

Digital Signal Conditioning

INSTRUCTIONAL OBJECTIVES

In this chapter the basic principles of digital signal processing will be studied. This includes digital-to-analog converters (DAC) and analog-to-digital converters (ADC), as well as the characteristics of digital data. After reading this chapter and working through the examples and problems at the end of the chapter you will be able to:

- Develop Boolean equations for multivariable alarms.
- Design an application using a comparator with hysteresis.
- Calculate the expected output of a bipolar DAC for a given input.
- Explain how a successive approximation ADC operates.
- Describe how a sample-and-hold circuit operates.
- Explain the operation of a frequency-based ADC.
- Describe the consequence of sampling rate on data acquisition.

3.1 INTRODUCTION

Why *digital signal conditioning*? And what is digital signal conditioning, anyway?

The answer to the first question is found in the recognition that digital electronics and digital computers have taken a major role in nearly every aspect of life in our modern world. Of course digital electronics is at the heart of computers, but there are lots of direct applications of digital electronics in our world. Everyday things like automatic door openers in stores, motion sensors in security systems, and seat-belt warning systems are implemented with digital electronics. All these digital electronic systems require data to be presented to them in a digital format (i.e., the data have to be *digitally conditioned*).

When most people think of a computer, they visualize the common home and office *personal computer*, or PC. In fact, however, one of the most common applications of computers is for controlling something, and we seldom ever even see the computer. When we learn that our microwave oven has a microprocessor-based computer inside, it is there for the purpose of controlling the operation of the oven. So it is with the use of computers in automobiles, washing machines, airplanes, and a vast host of other examples.

Computers are digital electronic devices and so all the information they work with has to be digitally formatted. Therefore, if they are used to control a variable such as temperature, then the temperature has to be represented digitally. So that's why we need digital signal conditioning—to condition process-control signals to be in an appropriate digital format.

The second question is the topic of this chapter. Digital signal conditioning in process control means finding a way to represent analog process information in a digital format. We will review some common digital electronics principles and then study how analog signals are converted into a digital format.

You should realize that there is no greater accuracy in using digital techniques to represent data; in fact, accuracy is usually lost. But digital data are much more immune from spurious influences that would cause subsequent inaccuracy, such as noise, amplifier gain changes, power supply drifts, and so on.

Use of computers in control systems is particularly valuable for a number of other reasons, however:

1. A computer can control multivariable process-control systems.
2. Nonlinearities in sensor output can be linearized by the computer.
3. Complicated control equations can be solved quickly and modified as needed.
4. Networking of control computers allows a large process-control complex to operate in a fully integrated fashion.

3.2 REVIEW OF DIGITAL FUNDAMENTALS

A working understanding of the application of digital techniques to process control requires a foundation in basic digital electronics. The design and implementation of control logic systems and microcomputer control systems require a depth of understanding that can be obtained only by taking several courses devoted to the subject. In this text, we assume a sufficient background that the reader can appreciate the essential features of digital electronic design and its application to process control. A summary of basic digital electronics concepts is presented in Appendix 2.

3.2.1 Digital Information

The use of digital techniques in process control requires that process variable measurements and control information be encoded into a digital form. Digital signals themselves are simply two-state (binary) levels. These levels may be represented in many ways (e.g.,

two voltages, two currents, two frequencies, or two phases). We speak, then, of the digital information as a high state (H, or **1**) or a low state (L, or **0**) on a wire that carries the digital signal.

Digital Words Given the simple binary information that is carried by a single digital signal, it is clear that multiple signals must be used to describe analog information. Generally, this is done by using an assemblage of digital levels to construct a binary number, often called a *word*. The individual digital levels are referred to as *bits* of the word. Thus, for example, a 6-bit word consists of six independent digital levels, such as **101011**₂, which can be thought of as a six-digit base 2 number. An important consideration, then, is how the analog information is encoded into this digital word.

Decimal Whole Numbers One of the most common schemes for encoding analog data into a digital word is to use the straight counting of decimal (or base 10) and binary number representations. The principles of this process are reviewed in Appendix 2, together with octal and hexadecimal representations.

EXAMPLE 3.1 Find the base 10 equivalent of the binary whole number **00100111**₂.

Solution

As in the base 10 system, zeros preceding the first significant digit do not contribute. Thus, the binary number is actually **100111**₂, and $n = 5$. To find the decimal equivalent, we use Appendix 2 and compute as follows:

$$\begin{aligned} N_{10} &= a_5 2^5 + a_4 2^4 + \cdots + a_1 2^1 + a_0 2^0 \\ N_{10} &= (1)2^5 + (0)2^4 + (0)2^3 + (1)2^2 + (1)2^1 + (1)2^0 \\ N_{10} &= 32 + 4 + 2 + 1 \\ N_{10} &= 39 \end{aligned} \quad (\text{A.2})$$

EXAMPLE 3.2 Find the binary equivalent of the base 10 number 47.

Solution

Starting the successive division, we get

$$\frac{47}{2} = 23 \text{ with a remainder of } \frac{1}{2} \text{ so that } a_0 = 1$$

then

$$\frac{23}{2} = 11 \text{ with a remainder of } \frac{1}{2} \text{ so that } a_1 = 1$$

then

$$\frac{11}{2} = 5 + \frac{1}{2} \quad \therefore a_2 = 1$$

$$\frac{5}{2} = 2 + \frac{1}{2} \quad \therefore a_3 = 1$$

$$\frac{2}{2} = 1 + 0 \quad \therefore a_4 = 0$$

$$\frac{1}{2} = 0 + \frac{1}{2} \quad \therefore a_5 = 1$$

We find that base 10 number 47 becomes binary number **101111**₂.

The representation of negative numbers in binary format takes on several forms, as discussed in Appendix 2.

Octal and Hex Numbers It is cumbersome for humans to work with digital words expressed as numbers in the binary representation. For this reason, it has become common to use either the octal (base 8) or hexadecimal (base 16, called hex) representations, which are reviewed in Appendix 2. Octal numbers are conveniently formed from groupings of three binary digits; that is, **000**₂ is 0₈ and **111**₂ is 7₈. Thus, a binary number like **101011**₂ is equivalent to 53₈. Hex numbers are formed easily from groupings of four binary digits; that is, **0000**₂ is 0H and **1111**₂ is FH. The letter H (or h) is used to designate a hex number instead of a subscript 16. Also recall that the hex counting sequence is 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F to cover the possible states. Because microcomputers must frequently use either 4-bit, 8-bit, or 16-bit words, the hex notation is commonly used with these machines. In hex, a binary number like **10110110**₂ would be written B6H.

3.2.2 Fractional Binary Numbers

Although not as commonly used, it is possible to define a fractional binary number in the same manner as whole numbers, using only the **1** and **0** of this counting system. Such numbers, just as in the decimal framework, represent divisions of the counting system to values less than unity. A correlation can be made to decimal numbers in a similar fashion to Equation (A.2.1), as

$$N_{10} = b_1 2^{-1} + b_2 2^{-2} + \cdots + b_m 2^{-m} \quad (3.1)$$

where

$$\begin{aligned} N_{10} &= \text{base 10 number less than 1} \\ b_1 b_2 \dots b_{m-1} b_m &= \text{base 2 number less than 1} \\ m &= \text{number of digits in base 2 number} \end{aligned}$$

EXAMPLE 3.3 Find the base 10 equivalent of the binary number 0.11010_2 .

Solution

This can be found most easily by using

$$N_{10} = b_1 2^{-1} + b_2 2^{-2} + \cdots + b_m 2^{-m} \quad (3.1)$$

with

$$m = 5$$

$$N_{10} = (1)2^{-1} + (1)2^{-2} + (0)2^{-3} + (1)2^{-4} + (0)2^{-5}$$

$$N_{10} = \frac{1}{2} + \frac{1}{4} + \frac{1}{16}$$

$$N_{10} = 0.8125_{10}$$

Converting a base 10 number that is less than 1 to a binary equivalent requires repeated multiplication by 2. The result of each multiplication is a fractional part and either a 0 or 1 whole-number part, which determines whether that digit is a 0 or a 1. The first multiplication gives the most significant bit, b_1 , and the last gives either a 0 or a 1 for the least significant bit, b_m .

EXAMPLE 3.4 Find the binary, octal, and hex equivalents of 0.3125_{10} .

Solution

Using successive multiplication, we find

$$2(0.3125) = 0.6250 \quad \text{so } b_1 = 0$$

$$2(0.625) = 1.250 \quad \text{so } b_2 = 1$$

$$2(0.25) = 0.5 \quad \text{so } b_3 = 0$$

$$2(0.5) = 1.0 \quad \text{so } b_4 = 1$$

Thus, we find that 0.3125_{10} is equivalent to 0.0101_2 . It can be represented as 0.010100_2 , because trailing zeros are not significant in a number less than 1, and thus as 0.24_{octal} , because $010_2 = 2_8$ and $100_2 = 4_8$. Similarly, this is 0.50_{H} .

3.2.3 Boolean Algebra

In process control, as well as in many other technical disciplines, action is taken on the basis of an evaluation of observations made in the environment. In driving an automobile, for example, we are constantly observing such external factors as traffic, lights, speed limits, pedestrians, street conditions, low-flying aircraft, and such internal factors as how fast we wish to go, where we are going, and many others. We evaluate these factors and take actions predicated on the evaluations. We may see that a light is green, streets are dry, speed

is low, there are no pedestrians or aircraft, we are late, and thus conclude that an action of pressing on the accelerator is required. Then we may observe a parked police radar unit with all other factors the same, negate the aforementioned conclusion, and apply the brake. Many of these parameters can be represented by a *true* or *not true* observation; in fact, with enough definition, all the observations could be reduced to simple true or false conditions. When we learn to drive, we are actually setting up internal responses to a set of such true/false observations in the environment.

In the industrial world, an analogous condition exists relative to the external and internal influences on a manufacturing process, and when we control a process, we are in effect teaching a control system response to a set of true/false observations. This teaching may consist of designing electronic circuits that can logically evaluate the set of true/false conditions and initiate some appropriate action. To design such an electronic system, we must first be able to mathematically express the inputs, the logical evaluation, and the corresponding outputs. *Boolean algebra* is a mathematical procedure that allows the combinations of true/false conditions in various logical operations by equations so that conclusions can be drawn. For the purposes of this text, we do not require expertise in Boolean technique, but only an operational familiarity with it that can be applied to a process-control environment.

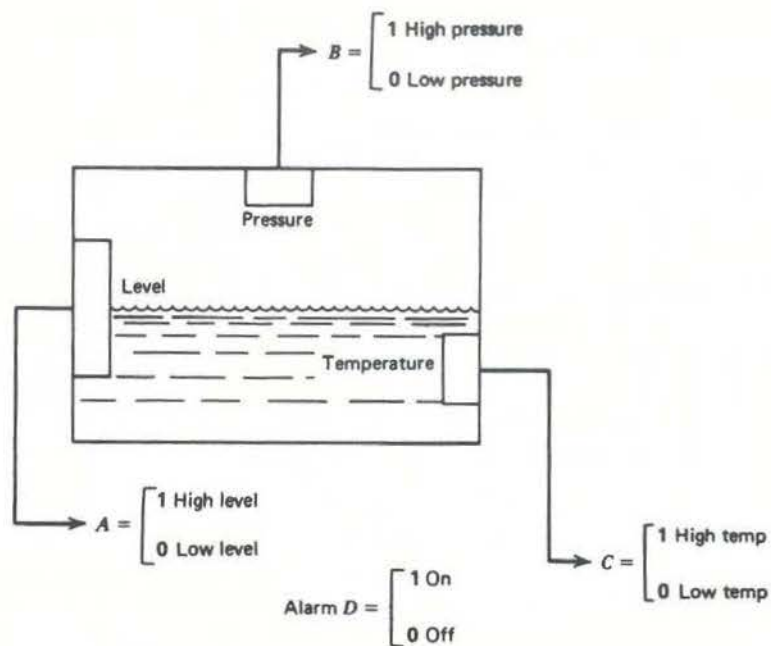
Before a particular problem in industry can be addressed using digital electronics, it must be analyzed in terms that are amenable to the binary nature of digital techniques. Generally, this is accomplished by stating the problem in the form of a set of true/false-type conditions that must be applied to derive some desired result. These sets of conditions are then stated in the form of one or more Boolean equations. We will see in Section 3.2.4 that a Boolean equation is in a form that is readily implemented with existing digital circuits. The mathematical approach of Boolean algebra allows us to write an analytical expression to represent these stipulations. The fundamentals of Boolean algebra are summarized in Appendix 2.

Let us consider a simple example of how a Boolean equation may result from a practical problem. Consider a mixing tank for which there are three variables of interest: liquid level, pressure, and temperature. The problem is that we must signal an alarm when certain combinations of conditions occur among these variables. Referring to Figure 3.1, we denote level by A , pressure by B , and temperature by C , and assume that setpoint values have been assigned for each variable so that the Boolean variables are either **1** or **0** as the physical quantities are above or below the setpoint values. The alarm will be triggered when the Boolean variable D goes to the logic true state. The alarm conditions are

1. Low level with high pressure
2. High level with high temperature
3. High level with low temperature and high pressure

We now define a Boolean expression with AND operations that will give a $D = 1$ for each condition:

1. $D = \bar{A} \cdot B$ will give $D = 1$ for condition 1.
2. $D = A \cdot C$ will give $D = 1$ for condition 2.
3. $D = A \cdot \bar{C} \cdot B$ will give $D = 1$ for condition 3.

**FIGURE 3.1**

System for illustrating Boolean applications to control.

The final logic equation results from combining all three conditions so that if any is true, the alarm will sound ($D = 1$). This is accomplished with the OR operation

$$D = \bar{A} \cdot B + A \cdot C + A \cdot \bar{C} \cdot B \quad (3.2)$$

This equation would now form the starting point for a design of electronic digital circuitry that would perform the indicated operations.

3.2.4 Digital Electronics

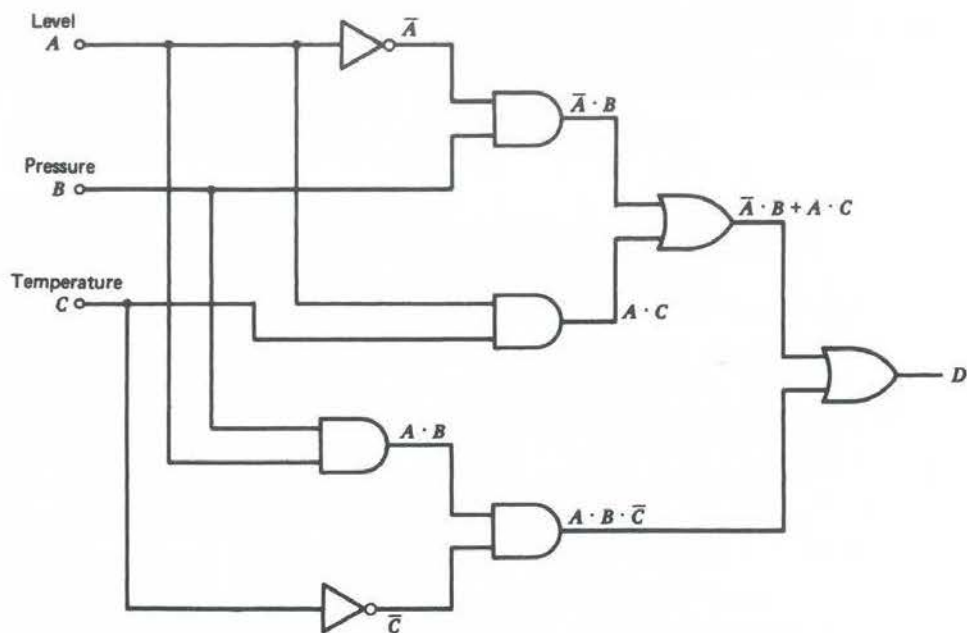
The electronic building blocks of digital electronics are designed to operate on the binary levels present on digital signal lines. These building blocks are based on families of types of electronic circuits, as discussed in Appendix 2, that have their specific stipulations of power supplies and voltage levels of the **1** and **0** states. The basic structure involves the use of AND/OR logic and NAND/NOR logic to implement Boolean equations.

EXAMPLE 3.5 Develop a digital circuit using AND/OR gates that implements Equation (3.2).

Solution

The problem posed in Section 3.2.3 (with Figure 3.1) has a Boolean equation solution of

$$D = \bar{A} \cdot B + A \cdot C + A \cdot \bar{C} \cdot B \quad (3.2)$$

**FIGURE 3.2**

Solution for Example 3.5.

The implementation of this equation using AND/OR gates is shown in Figure 3.2. The AND, OR, and inverter are used in a straightforward implementation of the equation. It should be noted that Equation (3.2) can be greatly reduced by someone skilled in the art of digital logic to: $D = A \cdot C + B$. The reader can show that this can be implemented by one AND gate and one OR gate!

EXAMPLE 3.6 Repeat Example 3.5 using NAND/NOR gates.

Solution

One way to implement the equation in NAND/NOR would be to provide inverters after every gate, in effect, to convert the devices back to AND/OR gates. In this case, the circuit developed would look like Figure 3.2 but with an inverter after every gate. A second approach is to use the Boolean theorems to reformulate the equation for better implementation using NAND/NOR logic. For example, if we are to get the desired equation for D as output from a NAND gate, the inputs must have been

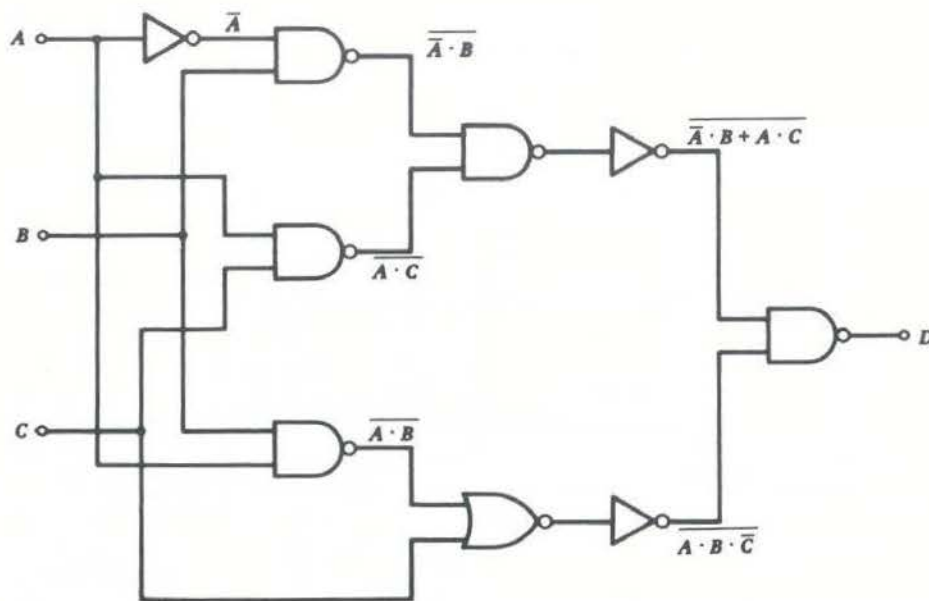
$$\overline{A \cdot B + A \cdot C}$$

and

$$\overline{A \cdot B \cdot \bar{C}}$$

because NAND between these produces

$$(\overline{A \cdot B + A \cdot C}) \cdot (\overline{A \cdot B \cdot \bar{C}})$$

**FIGURE 3.3**

Solution for Example 3.6.

which, by DeMorgan's theorem, becomes

$$\bar{A} \cdot B + A \cdot C + A \cdot B \cdot \bar{C}$$

that is, the desired output. Working backward from this result allows the circuit to be realized, as shown in Figure 3.3. However, using the simplified version given in Example 3.5, we can use DeMorgan's theorem to write

$$D = A \cdot C + B = \overline{(\bar{A} \cdot \bar{C}) \cdot \bar{B}}$$

The reader can now show that this can be provided by two NAND gates and one inverter.

3.2.5 Programmable Logic Controllers

The move toward digital logic techniques and computers in industrial control paralleled the development of special controllers called *programmable logic controllers* (PLCs), or simply *programmable controllers* (PCs). These devices are particularly suited to the solution control problems associated with Boolean equations and binary logic problems in general. They are a computer-based outgrowth of relay sequence controllers. Detailed treatment of this type of control system is given in Chapter 8.

3.2.6 Computer Interface

Figure 3.4 shows a simple model of a computer system. The processor is connected to external equipment via three parallel sets of digital lines. The *data lines* carry data to and from the processor. The *address lines* allow the computer to select external locations for input

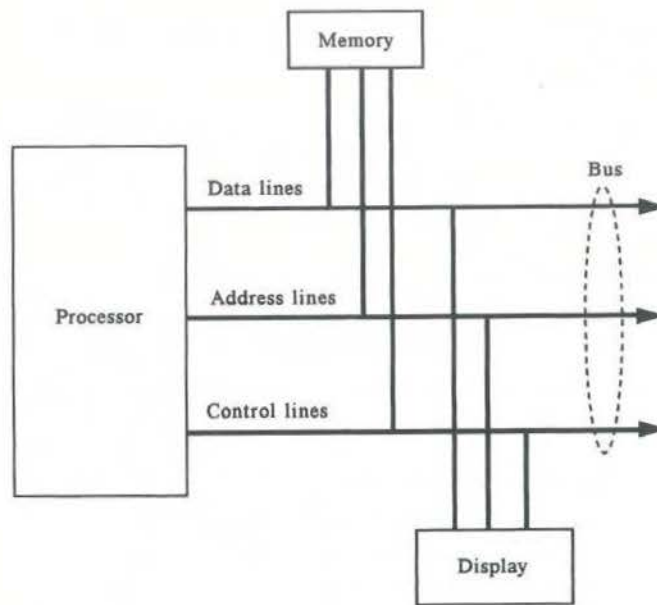


FIGURE 3.4
Generic model of a computer bus system.

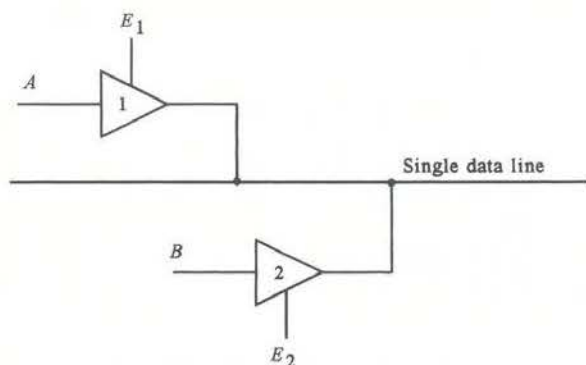
and output. The *control lines* carry information to and from the computer related to operations, such as reading, writing, interrupts, and so on. This collection of lines is called the *bus* of the computer.

The term *interface* refers to the hardware connections and software operations necessary to input and output data using connections to the bus. All of the equipment connected to the computer must share the bus lines.

It is an important consideration for interface hardware that a bus line not be compromised by some external connection. This means that the external equipment must not hold a bus line in a logic state when that equipment is not using the bus. If a data line is held at **0** by some equipment even when it is not performing data transfer, then no other equipment could raise that line to the **1** state during its data transfer operations. This problem is prevented by the use of tri-state buffers.

Tri-State Buffers Isolation of a bus line is accomplished by making all connections via a special digital device called a *tri-state buffer*. This device acts like a simple switch. When the switch is closed, the logic level on its input is impressed upon the output. When open, the output is placed in a high-impedance state—that is, an open circuit.

Figure 3.5 shows how two digital signals can both be connected to a single data line through tri-state buffers. Normally, both tri-states are disabled—that is, in the high-impedance state. When the computer needs to input signal *A*, an enable signal, E_1 , is sent to tri-state 1 so that the state of *A* is placed on the data line. After the computer reads the line, tri-state 1 is disabled again. Similarly, when the computer needs the state of signal *B*, an enable, E_2 , is sent to tri-state 2 to place *B* on the line.

**FIGURE 3.5**

Tri-state buffers allow multiple signals to share a single digital line in the bus.

3.3 CONVERTERS

The most important digital tool for the process-control technologist is one that translates digital information to analog and vice versa. Most measurements of process variables are performed by devices that translate information about the variable to an analog electrical signal. To interface this signal with a computer or digital logic circuit, it is necessary first to perform an analog-to-digital (A/D) conversion. The specifics of this conversion must be well known so that a unique, known relationship exists between the analog and digital signals. Often, the reverse situation occurs, where a digital signal is required to drive an analog device. In this case, a digital-to-analog (D/A) converter is required.

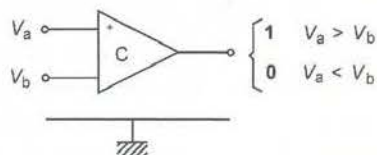
3.3.1 Comparators

The most elementary form of communication between the analog and digital is a device (usually an IC) called a *comparator*. This device, which is shown schematically in Figure 3.6, simply compares two analog voltages on its input terminals. Depending on which voltage is larger, the output will be a **1** (high) or a **0** (low) digital signal. The comparator is extensively used for alarm signals to computers or digital processing systems. This element is also an integral part of the analog-to-digital and digital-to-analog converter, to be discussed in Section 3.3.2.

One of the voltages on the comparator inputs, V_a or V_b in Figure 3.6, will be the variable input, and the other a fixed value called a trip, trigger, or reference voltage. The reference

FIGURE 3.6

A basic comparator compares voltages and produces a digital output.



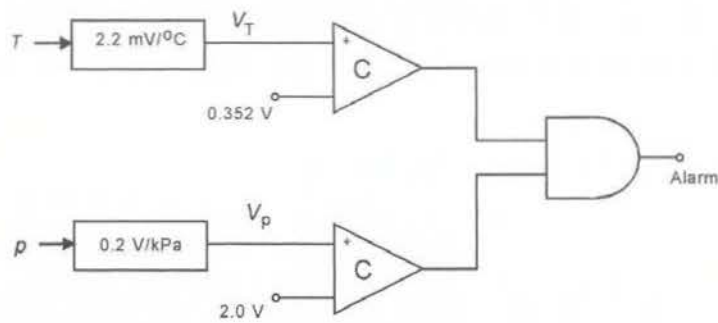
**FIGURE 3.7**

Diagram of a solution to Example 3.7.

value is computed from the specifications of the problem and then applied to the appropriate comparator input terminal, as illustrated in Example 3.7. The reference voltage may be provided from a divider using available power supplies.

EXAMPLE 3.7

A process-control system specifies that temperature should never exceed 160°C if the pressure also exceeds 10 kPa. Design an alarm system to detect this condition, using temperature and pressure transducers with transfer functions of $2.2 \text{ mV}/^{\circ}\text{C}$ and 0.2 V/kPa , respectively.

Solution

The alarm conditions will be a temperature signal of $(2.2 \text{ mV}/^{\circ}\text{C})(160^{\circ}\text{C}) = 0.352 \text{ V}$ coincident with a pressure signal of $(0.2 \text{ V/kPa})(10 \text{ kPa}) = 2 \text{ V}$. The circuit in Figure 3.7 shows how this alarm can be implemented with comparators and one AND gate. The reference voltages could be provided from dividers.

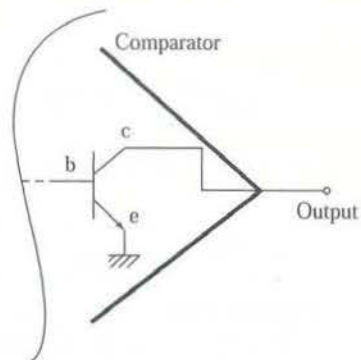
Open-Collector Comparators Some comparator models have a special method of providing the digital output signal. Figure 3.8a shows that the output terminal of the comparator is connected internally to the collector of a transistor in the comparator and nowhere else! This is called an *open-collector output* because it is just that. Of course, even if there is base-emitter current in the transistor, no voltage will show up on the collector until it is connected to a supply through some collector resistor. In fact, this is exactly what is done in an application. Figure 3.8b shows that an external resistor is connected from the output to an appropriate power supply. This is called a collector *pull-up* resistor. Now the output terminal will show either a **0** (0 V) if the internal transistor is ON or **1** (V_s) if the internal transistor is OFF.

There are a number of advantages to using the open-collector output:

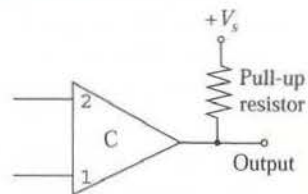
1. It is possible to use a different power source for the output. For example, suppose you want to activate a +12-V relay with the output of a comparator that operates on +5 V. By using an open-collector model, you can connect the pull-up resistor to a +12-V supply and power the relay directly from the output.

FIGURE 3.8

Many comparators use an open-collector output.



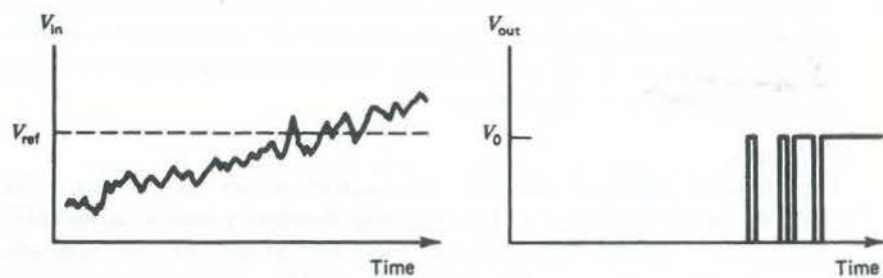
(a)



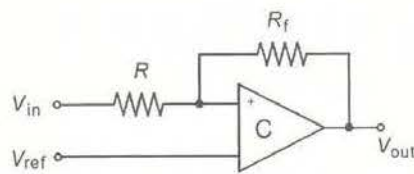
(b)

2. It is possible to OR together several comparators' outputs by connecting all open-collector outputs together and then using a common pull-up resistor. If any one of the comparator's output transistors is turned ON, the common output will go low.

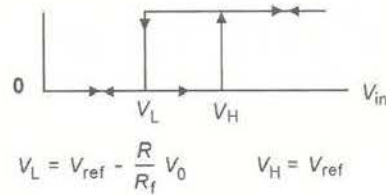
Hysteresis Comparator When using comparators, there is often a problem if the signal voltage has noise or approaches the reference value too slowly. The comparator output may "jiggle" back and forth between high and low as the reference level is reached. This effect is shown in Figure 3.9. Such fluctuation of output may cause problems with the equipment designed to interpret the comparator output signal.

**FIGURE 3.9**

A comparator output will "jiggle" when a noisy signal passes through the reference voltage level.



a) Hysteresis comparator circuit



b) Hysteresis comparator input-output relationship and equations

FIGURE 3.10

A generic DAC diagram, showing typical input and output signals.

This problem can often be solved by providing a *deadband* or *hysteresis* window to the reference level around which output changes occur. Once the comparator has been triggered high, the reference level is automatically reduced so that the signal must fall to some value below the old reference before the comparator goes to the low state.

There are many ways this hysteresis can be provided, but Figure 3.10a shows one common technique. Feedback resistor R_f is provided between the output and one of the inputs of the comparator, and that input is separated from the signal by another resistor, R . Under the condition that $R_f \gg R$, the response of the comparator is shown in Figure 3.10b.

The condition for which the output will go high (V_0) is defined by the condition

$$V_{in} \geq V_{ref} \quad (3.3)$$

Once having been driven high, the condition for the output to drop back to the low (0 V) state is given by the relation

$$V_{in} \leq V_{ref} - (R/R_f)V_0 \quad (3.4)$$

The deadband or hysteresis is given by $(R/R_f)V_0$, and is thus selectable by choice of the resistors, as long as this relation is satisfied. The response of this comparator is shown by the graph in Figure 3.10b. The arrows indicate increasing or decreasing input voltage.

EXAMPLE 3.8

A sensor converts the liquid level in a tank to voltage according to the transfer function (20 mV/cm). A comparator is supposed to go high (5 V) whenever the level becomes 50 cm. Splashing causes the level to fluctuate by ± 3 cm. Develop a hysteresis comparator to protect against the effects of splashing.

Solution

The nominal reference for the comparator occurs at 50 cm, which is $V_{ref} = (20 \text{ mV/cm})(50 \text{ cm}) = 1 \text{ V}$. The splashing, however, causes a “noise” of $(20 \text{ mV/cm}) \cdot (\pm 3 \text{ cm}) = \pm 60 \text{ mV}$. This is a total range of 120 mV. We need a deadband of at least 120 mV, but let us make it 150 mV for security. Thus, we have

$$\begin{aligned} (R/R_f)(5 \text{ V}) &= 150 \text{ mV} \\ (R/R_f) &= 0.03 \end{aligned}$$

If we make $R_f = 100 \text{ k}\Omega$, then $R = 3 \text{ k}\Omega$. Thus, use of these resistors, as shown in Figure 3.10, with a reference of 1 V will meet the requirement.

3.3.2 Digital-to-Analog Converters (DACs)

A DAC accepts digital information and transforms it into an analog voltage. The digital information is in the form of a binary number with some fixed number of digits. Especially when used in connection with a computer, this binary number is called a *binary word* or *computer word*. The digits are called *bits* of the word. Thus, an 8-bit word would be a binary number having eight digits, such as 10110110_2 . A unipolar DAC converts a digital word into an analog voltage by scaling the analog output to be zero when all bits are zero and some maximum value when all bits are one. This can be mathematically represented by treating the binary number that the word represents as a *fractional* number. In this context, the output of the DAC can be defined using Equation (3.1) as a *scaling* of some reference voltage:

$$V_{\text{out}} = V_R[b_1 2^{-1} + b_2 2^{-2} + \dots + b_n 2^{-n}] \quad (3.5)$$

where V_{out} = analog voltage output
 V_R = reference voltage
 $b_1 b_2 \dots b_n$ = n -bit binary word

The minimum V_{out} is zero, and the maximum is determined by the size of the binary word because, with all bits set to one, the decimal equivalent approaches V_R as the number of bits increases. Thus, a 4-bit word has a maximum of

$$V_{\text{max}} = V_R[2^{-1} + 2^{-2} + 2^{-3} + 2^{-4}] = 0.9375 V_R$$

and an 8-bit word has a maximum of

$$V_{\text{max}} = V_R[2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7} + 2^{-8}] = 0.9961 V_R$$

An alternative equation to Equation (3.5) is often easier to use. This is based on noting that the expression in brackets in Equation (3.5) is really just the fraction of total counting states possible with the n bits being used. With this recognition, we can write

$$V_{\text{out}} = \frac{N}{2^n} V_R \quad (3.6)$$

where N = base 10 whole-number equivalent of DAC input

Suppose an 8-bit converter with a 5.0-V reference has an input of 10100111_2 , or A7H. If this input is converted to base 10, we get $N = 167_{10}$ and $2^8 = 256$. From Equation (3.6), the output of the ADC will be

$$V_{\text{out}} = \frac{167}{256} 5.0 = 3.2617 \text{ volts}$$

EXAMPLE 3.9 What is the output voltage of a 10-bit DAC with a 10.0-V reference if the input is (a) 0010110101₂ = 0B5H, (b) 20FH? What input is needed to get a 6.5-V output?

Solution

Let's use Equation (3.5) for part (a) and Equation (3.6) for part (b). Thus, for the 0B5H input, we have

$$\begin{aligned}V_{\text{out}} &= 10.0[2^{-3} + 2^{-5} + 2^{-6} + 2^{-8} + 2^{-10}] \\V_{\text{out}} &= 10.0[0.1767578] \\V_{\text{out}} &= 1.767578 \text{ V}\end{aligned}$$

For (b), we have 20FH = 527₁₀ and 2¹⁰ = 1024, so

$$\begin{aligned}V_{\text{out}} &= (527/1024)10.0 \\V_{\text{out}} &= (0.514648)10.0 \\V_{\text{out}} &= 5.14648 \text{ V}\end{aligned}$$

We can use Equation (3.6) to determine the input needed to get a 6.5-V output by solving for N ,

$$\begin{aligned}N &= 2^n(V_{\text{out}}/V_R) \\N &= 1024(6.5/10) \\N &= 665.6\end{aligned}$$

The fact that there is a fractional remainder tells us that we cannot get exactly 6.5 V from the converter. The best we can do is get an output for $N = 665 = 299\text{H}$ or $666 = 29\text{AH}$. The outputs for these two inputs are 6.494 V and 6.504 V, respectively. The only way to get exactly 6.5 V of output would be to change the value of the reference slightly.

Bipolar DAC Some DACs are designed to output a voltage that ranges from plus to minus some maximum when the input binary ranges over the counting states. Although computers frequently use 2s complement to represent negative numbers, this is not common with DACs. Instead, a simple *offset-binary* is frequently used, wherein the output is simply biased by half the reference voltage of Equation (3.6). The bipolar DAC relationship is then given by

$$V_{\text{out}} = \frac{N}{2^n} V_R - \frac{1}{2} V_R \quad (3.7)$$

Notice that if $N = 0$, the output voltage will be given by the minimum value, $V_{\text{out}}(\text{min}) = -V_R/2$. However, the maximum value for N is equal to $(2^n - 1)$, so that the maximum value of output voltage will be

$$V_{\text{out}}(\text{max}) = \frac{(2^n - 1)}{2^n} V_R - \frac{1}{2} V_R = \frac{1}{2} V_R - \frac{V_R}{2^n}$$

EXAMPLE 3.10 A bipolar DAC has 10 bits and a reference of 5 V. What outputs will result from inputs of 04FH and 2A4H? What digital input gives a zero output voltage?

Solution

The inputs of 04FH and 2A4H can easily be converted to base 10 numbers 79_{10} and 676_{10} . Then, from Equation (3.7), we find

$$V_{\text{out}} = \frac{79}{1024}(5) - \frac{(5)}{2} = -2.1142578 \text{ V}$$

$$V_{\text{out}} = \frac{676}{1024}(5) - \frac{(5)}{2} = 0.80078 \text{ V}$$

The zero occurs when Equation (3.7) equals zero. Solving for N gives

$$0 = \frac{N}{1024}(5) - \frac{(5)}{2}$$

$$\text{or } N = 512_{10} = 200\text{H} = 100000000_2.$$

Conversion Resolution The conversion resolution is a function of the reference voltage and the number of bits in the word. The more bits, the smaller the change in analog output for a 1-bit change in binary word, and hence the better the resolution. The smallest possible change is simply given by

$$\Delta V_{\text{out}} = V_R 2^{-n} \quad (3.8)$$

where ΔV_{out} = smallest output change
 V_R = reference voltage
 n = number of bits in the word

Thus, a 5-bit word D/A converter with a 10-V reference will provide changes of $\Delta V_{\text{out}} = (10)(2^{-5}) = 0.3125 \text{ V per bit}$.

EXAMPLE 3.11

Determine how many bits a D/A converter must have to provide output increments of 0.04 V or less. The reference is 10 V.

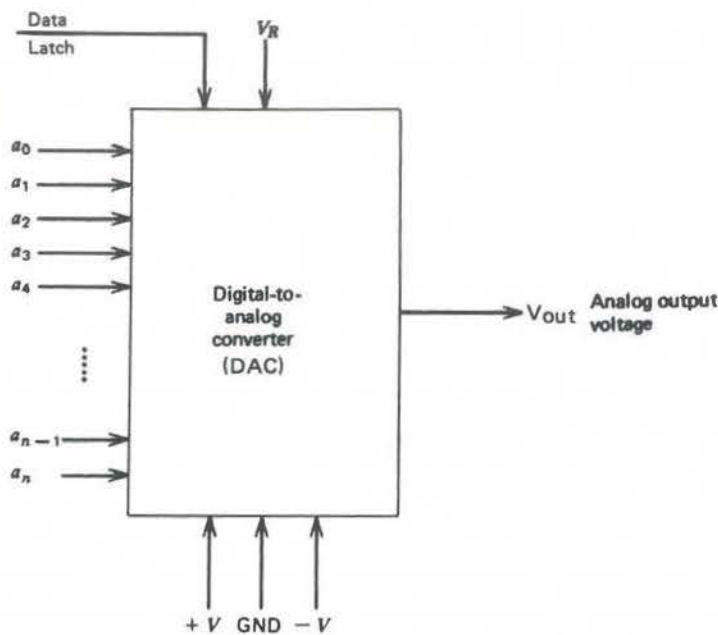
Solution

One way to find the solution is to continually try word sizes until the resolution falls below 0.04 V per bit. A more analytical procedure is to use Equation (3.8):

$$\Delta V = 0.04 = (10)(2^{-y})$$

Any n larger than the integer part of the exponent of two in this equation will satisfy the requirement. Taking logarithms

$$\begin{aligned} \log(0.04) &= \log[(10)(2^{-y})] \\ \log(0.04) &= \log(10) - y \log 2 \\ y &= \frac{\log(10) - \log(0.04)}{\log 2} \\ y &= 7.966 \end{aligned}$$

**FIGURE 3.11**

A generic DAC diagram, showing typical input and output signals.

Thus, a $n = 8$ will be satisfactory. This can be proved by Equation (3.8):

$$\begin{aligned}\Delta V_{\text{out}} &= (10)(2^{-8}) \\ \Delta V_{\text{out}} &= 0.0390625 \text{ V}\end{aligned}$$

DAC Characteristics For modern applications, most DACs are integrated circuit (IC) assemblies, viewed as a black box having certain input and output characteristics. In Figure 3.11, we see the essential elements of the DAC in terms of required input and output. The associated characteristics can be summarized as follows by referring to this figure:

1. **Digital input** Typically, digital input is a parallel binary word composed of a number of bits specified by the device specification sheet. TTL logic levels are usually required, unless otherwise noted.
2. **Power supply** The power supply is bipolar at a level of ± 12 to ± 18 V as required for internal amplifiers. Some DACs operate from a single supply.
3. **Reference supply** A reference supply is required to establish the range of output voltage and resolution of the converter. This must be a stable, low-ripple source. In some units, an internal reference is provided.
4. **Output** The output is a voltage representing the digital input. This voltage changes in steps as the digital input changes by bits, with the step determined by Equation (3.8). The actual output may be bipolar if the converter is designed to interpret negative digital inputs.

5. *Offset* Because the DAC is usually implemented with op amps, there may be the typical output offset voltage with a zero input (see Section 2.4.2). Typically, connections will be provided to facilitate a zeroing of the DAC output with a zero word input.
6. *Data latch* Many DACs have a data latch built into their inputs. When a logic command is given to latch data, whatever data are on the input bus will be latched into the DAC, and the analog output will be updated for that input data. The output will stay at that value until new digital data are latched into the input. In this way, the input of the DAC can be connected directly onto the data bus of a computer, but it will be updated only when a latch command is given by the computer.
7. *Conversion time* A DAC performs the conversion of digital input to analog output virtually instantaneously. From the moment that the digital signal is placed on the inputs to the presence of the analog output voltage is simply the propagation time of the signal through internal amplifiers. Typically, settling time of the internal amplifiers will be a few microseconds.

DAC Structure Generally speaking, a DAC is used as a black box, and no knowledge of the internal workings is required. There is some value, however, in briefly showing how such conversion can be implemented. The simplest conversion uses a series of op amps for input for which the gains have been selected to provide an output as given by Equation (3.5). The most common variety, however, uses a *resistive ladder network* to provide the transfer function. This is shown in Figure 3.12 for the case of a 4-bit converter. With the R - $2R$ choice of resistors, it can be shown through network analysis that the output voltage is given by Equations (3.5) or (3.6). The switches are analog electronic switches.

EXAMPLE 3.12

A control valve has a linear variation of opening as the input voltage varies from 0 to 10 V. A microcomputer outputs an 8-bit word to control the valve opening using an 8-bit DAC to generate the valve voltage.

- a. Find the reference voltage required to obtain a full open valve (10 V).
- b. Find the percentage of valve opening for a 1-bit change in the input word.

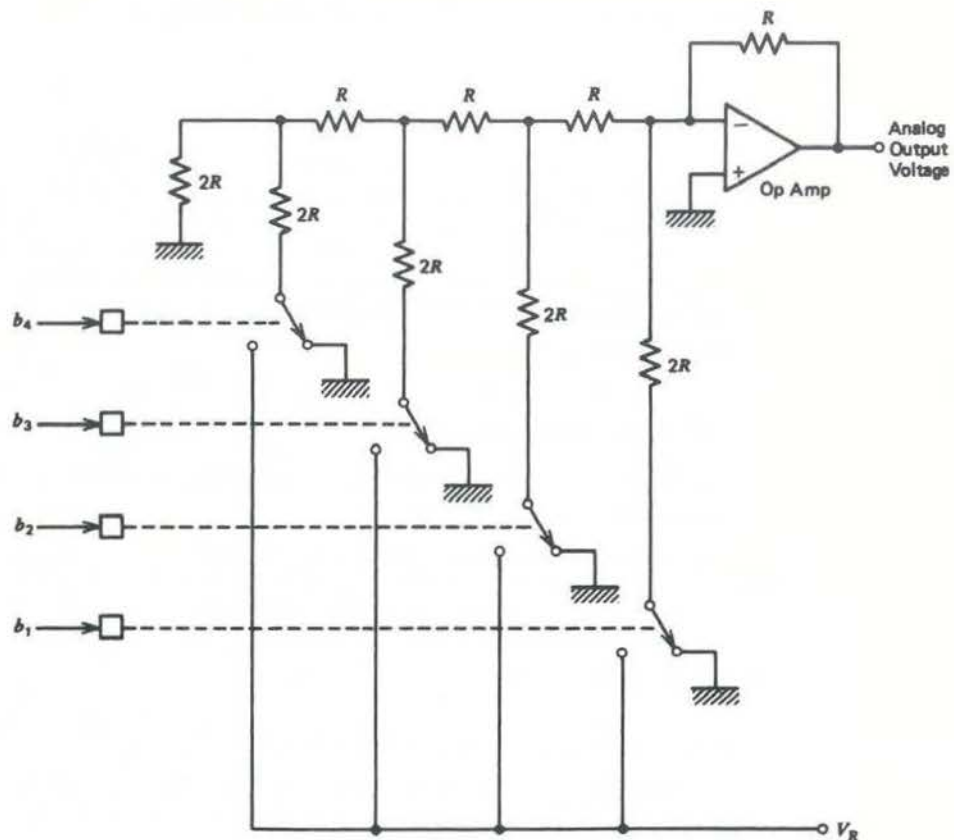
Solution

- a. The full open-valve condition occurs with a 10-V input. If a 10-V reference is used, a full digital word 11111111_2 will not quite give 10 V, so we use a larger reference. Thus, we have

$$V_{\text{out}} = V_R(b_1 2^{-1} + b_2 2^{-2} + \cdots + b_8 2^{-8})$$

$$10 = V_R \left(\frac{1}{2} + \frac{1}{4} + \cdots + \frac{1}{256} \right)$$

$$V_R = \frac{10}{0.9961} = 10.039 \text{ V}$$

**FIGURE 3.12**

A typical DAC is often implemented using a ladder network of resistors.

- b. The percentage of value change per step is found first from

$$\Delta V_{\text{out}} = V_R 2^{-8}$$

$$\Delta V_{\text{out}} = (10.039) \frac{1}{256}$$

$$\Delta V_{\text{out}} = 0.0392 \text{ V}$$

Thus,

$$\text{percent} = \frac{(0.0392)(100)}{10} = \mathbf{0.392\%}$$

Data Output Boards It is now common and convenient to obtain a printed circuit board that plugs into a personal computer expansion slot and is a complete data output system. The board has all necessary DACs, address decoding, and bus interface. In most cases, the supplier of the board also provides elementary software—often written in C, BASIC, or assembly language—as necessary to use the board for data output.

3.3.3 Analog-to-Digital Converters (ADCs)

Although there are sensors that provide a direct digital signal output and more are being developed, most still convert the measured variable into an analog electrical signal. With the growing use of digital logic and computers in process control, it is necessary to employ an ADC to provide a digitally encoded signal for the computer. The transfer function of the ADC can be expressed in a similar way to that of the DAC as given in Equation (3.5). In this case, however, the interpretation is reversed. The ADC will find a fractional binary number that gives the closest approximation to the fraction formed by the input voltage and reference.

$$b_1 2^{-1} + b_2 2^{-2} + \dots + b_n 2^{-n} \leq \frac{V_{in}}{V_R} \quad (3.9)$$

where $b_1 b_2 \dots b_n = n\text{-bit digital output}$
 $V_{in} = \text{analog input voltage}$
 $V_R = \text{analog reference voltage}$

We use an *inequality* in this equation because the fraction on the right can change continuously over all values, but the fraction derived from the binary number on the left can change only in fixed increments of $\Delta N = 2^{-n}$. In other words, the only way the left side can change is if the LSB changes from 1 to 0 or from 0 to 1. In either case, the fraction changes by only 2^{-n} and nothing in between. Therefore, there is an inherent uncertainty in the input voltage producing a given ADC output, and that uncertainty is given by

$$\Delta V = V_R 2^{-n} \quad (3.10)$$

Minimum and Maximum Voltages Equation (3.9) shows that if the ratio of input voltage to reference is less than ΔV , then the digital output will be all 0s, (i.e., 0000 ... 000₂). The LSB will not change until the input voltage becomes at least equal to ΔV , and then the output will be 0000 ... 001₂. Therefore, if the ADC output is all zeros, you know *only* that V_{in} is less than $V_R 2^{-n}$, so it could even be a negative voltage, for example.

Now notice that the MSB changes from 0 to 1 when the input voltage becomes equal to or greater than $V_R - \Delta V$. Therefore, if the ADC output is all 1s (i.e., 1111 ... 111₂) then you know *only* that V_{in} is greater than $V_R(1 - 2^{-n})$.

This uncertainty must be taken into account in design applications. If the problem under consideration specifies a certain resolution in analog voltage, then the word size and reference must be selected to provide this in the converted digital number.

EXAMPLE 3.13 Temperature is measured by a sensor with an output of 0.02 V/°C. Determine the required ADC reference and word size to measure 0° to 100°C with 0.1°C resolution.

Solution

At the maximum temperature of 100°C, the voltage output is

$$(0.02 \text{ V/}^\circ\text{C})(100^\circ\text{C}) = 2 \text{ V}$$

so a 2-V reference is used.

A change of 0.1°C results in a voltage change of

$$(0.1^\circ\text{C})(0.02 \text{ V}/^\circ\text{C}) = 2 \text{ mV}$$

so we need a word size where

$$0.002 \text{ V} = (2)(2^{-y})$$

Choose a size n that is the integer part of y plus 1. Thus, solving with logarithms, we find

$$y = \frac{\log(2) - \log(0.002)}{\log 2}$$

$$y = 9.996 \approx 10$$

so a **10-bit** word is required for this resolution. A 10-bit word has a resolution of

$$V = (2)(2^{-10})$$

$$V = 0.00195 \text{ V}$$

which is better than the minimum required resolution of 2 mV.

Notice that the output actually changes from 111111110_2 to 111111111_2 at a voltage of

$$V_R(1 - 2^{-n}) = (2 \text{ V})(1 - 2^{-10}) = 1.9980 \text{ V}$$

which corresponds to a temperature of $(1.9980 \text{ V})/(0.02 \text{ V}/^\circ\text{C}) = 99.90^\circ\text{C}$. This means we are actually measuring temperature between 0.1°C and 99.9°C .

EXAMPLE 3.14 Find the digital word that results from a 3.127-V input to a 5-bit ADC with a 5-V reference.

Solution

The relationship between input and output is given by Equation (3.9). Thus, we are to encode a fractional number of V_{in}/V_R , or

$$b_1 2^{-1} + b_2 2^{-2} + \cdots + b_5 2^{-5} = \frac{3.127}{5} = 0.6254$$

Using the method of successive multiplication defined in Section 3.2.2, we find

$$\begin{array}{ll} 0.6254(2) = 1.2508 & \therefore b_1 = 1 \\ 0.2508(2) = 0.5016 & \therefore b_2 = 0 \\ 0.5016(2) = 1.0032 & \therefore b_3 = 1 \\ 0.0032(2) = 0.0064 & \therefore b_4 = 0 \\ 0.0064(2) = 0.0128 & \therefore b_5 = 0 \end{array}$$

so that the output is **10100₂**.

Equation (3.9) can be written in a simpler fashion by expressing the fractional binary number as the fraction of counting states, as was done for the DAC. In this case, the base 10 value of the digital output can be expressed as

$$N = \text{INT}\left(\frac{V_{\text{in}}}{V_R} 2^n\right) \quad (3.11)$$

where $\text{INT}()$ means to take the integer part of the quantity in brackets. This is not a round-off, but rather a truncation, so that $\text{INT}(3.3) = 3$ and $\text{INT}(3.99) = 3$ also. The value of N is then converted to hex and/or binary to demonstrate the ADC output. In the previous example, we would have

$$N = \text{INT}\left(\frac{3.127}{5} 2^5\right) = \text{INT}(20.0128) = 20_{10}$$

or $14\text{H} = 10100_2$, as already found.

EXAMPLE 3.15 The input to a 10-bit ADC with a 2.500-V reference is 1.45 V. What is the hex output? Suppose the output was found to be 1B4H. What is the voltage input?

Solution

We will use Equation (3.11) to find the solution to these questions. For the first part, we can form the expression

$$N = \text{INT}((1.45/2.5)2^{10})$$

$$N = \text{INT}(593.92)$$

$$N = 593$$

$$N = 251\text{H}$$

So the output of the ADC is 251H for a 1.45-V input. To get the voltage input for a 1B4H-output, we solve Equation (3.11) for the voltage:

$$V_{\text{in}} = \frac{N}{2^n} V_R$$

A conversion yields $1\text{B}4\text{H} = 436_{10}$.

$$V_{\text{in}} = (436/1024)2.50$$

$$V_{\text{in}} = 1.06445\text{ V}$$

However, it is important to realize that any voltage from this to $1.06445 + 2.5/1024 = 1.06689$ will give the same output, 1B4H. So the correct answer to the question is that the input voltage lies in the range 1.06445 to 1.06689 V.

Bipolar Operation A bipolar ADC is one that accepts bipolar input voltage for conversion into an appropriate digital output. The most common bipolar ADCs provide an output called *offset-binary*. This simply means that the normal output is shifted by half the

scale so that all-zeros corresponds to the negative maximum input voltage instead of zero. In equation form, the relation would be written as

$$N = \text{INT} \left[\left(\frac{V_{\text{in}}}{V_R} + \frac{1}{2} \right) 2^n \right] \quad (3.12)$$

From this equation, you can see that if $V_{\text{in}} = -V_R/2$, the output is zero, $N = 0$. If $V_{\text{in}} = 0$, the output is half of 2^n . The output will be the maximum count when the input is $V_R/2 - V_R/2^n$. For example, for 8 bits with a 10.0-V reference, the step size is $\Delta V_{\text{in}} = (10)2^8 \approx 0.039 \text{ V}$. Looking at the possible states, we would have

$V_{\text{in}} = -5.000$	$N = 00000000_2$
$V_{\text{in}} = -4.961$	$N = 00000001_2$
etc.	
$V_{\text{in}} = -0.039$	$N = 01111111_2$
$V_{\text{in}} = 0.000$	$N = 10000000_2$
$V_{\text{in}} = +0.039$	$N = 10000001_2$
etc.	
$V_{\text{in}} = +4.961$	$N = 11111111_2$

There is an asymmetry to the result so that the converter cannot represent the full range from minus to plus $V_R/2$.

EXAMPLE 3.16

What are the hex and binary output of a bipolar 8-bit ADC with a 5.00-V reference for inputs of -0.85 V and $+1.5 \text{ V}$? What input voltage would cause an output of 72H?

Solution

Using Equation (3.12), we get

$$\begin{aligned} N &= \text{INT}((1/5.00)[-0.85 + 2.50]2^8) \\ N &= \text{INT}(84.48) \\ N &= 84_{10} \\ N &= 54\text{H} = 01010100_2 \end{aligned}$$

and

$$\begin{aligned} N &= \text{INT}((1/5.00)[1.5 + 2.50]256) \\ N &= \text{INT}(204.8) \\ N &= 204_{10} \\ N &= \text{CCH} = 11001100_2 \end{aligned}$$

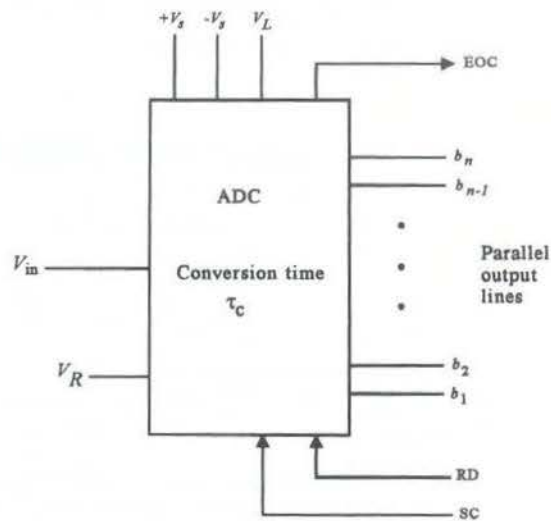
To get an output of 72H, we solve Equation (3.12) for V_{in} :

$$\begin{aligned} V_{\text{in}} &= (N/2^n)V_R - V_R/2 \\ V_{\text{in}} &= (114/256)5.00 - 2.50 \\ V_{\text{in}} &= -0.2734 \text{ V} \end{aligned}$$

But of course the actual answer is any voltage between -0.2734 V and $(-0.2734 + 5/256) = -0.2539 \text{ V}$.

FIGURE 3.13

A generic ADC diagram, showing typical input and output signals and noting the conversion time.



ADC Characteristics Figure 3.13 shows a generic ADC with all the typical connections. It is quite possible, and even appropriate in many cases, to regard the ADC as simply a black box with certain input and output characteristics. The following list summarizes the important characteristics of the ADC:

1. *Analog voltage input* This is for connection of the voltage to be converted. As will be explained later, it is important that this voltage be constant during the conversion process.
2. *Power supplies* Generally, an ADC requires bipolar supply voltages for internal op amps and a digital logic supply connection.
3. *Reference voltage* The reference voltage must be from a stable, well-regulated source. Special, integrated circuit reference-source voltages are available for this purpose.
4. *Digital outputs* The converter will have n output lines for connection to digital interface circuitry. Generally, the levels are typical TTL values for definition of the high and low states. It is common for the output lines to be tri-state outputs so that the ADC can be connected directly to a bus.
5. *Control lines* The ADC has a number of control lines that are single-bit digital inputs and outputs designed to control operation of the ADC and allow for interface to a computer. The most common lines are:
 - a. **SC** (Start-convert) This is a digital input to the ADC that starts the converter on the process of finding the correct digital outputs for the given analog voltage input. Typically, conversion starts on a falling edge.
 - b. **EOC** (End-of-convert) This is a digital output from the ADC to receiving equipment, such as a computer. Typically, this line will be high during the conversion process. When the conversion is complete, the line will go low. Thus, the falling edge indicates that the conversion is complete.
 - c. **RD** (Read) Since the output is typically buffered with tri-states, even though the conversion is complete, the correct digital results do not appear on the

output lines. The receiving equipment must take the RD line low to enable the tri-states and place the data on the output lines.

6. **Conversion time** This is not an input or an output, but a very important characteristic of ADCs. A typical ADC does not produce the digital output instantaneously when the analog voltage is applied to its input terminal. The ADC must sequence through a process to find the appropriate digital output, and this process takes time. This is one of the reasons that handshaking lines are required. Figure 3.14 shows a typical timing diagram for taking a sample of data via an ADC.

The existence of a finite conversion time complicates the use of ADCs in data acquisition. The computer cannot have a data input at any time; rather, it must request an input, wait for the ADC to perform a conversion, and then input the data.

ADC Structure Most ADCs are available in the form of integrated circuit (IC) assemblies that can be used as a black box in applications. To fully appreciate the characteristics of these devices, however, it is valuable to examine the standard techniques employed to perform the conversions. There are two methods in use that represent very different approaches to the conversion problem.

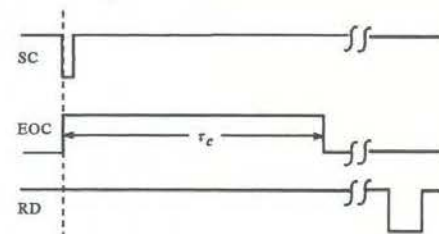
Parallel-Feedback ADC The parallel-feedback A/D converter employs a feedback system to perform the conversion, as shown in Figure 3.15. Essentially, a *comparator* is used to compare the input voltage, V_x , to a feedback voltage, V_F , that comes from a DAC as shown. The comparator output signal drives a logic network that steps the digital output (and hence DAC input) until the comparator indicates the two signals are the same within the resolution of the converter. The most popular parallel-feedback converter is the *successive approximation* device. The logic circuitry is such that it successively sets and tests each bit, starting with the most significant bit of the word. We start with all bits zero. Thus, the first operation will be to set $b_1 = 1$ and test $V_F = V_R 2^{-1}$ against V_x through the comparator.

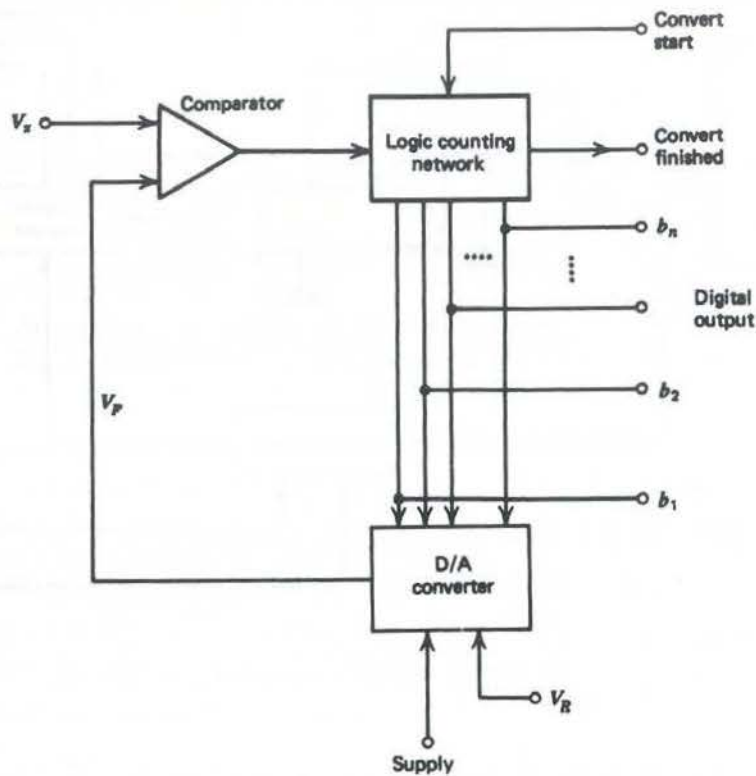
If V_x is greater, then b_1 will be 1, b_2 is set to 1, and a test is made of V_x versus $V_F = V_R(2^{-1} + 2^{-2})$, and so on.

If V_x is less than $V_R 2^{-1}$, then b_1 is reset to zero, b_2 is set to 1, and a test is made for V_x versus $V_R 2^{-2}$. This process is repeated to the least significant bit of the word. The operation can be illustrated best through an example.

FIGURE 3.14

A typical data-acquisition timing diagram using an ADC. The read operation may occur at any time after the end-of-convert has been issued by the ADC.



**FIGURE 3.15**

One common method of implementing an ADC is the successive approximation of parallel-feedback system using an internal DAC.

EXAMPLE 3.17

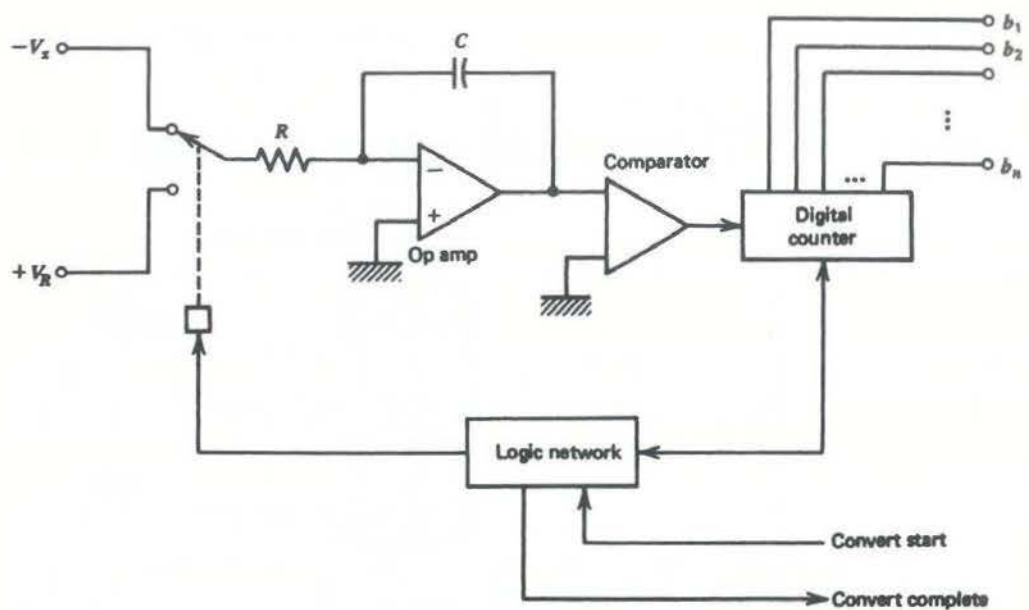
Find the successive approximation ADC output for a 4-bit converter to a 3.217-V input if the reference is 5 V.

Solution

Following the procedure outlined, we have the following operations: Let $V_x = 3.217$; then

- | | |
|-------------------------------------|--|
| (1) Set $b_1 = 1$
$V_x > 2.5$ | $V_F = 5(2^{-1}) = 2.5$ V
leave $b_1 = 1$ |
| (2) Set $b_2 = 1$
$V_x < 3.75$ | $V_F = 2.5 + 5(2^{-2}) = 3.75$
reset $b_2 = 0$ |
| (3) Set $b_3 = 1$
$V_x > 3.125$ | $V_F = 2.5 + 5(2^{-3}) = 3.125$
leave $b_3 = 1$ |
| (4) Set $b_4 = 1$
$V_x < 3.4375$ | $V_F = 3.125 + 5(2^{-4})$
reset $b_4 = 0$ |

By this procedure, we find the output is a binary word of **1010₂**.

**FIGURE 3.16**

The dual-slope ADC uses an op amp integrator, comparator, and counter. This is commonly used in digital voltmeters.

The conversion time of successive approximation-type ADCs is on the order of 1 to $5 \mu\text{s}$ per bit. Thus, a low-priced 8-bit ADC might require $5 \mu\text{s/bit}$ for a total conversion time of about $40 \mu\text{s}$. A higher-quality (and price) 12-bit might be able to perform the full conversion in only $15 \mu\text{s}$.

These conversion times depend on a clock that is internal to the ADC and not crystal controlled. Thus, there will be variation of the conversion time from unit to unit.

Ramp ADC The ramp-type A/D converters essentially compare the input voltage against a linearly increasing ramp voltage. A binary counter is activated that counts ramp steps until the ramp voltage equals the input. The output of the counter is then the digital word representing conversion of the analog input. The ramp itself is typically generated by an op amp integrator circuit, discussed in Section 2.5.8.

Dual-Slope Ramp ADC This ADC is the most common type of ramp converter. A simplified diagram of this device is shown in Figure 3.16. The principle of operation is based on allowing the input signal to drive the integrator for a fixed time, T_1 , thus generating an output of

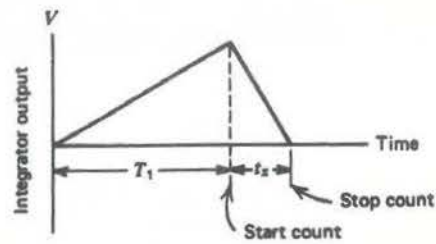
$$V_1 = \frac{1}{RC} \int V_x dt \quad (3.13)$$

or, because V_x is constant,

$$V_1 = \frac{1}{RC} = T_1 V_x \quad (3.14)$$

FIGURE 3.17

A typical timing diagram of a dual-slope ADC. Since both slopes depend upon R and C , the ADC output is independent of the values of these components.



After time T_1 , the input of the integrator is electronically switched to the reference supply. The comparator then sees an input voltage that decreases from V_1 as

$$V_2 = V_1 - \frac{1}{RC} \int V_R dt \quad (3.15)$$

or, because V_R is constant and V_1 is given from Equation (3.14),

$$V_2 = \frac{1}{RC} T_1 V_x - \frac{1}{RC} t V_R \quad (3.16)$$

A counter is activated at time T_1 and counts until the comparator indicates $V_2 = 0$, at which time t_x [Equation (3.16)] indicates that V_x will be

$$V_x = \frac{t_x}{T_1} V_R \quad (3.17)$$

Thus, the counter time, t_x , is linearly related to V_x and is independent of the integrator characteristics—that is, R and C . This procedure is shown in the timing diagram in Figure 3.17. Conversion *start* and *stop* digital signals are also used in these devices, and (in many cases) internal or external references may be used.

EXAMPLE 3.18

A dual-slope ADC as shown in Figure 3.16 has $R = 100 \text{ k}\Omega$ and $C = 0.01 \text{ }\mu\text{F}$. The reference is 10 V, and the fixed integration time is 10 ms. Find the conversion time for a 6.8-V input.

Solution

We find the voltage after an integration time of 10 ms as

$$\begin{aligned} V_1 &= \frac{1}{RC} T_1 V_x \\ V_1 &= \frac{(10 \text{ ms})(6.8 \text{ V})}{(100 \text{ k}\Omega)(0.1 \text{ }\mu\text{F})} \\ V_1 &= 6.8 \text{ V} \end{aligned}$$

Then we find the time required to integrate this to zero as $V_2 = 0$ in

$$V_2 = \frac{T_1 V_x}{RC} - \frac{t_x}{RC} V_R$$

Thus,

$$t_x = \frac{T_1 V_x}{V_R}$$

$$t_x = \frac{(10 \text{ ms})(6.8 \text{ V})}{10 \text{ V}}$$

$$t_x = 6.8 \text{ ms}$$

The total conversion time is then $10 \text{ ms} + 6.8 \text{ ms} = 16.8 \text{ ms}$.

Example 3.18 illustrates an important characteristic of the dual-slope ADC. It has a much longer conversion time than a successive approximation type. In fact, the conversion time for the dual-slope is frequently from tens to several hundreds of milliseconds.

One of the most common applications of the dual-slope ADCs is in digital multi-meters. Here, input circuitry converts the input voltage into an appropriate range for the ADC. The ADC performs conversions continuously; that is, when one conversion is finished, the output is latched into a display register and another conversion is started. In applications such as this, a few hundred milliseconds' conversion time is plenty fast and allows for the display to be updated several times per second.

EXAMPLE 3.19

A measurement of temperature using a sensor that outputs $6.5 \text{ mV}/^\circ\text{C}$ must measure to 100°C . A 6-bit ADC with a 10-V reference is used. (a) Develop a circuit to interface the sensor and the ADC. (b) Find the temperature resolution.

Solution

To measure to 100°C means the sensor output at 100°C will be

$$(6.5 \text{ mV}/^\circ\text{C})(100^\circ\text{C}) = 0.65 \text{ V}$$

- a. The interface circuit must provide a gain so that at 100°C the ADC output is **111111**. The input voltage that will provide this output is found from

$$V_x = V_R(a_1 2^{-1} + a_2 2^{-2} + \cdots + a_6 2^{-6})$$

$$V_x = 10 \left(\frac{1}{2} + \frac{1}{4} + \cdots + \frac{1}{64} \right)$$

$$V_x = 9.84375 \text{ V}$$

Thus, the required gain must provide this voltage when the temperature is 100°C .

$$\text{gain} = \frac{9.84375}{0.65}$$

$$\text{gain} = 15.14$$

The op amp circuit of Figure 3.18 will provide this gain.

- b. The temperature resolution can be found by working backward from the least significant bit (LSB) voltage change of the ADC:

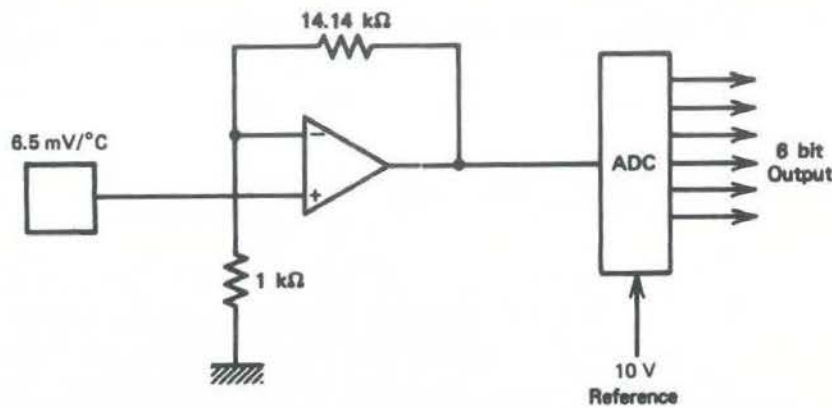


FIGURE 3.18
Analog circuit for Example 3.19.

$$\Delta V = V_R 2^{-n}$$

$$\Delta V = (10)(2^{-6}) = 0.15625 \text{ V}$$

Working back through the amplifier, this corresponds to a sensor change of

$$\Delta V_T = \frac{0.15625}{15.14} = 0.01032 \text{ V}$$

or a temperature of

$$\Delta T = \frac{0.01032 \text{ V}}{0.0065 \text{ V/}^\circ\text{C}} = 1.59^\circ\text{C}$$

Conversion-Time Consequences The finite conversion time of the ADC has serious consequences on the rate of change of signals presented for conversion. An ADC performs the conversion process by referring back to the input signal while the conversion is taking place. Obviously, if the input is changing while this process is taking place, errors will occur.

Consequently, the ADC output will be in error if the magnitude of the input voltage changes by more than one LSB voltage, ΔV , during the time of conversion, τ_c . This is a serious limitation. Since the change in time of the input voltage is just the derivative, this condition can be written in the form

$$\frac{dV_{in}}{dt} \leq \frac{\Delta V}{\tau_c} = \frac{V_R}{2^n \tau_c} \quad (3.18)$$

Consider, for example, a 10-bit ADC with a 5.0-V reference and a 20- μs conversion time. According to Equation (3.18), the maximum rate of change of the input voltage for this converter would be

$$\frac{dV_{in}}{dt} \leq \frac{5.0}{2^{10}(20 \times 10^{-6})} \approx 244 \text{ V/s}$$

This result, 244 V/s, doesn't seem so bad. Let's put this in perspective by asking what frequency it would correspond to if the input were a sinusoidal voltage,

$$V_{in} = V_0 \sin(\omega t)$$

Taking the derivative and using Equation (3.18) gives the result

$$\omega V_0 \cos(\omega t) \leq \frac{V_R}{2^n \tau_c}$$

or, since the maximum value of the cosine function is unity, we get the condition

$$\omega \leq \frac{V_R}{2^n \tau_c V_0} \quad (3.19)$$

or, in terms of frequency where $\omega = 2\pi f$,

$$f \leq \frac{V_R}{2^{n+1} \pi \tau_c V_0} \quad (3.20)$$

Returning to the 244 V/s, let's assume that the full range is in use, $V_0 = V_R$. Then we find that the maximum angular frequency is

$$\omega \leq \frac{1}{2^{10}(20 \times 10^{-6})} \approx 48.8 \text{ rad/s}$$

or, in terms of frequency,

$$f \leq \frac{\omega}{2\pi} \approx 7.8 \text{ Hz}$$

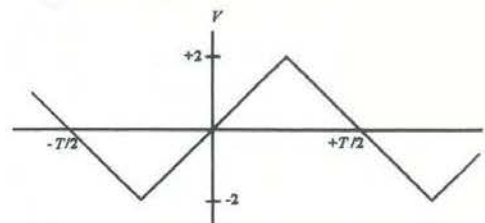
This is a remarkable result! It says that this 20- μ s converter cannot find a 10-bit representation of an oscillating signal greater than 7.8 Hz. Yet it is true, and means that if the frequency is greater than this, there will be errors in the lower-order bits; that is, it is no longer converting to 10 bits.

EXAMPLE 3.20

An 8-bit, bipolar ADC with a 5-V reference will be used to take samples of a triangular wave as shown in Figure 3.19. What is the maximum frequency of the wave if the ADC conversion time is 12 μ s?

FIGURE 3.19

Input signal for Example 3.20.



Solution

The solution is found from the condition expressed by Equation (3.18). From Figure 3.19 it is clear that the derivative of the input signal is simply the slope of the triangular wave,

$$\frac{dV_{in}}{dt} = \frac{2}{T/4} = \frac{8}{T} = 8f$$

where T is the period and f is the frequency. Then, from Equation (3.18),

$$8f \leq \frac{5}{2^8(12 \times 10^{-6})} = 1627.6 \text{ Hz}$$

or $f \leq 203.5 \text{ Hz}$.

Obviously, the limitations on frequency described previously severely limit the application of ADCs and computer data-acquisition systems. There is a solution, however. What is needed is simply that the signal not change during the conversion process. Therefore, the answer is to hold the value constant during that process. This is accomplished with a sample-and-hold (S/H) circuit.

Sample-and-Hold The basic concept of the sample-and-hold circuit is shown in Figure 3.20, where the S/H is connected to the input of an ADC. When the electronic switch is closed, the capacitor voltage will “track” the input voltage, $V_c(t) = V_{in}(t)$. At some time, t_s , when a conversion of the input voltage is desired, the electronic switch is opened, isolating the capacitor from the input. Thus, the capacitor will hold (stay charged) to the voltage when the switch opened, $V_c = V_{in}(t_s)$.

The voltage follower allows this voltage to be impressed upon the ADC input, but the capacitor does not discharge because of the very high input impedance of the follower. The start-convert is then issued, and the conversion proceeds with the input voltage remaining constant, so the problem of Equation (3.18) does not arise.

When the conversion is complete, the electronic switch is reclosed, and tracking continues until another conversion is needed. Figure 3.21 shows how $V_{in}(t)$ and $V_c(t)$ would appear during a sample collection sequence of a sinusoidal signal.

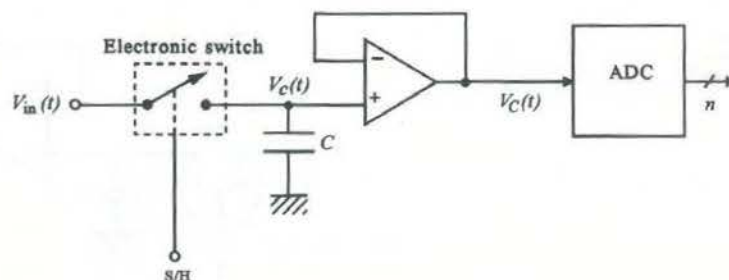
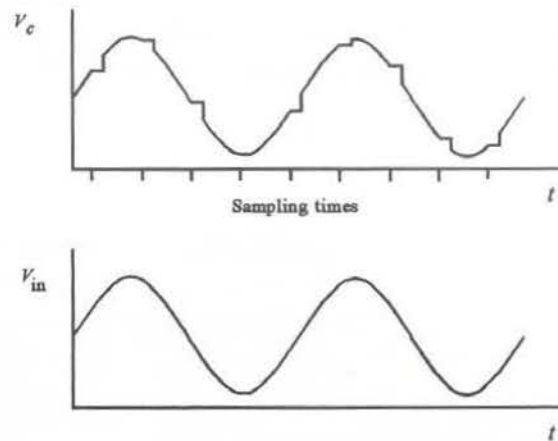


FIGURE 3.20

The basic concept of a sample-and-hold circuit for use with the ADC.

FIGURE 3.21

The sampled signal is literally “held” during the ADC conversion process.



Practical S/H Issues Because of the severe limitations of Equation (3.18), ADC systems virtually always use S/H circuits on the input. Whenever an SC is issued to the ADC, a “hold” is also issued to the S/H circuit. When the EOC is issued by the ADC, this usually automatically switches the S/H back to the “sample” mode. Figure 3.22 shows how IC S/H circuits are often implemented using FETs as the electronic switches.

There are several practical issues associated with the nonideal electrical characteristics of the elements involved in S/H circuits.

- There is a nonzero resistance path from the input voltage to the capacitor. This resistance consists of the output resistance of the source of V_{in} and the finite “ON” resistance of the electronic switch (the FET, for example).

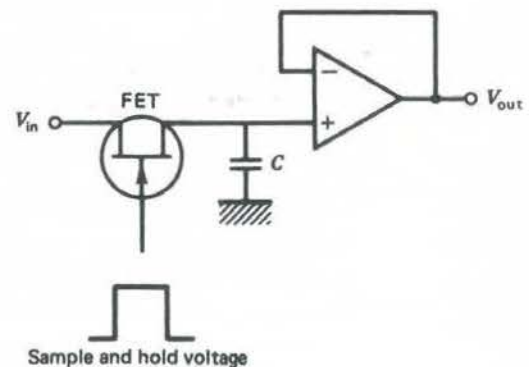
Figure 3.23a shows a model of the sample mode of the S/H. You can see that this constitutes a low-pass filter with $R = R_s + R_{ON}$ and the capacitor. Therefore, there will be a limitation on the frequency that the system can track. The capacitor voltage will be down 3 dB (0.707) at the critical frequency given by

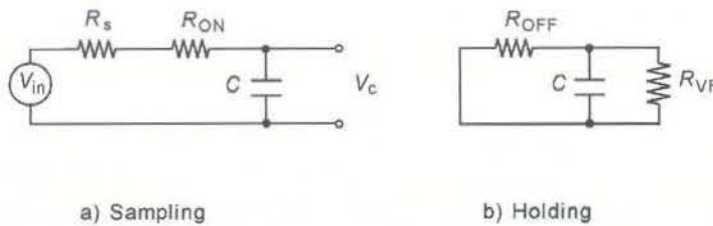
$$f_c = \frac{1}{2\pi(R_s + R_{ON})C} \quad (3.21)$$

Most commercial S/H circuits reduce this limitation by using a voltage follower before the switch, since it has very low output resistance.

FIGURE 3.22

A S/H often uses a FET as an electronic switch.



**FIGURE 3.23**

During (a) sampling and (b) holding, equivalent circuit resistance creates nonideal effects.

- During the hold mode, the capacitor is shunted by the parallel combination of the “OFF” resistance of the switch and the input resistance of the voltage follower, as shown in Figure 3.23b. In this case, there will be a gradual discharge, or “droop,” of the capacitor voltage. The droop time constant for this discharge is simply

$$\tau_D = \frac{R_{\text{OFF}} R_{\text{VF}}}{R_{\text{OFF}} + R_{\text{VF}}} C \quad (3.22)$$

The condition here is that the droop of the capacitor voltage cannot have a slope larger than that given by Equation (3.18). For the exponential decay of the capacitor voltage, this leads to the condition

$$\frac{V_C}{\tau_D} \leq \frac{V_R}{2^n \tau_c} \quad (3.23)$$

or

$$\tau_D \geq 2^n \tau_c \frac{V_C}{V_R} \quad (3.24)$$

Equation (3.24) shows how the value of capacitor C can be selected to ensure that droop will not exceed the limitation imposed by Equation (3.18). Usually, the equation is evaluated for $V_C = V_R$ as a worst-case condition.

EXAMPLE 3.21

A S/H will be used with a 12-bit, unipolar ADC with a $30\text{-}\mu\text{s}$ conversion time. The S/H switch ON resistance is $10\text{ }\Omega$, and its OFF resistance is $10\text{ M}\Omega$. The voltage follower input resistance is also $10\text{ M}\Omega$, while the signal source output resistance is $50\text{ }\Omega$.

- What value of capacitor should be used?
- Determine the sampling cutoff frequency.

Solution

- Equation (3.24), under the worst-case condition that $V_C = V_R$, will determine the minimum droop time, $\tau_D \geq 2^{12}(30 \times 10^{-6}) = 0.12288\text{ s}$. Now Equation (3.22) will allow determination of C ,

$$\frac{(10^7)(10^7)}{(10^7 + 10^7)} C \geq 0.12288$$

which gives $C \geq 0.025\text{ }\mu\text{F}$.

- b. From this result and Equation (3.22), the critical frequency during sampling is found to be

$$f_c = \frac{1}{(2\pi)(50 + 10)(0.025 \times 10^{-6})} = 108 \text{ kHz}$$

- Other important characteristics of the S/H circuit are the acquisition time (τ_{acq}) and the aperture time (τ_{ap}). The acquisition time is the time required for the S/H to reacquire the signal when changing from the hold to the sample mode. Typical times are on the order of microseconds. This places limits on the frequency with which samples can be taken.

The aperture time is the time between when a command to hold is given and the actual signal level is held. This delay means that the value presented to the ADC is not exactly the value at the time the sample was requested. This time is typically less than $1 \mu\text{s}$.

In general, the minimum time between samples taken from a S/H and ADC system is given by the sum

$$T = \tau_c + \tau_{acq} + \tau_{ap} \quad (3.25)$$

or, if expressed in terms of the maximum throughput frequency, $f_{\max} = 1/T$.

EXAMPLE 3.22

A S/H has a 50-ns aperture time and a $4\text{-}\mu\text{s}$ acquisition time, and the ADC has a $40\text{-}\mu\text{s}$ conversion time. What is the maximum throughput frequency?

Solution

The frequency is given by the inverse of the time between samples, given by Equation (3.25):

$$T = 40 \mu\text{s} + 0.05 \mu\text{s} + 4 \mu\text{s} = 44.05 \mu\text{s}$$

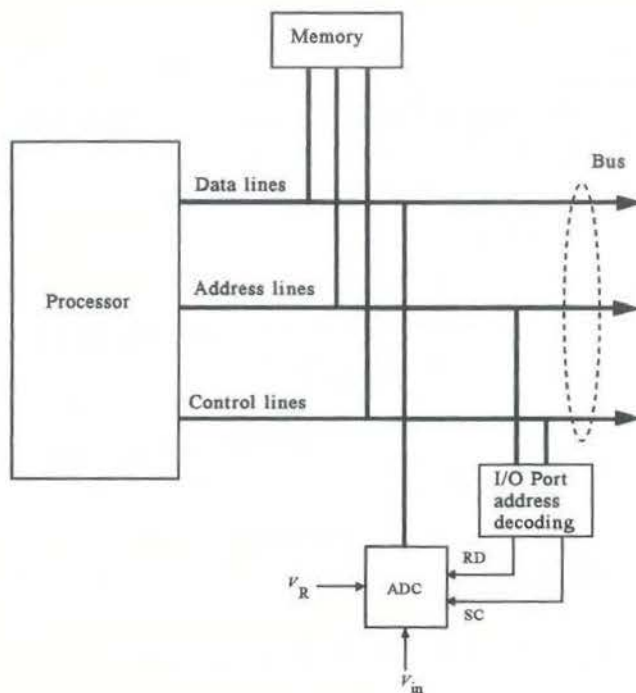
Thus, the frequency is $f_{\max} = 22.7 \text{ kHz}$.

Microprocessor-Compatible ADCs Just as with DACs, a whole line of ADCs have been developed that interface easily with microprocessor-based computers. The ADCs have built-in tri-state outputs so that they can be connected directly to the data bus of the computer. Data from the ADC are placed on the data bus lines only when the computer issues an appropriate enable command (often called a READ). Figure 3.24 shows how the ADC appears when connected to the environment of the microprocessor-based computer. The ADC appears much the same as memory. In some cases, an ADC input is actually taken by the computer using a memory-read instruction.

The decoding circuitry is necessary to provide the start-convert command, to input the convert-complete response from the ADC, and to issue the tri-state enable back to the ADC.

3.3.4 Frequency-Based Converters

There is another important method by which an analog sensor signal can be converted into a digital signal. This is based upon converting the sensor signal into a variable frequency

**FIGURE 3.24**

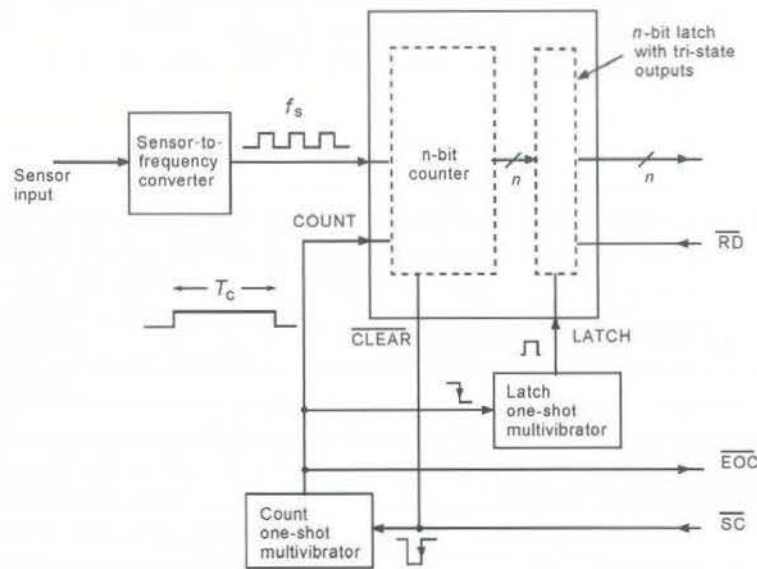
An ADC can be interfaced directly to the computer bus if it has tri-state outputs. Address decoding is required so the ADC can be operated by computer software.

and then using this frequency as input to a counter for a fixed interval of time. The output of the counter is then a measure of the frequency and thus the sensor signal.

Figure 3.25 shows a diagram that describes the essential elements of this type of analog-to-digital converter. An as yet not identified device converts the sensor signal into a proportional frequency, f_s . This frequency signal is typically a square wave, as suggested in the figure. The square wave is fed to an n -bit counter, which counts every rising (or falling) edge of the wave and hence every cycle. The counter often has a latch on the output that allows the counter to be accumulating a new count of input frequency while still maintaining the previous output.

A conversion cycle starts with a start-convert (SC) signal from the computer. This clears the counter and triggers a one-shot convert multivibrator (MV), which controls the operation. The MV is a simple digital IC that, when triggered, outputs a single pulse of some desired time duration. The count MV outputs a pulse of duration T_c when triggered by the SC. This pulse acts as a start/stop signal to the counter and so defines the time over which the frequency signal will be counted. If the input frequency is high, the count will be high; if the frequency is low, the count will be low. The latch MV issues a short pulse to latch the latest count into the output latches when the count time, T_c , is finished. The falling edge of T_c also signals the computer that a conversion is complete (EOC). The computer can then read the count by enabling the tri-state output of the counter latch with the RD signal taken low.

A typical design starts from the range of frequency of the converted sensor signal, f_{min} to f_{max} . For maximum resolution, we then make the count time, T_c , such that if the

**FIGURE 3.25**

General diagram of a frequency-based analog-to-digital converter.

sensor signal produces the maximum frequency, the count will also be at its maximum. For an n -bit binary counter, the maximum count is $2^n - 1$, so the relation we need is

$$T_c = \frac{2^n - 1}{f_{\max}} \quad (3.26)$$

The counter output for any other frequency is simply $N = fT_c$.

Example 3.23 illustrates this concept.

EXAMPLE 3.23

A sensor signal is converted to a frequency that varies from 2.0 to 20 kHz. This signal is to be converted into an 8-bit digital signal. Specify the count time, T_c . What is the range of count output for the sensor signal's frequency range?

Solution

An 8-bit counter has a maximum output of 255_{10} (11111111_2). Therefore, when the frequency is at its maximum, the count time must allow the counter to reach a full 255. From Equation (3.26),

$$T_c = \frac{2^8 - 1}{20,000 \text{ Hz}} = 0.01275 \text{ s}$$

So the one-shot multivibrator is configured to provide a 12.75-ms pulse. When the frequency is at the minimum of 2.0 kHz, the count will be

$$N = (2000 \text{ Hz})(0.01275 \text{ s}) = 25.5$$

or simply $25_{10} = 00011001_2$, since the counter can count only in integers.

Sensor-to-Frequency Conversion Of course, this technique of analog-to-digital conversion depends upon converting the measured variable information into a variable frequency. In fact, this is not so hard to do. Common ICs exist that readily convert voltage or current to frequency. An example is the LM331 voltage-to-frequency IC. Figure 3.26 shows a generic LM331 circuit for controlling output frequency with an input voltage. For this circuit, the output frequency is determined by the relation

$$f_{\text{out}} = \frac{R_S}{R_L} \frac{1}{R_f C_f} \frac{V_{\text{in}}}{2.09} \quad (3.27)$$

where the components are defined in the schematic. Actually, the frequency cannot really be zero even when V_{in} is zero. It is typically small, perhaps 10 Hz. Generally R_S is in the range of 10 to 20 k Ω and is adjustable to allow fine-tuning of the frequency-to-voltage scale factor. R_L is used to provide a discharge path for the 1- μF capacitor. It typically has a value of about 100 k Ω . The supply voltage can be up to 40 V. The output is open collector (OC), so a pull-up resistor is required to the logic supply voltage (for example, +5 V for TTL).

As an example, suppose we want an input voltage of 0 to 5.0 V to generate a frequency from ≈ 0 to 10 kHz. If we use typical values of $R_S = 15$ k Ω and $R_L = 100$ k Ω , Equation (3.27) provides

$$10,000 = \frac{15 \text{ k}\Omega}{100 \text{ k}\Omega} \frac{5.0}{2.09} \frac{1}{R_f C_f}$$

so we get $R_f C_f = 3.59 \times 10^{-5}$ s. Thus, picking $C_f = 0.01$ μF means $R_f = 3.59$ k Ω .

The value of output frequency also depends upon resistors and a capacitor, so by fixing the control voltage, the frequency can also be made to depend upon these values. Therefore, if the sensor is a varying resistance or capacitance, it can often be used directly in the circuit to provide a frequency that varies with the sensed quantity. You must remember, however, that the variation of frequency with either resistance or capacitance is often nonlinear.

Another common IC that can be used for converting resistance or capacity to a frequency is the 555 timer. This IC has an output frequency that depends upon an external

FIGURE 3.26

The LM331 is a common voltage-to-frequency converter useful in frequency-based ADCs.

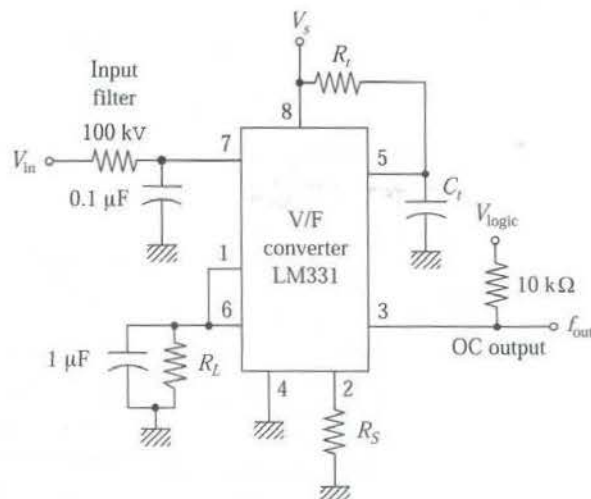
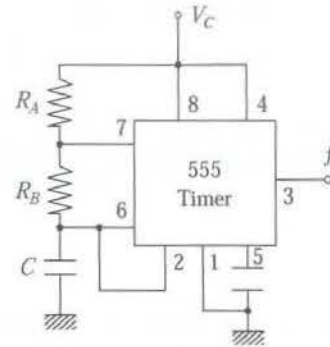


FIGURE 3.27

The 555 timer is useful for generation of a frequency that depends upon resistance or capacity.



resistor and capacitor. Figure 3.27 shows a standard 555 timer circuit that produces a steady, nonsymmetrical rectangular wave with frequency

$$f = \frac{1}{0.693(R_A + 2R_B)C} \quad (3.28)$$

The following example illustrates how the 555 can be used to produce a varying frequency and analog-to-digital conversion.

**EXAMPLE
3.24**

A resistor varies from 36 to 4 k Ω as light intensity varies from 1.5 to 10 W/m². Devise a frequency-based 10-bit ADC of this signal. What are the frequencies at 1.5 and 10 W/m²? Plot the counter output versus light intensity. Notice the nonlinearity.

Solution

We want the maximum frequency to produce a count of $2^{10} - 1 = 1023$, but maximum frequency is unspecified, so we cannot determine the count time. We get to pick one, so let us pick the count time to be, say, 10 ms. Then we get the maximum frequency as $f_{\max} = (1023/10 \text{ ms}) = 102,300 \text{ Hz}$. This frequency will occur when the sensor is at 4 k Ω :

$$102,300 \text{ Hz} = \frac{1}{0.693(4 \text{ k}\Omega + 2R_B)C}$$

This leaves another value we can pick. Let's make $R_B = 2 \text{ k}\Omega$; then we get

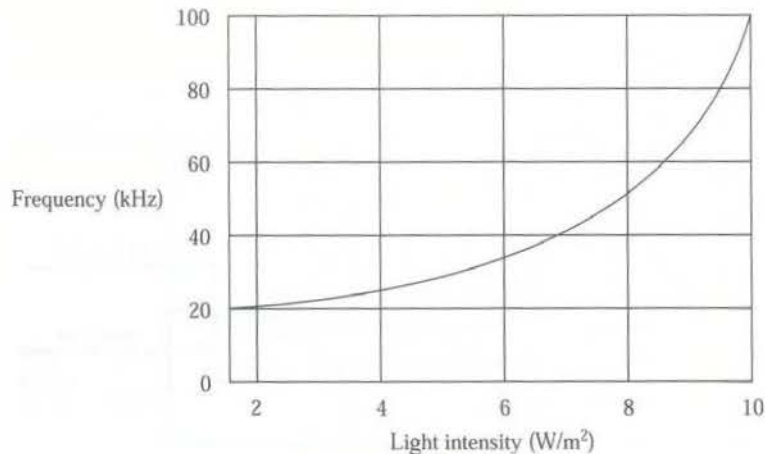
$$C = \frac{1}{0.693(4 \text{ k}\Omega + 4 \text{ k}\Omega)(102,300 \text{ Hz})} = 0.0018 \mu\text{F}$$

We get the frequency at 1.5 W/m² by using 36 k Ω for R_A .

$$f_{\min} = \frac{1}{0.693(36 \text{ k}\Omega + 4 \text{ k}\Omega)(0.0018 \mu\text{F})} = 20,042 \text{ Hz}$$

So the count is $N = (20,042 \text{ Hz})(0.01 \text{ s}) = 200.42$, or simply 200. A plot of the count versus light intensity is shown in Figure 3.28. You can see the nonlinearity, which results from the resistor being in the denominator of the frequency equation.

In general, it is important for you to realize that there are simple IC circuits that allow conversion of the sensed quantity directly to frequency. If this frequency is counted using a

**FIGURE 3.28**

Response from Example 3.24.

simple system, as shown in Figure 3.26, then analog-to-digital conversion has resulted. The conversion time of these circuits is determined by T_c and the maximum sensor frequency. Most of the IC circuits can have a maximum frequency of around 100 kHz. This means, for example, that an 8-bit converter would need a count time of about 2.55 ms. So you can see that conversion times are long compared with successive approximation types of ADCs. For most process-control applications, the slower conversion time is not a limitation.

3.4 DATA-ACQUISITION SYSTEMS

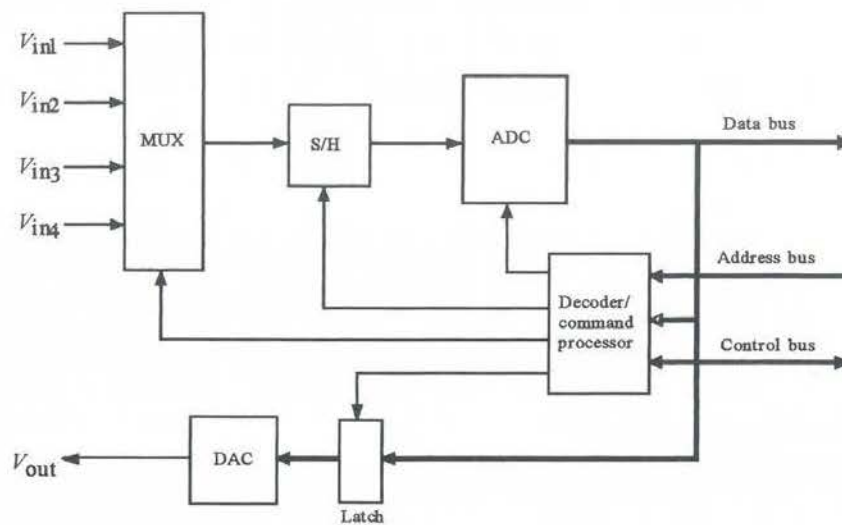
Microprocessor-based personal computers (PCs) are used extensively to implement direct digital control in the process industries. These familiar desktop computers are designed much like the system shown in Figure 3.24, using a bus that consists of the data lines, address lines, and control lines. All communication with the processor is via these bus lines. This includes essential equipment such as RAM, ROM, disk, and CD-ROM.

The PC also connects the bus lines to a number of printed circuit board (PCB) sockets, using an industry standard configuration of how the bus lines are connected to the socket. These sockets are referred to as *expansion slots*. Many special types of peripheral equipment such as fax/modem boards, game boards, and network connection boards are designed on PCBs that plug into these expansion slots.

Special PCBs called data-acquisition systems (DASs) have been developed for the purpose of providing for input and output of analog data. These are used when the PC is to be used in a control system. The following paragraphs provide general information about the hardware and software of data-acquisition systems.

3.4.1 DAS Hardware

The hardware features of a general data-acquisition system are shown in Figure 3.29. Although there is variation from manufacturer to manufacturer, the system shown in this figure and described herein demonstrates the essential features of DASs.

**FIGURE 3.29**

Typical layout of a data-acquisition board for use in a personal computer expansion slot.

ADC and S/H The DAS typically has a high-speed, successive approximation-type ADC and a fast S/H circuit. Whenever the DAS is requested to obtain a data sample, the S/H is automatically incorporated into the process. The ADC conversion time constitutes the major part of the data sample acquisition time, but the S/H acquisition time must also be considered to establish maximum throughput.

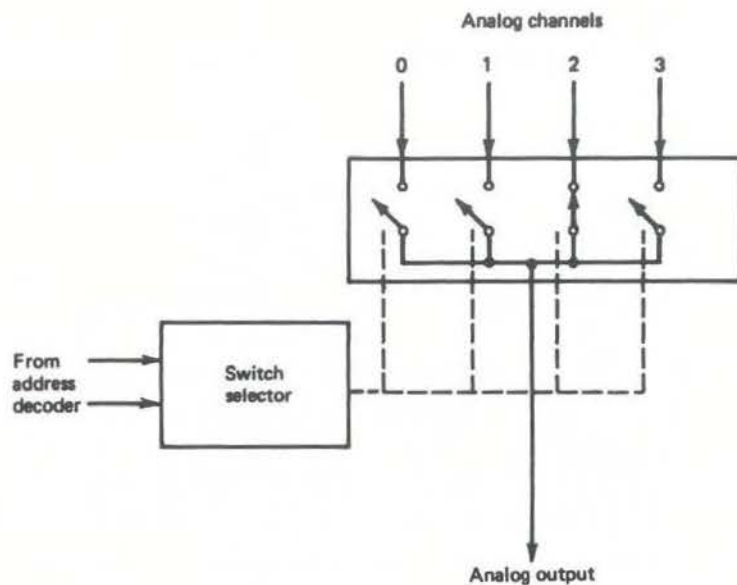
Analog Multiplexer The analog multiplexer (MUX) allows the DAS to select data from a number of different sources. The MUX has a number of input channels, each of which is connected to a different analog input voltage source. The MUX acts like a multiple set of switches, as illustrated in Figure 3.30, arranged in such a fashion that any one of the input channels can be selected to provide its voltage to the S/H and ADC. In some cases, the DAS can be programmed to take channel samples sequentially.

Address Decoder/Command Processor The computer can select to input a sample from a given channel by sending an appropriate selection on the address lines and control lines of the computer bus. These are decoded to initiate the proper sequence of commands to the MUX, ADC, and S/H. Another common feature is the ability to program the DAS to take a number of samples from a channel with a specified time between samples. In this case, the computer is notified by interrupt when a sample is ready for input.

DAC and Latch For output purposes, the DAS often includes a latch and DAC. The address decoder/command processor is used to latch data written to the DAS, which is then converted to an appropriate analog signal by the DAC.

3.4.2 DAS Software

The process of selecting a channel and initiating a data input from that channel involves some interface between the computer and the DAS. This interface is facilitated by software

**FIGURE 3.30**

An analog multiplexer acts as a multiposition switch for selecting particular inputs to the ADC.

that the computer executes. The software can be written by the user, but is often also provided by the DAS manufacturer in the form of programs on disk.

Figure 3.31 is a flowchart of the basic sequence of operations that must occur when a sample is required from the DAS. The following paragraphs describe each element of the sequence.

Generally, the DAS is mapped into a base port address location in the PC system. In the PC, this address can be from 000H to FFFH, but many addresses are reserved for use by the processor and other peripherals. A common address for input/output (I/O) systems such as the DAS is port 300H.

The sequence starts with selection of a channel for input. This is accomplished by a write to the DAS decoder that identifies the required channel. The MUX then places that channel input voltage at the S/H input.

The software then issues a start-convert (SC) command according to the specifications of the DAS. This is often accomplished by a write to some base + offset address. The DAS internally activates the hold mode of the S/H and starts the converter.

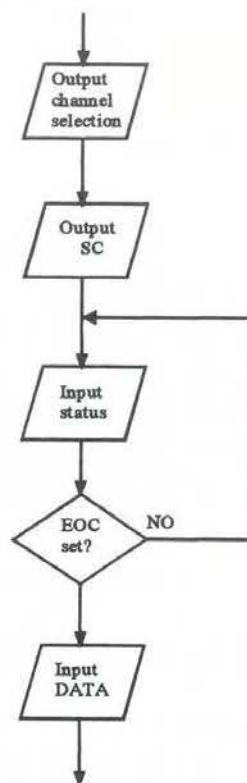
The end-of-convert (EOC) is provided in a status register in the DAS. The contents of this status register can be read by the processor by a port input of a base + offset address. The appropriate bit is then tested by the software to deduce whether the EOC has been issued.

Once the EOC has been issued, the software can input the data itself by a read of an appropriate address, again a base + offset, which enables tri-states, placing the ADC output on the data bus.

There is one problem with the operations described, shown in Figure 3.31. If the DAS fails, the computer will be locked in the loop waiting for the EOC to be issued. One way to resolve this is to add an additional timer loop for a time greater than the conversion time of

FIGURE 3.31

Software for data acquisition involves operations to start the ADC, test the EOC, and input the data.



the ADC. If the EOC is not detected prior to time-out, an error is announced, and the computer is returned to an error-handling routine.

In some cases, the EOC detection is handled by an interrupt service routine. In this way, the computer is free to execute other software until the interrupt occurs. Then the data is input. Again, there needs to be a system to detect that an EOC was not provided to protect against DAS failure.

EXAMPLE 3.25 A DAS has the following specifications:

1. 8 channels
2. 8-bit, bipolar ADC with a 5.0-V reference and a 25- μ s conversion time
3. S/H with a 10- μ s acquisition time
4. 8-bit unipolar DAC with a 10.0-V reference.

ADDRESSING: BASE = 000H to FFFH by switches

BASE + 0: READ inputs data sample

WRITE selects input channel:

b_0 set selects channel 0, and so forth, to

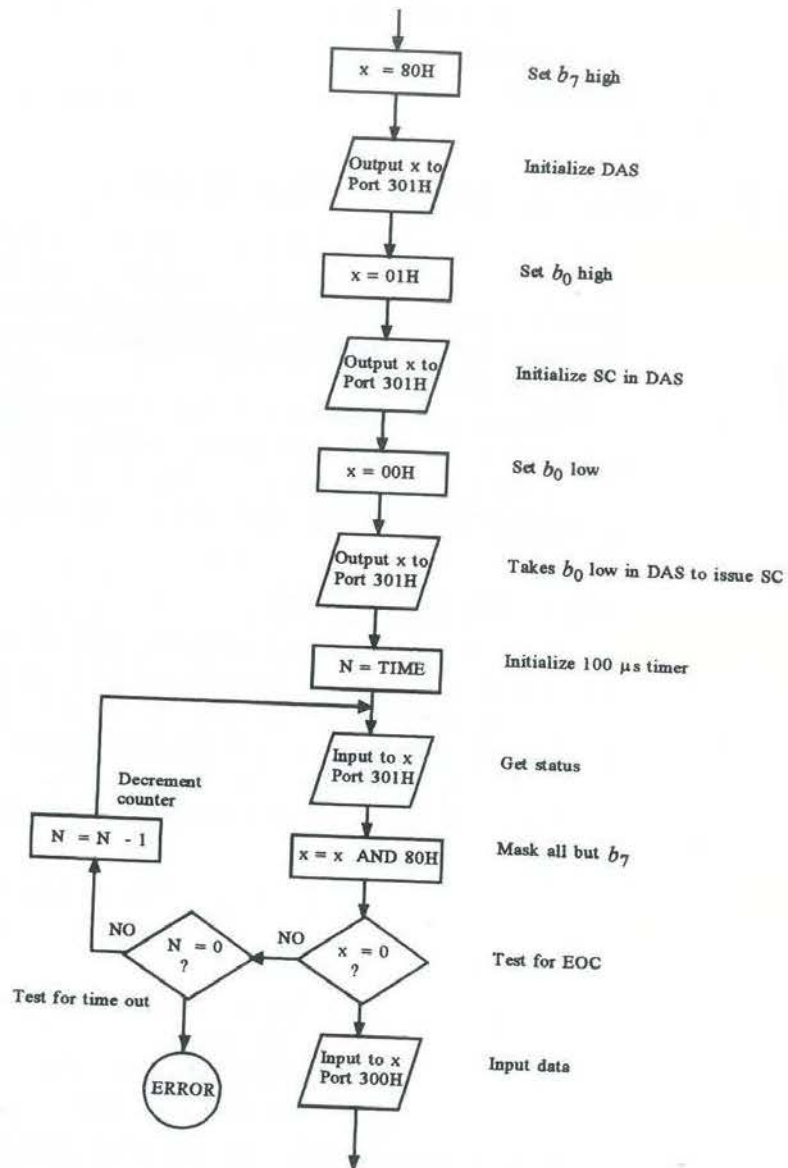
b_7 set selects channel 7

BASE + 1: READ inputs ADC status with EOC indicated by b_7 going low

WRITE initializes the DAS if b_7 is high, issues SC by taking b_0 low when b_7 is low

FIGURE 3.32

Solution to Example 3.25.



BASE + 2: READ has no action
WRITE sends data to DAC

Prepare a flowchart showing how a program would take a sample from channel 3. Include a time-out routine that jumps to ERROR (some unspecified routine) if the EOC is not issued after 100 μs . Use port 300H as the base address.

Solution

Figure 3.32 shows a flowchart indicating how the software would provide for an input from channel 3. It would be necessary to discover the value N to provide the 100- μs delay

by determining the time to sequence through the EOC testing loop. Such a flowchart could be implemented in assembly language or a higher-level language such as C, BASIC, or FORTRAN.

3.5 CHARACTERISTICS OF DIGITAL DATA

There are many advantages to using computers for the controller function. For example, one computer can handle many loops, interaction between loops can be accounted for in the software, data are less susceptible to noise-induced errors, and linearization can be easily provided by software. Other advantages include self-tuning, error correction, and automatic failure recovery.

It seems there should be a price for these advantages, and indeed, there are some disadvantages to the use of computers in controller operations. A serious disadvantage is that conversion of analog data into digital data results in a loss of knowledge about the value of the variable. The nature and consequences of this will be considered.

3.5.1 Digitized Value

Consider first analog-to-digital conversion (ADC) of analog data into a digital format. The format of the ADC output is an n -bit binary representation of the data. With n -bits it is possible to represent 2^n values, including zero. There is a finite *resolution* of the physical data being represented of one part in 2^n , and that means we now are ignorant about the value of the variable after it has been converted into the binary representation.

In equation form, we can write the relation between a physical variable and its n -bit digital representation as

$$N = \frac{(V - V_{\min})}{(V_{\max} - V_{\min})} 2^n \quad (3.29)$$

where N = base 10 equivalent of binary representation
 V = input value
 $V_{\max}$ = maximum input value
 $V_{\min}$ = minimum input value

Only the integer part of the right side of Equation (3.29) is used to determine N . Equation (3.29) assumes that the measurement system and ADC have been designed so that the binary output switches from the equivalent of $2^n - 1$ to 2^n just at $V_{\max}$. The resolution of the measurement can be found by noting what change in V will produce a single-integer (bit) change of N . This is easily seen to be

$$\Delta V = (V_{\max} - V_{\min})/2^n \quad (3.30)$$

The following example illustrates some of the consequences of the digital conversion of data. A careful study of this example will help you understand the limitations of digital representation.

**EXAMPLE
3.26**

A temperature between 100°C and 300°C is converted into a 0- to 5.0-V signal. This signal is fed to an 8-bit ADC with a 5.0-V reference. What is the actual measurement range of the system?

- What is the resolution?
- What hex output results from 169°C?
- What temperature does a hex output of C5H represent?

Solution

The nature of the ADC is that the output will change to FFH at a voltage of $5.0 - 5/256 = 4.98 \text{ V}$ (299.22°C) and would change to 100H at exactly 5.0 V (300°C). Thus, FFH would seem to mean any temperature between 299.22°C and 300°C. Because there is no 100H or higher (only 8 bits), FFH actually means any temperature *greater* than 299.22°C. Similarly, the output will be 00H for any voltage less than $5/256 = 0.0195 \text{ V}$, which is a temperature of 100.78°C, so 00H is output for any temperature *less* than 100.78°C. Thus, the actual measurement range is from 100.78°C to 299.22°C.

- The temperature span of $(300^\circ\text{C} - 100^\circ\text{C}) = 200^\circ\text{C}$ is divided into $2^8 = 256$ values. Therefore, the resolution is given by Equation (3.30).

$$\Delta T = 200^\circ\text{C}/256 = 0.78^\circ\text{C/bit}$$

Let us make sure we understand what this means. If the temperature is 100°C, the output will be 00H. It will stay at this value until the temperature reaches 100.78°C; then it will change to 01H. Thus, the resolution of 0.78°C means that we are ignorant about the value of the temperature by this amount with any reading.

- For every temperature within the range, there is one specific output value. Thus, for a temperature of 169°C, we can find that value by Equation (3.29). We find the fraction of the measurement range 169°C represented by

$$N = \frac{(169 - 100)}{(300 - 100)} 2^8 = \frac{69}{200} 256 = (0.345)(256)$$

$$N = 88.32$$

But only the integer part is used, so $88_{10} \rightarrow 58\text{H}$ and, therefore, $169^\circ\text{C} \rightarrow 58\text{H}$. Another way to get this result is to divide the quantity $(169 - 100)$ by the resolution to find the fraction of the binary number, $69/0.78 = 88.46 \rightarrow 58\text{H}$ (round-off error accounts for the difference).

- Now our ignorance of value really shows up. What temperature does C5H represent? The procedure is quite straightforward. We simply solve Equation (3.29) for T knowing that $\text{C5H} \rightarrow 197_{10}$.

$$197 = \frac{(T - 100)}{(300 - 100)} 256$$

$$T = (197)(200)/256 + 100$$

$$T = 253.9^\circ\text{C}$$

But wait! The fact is that the hex value will stay C5H until the temperature increases by the resolution, 0.78°C . So the actual answer can only be correctly stated as: the temperature is between 253.9°C and 254.68°C .

One of the consequences of the digitizing resolution is that we cannot be expected to control a value any more closely than this resolution. If we were supposed to control temperature to within $\pm 0.2^{\circ}\text{C}$ using the measurement system of Example 3.26, it would be impossible, because we do not know its value within that tolerance.

Problems of resolution are reduced by using more bits in the digital word. With 16 bits, for example, the resolution is one part in 65,536. With a 16-bit ADC using a 5.0-V reference, the least significant bit is toggled for voltage changes of only $5.0 \text{ V}/65,536 = 76.3 \mu\text{V}$! Thus, noise becomes a severe problem in a typical industrial environment.

3.5.2 Sampled Data Systems

The previous section dealt with the consequences of having only discrete knowledge of the value of the physical variable. Consider also that we have only discrete knowledge of the value *in time*. That is, the computer control system takes only periodic samples of the variable value. Thus, we are ignorant of the value or variation of the variable between samples. For the control system to function correctly, certain conditions must be assumed about variations between samples. That is what sampled data systems are all about. In the following section, we will consider the nature and consequences of having only periodic samples of the physical variable.

Sampling Rate The key issue with respect to sampling in a computer-based controller is the rate at which samples must be taken. The sample rate is expressed either through t_s , the time between samples, or $f_s = 1/t_s$, the sampling frequency.

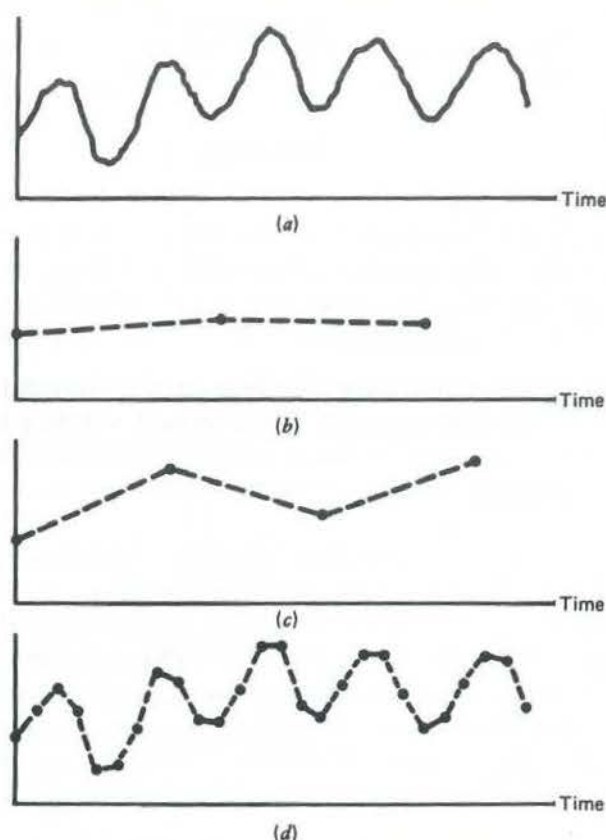
There is a maximum sampling rate in any system—that is, the time required to take a sample (ADC conversion time) plus the time required to solve the controller equations to determine the appropriate output (program execution time).

There is a minimum sampling rate in any system that depends on the nature of the time variation of the sampled variable. Simply put, samples must be taken at a high enough rate so that the signal can be reconstructed from the samples. There are serious consequences to sampling at too small a rate. For example, the control system will not be able to correct variations of the controlled variable that are missed because too few samples were taken.

Figure 3.33 illustrates the consequences of sampling rate on knowledge of signal variation. The actual signal is shown in Figure 3.33a. Figures 3.33b, 3.33c, and 3.33d illustrate knowledge about signal variation deduced from various sample rates. Reconstructions of the original signal are indicated by the dashed lines between samples.

The sampling rate of Figure 3.33b is much too slow, because little information about the actual signal variation is contained in the reconstruction from the samples.

For Figure 3.33c, the signal seems to possess a frequency of variation that is not in fact present in the actual signal. This is called *aliasing*, and it is one consequence of too small a sampling rate.

**FIGURE 3.33**

The sampling rate can disguise actual signal details.

The sampling rate of Figure 3.33d shows that the essential features of the signal can be reconstructed from the samples. A general rule for the minimum sampling rate can be deduced from the maximum frequency of the signal. The rule is that for adequate reconstruction of the signal from samples, the samples must be taken at a frequency that is about 10 times the maximum frequency of the signal.

$$f_s = 10f_{\max} \quad (3.31)$$

where f_s = sampling frequency
 $f_{\max}$ = maximum signal frequency

This is, of course, equivalent to taking 10 samples within the shortest period of the signal.

To determine the minimum sampling rate, an estimate must be made of the highest possible frequency (shortest possible period) of the signal. The sampling frequency will be 10 times that value. For Figure 3.33d, 6 samples are taken instead of 10 in one basic period of the data. The reconstruction from samples is somewhat crude, but the basic structure is present.

EXAMPLE 3.27 The plot of Figure 3.34 shows typical data taken from pressure variations in a reaction vessel. Determine the maximum time between samples for a computer control system to be used with this system.

Solution

An examination of the signal of Figure 3.34 shows that the shortest time between any two peaks is 0.15 s. This gives a maximum signal frequency of $f_{\max} = 1/0.15 = 6.7$ Hz. From Equation (3.31), the minimum sampling frequency is given by $f_s = 10f_{\max} = (10)(6.7 \text{ Hz}) = 67 \text{ Hz}$. The maximum time between samples is $t_s = 1/f_s = 15 \text{ ms}$.

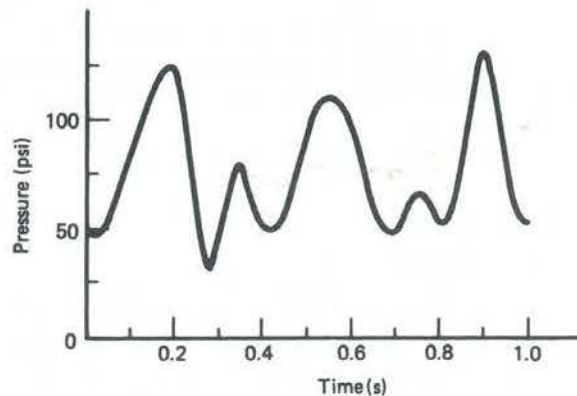
It is important to understand the connection between the suggested sampling frequency of Equation (3.31) and that suggested by the Nyquist sampling frequency. The Nyquist sampling theorem presents a result that, if a signal is bandlimited to a frequency, $f_{\max}$, then the signal can be reconstructed if sampled at a rate of twice this maximum—that is, $f_{\text{sample}} = 2f_{\max}$. This is an apparent contradiction to Equation (3.31).

This contradiction is resolved by two observations. First, practical signals in industry are not bandlimited, which would mean the maximum frequency was infinite. This can be alleviated by filtering the signal to block frequencies beyond some value $f_{\max}$, effectively providing an artificial band limit, but beyond that which presents real, practical data.

Second, the Nyquist criterion does not specify how many samples must be taken to reconstruct the signal. Indeed, in the limit of sampling at $2f_{\max}$, an infinite number of samples must be taken. In the practical world, the signal is not regular and periodic. Equation (3.31) represents a compromise that has been found to provide sufficient samples for practical signal reconstruction.

The cycle time of a process-control fieldbus can also impact the issue of the maximum signal frequency. If the fieldbus is used to take measurement data from a sensor and send the response back to the final control element, the potential delays because of bus traffic limit the controllable maximum frequency.

FIGURE 3.34
Pressure data for Example 3.27.



**EXAMPLE
3.28**

A network application is used with a control system that limits cycle time to 450 ms because of heavy bus usage. One of the control systems is to regulate periodic variations of temperature in a reaction chamber. Assuming the computer processing time is negligible, what is the maximum temperature variation frequency that can be controlled?

Solution

The 450 ms corresponds to the time for requesting a sample from the sensor and then receiving this sample. The total reaction time of the control system should include the time to get a signal back to the final control element. Thus, we add another 225 ms to the total processing time. In this case, the sampling frequency is given by

$$f_s = 1/(0.45 + 0.225) = 1.48 \text{ Hz}$$

The maximum frequency of the temperature variation is found from Equation (3.31):

$$f_{\max} = f_s/10 = 0.148 \text{ Hz,}$$

or a period of about 6.8 seconds.

3.5.3 Linearization

In many cases, the input binary number and the controlled variable are not linearly related. In such cases, it is necessary to execute a program that will linearize the binary number so that it is proportional to the controlled variable value. There are two common approaches: equation inversion and table look-up.

Linearization by Equation When an equation is known that relates the value of the controlled variable and the binary number in the computer, an equation can be developed to determine the linearized value of the variable. For example, suppose that a transducer outputs a voltage related to pressure by

$$V = K[p]^{1/2} \quad (3.32)$$

This voltage is converted to a binary number, DV , by an ADC. Then it is also true that the binary number and pressure are still related by the square root:

$$DV \text{ varies as } [p]^{1/2} \quad (3.33)$$

What we want is a binary number that is linearly related to pressure. The way to get this is to square DV :

$$DP = DV * DV \text{ varies as } p \quad (3.34)$$

Thus, the program would input a sample DV and multiply it by itself. The resulting number would be linearly related to the pressure. Of course, there may have to be scale shifts and offsets before we have a number equal to the pressure, as the following example shows.

EXAMPLE 3.29 Pressure from 50 to 400 psi is converted to voltage by the relation

$$V = 0.385[p]^{1/2} - 2.722$$

This is input to an ADC with a 5.0-V reference, which provides 00H to FFH over the pressure range. A program uses an instruction $DV = UDF(1)$ to input the data from the ADC as a base 10 number DV that varies from 0 to 255 over the pressure range. Develop a linearization equation to give a quantity, p , in the program that is equal to the actual pressure.

Solution

We have enough information to work backward through the ADC, signal conditioning, and measurement. Thus, we know the voltage is related to DV by the ADC transformation

$$DV = (V/V_{ref})256 = (V/5)256$$

or

$$V = (5/256)DV$$

Then, using the known relation for V in terms of p ,

$$0.385[p]^{1/2} - 2.722 = (5/256)DV$$

and solving for p ,

$$p = (0.0507 DV + 7.071)^2$$

This number p in the program is equal to the actual pressure value, 50 to 400 psi.

Linearization by Table Look-up There are many measurement processes where it is impossible to find a simple equation such as Equation (3.32) to relate the controlled variable and the binary number. Also, when the program must be written in assembly language, it may be difficult to evaluate even simple equations such as that of Example 3.29. In these cases, it becomes much easier to use the look-up table approach.

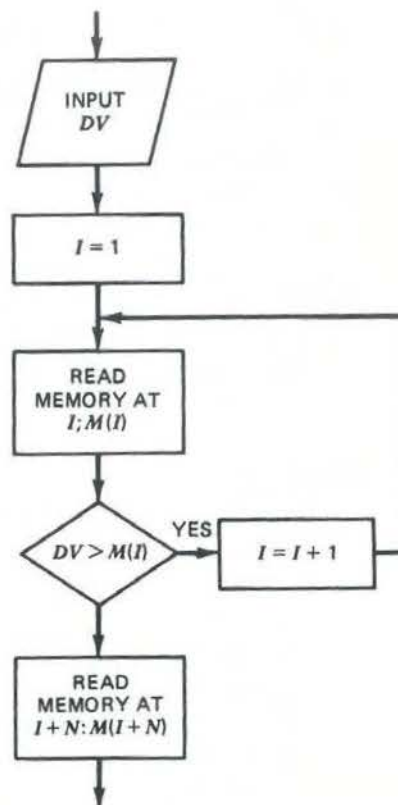
This is really just what we humans do when we use thermocouple tables, for example. We measure the voltage, go to a table and look up the temperature, and sometimes interpolate. It is the exact same thing with the software approach. The table of input values and corresponding physical variable values are stored in a table in memory. Following a measurement, the input value is looked up in the table and the correct measured value found.

Figure 3.35 shows a software approach for table look-up in flowchart format. It is assumed that the input values are stored in ascending value in N memory locations and the corresponding physical variable values are stored in the following N locations. Thus, if the input is found at the I th memory location from the start of the table, then the actual variable value is found at the $N + I$ location.

Of course, there are many other methods of table construction and search. In many cases, it is necessary to write interpolation routines by programming equations such as Equation (4.14) to refine the values between table values.

FIGURE 3.35

Linearization by table look-up can be accomplished by the operations in this flowchart.



SUMMARY

This chapter provides a digital electronics background to make the reader conversant with the elements of digital signal conditioning and able to perform simple analysis and design as associated with process control.

1. The use of digital words enables the encoding of analog information into a digital format.
2. It is possible to encode fractional decimal numbers as binary, and vice versa, using

$$N_{10} = b_1 2^{-1} + b_2 2^{-2} + \cdots + b_m 2^{-m} \quad (3.1)$$

3. Boolean algebraic techniques can be applied to the development of process alarms and elementary control functions.
4. Digital electronic gates and comparators allow the implementation of process Boolean equations.
5. DACs are used to convert digital words into analog numbers using a fractional-number representative. The resolution is

$$\Delta V = V_R 2^{-n} \quad (3.8)$$

6. An ADC of the successive approximations type determines an output digital word for an input analog voltage in as many steps as bits to the word.

7. The dual-slope ADC converts analog to digital information by a combination of integration and time counting.
8. The data-acquisition system (DAS) is a modular device that interfaces many analog signals to a computer. Signal address decoding, multiplexing, and ADC operations are included in the device.
9. The sampling rate of a signal must be high enough to assure the signal can be reconstructed from the samples. Generally we must sample about 10 times the maximum signal frequency.
10. One of the great advantages of digitizing data and feeding it into a computer is that nonlinearities can then be removed by software. This is done by either an equation or by a table look-up process.

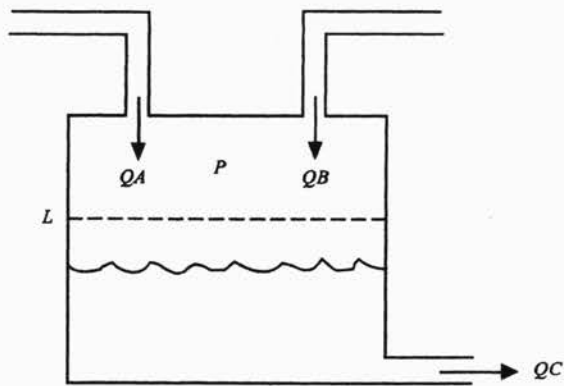
PROBLEMS

Section 3.2

- 3.1 Convert the following binary numbers into decimal, octal, and hex:
 - a. 1010_2
 - b. 111011_2
 - c. 010110_2
- 3.2 Convert the following binary numbers into decimal, octal, and hex:
 - a. 1011010_2
 - b. 0.1101_2
 - c. 1011.0110_2
- 3.3 Convert the following decimal numbers into binary, octal, and hex:
 - a. 21_{10}
 - b. 630_{10}
 - c. 427_{10}
- 3.4 Convert 27.156_{10} into a binary number with the fractional binary part expressed in 6 bits. What actual decimal does this binary fraction equal?
- 3.5 Find the 2s complement of
 - a. 1011_2
 - b. 10101100_2
- 3.6 Prove by a table of values that $\overline{A \cdot B} = \overline{A} + \overline{B}$ (DeMorgan's theorem).
- 3.7 Show that the Boolean equation $A \cdot B + A \cdot \overline{A \cdot B}$ reduces to A .
- 3.8 A process involves moving speed, load weight, and rate of loading in a conveyor system. The variables are provided as high (1) and low (0) levels for digital control. An alarm should be initiated whenever any of the following occur:
 - a. Speed is low; both weight and loading rate are high.
 - b. Speed is high; loading rate is low.
 Find a Boolean equation describing the required alarm output. Let the variables be S for speed, W for weight, and R for loading rate.
- 3.9 Implement Problem 3.8 with
 - a. AND/OR logic and
 - b. NAND/NOR logic

FIGURE 3.36

System for Problems 3.10 and 3.14.



- 3.10** A tank shown in Figure 3.36 has the following Boolean variables: flow rates, Q_A , Q_B , and Q_C ; pressure, P ; and level, L . All are high if the variable is high and low otherwise. Devise Boolean equations for two alarm conditions as follows:
- OV = overfill alarm**
 - If either input flow rate is high while the output flow rate is low, the pressure is low and the level is high.
 - [If both input flow rates are high while the output flow rate is low and the pressure is low.]
 - EP = empty alarm**
 - If both input flow rates are low, the level is low and the output flow rate is high.
 - If either input flow rate is low, the output flow rate is high and the pressure is high.
- 3.11** Devise logic circuits using NAND/NOR logic that will provide the two alarms of Problem 3.10.

Section 3.3

- 3.12** A sensor provides temperature data as $360 \mu\text{V}/^\circ\text{C}$. Develop a comparator circuit that goes high when the temperature reaches 530°C .
- 3.13** A light level is to trigger a comparator high (5 V) when the intensity reaches $30 \text{ W}/\text{m}^2$. Intensity is converted to voltage according to a transfer function of $0.04 \text{ V}/(\text{W}/\text{m}^2)$. Noise is found to contribute $\pm 1.6 \text{ W}/\text{m}^2$ of intensity fluctuations. Develop a hysteresis comparator to provide the required output and immunity for noise.
- 3.14** Sensors are available for the system shown in Figure 3.36 that provide voltage outputs with the following transfer functions:

$$\text{Flow rate: } V_Q = 0.15\sqrt{Q} \quad (Q \text{ in gal/min})$$

$$\text{Pressure: } V_P = \frac{20}{20 + p} \quad (p \text{ in psi})$$

$$\text{Level: } V_L = 0.05L \quad (L \text{ in meters})$$

The critical values for the high/low conditions are $QA = 55$ gal/min, $QB = 30$ gal/min, $QC = 100$ gal/min, $L = 3.6$ m, and $P = 120$ psi. Design comparator circuits to provide the Boolean variable values. Include hysteresis of 1% of the reference.

- 3.15** Design hardware for a liquid delivery system alarm that provides a binary high output when the system temperature is between 40° and 50°C and flow is between 2.0 and 3.0 L/min. Sensors to be used have transfer functions of

$$R_T = 1000e^{-.05(T-25)} \text{ with } R_T \text{ in } \Omega \text{ and } T \text{ in } ^\circ\text{C}$$

$$V_Q = \frac{5}{Q + 5} \text{ with } V_Q \text{ in V and } Q \text{ in L/min}$$

- 3.16** A 6-bit DAC has an input of 100101_2 and uses a 10.0-V reference.
- Find the output voltage produced.
 - Specify the conversion resolution.
- 3.17** A 4-bit DAC must have an 8.00-V output when all inputs are high. Find the required reference.
- 3.18** An 8-bit DAC with a 5.00-V reference connects to a light source with an intensity given by $I_L = 45V^{(3/2)}\text{W/m}^2$.
- What is the range of intensity that can be produced?
 - What intensities are produced by digital inputs of 1BH, 7AH, 9FH, and E5H?
 - Plot the intensity versus hex input, and comment on linearity.
- 3.19** A 12-bit bipolar DAC has a 10.00-V reference.
- What output voltage results from digital inputs of 4A6H, 02BH, and D5DH?
 - An output of 4.740 V is needed. What digital input would come closest to this value? By what percentage is the actual output different?
 - Suppose the output picks up a high-frequency noise of 50 mV rms. How many output bits are obscured by this noise?
- 3.20** A computer will be used to generate a ramp voltage signal in time (i.e., a triangular wave like Figure 3.19), with the following specifications: (1) -5 to $+5$ V, (2) period of 2.5 ms, (3) step size of 10 mV minimum.
- Specify the requirements of the DAC.
 - Prepare flowcharts of the software that will generate the wave.
- 3.21** An 8-bit ADC with a 10.0-V reference has an input of 3.797 V. Find the digital output word. What range of input voltages would produce this same output? Suppose the output of the ADC is 10110111_2 . What is the input voltage?
- 3.22** An ADC that will encode pressure data is required. The input signal is 666.6 mV/psi.
- If a resolution of 0.5 psi is required, find the number of bits necessary for the ADC. The reference is 10.0 V.
 - Find the maximum measurable pressure.
- 3.23** A bipolar ADC has 10 bits and a 10.00-V reference. What output is produced by inputs of -4.3 V, -0.66 V, $+2.4$ V, and $+4.8$ V? What is the input voltage if the output is 30BH?

- 3.24** A 10-bit, unipolar ADC with a reference of 5.00 V and a 44- μ s conversion time will be used to collect data on time constant measurement. Thus, the input will be of the form

$$V(t) = 4(1 - e^{t/\tau})$$

What is the minimum value of τ for which reliable data samples can be taken if no S/H circuit is used?

- 3.25** An 8-bit, 20- μ s bipolar ADC with a 5.00-V reference will monitor a sinusoidal signal with a 3.00-V peak amplitude. What is the maximum frequency that can be tracked to 8-bit accuracy?
- 3.26** A sample-and-hold circuit like the one shown in Figure 3.22 has $C = 0.47 \mu\text{F}$, and the ON resistance of the FET is 75Ω . For what signal frequency is the sampling capacitor voltage down 3 dB from the signal voltage? How does this limit the application of the sample hold?
- 3.27** A S/H and 12-bit bipolar ADC combination has the following characteristics: ADC conversion time = 35 μ s, S/H aperture time = 0.8 μ s, acquisition time = 5 μ s, ADC reference = 5.00 V, S/H ON resistance = 75Ω , OFF resistance = 20 M Ω , and source resistance = 1 Ω . A 0.0075- μ F capacitor is used for the S/H function, and the voltage follower has an input resistance of 7.5 M Ω . Determine:
- the sampling cut-off frequency.
 - the effect of droop on the conversion process.
 - the effect of the S/H ON resistance on the sampling process.
- 3.28** A S/H and ADC combination has a throughput expressed as 100,000 samples per second. Explain the consequences of using this system to take samples every 5 ms.
- 3.29** A sensor signal is converted to a frequency that varies from 4.6 to 37 kHz. This is to be used with a counter-based ADC of 10 bits. Specify the count time, T_c . What is the count for the minimum frequency of 4.6 kHz?
- 3.30** A capacitor varies linearly from 0.04 to 0.26 μ F as pressure varies from 0 to 100 kPa gauge. Devise a circuit using an LM331 that converts this to a frequency with a maximum of 20 kHz at 0.04 μ F. Then determine the appropriate T_c of a counter-based ADC that will provide an 8-bit output of 255 at 20 kHz. What is the frequency at 0.26 μ F? Plot the frequency versus pressure. Comment on the linearity.

Section 3.4

- 3.31** A DAS has an ADC of 8 bits with a 0- to 2.5-V range of input for 00H to FFH output (i.e., FEH to FFH occurs at 2.50 V). Inputs are temperature from 20° to 100°C scaled at 40 mV/°C, pressure from 1 to 100 psi scaled at 100 mV/psi, and flow from 30 to 90 gal/min scaled at 150 mV/(gal/min). Develop a block diagram of the required signal conditioning so that each of these can be connected to the DAS and so that the indicated variable range corresponds to 00H to FFH. Specify the resolution of each in terms of the change in each variable that corresponds to an LSB change in the ADC output.
- 3.32** Design op amp circuits that will provide the signal conditioning specified in Problem 3.31.

- 3.33** A data-acquisition system has eight input channels to be sampled continuously and sequentially. The multiplexer can select and settle on a channel in $3.1 \mu\text{s}$, the ADC converts in $33 \mu\text{s}$, and the computer processes a single channel of data in $450 \mu\text{s}$. What is the minimum time between samples for a particular channel?
- 3.34** Flow is to be measured and input to a computer. For maximum resolution, we want the minimum flow, $30 \text{ m}^3/\text{h}$, to correspond to 00H, and the maximum, $60 \text{ m}^3/\text{h}$, to correspond to FFH from an 8-bit converter. The flow is measured as a voltage given by $V = 0.0022Q^2$, where Q is the flow in m^3/h . Develop the signal conditioning and ADC reference that will provide this specification. What is the resolution of flow when the flow is at the lower limit and at the upper limit? Explain why there is a difference.
- 3.35** An 8-bit ADC has an 8.00-V reference.
- Find the output for inputs of 3.4 V and 6.7 V.
 - What range of inputs could have caused the output to become B7H?
- 3.36** Tests show that the output of a position sensor is 12 mV/mm, but there is 60-Hz noise on the output of a constant 5-mV rms. The sensor output impedance is $2.5 \text{ k}\Omega$. This sensor is to measure work-piece motion, which oscillates between -10 mm and $+10 \text{ mm}$ with a period of 1.5 s. The position is to be interfaced to a 12-bit bipolar, offset binary ADC with a 5.000-V reference. Design the interface system such that -10 mm corresponds to 000H and $+10 \text{ mm}$ corresponds to FFFH.
- With no filter, determine how many bits are being toggled by noise (i.e., are lost to any real data).
 - Introduce a filter, reevaluate the effect of noise on the ADC output, and determine how many bits represent real data.
- 3.37** A sensor linearly changes resistance from 2.35 to $3.57 \text{ k}\Omega$ over a range of some measured variable. The measurement must have a resolution of at least 1.25Ω and be interfaced to a computer. Design the signal conditioning and specify the characteristics of the required ADC.
- 3.38** Using the DAS of Example 3.25, prepare a flowchart of a system that inputs a sample from channel 5, decrements it by one, and outputs to the DAC. Use port 300H for the base address.
- 3.39** What is the maximum rate that data can be taken from one channel of the DAS of Example 3.25, apart from software time? What is the maximum rate that data can be taken if all eight channels are sequentially sampled?
- 3.40** Prepare a flowchart of software that inputs data from channel 2 and then channel 6. These are multiplied, and the higher 8 bits of the 16-bit product are sent to the DAC for output. Develop an equation that relates the analog output voltage to the two analog input voltages.

Section 3.5

- 3.41** A digital control system is to provide regulation of pressure within 1.2 kPa in the range 30 to 780 kPa . How many bits must be used for the data acquisition?
- 3.42** A computer will be used to control flow through 10 pumping stations. The pumps exhibit a surging effect with a period of 2.2 s . What is the minimum sampling rate to ensure quality data? How much time can be spent processing each station's data? Data-acquisition hardware and software take 200μ for a channel.

- 3.43** The output voltage of a silicon photovoltaic cell is given by $V = 0.11 \log_e(I_L)$, where I_L is the light intensity in W/m^2 . Intensity is to be measured from 100 to 400 W/m^2 , and input to a computer is via an 8-bit ADC with a 5.00-V reference.
- Develop signal conditioning to interface the cell to the ADC so that 00H to 01H occurs at 100 W/m^2 and FEH to FFH occurs at 400 W/m^2 .
 - Develop software in the language of your choice to linearize the input data so that a program variable is the intensity in W/m^2 .
- 3.44** A type-K TC with a 20°C reference will be used to measure temperature from 200°C to 280°C in a computer controller application. The data-acquisition hardware and software make the actual measured voltage, in mV, available in software as a variable VTC. Devise a flowchart and/or program in the language of your choice that makes the reference correction and performs linearization by table look-up and interpolation. Use table values in 5°C increments for a type-K TC with a 0°C reference.
- 3.45** A gauge is used in a bridge circuit as in Figure 2.5 with $R_1 = R_2 = R_3 = 120 \Omega$, $V = 10.0 \text{ V}$. Assume the bridge offset voltage is provided as input to a computer and appears in a variable DELTAV. Prepare a flowchart and/or program that will obtain the value of R_4 from this voltage.

SUPPLEMENTARY PROBLEMS

- S3.1** A monitoring station has a flow sensor that outputs a voltage proportional to the square root of high-pressure gas-flow rate, as given by $V_Q = 0.45 \sqrt{(Q + 9)}$ volts with the flow rate in kg/h. Normal flow is $4.0 \pm 1.0 \text{ kg/h}$, and the flow typically surges periodically within these limits. There are two alarm conditions: (1) if the flow surges above and below 6 kg/h twice without dropping below 4 kg/h in between, and (2) if a surge above 5.1 kg/h is followed by a drop below 2.9 kg/h within 2 s. Devise a system of comparators, F/F_s s, one-shot multivibrators, and any other needed circuitry to provide the alarms.
- S3.2** A pressure sensor outputs 2.5 mV/kPa , but there is 60-Hz noise of a steady peak 55 mV on the line. Pressure in the range 0.0 to 70 kPa must be converted by an ADC and presented to a computer. The resolution must be at least 0.2 kPa.
- Design a system of filter, signal-conditioning, and ADC specifications to satisfy this requirement.
 - Suppose the pressure suddenly changed from 20 to 50 kPa. How long would it be before the ADC presented the correct digital signal for this pressure to the computer?
- S3.3** A system is presented in Figure 3.37 to provide computer temperature control. The heater requires a voltage from 0.0 to 5.0 V for off (0%) to full on (100%). The following algorithm will be used by the computer to determine the heater setting:
- If $T < 14^\circ\text{C}$: heater at 100%
 - If $14^\circ\text{C} < T < 19^\circ\text{C}$: heater at 70%
 - If $19^\circ\text{C} < T < 22^\circ\text{C}$: heater at 38%
 - If $T > 22^\circ\text{C}$: heater at 0%

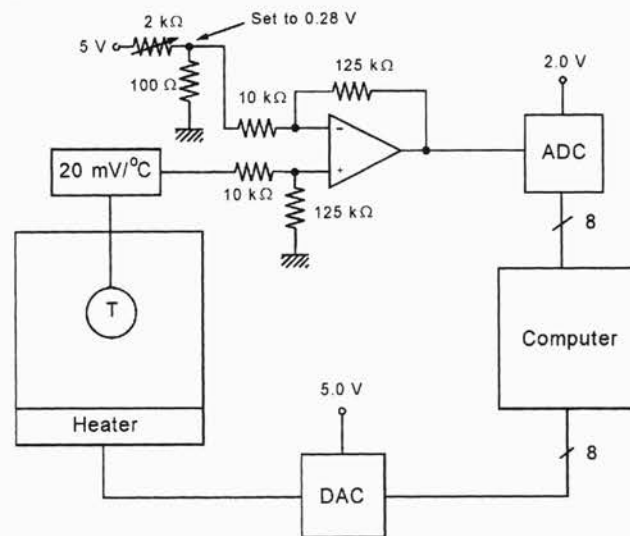


FIGURE 3.37
System for Problem S3.3.

- Determine the ADC output for each critical temperature.
- Determine the necessary heater voltages and the proper DAC input to produce those voltages.
- Prepare a flowchart showing how a computer program would satisfy the algorithm requirements.