuit are two functions typically found in a digital signal processing system.
- The sample-and-hold function does two operations, the first of which is sampling. Sampling is the process of taking a sufficient number of discrete values at points on a waveform that will define the shape of the waveform.
- The more samples you take, the more accurately you can define a waveform.
- Sampling converts an analog signal into a series of impulses, each representing the amplitude of the signal at a given instant in time.

Sampling and Filtering

- All analog signals (except a pure sine wave) contain a spectrum of component frequencies. For a pure sine wave, these frequencies appear in multiples called harmonics.
- The harmonics of an analog signal are sine waves of different frequencies and amplitudes.
- When the harmonics of a given periodic waveform are added, the result is the original signal.

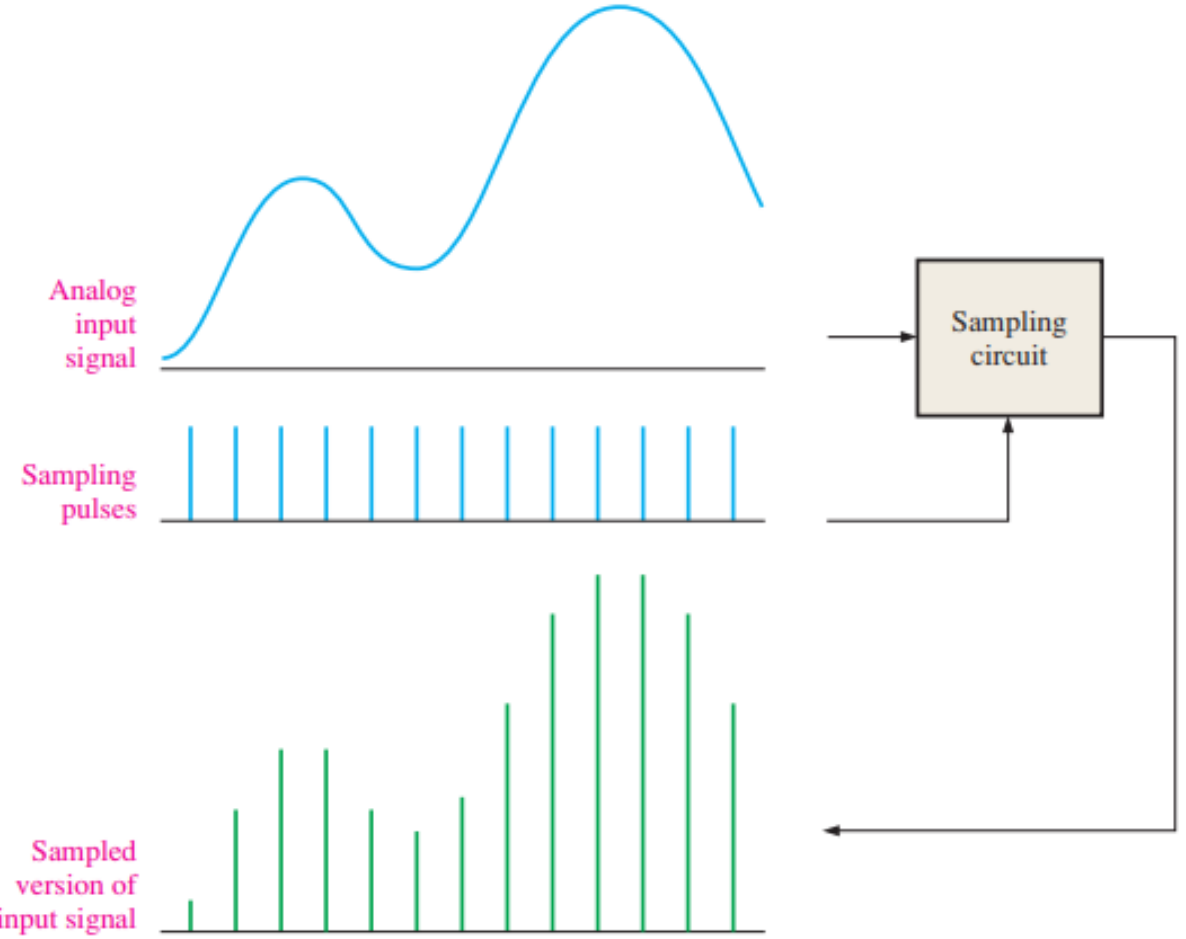
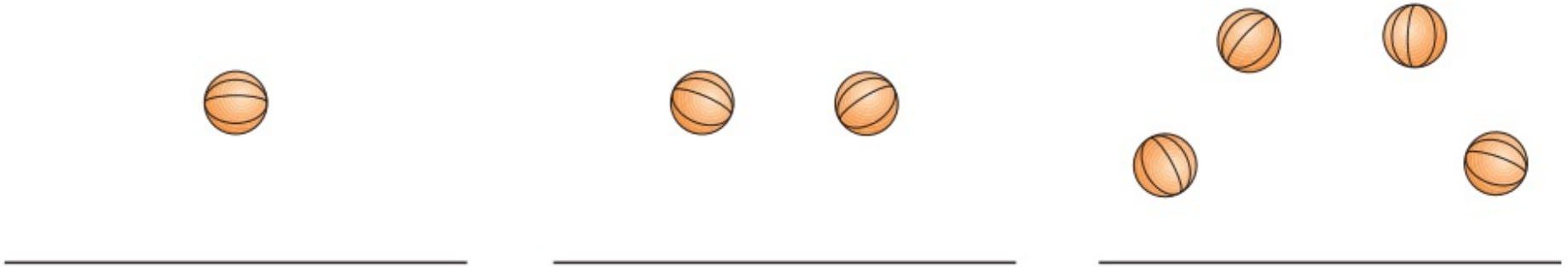


FIGURE 12–1 Illustration of the sampling process.

The Sampling Theorem

- The sampling theorem states that, in order to represent an analog signal, the sampling frequency, f_{sample}, must be at least twice the highest frequency component $f_{a(max)}$ of the analog signal.
- The frequency $f_{a(max)}$ is known as the Nyquist frequency

$$f_{sample} > 2f_{a(max)}$$



(a) One sample of a ball during a single bounce

(b) Two samples of a ball during a single bounce. This is the absolute minimum required to tell anything about its movement, but generally insufficient to describe its path.

(c) Four samples of a ball during a single bounce form a rough picture of the path of the ball.

FIGURE 12–2 Bouncing ball analogy of sampling theory.

The Need for Filtering

- Low-pass filtering is necessary to remove all frequency components (harmonics) of the analog signal that exceed the Nyquist frequency. If there are any frequency components in the analog signal that exceed the Nyquist frequency, an unwanted condition known as **aliasing** will occur. An alias is a signal produced when the sampling frequency is not at least twice the signal frequency

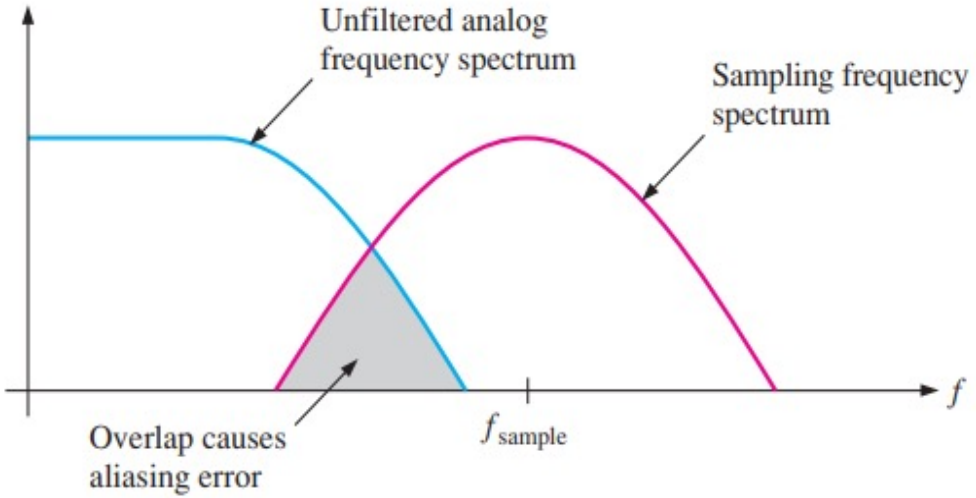


FIGURE 12–3 A basic illustration of the condition $f_{sample} < 2f_{a(max)}$.

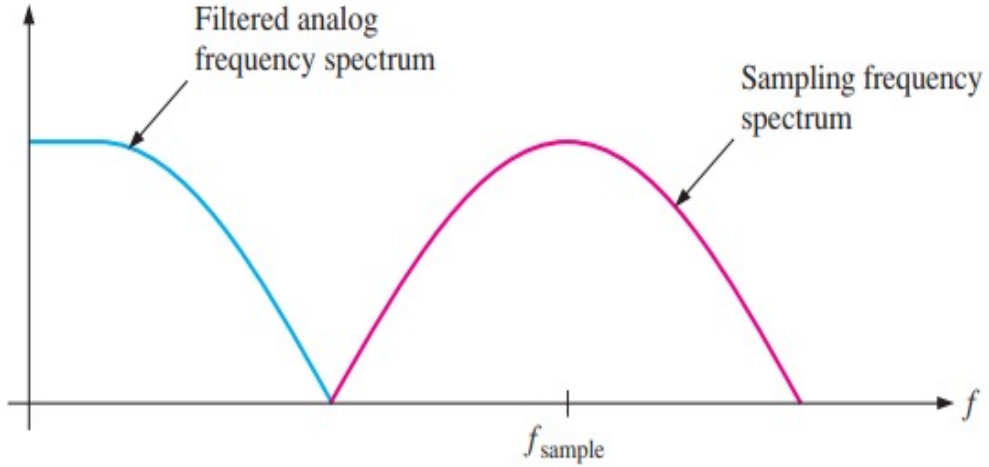


FIGURE 12–4 After low-pass filtering, the frequency spectra of the analog and the sampling signals do not overlap, thus eliminating aliasing error.

Holding the Sampled Value

- The holding operation is the second part of the sample-and-hold function. After filtering and sampling, the sampled level must be held constant until the next sample occurs. This is necessary for the ADC to have time to process the sampled value. This sample-and-hold operation results in a "stairstep" waveform that approximates the analog input waveform.

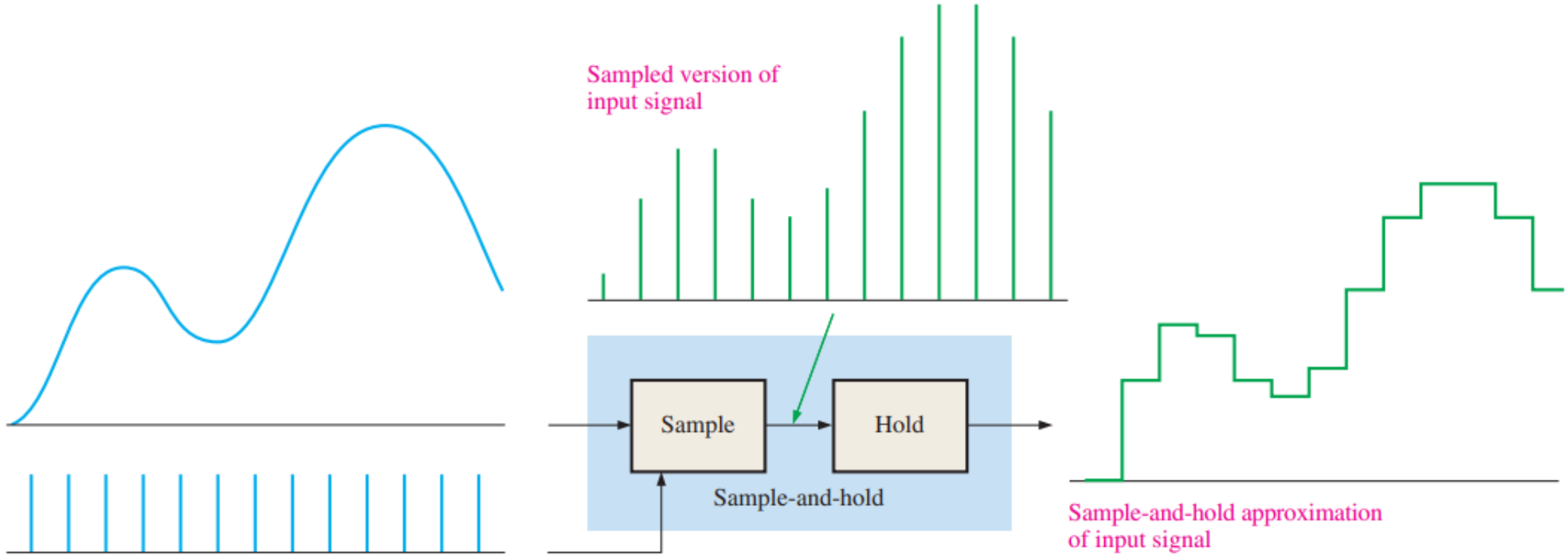


FIGURE 12–5 Illustration of a sample-and-hold operation.

Analog-to-Digital Conversion

- Analog-to-digital conversion is the process of converting the output of the sample and hold circuit to a series of binary codes that represent the amplitude of the analog input at each of the sample times.

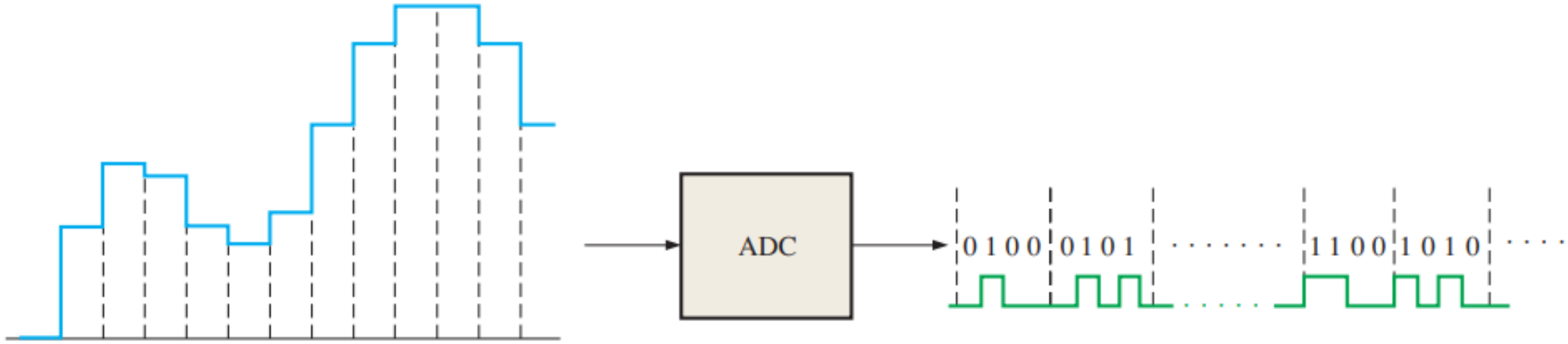


FIGURE 12–6 Basic function of an analog-to-digital converter (ADC) (The binary codes and number of bits are arbitrarily chosen for illustration only). The ADC output waveform that represents the binary codes is also shown.

Quantization

The process of converting an analog value to a code is called quantization. During the quantization process, the ADC converts each sampled value of the analog signal to a binary code. The more bits that are used to represent a sampled value, the more accurate is the representation.

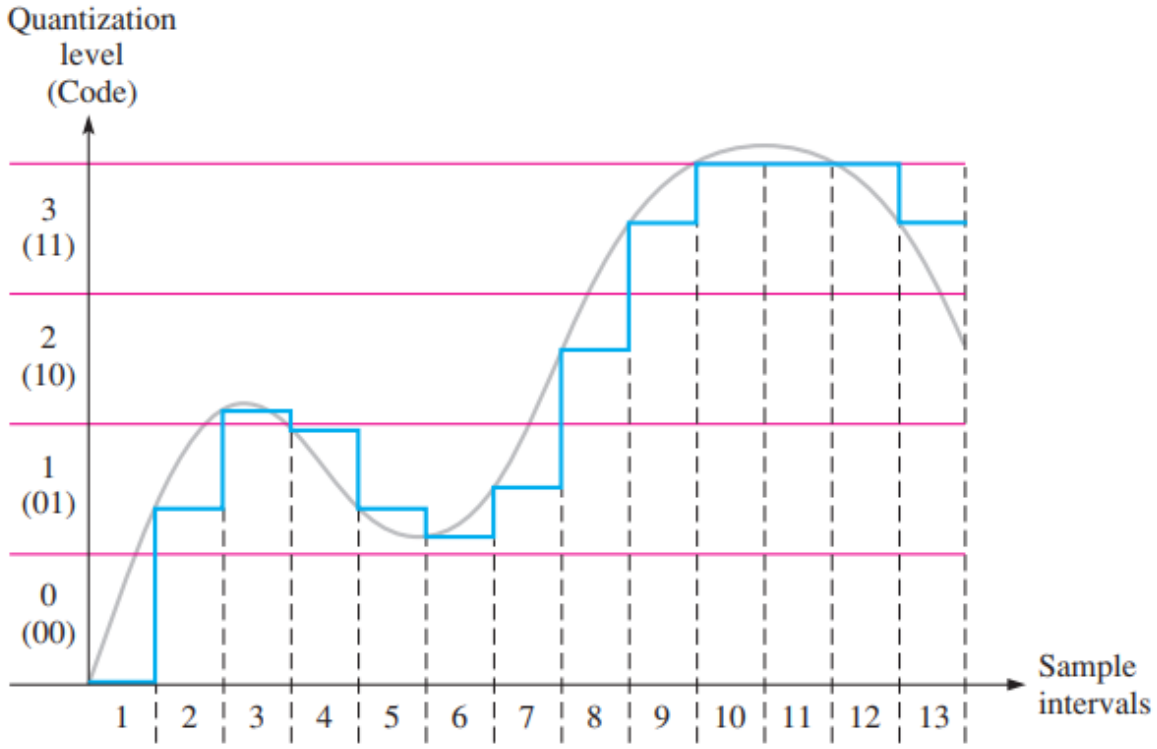


FIGURE 12–7 Sample-and-hold output waveform with four quantization levels. The original analog waveform is shown in light gray for reference.

TABLE 12–1

Two-bit quantization for the waveform in Figure 12–7.

Sample Interval	Quantization Level	Code
1	0	00
2	1	01
3	2	10
4	1	01
5	1	01
6	1	01
7	1	01
8	2	10
9	3	11
10	3	11
11	3	11
12	3	11
13	3	11

Quantization

The process of converting an analog value to a code is called quantization. During the quantization process, the ADC converts each sampled value of the analog signal to a binary code. The more bits that are used to represent a sampled value, the more accurate is the representation.

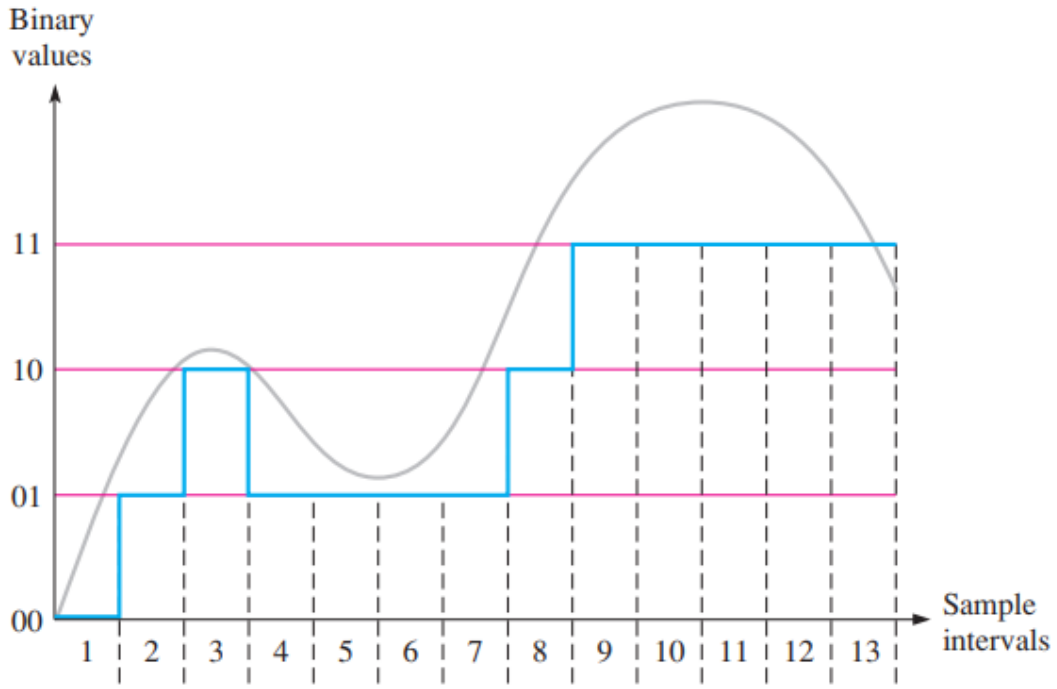


FIGURE 12–8 The reconstructed waveform in Figure 12–7 using four quantization levels (2 bits). The original analog waveform is shown in light gray for reference.

TABLE 12–1

Two-bit quantization for the waveform in Figure 12–7.

Sample Interval	Quantization Level	Code
1	0	00
2	1	01
3	2	10
4	1	01
5	1	01
6	1	01
7	1	01
8	2	10
9	3	11
10	3	11
11	3	11
12	3	11
13	3	11

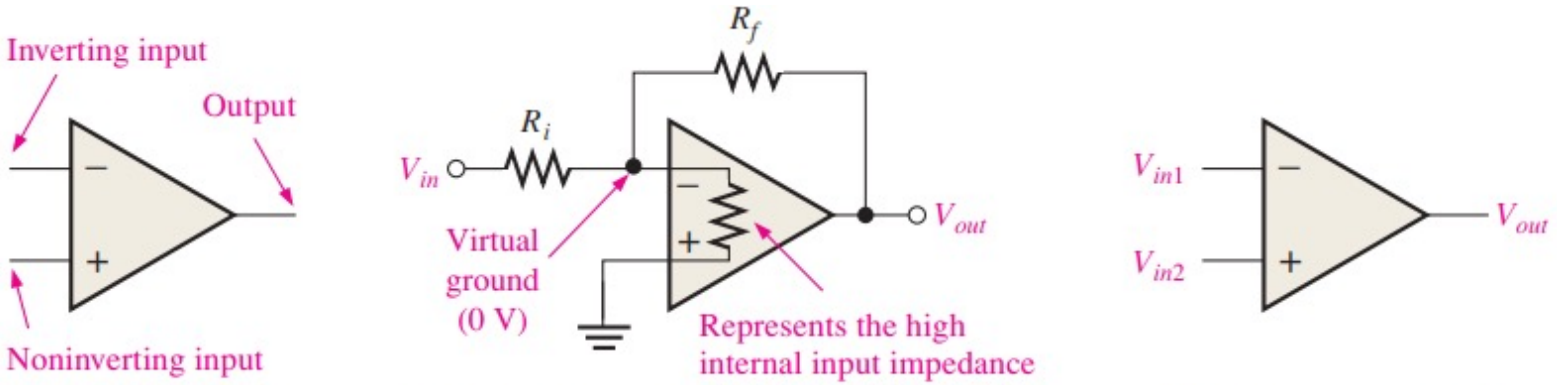
2. The Methods of Analog-to-Digital Conversion

Two important ADC parameters are resolution, which is the number of bits, and throughput, which is the sampling rate an ADC can handle in units of samples per second (sps).

A Quick Look at an Operational Amplifier

An op-amp is a linear amplifier that has two inputs (inverting and noninverting) and one output.

$$\frac{V_{out}}{V_{in}} = -\frac{R_f}{R_i}$$



(a) Op-amp symbol

(b) Op-amp as an inverting amplifier with gain of R_f/R_i

(c) Op-amp as a comparator

FIGURE 12–11 The operational amplifier (op-amp).

Flash (Simultaneous) Analog-to-Digital Converter

- The number of bits used in an ADC is its resolution.
- The large number of comparators necessary for a reasonable-sized binary number is one of the disadvantages of the flash ADC.
- Its chief advantage is that it provides a fast conversion time because of a high throughput, measured in samples per second (sps).

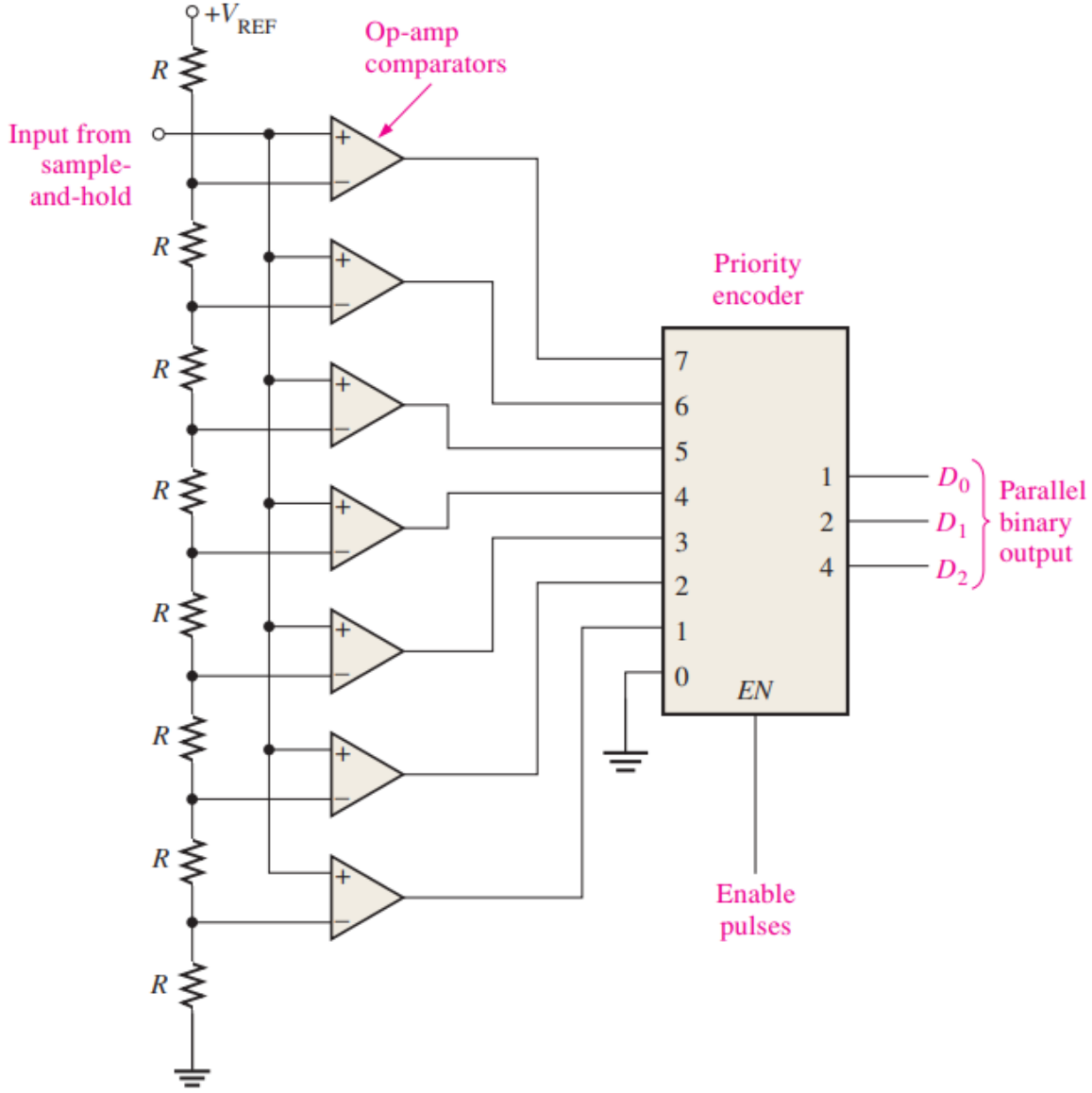


FIGURE 12–12 A 3-bit flash ADC.

EXAMPLE 12–1

Determine the binary code output of the 3-bit flash ADC in Figure 12–12 for the input signal in Figure 12–13 and the encoder enable pulses shown. For this example, $V_{REF} = +8$ V.

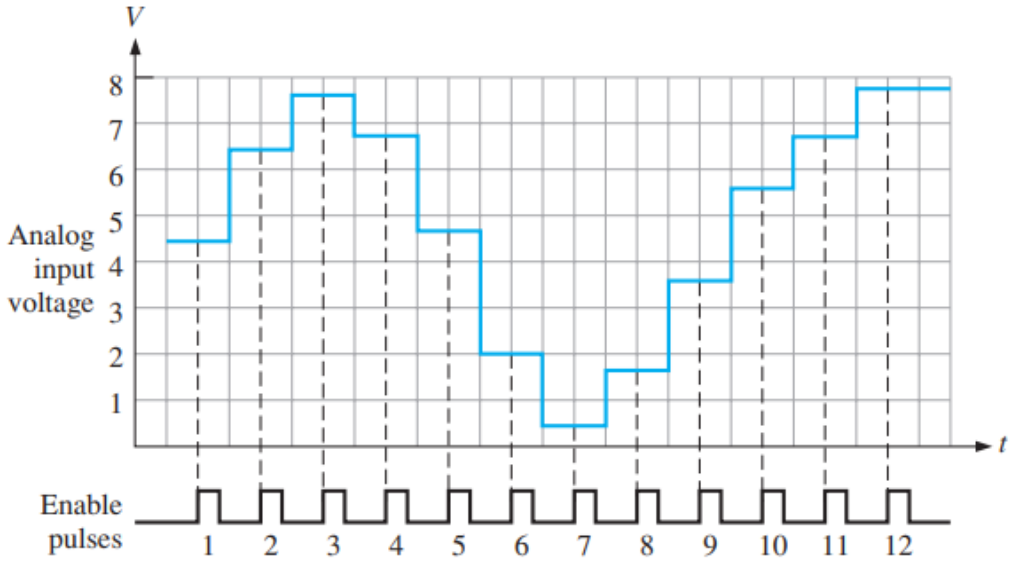


FIGURE 12–13 Sampling of values on a waveform for conversion to binary code.

Solution

The resulting digital output sequence is listed as follows and shown in the waveform diagram of Figure 12–14 in relation to the enable pulses:

100, 110, 111, 110, 100, 010, 000, 001, 011, 101, 110, 111

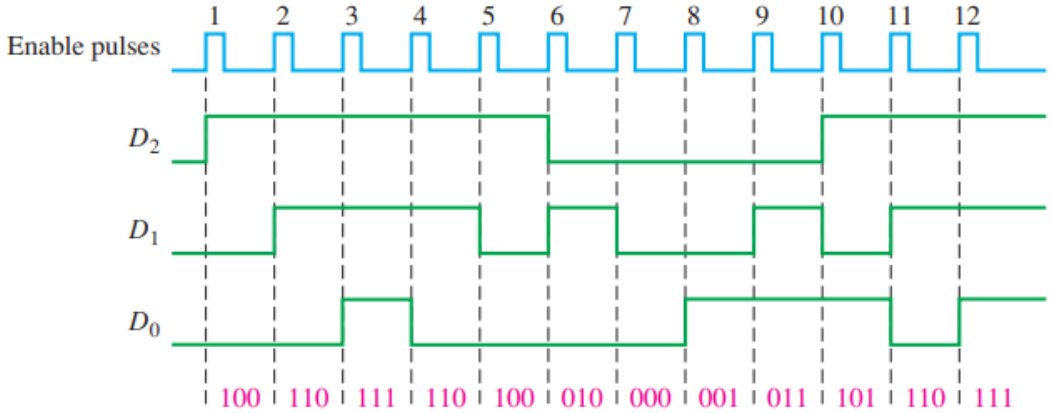


FIGURE 12–14 Resulting digital outputs for sample-and-hold values. Output D_0 is the LSB of the 3-bit binary code.

Dual-Slope Analog-to-Digital Converter

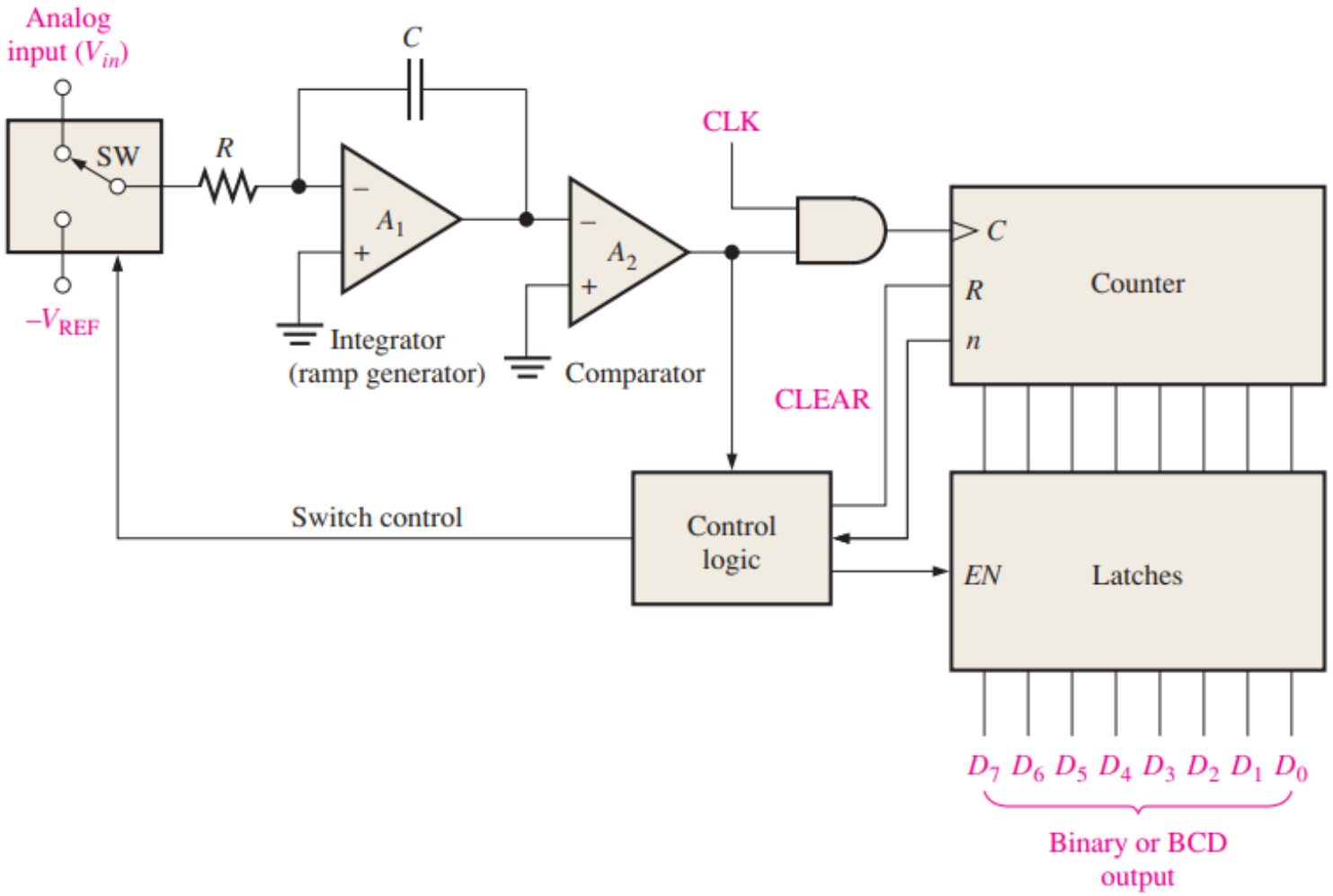
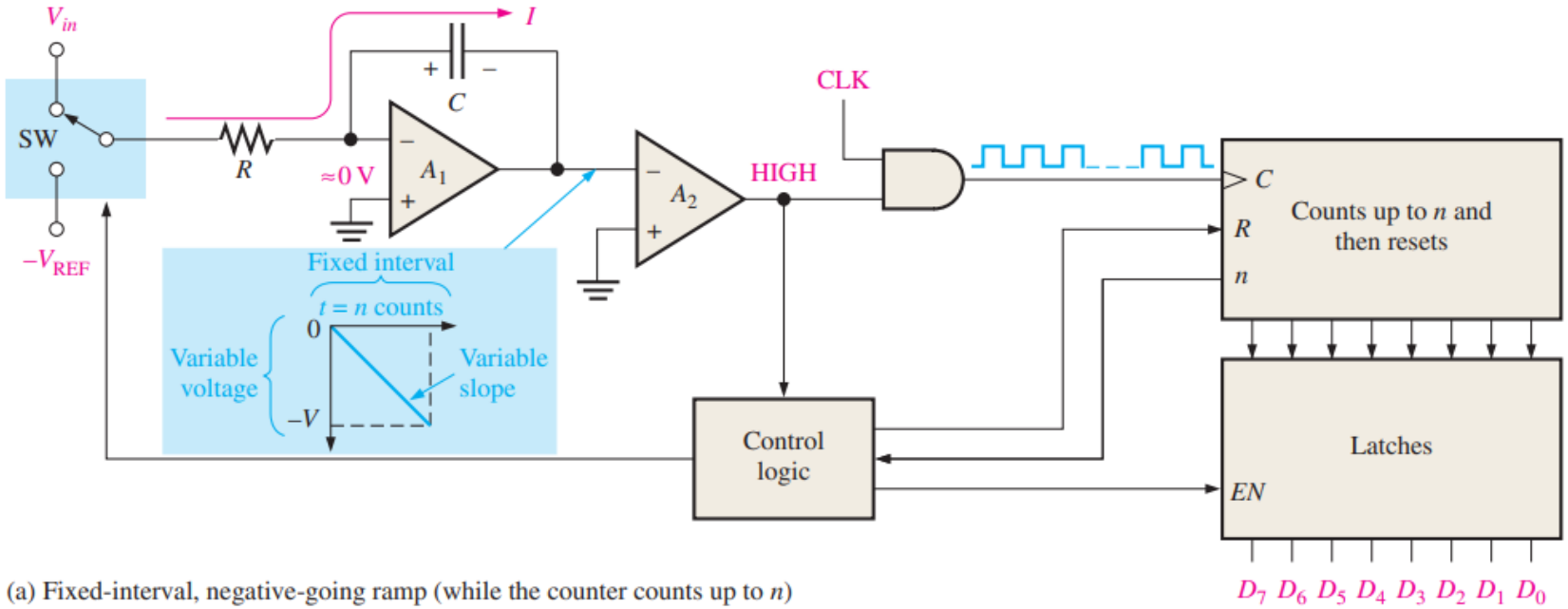
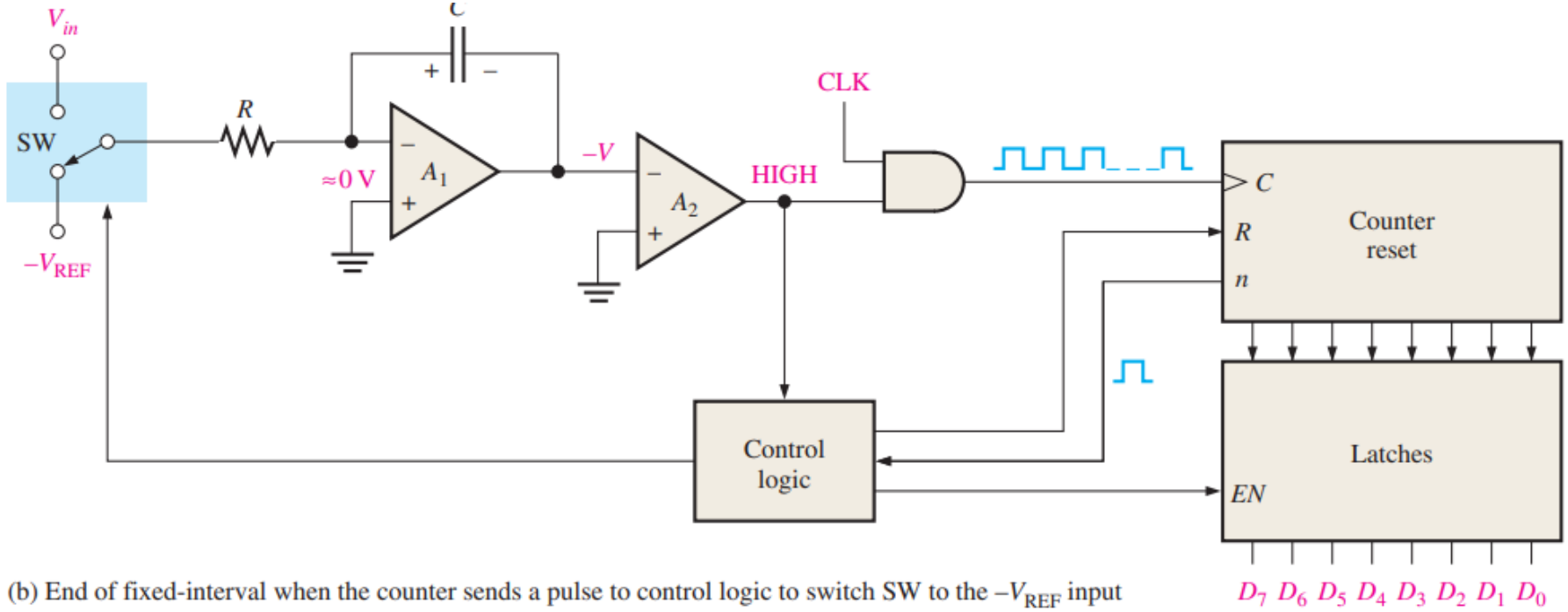


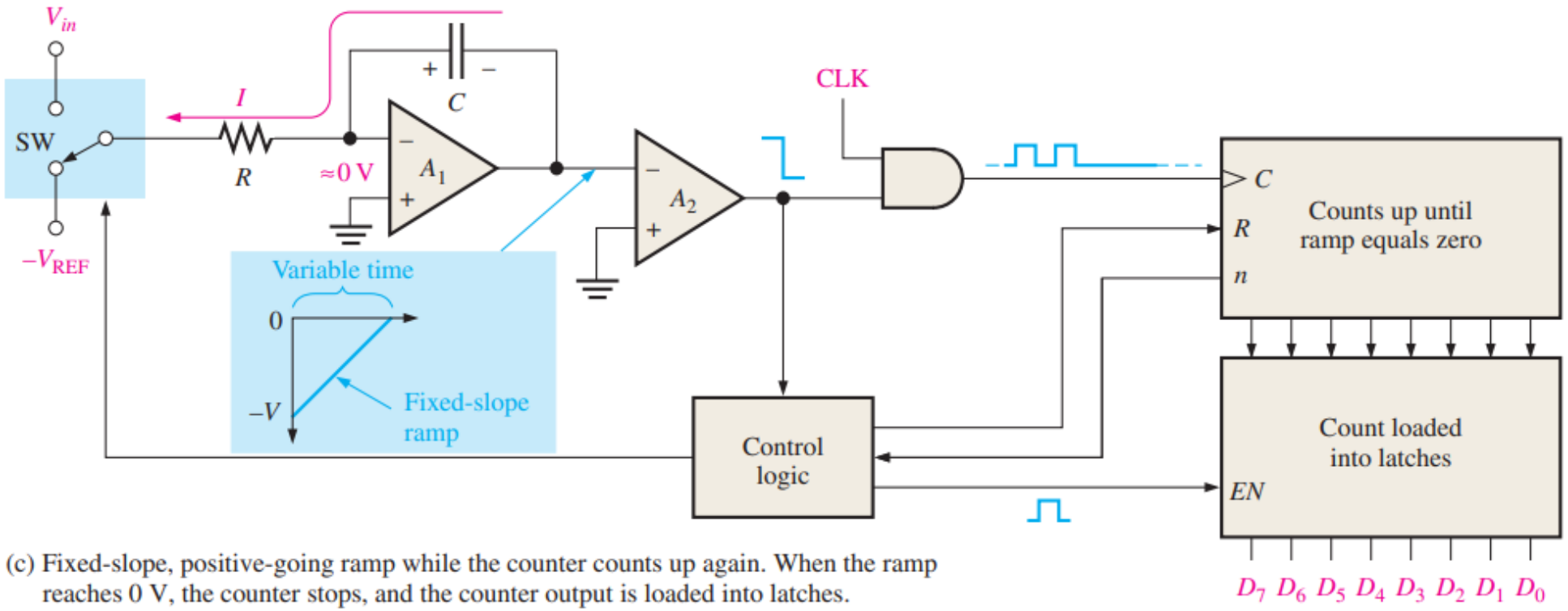
FIGURE 12–15 Basic dual-slope ADC.



(a) Fixed-interval, negative-going ramp (while the counter counts up to n)



(b) End of fixed-interval when the counter sends a pulse to control logic to switch SW to the $-V_{REF}$ input



(c) Fixed-slope, positive-going ramp while the counter counts up again. When the ramp reaches 0 V, the counter stops, and the counter output is loaded into latches.

FIGURE 12–16 Illustration of dual-slope conversion.

Successive-Approximation Analog-to-Digital Converter

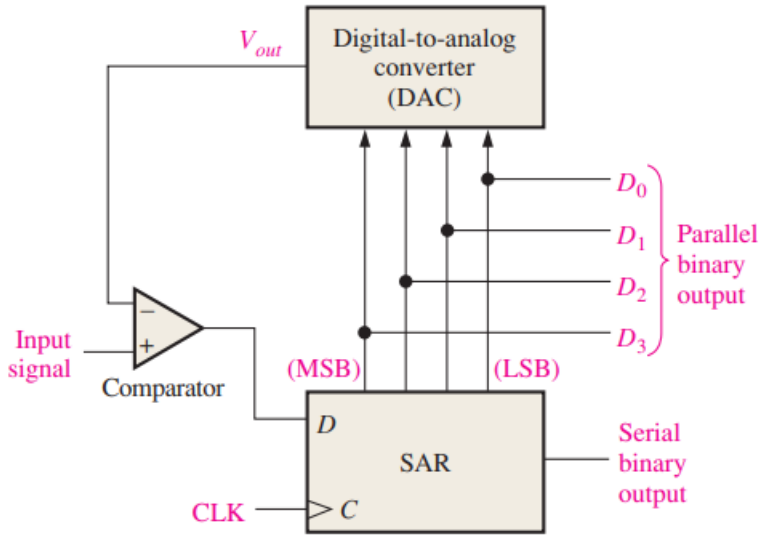
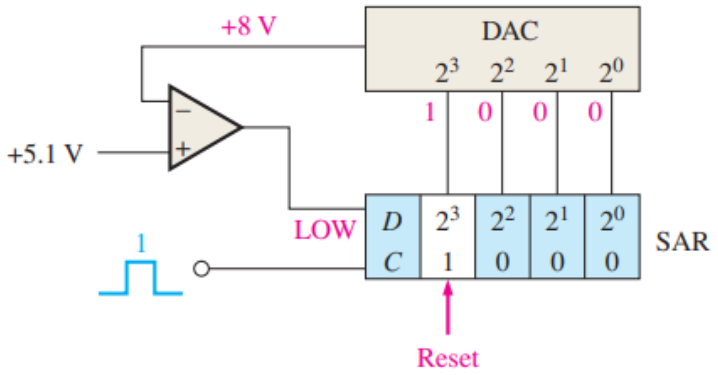
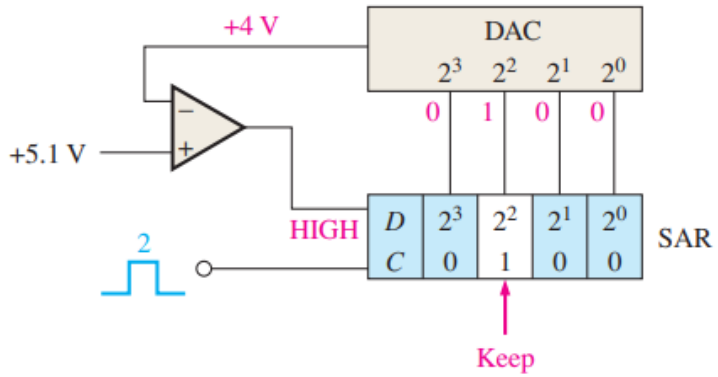


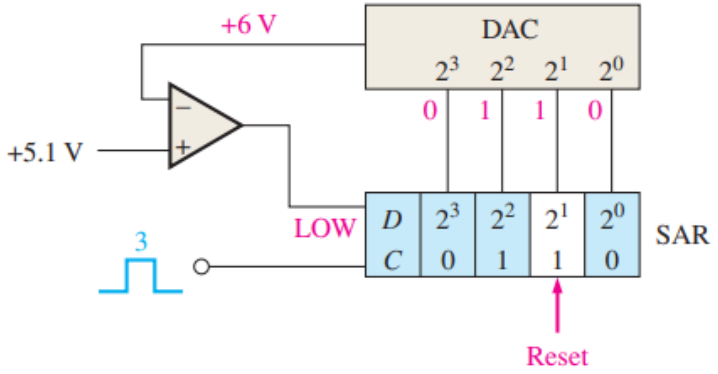
FIGURE 12–17 Successive-approximation ADC.



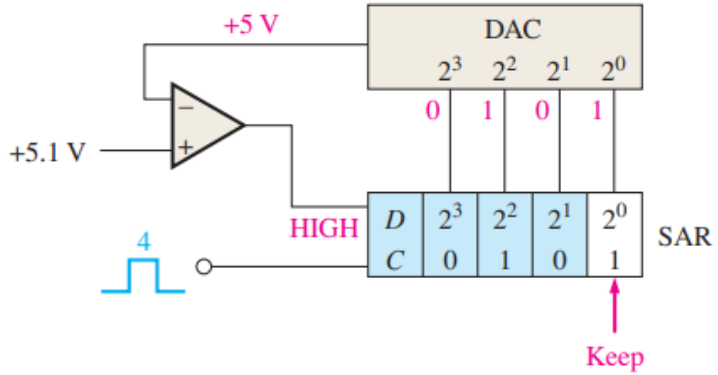
(a) MSB trial



(b) 2^2-bit trial



(c) 2^1-bit trial



(d) LSB trial (conversion complete)

FIGURE 12–18 Illustration of the successive-approximation conversion process.

Sigma-Delta Analog-to-Digital Converter

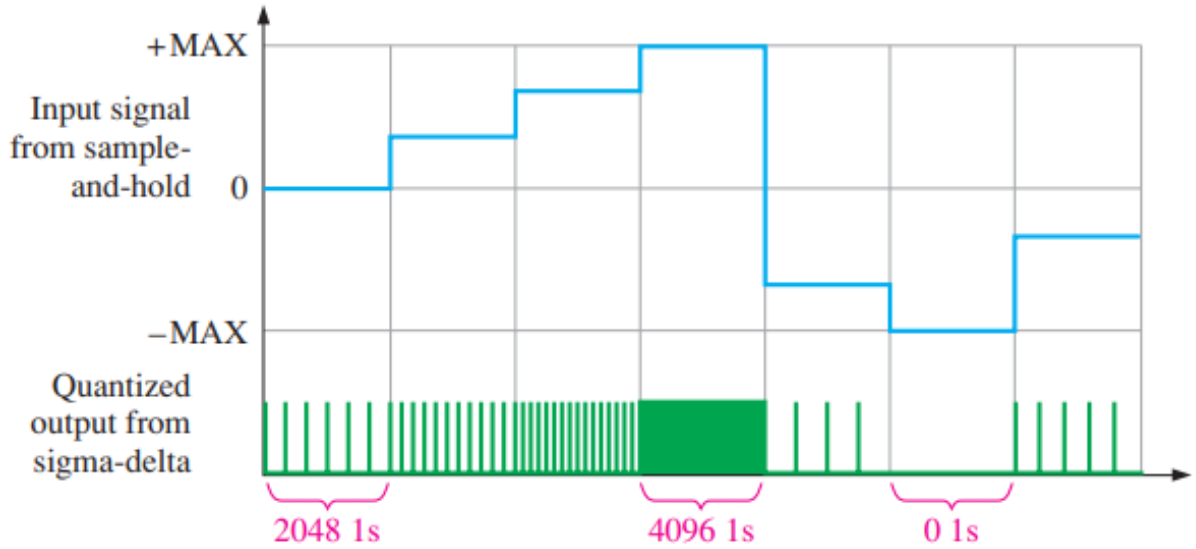


FIGURE 12–20 A simplified illustration of sigma-delta analog-to-digital conversion.

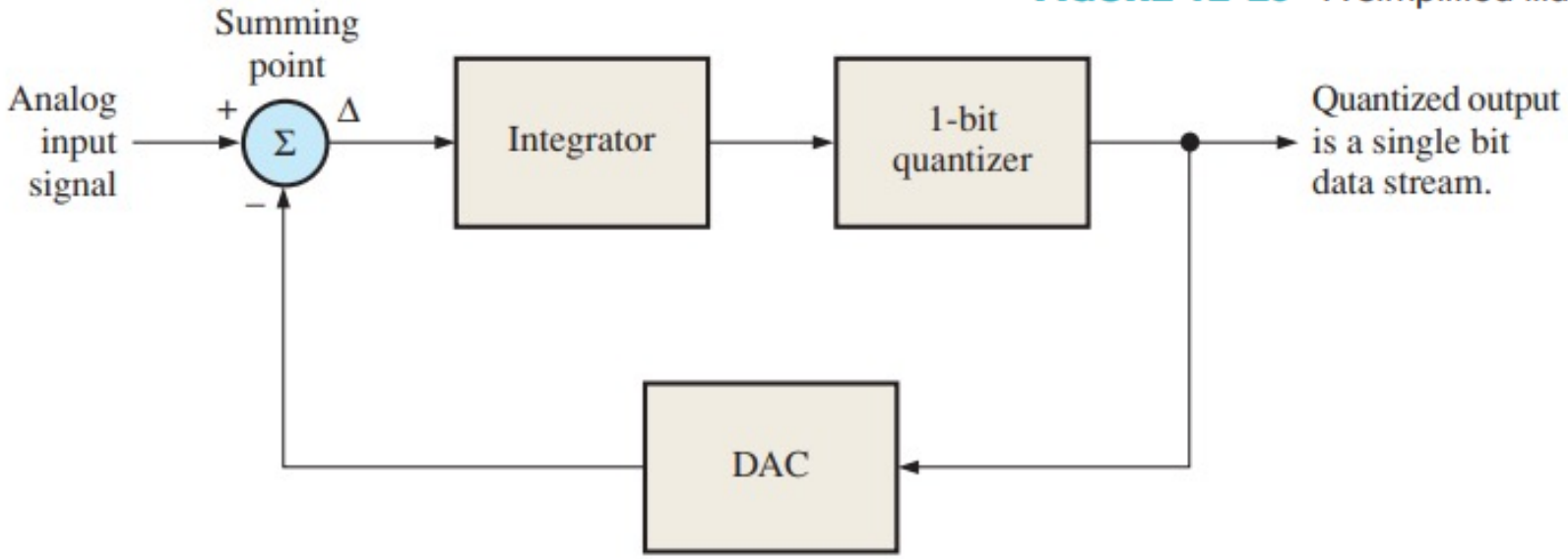


FIGURE 12–21 Partial functional block diagram of a sigma-delta ADC.

Testing Analog-to-Digital Converters

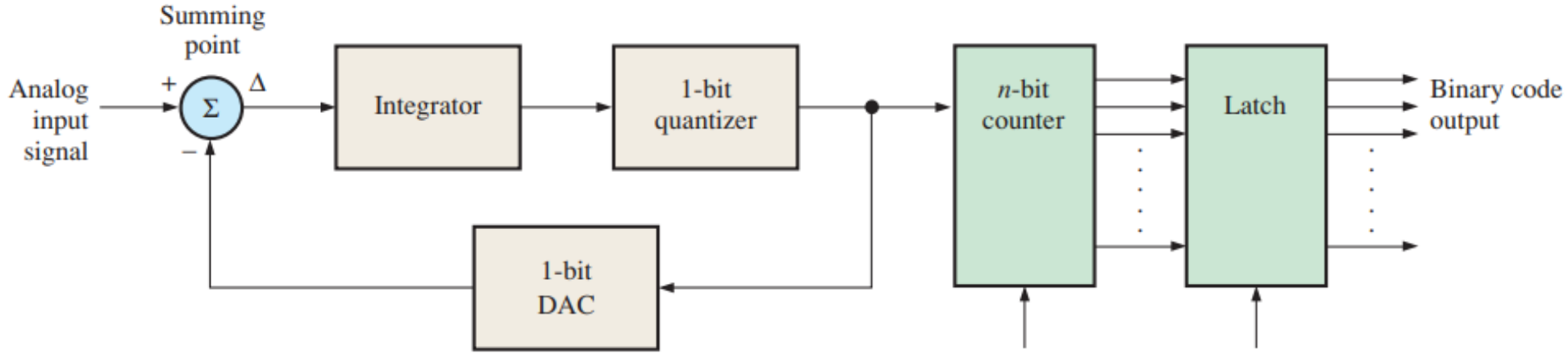


FIGURE 12–22 One type of sigma-delta ADC.

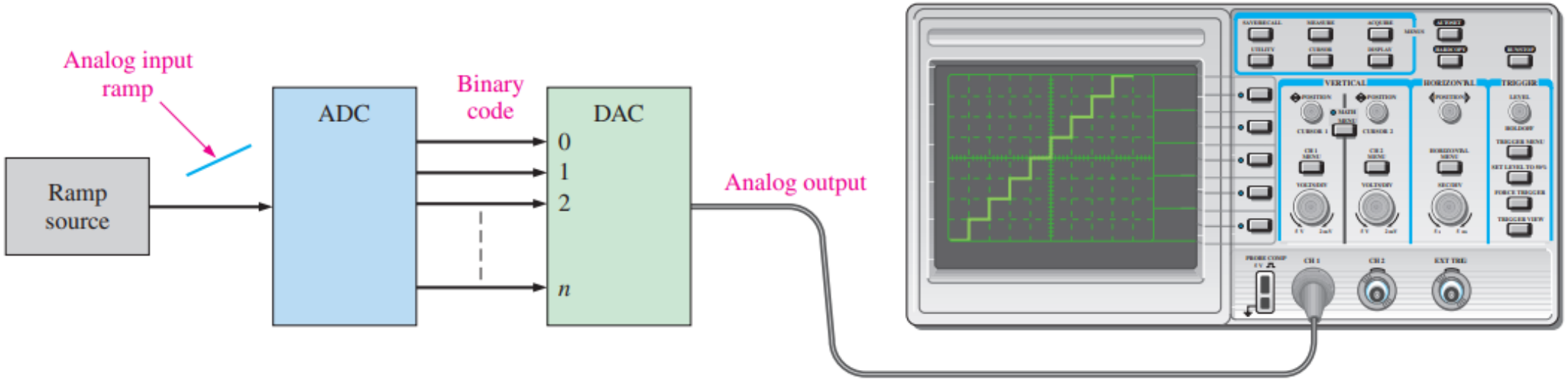


FIGURE 12–23 A method for testing ADCs.

Analog-to-Digital Conversion Errors

Missing Code

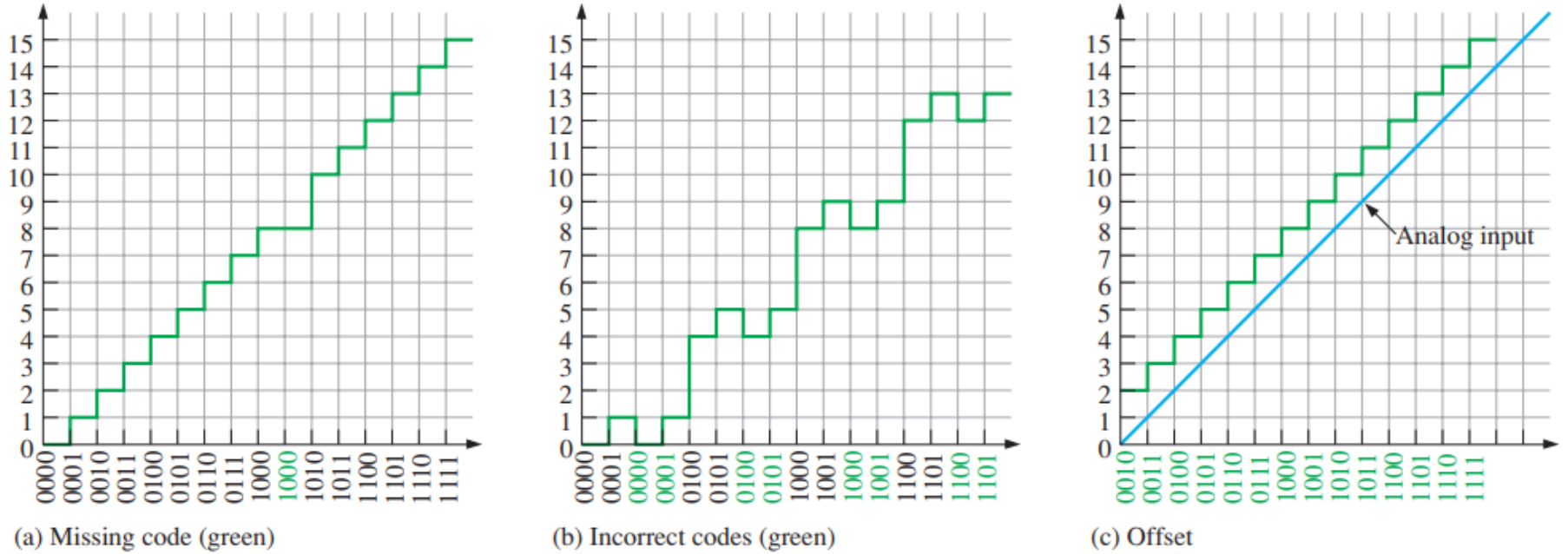


FIGURE 12–24 Illustrations of analog-to-digital conversion errors.

In a flash ADC, for example, a failure of one of the op-amp comparators can cause a missing-code error.

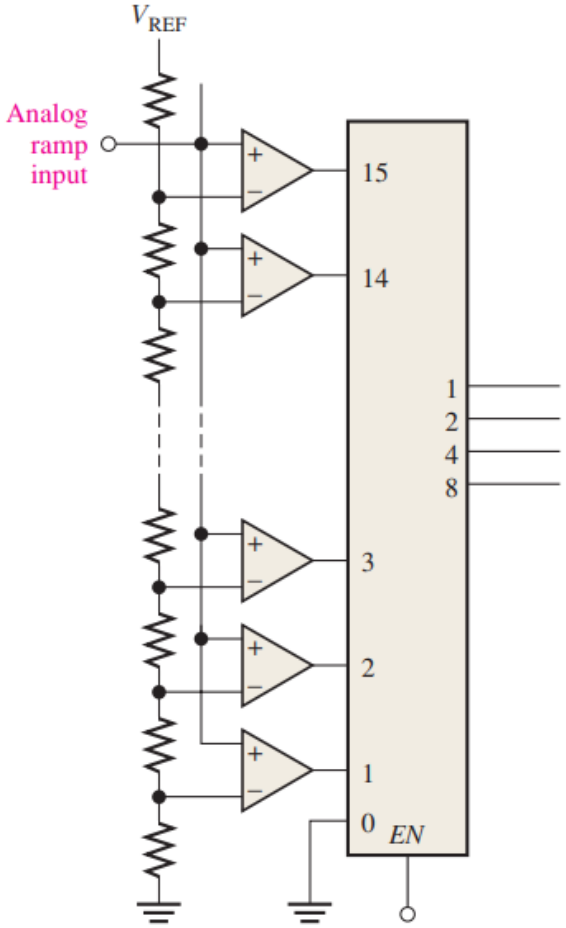
Incorrect Code

The stairstep output in Figure 12–24(b) indicates that several of the binary code words coming out of the ADC are incorrect.

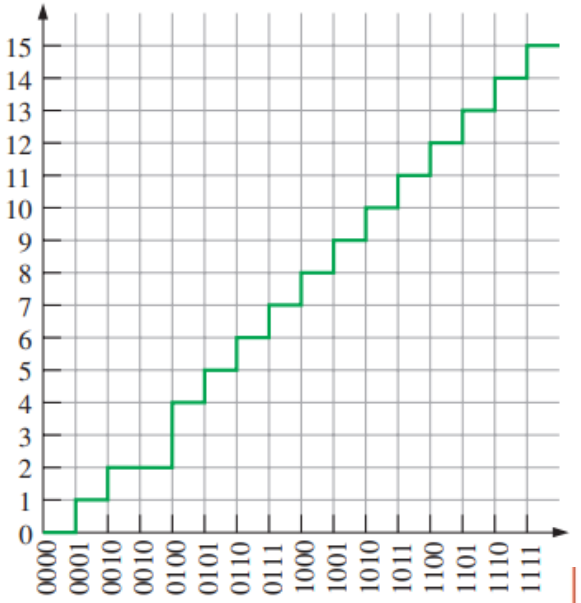
Offset

EXAMPLE 12–2

A 4-bit flash ADC is shown in Figure 12–25(a). It is tested with a setup like the one in Figure 12–23. The resulting reconstructed analog output is shown in Figure 12–25(b). Identify the problem and the most probable fault.



(a)



(b)

FIGURE 12–25

Solution

The binary code 0011 is missing from the ADC output, as indicated by the missing step. Most likely, the output of comparator 3 is stuck in its inactive state (LOW).

3. Methods of Digital-to-Analog Conversion

Digital-to-analog conversion is an important part of a digital processing system. Once the digital data has been processed, it is converted back to analog form.

Binary-Weighted-Input Digital-to-Analog Converter

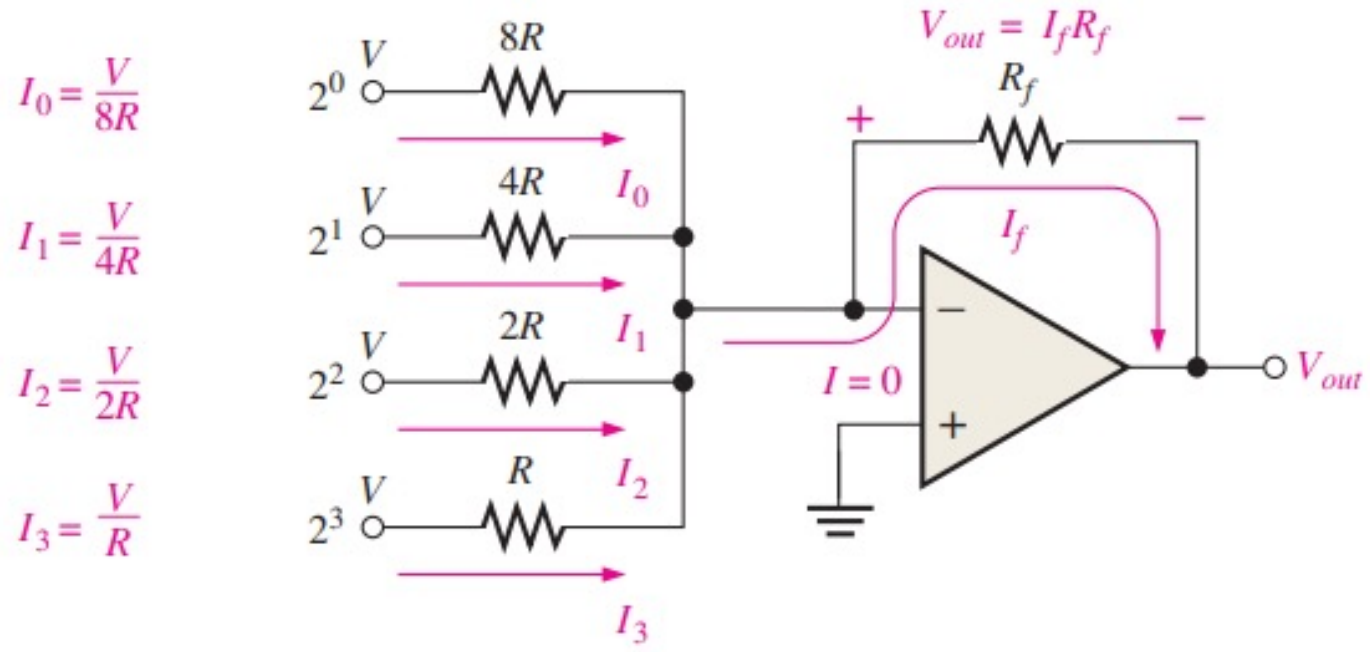
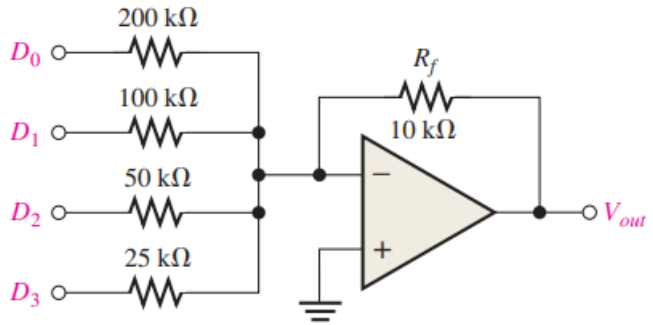


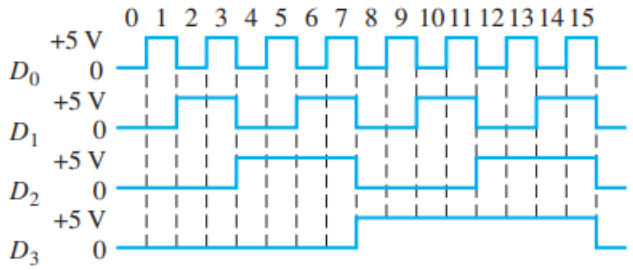
FIGURE 12–26 A 4-bit DAC with binary-weighted inputs.

EXAMPLE 12–3

Determine the output of the DAC in Figure 12–27(a) if the waveforms representing a sequence of 4-bit numbers in Figure 12–27(b) are applied to the inputs. Input D_0 is the least significant bit (LSB).



(a)



(b)

FIGURE 12–27

Solution

First, determine the current for each of the weighted inputs. Since the inverting (−) input of the op-amp is at 0 V (virtual ground) and a binary 1 corresponds to +5 V, the current through any of the input resistors is 5 V divided by the resistance value.

$$I_0 = \frac{5\text{ V}}{200\text{ k}\Omega} = 0.025\text{ mA}$$

$$I_1 = \frac{5\text{ V}}{100\text{ k}\Omega} = 0.05\text{ mA}$$

$$I_2 = \frac{5\text{ V}}{50\text{ k}\Omega} = 0.1\text{ mA}$$

$$I_3 = \frac{5\text{ V}}{25\text{ k}\Omega} = 0.2\text{ mA}$$

Almost no current goes into the inverting op-amp input because of its extremely high impedance. Therefore, assume that all of the current goes through the feedback resistor R_f. Since one end of R_f is at 0 V (virtual ground), the drop across R_f equals the output voltage, which is negative with respect to virtual ground.

$$V_{out(D0)} = (10\text{ k}\Omega)(-0.025\text{ mA}) = -0.25\text{ V}$$

$$V_{out(D1)} = (10\text{ k}\Omega)(-0.05\text{ mA}) = -0.5\text{ V}$$

$$V_{out(D2)} = (10\text{ k}\Omega)(-0.1\text{ mA}) = -1\text{ V}$$

$$V_{out(D3)} = (10\text{ k}\Omega)(-0.2\text{ mA}) = -2\text{ V}$$

From Figure 12–27(b), the first binary input code is 0000, which produces an output voltage of 0 V. The next input code is 0001, which produces an output voltage of −0.25 V. The next code is 0010, which produces an output voltage of −0.5 V. The next code is 0011, which produces an output voltage of −0.25 V + −0.5 V = −0.75 V. Each successive binary code increases the output voltage by −0.25 V, so for this particular straight binary sequence on the inputs, the output is a stairstep waveform going from 0 V to −3.75 V in −0.25 V steps. This is shown in Figure 12–28.

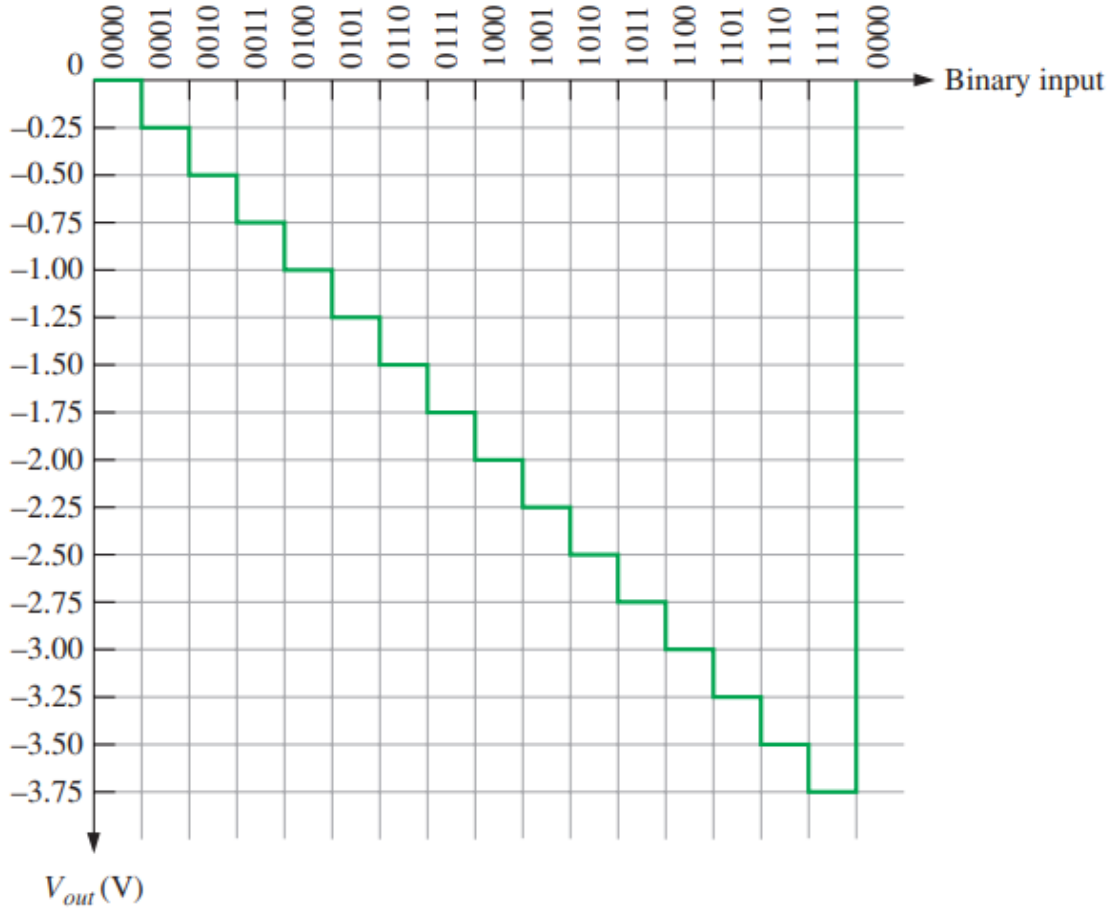


FIGURE 12–28 Output of the DAC in Figure 12–27.

The R/2R Ladder Digital-to-Analog Converter

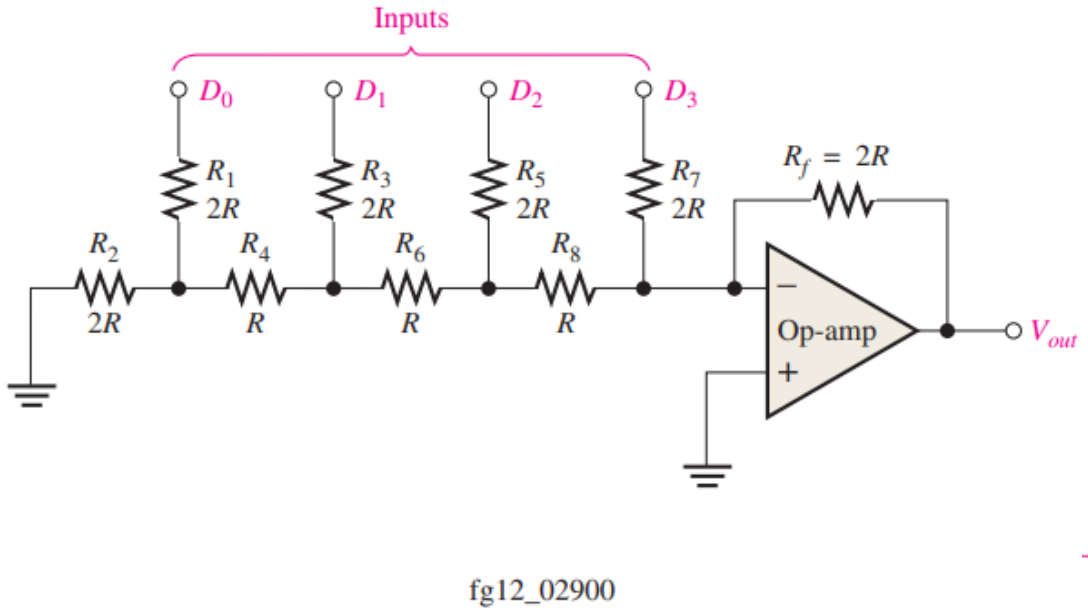
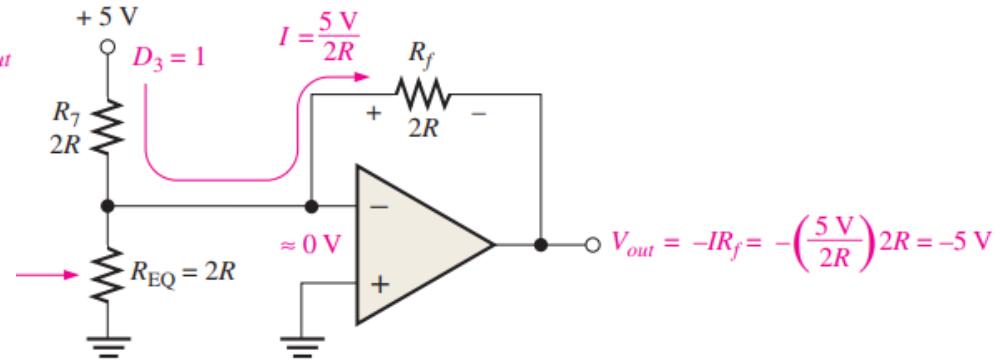
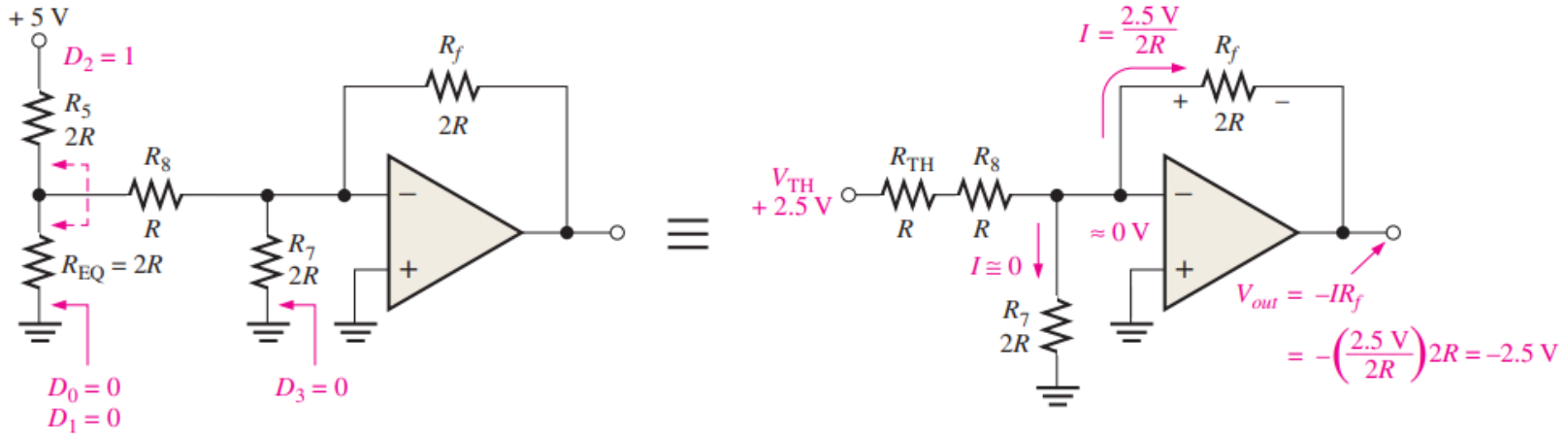


FIGURE 12–29 An *R/2R* ladder DAC.

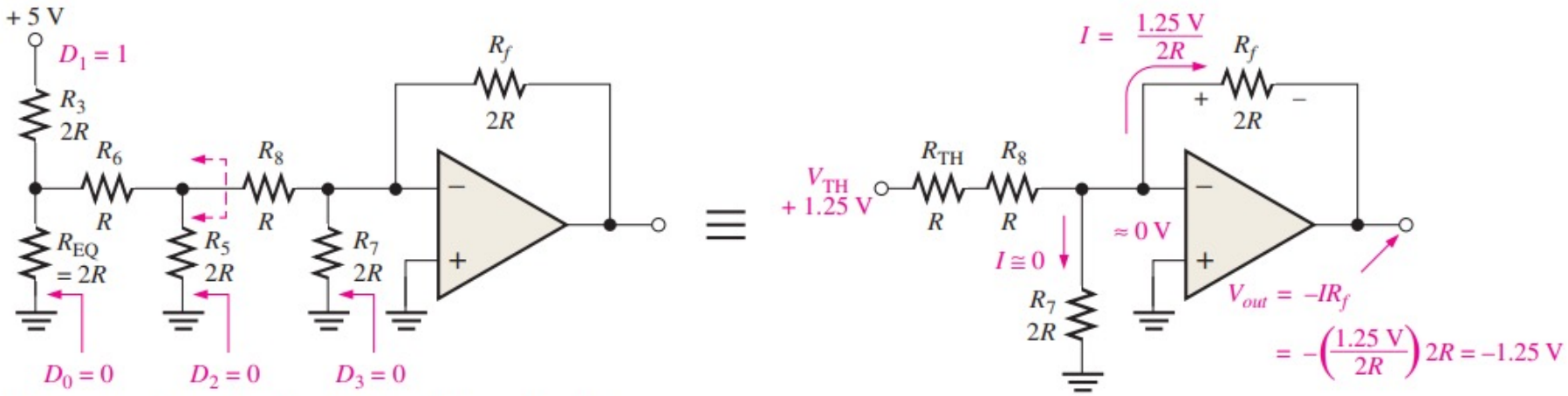


(a) Equivalent circuit for $D_3 = 1, D_2 = 0, D_1 = 0, D_0 = 0$

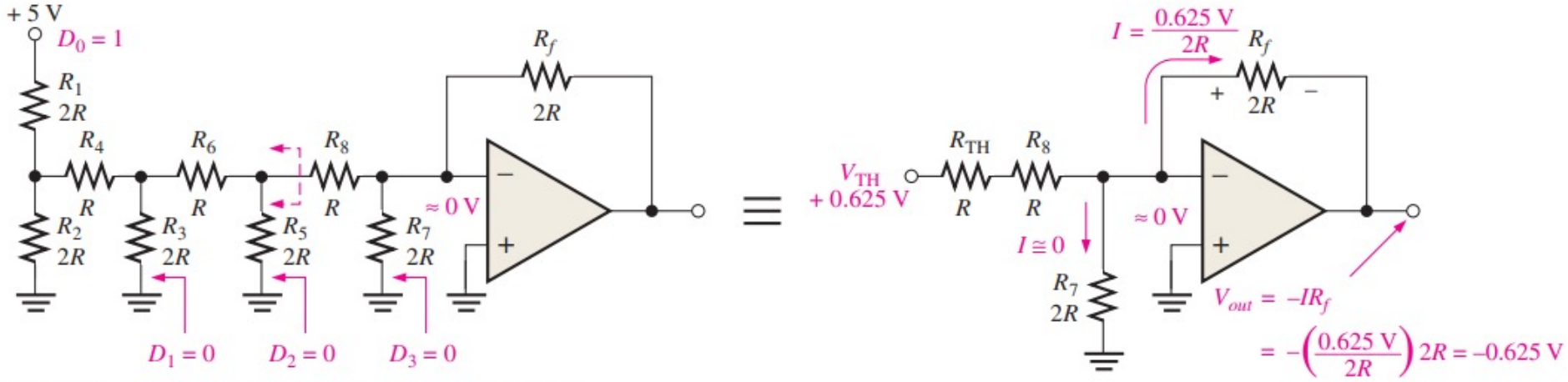


(b) Equivalent circuit for $D_3 = 0, D_2 = 1, D_1 = 0, D_0 = 0$

The R/2R Ladder Digital-to-Analog Converter



(c) Equivalent circuit for $D_3 = 0$, $D_2 = 0$, $D_1 = 1$, $D_0 = 0$



(d) Equivalent circuit for $D_3 = 0$, $D_2 = 0$, $D_1 = 0$, $D_0 = 1$

FIGURE 12–30 Analysis of the *R*/2*R* ladder DAC.

Performance Characteristics of Digital-to-Analog Converters

- ***Resolution***. The resolution of a DAC is the reciprocal of the number of discrete steps in the output. This, of course, is dependent on the number of input bits.
- ***Accuracy***. Accuracy is derived from a comparison of the actual output of a DAC with the expected output. It is expressed as a percentage of a full-scale, or maximum, output voltage.
- ***Linearity***. A linear error is a deviation from the ideal straight-line output of a DAC. A special case is an offset error, which is the amount of output voltage when the input bits are all zeros.
- ***Monotonicity***. A DAC is monotonic if it does not take any reverse steps when it is sequenced over its entire range of input bits.
- ***Settling time.*** Settling time is normally defined as the time it takes a DAC to settle within +-1/2 LSB of its final value when a change occurs in the input code.

EXAMPLE 12–4

Determine the resolution, expressed as a percentage, of the following:

(a) an 8-bit DAC

(b) a 12-bit DAC

Solution

(a) For the 8-bit converter,

$$\frac{1}{2^8 - 1} \times 100 = \frac{1}{255} \times 100 = \mathbf{0.392\%}$$

(b) For the 12-bit converter,

$$\frac{1}{2^{12} - 1} \times 100 = \frac{1}{4095} \times 100 = \mathbf{0.0244\%}$$

Related Problem

Calculate the resolution for a 16-bit DAC.

Digital-to-Analog Conversion Errors

Nonmonotonicity

Differential Nonlinearity

Low or High Gain

Offset Error

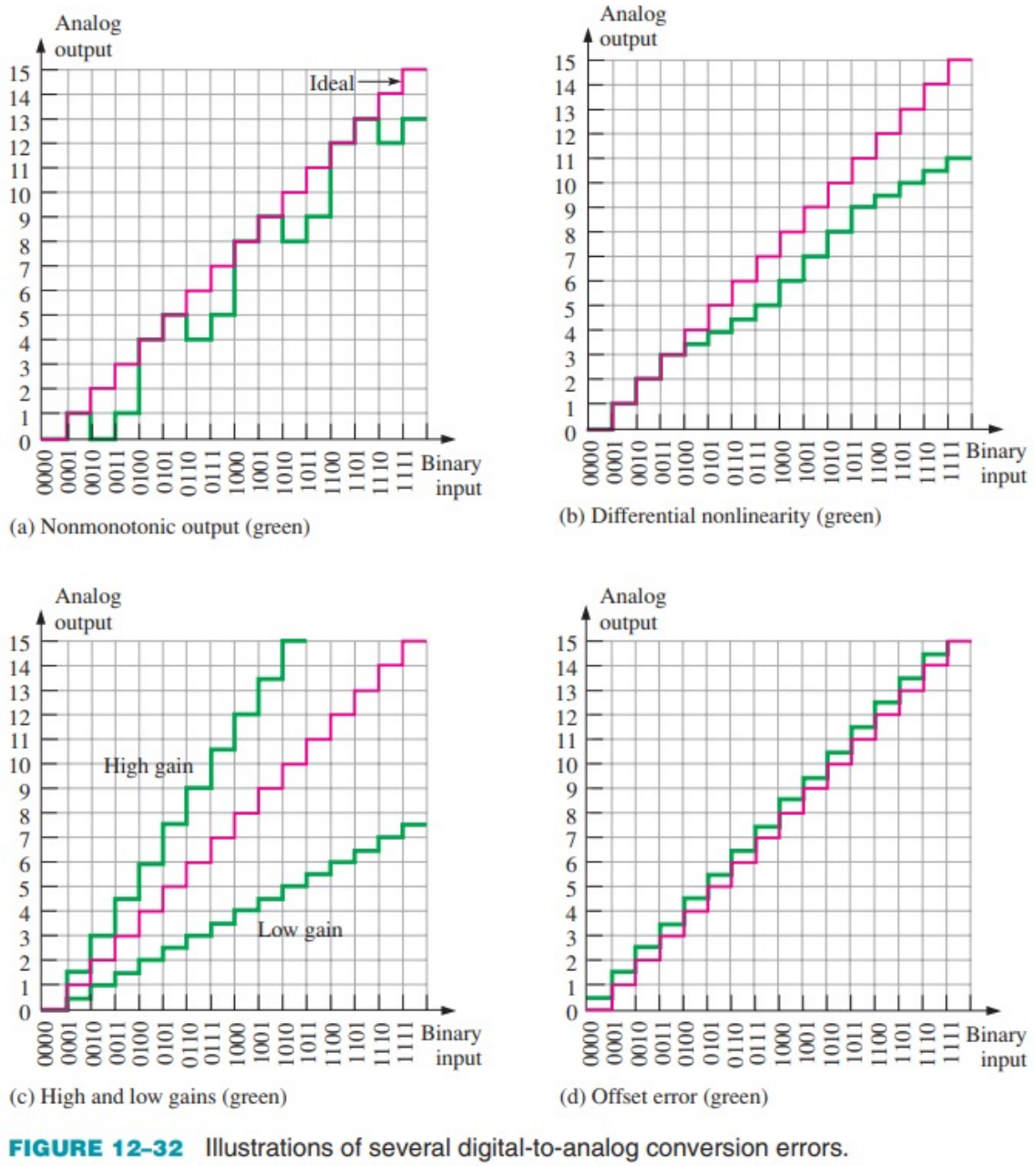


FIGURE 12–32 Illustrations of several digital-to-analog conversion errors.

The Reconstruction Filter

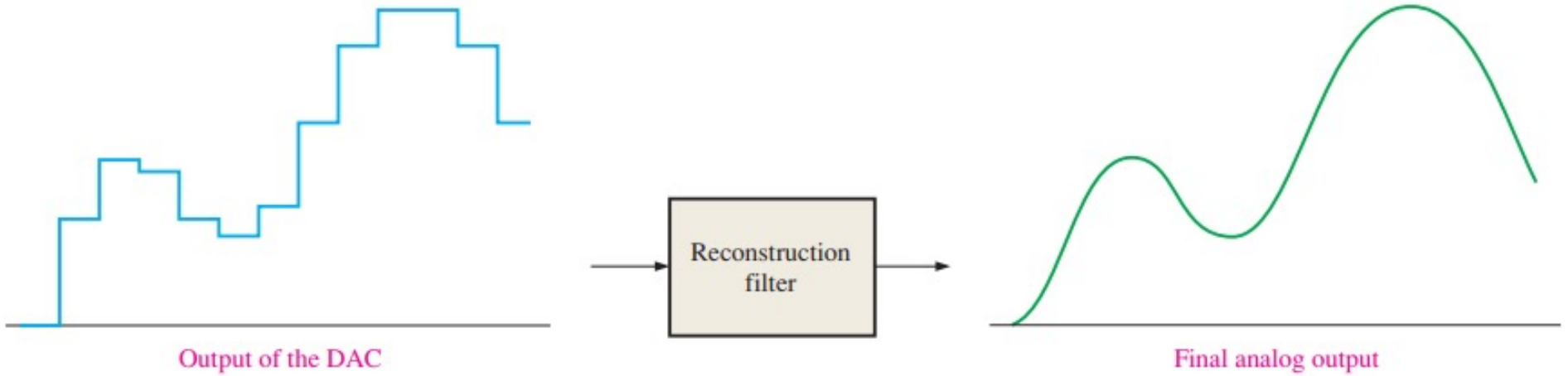


FIGURE 12–34 The reconstruction filter smooths the output of the DAC.

4. Digital Signal Processing

Digital signal processing converts signals that naturally occur in analog form to digital form and uses digital techniques to enhance and modify analog signal data for various applications.

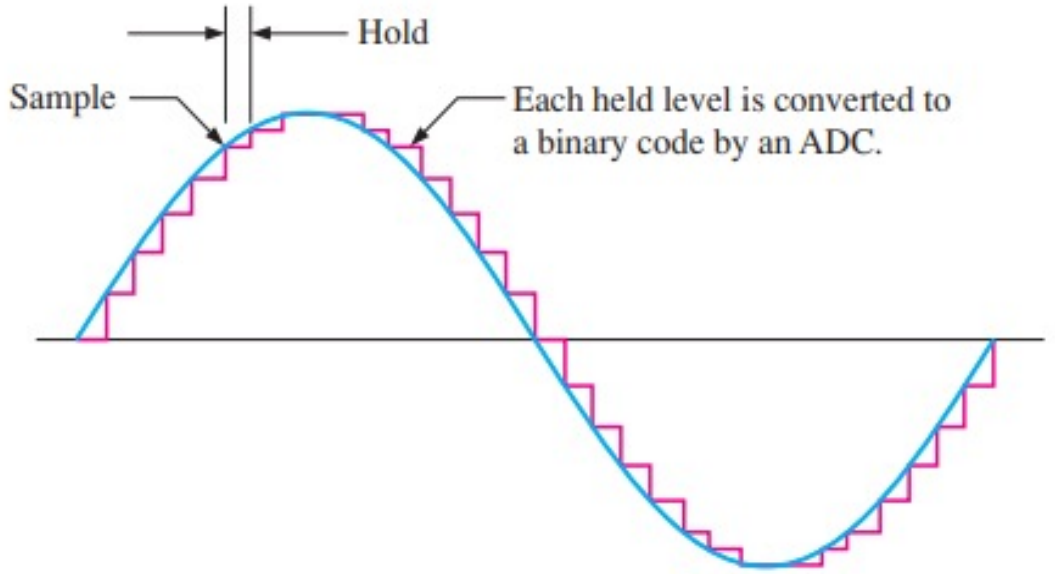


FIGURE 12–35 An original analog signal (sine wave) and its "stairstep" approximation.

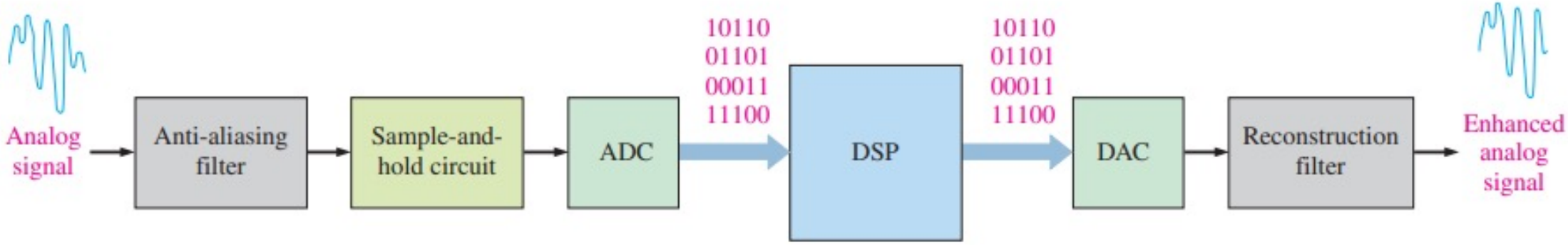


FIGURE 12–36 Basic block diagram of a typical digital signal processing system.

5. The Digital Signal Processor (DSP)

A digital signal processor (DSP) is a special type of microprocessor that processes data in real time

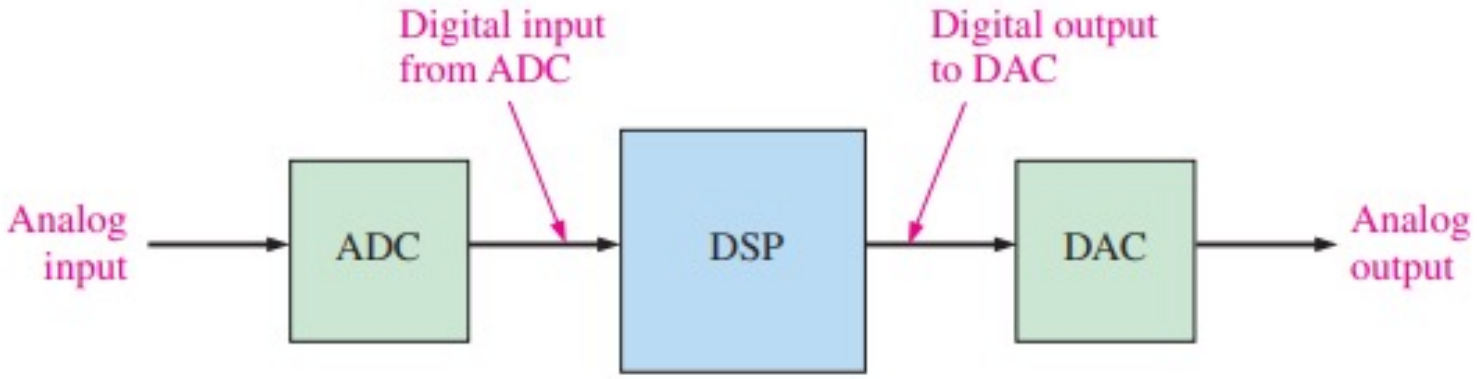


FIGURE 12–37 The DSP has a digital input and produces a digital output.

The DSP in a Cellular Telephone

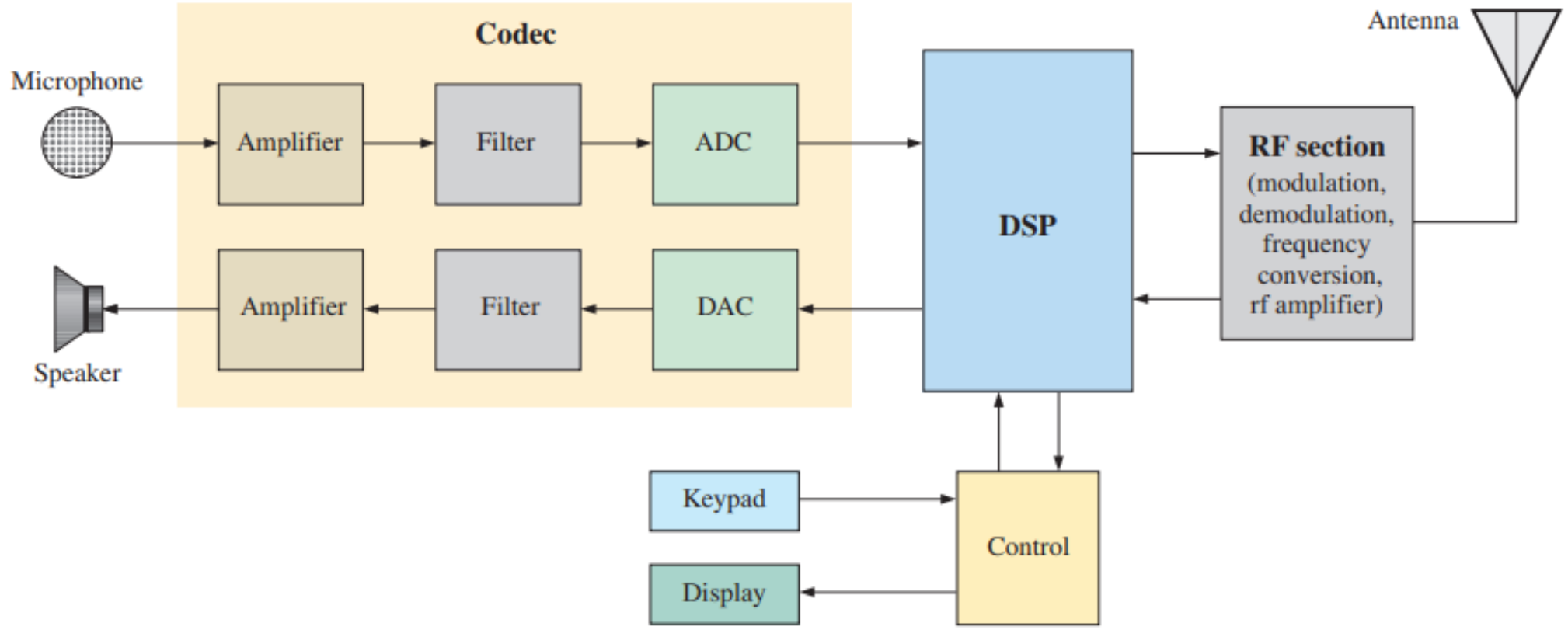


FIGURE 12–38 Simplified block diagram of a digital cellular phone.

Functions Performed by the DSP: Speech compression; Speech decompression; Protocol handling; Error detection and correction; Encryption

Basic DSP Architecture

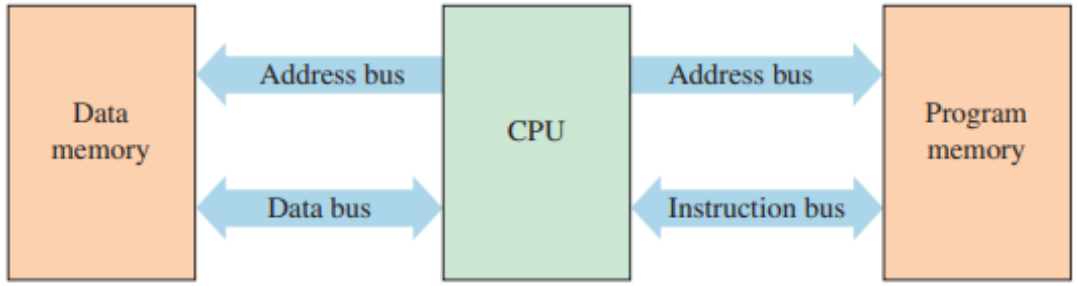


FIGURE 12–39 Many DSPs use the Harvard architecture (two memories).

TABLE 12–3

TMS320C6000 series DSP data processing performance.

DSP	Type	Application	Processing Speed	Multiply/ Accumulate Speed
C62xx	Fixed-point	General-purpose	1200–2400 MIPS	300–600 MMACS
C64xx	Fixed-point	Special-purpose	3200–4800 MIPS	1600–2400 MMACS
C67xx	Floating-point	General-purpose	600–1000 MFLOPS	200–333 MMACS

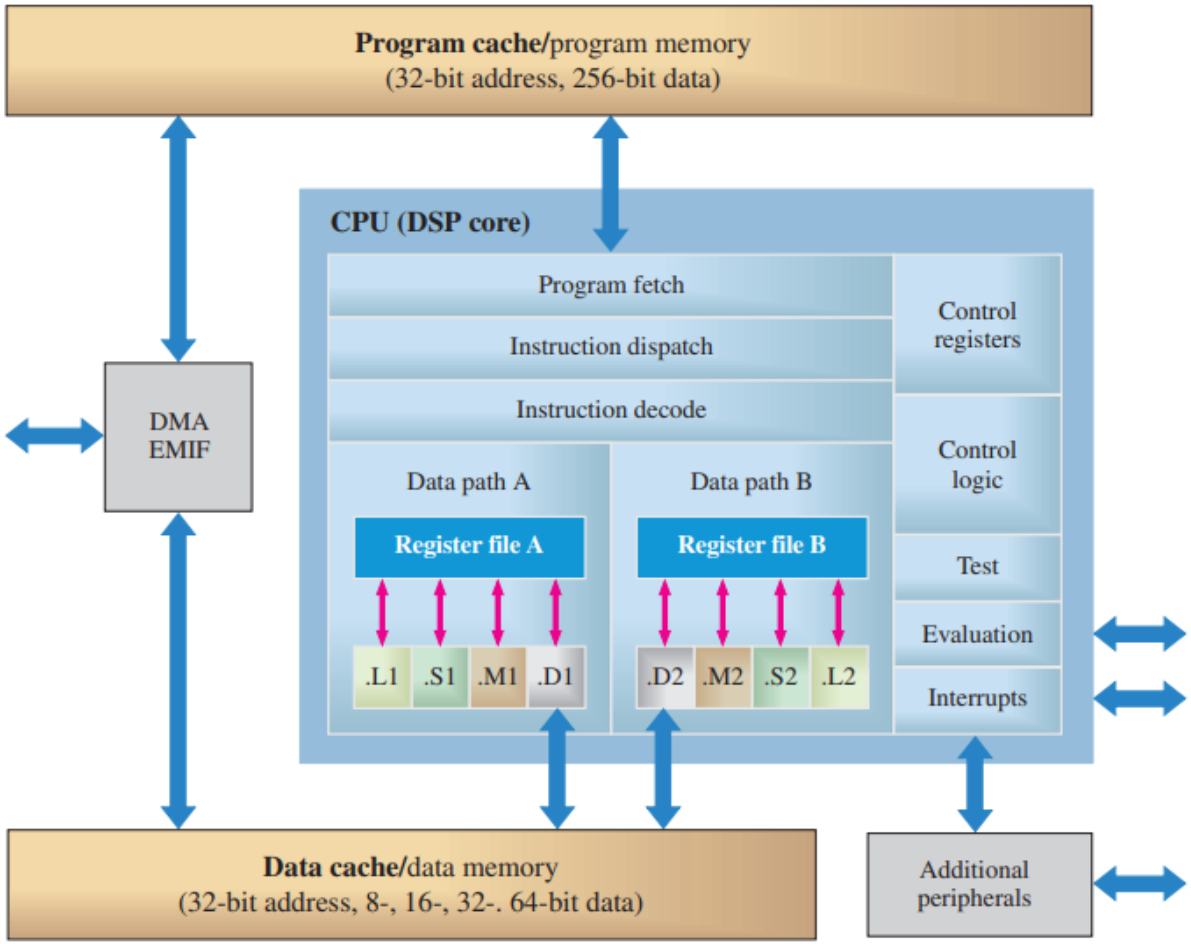
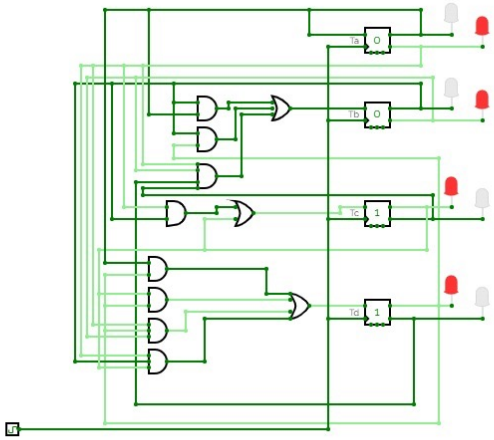


FIGURE 12–40 General block diagram of the TMS320C6000 series DSP.

THE END

Lecture 12: Signal Conversion and Processing



INSTRUCTOR: Dr. Vuong Quoc Bao