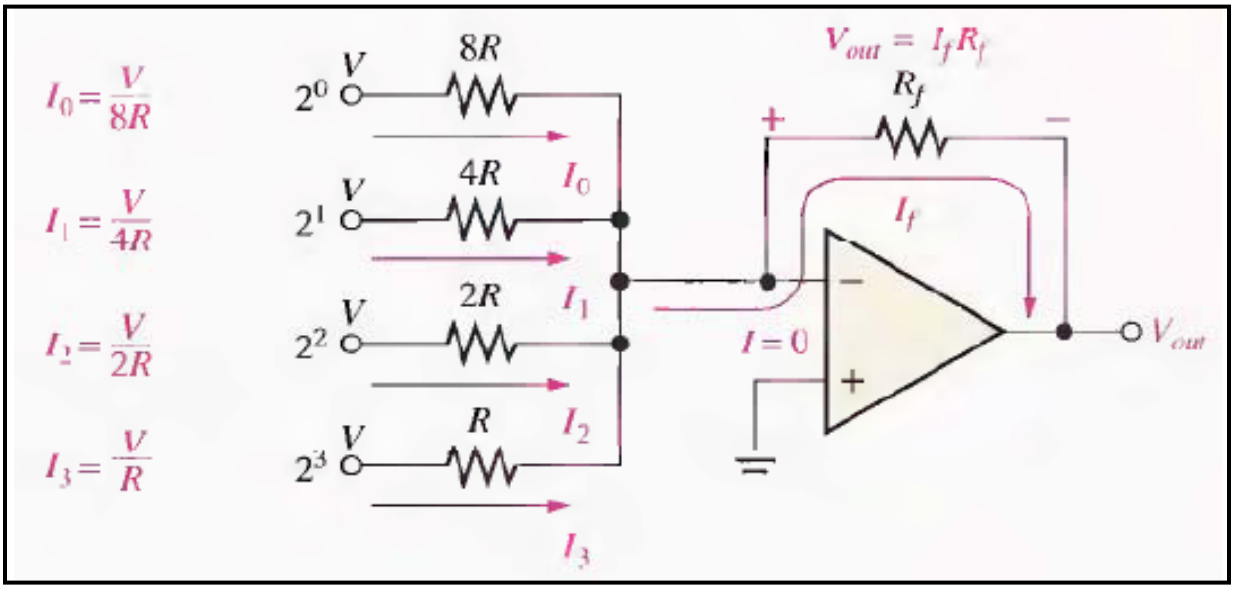
Digital and Analog Representation

Digital-to-Analog Conversion Methods

Digital-to-analog conversion is an important part of the digital processing system. Once the digital data have been processed by the digital signal processing (DSP), they are converted back to analog form. In this section, we will examine the theory of operation of two basic types of digital-to-analog converters (DACs) and learn about their performance characteristics.

1. Binary-Weighted-Input Digital-to-Analog Converter

One method of digital-to-analog conversion uses a resistor network with resistance values that represent the binary weights of the input bits of the digital code. Figure below shows a 4-bit **DAC** of this type. Each of the input resistors will either have current or have no current, depending on the input voltage level. If the input voltage is zero (**binary 0**), the current is also zero. If the input voltage is ***HIGH*** (**binary 1**), the amount of current depends on the input resistor value and is different for each input resistor, as indicated in the figure below.



A 4-bit DAC with binary-weighted inputs.

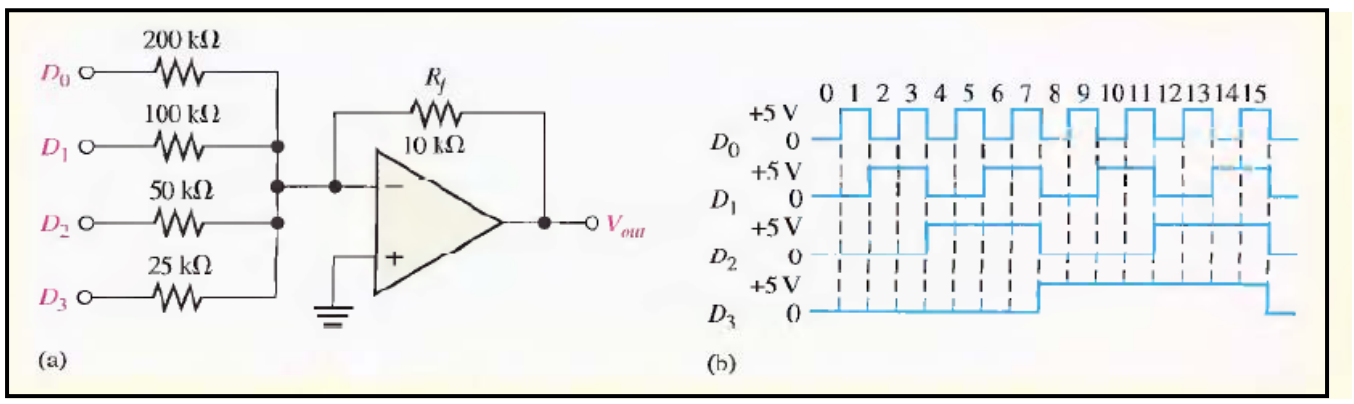
Since there is practically no current into the op-amp inverting (-) input, all of the input currents sum together and go through R_f. Since the inverting input is at (***0 V***) (*virtual ground*), the drop across R_f is equal to the output voltage, so $V_{out} = I_f R_f$.

The values of the input resistors are chosen to be inversely proportional to the binary weights of the corresponding input bits. The lowest-value resistor (***R***) corresponds to the highest binary-weighted input (2^3). The other resistors are multiples of ***R*** (that is, ***2R, 4R***, and ***8R)*** and correspond to the binary weights (2^2, 2^1, and 2^0, respectively). The input currents are also proportional to the binary weights. Thus, the output voltage is proportional to the sum of the binary weights because the sum of the input currents is through R_f.

Disadvantages of this type of ***DAC*** are the number of different resistor values and the fact that the voltage levels must be exactly the same for all inputs. For example, an 8-bit converter requires eight resistors, ranging from some value of ***R*** to ***128R*** in binary-weighted steps. This range of resistors requires tolerances of one part in 255 (less than 0.5%) to accurately convert the input, *making* this type of ***DAC*** very difficult to mass-produce.

Example:

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



Solution:

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

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

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

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

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

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

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

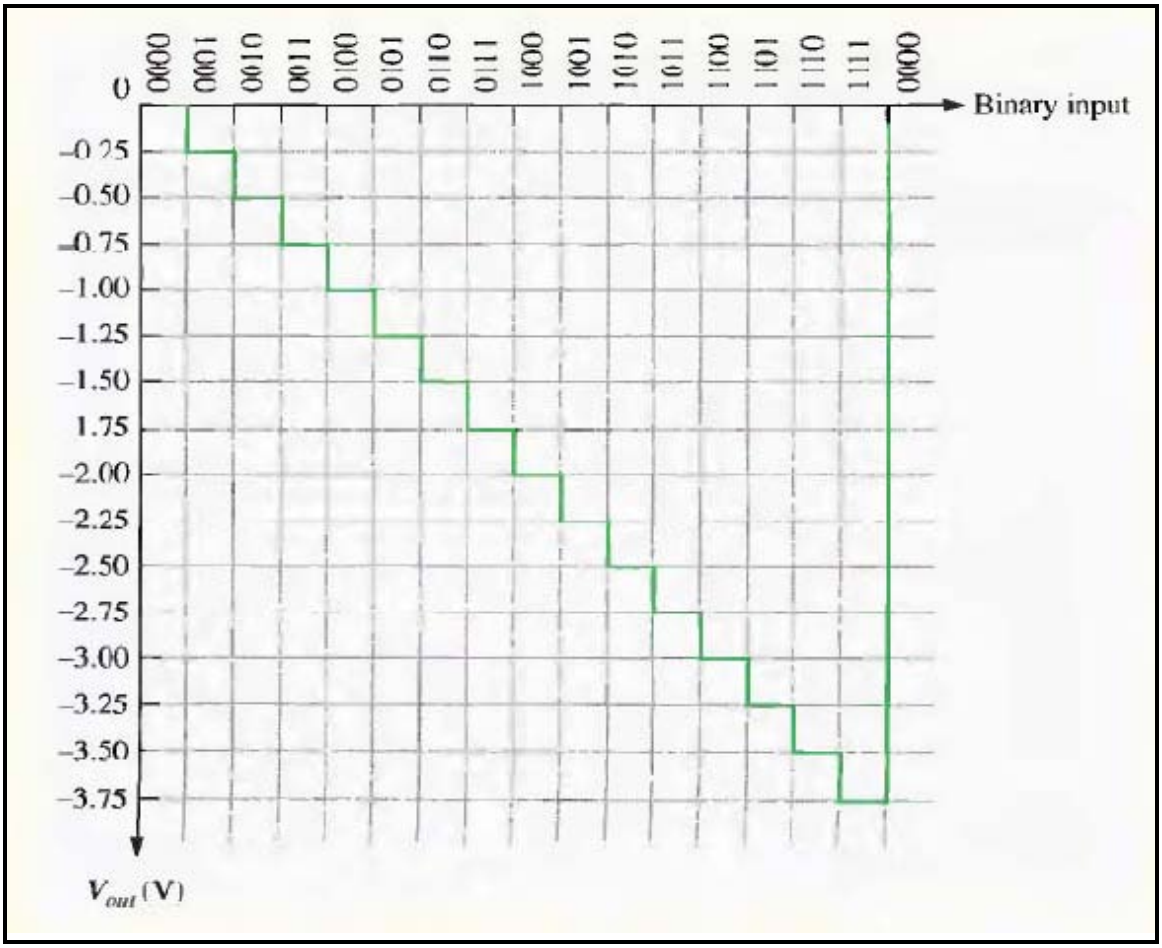
$$\boldsymbol{V_{out} = (10\ k\Omega)(-0.05\ mA) = -\ 0.5\ V.}$$

$$\boldsymbol{V_{out} = (10\ k\Omega)(-0.1\ mA) = -\ 1.0\ V.}$$

$$\boldsymbol{V_{out} = (10\ k\Omega)(-0.2\ mA) = -\ 2.0\ V.}$$

From figure (b), the first binary input code is *0000*, which produces an output voltage of (*0 V*). The next input code is *0001*, which produces an output voltage of (*-0.25 V*). For this, the output voltage is (*-0.25 V*). The next

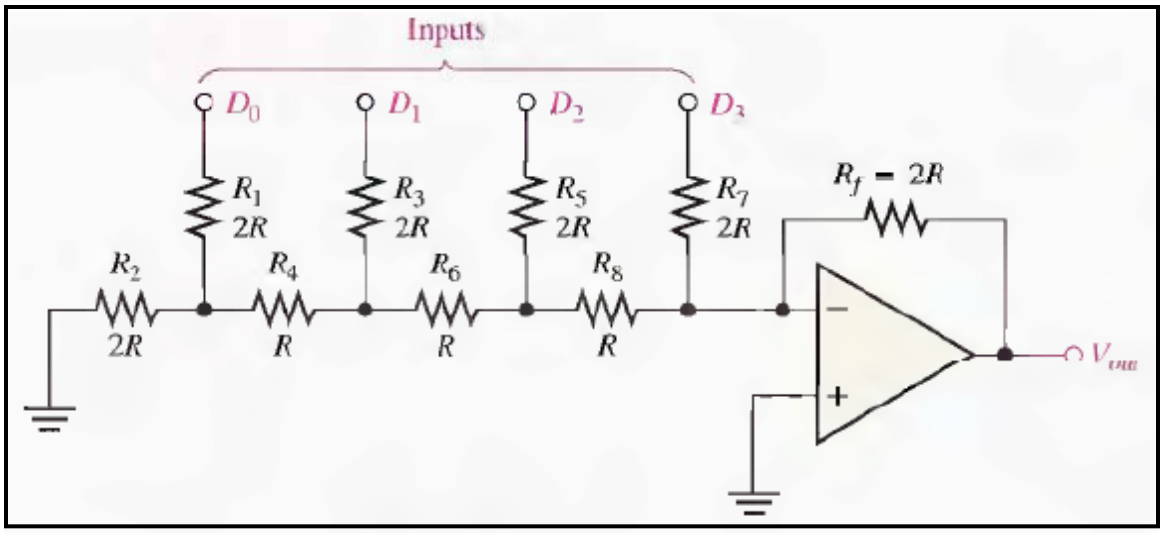
code is *0010*, which produces an output voltage of (*-0.5 V*). The next code is *0011*, which produces an output voltage of [(*-0.25 V*) + **(***-0.5 V***)** = *-0.75 V*]. Each successive binary code increases the output voltage by (*-0.25 V*), so that this particular straight binary sequence on the inputs, the output is a stair step waveform going from [*0 V to -3.75 V in -0.25 V steps*]. This is shown in figure below.



The output of the DAC

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

Another method of digital-to-analog conversion is the ***R / 2R*** ladder, as shown in figure below for four bits. It overcomes one of the problems in the binary-weighted-input ***DAC*** in that it requires only two resistor values.



An R / 2R Ladder DAC.

Start by assuming that the D_3 input is ***HIGH*** (***+5 V***) and the others are ***LOW*** (ground, ***0 V***). This condition represents the binary number *1000*. A circuit analysis will show that this reduces to the equivalent form shown in figure (a). Essentially no current goes through the (***2R***) equivalent resistance because the inverting input is at virtual ground. Thus, all of the current ***(I*** = 5 ***V / 2R)*** through ***R***, also goes through (R_f), and the output voltage is (***-5 V)***. The operational amplifier keeps the inverting (-) input near zero volts (***0 V***) because of negative feedback. Therefore**,** all current goes through (R_f) rather than into the inverting input.

Figure (b) shows the equivalent circuit when the D_2 input is at (***+5 V)*** and the others are at ground. This condition represents *0100*. If we thevenize looking from R_8, we get (***2.5 V***) in series with ***R***, as shown. This results in a

current through R_f of [$I = 2.5\ V / 2R$], which gives an output voltage of (***-2.5 V***). Keep in mind that there is no current into the op-amp inverting input and that there is no current through the equivalent resistance to ground because it has (***0 V***) across it, due to the virtual ground.

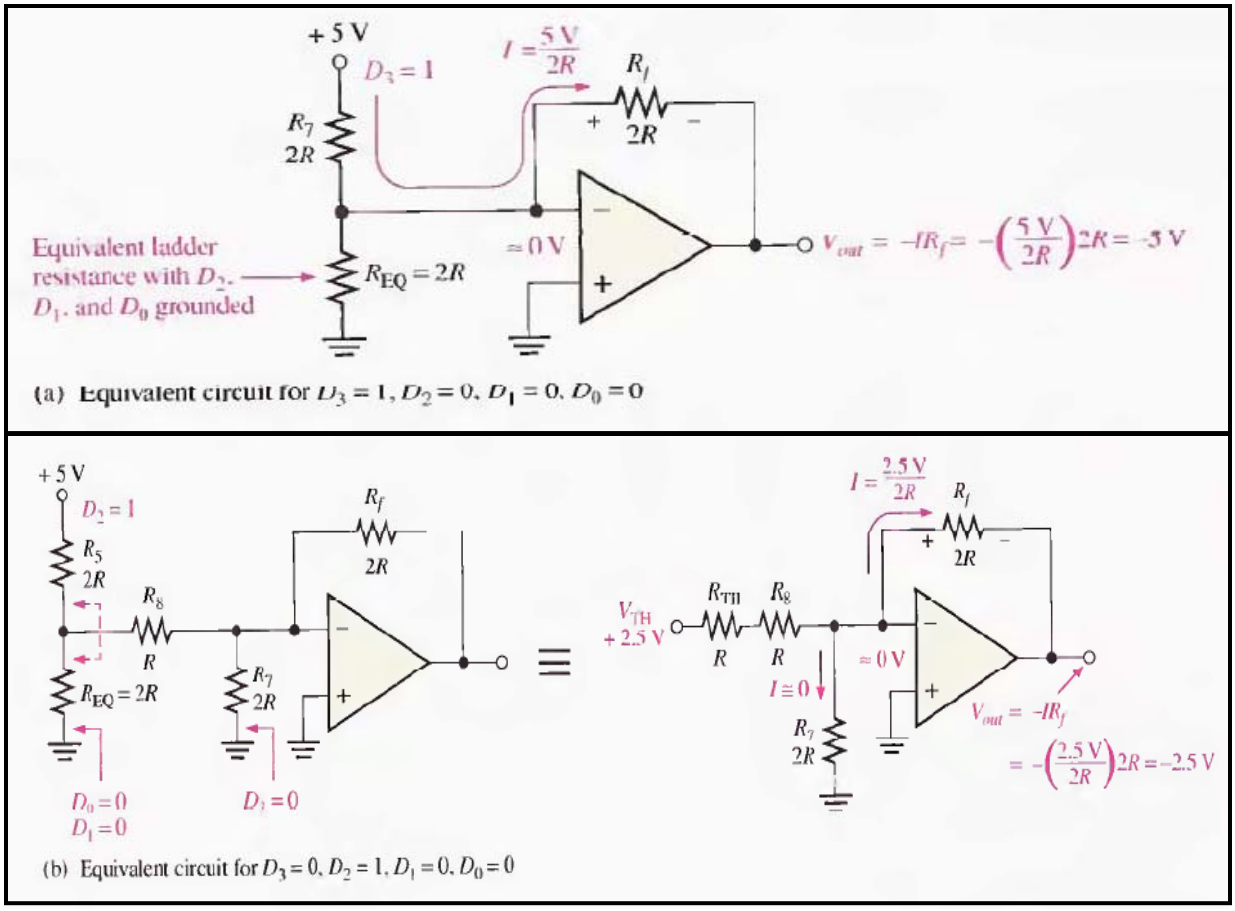
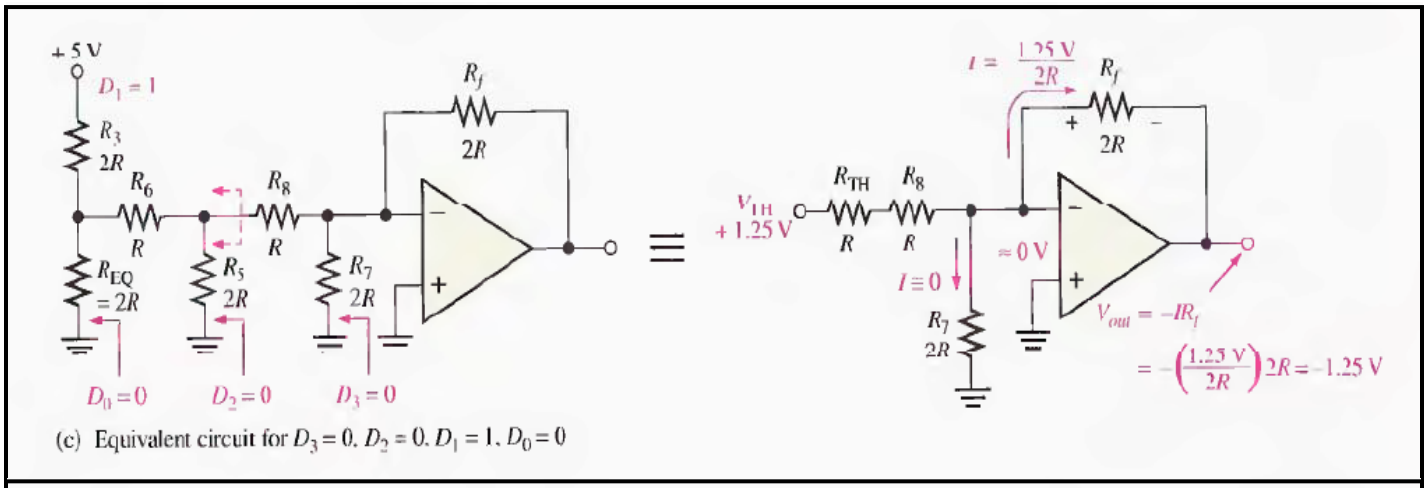


Figure (c) shows the equivalent circuit when the D_1 input is at (***+5*** *V*) and the others are at ground. This condition represents *0010*. Again thevenizing looking from R_8 you get (***1.25 V***) in series with ***R*** as shown. This results in a current through R_f of [I = ***1.25 V / 2R***], which gives an output voltage of (***- 1.25 V***).

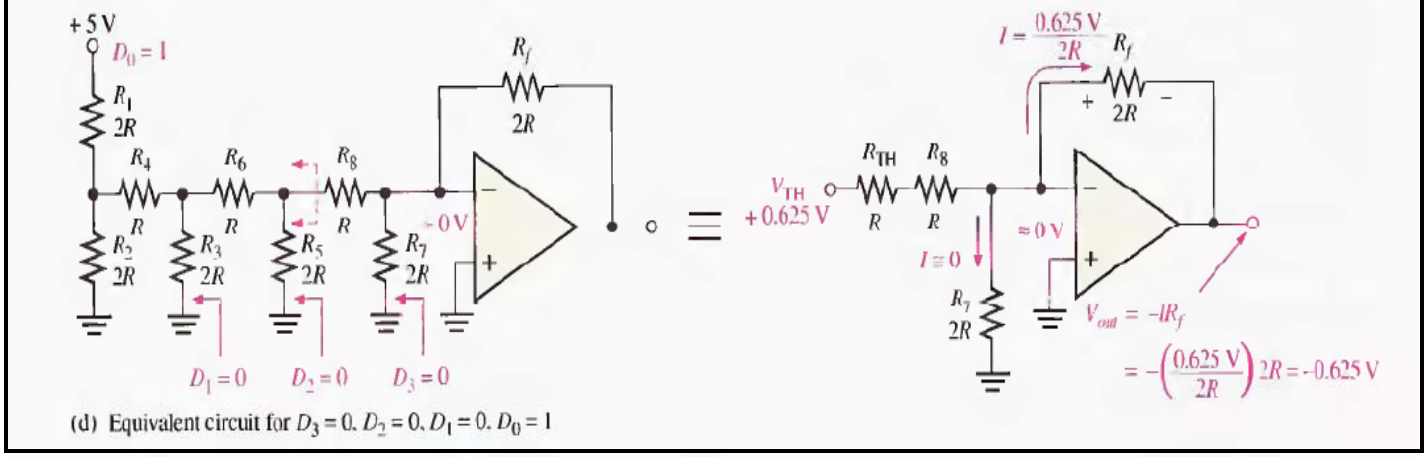
In part (d), the equivalent circuit representing the case where D_0 is at (***+5 V***) and the other inputs are at ground. This condition represents *0001*. Thevenizing from R_8 gives an equivalent of (***0.625 V***) in series with ***R*** as

shown. The resulting current through R_f is [I = **0.625 V / 2R]**, which gives an output voltage of (**- *0.625 V***).

Notice that each successively lower-weighted input produces an output voltage that is halved, so that the output voltage is proportional to the binary weight of the input bits.



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



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

Performance Characteristics of Digital-to-Analog Converters:

The performance characteristics of a ***DAC*** include ***resolution***, ***accuracy***, ***linearity***, ***monotonicity***, and ***settling time***, each of which is discussed in the following list:

- ***Resolution:*** The resolution of a ***DAC*** is the reciprocal of the number of discrete steps in the output. This, of course, is dependent on the number of input bits. For example, a 4-bit ***DAC*** has a resolution of one part in $[2^4 - 1]$ (one part in fifteen). Expressed as a percentage, this is [***(1/15)*100*** = ***6.67%***]. The total number of discrete steps equals **[**$2^n - 1$**]**, where ***n*** is the number of bits. Resolution can also be expressed as the number of bits that are converted.

- ***Accuracy:*** Accuracy is derived from a comparison of the actual output of a ***DAC*** with the expected output. It is expressed as a percentage of a full-scale, or maximum, output voltage. For example, if a converter has a full-scale output of 10 V and the accuracy is (±0.1%), then the maximum error for any output voltage is [***(10 V) (0.001)*** = ***10 mV***]. Ideally, the accuracy should be no worse than [**± *1 /2***] of a least significant. For an 8-bit converter, the least significant bit is (0.39%) of full scale. The accuracy should be approximately (±0.2%).

- ***Linearity:*** A linear error is a deviation from the ideal straight-line output of a ***DAC***. A special case is an offset error, which is the amount of output voltage when the input bits are all zeros.

- ***Monotonicity:*** A ***DAC*** is **monotonic** if it does not take any reverse steps when it is sequenced over its entire range of input bits.

- ***Settling time:*** Settling time is normally defined as the time it takes a ***DAC*** to settle within (**± 1/2 LSB**) of its final value when a change occurs in the input code.

Example: Determine the resolution, expressed as a percentage, of the following: ***a)*** an 8-bit DAC. ***a)*** an 12-bit DAC.

Solution:

(a) For the 8-bit converter.

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

(b) For the 12-bit converter.

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

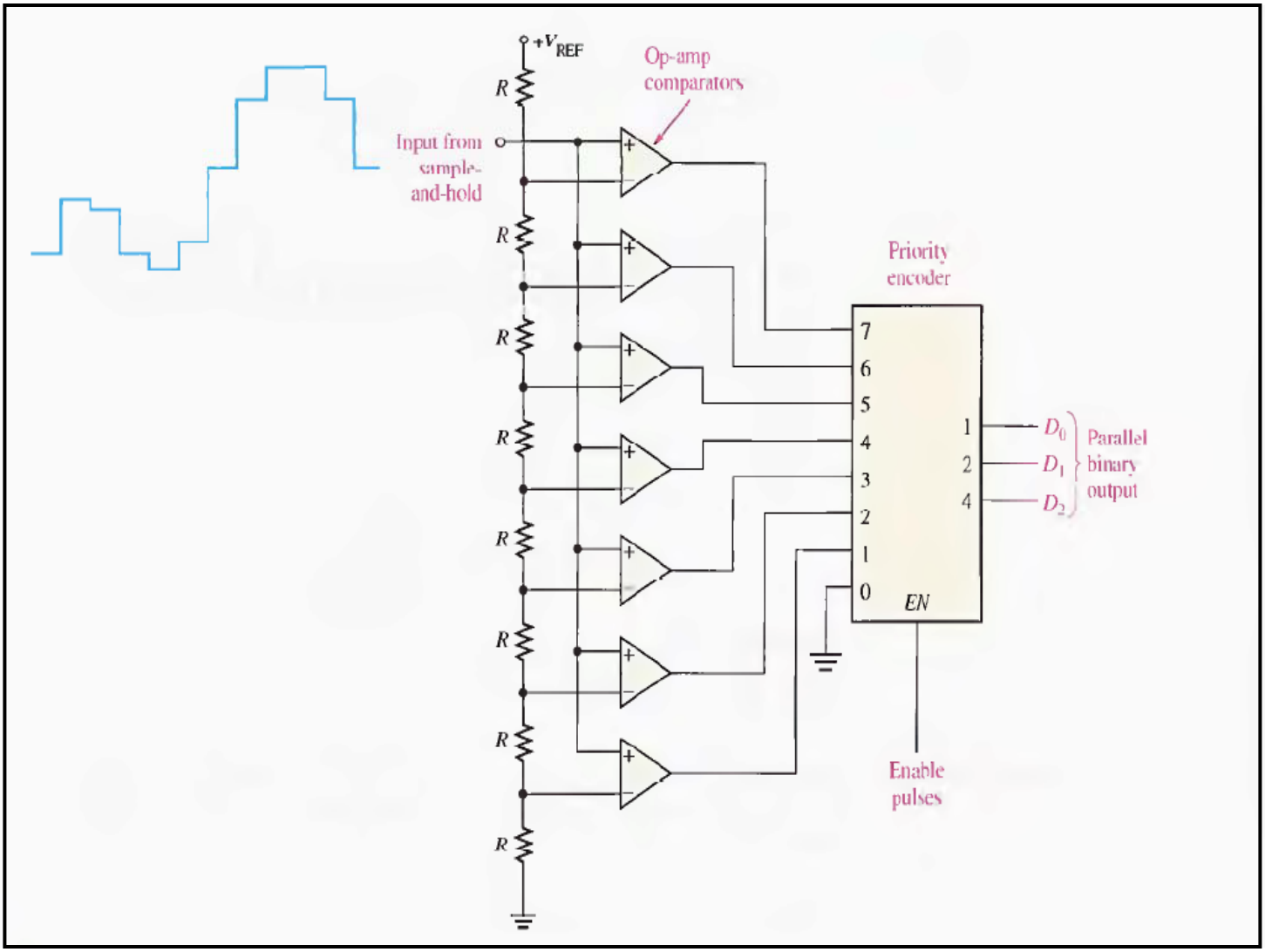
Analog-to-Digital Converter Methods:

As you have seen, analog-to-digital conversion is the process by which an analog quantity is converted to digital form. It is necessary when measured quantities must be in digital form for processing or for display or storage. Some common types of analog-to-digital converters (***ADCs***) are now examined. Two important ***ADC*** parameters are ***resolution***, which is the number of bits, and ***throughput***. This is the sampling rate an ***ADC*** can handle in units of samples per second (***sps***).

Flash (Simultaneous) Analog-to-Digital Converter:

The flash method utilizes comparators that compare reference voltages with the analog input voltage. When the input voltage exceeds the reference voltage for a given comparator, a HIGH is generated. Figure below

shows a 3-bit converter that uses seven comparator circuits; a comparator is not needed for the all-0s condition. A 4-bit converter of this type requires fifteen comparators. In general, $[2^n - 1]$ comparator are required for conversion to an n-bit binary code. The number of bits used in an ADC is its ***resolution***. The large number of ***comparators*** necessary for a reasonable-sized binary number is one of the disadvantages of the ***flash ADC***. Its chief advantage is that it provides a fast conversion time because of a high ***throughput***, measured in samples per second (sps).

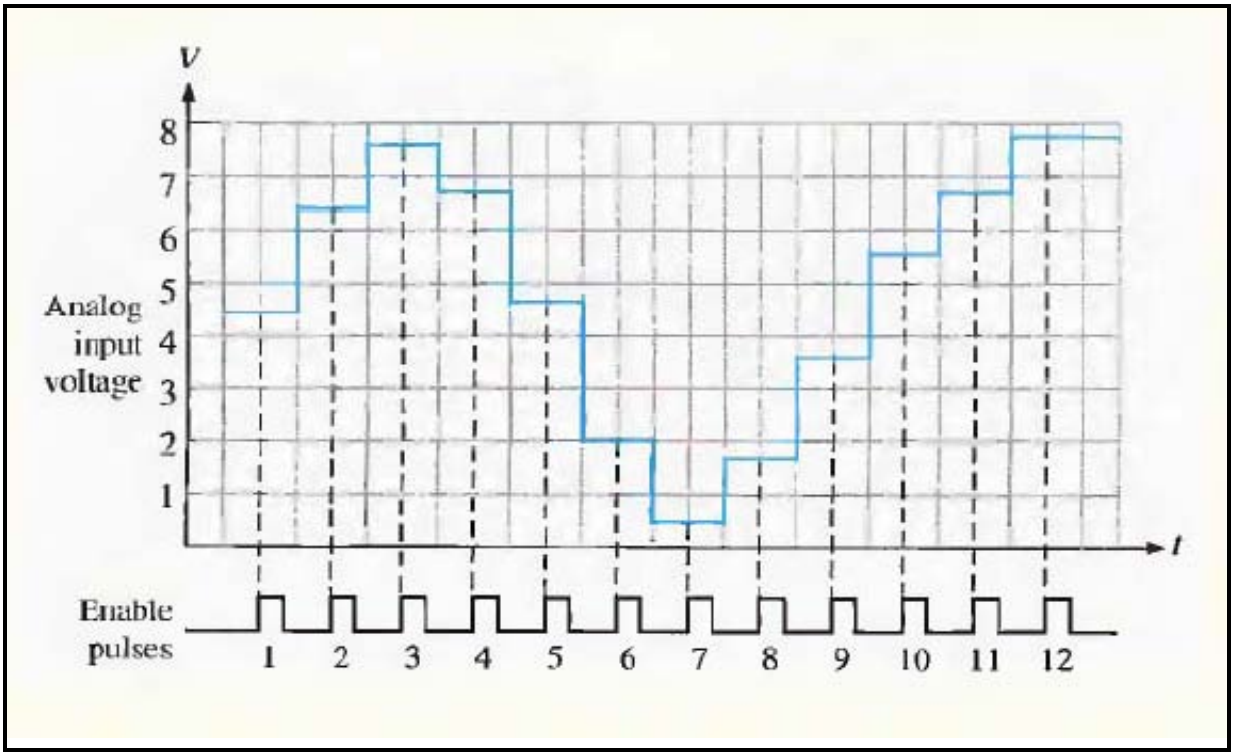


A 3-bit flash ADC.

The reference voltage for each comparator is set by the resistive voltage-divider circuit. The output of each comparator is connected to an input of the priority encoder. The encoder is enabled by a pulse on the ***EN*** input, and a 3-bit code representing the value of the input appears on the encoder's outputs. The binary code is determined by the highest-order input having a HIGH level.

The frequency of the enable pulses and the number of bits in the binary code determine the accuracy with which the sequence of binary codes represents the input of the ADC. There should be one enable pulse for each sampled level of the input signal.

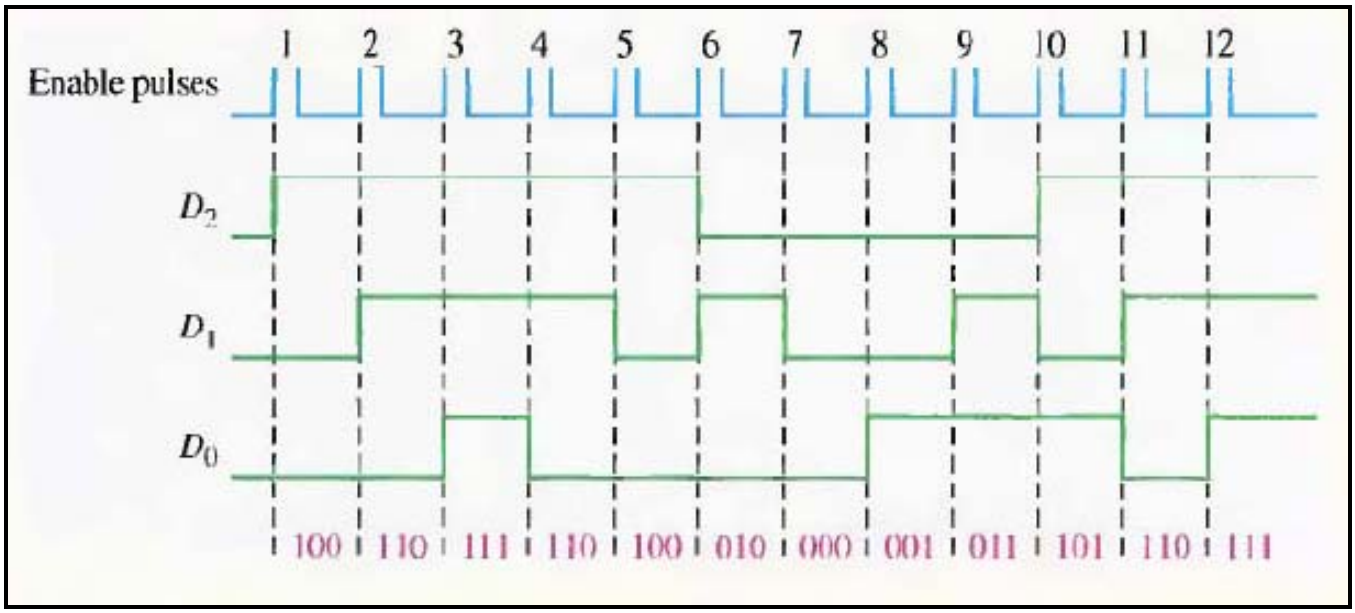
Example: Determine the binary code output of the 3-bit flash ADC in figure above for the input signal in figure below and the encoder enable pulses shown. For this example, $V_{REF.} = +8V$.



Sampling of values on a waveform for conversion to binary code.

Solution: The resulting digital output sequence is listed as follows and shown in the waveform diagram of figure below in relation to the enable pulses:

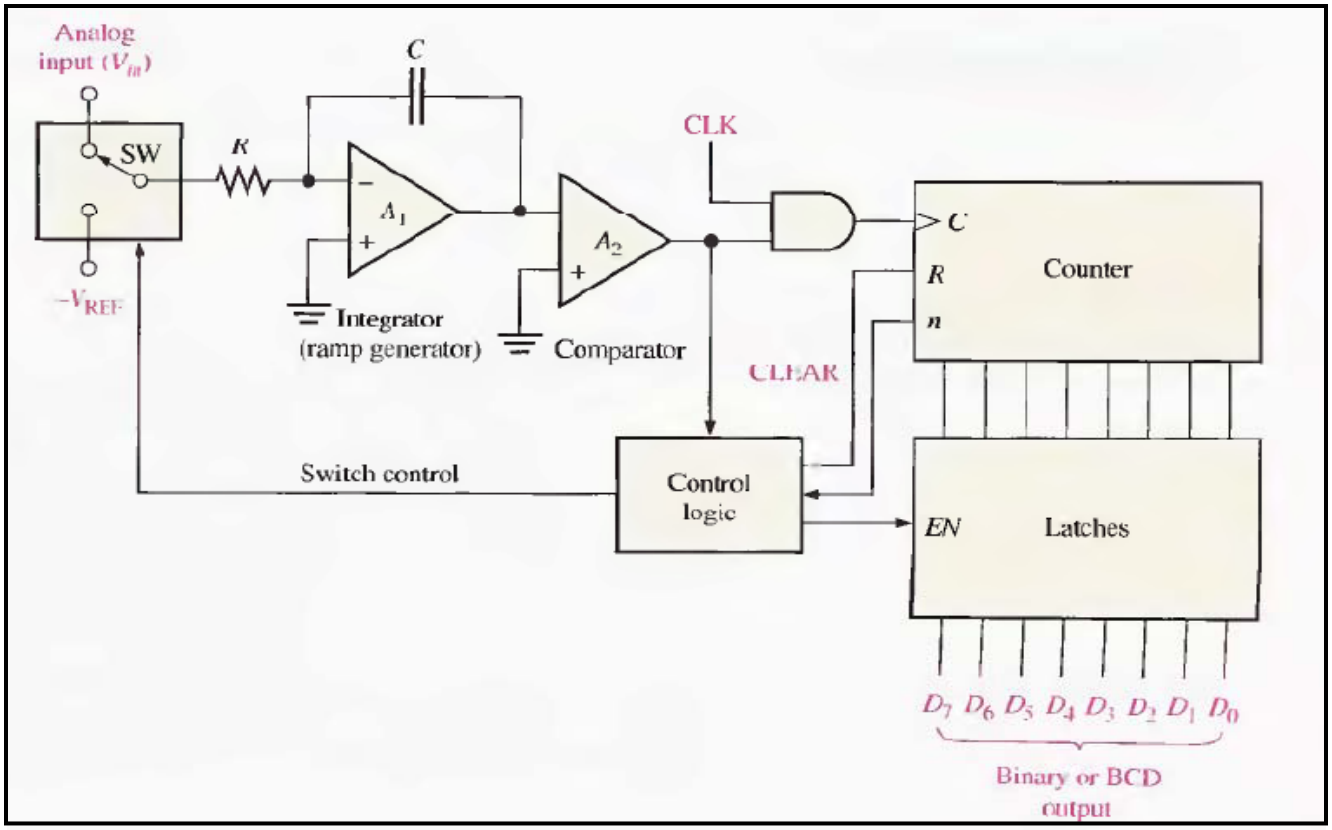
100, 110, 111, 110, 100, 010, 000, 001, 011, 101, 110, 111.



Resulting digital output for sample-and-hold values. Output D_0, is the LSB of the 3-it binary code.

Dual-Slope Analog-to-Digital Converter:

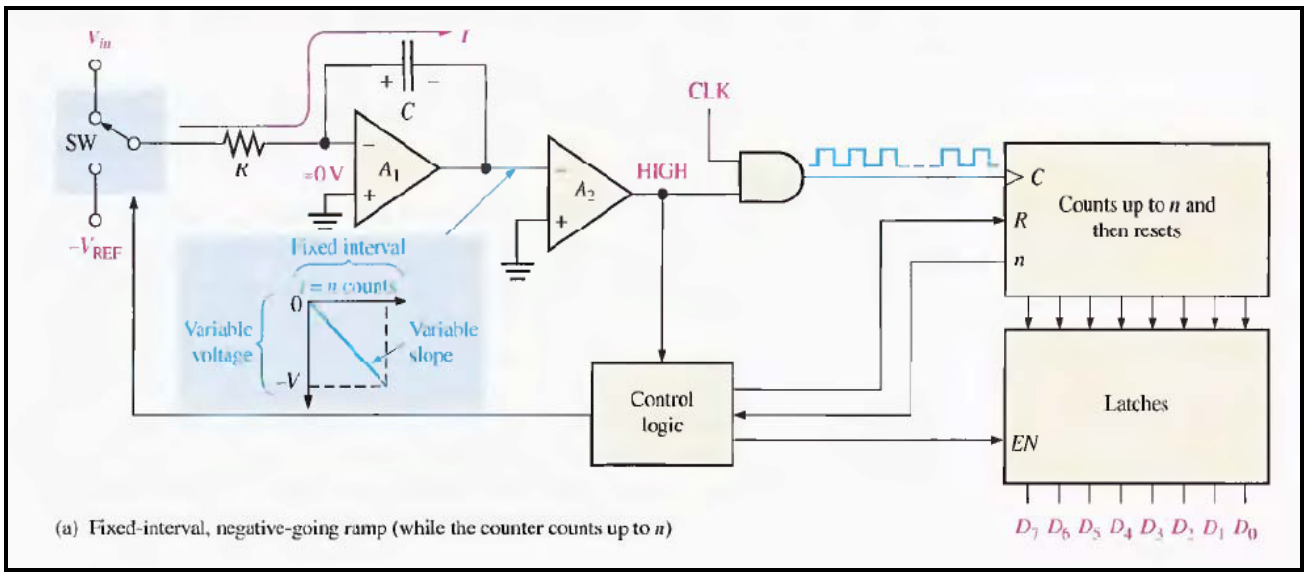
A dual-slope ADC is common in digital voltmeters and other types of measurement instruments. A ramp generator (integrator) is used to produce the dual-slope characteristic. A block diagram of a dual-slope ADC is shown in figure below.



Basic Dual-Slope ADC.

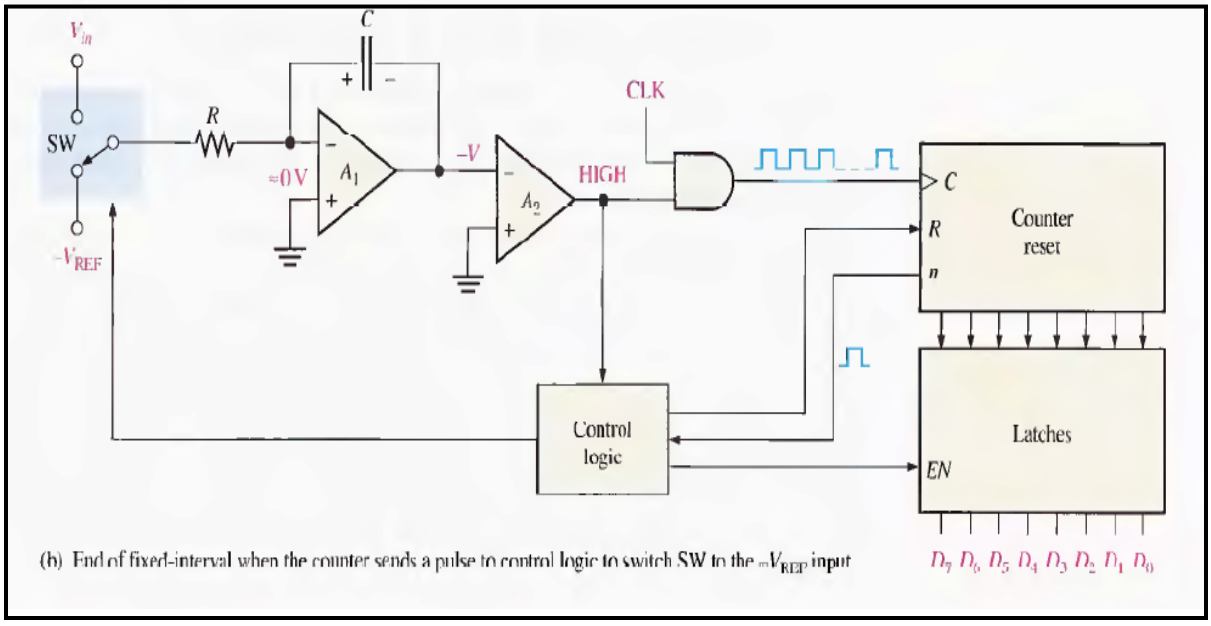
Figure (a) below illustrates dual-slope conversion. Start by assuming that the counter is reset and the output of the integrator is *m*. Now assume that a positive input voltage is applied to the input through the switch (SW) as selected by the control logic. Since the inverting input of $\boldsymbol{A_1}$, is at virtual ground, and assuming that $\boldsymbol{V_{in}}$ is constant for a period of time, there will be constant current through the input resistor ***R*** and therefore through the capacitor ***C***. Capacitor ***C*** will charge linearly because the current is constant,

and as a result, there will be a negative-going linear voltage ramp on the output of A_1 as illustrated in figure (a).



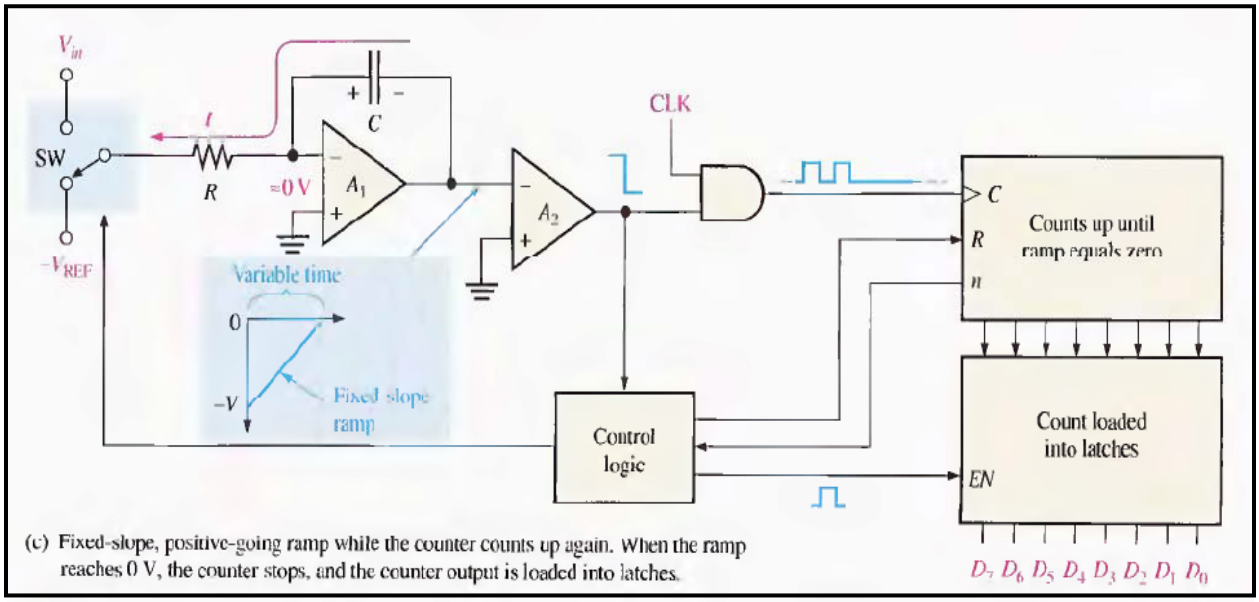
(a) Fixed-interval, negative-going ramp (while the counter counts up to n)

When the counter reaches a specified count, it will be reset, and the control logic will switch the negative reference voltage ($-V_{REF}$) to the input of A_1 as shown in figure (b). At this point the capacitor is charged to a negative voltage ($-V$) proportional to the input analog voltage.



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

Now the capacitor discharges linearly because of the constant current from the ($-V_{REF}$), as shown in figure (c). This linear discharge produces a positive-going ramp on the A_1 output, starting at ($-V$) and having a constant slope that is independent of the charge voltage. As the capacitor discharges, the counter advances from its ***RESET*** state. The time it takes the capacitor to discharge to zero depends on the initial voltage ($-V$) (proportional to V_{in}) because the discharge rate (slope) is constant. When the integrator (A_1) output voltage reaches zero, the comparator (A_2) switches to the LOW state and disables the clock to the counter. The binary count is latched, thus completing one conversion cycle. The binary count is proportional to V_{in} because the time it takes the capacitor to discharge depends only on ($-V$), and the counter records this interval of time.



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

Successive-Approximation Analog-to-Digital Converter:

One of the most widely used methods of analog-to-digital conversion is successive-approximation. It has a much faster conversion time than the dual-slope conversion, but it is slower than the flash method. It also has a fixed conversion time that is the same for any value of the analog input.

Figure (1) shows a basic block diagram of a 4-bit successive approximation ***ADC***. It consists of a ***DAC***, a successive-approximation register (***SAR***), and a comparator. The basic operation is as follows: The input bits of the DAC are enabled (made equal to a 1) one at a time, starting with the most significant bit (MSB). As each bit is enabled, the comparator produces an output that indicates whether the input signal voltage is greater or less than the output of the ***DAC***. If the ***DAC*** output is greater than the input signal, the comparator's output is LOW, causing the bit in the register to reset. If the output is less than the input signal, the I bit is retained in the register. The system does this with the MSB first, then the next most significant bit, then the next, and so on. After all the bits of the DAC have been tried, the conversion cycle is complete.

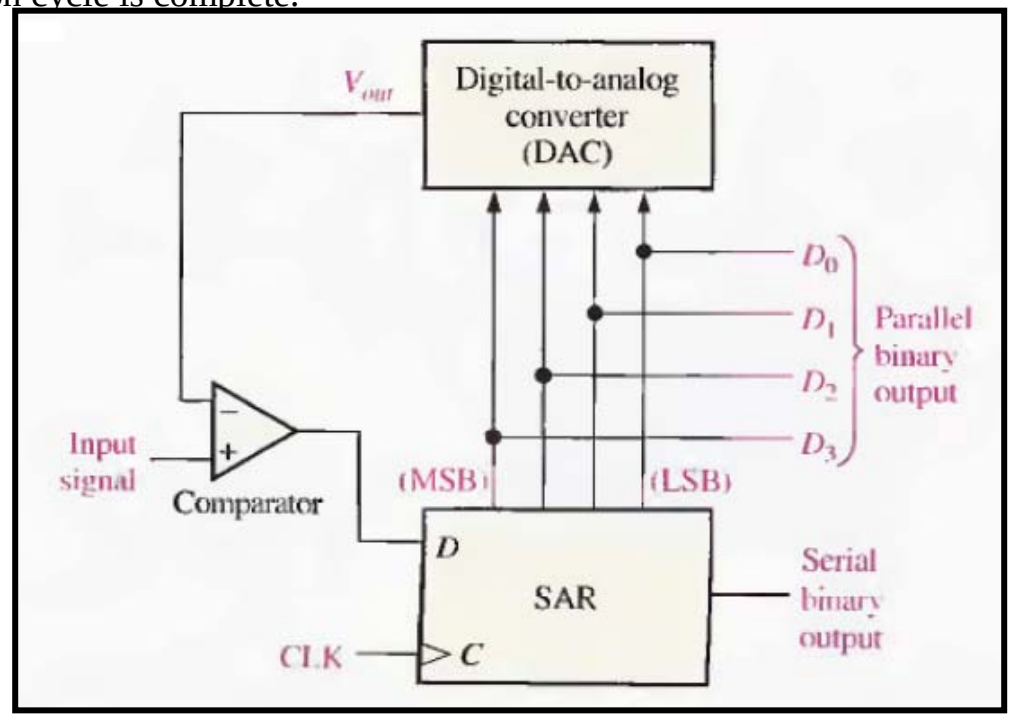


Figure (1): Successive-Approximation ADC.

In order to better understand the operation of the successive-approximation ***ADC***, let's take a specific example of a 4-bit conversion. Figure (2) illustrates the step-by-step conversion of a constant input voltage (5.1 V in this case). Let's assume that the DAC has the following output characteristic: V_{out} = 8 V for the 2^3 bit (MSB), V_{out} = 4 V for the 2^2 bit, V_{out} = 2 V for the 2^1 bit, and V_{out}=1 V for the 2^0 bit (LSB).

Figure 2(a) shows the first step in the conversion cycle with the MSB =1. The output of the ***DAC*** is 8 V. Since this is greater than the input of 5.1 V, the output of the comparator is LOW, causing the MSB in the ***SAR*** to be reset to a 0.

Figure 2(b) shows the second step in the conversion cycle with the 2^2 bit equal to a 1. The output of the ***DAC*** is 4 V. Since this is less than the input of 5.1 V, the output of the comparator switches to a HIGH, causing this bit to be retained in the ***SAR***.

Figure 2(c) shows the this step in the conversion cycle with the 2^1 bit equal to a 1. The output of the ***DAC*** is 6 V because there is a 1 on the 2^2 bit input and on the 2' bit input; 4V + 2V = 6 V. Since this is greater than the input of 5.1 V. the output of the comparator switches to a LOW, causing this bit to be reset to a 0.

Figure 2(d) shows the fourth and final step in the conversion cycle with the 2" bit equal to a 1. The output of the ***DAC*** is 5 V because there is a 1 on the 2^2 bit input and on the 2^0 bit input; 4 V + 1 V = 5 V.

The four bits have all been tried, thus completing the conversion cycle. At this point the binary code in the register is 0101, which is approximately the binary value of the input of 5.1 V. Additional bits will produce an even more accurate result. Another conversion cycle now begins, and the basic process is repeated. The ***SAR*** is cleared at the beginning of each cycle.

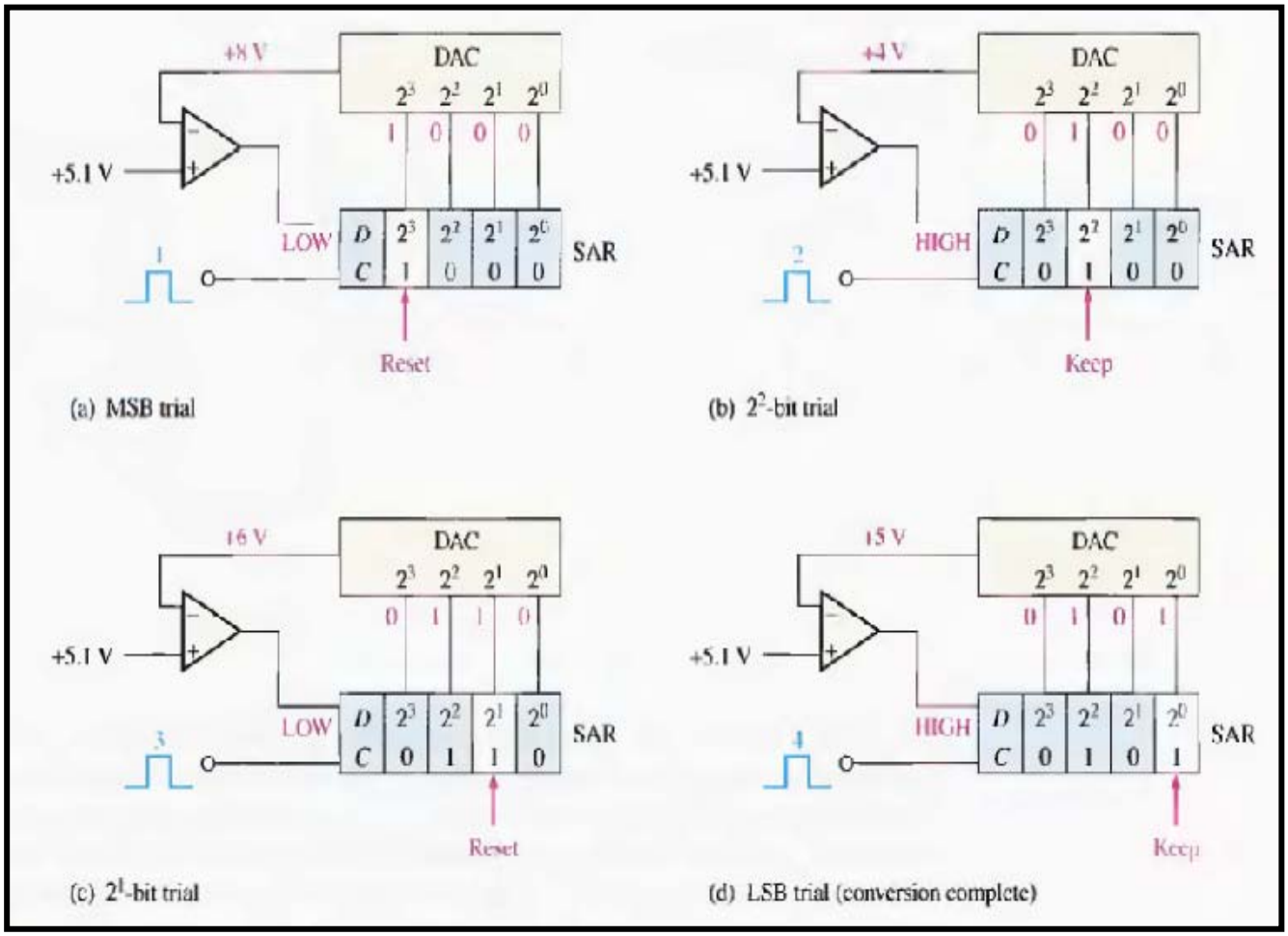


Figure (2): Illustration of the successive-approximation conversion process.