

Bangladesh Army University of Engineering & Technology (BAUET)
Department of Computer Science and Engineering

Interfacing with the Analog World

Prof. Dr. Md. Rabiul Islam
Dept. of Computer Science & Engineering
Rajshahi University of Engineering & Technology
Rajshahi-6204, Bangladesh.
rabiul_cse@yahoo.com

Outline

Overview of Interfacing with the Analog World

Digital to Analog Conversion

- Process of Digital to Analog Conversion
- Resolution or Step Size
- Percentage Resolution
- D/A Conversion Circuitry
- Mathematical Examples

Analog to Digital Conversion

- Process of Analog to Digital Conversion
- Digital Ram ADC
- Implementation of DAC and ADC
 - ✓ Data Acquisition
 - ✓ Reconstructing Digital Signal
 - ✓ Aliasing
- Successive Approximation ADC
 - ✓ Overview of successive Approximation ADC
 - ✓ Example of 4bit Successive Approximation ADC
- IC ADC0804 Successive Approximation ADC
 - ✓ Pin Diagram
 - ✓ Important Characteristics
 - ✓ Pin Function
- Flash ADC
- Up/Down Digital Ramp ADC
- Mathematical Examples

Application areas of DACs and ADCs

References

Chapter 10 Interfacing with the Analog World, Ronald J. Tocci and Neal S. Widmer, "Digital Systems Principles and Applications", Eighth Edition, Prentice Hall.

Overview of Interfacing with the Analog World

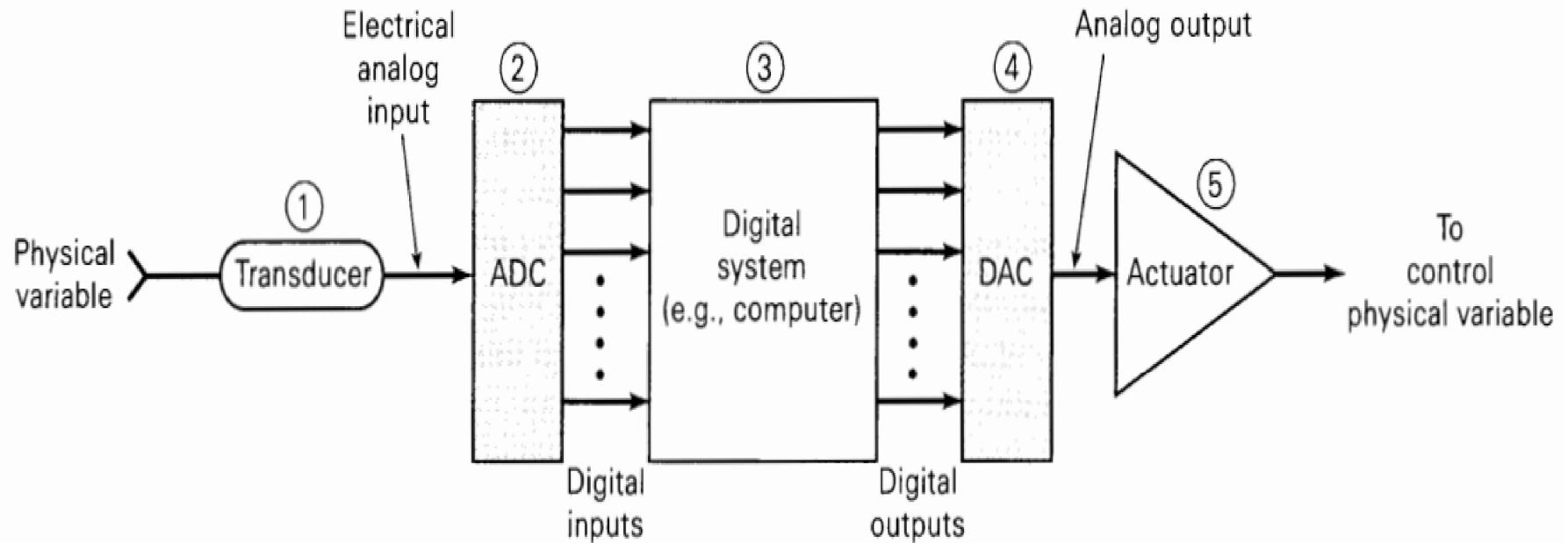


Fig. Analog to Digital Converter (ADC) and Digital to Analog Converter (DAC) is used to interface a computer with the real world so that the computer can monitor and control the physical variable.

Overview of Interfacing with the Analog World

Analog-to-digital converter (ADC). The transducer's electrical analog output serves as the analog input to the ADC. The ADC converts this analog input to a digital output. This digital output consists of a number of bits that represent the value of the analog input. For example, the ADC might convert the transducer's 800- to 1500-mV analog values to binary values ranging from 01010000 (80) to 10010110 (150). Note that the binary output from the ADC is proportional to the analog input voltage so that each unit of the digital output represents 10 mV.

Digital-to-analog converter (DAC). This digital output from the computer is connected to a DAC, which converts it to a proportional analog voltage or current. For example, the computer might produce a digital output ranging from 00000000 to 11111111, which the DAC converts to a voltage ranging from 0 to 10 V.

Digital to Analog Conversion

Process of Digital to Analog Conversion:

Basically, D/A *conversion* is the process of taking a value represented in *digital* code (such as straight binary or BCD) and converting it to a voltage or current that is proportional to the digital value.

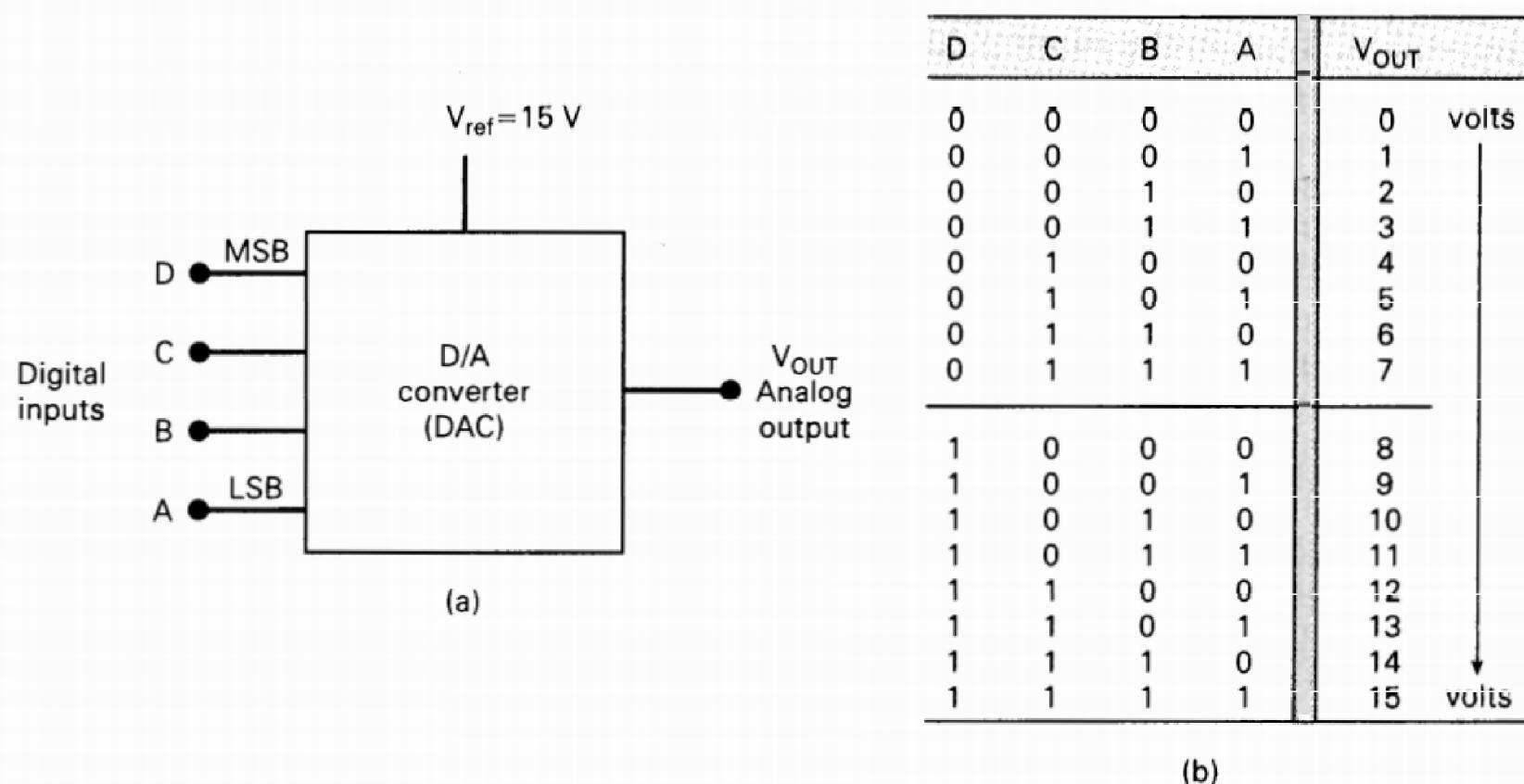


FIGURE Four-bit DAC with voltage output.

Digital to Analog Conversion

Notice that there is an input for a voltage reference, V_{ref} . This input is used to determine the **full-scale output** or maximum value that the D/A converter can produce. The digital inputs D , C , B , and A are usually derived from the output register of a digital system. The $2^4 = 16$ different binary numbers represented by these four bits are listed in Figure 10-2(b). For each input number, the D/A converter output voltage is a unique value. In fact, for this case, the analog output voltage V_{OUT} is equal in volts to the binary number. It could also have been twice the binary number or some other proportionality factor. The same idea would hold true if the D/A output were a current I_{OUT} .

In general,

$$\text{analog output} = K \times \text{digital input}$$

Digital to Analog Conversion

where K is the proportionality factor and is a constant value for a given DAC connected to a fixed reference voltage. The analog output can, of course, be a voltage or a current. When it is a voltage, K will be in voltage units, and when the output is a current, K will be in current units. For the DAC of Figure , $K = 1 \text{ V}$, so that

$$V_{\text{OUT}} = (1 \text{ V}) \times \text{digital input}$$

We can use this to calculate V_{OUT} for any value of digital input. For example, with a digital input of $1100_2 = 12_{10}$, we obtain

$$V_{\text{OUT}} = 1 \text{ V} \times 12 = 12 \text{ V}$$

Digital to Analog Conversion

Example 1:

A five-bit DAC has a current output. For a digital input of 10100, an output current of 10 mA is produced. What will I_{OUT} be for a digital input of 11101?

Solution

The digital input 10100_2 is equal to decimal 20. Since $I_{\text{OUT}} = 10 \text{ mA}$ for this case, the proportionality factor must be 0.5 mA. Thus, we can find I_{OUT} for any digital input such as $11101_2 = 29_{10}$ as follows:

$$\begin{aligned}I_{\text{OUT}} &= (0.5 \text{ mA}) \times 29 \\&= 14.5 \text{ mA}\end{aligned}$$

Remember, the proportionality factor, K , varies from one DAC to another and depends on the reference voltage.

Digital to Analog Conversion

Example 2:

What is the largest value of output voltage from an eight-bit DAC that produces 1.0 V for a digital input of 00110010?

Solution

$$\begin{aligned}00110010_2 &= 50_{10} \\1.0 \text{ V} &= K \times 50\end{aligned}$$

Therefore,

$$K = 20 \text{ mV}$$

The largest output will occur for an input of $11111111_2 = 255_{10}$.

$$\begin{aligned}V_{\text{OUT}}(\text{max}) &= 20 \text{ mV} \times 255 \\&= 5.10 \text{ V}\end{aligned}$$

Digital to Analog Conversion

Resolution/Step Size:

Resolution of a D/A converter is defined as the smallest change that can occur in the analog output as a result of a change in the digital input. Referring to the table in Figure _____, we can see that the resolution is 1 V, since V_{OUT} can change by no less than 1 V when the digital input value is changed. The resolution is always equal to the weight of the LSB and is also referred to as the **step size**, since it is the amount that V_{OUT} will change as the digital input value is changed from one step to the next. This is illustrated better in Figure _____, where the outputs from a four-bit binary counter provide the inputs to our DAC. As the counter is being continually cycled through its 16 states by the clock signal, the DAC output is a **staircase** waveform that goes up 1 V per step. When the counter is at 1111, the DAC output is at its maximum value of 15 V; this is its **full-scale output**. When the counter recycles to 0000, the DAC output returns to 0 V. The resolution or step size is the size of the jumps in the staircase waveform; in this case, each step is 1 V.

Digital to Analog Conversion

Resolution/Step Size:

Note that the staircase has 16 levels corresponding to the 16 input states, but there are only 15 steps or jumps between the 0-V level and full-scale. In general, for an N -bit DAC the number of different levels will be 2^N , and the number of steps will be $2^N - 1$.

You may have already figured out that resolution (step size) is the same as the proportionality factor in the DAC input/output relationship:

$$\text{analog output} = K \times \text{digital input}$$

A new interpretation of this expression would be that the digital input is equal to the number of steps, K is the amount of voltage (or current) per step, and the analog output is the product of the two. We now have a convenient way of calculating the value of K for the D/A:

$$\text{resolution} = K = \frac{A_{fs}}{(2^n - 1)}$$

where A_{fs} is the analog full-scale output and n is the number of bits.

Digital to Analog Conversion

Resolution/Step Size:

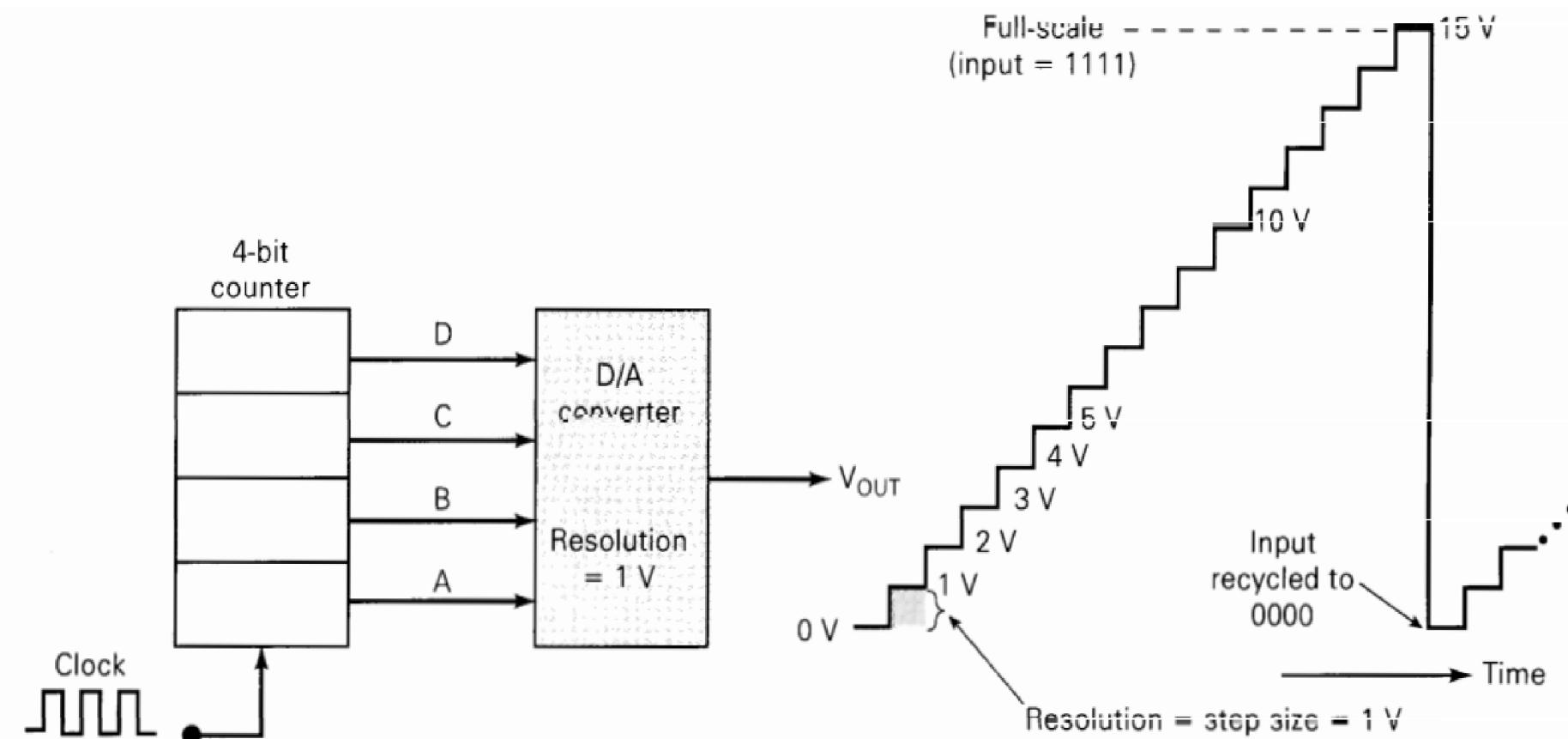


FIGURE Output waveforms of a DAC as inputs are provided by a binary counter.

Digital to Analog Conversion

Percentage Resolution:

Although resolution can be expressed as the amount of voltage or current per step, it is also useful to express it as a percentage of the *full-scale output*. To illustrate, the DAC of Figure 10-3 has a maximum full-scale output of 15 V (when the digital input is 1111). The step size is 1 V, which gives a percentage resolution of

$$\begin{aligned}\% \text{ resolution} &= \frac{\text{step size}}{\text{full scale (F.S.)}} \times 100\% \\ &= \frac{1 \text{ V}}{15 \text{ V}} \times 100\% = 6.67\%\end{aligned}$$

Digital to Analog Conversion

Example 1:

A 10-bit DAC has a step size of 10 mV. Determine the full-scale output voltage and the percentage resolution.

Solution

With 10 bits, there will be $2^{10} - 1 = 1023$ steps of 10 mV each. The full-scale output will therefore be $10 \text{ mV} \times 1023 = 10.23 \text{ V}$, and

$$\% \text{ resolution} = \frac{10 \text{ mV}}{10.23 \text{ V}} \times 100\% \approx 0.1\%$$

Digital to Analog Conversion

Example helps to illustrate the fact that the percentage resolution becomes smaller as the number of input bits is increased. In fact, the percentage resolution can also be calculated from

$$\% \text{ resolution} = \frac{1}{\text{total number of steps}} \times 100\%$$

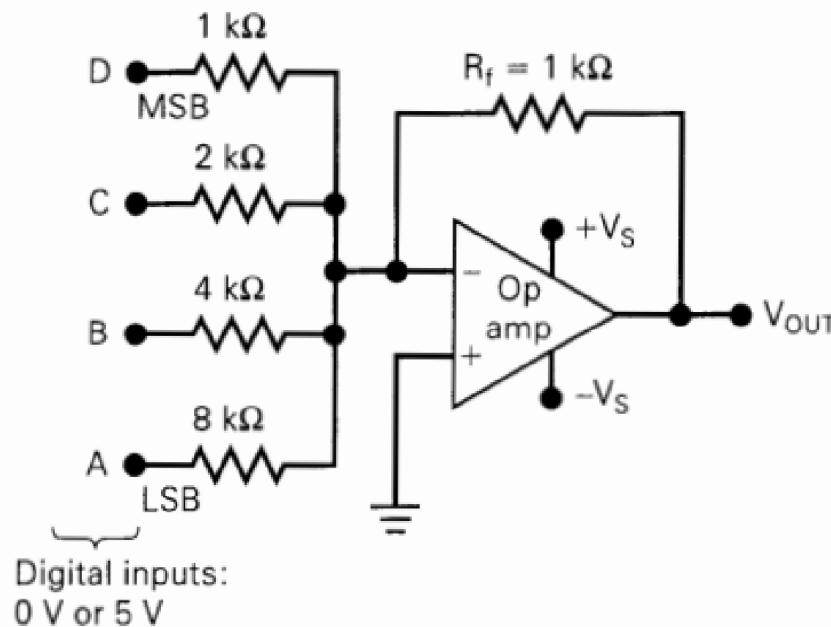
For an N -bit binary input code the total number of steps is $2^N - 1$. Thus, for the previous example,

$$\begin{aligned}\% \text{ resolution} &= \frac{1}{2^{10} - 1} \times 100\% \\ &= \frac{1}{1023} \times 100\% \\ &\approx 0.1\%\end{aligned}$$

This means that it is *only the number of bits* that determines the *percentage* resolution. Increasing the number of bits increases the number of steps to reach full scale, so that each step is a smaller part of the full-scale voltage. Most DAC manufacturers specify resolution as the number of bits.

Digital to Analog Conversion

D/A Conversion Circuitry:



(a)

Input code				
D	C	B	A	V_{OUT} (volts)
0	0	0	0	0
0	0	0	1	-0.625 ← LSB
0	0	1	0	-1.250
0	0	1	1	-1.875
<hr/>				
0	1	0	0	-2.500
0	1	0	1	-3.125
0	1	1	0	-3.750
0	1	1	1	-4.375
<hr/>				
1	0	0	0	-5.000
1	0	0	1	-5.625
1	0	1	0	-6.250
1	0	1	1	-6.875
<hr/>				
1	1	0	0	-7.500
1	1	0	1	-8.125
1	1	1	0	-8.750
1	1	1	1	-9.375 ← Full-scale

(b)

FIGURE
resistors.

Simple DAC using an op-amp summing amplifier with binary-weighted

Digital to Analog Conversion

D/A Conversion Circuitry:

Figure shows the basic circuit for one type of four-bit DAC. The inputs A , B , C , and D are binary inputs that are assumed to have values of either 0 or 5 V. The *operational amplifier* is employed as a summing amplifier, which produces the weighted sum of these input voltages. It may be recalled that the summing amplifier multiplies each input voltage by the ratio of the feedback resistor R_F to the corresponding input resistor R_{IN} . In this circuit $R_F = 1 \text{ k}\Omega$, and the input resistors range from 1 to 8 $\text{k}\Omega$. The D input has $R_{IN} = 1 \text{ k}\Omega$, so the summing amplifier passes the voltage at D with no attenuation. The C input has $R_{IN} = 2 \text{ k}\Omega$, so that it will be attenuated by $\frac{1}{2}$. Similarly, the B input will be attenuated by $\frac{1}{4}$, and the A input by $\frac{1}{8}$. The amplifier output can thus be expressed as

$$V_{OUT} = -(V_D + \frac{1}{2}V_C + \frac{1}{4}V_B + \frac{1}{8}V_A)$$

The negative sign is present because the summing amplifier is a polarity-inverting amplifier, but it will not concern us here.

Digital to Analog Conversion

Clearly, the summing amplifier output is an analog voltage which represents a weighted sum of the digital inputs, as shown by the table in Figure . This table lists all of the possible input conditions and the resultant amplifier output voltage. The output is evaluated for any input condition by setting the appropriate inputs to either 0 or 5 V. For example, if the digital input is 1010, then $V_D = V_B = 5$ V and $V_C = V_A = 0$ V. Thus, using equation ,

$$\begin{aligned}V_{\text{OUT}} &= -(5 \text{ V} + 0 \text{ V} + \frac{1}{4} \times 5 \text{ V} + 0 \text{ V}) \\&= -6.25 \text{ V}\end{aligned}$$

The resolution of this D/A converter is equal to the weighting of the LSB, which is $\frac{1}{8} \times 5 \text{ V} = 0.625 \text{ V}$. As shown in the table, the analog output increases by 0.625 V as the binary input number advances one step.

Digital to Analog Conversion

D/A Conversion Circuitry:

Since the noninverting input of the op amp is grounded, the op amp will work day and night to hold the inverting input also at 0 V. Remember that the inverting input in this circuit is referred to as the summing point. When one of the switches is closed, a current will flow from -5 V (V_{REF}) through that resistor to the summing point. The op amp will pull the current on through the feedback resistor to produce a proportional output voltage. If you close switch D0, for example, a current of 0.05 mA will flow into the summing point. In order to pull this current through the feedback resistor, the op amp must put a voltage of $0.05\text{ mA} \times 10\text{ k}\Omega$ or 0.5 V on its output. If you also close switch D1, it will send another 0.1 mA into the summing point. In order to pull the sum of the currents through the feedback resistor, the op amp has to output a voltage of $0.15\text{ mA} \times 10\text{ k}\Omega$ or 1.5 V .

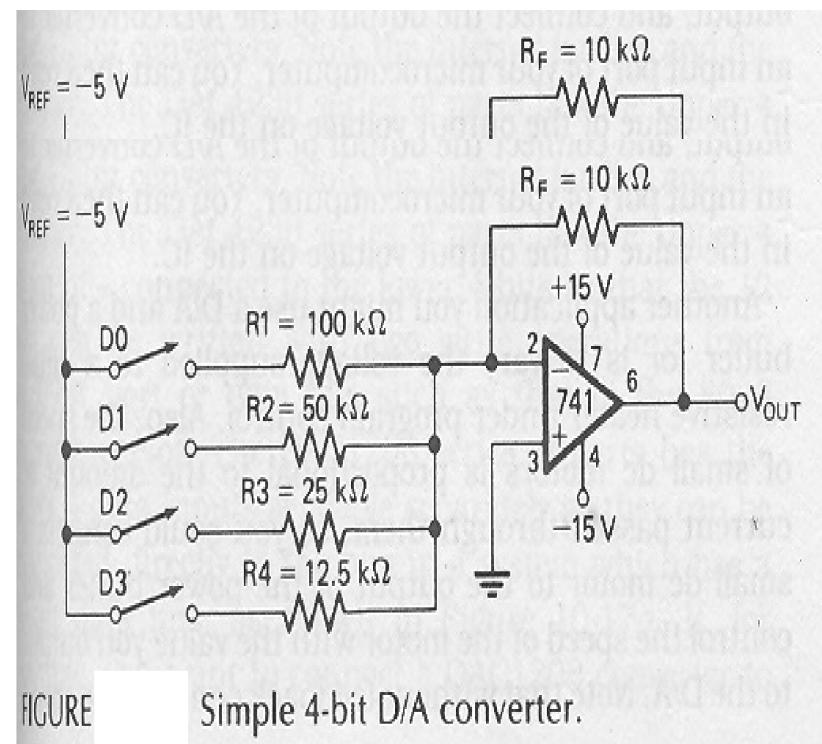


FIGURE Simple 4-bit D/A converter.

Digital to Analog Conversion

D/A Conversion Circuitry:

The point here is that the binary-weighted resistors produce binary-weighted currents which are summed by the op amp to produce a proportional output voltage. The binary word applied to the switches produces a proportional output voltage. Technically the output voltage is "digital" because it can only have certain fixed values, just as the display on a digital voltmeter can. However, the output simulates an analog signal, so we refer to it as analog. Switch D3 in Figure 10-14 represents the most significant bit because closing it produces the largest current. Note that since V_{REF} is negative, the output will go positive as switches are closed.

Digital to Analog Conversion

D/A Conversion Circuitry:

Example 3:

- (a) Determine the weight of each input bit of Figure mentioned above
- (b) Change R_F to 250Ω and determine the full-scale output.

Solution

- (a) The MSB passes with gain = 1, so its weight in the output is 5 V. Thus,

$$\begin{aligned}\text{MSB} &\rightarrow 5 \text{ V} \\ \text{2nd MSB} &\rightarrow 2.5 \text{ V} \\ \text{3rd MSB} &\rightarrow 1.25 \text{ V} \\ \text{4th MSB} = \text{LSB} &\rightarrow 0.625 \text{ V}\end{aligned}$$

- (b) If R_F is reduced by a factor of 4, to 250Ω , each input weight will be four times *smaller* than the values above. Thus, the full-scale output will be reduced by this same factor and becomes $-9.375/4 = -2.344 \text{ V}$.

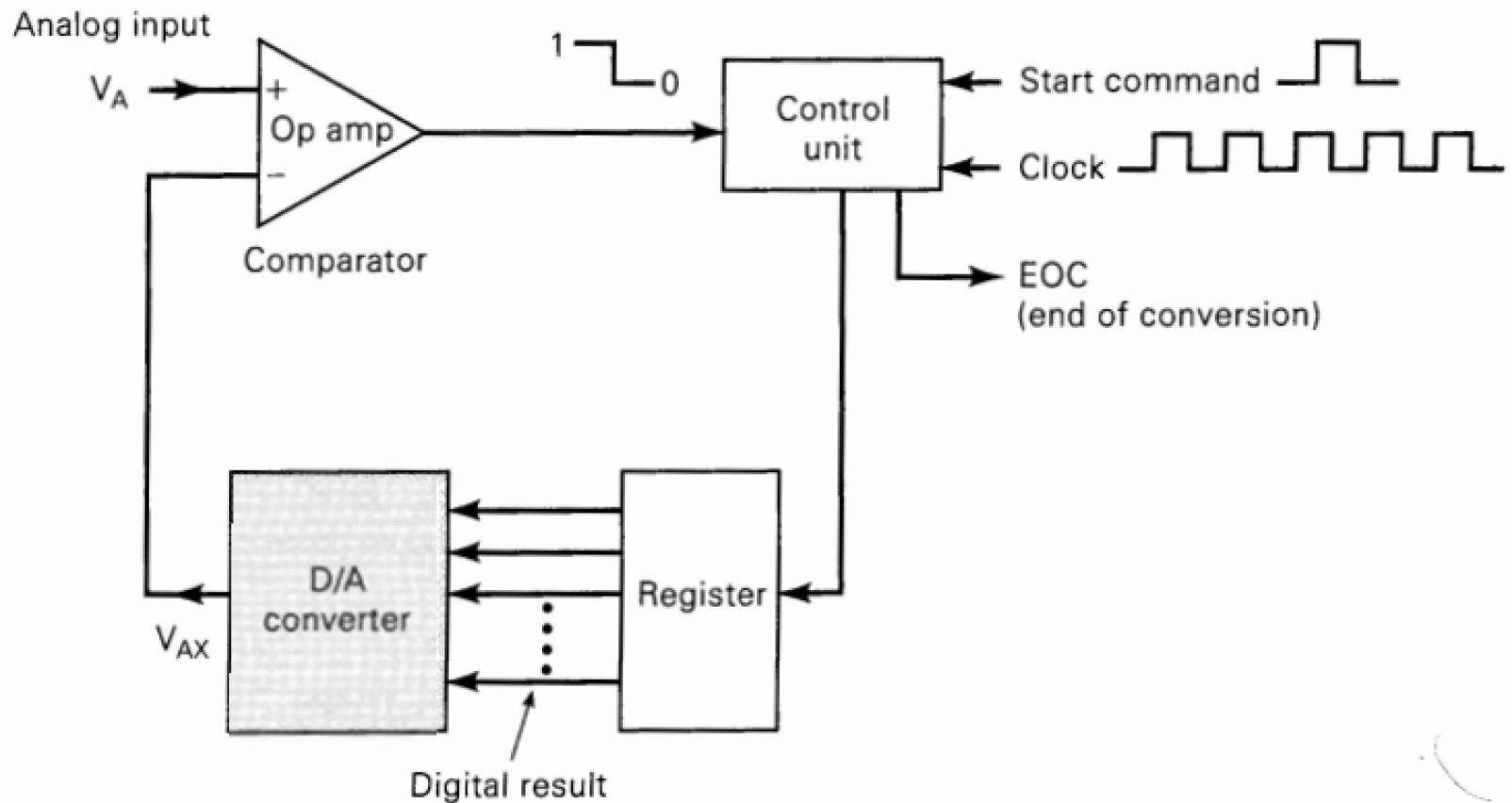
Analog to Digital Conversion

Process of Analog to Digital Conversion:

An **analog-to-digital converter** takes an analog input voltage and after a certain amount of time produces a digital output code that represents the analog input. The A/D conversion process is generally more complex and time-consuming than the D/A process, and many different methods have been developed and used.

Several important types of ADCs utilize a DAC as part of their circuitry. Figure is a general block diagram for this class of ADC. The timing for the operation is provided by the input clock signal. The control unit contains the logic circuitry for generating the proper sequence of operations in response to the START COMMAND, which initiates the conversion process. The op-amp comparator has two *analog* inputs and a *digital* output that switches states, depending on which analog input is greater.

Analog to Digital Conversion



FIGURE

General diagram of one class of ADCs.

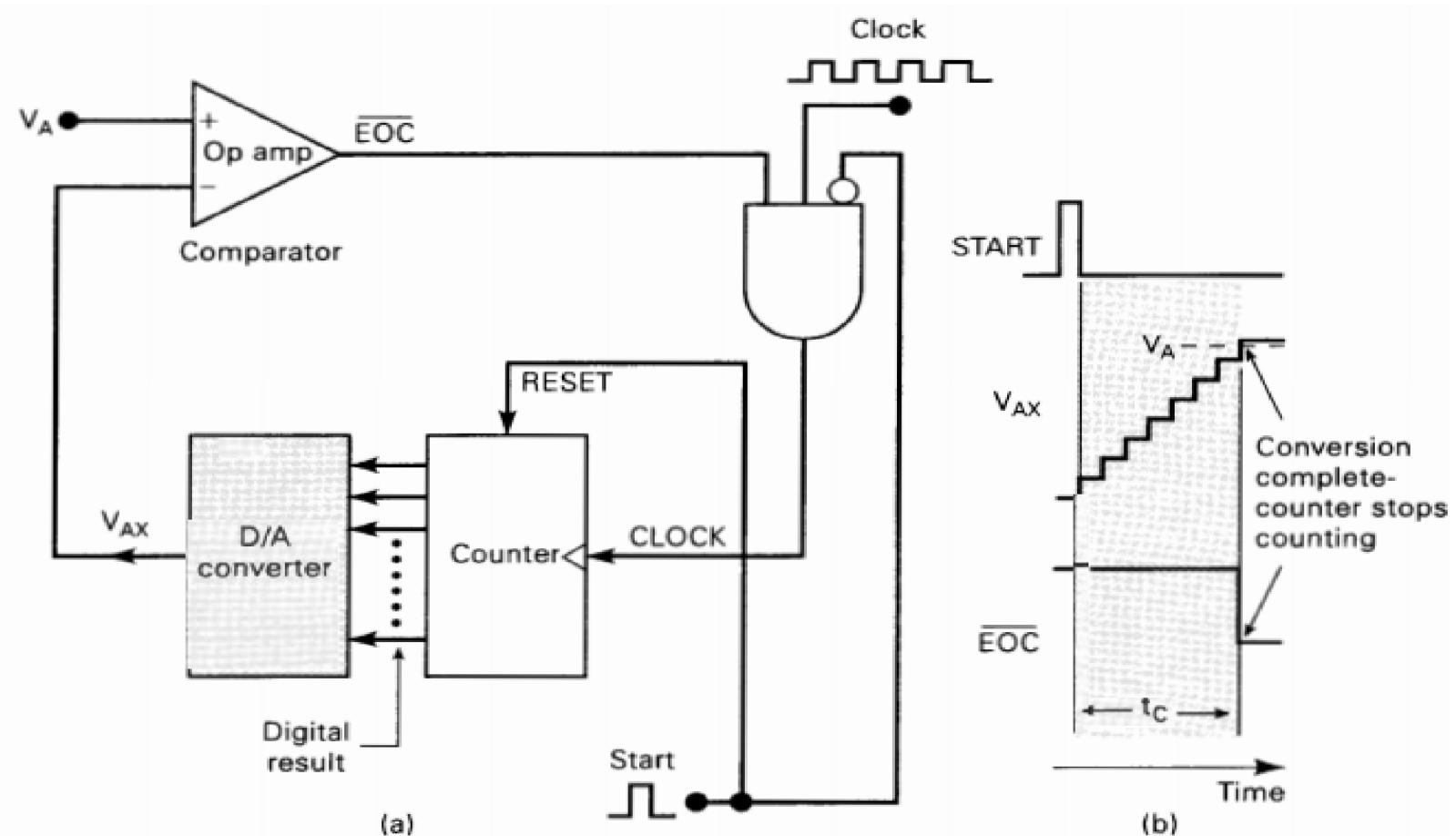
Analog to Digital Conversion

The basic operation of ADCs of this type consists of the following steps:

1. The START COMMAND pulse initiates the operation.
2. At a rate determined by the clock, the control unit continually modifies the binary number that is stored in the register.
3. The binary number in the register is converted to an analog voltage, V_{AX} , by the DAC.
4. The comparator compares V_{AX} with the analog input V_A . As long as $V_{AX} < V_A$, the comparator output stays HIGH. When V_{AX} exceeds V_A by at least an amount equal to V_T (threshold voltage), the comparator output goes LOW and stops the process of modifying the register number. At this point, V_{AX} is a close approximation to V_A . The digital number in the register, which is the digital equivalent of V_{AX} , is also the approximate digital equivalent of V_A , within the resolution and accuracy of the system.
5. The control logic activates the end-of-conversion signal, EOC , when the conversion is complete.

Analog to Digital Conversion

Digital Ramp ADC:



FIGURE

Digital-ramp ADC.

Analog to Digital Conversion

Digital Ramp ADC:

One of the simplest versions of the general ADC of Figure 1 uses a binary counter as the register and allows the clock to increment the counter one step at a time until $V_{AX} \geq V_A$. It is called a **digital-ramp ADC** because the waveform at V_{AX} is a step-by-step ramp (actually a staircase) like the one shown in Figure 2. It is also referred to as a *counter-type* ADC.

Figure 3 is the diagram for a digital-ramp ADC. It contains a counter, a DAC, an analog comparator, and a control AND gate. The comparator output serves as the active-LOW end-of-conversion signal \overline{EOC} . If we assume that V_A , the analog voltage to be converted, is positive, the operation proceeds as follows:

Analog to Digital Conversion

Digital Ramp ADC:

1. A START pulse is applied to reset the counter to 0. The HIGH at START also inhibits clock pulses from passing through the AND gate into the counter.
2. With all 0s at its input, the DAC's output will be $V_{AX} = 0 \text{ V}$.
3. Since $V_A > V_{AX}$, the comparator output, \overline{EOC} , will be HIGH.
4. When START returns LOW, the AND gate is enabled and clock pulses get through to the counter.
5. As the counter advances, the DAC output, V_{AX} , increases one step at a time as shown in Figure .
6. This continues until V_{AX} reaches a step that exceeds V_A by an amount equal to or greater than V_T (typically 10 to 100 μV). At this point, \overline{EOC} will go LOW and inhibit the flow of pulses into the counter, and the counter will stop counting.
7. The conversion process is now complete as signaled by the HIGH-to-LOW transition at \overline{EOC} , and the contents of the counter are the digital representation of V_A .
8. The counter will hold the digital value until the next START pulse initiates a new conversion.

Analog to Digital Conversion

Digital Ramp ADC:

Exercise:

Assume the following values for the ADC of Figure : clock frequency = 1 MHz; $V_T = 0.1$ mV; DAC has F.S. output = 10.23 V and a 10-bit input. Determine the following values.

- (a) The digital equivalent obtained for $V_A = 3.728$ V
- (b) The conversion time
- (c) The resolution of this converter

Solution

- (a) The DAC has a 10-bit input and a 10.23-V F.S. output. Thus, the number of total possible steps is $2^{10} - 1 = 1023$, and so the step size is

$$\frac{10.23 \text{ V}}{1023} = 10 \text{ mV}$$

This means that V_{AX} increases in steps of 10 mV as the counter counts up from 0. Since $V_A = 3.728$ V and $V_T = 0.1$ mV, V_{AX} must reach 3.7281 V or more before the comparator switches LOW. This will require

$$\frac{3.7281 \text{ V}}{10 \text{ mV}} = 372.81 = 373 \text{ steps}$$

At the end of the conversion, then, the counter will hold the binary equivalent of 373, which is 0101110101. This is the desired digital equivalent of $V_A = 3.728$ V, as produced by this ADC.

Analog to Digital Conversion

Digital Ramp ADC:

Exercise:

- (b) Three hundred seventy-three steps were required to complete the conversion. Thus, 373 clock pulses occurred at the rate of one per microsecond. This gives a total conversion time of $373 \mu\text{s}$.
- (c) The resolution of this converter is equal to the step size of the DAC, which is 10 mV. In percent it is $1/1023 \times 100\% \approx 0.1\%$.

Analog to Digital Conversion

Implementation of DAC and ADC:

Data Acquisition:

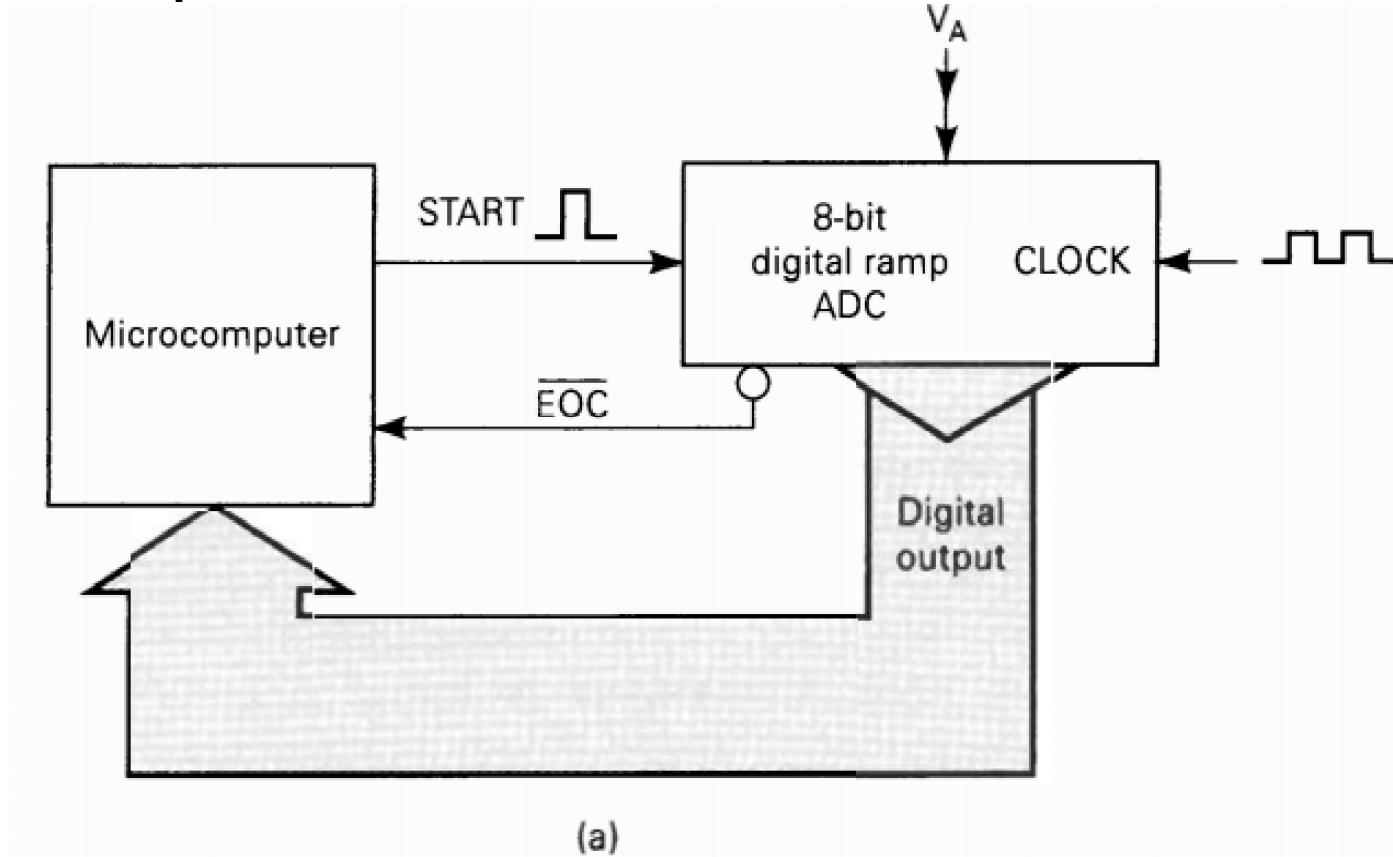
Sampling:

There are many applications in which analog data must be *digitized* (converted to digital) and transferred into a computer's memory. The process by which the computer acquires these digitized analog data is referred to as *data acquisition*. Acquiring a single data point's value is referred to as **sampling** the analog signal and that data point is often called a *sample*. The computer can do several different things with the data, depending on the application. In a storage application, such as digital audio recording, video recording, or a digital oscilloscope, the internal microcomputer will store the data and then transfer them to a DAC at a later time to reproduce the original analog signal. In a process control application, the computer can examine the data or perform computations on them to determine what control outputs to generate.

Analog to Digital Conversion

Implementation of DAC and ADC:

Data Acquisition:



Analog to Digital Conversion

Implementation of DAC and ADC:

Data Acquisition:

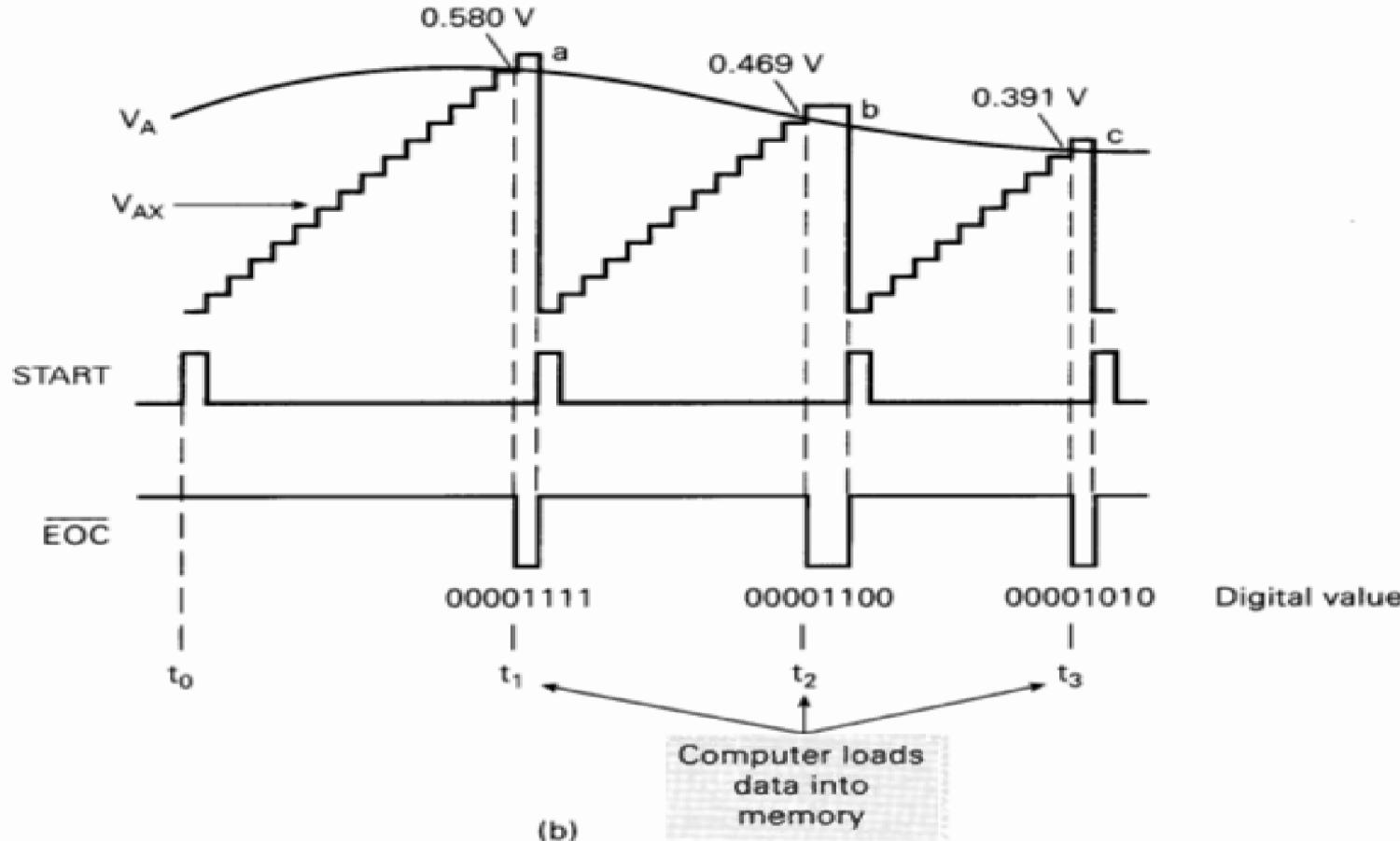


FIGURE (a) Typical computer data acquisition system; (b) waveforms showing how the computer initiates each new conversion cycle and then loads the digital data into memory at end of conversion.

Analog to Digital Conversion

Implementation of DAC and ADC:

Data Acquisition:

Figure (a) shows how a microcomputer is connected to a digital-ramp ADC for the purpose of data acquisition. The computer generates the START pulses that initiate each new A/D conversion. The \overline{EOC} (end-of-conversion) signal from the ADC is fed to the computer. The computer monitors \overline{EOC} to find out when the current A/D conversion is complete; then it transfers the digital data from the ADC output into its memory.

The waveforms in Figure (b) illustrate how the computer acquires a digital version of the analog signal, V_A . The V_{AX} staircase waveform that is generated internal to the ADC is shown superimposed on the V_A waveform for purposes of illustration. The process begins at t_0 , when the computer generates a START pulse to start an A/D conversion cycle. The conversion is completed at t_1 , when the staircase first exceeds V_A , and \overline{EOC} goes LOW. This NGT at \overline{EOC} signals the computer that the ADC has a digital output that now represents the value of V_A at point a , and the computer will load these data into its memory.

The computer generates a new START pulse shortly after t_1 to initiate a second conversion cycle. Note that this resets the staircase to 0 and \overline{EOC} back HIGH because the START pulse resets the counter in the ADC. The second conversion ends at t_2 when the staircase again exceeds V_A . The computer then loads the digital data corresponding to point b into its memory. These steps are repeated at t_3 , t_4 , and so on.

The process whereby the computer generates a START pulse, monitors \overline{EOC} , and loads ADC data into memory is done under the control of a program that the computer is executing. This data acquisition program will determine how many data points from the analog signal will be stored in the computer memory.

Analog to Digital Conversion

Implementation of DAC and ADC:

Reconstructing Digital Signal:

In Figure (b) the ADC is operating at its maximum speed since a new START pulse is generated immediately after the computer acquires the ADC output data from the previous conversion. Note that the conversion times are not constant, because the analog input value is changing. The problem with this method of storing a waveform is that in order to reconstruct the waveform, we would need to know the point in time that each data value is to be plotted. Normally, when storing a digitized waveform the samples are taken at fixed intervals at a rate that is at least two times greater than the highest frequency in the analog signal. The digital system will store the waveform as a list of sample data values. Table shows the list of data that would be stored if the signal in Figure (a) were digitized.

TABLE
samples.

Digitized data

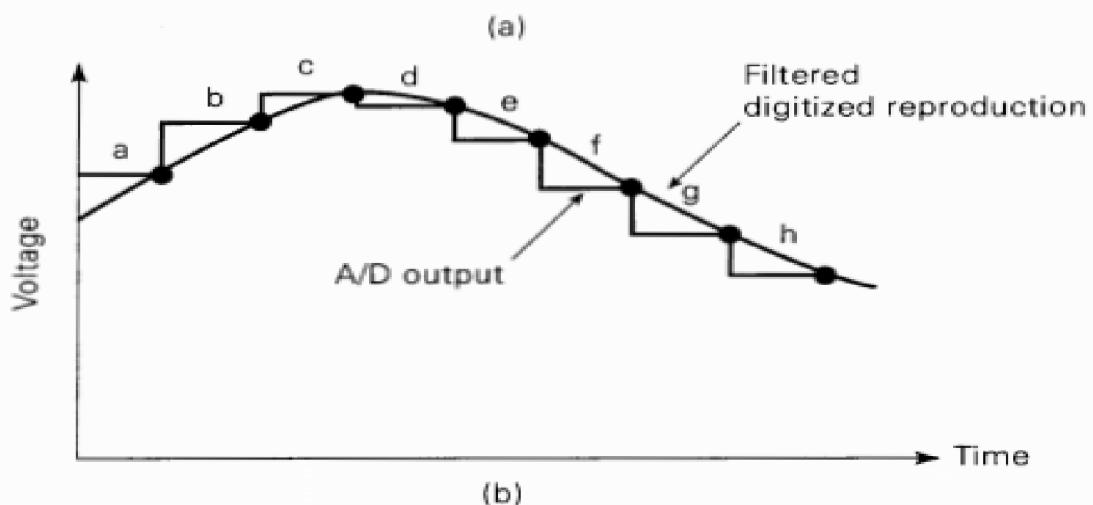
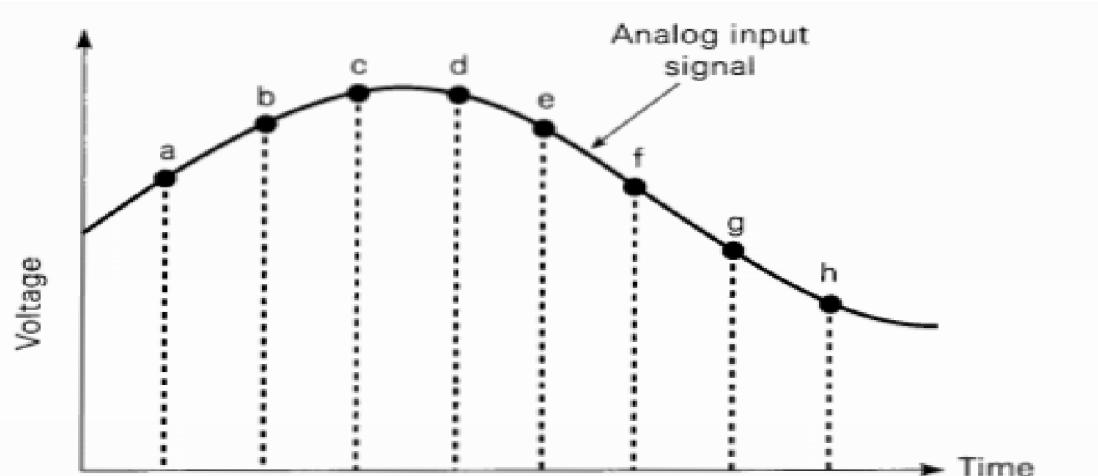
Point	Actual Voltage (V)	Digital Equivalent
a	1.22	01111010
b	1.47	10010011
c	1.74	10101110
d	1.70	10101010
e	1.35	10000111
f	1.12	01110000
g	0.91	01011011
h	0.82	01010010

Analog to Digital Conversion

Implementation of DAC and ADC:

Reconstructing Digital Signal:

FIGURE : (a) Digitizing an analog signal; (b) reconstructing the signal from the digital data.



Analog to Digital Conversion

Implementation of DAC and ADC:

Reconstructing Digital Signal:

In Figure (a) we see how the ADC continually performs conversions to digitize the analog signal at points *a*, *b*, *c*, *d*, and so on. If these digital data are used to reconstruct the signal, the result will look like that in Figure (b). The black line represents the voltage waveform that would actually come out of the D/A converter. The red line would be the result of passing the signal through a simple low-pass RC filter. We can see that it is a fairly good reproduction of the original analog signal. This is because the analog signal does not make any rapid changes between digitized points. If the analog signal contained higher-frequency variations, the ADC would not be able to follow the variations, and the reproduced version would be much less accurate.

Analog to Digital Conversion

Implementation of DAC and ADC:

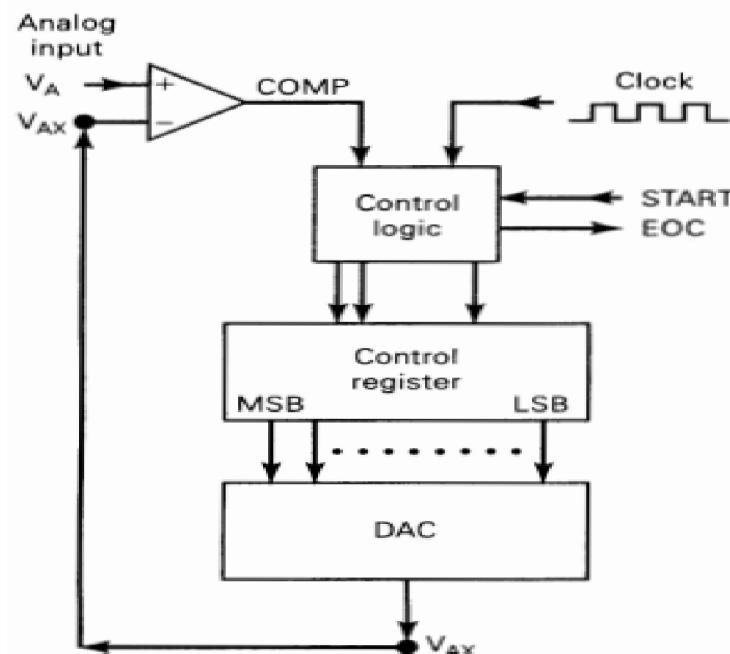
Aliasing:

The obvious goal in signal reconstruction is to make the reconstruction nearly identical to the original analog signal. In order to avoid loss of information, as has been proven by a man named Nyquist, the incoming signal must be sampled at a rate greater than two times the highest-frequency component in the incoming signal. For example, if you are pretty sure that the highest frequency in an audio system will be less than 10 kHz, you must sample the audio signal at 20,000 samples per second in order to be able to reconstruct the signal. The frequency at which samples are taken is referred to as the **sampling frequency**, F_s . What do you think would happen if for some reason a 12-kHz tone is present in the input signal? Unfortunately, the system would *not* simply ignore it because it is too high! Rather, a phenomenon called *aliasing* would occur. A signal **alias** is produced by sampling the signal at a rate less than the minimum rate identified by Nyquist (twice the highest incoming frequency). In this case any frequency over 10 kHz will produce an alias frequency. The alias frequency is always the difference between any integer multiple of the sample frequency F_s (20 kHz) and the incoming frequency that is being digitized (12 kHz). Instead of hearing a 12-kHz tone in the reconstructed signal, you would hear an 8-kHz tone which was not in the original signal.

Analog to Digital Conversion

Overview of Successive Approximation ADC:

The **successive-approximation converter** is one of the most widely used types of ADC. It has more complex circuitry than the digital-ramp ADC but a much shorter conversion time. In addition, successive-approximation converters (SACs) have a fixed value of conversion time that is not dependent on the value of the analog input.



(a)

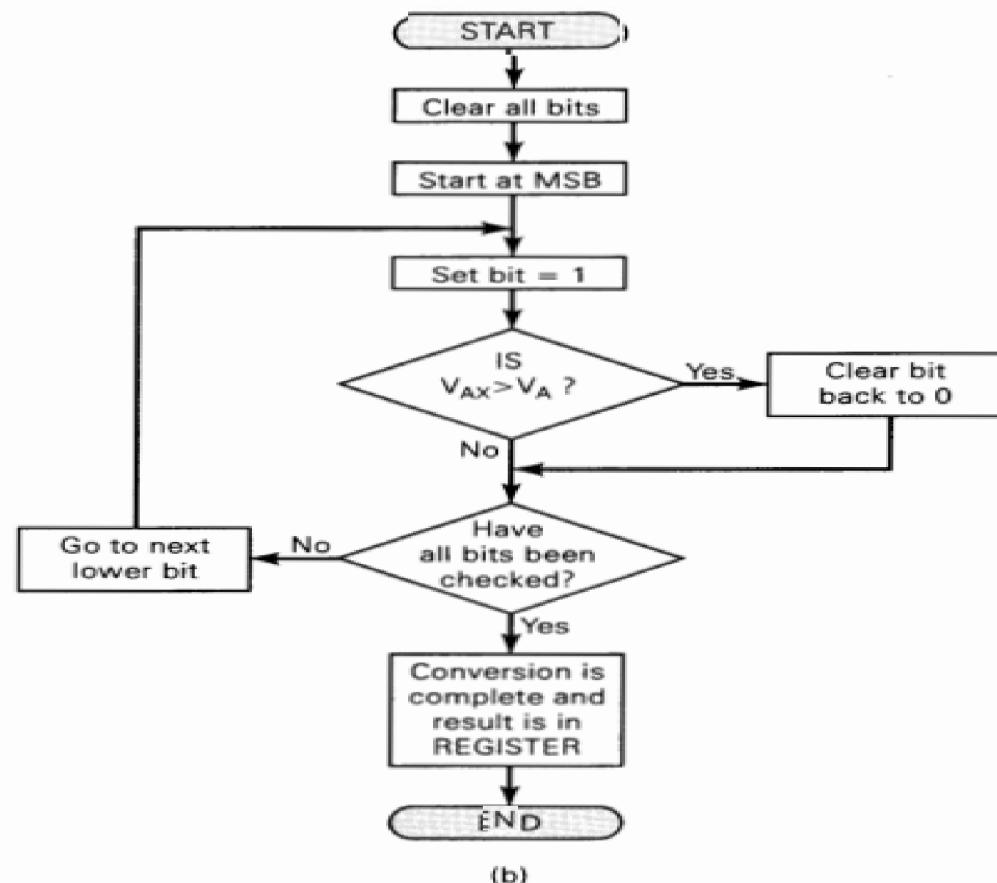


FIGURE
of operation.

Successive-approximation ADC: (a) simplified block diagram; (b) flowchart

Analog to Digital Conversion

Successive Approximation ADC:

Example of 4bit Successive Approximation ADC:

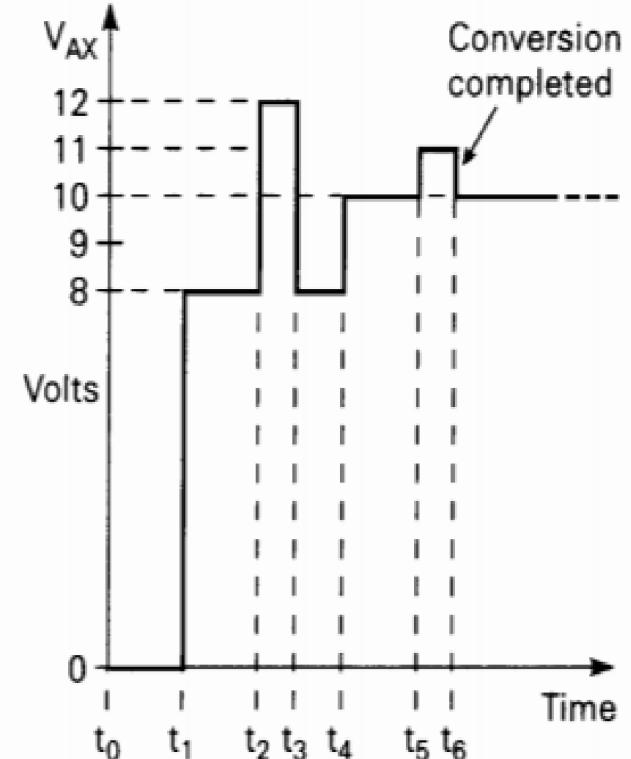
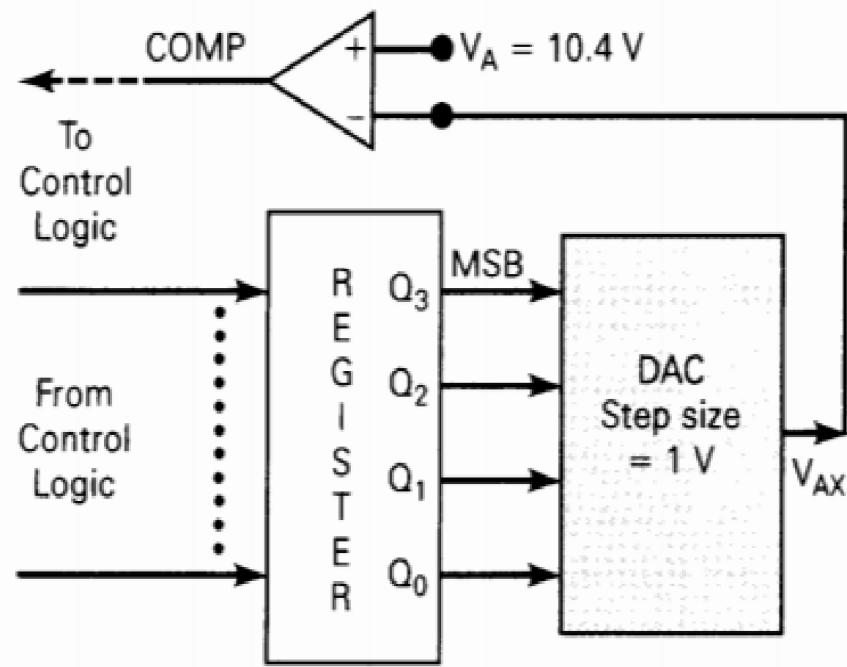


FIGURE
10.4 V.

Illustration of four-bit SAC operation using DAC step size of 1 V and $V_A = 10.4 \text{ V}$.

Analog to Digital Conversion

Successive Approximation ADC:

Example of 4bit Successive Approximation ADC:

Let's assume that the analog input is $V_A = 10.4$ V. The operation begins with the control logic clearing all of the register bits to 0 so that $Q_3 = Q_2 = Q_1 = Q_0 = 0$. We will express this as $[Q] = 0000$. This makes the DAC output $V_{AX} = 0$ V, as indicated at time t_0 on the timing diagram in Figure 10-19. With $V_{AX} < V_A$, the comparator output is HIGH.

At the next step (time t_1), the control logic sets the MSB of the register to 1 so that $[Q] = 1000$. This produces $V_{AX} = 8$ V. Since $V_{AX} < V_A$, the COMP output is still HIGH. This HIGH tells the control logic that the setting of the MSB did not make V_{AX} exceed V_A , so that the MSB is kept at 1.

The control logic now proceeds to the next lower bit, Q_2 . It sets Q_2 to 1 to produce $[Q] = 1100$ and $V_{AX} = 12$ V at time t_2 . Since $V_{AX} > V_A$, the COMP output goes LOW. This LOW signals the control logic that the value of V_{AX} is too large, and the control logic then clears Q_2 back to 0 at t_3 . Thus, at t_3 , the register contents are back to 1000 and V_{AX} is back to 8 V.

The next step occurs at t_4 , where the control logic sets the next lower bit Q_1 so that $[Q] = 1010$ and $V_{AX} = 10$ V. With $V_{AX} < V_A$, COMP is HIGH and tells the control logic to keep Q_1 set at 1.

Analog to Digital Conversion

Successive Approximation ADC:

Example of 4bit Successive Approximation ADC:

The final step occurs at t_5 , where the control logic sets the next lower bit Q_0 so that $[Q] = 1011$ and $V_{AX} = 11$ V. Since $V_{AX} > V_A$, COMP goes LOW to signal that V_{AX} is too large, and the control logic clears Q_0 back to 0 at t_6 .

At this point, all of the register bits have been processed, the conversion is complete, and the control logic activates its \overline{EOC} output to signal that the digital equivalent of V_A is now in the register. For this example, digital output for $V_A = 10.4$ V is $[Q] = 1010$. Notice that 1010 is actually equivalent to 10 V, which is *less than* the analog input; this is a characteristic of the successive-approximation method. Recall that in the digital-ramp method the digital output was always equivalent to a voltage that was on the step above V_A .

Analog to Digital Conversion

Successive Approximation ADC:

Exercise:

An eight-bit SAC has a resolution of 20 mV. What will its digital output be for an analog input of 2.17 V?

Solution

$$2.17 \text{ V}/20 \text{ mV} = 108.5$$

so that step 108 would produce $V_{AX} = 2.16 \text{ V}$ and step 109 would produce 2.18 V. The SAC always produces a final V_{AX} that is at the step *below* V_A . Therefore, for the case of $V_A = 2.17 \text{ V}$, the digital result would be $108_{10} = 01101100_2$.

Analog to Digital Conversion

IC ADC0804: Successive Approximation ADC

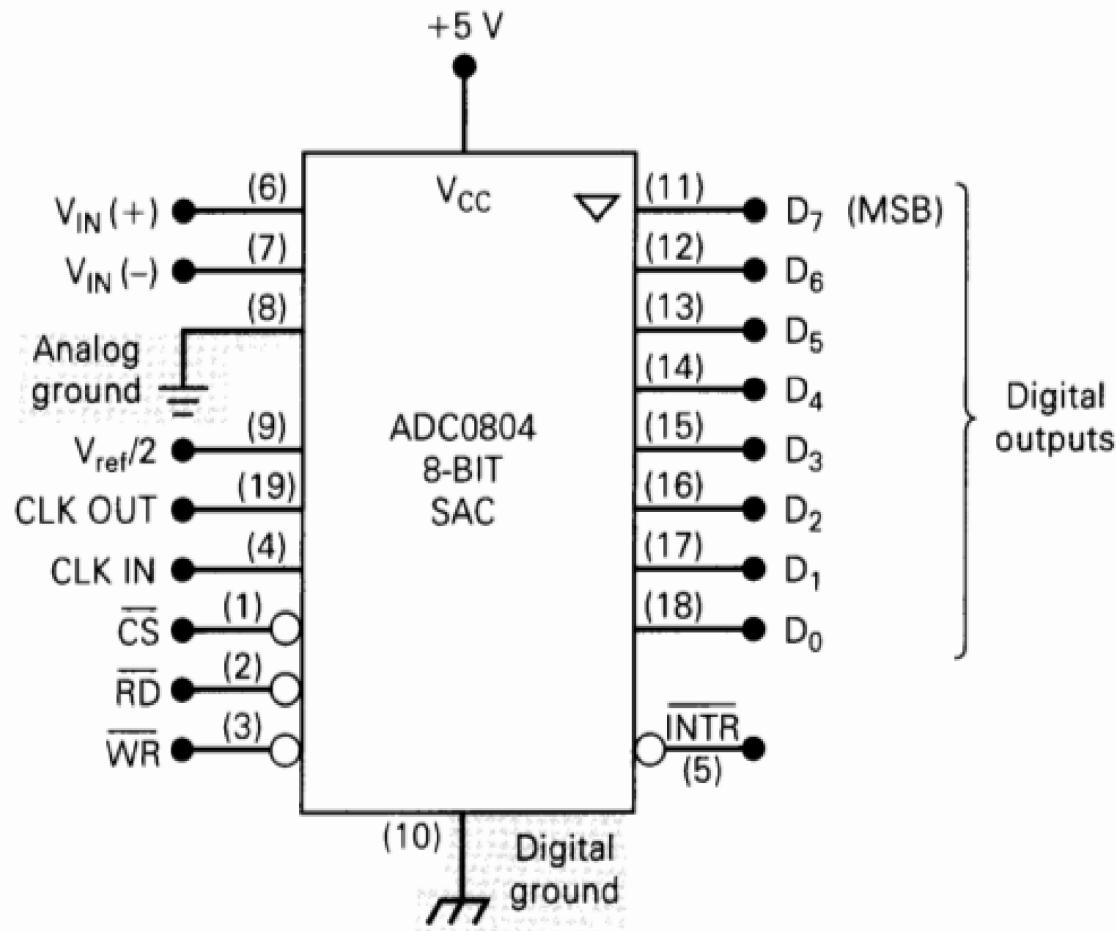
ADC0804 is a very commonly used 8-bit analog to digital convertor. It is a single channel IC, *i.e.*, it can take only one analog signal as input. The digital outputs vary from 0 to a maximum of 255. The step size can be adjusted by setting the reference voltage at pin9. When this pin is not connected, the default reference voltage is the operating voltage, *i.e.*, Vcc. The step size at 5V is 19.53mV (5V/255), *i.e.*, for every 19.53mV rise in the analog input, the output varies by 1 unit. To set a particular voltage level as the reference value, this pin is connected to half the voltage. For example, to set a reference of 4V (Vref), pin9 is connected to 2V (Vref/2), thereby reducing the step size to 15.62mV (4V/255).

ADC0804 needs a clock to operate. The time taken to convert the analog value to digital value is dependent on this clock source. An external clock can be given at the Clock IN pin. ADC 0804 also has an inbuilt clock which can be used in absence of external clock. A suitable RC circuit is connected between the Clock IN and Clock R pins to use the internal clock.

Analog to Digital Conversion

IC ADC0804: Successive Approximation ADC

FIGURE ADC0804 eight-bit successive-approximation ADC with tristate outputs. The numbers in parentheses are the IC's pin numbers.



Analog to Digital Conversion

IC ADC0804: Successive Approximation ADC

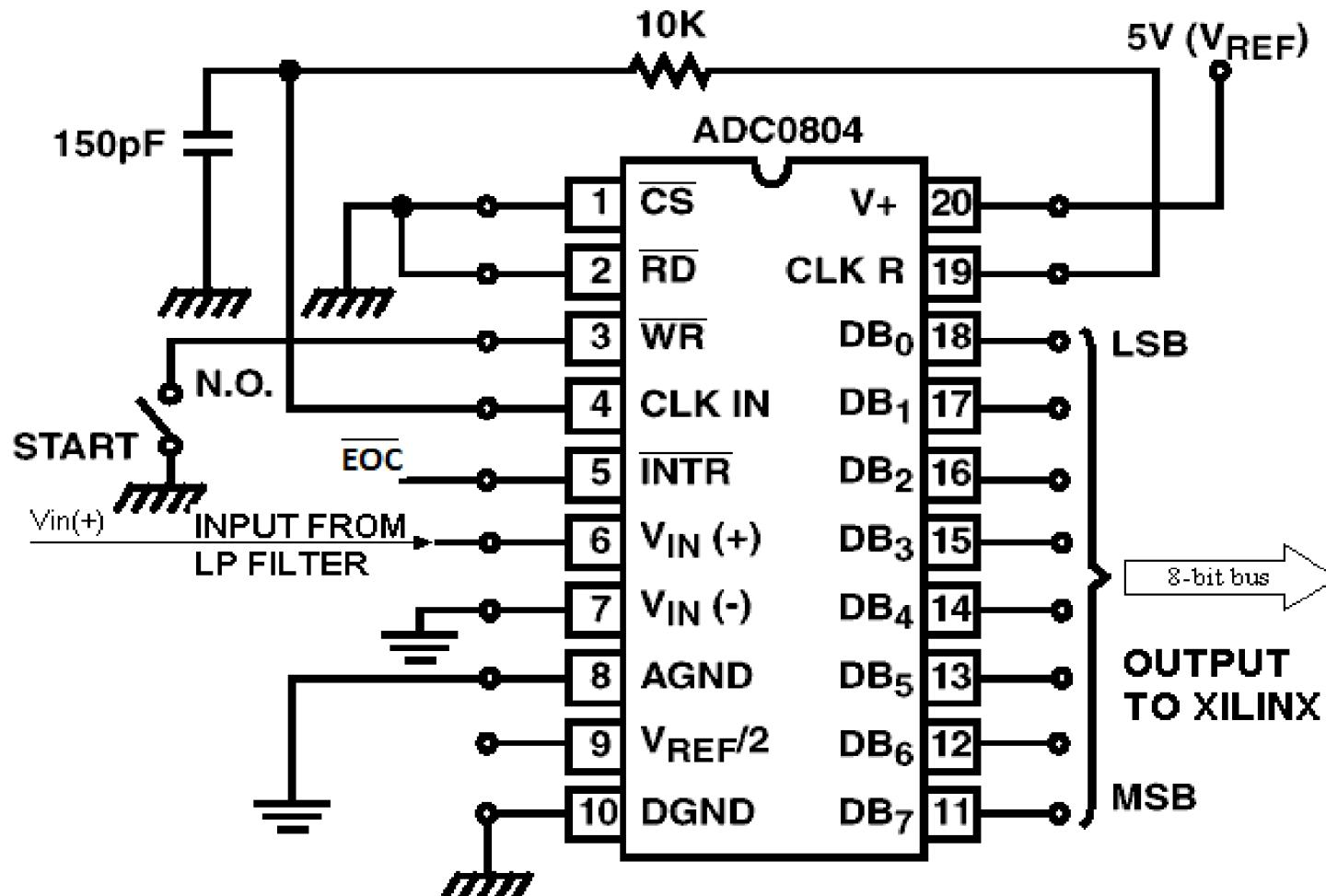
Pin Description:

Pin No	Function	Name
1	Activates ADC; Active low	Chip select
2	Input pin; High to low pulse brings the data from internal registers to the output pins after conversion	Read
3	Input pin; Low to high pulse is given to start the conversion	Write
4	Clock Input pin; to give external clock.	Clock IN
5	Output pin; Goes low when conversion is complete	Interrupt
6	Analog non-inverting input	Vin(+)
7	Analog inverting Input; normally ground	Vin(-)
8	Ground(0V)	Analog Ground
9	Input pin; sets the reference voltage for analog input	Vref/2
10	Ground(0V)	Digital Ground
11		D7
12		D6
13		D5
14		D4
15		D3
16		D2
17		D1
18		D0
19	Used with Clock IN pin when internal clock source is used	Clock R
20	Supply voltage; 5V	Vcc

Analog to Digital Conversion

IC ADC0804: Successive Approximation ADC

ADC 0804 Circuit Connection:



Analog to Digital Conversion

IC ADC0804: Successive Approximation ADC

Important Characteristics:

- It has two analog inputs, $V_{IN}(+)$ and $V_{IN}(-)$, to allow **differential inputs**. In other words, the actual analog input, V_{IN} , is the difference in the voltages applied to these pins [analog $V_{IN} = V_{IN}(+) - V_{IN}(-)$]. In single-ended measurements, the analog input is applied to $V_{IN}(+)$, while $V_{IN}(-)$ is connected to analog ground. During normal operation, the converter uses $V_{CC} = +5$ V as its reference voltage, and the analog input can range from 0 to 5 V full scale.
- It converts the analog input voltage to an eight-bit digital output. The digital outputs are tristate buffered so that they can be easily connected in a data bus arrangement. With eight bits, the resolution is $5\text{ V}/255 = 19.6\text{ mV}$.
- It has an internal clock generator circuit that produces a frequency of $f = 1/(1.1RC)$, where R and C are values of externally connected components. A typical clock frequency is 606 kHz using $R = 10\text{ k}\Omega$ and $C = 150\text{ pF}$. An external clock signal can be used, if desired, by connecting it to the CLK IN pin.
- Using a 606-kHz clock frequency, the conversion time is approximately $100\text{ }\mu\text{s}$.
- It has separate ground connections for digital and analog voltages. Pin 8 is the analog ground that is connected to the common reference point of the analog circuit that is generating the analog voltage. Pin 10 is the digital ground that is the one used by all of the digital devices in the system. (Note the different symbols used for the different grounds.) The digital ground is inherently noisy because of the rapid current changes that occur as digital devices change states. Although it is not necessary to use a separate analog ground, doing so ensures that the noise from digital ground is prevented from causing premature switching of the analog comparator inside the ADC.

Analog to Digital Conversion

IC ADC0804: Successive Approximation ADC

Pin Functions:

- **\overline{CS} (Chip Select)**. This input must be in its active-LOW state for \overline{RD} or \overline{WR} inputs to have any effect. With \overline{CS} HIGH, the digital outputs are in the Hi-Z state, and no conversions can take place.
- **\overline{RD} (READ)**. This input is used to enable the digital output buffers. With $\overline{CS} = \overline{RD} = \text{LOW}$, the digital output pins will have logic levels representing the results of the *last* A/D conversion. The microcomputer can then *read* (fetch) this digital data value over the system data bus.
- **\overline{WR} (WRITE)**. A LOW pulse is applied to this input to signal the start of a new conversion. This is actually a start conversion input. It is called a **WRITE** input because in a typical application the microcomputer generates a WRITE pulse (similar to one used for writing to memory) that drives this input.
- **\overline{INTR} (INTERRUPT)**. This output signal will go HIGH at the start of a conversion and will return LOW to signal the end of conversion. This is actually an end-of-conversion output signal, but it is called INTERRUPT because in a typical situation it is sent to a microprocessor's interrupt input to get the microprocessor's attention and let it know that the ADC's data are ready to be read.

Analog to Digital Conversion

IC ADC0804: Successive Approximation ADC

Pin Functions:

- $V_{ref}/2$. This is an optional input that can be used to reduce the internal reference voltage and thereby change the analog input range that the converter can handle. When this input is unconnected, it sits at 2.5 V ($V_{CC}/2$), since V_{CC} is being used as the reference voltage. By connecting an external voltage to this pin, the internal reference is changed to twice that voltage, and the analog input range is changed accordingly. Table shows this.
- CLK OUT. A resistor is connected to this pin to use the internal clock. The clock signal appears on this pin.
- CLK IN. Used for external clock input, or for a capacitor connection when the internal clock is used.

TABLE

$V_{ref}/2$	Analog Input Range (V)	Resolution (mV)
Open	0–5	19.6
2.25	0–4.5	17.6
2.0	0–4	15.7
1.5	0–3	11.8

Analog to Digital Conversion

IC ADC0804: Successive Approximation ADC

Pin Functions:

- $V_{ref}/2$. This is an optional input that can be used to reduce the internal reference voltage and thereby change the analog input range that the converter can handle. When this input is unconnected, it sits at 2.5 V ($V_{CC}/2$), since V_{CC} is being used as the reference voltage. By connecting an external voltage to this pin, the internal reference is changed to twice that voltage, and the analog input range is changed accordingly. Table shows this.
- CLK OUT. A resistor is connected to this pin to use the internal clock. The clock signal appears on this pin.
- CLK IN. Used for external clock input, or for a capacitor connection when the internal clock is used.

TABLE

$V_{ref}/2$	Analog Input Range (V)	Resolution (mV)
Open	0–5	19.6
2.25	0–4.5	17.6
2.0	0–4	15.7
1.5	0–3	11.8

Analog to Digital Conversion

Flash ADC:

Analog in V_A	Comparator outputs						Digital outputs			
	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C	B	A
0-1 V	1	1	1	1	1	1	1	0	0	0
1-2 V	0	1	1	1	1	1	1	0	0	1
2-3 V	0	0	1	1	1	1	1	0	1	0
3-4 V	0	0	0	1	1	1	1	0	1	1
4-5 V	0	0	0	0	1	1	1	1	0	0
5-6 V	0	0	0	0	0	1	1	1	0	1
6-7 V	0	0	0	0	0	0	1	1	1	0
> 7 V	0	0	0	0	0	0	0	1	1	1

(b)

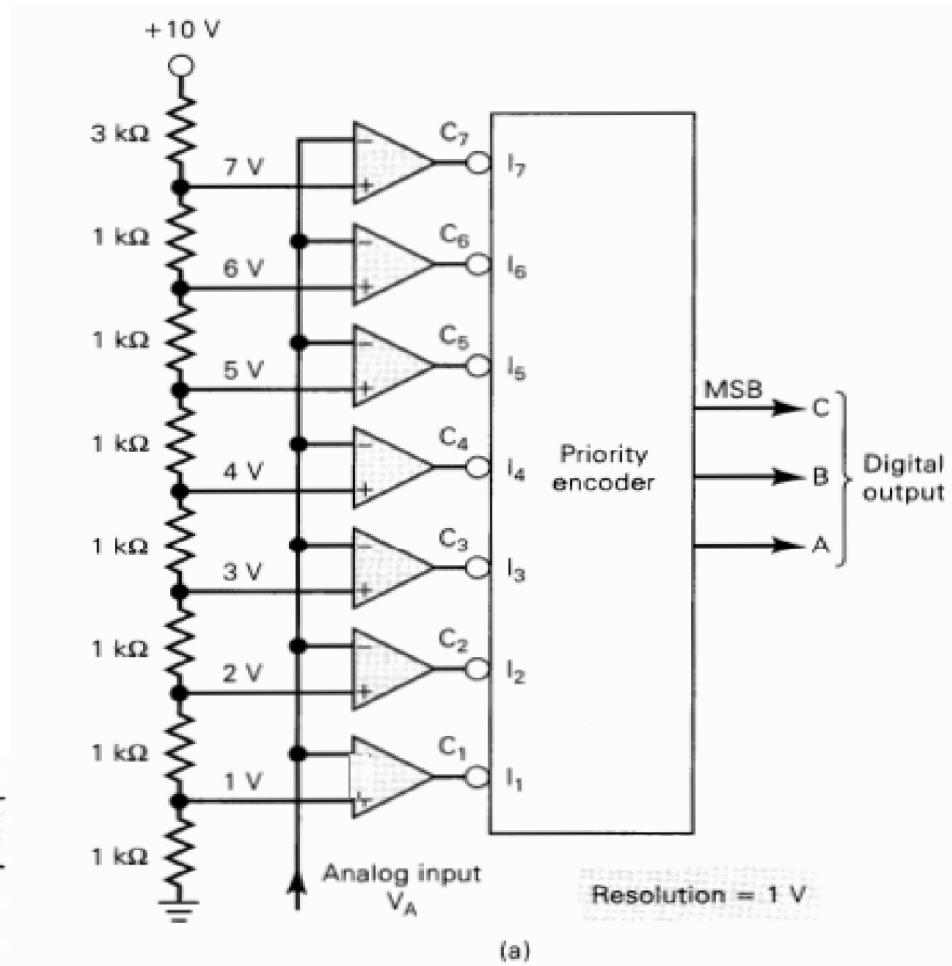


FIGURE (a) Three-bit flash ADC; (b) truth table.

Analog to Digital Conversion

Flash ADC:

The **flash converter** is the highest-speed ADC available, but it requires much more circuitry than the other types. For example, a six-bit flash ADC requires 63 analog comparators, while an eight-bit unit requires 255 comparators, and a ten-bit converter requires 1023 comparators. The large number of comparators has limited the size of flash converters. IC flash converters are commonly available in two- to eight-bit units, and most manufacturers offer nine- and ten-bit units as well.

The flash converter in Figure (a) has a three-bit resolution and a step size of 1 V. The voltage divider sets up reference levels for each comparator so that there are seven levels corresponding to 1 V (weight of LSB), 2V, 3V, . . . , and 7 V (full scale). The analog input, V_A , is connected to the other input of each comparator.

With $V_A < 1$ V, all of the comparator outputs C_1 through C_7 will be HIGH. With $V_A > 1$ V, one or more of the comparator outputs will be LOW. The comparator outputs are fed into an active-LOW priority encoder that generates a binary output corresponding to the highest-numbered comparator output that is LOW. For example, when V_A is between 3 and 4 V, outputs C_1 , C_2 , and C_3 will be LOW and all others will be HIGH. The priority encoder will respond only to the LOW at C_3 and will produce a binary output $CBA = 011$, which represents the digital equivalent of V_A , within the resolution of 1 V. When V_A is greater than 7 V, C_1 to C_7 will all be LOW, and the encoder will produce $CBA = 111$ as the digital equivalent of V_A . The table in Figure (b) shows the responses for all possible values of analog input.

Analog to Digital Conversion

Flash ADC:

The flash ADC of Figure 1 has a resolution of 1 V because the analog input must change by 1 V in order to bring the digital output to its next step. To achieve finer resolutions, we would have to increase the number of input voltage levels (i.e., use more voltage-divider resistors) and the number of comparators. For example, an eight-bit flash converter would require $2^8 = 256$ voltage levels, including 0 V. This would require 256 resistors and 255 comparators (there is no comparator for the 0-V level). The 255 comparator outputs would feed a priority encoder circuit that would produce an eight-bit code corresponding to the highest-order comparator output that is LOW. In general, an N -bit flash converter would require $2^N - 1$ comparators, 2^N resistors, and the necessary encoder logic.

Analog to Digital Conversion

Flash ADC:

Conversion Time:

The flash converter uses no clock signal, because no timing or sequencing is required. The conversion takes place continuously. When the value of analog input changes, the comparator outputs will change, thereby causing the encoder outputs to change. The conversion time is the time it takes for a new digital output to appear in response to a change in V_A , and it depends only on the propagation delays of the comparators and encoder logic. For this reason, flash converters have extremely short conversion times. For example, the MC10319 from Motorola is an eight-bit ADC that uses high-speed ECL circuits internally but has TTL-compatible logic inputs and outputs. Its typical conversion time is less than 20 ns. The Analog Devices AD9010 is a 10-bit flash converter with a conversion time under 15 ns.

Analog to Digital Conversion

Other ADCs:

Up/Down Digital-Ramp ADC (Tracking ADC)

As we have seen, the digital-ramp ADC is relatively slow because the counter is reset to 0 at the start of each new conversion. The staircase always begins at 0 V and steps its way up to the “switching point” where V_{AX} exceeds V_A and the comparator output switches LOW. The time it takes the staircase to reset to 0 and step back up to the new switching point is really wasted. The **up/down digital-ramp ADC** uses an up/down counter to reduce this wasted time.

The up/down counter replaces the up counter that feeds the DAC. It is designed to count up whenever the comparator output indicates that $V_{AX} < V_A$ and to count down whenever $V_{AX} > V_A$. Thus, the DAC output is always being stepped in the direction of the V_A value. Each time the comparator output switches states, it indicates that V_{AX} has “crossed” the V_A value, the digital equivalent of V_A is in the counter, and the conversion is complete.

When a new conversion is to begin, the counter is *not reset to 0* but begins counting either up or down from its last value, depending on the comparator output. It will count until the staircase crosses V_A again to end the conversion. The V_{AX} waveform, then, will contain both positive-going and negative-going staircase signals that “track” the V_A signal. For this reason, this ADC is often called a *tracking ADC*.

Clearly, the conversion times will generally be reduced because the counter does not start over from 0 each time but simply counts up or down from its previous value. Of course, the value of t_C will still depend on the value of V_A , and so it will not be constant.

Analog to Digital Conversion

Application Areas of DACs:

DACs are commonly used in music players to convert digital data streams into analog audio signals.

DACs are also used in televisions and mobile phones to convert digital video data into analog video signals which connect to the screen drivers to display monochrome or color images.

Analog to Digital Conversion

Application Areas of ADCs:

Digital Storage Oscilloscope

Digital Signal Processing

Multiplexing

Thank you
