Module 3

Embedded Systems I/O

Lesson 18

AD and DA Converters

Instructional Objectives

After going through this lesson the student would be able to

- Learn about Real Time Signal Processing
- Sampling Theorem
- DA Conversion
- Different Methods of AD Conversions
    - Successive Approximation
    - Flash
    - Sigma Delta

Pre-Requisite

Digital Electronics, Microprocessors

18 Introduction

The real time embedded controller is expected to process the real world signals within a specified time. Most of the real world signals are analog in nature. Take the examples of your mobile phone. The overall architecture is shown on Fig.18.1. The Digital Signal Processor (DSP) is fed with the analog data from the microphone. It also receives the digital signals after demodulation from the RF receiver and generates the filtered and noise free analog signal through the speaker. All the processing is done in real time.

The processing of signals in real time is termed as Real Time Signal Processing which has been coined beautifully in the Signal Processing industry.

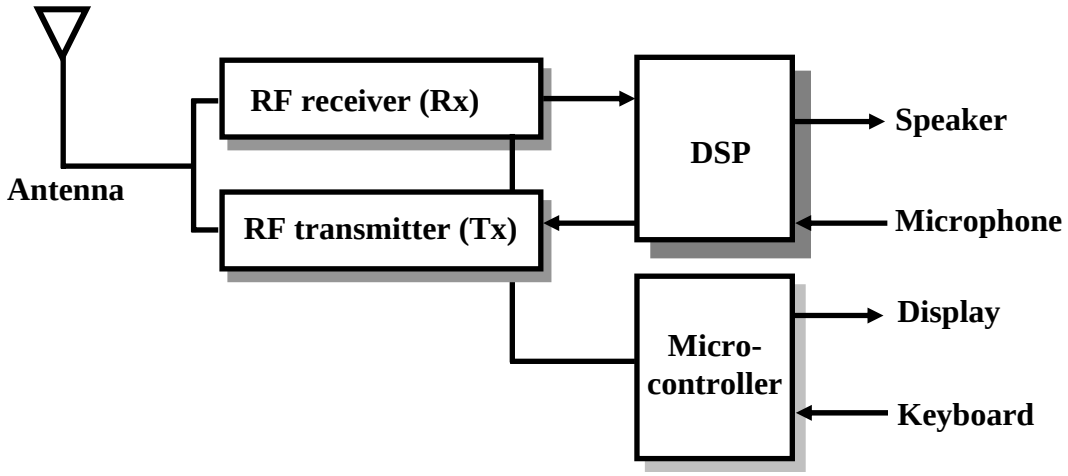


Fig. 18.1 The block diagram

The detailed steps of such a processing task is outlined in Fig.18.2

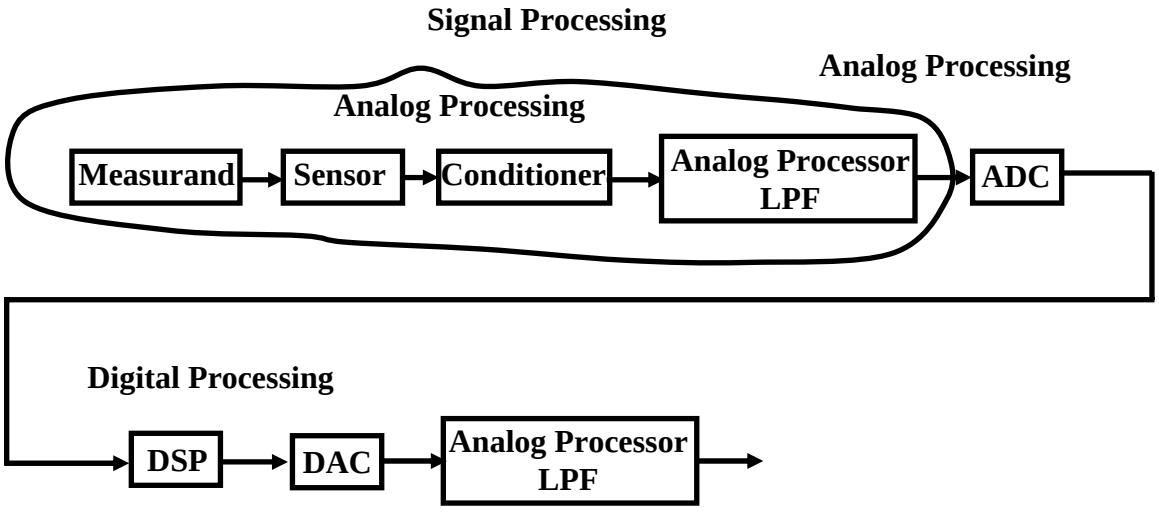


Fig. 18.2 Real Time Processing of Analog Signals

Measurand is the quantity which is measured. In this case it is the analog speech signal. The sensor is a microphone. In case of your mobile set it is the microphone which is embedded in it. The conditioner can be a preamplifier or a demodulator. The Analog Processor mostly is a Low Pass Filter (LPF). This is primarily used to prevent *aliasing* which is a term to be explained later in this chapter. The following is the Analog to Digital Converter which has a number of stages to convert an analog signal into digital form. The Digital Signal Processing is carried out by a system with a processor. Further the processed signal is converted into analog signal by the Digital to Analog Converter which finally sends the output to the real world through another Low Pass Filter.

The functional layout of the ADC and DAC is depicted in Fig.18.3

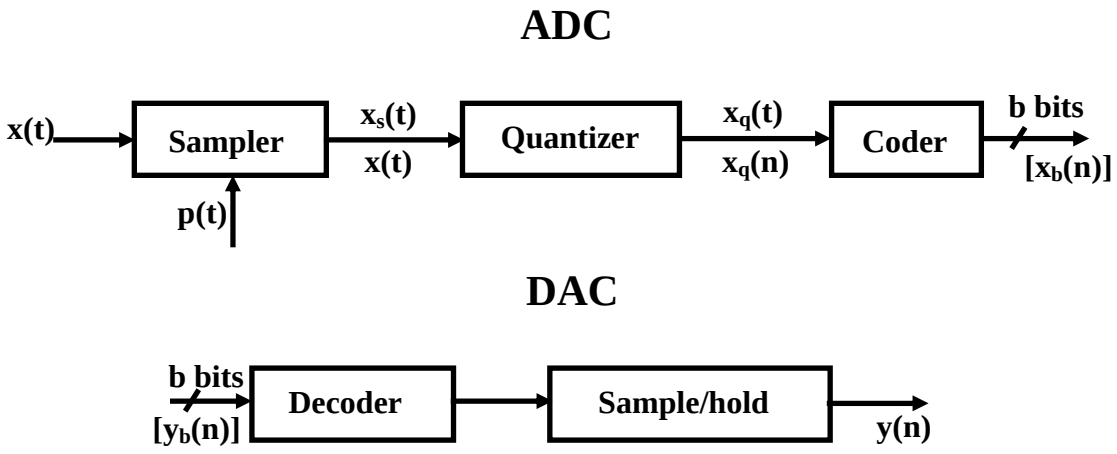


Fig. 18.3 The functional layout of the ADC and DAC

The DA Converter

In theory, the simplest method for digital-to-analog conversion is to pull the samples from memory and convert them into an impulse train.

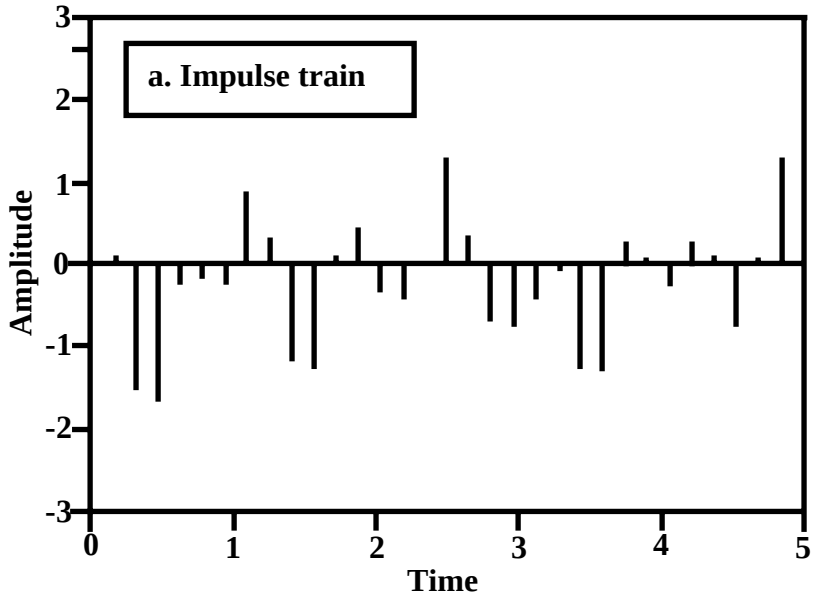


Fig. 18.4(a) The analog equivalent of digital words

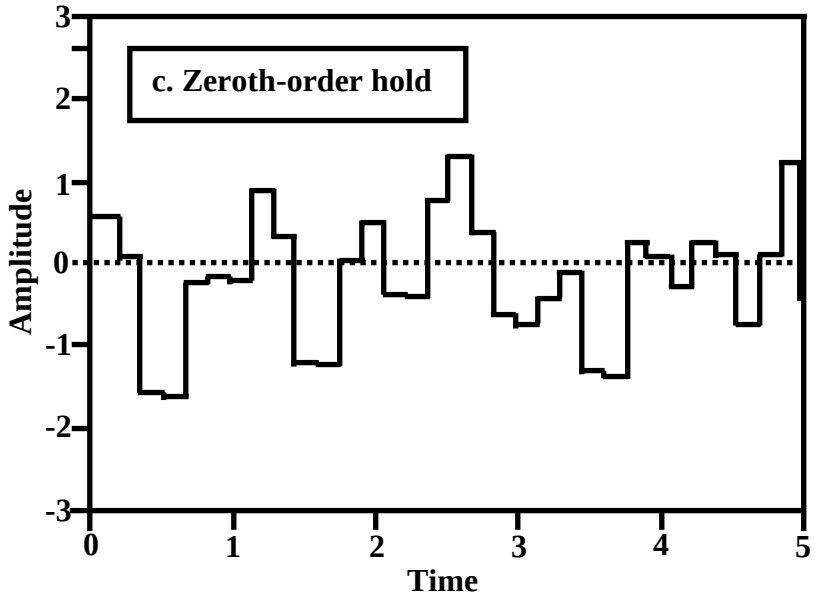


Fig. 18.4(b) The analog voltage after zero-order hold

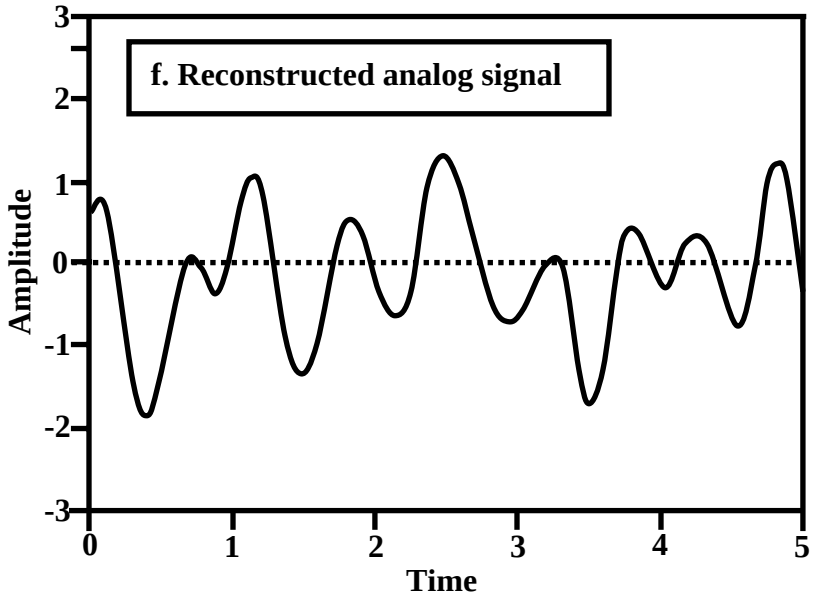
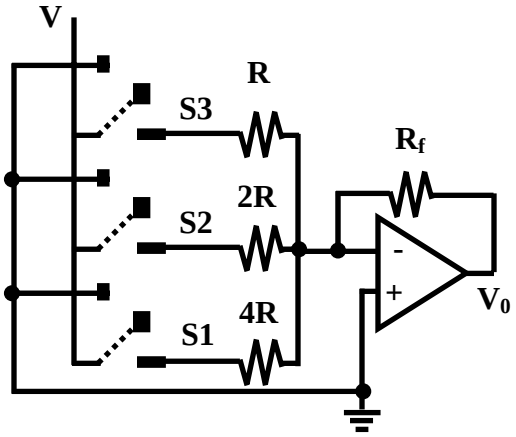


Fig. 18.4(c) The reconstructed analog signal after filtering

A digital word (8-bits or 16-bits) can be converted to its analog equivalent by weighted averaging. Fig. 18.5(a) shows the weighted averaging method for a 3-bit converter.

A switch connects an input either to a common voltage V or to a common ground. Only switches currently connected to the voltage source contribute current to the non-inverting input summing node. The output voltage is given by the expression drawn below the circuit diagram; SX = 1 if switch X connects to V, SX = 0 if it connects to ground. There are eight possible combinations of connections for the three switches, and these are indicated in the columns of the table to the right of the diagram. Each combination is associated with a decimal integer as shown. The inputs are weighted in a 4:2:1 relationship, so that the sequence of values for 4S3 + 2S2 + S1 form a binary-coded decimal number representation. The magnitude of Vo varies in units (steps) of (Rf/4R)V from 0 to 7. This circuit provides a simplified Digital to Analog Converter (DAC). The digital input controls the switches, and the amplifier provides the analog output.

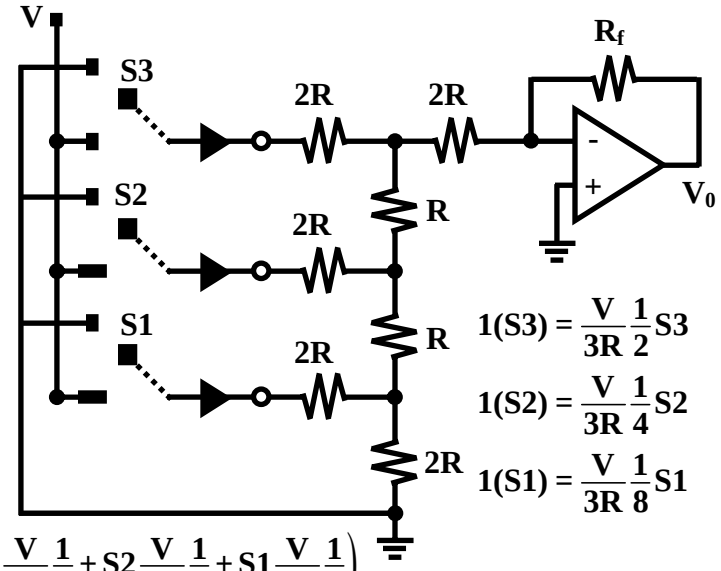


	S3	S2	S1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

$$V0 = -R_f\left(S3\frac{V}{R} + S2\frac{V}{2R} + S1\frac{V}{4R}\right)$$

$$= \frac{-R_f}{4R}V(4S3 + 2S2 + S1)$$

Fig. 18.5(a) The binary weighted register method



$$1(S3) = \frac{V}{3R}\frac{1}{2}S3$$

$$1(S2) = \frac{V}{3R}\frac{1}{4}S2$$

$$1(S1) = \frac{V}{3R}\frac{1}{8}S1$$

$$V0 = -R_f\left(S3\frac{V}{3R}\frac{1}{2} + S2\frac{V}{3R}\frac{1}{4} + S1\frac{V}{3R}\frac{1}{8}\right)$$

$$= -R_f\frac{V}{24R}(4S3 + 2S2 + S1)$$

Fig. 18.5(b) R-2R ladder D-A conversion circuit

Fig. 18.5(b) depicts the R-2R ladder network. The disadvantage of the binary weighted register is the availability and manufacturing of exact values of the resistances. Here also the output is proportional to the binary-coded decimal number.

The output of the above circuits as given in Fig. 18.5(a) and 18.5(b) is equivalent analog values as shown in Fig. 18.4(a). However to reconstruct the original signal this is further passed through a zero order hold (ZOH) circuit followed by a filter (Fig.18.2). The reconstructed waveforms are shown in Fig. 18.4(b) and 18.4(c).

The AD Converter

The ADC consists of a sampler, quantizer and a coder. Each of them is explained below.

Sampler

The sampler in the simplest form is a semiconductor switch as shown below. It is followed by a hold circuit which a capacitor with a very low leakage path.

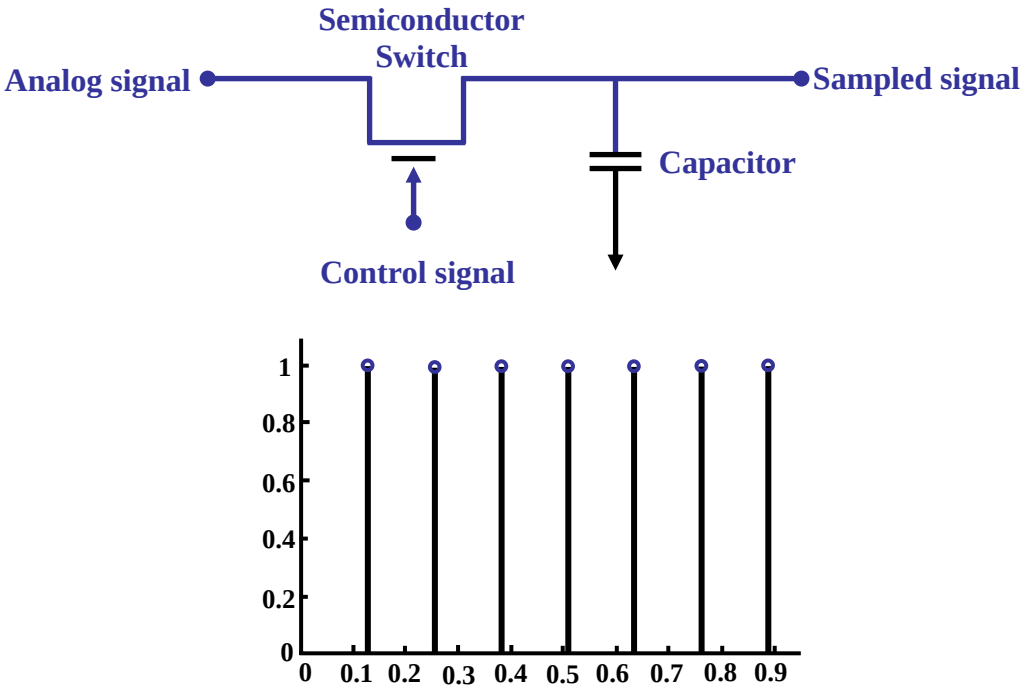


Fig. 18.6 The Sample and Hold Circuit

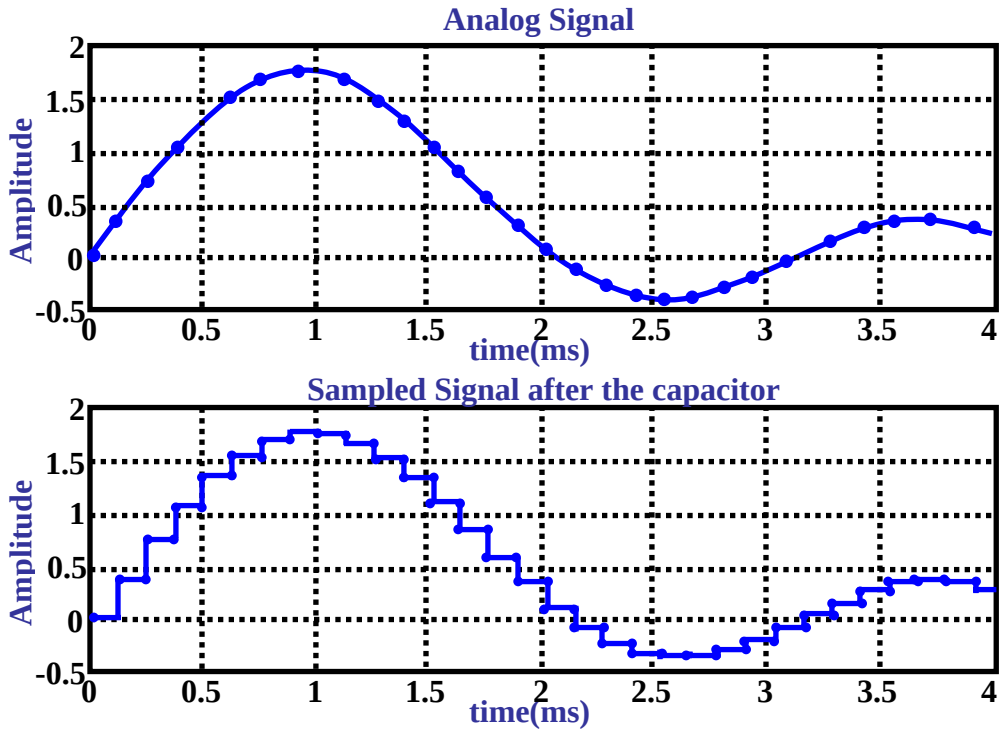


Fig. 18.7 Sample and Hold Signals

Quantizer

The hold circuit tries to maintain a constant voltage till the next switching. The quantizer is responsible to convert this voltage to a binary number. The number of bits in a binary number decides the approximation and accuracy.

The sample hand hold output can assume any real number in a given range. However because of finite number of bits (*say N*) the levels possible in the digital domain 0 to 2^{N-1} which corresponds to a voltage range of 0 to *V* volts

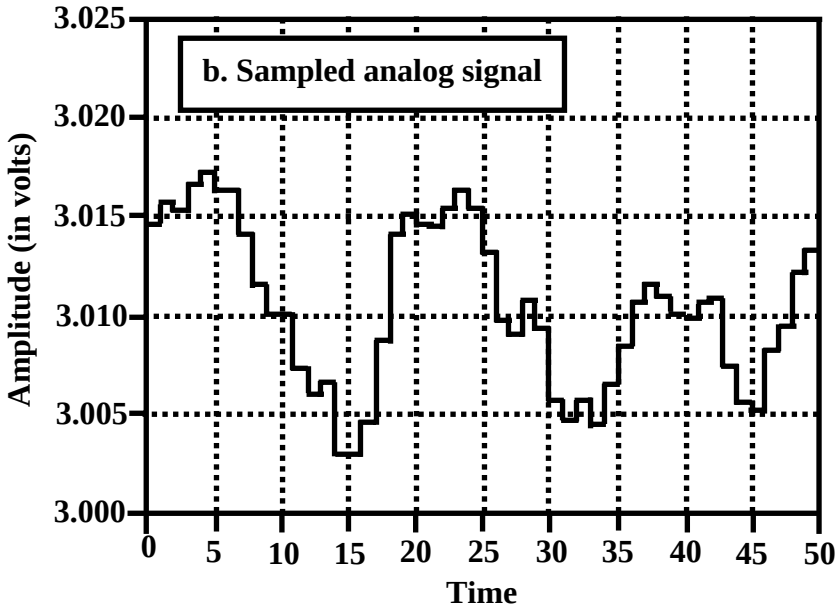


Fig. 18.8(a) Hold Circuit Output

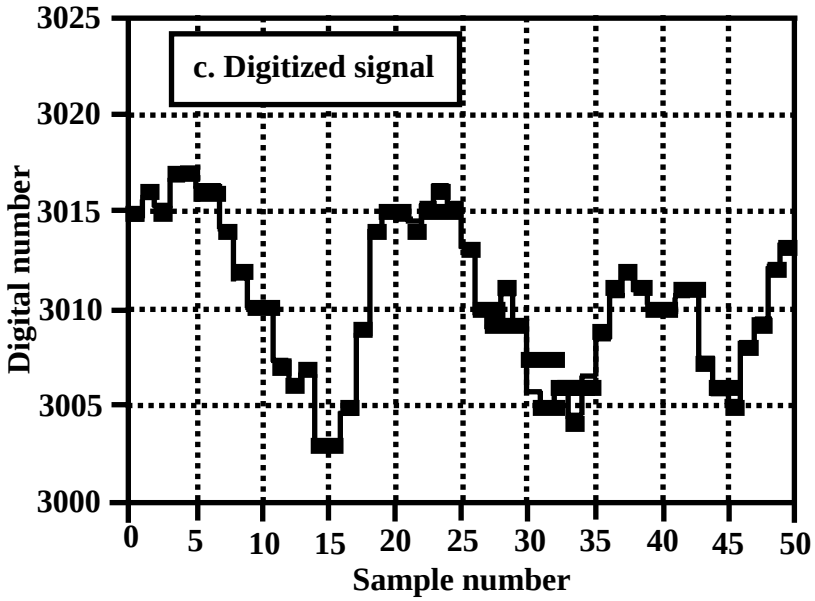


Fig. 18.8(b) The Quantized Value

Coder

This is an optional device which is used after the conversion is complete. In microprocessor based systems the Coder is responsible for packing several samples and transmitting them onwards either in synchronous or in asynchronous manner. For example in TI DSK kits you will find the AD converters with CODECs are interfaced to McBSP ports (short form of Multi-channel Buffered Serial Ports). Several 16-bit sampled values are packed into a frame and transmitted to the processor or to the memory by Direct Memory Access (DMA). The Coder is responsible for controlling the ADC and transferring the Data quickly for processing. Sometimes the Codec is responsible for compressing several samples together and transmitting them. In your desktop computers you will find audio interfaces which can digitize and record your voice and store them in .wav format. Basically this AD conversion followed by coding. The wav format is the Pulse-Code-Modulated (PCM) format of the original digital voice samples.

The Sampling Theorem

The definition of proper sampling is quite simple. Suppose you sample a continuous signal in some manner. If you can exactly reconstruct the analog signal from the samples, you must have done the sampling properly. Even if the sampled data appears confusing or incomplete, the key information has been captured if you can reverse the process. Fig.18.9 shows several sinusoids before and after digitization. The continuous line represents the analog signal entering the ADC, while the square markers are the digital signal leaving the ADC. In (a), the analog signal is a constant DC value, a cosine wave of zero frequency. Since the analog signal is a series of straight lines between each of the samples, all of the information needed to reconstruct the analog signal is contained in the digital data. According to our definition, this is proper sampling.

The sine wave shown in (b) has a frequency of 0.09 of the sampling rate. This might represent, for example, a 90cycle/second sine wave being sampled at1000 samples/second. Expressed in

another way, there are 11.1 samples taken over each complete cycle of the sinusoid. This situation is more complicated than the previous case, because the analog signal cannot be reconstructed by simply drawing straight lines between the data points. Do these samples properly represent the analog signal? The answer is yes, because no other sinusoid, or combination of sinusoids, will produce this pattern of samples (within the reasonable constraints listed below). These samples correspond to only one analog signal, and therefore the analog signal can be exactly reconstructed. Again, an instance of proper sampling. In (c), the situation is made more difficult by increasing the sine wave's frequency to 0.31 of the sampling rate. This results in only 3.2 samples per sine wave cycle. Here the samples are so sparse that they don't even appear to follow the general trend of the analog signal. Do these samples properly represent the analog waveform? Again, the answer is yes, and for exactly the same reason. The samples are a unique representation of the analog signal. All of the information needed to reconstruct the continuous waveform is contained in the digital data. Obviously, it must be more sophisticated than just drawing straight lines between the data points. As strange as it seems, this is proper sampling according to our definition. In (d), the analog frequency is pushed even higher to 0.95 of the sampling rate, with a mere 1.05 samples per sine wave cycle. Do these samples properly represent the data? No, they don't! The samples represent a different sine wave from the one contained in the analog signal. In particular, the original sine wave of 0.95 frequency misrepresents itself as a sine wave of 0.05 frequency in the digital signal. This phenomenon of sinusoids changing frequency during sampling is called aliasing. Just as a criminal might take on an assumed name or identity (an alias), the sinusoid assumes another frequency that is not its own. Since the digital data is no longer uniquely related to a particular analog signal, an unambiguous reconstruction is impossible. There is nothing in the sampled data to suggest that the original analog signal had a frequency of 0.95 rather than 0.05. The sine wave has hidden its true identity completely; the perfect crime has been committed! According to our definition, this is an example of improper sampling. This line of reasoning leads to a milestone in DSP, the sampling theorem. Frequently this is called the Shannon sampling theorem, or the Nyquist Sampling theorem, after the authors of 1940s papers on the topic. The sampling theorem indicates that a continuous signal can be properly sampled, only if it does not contain frequency components above one-half of the sampling rate. For instance, a sampling rate of 2,000 samples/second requires the analog signal to be composed of frequencies below 1000 cycles/second. If frequencies above this limit are present in the signal, they will be aliased to frequencies between 0 and 1000 cycles/second, combining with whatever information that was legitimately there.

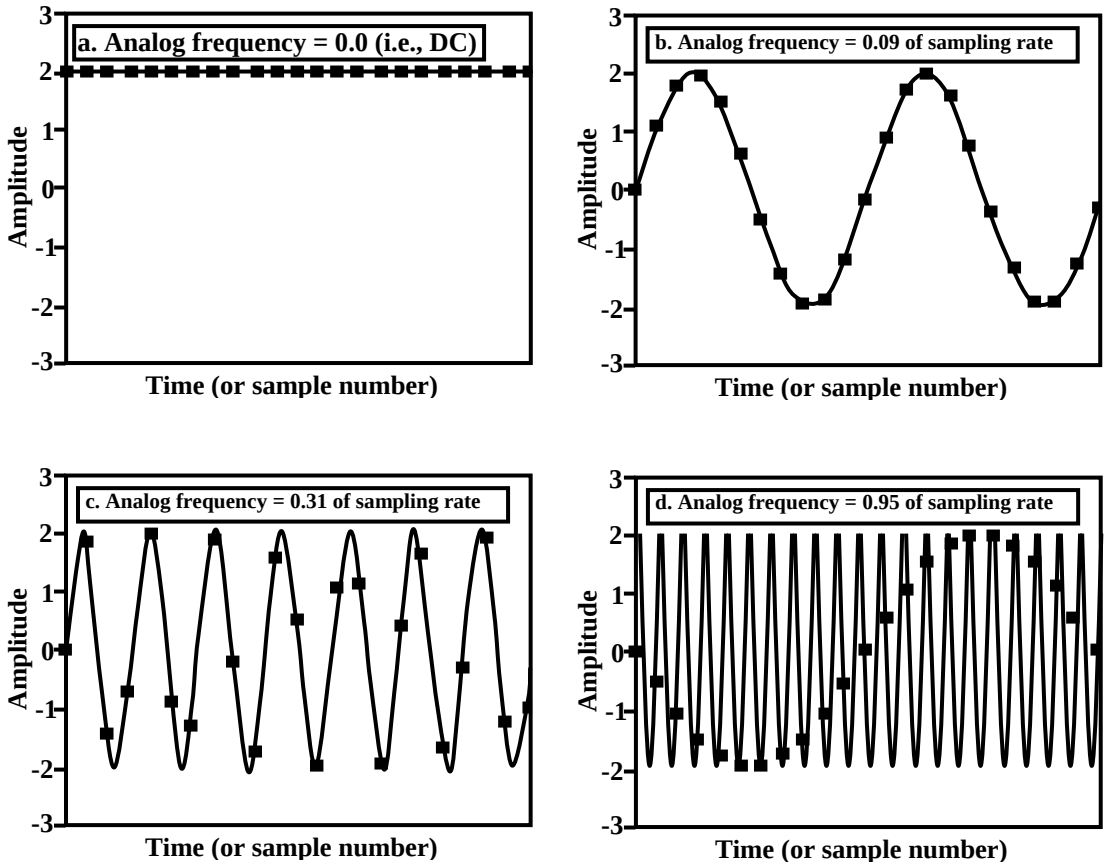


Fig. 18.9 Sampling a sine wave at different frequencies

Methods of AD Conversion

The analog voltage samples are converted to digital equivalent at the quantizer. There are various ways to convert the analog values to the nearest finite length digital word. Some of these methods are explained below.

Successive Approximation ADC

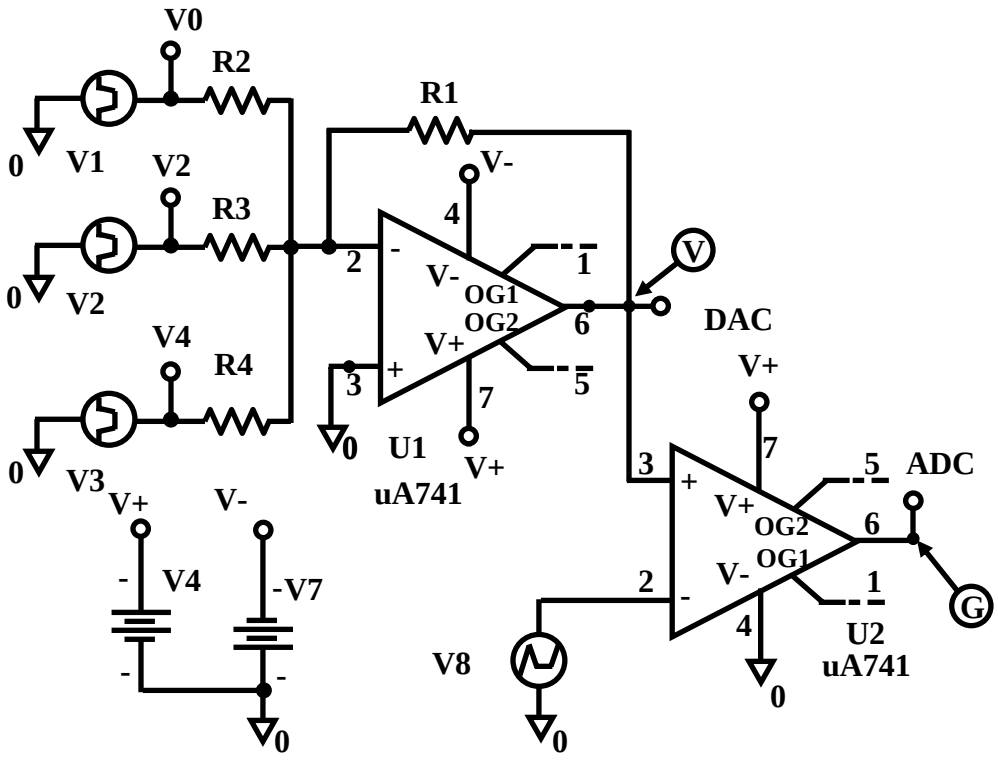


Fig. 18.10 The Counter Converter

The AD conversion is indirectly carried out through DA conversion. The 3-bit input as shown in Fig.18.10 to the DA converter may change sequentially from 000 to 111 by a 3-bit counter. The unknown voltage (V8) is applied to one input of the comparator. When the DA output exceeds the unknown voltage the comparator output which was negative becomes positive. This can be used to latch the counter value which is approximately equivalent digital value of the unknown voltage.

The draw back of sequential counting is the time taken to reach the highest count is large. For instance an 8-bit converter has to count 256 for converting the maximum input. It therefore has to consume 256 clock cycles which is large. Therefore, a different method called successive approximation is used for counting as shown in Fig.18.11.

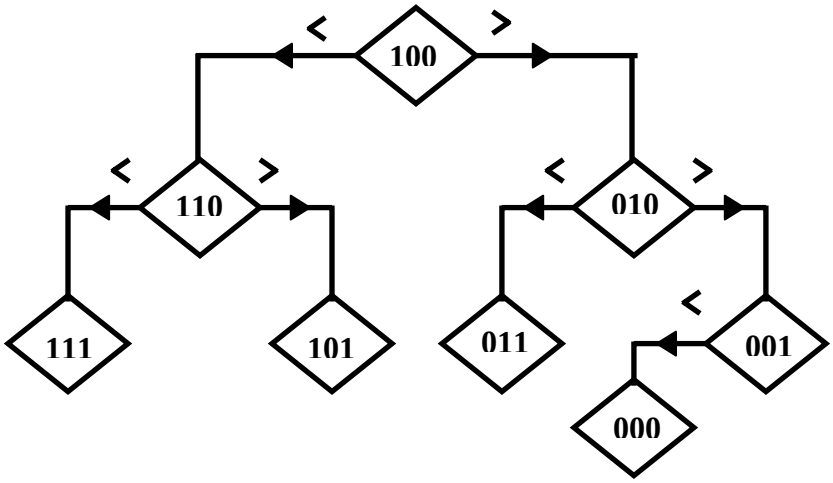


Fig. 18.11 The successive approximation counting

Consider a three-bit conversion for simplicity. The counting ADC must allow for up to eight comparisons (including zero). The search tree for an SAR search is illustrated in Fig.18.11. To start a conversion cycle a three-bit digital register is first cleared, and then loaded with the triplet 100. The register state provides the input to a DAC, and that provides a reference output. This output is compared to the analog signal to be converted, and a decision is made whether the analog signal is greater than or less than the reference signal. This comparison is essentially the same as that made for the previous ADC, except that because of the use of the 'half-way' code the result of this single comparison is used to eliminate concurrently half the possible DAC steps. As the tree suggests, if the analog signal is greater then all the smaller DAC outputs are eliminated from consideration. Digital logic associated with the comparison then either clears the MSB (Most Significant Bit) to 0 or simply leaves it unchanged. In either case the next bit is set to 1, i.e., to the mid-code of the selected half, and a new comparison made. Again half the remaining DAC states are eliminated from consideration. Depending on the result of the comparison the second bit is cleared to 0, or it is left unchanged at 1. In either case the third bit is set to 1 and the comparison step repeated. Each time a comparison is made half the remaining DAC output states will be eliminated. Instead of having to step through 2N states for an N bit conversion only N comparisons are needed. The SAR ADC is perhaps the most common of the converters, providing a relatively rapid and relatively inexpensive conversion

'Flash' Converter

Making all the comparisons between the digital states and the analog signal concurrently makes for a fast conversion cycle. A resistive voltage divider (see figure) can provide all the digital reference states required. There are eight reference values (including zero) for the three-bit converter illustrated. Note that the voltage reference states are offset so that they are midway between reference step values. The analog signal is compared concurrently with each reference state; therefore a separate comparator is required for each comparison. Digital logic then combines the several comparator outputs to determine the appropriate binary code to present.

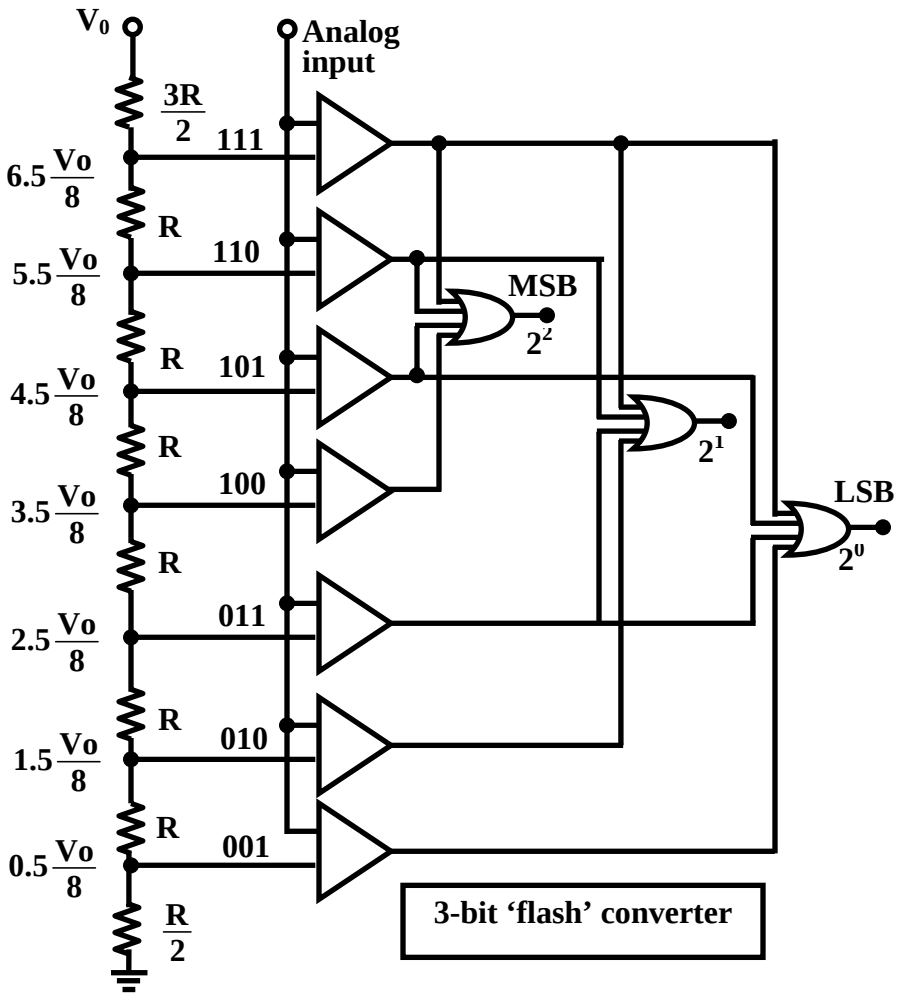


Fig. 18.12 Flash Converter

Sigma-Delta (ΣΔ) AD converters

The analog side of a sigma-delta converter (a 1-bit ADC) is very simple. The digital side, which is what makes the sigma-delta ADC inexpensive to produce, is more complex. It performs filtering and decimation. The concepts of over-sampling, noise shaping, digital filtering, and decimation are used to make a sigma-delta ADC.

Over-sampling

First, consider the frequency-domain transfer function of a traditional multi-bit ADC with a sine-wave input signal. This input is sampled at a frequency Fs. According to Nyquist theory, Fs must be at least twice the bandwidth of the input signal. When observing the result of an FFT analysis on the digital output, we see a single tone and lots of random noise extending from DC to Fs/2 (Fig.18.13). Known as quantization noise, this effect results from the following consideration: the ADC input is a continuous signal with an infinite number of possible states, but the digital output is a discrete function, whose number of different states is determined by the converter's

resolution. So, the conversion from analog to digital loses some information and introduces some distortion into the signal. The magnitude of this error is random, with values up to ±LSB.

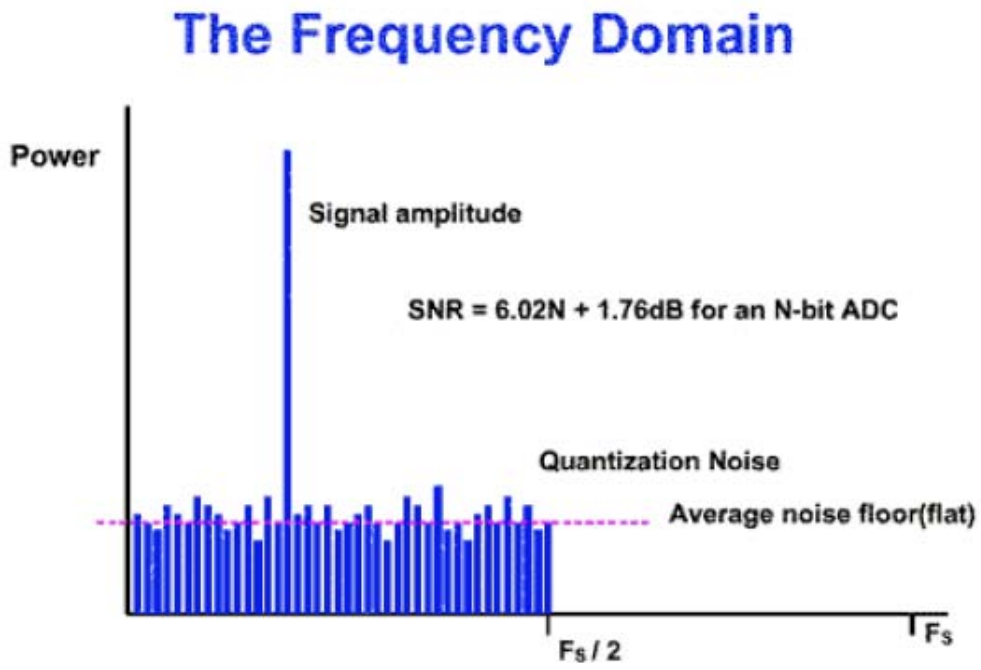


Fig. 18.13 FFT diagram of a multi-bit ADC with a sampling frequency F_S

If we divide the fundamental amplitude by the RMS sum of all the frequencies representing noise, we obtain the signal to noise ratio (SNR). For an N-bit ADC, SNR = 6.02N + 1.76dB. To improve the SNR in a conventional ADC (and consequently the accuracy of signal reproduction) you must increase the number of bits. Consider again the above example, but with a sampling frequency increased by the oversampling ratio k, to kFs (Fig.18.14). An FFT analysis shows that the noise floor has dropped. SNR is the same as before, but the noise energy has been spread over a wider frequency range. Sigma-delta converters exploit this effect by following the 1-bit ADC with a digital filter (Fig.18.14). The RMS noise is less, because most of the noise passes through the digital filter. This action enables sigma-delta converters to achieve wide dynamic range from a low-resolution ADC.

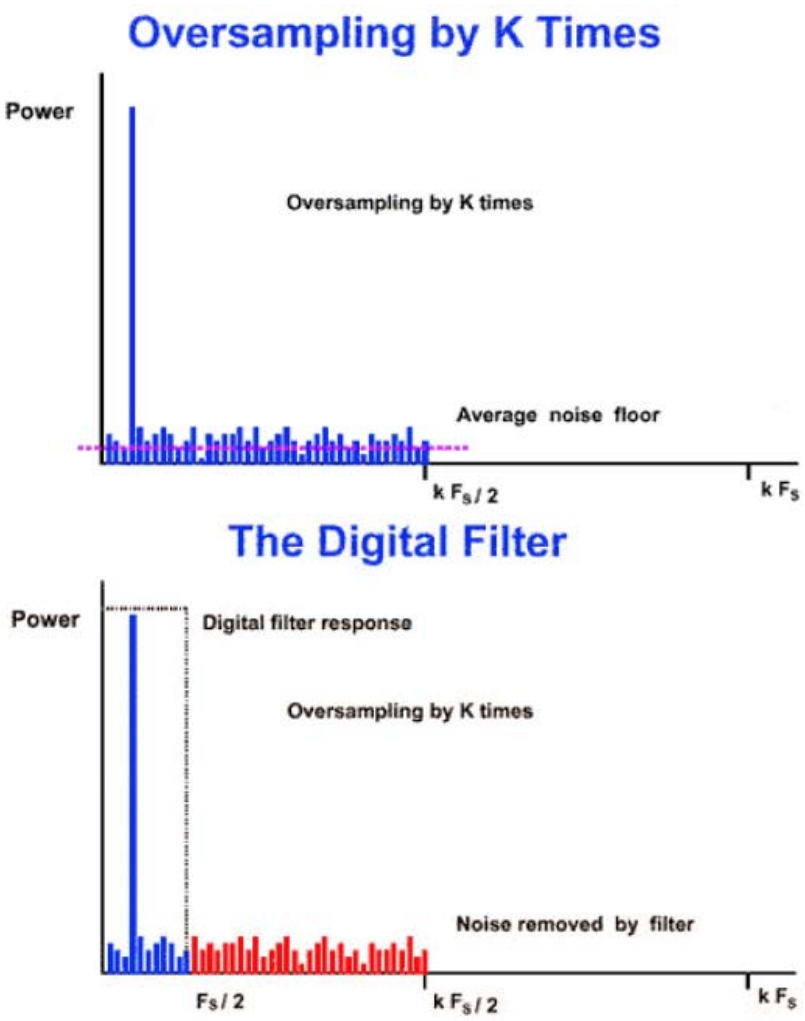


Fig. 18.14 FFT diagram of a multi-bit ADC with a sampling frequency kF_S and effect of Digital Filter on Noise Bandwidth

Noise Shaping

It includes a difference amplifier, an integrator, and a comparator with feedback loop that contains a 1-bit DAC. (This DAC is simply a switch that connects the negative input of the difference amplifier to a positive or a negative reference voltage.) The purpose of the feedback DAC is to maintain the average output of the integrator near the comparator's reference level. The density of "ones" at the modulator output is proportional to the input signal. For an increasing input the comparator generates a greater number of "ones," and vice versa for a decreasing input. By summing the error voltage, the integrator acts as a lowpass filter to the input signal and a highpass filter to the quantization noise. Thus, most of the quantization noise is pushed into higher frequencies. Oversampling has changed not the total noise power, but its distribution. If we apply a digital filter to the noise-shaped delta-sigma modulator, it removes more noise than does simple oversampling.(Fig.18.16).

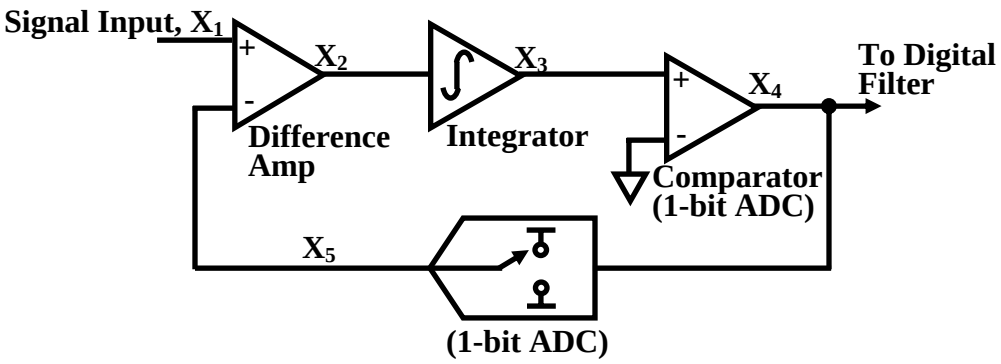


Fig. 18.15 Block Diagram of 1-bit Sigma Delta Converter

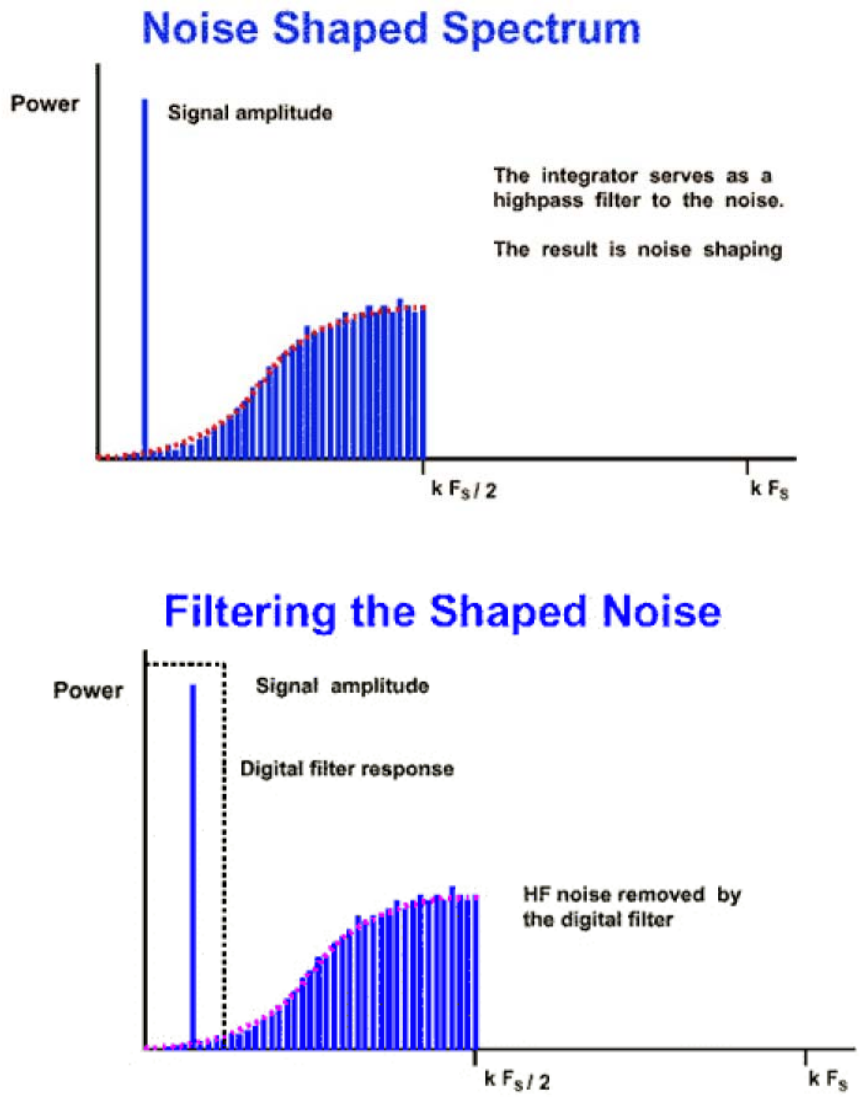


Fig. 18.16 The Effect of Integrator and Digital Filter on the Spectrum

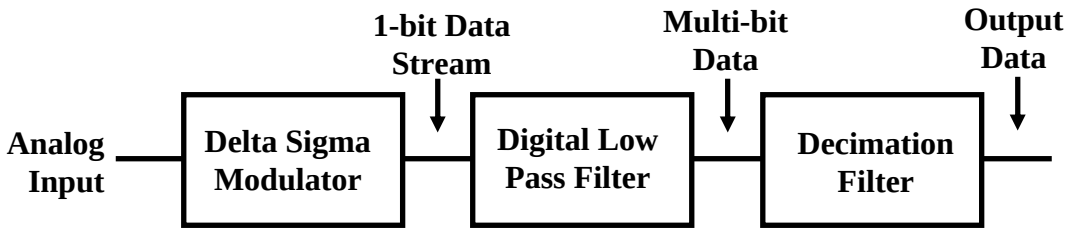


Fig. 18.17 The Digital Side of the Sigma-Delta modulator

Digital Filtering

The output of the sigma-delta modulator is a 1-bit data stream at the sampling rate, which can be in the megahertz range. The purpose of the digital-and-decimation filter (Fig.18.17) is to extract information from this data stream and reduce the data rate to a more useful value. In a sigma-delta ADC, the digital filter averages the 1-bit data stream, improves the ADC resolution, and removes quantization noise that is outside the band of interest. It determines the signal bandwidth, settling time, and stop band rejection.

Conclusion

In this chapter you have learnt about the basics of Real Time Signal Processing, DA and AD conversion methods. Some microcontrollers are already equipped with DA and AD converters on the same chip. Generally the real world signals are broad band. For instance a triangular wave though periodic will have frequencies ranging till infinite. Therefore anti-aliasing filter is always desirable before AD conversion. This limits the signal bandwidth and hence finite sampling frequency. The question answer session shall discuss about the quantization error, specifications of the AD and DA converters and errors at the various stages of real time signal processing. The details of interfacing shall be discussed in the next lesson.

The AD and DA converter fall under mixed VLSI circuits. The digital and analog circuits coexist on the same chip. This poses design difficulties for VLSI engineers for embedding fast and high resolution AD converters along with the processors. Sigma-Delta ADCs are most complex and hence rarely found embedded on microcontrollers.

Question Answers

Q1. What are the errors at different stages in a Real Time Signal Processing system? Elaborate on the quantization error.

Ans: *Refer to text*

Q2. What are the difference specifications of a D-A converter?

Ans: No. of bits (8-bits, 16-bits etc), Settling Time, Power Supply range, Power Consumption, Various Temperature ratings, Packaging

Q3. What are the various specifications of an A-D converter?

Ans: No. of bits (8-bits, 16-bits etc), No. of channels, Conversion Time, Power Supply range, Power Consumption, Various Temperature ratings, Packaging

Q4. How to construct a second order Delta-Sigma AD Converter.

Ans: *Refer to text and Fig.18.15*

Q5. What method you will adopt to digitize a slowly varying temperature signal without using AD converter?

Ans: Instead of AD Converters use Voltage to Frequency Converters followed by a counter

Source:

http://www.nptel.ac.in/courses/Webcourse-contents/IIT%20Kharagpur/Embedded%20systems/Pdf/Lesson-18.pdf