6 Digital Techniques and Software-Defined Radio

The advent of digital electronics, and computers, has revolutionised the world of radio. We can now perform nearly all the functions of a radio in the digital domain and functions that are difficult in the analogue domain can become easy in the digital domain. The key to all of this is the ability to transform radio signals back and forth between the digital and analogue domains through analogue-to-digital and digital-to-analogue converters. It is the advances in these converters that have turned radio from a technology of the analogue domain into one of modern computing. Not all analogue radio is redundant, since functions such as filtering, amplification and antennas will always be required. However, modern developments have brought the computer closer and closer to the antenna and have turned a large part of modern radio development into that of software development. In the current chapter we will discuss the elements of what has come to be known as *software-defined radio*.

6.1 Basic Digital Electronics

In the digital domain, a signal will be represented by a sequence of samples at intervals sufficiently small to capture the essential detail of the signal. Our normal way of representing such samples is with numbers in the decimal system with a base of 10. However, such a representation is inconvenient for digital systems and it turns out more convenient to use a binary system with base 2, i.e. numbers are represented in terms of zeros and ones. For example, the numbers 0 to 15 will take the form 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111. The rightmost digit will represent the number of 1s, the next the number of 2s, the next the number of 4s, the next the number of 8s and so on.

In the electronic world, the number 1 can take the form of an *on state* and 0 the form of an *off state*. On/off switches are clearly a key element in digital electronics and their simplest realisation is the NMOS transistor. In this device, a suitably positive voltage at the gate turns the device on and current flows, but a zero voltage switches the device off. A PMOS device also acts as a switch, but a negative voltage is required at the gate in order to switch the device on. These devices can be used to form *gates* that can be used to combine and manipulate digital signals. The simplest gate is known as a *NAND gate* and the circuit of such a device is shown in Figure 6.1a. Figure 6.1b shows the output Y of the gate for various combinations of its inputs A and B (we have given the

https://doi.org/10.1017/9781108684514.007 Published online by Cambridge University Press

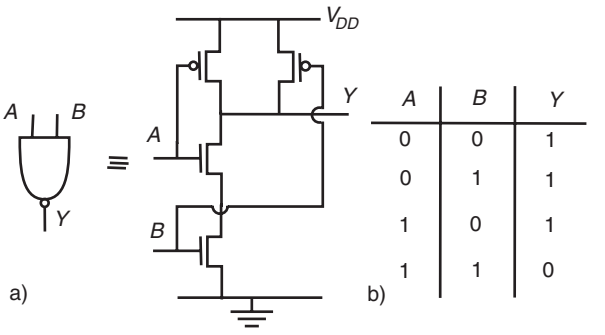


Fig. 6.1 NAND gate and its CMOS realisation.

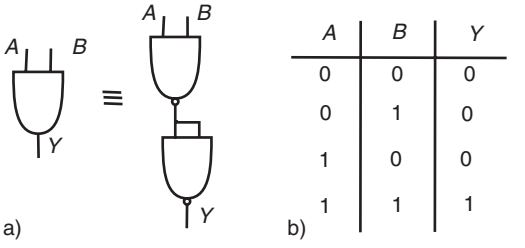


Fig. 6.2 AND gate in terms of NAND gates.

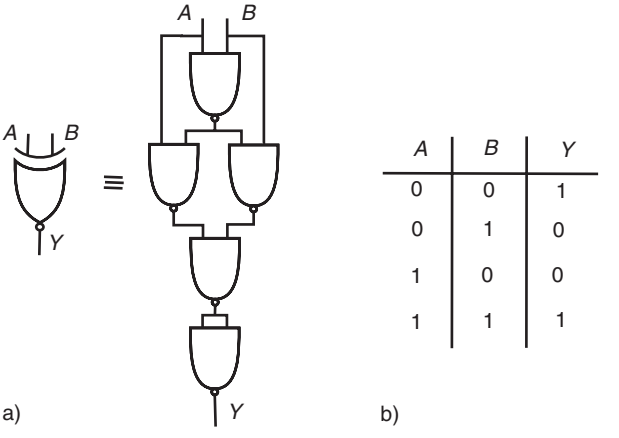
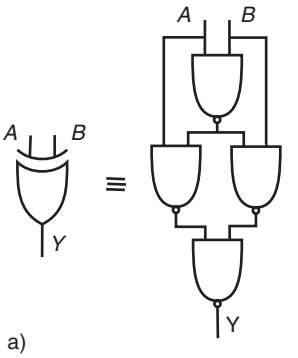


Fig. 6.3 XNOR gate in terms of NAND gates.

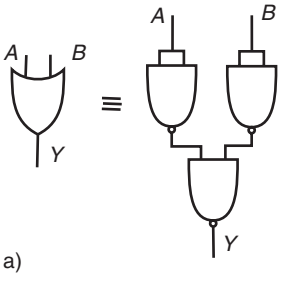
input and output voltages a nominal value of 1). Figures 6.2 to 6.4 show several other gates that can be built from the basic building block of a NAND gate (the additional gates are the *AND*, *XNOR* and *XOR* gates). The gates above are part of what is known as *combinatorial logic* since they are equivalent to a combination of logical operations with the 0 and 1 states regarded as the *false* and *true* states of logic.

https://doi.org/10.1017/9781108684514.007 Published online by Cambridge University Press



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Fig. 6.4 XOR gate in terms of NAND gates.



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Fig. 6.5 OR gate in terms of NAND gates.

If we are to use logic to represent numbers, we will need to represent a single number by several logic states. For example, a four-bit number A will need four logic states $A3$, $A2$, $A1$ and $A0$ (where $A0$ is the number of 1s, $A1$ is the number of 2s, $A2$ is the number of 4s and $A3$ is the number of 8s, i.e. the number $A_0 + 2A_1 + 4A_2 + 8A_3$ in decimal form). If we need to add this to another number B with the bits $B3$, $B2$, $B1$ and $B0$, then we can achieve this with combinatorial logic. We need to add the corresponding bits of both numbers but need to account for any carry from the previous bit and to produce any carry that might be needed for the next bit. This can be achieved with the one-bit adder shown in Figure 6.6a. For a complete addition, we will need to combine several of these one-bit adders as shown in Figure 6.6b. More complex operations, such as multiplication, will require us to perform a sequence of operations and this brings us to *sequential logic*. In sequential logic, the output depends not only on the input at the current time but also on past input and outputs of the circuit (see Figure 6.7). For example, multiplication can be regarded as addition which is repeated several times over.

An example of sequential logic is the JK flip-flop shown in Figure 6.8a. This device is able to maintain the state of its output Q ($\overline{Q}$ being the complementary state of Q) until suitably reset, the reset being enabled on the rising edge of a clock signal (the table shows that changes that can take place). Further, this device can act as a store for a single bit of data (the state of Q) and several of these devices can be used to

https://doi.org/10.1017/9781108684514.007 Published online by Cambridge University Press

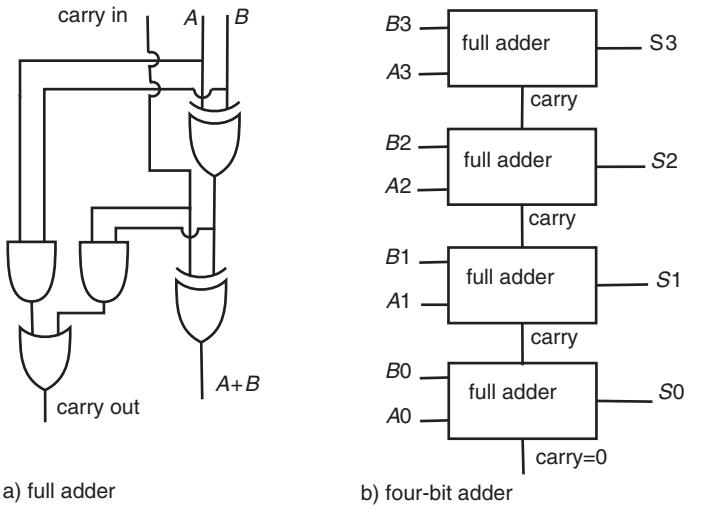


Fig. 6.6 Full adder and four-bit adder.

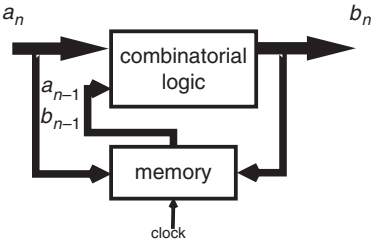
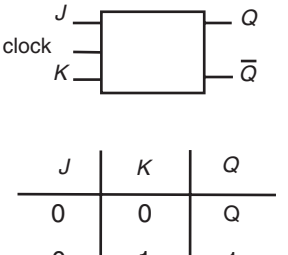
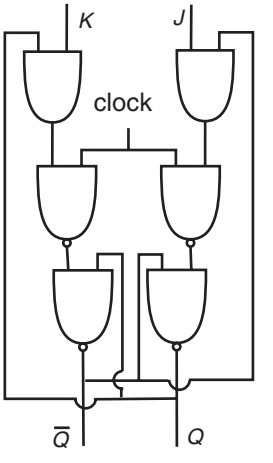


Fig. 6.7 Typical sequential logic.



J	K	Q
0	0	Q
0	1	1
1	0	0
1	1	$\overline{Q}$

a)



b)

Fig. 6.8 A JK flip-flop.

https://doi.org/10.1017/9781108684514.007 Published online by Cambridge University Press

store numbers in the binary system. Indeed such memory is an essential element in sequential logic. Another important element is the clock signal, often supplied by a quartz crystal oscillator, which is used to progress the signal through the logic. If a sequence of numbers is to be processed through several stages of combinatorial logic, the clock pulses will trigger the passage from one stage to the next. If a radio signal has been converted to a sequence of numbers, the various functions of a radio receiver will now become arithmetic operations. For example, amplification becomes multiplication by a fixed number and mixing becomes the multiplication of the sequences representing two different signals. Obviously, a typical radio will require many stages of sequential logic and this can become extremely complex. The logic, however, can be constructed as a combination of a vast number of simple logic elements such as NAND gates. This has been made easier with the advent of *field-programmable gate arrays* (FPGAs) . These are integrated circuits that contain millions of simple gates that can have their connections programmed and hence be made to perform the function of the desired software radio. These devices have revolutionised software radio since the operator can change the nature of the radio by simply reprogramming the FPGA.

6.2 Digital Signal Processing

We have already seen the convenience of analysing analogue RF signals in the complex domain and the same is also true for digital signals. In the complex domain, a real harmonic signal $s(t) = S\cos(2\pi ft)$ can be represented in terms of the complex harmonic signals $\exp(\mathrm{j}2\pi ft)$ and $\exp(-\mathrm{j}2\pi ft)$ as $s(t) = S(\exp(\mathrm{j}2\pi ft) + \exp(-\mathrm{j}2\pi ft))/2$, i.e. the real signal contains equal components on the frequencies f and $-f$. In general, a signal $s(t)$ will be a linear combination of complex harmonic signals at a variety of frequencies. Importantly, for a real signal, the distribution of positive frequency components will be a mirror image of those at negative frequencies (this is illustrated in Figure 6.9). Consequently, it is sufficient to represent a signal $s(t)$ by its positive frequency components $s_+(t)$. Signal $s_+(t)$ can be split into its real and imaginary parts, i.e. $s_+(t) = s^I(t) + \mathrm{j}s^Q(t)$ where s^I and s^Q are known as the *in-phase* and *quadrature* components respectively (note that the original real signal will be $2s^I$). In the case of a simple harmonic signal $s(t) = S\cos(2\pi ft)$, we have an in-phase component

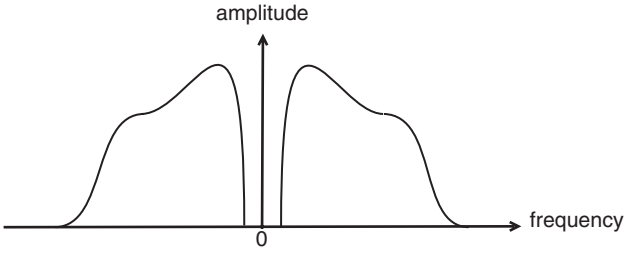


Fig. 6.9 The frequency content of a real signal.

https://doi.org/10.1017/9781108684514.007 Published online by Cambridge University Press

$s^I(t) = S\cos(2\pi ft)/2$ and quadrature component $s^Q(t) = S\sin(2\pi ft)/2$. From this it can be seen that the in phase component is obtained from the quadrature component by a $\pi/2$ shift in phase. Likewise, for a general signal, the in-phase component is obtained from the quadrature component by a $\pi/2$ phase shift for each of its harmonic components.

In the digital domain, an RF signal will be represented by a time-ordered sequence of binary numbers, the numbers having been sampled from the original RF signal. The sampling will need to have been carried out at a rate that gives an adequate representation of the signal and so we will need to determine the limitations on this rate. To this end, we consider a signal $s(t)$ that has been sampled (see Figure 6.10) at a rate F for a finite number N of samples ($T = \frac{1}{F}$ is the time interval between samples). Let $s_0, s_1, s_2, \ldots, s_{N-1}$ be the samples and form the linear combination

$$\frac{1}{N}\sum_{k=0}^{N-1} s_k \sum_{n=0}^{N-1} \exp\left(\mathrm{j}\frac{2\pi n}{N}(l-k)\right) \tag{6.1}$$

where l is an arbitrary integer. Noting the identity

$$\sum_{n=0}^{N-1} \exp(\mathrm{j}\frac{2\pi n}{N}(l-k)) = \frac{1-\exp(\mathrm{j}2\pi(l-k))}{1-\exp\left(\mathrm{j}\frac{2\pi}{N}(l-k)\right)} \tag{6.2}$$

(this is obtained from the geometric sum $\sum_{i=0}^{N-1} a^i = (1-a^N)/(1-a)$ with $a = \exp(\mathrm{j}2\pi n(l-k)/N)$), we will find that this sum takes the value N when $l = k$ and is zero for other integral combinations of l and k. As a consequence, (6.1) will evaluate to s_l and then

$$s_l = \sum_{n=0}^{N-1}\left(\frac{1}{N}\sum_{k=0}^{N-1} s_k \exp(-\mathrm{j}\frac{2\pi kn}{N})\right)\exp\left(\mathrm{j}\frac{2\pi ln}{N}\right) \tag{6.3}$$

on changing the order of summation. We can now make the following approximation to the signal $s(t)$

$$s(t) \approx \frac{1}{N}\sum_{n=0}^{N-1} a_n \exp\left(\mathrm{j}2\pi n\Delta F\, t\right), \tag{6.4}$$

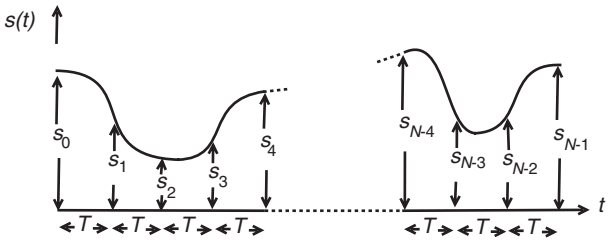


Fig. 6.10 The sampling of an analogue signal.

https://doi.org/10.1017/9781108684514.007 Published online by Cambridge University Press

where $\Delta F = \frac{F}{N}$ and the coefficients $a_0, a_1, \ldots, a_{N-1}$ are given by

$$a_n = \sum_{k=0}^{N-1} s_k \exp\left(-\mathrm{j}\frac{2\pi kn}{N}\right). \tag{6.5}$$

This gives an exact representation of the signal at the sample times $0, T, 2T, 3T, \ldots, (N-1)T$ and interpolated values at other times. The coefficients $a_0, a_1, a_2, a_3, \ldots, a_{N-1}$ are the magnitude of the signal at frequencies 0, ΔF, $2\Delta F$, $3\Delta F$, …, $(N-1)\Delta F$ respectively and together are known as the *discrete Fourier transform* (DFT). We now need to consider the limitations of the discrete representation of a signal and its spectral content. To this end, consider the complex harmonic signal $s(t) = \exp(\mathrm{j}2\pi ft)$ and form its DFT

$$\begin{aligned} a_n &= \sum_{k=0}^{N-1} \exp\left(-\mathrm{j}2\pi k\frac{n}{N}\right)\exp\left(\mathrm{j}2\pi k\frac{f}{F}\right) \\ &= \frac{1-\exp\left(-\mathrm{j}2\pi N\left(\frac{n}{N}-\frac{f}{F}\right)\right)}{1-\exp\left(-\mathrm{j}2\pi\left(\frac{n}{N}-\frac{f}{F}\right)\right)} \\ &= \exp\left(-\mathrm{j}\pi(N-1)\left(\frac{n}{N}-\frac{f}{F}\right)\right)\frac{\sin\left(\mathrm{j}\pi N\left(\frac{n}{N}-\frac{f}{F}\right)\right)}{\sin\left(\mathrm{j}\pi\left(\frac{n}{N}-\frac{f}{F}\right)\right)}. \end{aligned} \tag{6.6}$$

For frequencies that are a multiple of F/N (i.e. $f = m\Delta F$ where m is an integer), we have that $a_m = 1$ and that $a_n = 0$ when $n \neq m$. This means that the DFT has picked out the correct frequency content of the signal. However, when the frequency f is a non-integral multiple of F/N, the DFT will spread the energy around all the spectral coefficients. This is known as *spectral leakage* and is a particular problem will the DFT. Fortunately, the leakage will only be significant for coefficients associated with frequencies close to f. Clearly, ΔF sets the lower limit to the frequency resolution of the DFT.

We have seen that the sample rate and data length set a lower limit on frequency resolution, but the question arises as to what is the maximum frequency that can be resolved. It will be noted that the harmonic signal $s(t) = \exp(\mathrm{j}2\pi((f+mF)t)$ will have the same DFT as $s(t) = \exp(\mathrm{j}2\pi ft)$ when m is an integer. As a consequence the DFT will not distinguish between them, and their reconstructions (the inverse DFT) will be identical, i.e. both signals will be in the frequency range $[0, F]$. This phenomenon is known as *aliasing* and means that a DFT has the potential to *fold* signal content with frequencies outside the interval $[0, F]$ onto signals with frequency content within the interval. This is illustrated in Figure 6.11, which shows three signals with distinctly different frequency content but with the same DFT. Figure 6.12 shows an example of a signal with frequency content outside the range [0,F]. The signal can be pictured as two separate signals, one with frequency content within the interval [0,F] and one with all its frequency content outside this range. The DFT, however, will fold the signal with frequencies outside [0,F] onto one with content inside this interval and, when the signals

https://doi.org/10.1017/9781108684514.007 Published online by Cambridge University Press

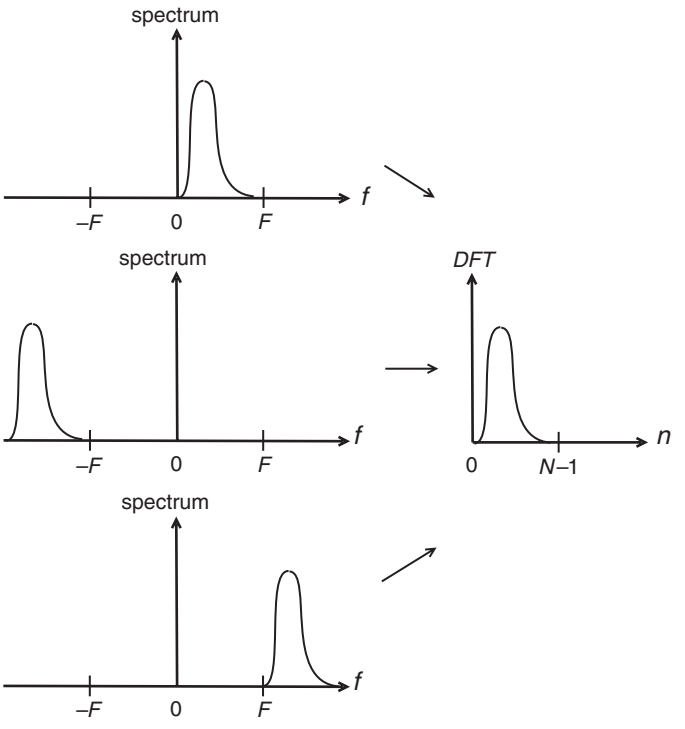


Fig. 6.11 Three signals with different spectral content, but identical DFT due to aliasing.

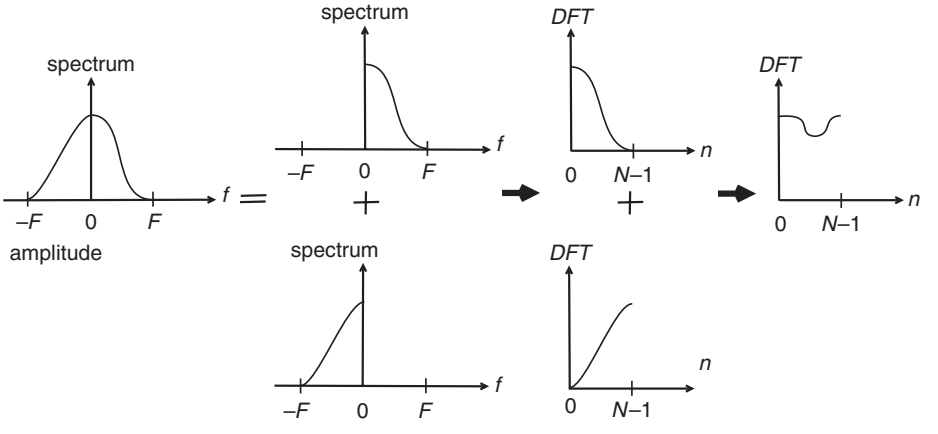


Fig. 6.12 Spectral contamination caused by aliasing.

are recombined, they will give a completely erroneous picture of frequency content. In order for the DFT to properly resolve the frequency content of the signal, the signal will need to be band-limited with the bandwidth dictated by the sampling rate. For a real signal, there will be both positive and negative frequencies and so the bandwidth needs

https://doi.org/10.1017/9781108684514.007 Published online by Cambridge University Press

to prevent the negative frequencies folding onto the positive frequencies and this leads to the *Nyquist sampling theorem* (Nyquist, 1928), i.e.

The sample rate F will need to be at least twice the maximum frequency present if aliasing is to be avoided.

As a consequence of this, it is normal to use an *anti-aliasing filter* before sampling in order to remove any signal components at frequencies above $F/2$.

As we have seen earlier, filtering is an important function in analogue RF signal processing and this obviously remains true in digital processing. The DFT suggests a possible approach to such filtering. Consider N consecutive samples be taken from a digitised signal and consider their DFT. If we want to pass the signal through a band-pass filter, lower and upper band edges $P\Delta F$ and $Q\Delta F$ respectively, we could do this by reconstructing the consecutive samples from the DFT coefficients a_P to a_Q alone, i.e.

$$s_l^F = \frac{1}{N}\sum_{n=P}^{Q} a_n \exp\left(\mathrm{j}2\pi\frac{ln}{N}\right), \tag{6.7}$$

where s^F denotes the filtered sequence. The coefficients a_n can be eliminated from (6.7) by means of (6.5) to yield

$$s_l^F = \sum_{k=0}^{N-1} s_k\, c_{k-l}, \tag{6.8}$$

where

$$c_m = \frac{1}{N}\sum_{n=P}^{Q} \exp\left(-\mathrm{j}2\pi\frac{mn}{N}\right), \tag{6.9}$$

i.e. the band-pass samples can be formed as linear combinations of the original samples. This suggests that, in the case of an infinite sequence, we form a filtered sequence by applying a process of the form of (6.8). (This is exemplified by a moving average that is used to smooth data, essentially a low-pass filtering process.) The filtered sequence (s_0^F, s_1^F, s_2^F, s_3^F, …) is generated according to

$$s_M^F = \sum_{k=-L}^{N-1-L} s_{k+M}\, c_k \tag{6.10}$$

for suitable L (note that the output sequence is delayed by $N-1-L$ samples). The filter is implemented as the sequential circuit shown in Figure 6.13 which consists of a combination of adder, multiplier and delay elements. The signal samples enter on the left and are moved through at a rate of one stage per sample interval; the filtered sequence then exits to the right. The output of a delay block (labelled Z^{-1}) consists of its input value at the previous step of the process and therefore will involve some sort of memory. It is important to note that the above filter is based upon ideas that come from the filtering of a finite-length sequence of data, i.e. each output value is derived from a truncated sequence of input data. Unfortunately, this *windowing* process has the effect of

https://doi.org/10.1017/9781108684514.007 Published online by Cambridge University Press

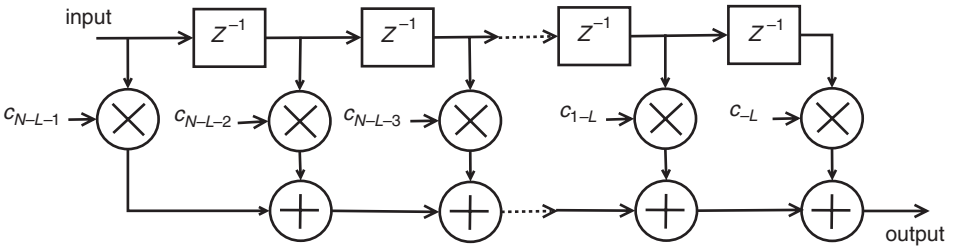


Fig. 6.13 A finite-length discrete signal filtering system.

spreading the spectrum of the output sequences outside the desired filter bounds (another example of spectral leakage). Fortunately, however, we can reduce this effect by making the window more benign. This can be achieved by making the filter coefficients (i.e. the c_k) taper towards zero as k approaches both 0 and N (to minimise the effects of the taper, it is advisable to choose L close to $N/2$).

Digital signal processing can be computationally expensive and this can make it hard to maintain the kind of real time throughput that is required by modern radio systems. One way of improving matters is to eliminate any unnecessary data processing. Fortunately, the Nyquist sampling theorem comes to the rescue as it tells us that we need only use a sampling rate that is twice the bandwidth. Consequently, providing we obey this limitation, we can reduce the computational requirement by only keeping every M^{th} sample, a process that is known as *down sampling*. To avoid any possibility of aliasing, such a process is often proceeded by a low-pass filter with bandwidth consistent with the new sampling rate (the combination is known as a *decimation filter*). If we need to change a signal back to a greater sample rate, we can repeat each sample M times, but we must then follow this process by a low-pass filter in order to remove any high-frequency component that is introduced by this interpolation process (this total process is known as an *interpolation filter*).

The computational expense of such processing has led to some clever ideas for filtering and one of the most important of these is the cascaded integrator comb (CIC) filter (Hogenauer, 1981). Figures 6.14a and 6.14b show an *integrator filter* and a *comb filter* respectively. The amplitude response of the integrator in terms of frequency is $1/2|\sin(\pi f/F)|$ and so it acts like a low-pass filter (F is the sampling rate). On the other hand, the comb filter has the amplitude response $2|\sin(\pi Mf/F)|$ and acts as a multiple notch filter that removes frequencies that are integral multiples of frequency F/M (i.e. it behaves like a comb). If we combine these filters in the manner of Figure 6.15, we obtain a filter with the frequency response that is also shown in this Figure 6.15 (the amplitude response is $|\sin(2\pi fRM/2F)/\sin(2\pi f/2F)|^N$). It will be noted that the filter does not contain any computationally expensive processes (only delays, additions and changing the sign of a number) and so can form the basis of an extremely efficient filter. One downside, however, is that the number of bits in the samples grows during passage through the filter by an amount of $N\log_2(RM)$. However, this growth can usually be handled by judicious truncation at various stages of the filter.

https://doi.org/10.1017/9781108684514.007 Published online by Cambridge University Press

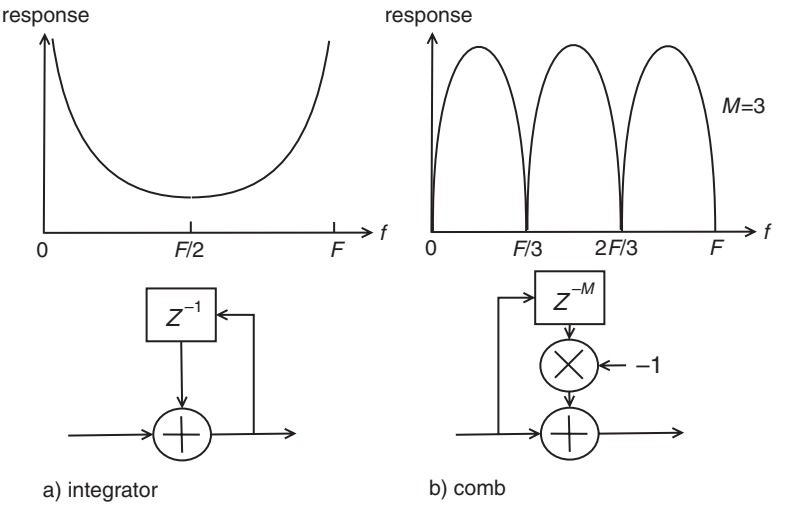


Fig. 6.14 Integrator and comb filters with their frequency response.

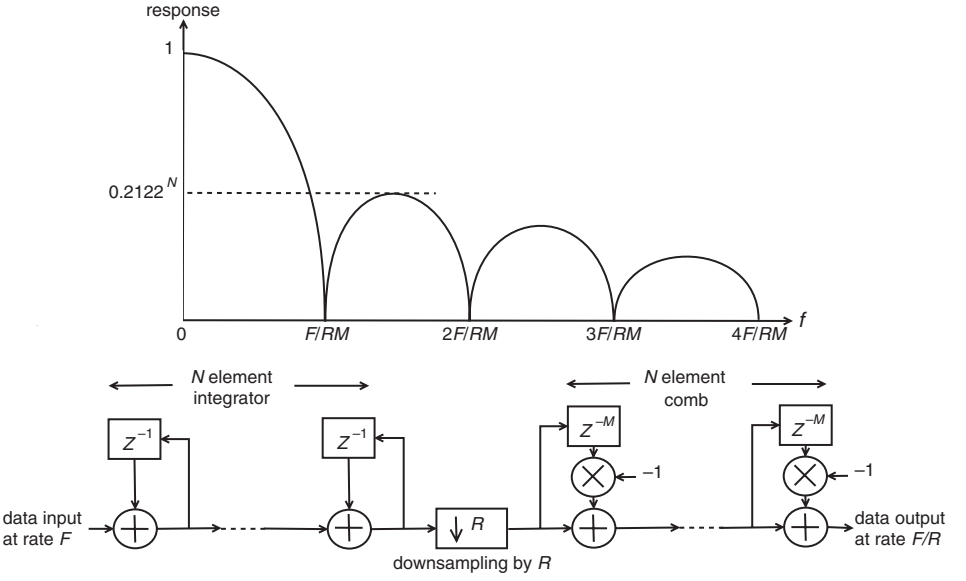


Fig. 6.15 CIC filter with its frequency response.

6.3 Analogue-to-Digital and Digital-to-Analogue Converters

Key to the realisation of software radio is the ability to convert an RF signal into a sequence of numbers through an appropriate *analogue-to-digital converter* (ADC). The most basic ADC is the threshold detector, an example of which is shown in example of Figure 6.16. This converter can be used to detect whether a pulse is present and can

https://doi.org/10.1017/9781108684514.007 Published online by Cambridge University Press

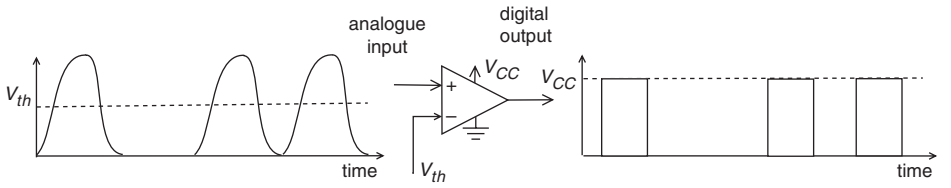


Fig. 6.16 A simple one-bit analogue-to-digital converter.

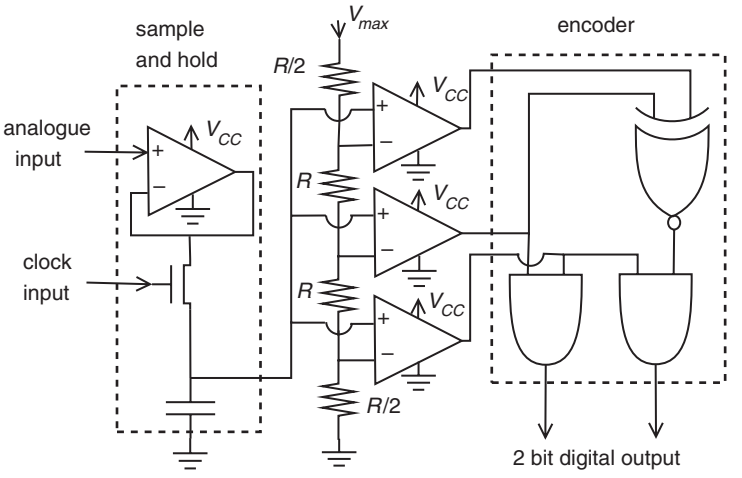


Fig. 6.17 A two-bit analogue-to-digital flash converter.

therefore be used to detect certain kinds of digital modulation. If the input voltage is above a certain threshold, its level is raised to that of the supply. However, if the input voltage is below the threshold, the output voltage will be that of the ground. Such a converter can only detect two levels of amplitude (i.e. it can provide one bit of data). For signals more complex than simple pulses, we will require a much greater amplitude resolution. Figure 6.17 shows a simple *flash converter* that can detect four levels of amplitude (i.e. it can provide two bits of data). At the input there is a *sample-and-hold* system to ensure that the samples have a constant amplitude during the acquisition period. Digitisation is achieved through a multi-threshold comparator and the logic circuits encode the output as a two-bit binary number. A two-bit converter is too crude (see Figure 6.18) for most software radio, but this simple converter demonstrates the general principles.

The conversion of an analogue signal, into one with only a finite number of fixed levels of voltage, will obviously lead to some inaccuracy and this is known as *quantisation noise* (see Figure 6.18). Let the sample levels have spacing Δ then, for a sampling interval length T, the signal will fluctuate up to $\pm\Delta/2$ about the mean. Assuming that the quantisation voltage error v_{qn} varies linearly between the quantisation levels, its mean square will be given by

$$\overline{v_{\mathrm{qn}}^2} = \frac{\Delta^2}{12} = \frac{V_{\mathrm{max}}^2 2^{-2b}}{12}, \tag{6.11}$$

https://doi.org/10.1017/9781108684514.007 Published online by Cambridge University Press

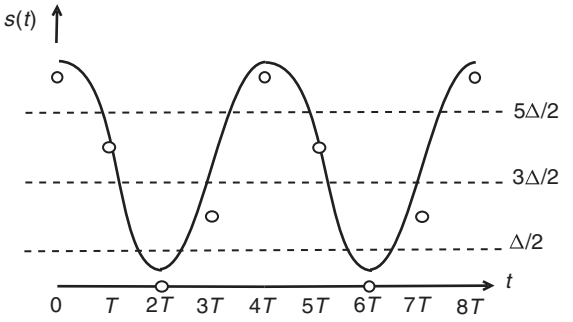


Fig. 6.18 Two-bit digitisation.

where b is the bit number resolution of the ADC and $V_{\max}$ is the voltage range. The power of the *quantisation noise* will therefore be

$$N_{qn} = \frac{V_{\max}^2 2^{-2b}}{12R_{\text{in}}}, \tag{6.12}$$

where R_{in} is the input resistance of the ADC. $V_{\max}$ will be the largest peak-to-peak voltage that ADC can accommodate and so the largest possible SNR will be

$$\begin{aligned} SNR &= \frac{(V_{\max}/2)^2}{2R_{\text{in}}}/N_{\text{qn}} = \frac{3}{2}2^{2b} \\ &= (6.02b + 1.75)\ \text{dB}. \end{aligned} \tag{6.13}$$

This is the *dynamic range* of the converter. If the quantisation noise is spread uniformly across the *Nyquist bandwidth* ($F/2$), the SNR will increase by factor $F/2B$ when the ADC output is processed through a digital filter with bandwidth B (quantity $10\log_{10}(F/2B)$ is known as the processing gain). However, if the input consists of relatively few signals, the quantisation noise can become highly correlated and cause a considerable number of high-amplitude *spurs* (see Figure 6.19a). Fortunately, this can be alleviated by adding some noise at the input to the ADC (a process known as *dithering*) in order to assist the decorrelation of the quantisation noise and hence reduce the spurs (see Figure 6.19b). Such noise can be added into a section of the Nyquist bandwidth that is to be removed by subsequent processing, e.g. where there are no signals of interest.

As with sequential digital electronics in general, the clock signal is an important element in analogue-to-digital conversion and its stability is an important issue. A quartz crystal oscillation is usually used for stability, but imperfections can still arise in the signal from a practical oscillator. The resulting fluctuations in sample interval are known as *aperture jitter* and cause distortions that translate into additional noise that increases with frequency f. For high-resolution converters, this noise will dominate and the largest possible SNR will take the form

$$SNR = 20\log_{10}\left(\frac{1}{2\pi f \Delta\tau}\right)\text{dB}, \tag{6.14}$$

where $\Delta\tau$ is the rms value of the sample interval fluctuations.

https://doi.org/10.1017/9781108684514.007 Published online by Cambridge University Press

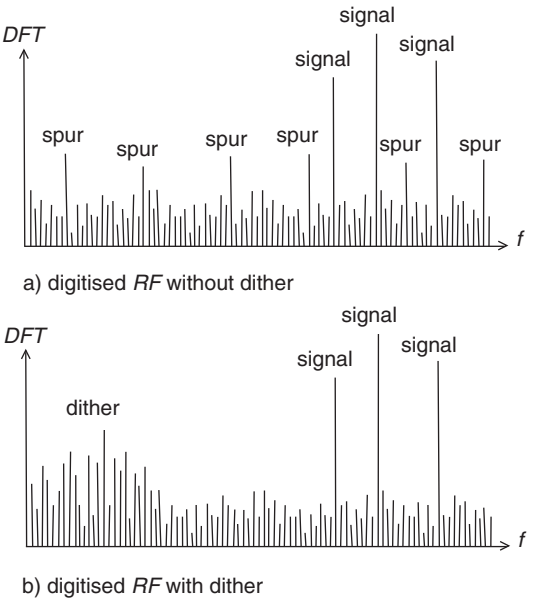


Fig. 6.19 Sampled signals with quantisation noise and spurs.

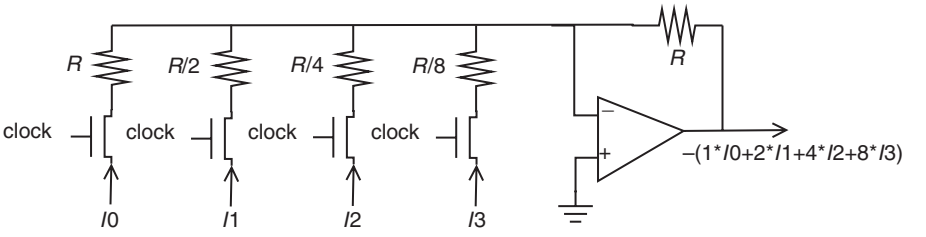


Fig. 6.20 Simple digital-to-analogue converter.

Once converted into digital form, the analogue signal can be processed by a computer or perhaps some programmable logic. Often, however, we will need to convert the processed signal back into analogue form (this is certainly the case for audio broadcast signals). To achieve this, we need a digital-to-analogue converter (DAC), a simple example being shown in Figure 6.20. This circuit adds suitable multiples of the input digits and produces an output that is proportional to the number represented by the digits.

We now consider the option of increasing the sample rate above that suggested by the Nyquist theorem, i.e. *over-sampling*. This might seem wasteful, but we will see that it does have some merit. We have already seen that the effective SNR is controlled by the bit number resolution of the ADC and Figure 6.21 illustrates a signal and accompanying quantisation noise. If we increase the sample rate by a factor k (i.e. sample at frequency kF) we obtain the same SNR, but now the noise is spread out over a wider frequency

https://doi.org/10.1017/9781108684514.007 Published online by Cambridge University Press

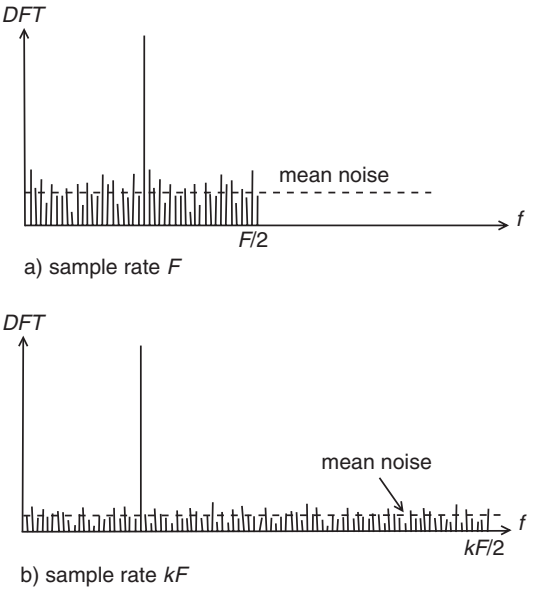


Fig. 6.21 The effect of over-sampling.

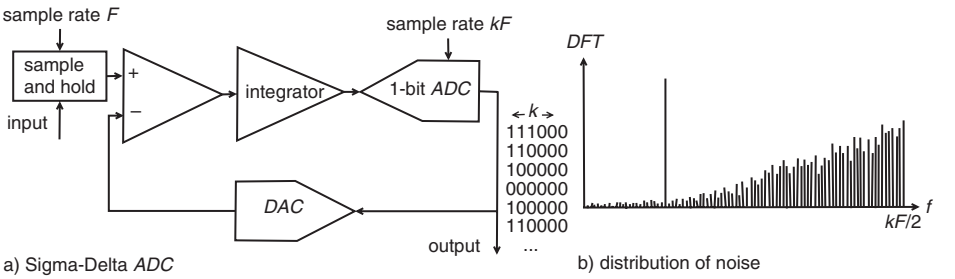


Fig. 6.22 Sigma-delta ADC.

range. Consequently, if we filter out the extra frequency range, we will obtain an increased SNR. This fact is used in what is known as a *sigma-delta* ($\Sigma\Delta$) ADC.

Figure 6.22a shows the circuit of a basic $\Sigma\Delta$ ADC. The signal is fed into a sample-and-hold circuit that operates at the desired sampling rate and then enters a comparator where the output of the converter (initially zero) is passed through a DAC and subtracted from the sampled input. This is then fed into a one-bit ADC with suitable threshold and the process repeated. The output will hence consist of a sequence of constant values (notionally 1s) until their combination has reached the level of the input signal and will thereafter be zero until k samples of the ADC have been achieved. The sequence of 1s and 0s will then constitute the digitised signal where the number of 1s is the value of the sample. The integrator before the ADC has the effect of moving the quantisation noise to the higher frequencies (see Figure 6.22b). Consequently, when the output is

https://doi.org/10.1017/9781108684514.007 Published online by Cambridge University Press

converted to a more conventional form by counting the number of 1s in the sequence of k digits, this noise will be averaged out. With the $\Sigma\Delta$ approach to ADCs, it is possible to make very simple high-resolution linear converters. However, their bandwidth is limited by the need to over-sample and so there is a trade-off between bandwidth and resolution.

6.4 Digital Receiver and Transmitter Architecture

In software radio, the various functions within an analogue radio become arithmetic operations performed on a sequence of samples representing the RF signal. Amplification becomes a multiplication by the voltage gain, mixing becomes the multiplication of two signal sequences and filtering becomes a weighted moving average. Essentially, the radio architecture remains the same, it is just the realisation that changes. The digital circuits can be realised as programs within a computer or, more often as not, constructed out of the logic within an FPGA.

Most software radio receivers are basically of the direct-conversion variety, i.e. they convert directly to baseband. Figure 6.23 is an example of such a receiver. It is essentially an analogue receiver with the analogue-to-digital conversion performed at the outputs of the mixers. We first analyse a simple cosine RF signal $s^{\mathrm{RF}} = S^{\mathrm{RF}}\cos(2\pi f_{\mathrm{RF}}t)$ and local oscillator signal $s^{\mathrm{LO}} = S^{\mathrm{LO}}\cos(2\pi f_{\mathrm{LO}}t)$. The in-phase signal output will be

$$\begin{aligned} s^{\mathrm{I}}(t) &= S^{\mathrm{RF}}S^{\mathrm{LO}}\cos(2\pi f_{\mathrm{LO}}t)\cos(2\pi f_{\mathrm{RF}}t) \\ &= \frac{S^{\mathrm{RF}}S^{\mathrm{LO}}}{2}(\cos(2\pi(f_{\mathrm{LO}}-f_{\mathrm{RF}})t)+\cos(2\pi(f_{\mathrm{LO}}+f_{\mathrm{RF}})t)) \end{aligned} \tag{6.15}$$

and the quadrature output

$$\begin{aligned} s^{\mathrm{Q}}(t) &= -S^{\mathrm{RF}}S^{\mathrm{LO}}\sin(2\pi f_{\mathrm{LO}}t)\cos(2\pi f_{\mathrm{RF}}t) \\ &= -\frac{S^{\mathrm{RF}}S^{\mathrm{LO}}}{2}(\sin(2\pi(f_{\mathrm{LO}}-f_{\mathrm{RF}})t)+\sin(2\pi(f_{\mathrm{LO}}+f_{\mathrm{RF}})t)). \end{aligned} \tag{6.16}$$

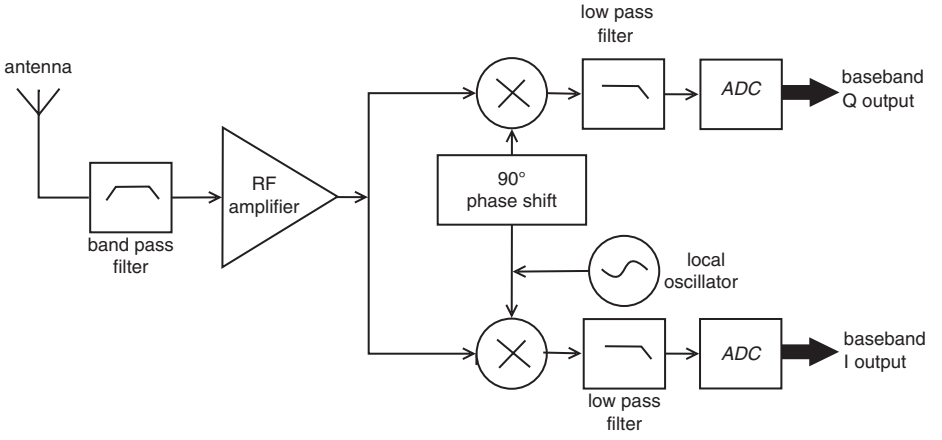


Fig. 6.23 A direct conversion receiver with baseband analogue-to-digital conversion.

https://doi.org/10.1017/9781108684514.007 Published online by Cambridge University Press

The low-pass filters at the mixer outputs are used to remove the high-frequency components from the outputs and so the receiver will produce digitised copies of the input RF signal that have been translated downward in frequency by f_{LO}. However, one copy will be in phase with the original signal and the other will be $\pi/2$ out of phase (i.e. in quadrature). In general, the RF signal will be modulated and so contain components over a range of frequencies, all of which will be translated downward by f_{LO} with the quadrature components $\pi/2$ out of phase.

We now need to consider the demodulation of the I and Q outputs. Let s_n^{I} be an element of the sequence that represents the in-phase output and s_n^{Q} be an element of the sequence that represents the quadrature output. For an amplitude modulated signal, the sequence that represents the original modulation will have the terms $S_n^{\mathrm{demod}} = \left(|s_n^{\mathrm{I}}|^2 + |s_n^{\mathrm{Q}}|^2\right)^{\frac{1}{2}}$ and, for a phase modulated signal, the sequence will have terms $S_n^{\mathrm{demod}} = \arctan(s_n^{\mathrm{Q}}/s_n^{\mathrm{I}})$. Frequency is the rate of change of phase and so frequency demodulation will consist of differentiating the phase demodulated output. In the case of discrete samples, however, this will consist of forming the difference sequence

$$S_n^{\mathrm{demod}} = \frac{1}{T}\left(\arctan(s_n^{\mathrm{Q}}/s_n^{\mathrm{I}}) - \arctan(s_{n-1}^{\mathrm{Q}}/s_{n-1}^{\mathrm{I}})\right). \tag{6.17}$$

In the case of single sideband modulated signals, the architecture of Figure 6.23 is ideal. We now further process the quadrature output by a digital filter that shifts the phase by $-\pi/2$ at positive frequencies and by $\pi/2$ at negative frequencies (this is known as a *Hilbert transform* filter). Then, for a simple cosine RF signal, the output will be the digitised form of

$$\begin{aligned} s^{\mathrm{H}}(t) &= -\frac{S^{\mathrm{RF}}S^{\mathrm{LO}}}{2}\cos\left(2\pi(f_{\mathrm{LO}} - f_{\mathrm{RF}})t\right) \quad f_{\mathrm{RF}} > f_{\mathrm{LO}} \\ &= \frac{S^{\mathrm{RF}}S^{\mathrm{LO}}}{2}\cos\left(2\pi(f_{\mathrm{LO}} - f_{RF)}t\right) \quad f_{\mathrm{RF}} < f_{\mathrm{LO}}. \end{aligned} \tag{6.18}$$

(The Hilbert transform has essentially multiplied positive frequency content by $-\mathrm{j}$ and negative frequency content by j.) We further note that the in-phase output will be the digitised form of

$$s^{\mathrm{I}}(t) = \frac{S^{\mathrm{RF}}S^{\mathrm{LO}}}{2}\cos\left(2\pi(f_{\mathrm{LO}} - f_{\mathrm{RF}})t\right). \tag{6.19}$$

In order to demodulate SSB signals, we will need to choose f_{LO} to be the carrier frequency; then the addition of $s^{\mathrm{I}}(t)$ and $s^{\mathrm{H}}(t)$ will eliminate signals above f_{LO} and subtraction will eliminate signals below. Consequently, we form output sequence $S_n^{\mathrm{demod}} = s_n^{\mathrm{I}} + s_n^{\mathrm{H}}$ for LSB and $S_n^{\mathrm{demod}} = s_n^{\mathrm{I}} - s_n^{\mathrm{H}}$ for USB (note that the terms s_n^{H} represent the quadrature samples s_n^{Q} after processing through a discrete version of the Hilbert transform filter).

The goal of software radio is to move the ADC closer and closer to the antenna until the function of the analogue electronics is to merely filter and amplify the incoming signal (these analogue functions are still necessary). In a fully digital receiver, the mixer function is performed by multiplying each term s_n^{RF} of the digitised RF signal by the corresponding term of a digital local oscillator sequence s_n^{LO}. A typical direct digitising

https://doi.org/10.1017/9781108684514.007 Published online by Cambridge University Press

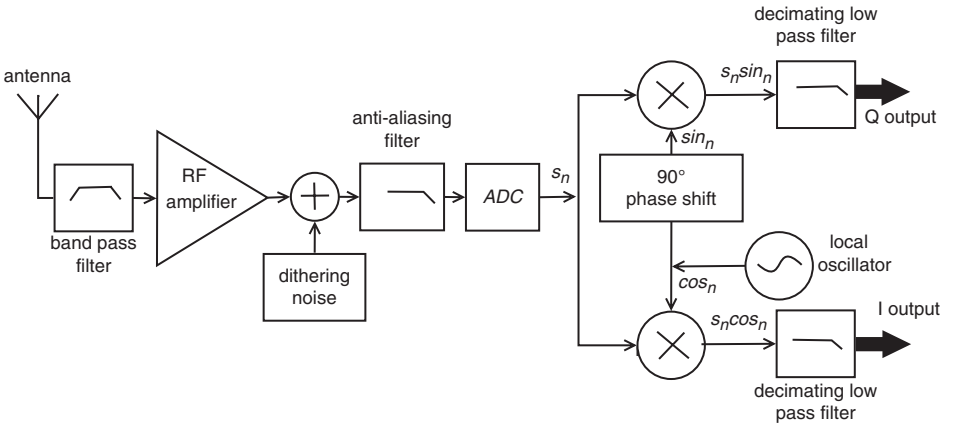


Fig. 6.24 A direct digitising receiver.

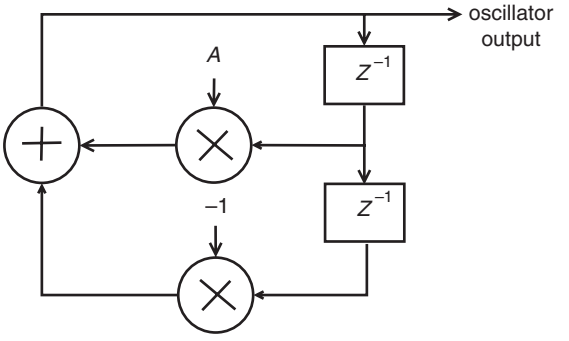


Fig. 6.25 A digital oscillator ($A = 2\cos(2\pi f_{\mathrm{LO}}/F)$).

receiver is shown in Figure 6.24. It will be noted the mixers on the I and Q channels are followed by *decimating filters* since the sampling rate requirement at this stage will often be much lower than that of the original RF signal (according to the Nyquist theorem it will be twice the baseband bandwidth). This helps reduce the computational demand of subsequent processing. Obviously, in order to achieve the mixing in the digital domain, we will also need to generate discrete samples corresponding to the local oscillator signal. We can generate a cosine LO sequence by starting with $s_0^{\mathrm{LO}} = 1$ and $s_1^{\mathrm{LO}} = A\cos(2\pi f_{\mathrm{LO}}/F)$ and then generating subsequent values through the recurrence relation $s_n^{\mathrm{LO}} = 2\cos(2\pi f_{\mathrm{LO}}/F)s_{n-1}^{\mathrm{LO}} - s_{n-2}^{\mathrm{LO}}$ where f_{LO} is the required oscillator frequency and F is the sample rate. The oscillator can be realised through the circuit of Figure 6.25 where the memory of the delays will be initially seeded with s_0^{LO} and s_1^{LO} (s sine sequence can be generated by as suitably delayed cosine sequence).

The major limitation of the above procedures is the availability of ADCs with suitable dynamic range and high enough sampling rate. Analogue down conversion before digitisation is one possibility, but another is to use what are commonly known as

https://doi.org/10.1017/9781108684514.007 Published online by Cambridge University Press

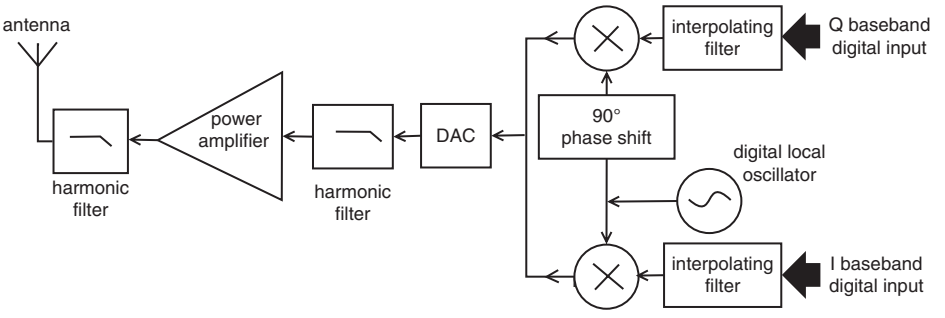


Fig. 6.26 A direct digitising transmitter.

under-sampling techniques. As we have noted previously, sampling a signal at frequency F will cause components at frequencies above $F/2$ to fold onto lower frequencies. In the case that these lower frequencies are unoccupied, we will have down conversion without the use of mixers. However, such an approach will require the use of a band-pass anti-aliasing filter (see Figure 6.24) with bandwidth less than the Nyquist bandwidth of the ADC. In addition, the sample and hold circuits in the ADC must have a response time that is fast enough to handle the input signal.

In the case of transmission, the architectures are once again based on those in the analogue domain. A typical architecture for a direct digitising transmitter is shown in Figure 6.26. We generate the desired modulation by forming the appropriate baseband I and Q sequences. In the case of SSB, we form these sequences from digitised audio with the quadrature input processed through a Hilbert transform filter. We have a digitised form of the phasing method. The audio will usually be sampled at a low rate due to its low frequency and so we use an *interpolating filter* to convert the samples into an equivalent sequence with the same the rate as those of the RF signals (this is achieved by interpolating between the low-rate samples). The output of the digital SSB generator is fed into a DAC after which it is filtered in order to remove any unwanted harmonics. Consequently, the output of the filter is the desired analogue RF signal which can then be amplified to the appropriate power level.

6.5 Conclusion

In the current chapter we have shown that many of the traditional functions of a radio can be performed within the digital domain. Indeed, the digitisation process can now almost reach the radio antenna. However, at the input of the receiver, and at the output of a transmitter, analogue filtering and amplification will always be necessary. Whilst we can now generate, and detect, radio signals with great fidelity, this is not much use if our ability to launch, propagate and capture radio waves is poor. As a consequence, an understanding of transmission lines, antennas and propagation is crucial to the design and implementation of an effective radio system. These topics are the subject of the next three chapters.

https://doi.org/10.1017/9781108684514.007 Published online by Cambridge University Press