Department of Computer Science & Engineering

LAB -FPGA

Introduction to DSP architectures and programming

Digital signal processing algorithms typically require a large number of mathematical operations to be performed quickly and repeatedly on a series of data samples. Signals (perhaps from audio or video sensors) are constantly converted from analog to digital, manipulated digitally, and then converted back to analog form. Many DSP applications have constraints on latency; that is, for the system to work, the DSP operation must be completed within some fixed time, and deferred (or batch) processing is not viable.

Most general-purpose microprocessors and operating systems can execute DSP algorithms successfully, but are not suitable for use in portable devices such as mobile phones and PDAs because of power efficiency constraints.[5] A specialized DSP, however, will tend to provide a lower-cost solution, with better performance, lower latency, and no requirements for specialised cooling or large batteries.

Such performance improvements have led to the introduction of digital signal processing in commercial communications satellites where hundreds or even thousands of analog filters, switches, frequency converters and so on are required to receive and process the uplinked signals and ready them for downlinking, and can be replaced with specialised DSPs with significant benefits to the satellites' weight, power consumption, complexity/cost of construction, reliability and flexibility of operation. For example, the SES-12 and SES-14 satellites from operator SES launched in 2018, were both built by Airbus Defence and Space with 25% of capacity using DSP.

By the standards of general-purpose processors, DSP instruction sets are often highly irregular; while traditional instruction sets are made up of more general instructions that allow them to perform a wider variety of operations, instruction sets optimized for digital signal processing contain instructions for common mathematical operations that occur frequently in DSP calculations.

Following operations must be known in understanding the dsp

Instruction sets

- multiply–accumulates (MACs, including fused multiply–add, FMA) operations
  - used extensively in all kinds of matrix operations
    - convolution for filtering
    - dot product

        - polynomial evaluation
    - Fundamental DSP algorithms depend heavily on multiply–accumulate performance
        - FIR filters
        - Fast Fourier transform (FFT)

Analog-to-Digital Converter (ADC)

Real signals (e.g., a voltage measured with a thermocouple or a speech signal recorded with a microphone) are analog quantities, varying continuously with time.

Digital format offers several advantages: digital signal processing, storage, use of computers, robust transmission, etc.

* An ADC (Analog-to-Digital Converter) is used to convert an analog signal to the digital format.

Analog-to-Digital converters (ADC) translate analog signals, real world signals like temperature, pressure, voltage, current, distance, or light intensity, into a digital representation of that signal. This digital representation can then be processed, manipulated, computed, transmitted or stored.

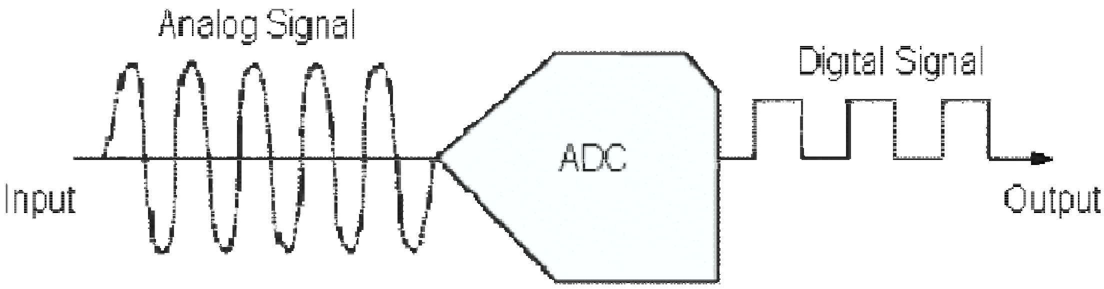
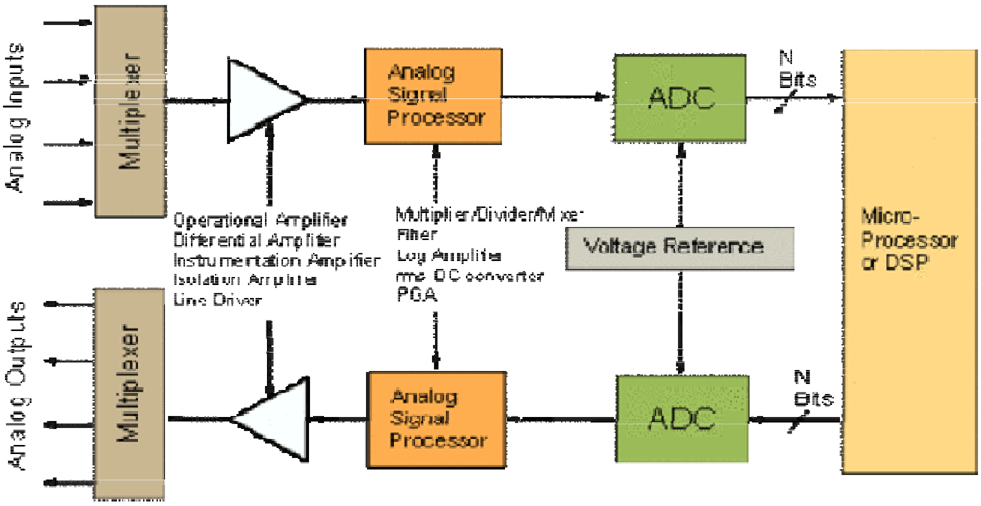


Figure 20.1 Analog to Digital conversion

In many cases, the analog to digital conversion process is just one step within a larger measurement and control loop where digitized data is processed and then reconverted back to analog signals to drive external transducers. These transducers can include things like motors, heaters and acoustic divers like loudspeakers. The performance required of the ADC will reflect the performance goals of the measurement and control loop. ADC performance needs will also reflect the capabilities and requirements of the other signal processing elements in the loop.



An ADC samples an analog waveform at uniform time intervals and assigns a digital value to each sample. The digital value appears on the converter's output in a binary coded format. The value is obtained by dividing the sampled analog input voltage by the reference voltage and them multiplying by the number of digital codes. The resolution of converter is set by the number of binary bits in the output code.

$$\text{Digital Output Code} = \frac{\text{Analog Input}}{\text{Reference Input}} \times (2^N - 1)$$

Figure 20.3 Digital output code

An ADC carries out two processes, sampling and quantization. The ADC represents an analog signal, which has infinite resolution, as a digital code that has finite resolution. The ADC produces 2N digital values where N represents the number of binary output bits. The analog input signal will fall between the quantization levels because the converter has finite resolution resulting in an inherent uncertainty or quantization error. That error determines the maximum dynamic range of the converter.

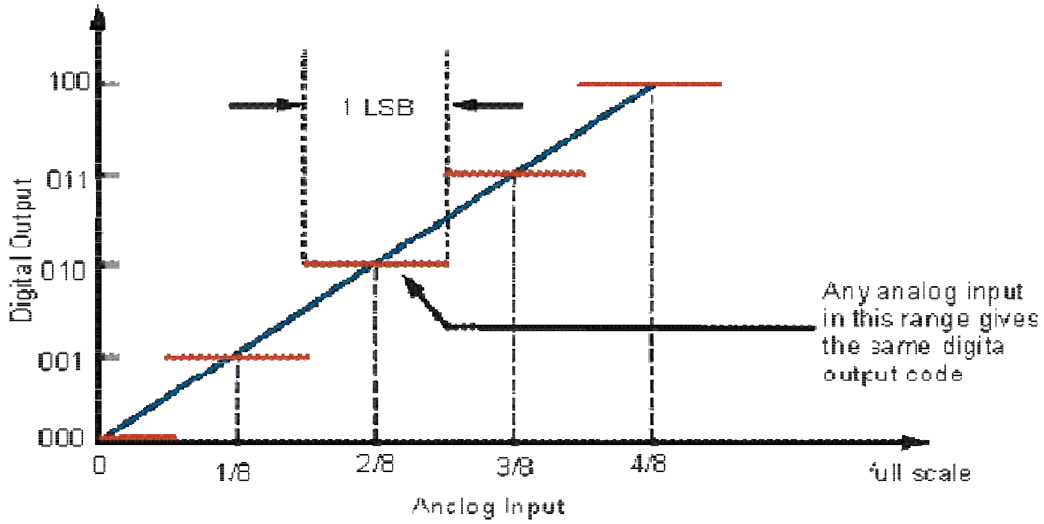


Figure 20.4 Quantization Process

The **sampling** process represents a continuous time domain signal with values measured at discrete and uniform time intervals. This process determines the maximum bandwidth of the sampled signal in accordance with the Nyquist Theory. **This theory states that the signal frequency must be less than or equal to one half the sampling frequency to prevent aliasing**. Aliasing is a condition in which frequency signals outside the desired signal band will, through the sampling process, appear within the bandwidth of interest. However, this aliasing process can be exploited in communications systems design to down-convert a high frequency signal to a lower frequency. This technique is known as under-sampling. A criterion for under-sampling is that the ADC has sufficient input bandwidth and dynamic range to acquire the highest frequency signal of interest.

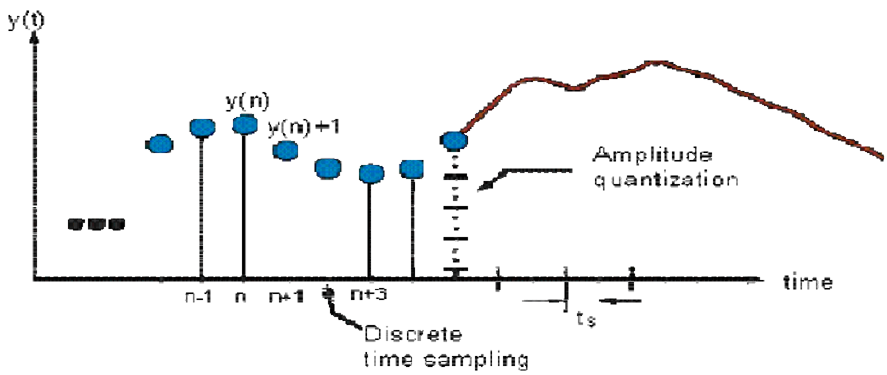


Figure 20.5 Sampling Process

Sampling and quantization are important concepts because they establish the performance limits of an ideal ADC. In an ideal ADC, the code transitions are exactly 1 least significant bit (LSB) apart. So, for an N-bit ADC, there are 2N codes and 1 LSB = FS/2N, where FS is the full-scale analog input voltage. However, ADC operation in the real world is also affected by non-ideal effects, which produce errors beyond those dictated by converter resolution and sample rate. These errors are reflected in a number of AC and DC performance specifications associated with ADCs.

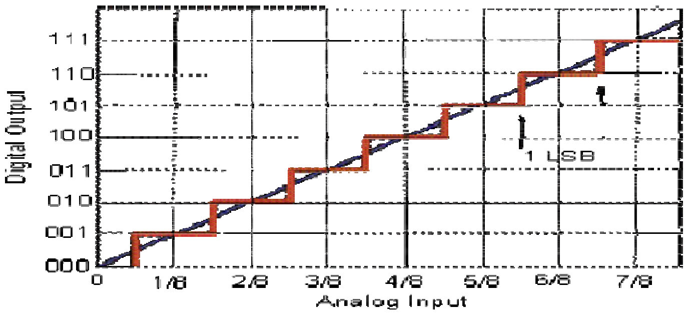


Figure 20.6 Transfer Function for an Ideal ADC

Decimation – Reduce the sampling rate of a discrete-time signal. – Low sampling rate reduces storage and computation requirements. • **Interpolation** – Increase the sampling rate of a discrete-time signal. – Higher sampling rate preserves fidelity.

Moving averages smooth the price data to form a trend following indicator. They do not predict price direction, but rather define the current direction, though they lag due to being based on past prices. Despite this, moving averages help smooth price action and filter out the noise. They also form the building blocks for many other technical indicators and overlays, such as Bollinger Bands, MACD and the McClellan Oscillator. The two most popular types of moving averages are the **Simple Moving Average (SMA)** and the **Exponential Moving Average (EMA)**. These moving averages can be used to identify the direction of the trend or define potential support and resistance levels.

Here's a chart with both an SMA and an EMA on it:

Simple Moving Average Calculation

A simple moving average is formed by computing the average price of a security over a specific number of periods. Most moving averages are based on closing prices; for example, a 5-day simple moving average is the five-day sum of closing prices divided by five. As its name implies, a moving average is an average that moves. Old data is dropped as new data becomes available, causing the average to move along the time scale. The example below shows a 5-day moving average evolving over three days.

Daily Closing Prices: 11,12,13,14,15,16,17

First day of 5-day SMA: (11 + 12 + 13 + 14 + 15) / 5 = 13

Second day of 5-day SMA: (12 + 13 + 14 + 15 + 16) / 5 = 14

Third day of 5-day SMA: (13 + 14 + 15 + 16 + 17) / 5 = 15

The first day of the moving average simply covers the last five days. The second day of the moving average drops the first data point (11) and adds the new data point (16). The third day of the moving average continues by dropping the first data point (12) and adding the new data point (17). In the example above, prices gradually increase from 11 to 17 over a total of seven days. Notice that the moving average also rises from 13 to 15 over a three-day calculation period. Also, notice that each moving average value is just below the last price. For example, the moving average for day one equals 13 and the last price is 15. Prices the prior four days were lower and this causes the moving average to lag.

Periodic Signals

A periodic signal is one that repeats the sequence of values exactly after a fixed length of time, known as the period. In mathematical terms a signal x(t) is periodic if there is a number T such that for all t Equation holds the following.

$x(t)=x(t+T)$

The smallest positive number T that satisfies Equation above is the period and it defines the duration of one complete cycle. The fundamental frequency of a periodic signal is given by Equation below

$f=1/T$

It is important to distinguish between the real signal and the quantitative representation, which is necessarily an approximation. The amount of error in the approximation depends on the complexity of the signal, with simple waveforms, such as the sinusoid, having less error than complex waveforms.

A harmonic is a signal or wave whose frequency is an integral (whole-number) multiple of the frequency of some reference signal or wave. The term can also refer to the ratio of the frequency of such a signal or wave to the frequency of the reference signal or wave.

Let *f* represent the main, or fundamental, frequency of an alternating current (AC) signal, electromagnetic field , or sound wave. This frequency, usually expressed in hertz , is the frequency at which most of the energy is contained, or at which the signal is defined to occur. If the signal is displayed on an oscilloscope, the waveform will appear to repeat at a rate corresponding to *f* Hz.

For a signal whose fundamental frequency is f, the second harmonic has a frequency $2f$, the third harmonic has a frequency of $3f$, and so on. Let w represent the wavelength of the signal or wave in a specified medium. The second harmonic has a wavelength of $w/2$, the third harmonic has a wavelength of $w/3$, and so on. Signals occurring at frequencies of $2f$, $4f$, $6f$, etc. are called even harmonics; the signals at frequencies of $3f$, $5f$, $7f$, etc. are called odd harmonics. A signal can, in theory, have infinitely many harmonics.

Nearly all signals contain energy at harmonic frequencies, in addition to the energy at the fundamental frequency. If all the energy in a signal is contained at the fundamental frequency, then that signal is a perfect sine wave. If the signal is not a perfect sine wave, then some energy is contained in the harmonics. Some waveforms contain large amounts of energy at harmonic frequencies. Examples are square waves, sawtooth waves, and triangular waves.

IIR vs FIR Filters

IIR filters are difficult to control and have no particular phase, whereas FIR filters make a linear phase always possible. IIR can be unstable, whereas FIR is always stable. IIR, when compared to FIR, can have limited cycles, but FIR has no limited cycles. IIR is derived from analog, whereas FIR has no analog history. IIR filters make polyphase implementation possible, whereas FIR can always be made casual.

FIR filters are helpful to achieve fractional constant delays. #MAD stands for a number of multiplications and additions, and is used as a criterion for an IIR and FIR filter comparison. IIR filters require more #MAD when compared to FIR, because FIR is of a higher order in comparison to IIR, which is of lower order, and uses polyphase structures.

FIR filters are dependent upon linear-phase characteristics, whereas IIR filters are used for applications which are not linear. FIR's delay characteristics is much better, but they require more memory. On the other hand, IIR filters are dependent on both i/p and o/p, but FIR is dependent upon i/p only. IIR filters consist of zeros and poles, and require less memory than FIR filters, whereas FIR only consists of zeros. IIR filters can become difficult to implement, and also delay and distort adjustments can alter the poles & zeroes, which make the filters unstable, whereas FIR filters remain stable. FIR filters are used for tapping of a higher-order, and IIR filters are better for tapping of lower-orders, since IIR filters may become unstable with tapping higher-orders.

Analog to Digital Converter

From the name itself it is clear that it is a converter which converts the analog (continuously variable) signal to digital signal. This is really an electronic integrated circuit which directly converts the continuous form of signal to discrete form. It can be expressed as A/D or A-to-D or A-D or ADC. The input (analog) to this system can have any value in a range and are directly measured. But for output (digital) of an N-bit A/D converter, it should have only 2^N discrete values. This **A/D converter** is a linkage between the analog (linear) world of transducers and discreet world of processing the signal and handling the data. The digital to analog converter (DAC) carry out the inverse function of the ADC. The schematic representation of ADC is shown below.

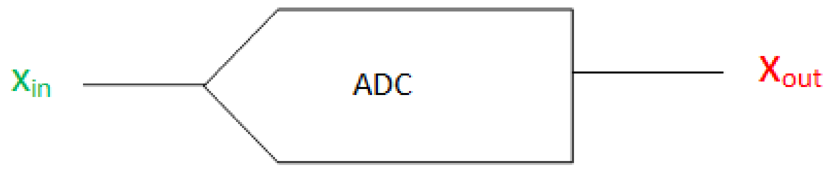


Figure 1

ADC Process

There are mainly two steps involves in the process of conversion. They are

- Sampling and Holding
- Quantizing and Encoding

The whole ADC conversion process is shown in figure 2.

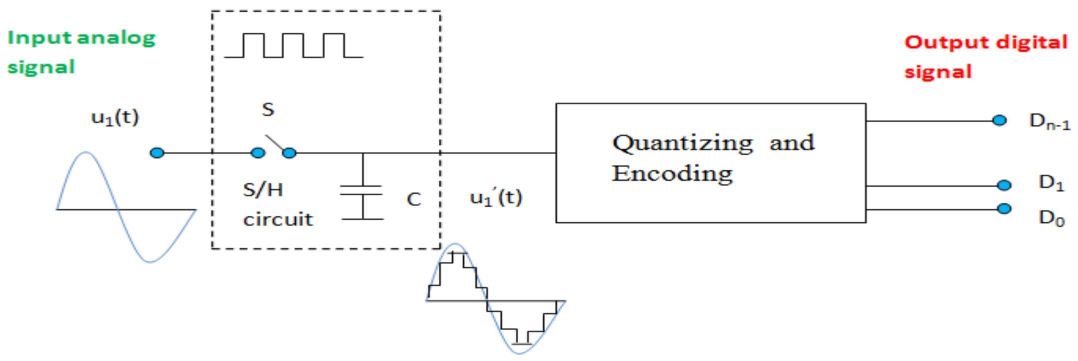


Figure 2

Sampling and Holding

In the process of Sample and hold (S/H), the continuous signal will gets sampled and freeze (hold) the value at a steady level for a particular least period of time. It is done to remove variations in input signal which can alter the conversion process and thereby increases the accuracy. The minimum sampling rate has to be two times the maximum data frequency of the input signal.

Quantizing and Encoding

For understanding quantizing, we can first go through the term Resolution used in ADC. It is the smallest variation in analog signal that will result in a variation in the digital output. This actually represents the quantization error.

$$Resolution, \triangle V = \frac{V_r}{2^N}$$

V → Reference voltage range

2^N → Number of states

N → Number of bits in digital output

Quantizing: It is the process in which the reference signal is partitioned into several discrete quanta and then the input signal is matched with the correct quantum.

Encoding: Here; for each quantum, a unique digital code will be assigned and after that the input signal is allocated with this digital code. The process of quantizing and encoding is demonstrated in the table below.

Analog signal			Digital o/p
7.5	7	7Δ=7V	111
6.5	6	6Δ=6V	110
5.5	5	5Δ=5V	101
4.5	4	4Δ=4V	100
3.5	3	3Δ=3V	011
2.5	2	2Δ=2V	010
1.5	1	1Δ=1V	001
0.5	0	0Δ=0V	000

From the above table we can observe that only one digital value is used to represent the whole range of voltage in an interval. Thus, an error will occur and it is called quantization error. This is the noise introduced by the process of quantization. Here the maximum quantization error is $\pm \frac{1}{2}\Delta V = \pm 0.5V$

Improvement of Accuracy in ADC

Two important methods are used for improving the accuracy in ADC. They are by increasing the resolution and by increasing the sampling rate. This is shown in figure below (figure 3).

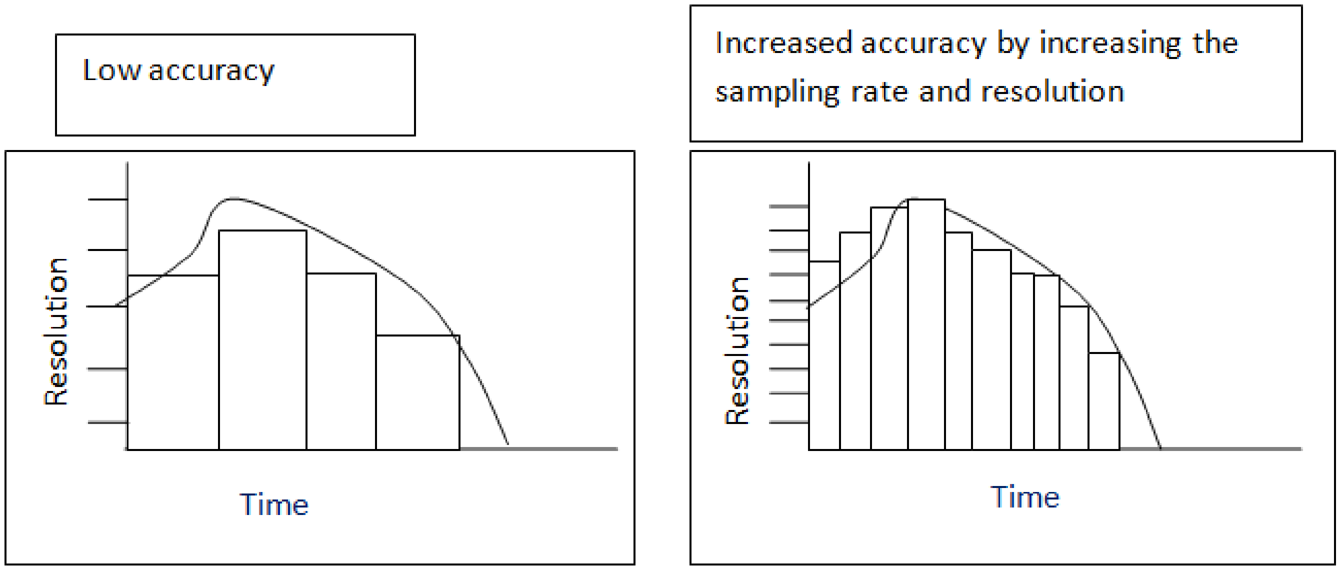


Figure 3

Types of Analog to Digital Converter

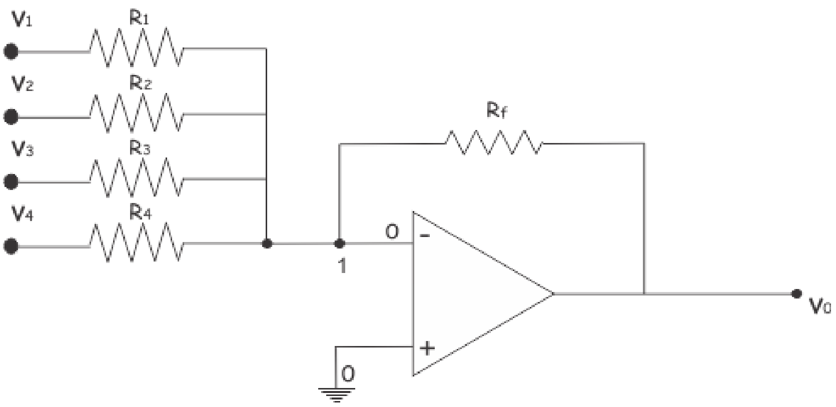
- **Successive Approximation ADC:** This converter compares the input signal with the output of an internal DAC at each successive step. It is the most expensive type.
- **Dual Slope ADC:** It have high accuracy but very slow in operation.
- **Pipeline ADC:** It is same as that of two step Flash ADC.
- **Delta-Sigma ADC:** It has high resolution but slow due to over sampling.
- **Flash ADC:** It is the fastest ADC but very expensive.
- **Other:** Staircase ramp, Voltage-to-Frequency, Switched capacitor, tracking, Charge balancing, and resolver.

Application of ADC

- Used together with the transducer.
- Used in computer to convert the analog signal to digital signal.
- Used in cell phones.
- Used in microcontrollers.
- Used in digital signal processing.
- Used in digital storage oscilloscopes.
- Used in scientific instruments.
- Used in music reproduction technology etc.

Digital to Analog Converter or DAC

Op amp is extensively used as main building block of **digital to analog convertor**. **Digital to analog convertor** is an electronics device in form of IC, which converts digital signal to its equivalent analog signal. The **DAC** can be realized in many ways. One of the popular **digital to analog convertor** circuit is **binary weighted ladder**. This is basically a summing amplifier designed with suitable resistances, as shown below.



Now, applying Kirchhoff Current Law at node 1 of the above circuit, we get,

$$\frac{v_1}{R_1} + \frac{v_2}{R_2} + \frac{v_3}{R_3} + \frac{v_4}{R_4} = -\frac{v_0}{R_f}$$

$$\Rightarrow -v_0 = \frac{R_f}{R_1}v_1 + \frac{R_f}{R_2}v_2 + \frac{R_f}{R_3}v_3 + \frac{R_f}{R_4}v_4 \cdots\cdots(i)$$

Before going through the above circuit of **digital to analog convertor**, Let us put some suitable values of different resistors connected in the circuit. Such as, R_f = 10KΩ, R_1 = 10KΩ, R_2 = 20KΩ, R_3 = 40KΩ and R_4= 80KΩ.

Putting these values in equation (i) we get,

$$-v_0 = v_1 + 0.5v_2 + 0.25v_3 + 0.125v_4$$

Now, let us also apply voltage at input terminals either 0 or 1 volt. Putting , 0 volt at all inputs,(i.e. v_1 = 0, v_2 = 0, v_3 = 0 and v_4 = 0) we get,

$$-v_0 = 0 + 0.5 \times 0 + 0.25 \times 0 + 0.125 \times 0$$

$$\Rightarrow v_0 = 0\ v$$

So, for digital input 0000, we get analog output 0 volt. Putting, 1V at last input only, (i.e. v_1 = 0, v_2 = 0, v_3 = 0 and v_4 = 1V), we get,

$$-v_0 = 0 + 0.5 \times 0 + 0.25 \times 0 + 0.125 \times 1$$

$$\Rightarrow v_0 = -0.125\ volt$$

Similarly, for $v_1 = 0$, $v_2 = 0$, $v_3 = 1$, $v_4 = 0$

$$-v_0 = 0 + 0.5 \times 0 + 0.25 \times 1 + 0.125 \times 0$$

$$\Rightarrow v_0 = -0.25\ volt$$

For, $v_1 = 0$, $v_2 = 0$, $v_3 = 1$, $v_4 = 1$

$$-v_0 = 0 + 0.5 \times 0 + 0.25 \times 1 + 0.125 \times 1$$

$$\Rightarrow v_0 = -0.375\ volt$$

In this way the inputs and corresponding outputs can be represented in a table as shown below.

Binary Input [v_1 v_2 v_3 v_4]	Decimal Value	Output($-v_0$)
0000	0	0
0001	1	0.125
0010	2	0.25
0011	3	0.375
0100	4	0.5
0101	5	0.625
0110	6	0.75
0111	7	0.875
1000	8	1.0
1001	9	1.125
1010	10	1.25

1011	11	1.375
1100	12	1.5
1101	13	1.625
1110	14	1.75
1111	15	1.875

So, for each decimal number there is one unique output voltage level. From the table it is also seen that, form 0 to 15, for each increment there is an increase of output voltage level by 0.125 volt. So, the output is analog and it is linearly proportional the decimal equivalent of digital inputs. The above example was of a four bit **DAC**. A four bit **DAC** can be represented as shown below.

Sampling Theory

The signals we use in the real world, such as our voices, are called "analog" signals. To process these signals in computers, we need to convert the signals to "digital" form. While an analog signal is continuous in both time and amplitude, a digital signal is discrete in both time and amplitude. To convert a signal from continuous time to discrete time, a process called sampling is used. The value of the signal is measured at certain intervals in time. Each measurement is referred to as a sample. (The analog signal is also quantized in amplitude, but that process is ignored in this demonstration. See the Analog to Digital Conversion page for more on that.)

When the continuous analog signal is sampled at a frequency F, the resulting discrete signal has more frequency components than did the analog signal. To be precise, the frequency components of the analog signal are repeated at the sample rate. That is, in the discrete frequency response they are seen at their original position, and are also seen centered around +/- F, and around +/- 2F, etc.

How many samples are necessary to ensure we are preserving the information contained in the signal? If the signal contains high frequency components, we will need to sample at a higher rate to avoid losing information that is in the signal. In general, to preserve the full information in the signal, it is necessary to sample at twice the maximum frequency of the signal. This is known as the Nyquist rate. The Sampling Theorem states that a signal can be exactly reproduced if it is sampled at a frequency F, where F is greater than twice the maximum frequency in the signal.

What happens if we sample the signal at a frequency that is lower that the Nyquist rate? When the signal is converted back into a continuous time signal, it will exhibit a phenomenon called ***aliasing***. Aliasing is the presence of unwanted components in the reconstructed signal. These components were not present when the original signal was sampled. In addition, some of the frequencies in the original signal may be lost in the reconstructed signal. Aliasing occurs because signal frequencies can overlap if the sampling frequency is too low. Frequencies "fold" around half the sampling frequency - which is why this frequency is often referred to as the folding frequency.

Sometimes the highest frequency components of a signal are simply noise, or do not contain useful information. To prevent aliasing of these frequencies, we can filter out these components before sampling the signal. Because we are filtering out high frequency components and letting lower frequency components through, this is known as low-pass filtering.

Demonstration of Sampling

The original signal in the applet below is composed of three sinusoid functions, each with a different frequency and amplitude. The example here has the frequencies 28 Hz, 84 Hz, and 140 Hz. Use the filtering control to filter out the higher frequency components. This filter is an ideal low-pass filter, meaning that it exactly preserves any frequencies below the cutoff frequency and completely attenuates any frequencies above the cutoff frequency.

Notice that if you leave all the components in the original signal and select a low sampling frequency, aliasing will occur. This aliasing will result in the reconstructed signal not matching the original signal. However, you can try to limit the amount of aliasing by filtering out the higher frequencies in the signal. Also important to note is that once you are sampling at a rate above the Nyquist rate, further increases in the sampling frequency do not improve the quality of the reconstructed signal. This is true because of the ideal low-pass filter. In real-world applications, sampling at higher frequencies results in better reconstructed signals. However, higher sampling frequencies require faster converters and more storage. Therefore, engineers must weigh the advantages and disadvantages in each application, and be aware of the tradeoffs involved.

The importance of frequency domain plots in signal analysis cannot be understated. The three plots on the right side of the demonstration are all Fourier transform plots. It is easy to see the effects of changing the sampling frequency by looking at these transform plots. As the sampling frequency decreases, the signal separation also decreases. When the sampling frequency drops below the Nyquist rate, the frequencies will crossover and cause aliasing.