PSpice for Digital Communications Engineering
PSpice for Digital Communications Engineering

Paul Tobin
School of Electronic and Communications Engineering
Dublin Institute of Technology
Ireland

SYNTHESIS LECTURES ON DIGITAL CIRCUITS AND SYSTEMS #10

MORGAN & CLAYPOOL PUBLISHERS
ABSTRACT
PSpice for Digital Communications Engineering shows how to simulate digital communication systems and modulation methods using the very powerful Cadence Orcad PSpice version 10.5 suite of software programs. Fourier series and Fourier transform are applied to signals to set the ground work for the modulation techniques introduced in later chapters. Various baseband signals, including duo-binary baseband signaling, are generated and the spectra are examined to detail the unsuitability of these signals for accessing the public switched network. Pulse code modulation and time-division multiplexing circuits are examined and simulated where sampling and quantization noise topics are discussed. We construct a single-channel PCM system from transmission to receiver i.e. end-to-end, and import real speech signals to examine the problems associated with aliasing, sample and hold.

Companding is addressed here and we look at the A and mu law characteristics for achieving better signal to quantization noise ratios. Several types of delta modulators are examined and also the concept of time division multiplexing is considered. Multi-level signaling techniques such as QPSK and QAM are analyzed and simulated and ‘home-made meters’, such as scatter and eye meters, are used to assess the performance of these modulation systems in the presence of noise. The raised-cosine family of filters for shaping data before transmission is examined in depth where bandwidth efficiency and channel capacity is discussed. We plot several graphs in Probe to compare the efficiency of these systems. Direct spread spectrum is the last topic to be examined and simulated to show the advantages of spreading the signal over a wide bandwidth and giving good signal security at the same time.

KEYWORDS
Fourier series and Fourier transforms, baseband and passband modulation, pulse code modulation, time-division multiplexing, quantization noise, M-ary signaling, QPSK, QAM, eye-meter, scatter diagrams, spread spectrum, raised cosine filter.
I would like to dedicate this book to my wife and friend, Marie and sons Lee, Roy, Scott and Keith and my parents (Eddie and Roseanne), sisters, Sylvia, Madeleine, Jean, and brother, Ted.
Contents

Preface .................................................................................................................. xi

1. Fourier Analysis, Signals, and Bandwidth ......................................................... 1
   1.1 Digital Signals ............................................................................................. 1
   1.2 Bandwidth .................................................................................................. 2
   1.3 Pulse Spectra for Different Pulse Widths and Period .................................. 2
       1.3.1 Average and RMS Pulse Power ....................................................... 3
       1.3.2 Unsynchronizing Probe–Plot Axis .................................................... 4
       1.3.3 Fourier Transform ........................................................................... 6
       1.3.4 Fourier Series Analysis .................................................................... 8
   1.4 The Vector Part .......................................................................................... 11
   1.5 Exercise ..................................................................................................... 13

2. Baseband Transmission Techniques .................................................................... 15
   2.1 Baseband Signals ....................................................................................... 15
   2.2 Baseband Signal Formats ........................................................................... 15
       2.2.1 NonReturn to Zero (NRZ) Coding ..................................................... 16
       2.2.2 FileStim Generator .......................................................................... 16
       2.2.3 NRZ-B .............................................................................................. 17
   2.3 RZ Encoding and Decoding ........................................................................ 18
       2.3.1 RZ to NRZ Decoder ........................................................................... 19
   2.4 Manchester Encoding and Decoding ......................................................... 21
       2.4.1 Manchester Unipolar to Bipolar Encoding ........................................ 21
       2.4.2 Manchester (Biphase) Decoding ...................................................... 23
       2.4.3 Differential Manchester Coding ...................................................... 24
   2.5 Alternate Mark Inversion Encoding .......................................................... 25
       2.5.1 AMI Decoding .................................................................................. 25
   2.6 DUO–Binary Baseband Signaling .............................................................. 27
       2.6.1 Use of a Precoder in Duo–Binary Signaling ...................................... 29
   2.7 Integrate and Dump Matched Filter Baseband Receiver ............................. 30
       2.7.1 Example .......................................................................................... 32
   2.8 Exercises .................................................................................................... 34
3. Sampling and Pulse Code Modulation
   3.1 Single-Channel Pulse Code Modulation
   3.2 Companding Characteristics
   3.3 Sampling
      3.3.1 Sallen and Key Antialiasing Active Filter
      3.3.2 Speech Signals
      3.3.3 Sample and Hold
   3.4 Quantization Noise
   3.5 Analog to Digital Conversion
      3.5.1 DAC Resolution
      3.5.2 Band-Stop Filter
   3.6 Pulse Code Modulation
      3.6.1 Universal Shift Register
      3.6.2 74194 Universal Shift Register
   3.7 Single-Channel 4-Bit PCM Transmitter
   3.8 Time-Division Multiplexing and Demultiplexing
      3.8.1 Time-division Multiplexing of Two PAM Signals
   3.9 Linear Delta Modulation
      3.9.1 Delta Demodulation
   3.10 Exercises

4. Passband Transmission Techniques
   4.1 Baseband to Passband
   4.2 Amplitude Shift Keying
   4.3 Frequency Shift Keying
      4.3.1 Frequency Shift Keying Spectrum
      4.3.2 FSK Receiver
      4.3.3 Infinite Gain Multiple Feedback Active Filter
   4.4 Phase Shift Keying
      4.4.1 PSK Receiver
   4.5 Differential Phase Shift Keying (DPSK)
   4.6 Differential Phase Shift Receiver
   4.7 Exercises

5. Multilevel Signaling and Bandwidth Efficiency
   5.1 Channel Capacity
   5.2 Multilevel Encoding: Bandwidth Efficiency
   5.3 Bit Error Rate
5.4 Quadrature Phase Shift Keying ......................................... 96
5.5 QPSK Modulation Using ABM Parts ................................... 97
  5.5.1 QPSK Modulation Using a Bit-Splitter ......................... 98
  5.5.2 QPSK Receiver .................................................. 101
5.6 Offset Quadrature Phase Shift Keying ................................ 102
5.7 Gaussian Minimum Shift Keying ...................................... 106
5.8 Eight-PSK .................................................................. 107
5.9 Quadrature Amplitude Modulation .................................... 110
  5.9.1 Eight-QAM ....................................................... 112
  5.9.2 Comparison of PSK and QAM ................................. 112
  5.9.3 Sixteen-Quadrature Amplitude Modulation .................. 114
  5.9.4 Two-to-Four Level Conversion ................................. 116
5.10 Clock Extraction ....................................................... 116
5.11 Costas Loop ............................................................. 120
5.12 Exercises .................................................................. 122

6. System Performance and Test Instruments ................................. 127
6.1 Noise Generator .......................................................... 127
6.2 Eye Diagram .................................................................. 127
  6.2.1 The Eye Meter ..................................................... 128
  6.2.2 Eye Diagram Application ....................................... 129
6.3 Vector/Scatter Diagram .................................................. 131
  6.3.1 Noisy QPSK Scatter Diagram ................................. 135
  6.3.2 Noisy 16-QAM Scatter Diagram .............................. 135
6.4 Clock with Jitter ............................................................ 135
6.5 Intersymbol Interference .................................................. 137
6.6 Nyquist Signal Criterion ................................................ 138
6.7 Raised Cosine Filter ...................................................... 141
  6.7.1 Square Root-Raised Cosine Filter ............................. 141
  6.7.2 Raised Cosine Filter Response ............................... 142
  6.7.3 Transmitter and Receiver Filter Impulse Response ........ 143
  6.7.4 Example ............................................................ 146
  6.7.5 Solution .............................................................. 146
6.8 Errors, Noise, and Matched Filters ..................................... 147
  6.8.1 Importing Noise into a Schematic ............................ 148
  6.8.2 Gaussian Noise Distribution Plot Using a Macro .......... 148
  6.8.3 Example ............................................................ 153
Before each simulation session, it is necessary to create a project file as shown in Figure 1. Select the small folded white sheet icon at the top left hand corner of the display.

**FIGURE 1:** Creating new project file.

Enter a suitable name in the Name box and select Analog or Mixed A/D and specify a Location for the file. Press OK and a further menu will appear so tick Create a blank project as shown in Figure 2.

This produces an empty schematic area called Page 1 where component are placed. Libraries have to be added, (Add library) by selecting the little AND symbol in the right toolbar icons. The easiest method is to select all the libraries. However, if you select Create based upon an existing project, then all previously used libraries associated with that project will be loaded. Chapter 1 uses the Fourier series expansion and Fourier transform to show the relationship between pulse width and pulse period by examining the spectra for different pulse signals. In chapter 2 we generate baseband signals and again examine the spectra for these signals. Chapter 3 examines another baseband modulation technique – the important topic called pulse code modulation (PCM). In this chapter we examine sampling, anti-aliasing
filters, quantization noise and sample and hold. Also investigated in this chapter is time division multiplexing and we construct a single channel PCM system from transmission to receiver i.e. end-to-end. Passband systems are considered in chapter 4 where systems such as frequency shift keying (FSK), amplitude shift keying (ASK) and phase shift keying (PSK) and differential forms are considered. Chapter 5 considers multi-level systems (M-ary) such as quadrature phase shift keying (QPSK) and quadrature amplitude modulation (QAM). The hierarchical method of construction is used in these systems because of the system complexity. Figure 3 shows a QPSK modulator system broken into manageable blocks. The main schematic is named Figure 5-005 and the sub-circuits in different pages are named Figure 5-005a to Figure 5-005g.

Chapter 6 looks at systems performance where we introduce home-made PSpice meters for producing eye and scatter diagrams to assess the performance of the modulation techniques in the presence of noise deliberately introduced. The raised-cosine filter family, the integrate-and-dump filters, and zero forcing equalizers are also investigated in this chapter. The power of the macro is introduced to plot statistical probability curves and a bit error rate (BER) meter.
is investigated. Chapter 7 looks at spread spectrum transmission methods showing how this technique excels in detecting and recovering wanted signals buried in noise.

ACKNOWLEDGEMENTS
I would like to acknowledge certain people who helped me directly and indirectly in producing this book. Many years ago two gentlemen, Tim O’Brien and Paddy Murray introduced me to the world of communications engineering. Much earlier than this though, when I was not quite a teenager, my dad Eddie Tobin (Wizard of the G-banjo), showed me how to build a crystal set and probably this, more than anything else, left me with a passion for this subject which continues to this day. I would like to thank Dr Gerald Farrell, Head of School of Electronics and Communications Engineering DIT, for his help and encouragement. Last, but not least, I would like to thank Joel Claypool of Morgan and Claypool publishing for taking on my five books.
CHAPTER 1

Fourier Analysis, Signals, and Bandwidth

1.1 DIGITAL SIGNALS

Data signals on a limited-bandwidth communications channel are distorted, attenuated, phase shifted, and the ever-present noise added at different stages in the transmission path. There are two basic communication channels: free space, where passband techniques transport the data in an unguided fashion (microwave point-to-point systems are not really in this category), and guided transmission systems such as cables, optical fibers, coax, etc. For example, signals received from an unguided free-space channel are a composite signal comprising signals reflected from buildings, structures, i.e., reflected multipath signals. We should be thankful for this phenomenon because without it we might not receive signals on our mobile phones where buildings, walls, etc. would block line-of-sight to the transmitter.

However, multipath signals also cause signal distortion and have to be removed in the receiver. The channel also distorts the transmitted signal with energy from each transmitted pulse “leaking” into the next transmitted pulse making it difficult for correct pulse identification in the receiver. This is a phenomenon called intersymbol interference (ISI) and the receiver cannot make correct decisions whether a pulse is present or not. One technique for reducing ISI and producing a maximum transmission symbol rate is to shape, or predistort, pulses before transmission. This removes high frequencies from the signal prior to transmission. Fig. 1.1 shows the elements of a digital transmission system.

The encoded data from the modulator is prefiltered before transmission using a root-raised cosine (RCC) filter and attempts to mimic the ideal brick-wall Nyquist filter. At the

FIGURE 1.1: Transmission system.
receiver is another RCC filter so that the overall path response has a raised cosine filter response. These filters are implemented using digital signal processors that can implement high-order cosine filters that approach the ideal filter response (8). However, transmission path characteristics are generally unknown and vary with time, so adaptive equalizers in the receiver track these changes and subsequently change the filter characteristics as needed. The equalizer flattens the channel response thus ensuring that the RCC filter works efficiently. After equalization, the received signal is root-raised cosine filtered and the demodulator can then make decisions to determine if a pulse is present or not. It does this by sampling the received signal at the center of a pulse and uses logic circuits to ascertain the presence or absence of a pulse.

1.2 BANDWIDTH
In analog communications system, bandwidth is defined as the range of frequencies over which a signal may be transmitted and received with reasonable fidelity, or minimum errors, if the signal is digital. The –3 dB bandwidth is defined as that frequency where the signal is attenuated to half of the value in the passband region. However, this is only one of many bandwidth definitions used in a digital communication channel. For example, data signals have sinc-shaped frequency spectra and one bandwidth definition is the width of the first spectral lobe measured in the spectrum. A transmission line channel has low-pass filtering characteristics and will distort a transmitted pulse by causing it to spread out in time. In the receiver, these smeared pulses will overlap making it difficult in the receiver to ascertain whether a pulse is present or not. We need to examine the shape and spectrum of different types of pulses and see how, by reshaping them using certain filters, we can minimize the problems of symbol interference (ISI). To examine channel limitations, we consider the spectrum of data signals using Fourier series/Fourier transform analysis. The Fourier series is used for periodic functions, whereas the Fourier transform (FT) is for nonperiodic functions. Thus, the FT is a generalization of the Fourier series.

1.3 PULSE SPECTRA FOR DIFFERENT PULSE WIDTHS AND PERIOD
Signal spectra are examined by selecting the fast Fourier transform (FFT) PROBE icon after simulation. The objective of this experiment is to investigate the effect on the overall shape of the frequency spectrum when the pulse width and period are changed. Set the VPULSE generator part parameters as shown in Fig. 1.2.

Be careful about the rise and fall time parameters as simulation times are increased if these values are too small. The VPULSE generator parameters are shown in Fig. 1.3.
1.3.1 Average and RMS Pulse Power

The average power for a pulse waveform with pulse width $\tau$ and period $T$ is connected across a resistance, $R$,

$$ P = \frac{1}{T} \int_0^\tau \frac{V^2}{R} \, dt = \frac{1}{TR} \left[ \frac{V^2 t}{T} \right]_0^\tau = \left( \frac{\tau}{T} \right) \frac{V^2}{R} \, \text{W}. \quad (1.1) $$

The RMS value for a pulse with $\tau = 10 \text{ us}$ and period $T = 125 \text{ us}$ is

$$ V_{\text{RMS}} = \sqrt{\frac{\tau}{T}} V = \sqrt{\frac{10 \text{ u}}{125 \text{ u}}} = 1.414 \text{ V}. \quad (1.2) $$
Thus, the RMS value for a square wave, where \( \tau = T/2 \), is \( V/\sqrt{2} \); the same as a sinusoidal signal. The transient parameters are as follows: Output File Options/Print values in the output file = 1ms, Run to time = 20ms, Maximum step size = 0.01 µs. Press F11 to simulate and from PROBE add another axis using alt PP and add the root mean square of the load voltage rms(V(RL:2)) as shown in Fig. 1.4. Note that setting the Print values in the output file to a higher value than the default value speeds up the simulation but it must be less than the Run to time.

To obtain good resolution in the spectra, we must ensure that the Run to time is large and the Maximum step size is small. This results in longer simulation times, however. Reset the pulse period to a square wave pulse, i.e., PER = PW, and simulate. Click the FFT icon and measure the spectral components shown in Fig. 1.5.

A rectangular pulse, with amplitude \( h \) and width \( \tau \) has a sinc-shaped spectrum \( \sin 2\pi f t /2\pi f t \) with magnitude \( h\tau \) at DC and spectral nulls at frequencies \( f = 1/\tau, 2/\tau, 3/\tau \), etc. If the area \( h\tau \) of the voltage pulse is constant and the width is decreased, then the spectrum widens but the peak at DC remains at \( h\tau \).

1.3.2 Unsynchronizing PROBE–Plot Axis

We may display signals in time and frequency simultaneously using the following methods. The first method is the simplest: from PROBE, select the Windows menu, create a New window and Tile Vertically. Copy the time variable across to the new window and click the FFT icon.

The second method uses the Unsynchronizing axis facility in PROBE/Plot. After simulation,
we should observe the square wave signal across the load resistance. Press in sequence, the short-cut keys `alt PP` to produce an extra plot above the original. Position the mouse on the variable `v(Rload:1)` at the bottom where it should turn red. Apply the short-cut keystrokes `ctrl C` and `ctrl V` to copy and paste the signal into the new plot. To display the two signals in the time and frequency domains simultaneously, as shown in Fig. 1.6, select from the Plot

![Figure 1.5: Spectrum of a square wave.](image1)

**FIGURE 1.5:** Spectrum of a square wave.

![Figure 1.6: PW = 10 µs and PER = 125 µs.](image2)

**FIGURE 1.6:** \( PW = 10 \, \mu s \) and \( PER = 125 \, \mu s \).
FIGURE 1.7: \(PW = 10\, \mu s\) and \(PER = 1000\, s\).

menu, **Unsynchronize Plot** and click the **FFT** icon to display the pulse spectrum (sinc-shaped discrete lines) together with the time-domain pulse train.

The spectrum in Fig. 1.7 shows that the sinc lobes are continuous when an impulse is created by making the **VPULSE** generator period (PER) much longer when compared to the pulse width (PW). The resultant spectrum flattens out around the origin (DC).

If we could make the impulse infinitely thin then the spectrum would flatten out to infinity and the spectral components will also get smaller. To investigate the effects on the spectrum, reduce the pulse width, increase the pulse period and simulate.

### 1.3.3 Fourier Transform

The Fourier transform (FT) is a generalization of the Fourier series as outlined in Section 1.3.4 to examine periodic time function, and is used when a signal is not periodic. Applying the FT to a square wave, of duration \(\tau\) and amplitude \(A\) results in a spectrum which is sinc-shaped. Consider the following analysis:

\[
v(f) = \int_{-\infty}^{\infty} v(t) e^{-j2\pi ft} dt =\int_{-\tau/2}^{\tau/2} A e^{-j2\pi ft} dt = A \left[ \frac{e^{-j2\pi ft}}{-j2\pi f} \right]_{-\tau/2}^{\tau/2} \\
= A\frac{(e^{j\pi f\tau} - e^{-j\pi f\tau})}{j2\pi f\tau} = A\tau \frac{\sin \pi f\tau}{\pi f\tau} = A\tau \sin c(\pi f\tau). \quad (1.3)
\]

The power spectral density (PSD) is defined as

\[
\{v(f)\}^2 = (A\tau)^2 \left( \frac{\sin \pi f\tau}{\pi f\tau} \right)^2. \quad (1.4)
\]
From (1.4), we see that the PSD has a maximum value of $(A\tau)^2$ at 0 Hz (DC), and the first null (a zero crossing) occurs at $\sin \pi f \tau = 0$ (a frequency $f = 1/\tau$), with 90% of the signal energy in the first lobe of the spectrum. As the pulse narrows, the main spectral lobe widens and increases the channel bandwidth requirements. Thus, transmitting infinitely thin digital pulses with no distortion requires a channel with an infinite bandwidth and a linear phase response—conditions that are not physically realizable. We may now apply the inverse Fourier transform to a narrow pulse processed through an ideal low-pass filter with cut-off frequency $f_c \ll 1/\tau$ to get back to the time domain:

$$v(t) = \int_{-\infty}^{\infty} v(f)e^{j2\pi ft}df = A\tau \int_{-f_i}^{f_i} \frac{\sin \pi ft}{\pi ft} e^{j2\pi ft}df.$$ 

(1.5)

The sinc function $\frac{\sin \pi ft}{\pi ft}$ is 1 (prove this by applying L'Hopital's rule) for small values of $f_c \tau$, so (1.5) becomes

$$v(t) = A\tau \left[ e^{j2\pi ft} \right]_{-f_i}^{f_i} = A\tau \frac{e^{j2\pi f_c t} - e^{-j2\pi f_c t}}{j2\pi t} = 2Af_c \tau \sin 2\pi f_c t.$$ 

(1.6)

A time-domain sinc signal is simulated using an equation defined in an ABM part as shown in Fig. 1.8. Enter the equation $2*A*f_c*tau*sin(2*pi*f_c*(time-k))/(2*pi*f_c*(time-k))$ by clicking the pi value to bring you into the Value box of EXP1. You could simplify this equation by canceling out the $2f_c$ terms (they were left in to avoid confusion). The first zero occurs at $\sin 2\pi f_c t = 1$, or $t = 1/2 f_c$, and other zero crossings at $n/2 f_c$. The limited channel bandwidth causes pulses in the transmitted pulse stream to spread in time and overlap so that the receiver might not be able to distinguish between 0 and 1. This is called intersymbol interference (ISI) and together with the presence of noise and other interference signals produces errors in the bit stream. The PARAM part is used to define constants including a delay factor, $k$, which is
FIGURE 1.9: Sinc signals and pulses.

included to realize a causal sinc function (note that it is not now necessary to define $\pi$ as was required in previous editions of PSpice).

Set the transient parameter **Run to time** = 100 s. The sinc-shaped spectrum is observe in Fig. 1.9. Click the **FFT** icon and use the magnifying icon to observe the duality that exists between the sinc waveforms and pulses. We also observe the Gibbs effect occurring in the passband region.

1.3.4 Fourier Series

We can use the Fourier series to synthesize a function, $f(t)$, with period $T$ and fundamental frequency $f_0$, as the sum of weighted sine and cosine functions and a DC term. For example, a square wave can be reconstructed accurately as more and more cosine and sine components are added. The signal in Fig. 1.5 is defined as

$$x(t) = \begin{cases} 
A \to & -\tau/2 < t < \tau/2 \\
0 \to & \tau/2 < t < (T - \tau/2). 
\end{cases} \quad (1.7)$$

The Fourier series for a periodic function, $f(t)$, is

$$f(t) = A_0 + \sum_{n=1}^{N} (a_n \cos 2\pi nf_0t + b_n \sin 2\pi nf_0t). \quad (1.8)$$
Here $A_0$ is a DC term and $f_0$ is the fundamental frequency equal to $1/T$. The DC coefficient is calculated for the signal defined in (1.7) as

$$A_0 = \frac{1}{T} \int_0^T f(t) \, dt = \frac{1}{T} \int_{-\tau/2}^{\tau/2} A \, dt = \left[ \frac{A \tau}{T} \right]_{-\tau/2}^{\tau/2} = \frac{A \tau}{T}. \quad (1.9)$$

If the signal is a square wave, then $T = 2\tau$, so that (1.9) is

$$A_0 = \frac{A}{2}. \quad (1.10)$$

The $a_n$ coefficient is calculated as

$$a_n = \frac{2}{T} \int_0^T f(t) \cos 2\pi nf_0 t \, dt = \frac{2}{T} \int_{-\tau/2}^{\tau/2} A \cos 2\pi nf_0 t \, dt = \frac{2A}{2\pi nf_0 T} \left[ \sin 2\pi nf_0 t \right]_{-\tau/2}^{\tau/2}. \quad (1.10)$$

Since $\sin(-x) = -\sin(x)$ and $f_0 = 1/T$,

$$a_n = \frac{2A}{2\pi nf_0 T} \left[ \sin(2\pi nf_0 \tau/2) - \sin(-2\pi nf_0 \tau/2) \right] = \frac{2A \tau}{\pi nf_0} \left[ \sin(2\pi nf_0 \tau/2) \right] = \frac{2A \pi nf_0 \tau/2}{\pi \pi \tau/2}. \quad (1.11)$$

The $b_n$ coefficient is calculated as

$$b_n = \frac{2}{T} \int_0^T f(t) \sin 2\pi nf_0 t \, dt = \frac{2}{T} \int_{-\tau/2}^{\tau/2} A \sin(2\pi nf_0 t) \, dt = \frac{-A}{\pi nf_0 T} \left[ \cos(2\pi nf_0 t) \right]_{-\tau/2}^{\tau/2} \quad (1.12)$$

$$b_n = \frac{-A}{\pi n} \left\{ \cos(2\pi nf_0 \tau/2) - \cos(-2\pi nf_0 \tau/2) \right\} = 0, \quad (1.13)$$

since $\cos(x) = \cos(-x)$. Thus the Fourier series expansion is written as

$$f(t) = \frac{A \tau}{T} + \frac{2A \tau}{T} \sum_{n=1}^{\infty} \frac{\sin(\pi n \tau/T)}{\pi n \tau/T} \cos(2\pi nf_0 t). \quad (1.14)$$

The schematic in Fig. 1.2 has the following parameters: \textbf{VPULSE} generator pulse width $PW = 1 \mu s$ equal to half the pulse period $PER = 2 \mu s$, thus the fundamental frequency of the pulse signal is 500 kHz. To display harmonic information in the output file, select the simulation menu and tick \textbf{Perform Fourier Analysis}, \textbf{Center Frequency} = 500 kHz, \textbf{Number of Harmonics} = 10, and \textbf{Output Variables}: $v$\textit{(squarewave)}. Select \textbf{Analysis/Examine Output} to display the harmonic analysis information at the end of the “.out” text file. Press \textbf{F11} and examine from \textbf{PROBE/View} the harmonic details such as amplitude, frequency, and phase in the output file as shown in Fig. 1.10. Write down the frequency, amplitude, and phase for each of the odd harmonics, i.e., 1, 3, 5, 7, etc.
We will now attempt to reconstruct this square wave using the fundamental frequency and the first three odd harmonics shown in the output file. In Fig. 1.11, connect the generators in series and set the frequency and amplitude values for each VSIN generator part as shown. Set the Analysis tab to Analysis Type: Time Domain (Transient), Run to time = 0.5ms, Maximum step size = 1 μs, and press F11 to simulate. The Run to time is set to a much larger value than the square wave period to achieve good FFT resolution. After simulation we can see how a square wave is synthesized by adding the odd harmonics. The sum of these generators is connected to a low-pass CR filter to mimic the limited low-pass filter bandwidth characteristics of a transmission line channel. Adding more generators of course would produce a better approximation to the Vin square wave included in the schematic. The synthesized and generator square wave signals are shown in Fig. 1.12.
1.4 THE VECTOR PART

A VECTOR part records digital signals at a location when placed on a schematic. Placing a VECTOR part on a schematic records digital signals only and creates a file “myfilename.vec” that may be applied as an input signal source using the FileStim generator part. This is useful for breaking larger digital circuits into smaller subcircuits to satisfy the limitations of the evaluation version. PSpice creates two columns of time–voltage pairs in the VECTOR file. To see how this part is used, create a pseudorandom binary sequence (PRBS) generator using the circuit shown and place the VECTOR part (a little square box symbol) as shown in Fig. 1.13.

Fig. 1.14 shows the necessary parameters to be set when simulating digital circuits. The flip-flops must be initialized into a certain state. Select the simulation setting menu and then select the Options tab. In the Category box select Gate-Level Simulation. This will then show the Timing Mode- select Typical. Make sure to select Initialization all flip-flops to 1. Select and right click the VECTOR part. Select Edit Properties and the directory and file name are specified in the FILE parameter. Select FILE and in the second column enter C:\Pspice\Circuits\signalsources\data\prbs1.txt.
Other VECTOR parameters are as follows:

- **POS**: this is the column position in the file with values ranging from 1 to 255.
- **FILE**: the location and file name must be specified, e.g., C:\signalsources\prbs.txt.
- **RADIX**: valid values for VECTOR symbol attached to a bus are B[inary], O[ctal], and H[ex].
- **BIT**: if the VECTOR symbol is attached to a wire, the bit position within a single hex or octal digit.
- **SIGNAMES**: this is the wire segment name in the file header where the VECTOR part is connected.

The PRBS clock and output signals are shown in Fig. 1.16.
The VECTOR data file created, as shown in Fig. 1.15, is 01010101 and consists of three parts: (1) * Created by PSpice (a comment line), (2) the header (the wire segment name is called out), and (3) a column pair consisting of time and data level amplitudes (i.e., 1 or 0).

The length of the data file, created using the VECTOR part, depends on the transient Run to time value.

1.5 EXERCISE
1. Investigate Fourier analysis for Triangle, Sawtooth, and rectified sinusoid signals.
CHAPTER 2

Baseband Transmission Techniques

2.1 BASEBAND SIGNALS

Fig. 2.1 shows a digital data sequence \( x(n) \) encoded onto a line code \( s(t) \) for transmission over a limited-bandwidth channel. Signals not carrier modulated are referred to as baseband signals. The encoded data, in unipolar or bipolar format, is sent directly to the channel so that the data spectrum starts at DC.

![Figure 2.1: Baseband coder.](image)

Line coding a data stream enables a receiver to extract a timing clock signal from the received signal so that the transmitter and receiver operate in synchronism. The code should produce a signal with a suitable spectrum consistent with minimum bandwidth. For example, long sequences of 0’s or 1’s in the transmitted sequence produce DC in the spectrum that should be avoided as the public telephone network contains transformer and capacitive-coupled networks that will not transmit DC. Examples of baseband codes are nonreturn to zero (NRZ—also called NRZ-L), Manchester (or biphase, Bi\( \phi \)), differential Manchester, and alternate mark inversion (AMI). A communication channel has many forms: free-space, twin-pair cable, coax cable, optical fiber, and the bandwidth for each channel is different placing limits on the amount of information it can carry at any one time. Channel noise also limits the information transmission rate and causes errors to be detected in the receiver.

2.2 BASEBAND SIGNAL FORMATS

Baseband signals have two formats: polar and bipolar. The bipolar format has advantages over the polar format because it has no DC content when equal numbers of 0’s and 1’s occur in the transmitted message signal. The polar format (also referred to as unipolar) contains DC and cannot be transmitted over a telephone network that uses transformer/capacitor coupling. Errors occur in noisy systems where strings of 1’s or 0’s change the decision threshold making it difficult for the receiver to detect whether a 1, or 0, is present. The bipolar format, for
the same signal to noise ratio, requires half the average power compared to polar signals. TTL (transistor–transistor logic—same format as nonreturn to zero) represents a logic level 0 (a space) as 0–0.8 V, and a logic level 1 (a mark) as 2–5 V, with output current less than 15 mA.

2.2.1 NonReturn to Zero (NRZ) Coding

The circuit in Fig. 2.2 produces a two-level polar nonreturn to zero (NRZ) data that has poor coding properties and with poor clock extraction properties. Here the pulse width is equal to the bit interval, and we will see from the spectrum plotted in PROBE using the FFT icon that there is a DC component making it unsuitable for transmitting data over the public switched telephone network (PSTN).

The input data is an ASCII file 00100101…NRZ1.txt created in Notepad© (see Fig. 1.15 in Chapter 1). In this example, the header name is the same as the input wire segment name NRZdata and the header filename must be separated from the first file pair “0s 0” by a blank line.

2.2.2 FileStim Generator

The FileStim generator part applies the signal recorded by the VECTOR part, examined in Chapter 1, as a digital input source. The FileStim generator has two attributes: the first is the FileName attribute, where the file location and name is entered, e.g., C:\Pspice\Circuits\signalsources\data\NRZ1.txt; the second attribute is the SigName attribute that specifies the name of the wire where it is attached (select a wire and enter a name using the Net Alias icon on the right toolbar). If you use FileStim with a nondigital signal, you will get “Circuit Too Large” error message displayed. Set the transient Run to time to 20ms and simulate to produce the signals shown in Fig. 2.3.

To investigate the NRZ signal spectrum, we need to use a much longer signal, so replace the FileStim generator with a 1ms DigClock. Set Output File Options/Print values in the
output file to 100ms, Run to time to 1 s, and Maximum step size = 1 µs and press the F11 key to simulate. Select the FFT icon and observe a 2.5 V DC component present in the spectrum (2.5 V is the average value for a 0–5 V pulse).

2.2.3 NRZ-B

Turning NRZ into bipolar NRZ-B (B stands for bipolar) form eliminates DC from the spectrum. Fig. 2.4 shows how to do this conversion with the input data applied using a STIM1 part. Set Run to time to 1ms, Maximum step size (left blank), and simulate with the F11 key.
2.3 RZ ENCODING AND DECODING

Clock signals are generated in the receiver from positive or negative transition levels to zero. The pulses are shifted back by a quarter of a clock period to ensure that the sampling points occur in the centers of the first halves of the bit intervals. This coding requires extra bandwidth because the actual pulse is half the size of the bit interval, and wastes power in transmitting a three-level signal. A continuous stream of 1’s produces a DC level and causes problems in communication networks that cannot transmit DC. Fig. 2.6 applies the NRZ signal from the
FIGURE 2.7: RZ and RZ-B signals.

previous schematic to produce a Return to Zero (RZ) line coding with a transition in the middle of every bit.

Set Run to time = 10ms and simulate with F11 key. The RZ and RZ-B signals are shown in Fig. 2.7. However, to display the FFT of a digital signal in PROBE means attaching a resistor from the required node to ground; otherwise you get the message informing you that the FFT of a digital trace will not be displayed. The output from the comparator is RZ-B with no DC in the spectrum.

2.3.1 RZ to NRZ Decoder

Fig. 2.8 shows an RZ to NRZ signal decoder. The monostable (74121) pulse width is set by C1 and R1 connected to pins 10 and 11. However, these components are not modeled in PSpice and are included only for completeness, so we select the 74121 component and set the pulse width to 25 µs.

When using digital devices such as registers, flip-flops, etc., it is important to reset Initialize all flip flops from X to 0, or 1. Failing to do this may result in red lines displayed in PROBE because the output impedance levels are indeterminate. Select the Simulation Setting menu and then select the Options tab. In the Category box select Gate-Level Simulation.
FIGURE 2.8: RZ to NRZ decoder.

This will then show the **Timing Mode** - select **Typical**. Make sure to select **Initialization all flip-flops to 1**. For most simulations, set the **Timing Mode** to **Typical** and the **Default A/D Interface** to **Level 2**. An alternative is to attach an initialization circuit to the CLR pin. Such a circuit may be a CR low-pass filter attached to a 5 V DC source that charges up the capacitor on switch-on to the 5 V source. The capacitor will charge up in approximately five time constants to 5 V and is equivalent to applying a HI condition to the IC CLR pin.

Set **Output File Options/Print values in the output file** = 100 ns, **Run to time** = 1ms, **Start saving data after** = 0.2ms, and **Maximum step size** = 1 µs, and simulate with the **F11** key. Position the cursor accurately on a leading, or lagging edge, using the icon that moves the cursor to the next digital transition. The NRZ output signal is clearly seen in Fig. 2.9.

FIGURE 2.9: NRZ output signal.
2.4 MANCHESTER ENCODING AND DECODING

It is not desirable to have DC content in signals used in the public switched telephone network (PSTN), especially where lines are transformer coupled. The unipolar Manchester code is a two-level code with transitions at the bit centers between two levels (high–low and low–high). Each “1” has a transition from high to low and each “0” has a transition from low to high. The transitions make for easier clock extraction at the receiver. The Manchester code (biphase) requires twice the bandwidth compared to NRZ and NRZ-B and has a DC component. Fig. 2.10 shows a schematic for producing a unipolar Manchester signal.

The file “Manchesterdata3.txt,” a data file — “101011100,” was created in a text editor Notepad (see Section 1.4) consisting of a header line “* Created by PSpice”; the line segment name data is separated by a space from the time–voltage data. The Manchester data input is applied using a FileStim part where the second line, SigName = data, is the wire segment name connecting the FileStim. Filename = C\signalsources\manchesterdata3.txt. Set the Output File Options/Print values in the output file to 1ms, Run to time to 10 ms, and Maximum step size (left blank). Press F11 to display the signals as in Fig. 2.11.

2.4.1 Manchester Unipolar to Bipolar Encoding

Fig. 2.12 shows how to transform a unipolar Manchester code into bipolar form. The bipolar Manchester code has no DC in the spectrum and could be used in the public switched telephone network (PSTN).

Set the Output File Options/Print values in the output file to 1ms, Run to time to 200ms, and Maximum step size (left blank). Simulate with F11 key to display the bipolar Manchester signals shown in Fig. 2.13.
22 PSPICE FOR DIGITAL COMMUNICATIONS ENGINEERING

FIGURE 2.11: Unipolar Manchester signals and the DC component.

FIGURE 2.12: Bipolar Manchester code.

FIGURE 2.13: Bipolar Manchester signals.
FIGURE 2.14: Biphase production.

An impulse at 3 ms occurs because the two input signals are separated slightly in time and when applied to the XOR gate cause an ambiguity resulting in an impulse. Use the magnifying tool to measure the difference between the two pulses (approximately 31 ns between the two edges). Observe how the bipolar Manchester code signal has no DC in the spectrum.

2.4.2 Manchester (Biphase) Decoding

Two XOR gates connected as shown in Fig. 2.14 will generate a biphase signal for the input data applied using a stimulus STIM1 part named Datagen.

Fig. 2.15 shows how looping functions create a repeating input data pattern. The first command is STARTLOOP and ended by GOTO STARTLOOP 100 times. Note that the command GOTO LOOP 2-1 TIMES produces an infinite loop. Enter the test signal “1011011.” In this example, command2 is +0us 0, command3 is +100 us 1, etc., and repeated 100 times.

The clock signal is a DigClock part with parameters as shown in the schematic. The input line to the NOR device is labeled using the Net Alias icon, typing in a name such as Data and dragging the little box to the wire segment. This is useful for identifying signals when plotted in PROBE. Draw a box around a component, or a group of components, by selecting the box icon from the right-hand menu. After drawing the box, click on a box line to change the line properties if you so wish, i.e., to dotted format as shown above. To observe the DC content in the spectrum, terminate the output with a resistance in order to use the FFT function on digital level signals. Set Output File Options/Print values in the output file to

FIGURE 2.15: STIM1 parameters.
2.4.3 Differential Manchester Coding

The schematic in Fig. 2.17 uses the data text signal from the previous circuit. Differential Manchester code, used in the Ethernet, has transitions in the middle of the pulse.

Digital warnings after simulation are often associated with PROBE screen waveforms that have indeterminate states. The flip-flops must be initialized into a certain state. Select the Simulation Setting menu and then select the Options tab. In the Category box select Gate-Level Simulation. This will then show the Timing Mode - select Typical. In the Timing Mode change Digital Setup from Typical to Maximum. The default Initialize all flip flops is set to X. Failing to change this will result in the flip-flop being in an undetermined state and shows up in the plot as two red lines, so change Initialize all flip flops to 0. The differential Manchester signal in Fig. 2.18 shows transitions occurring in the middle of the pulse that are determined by 1 or 0 and the preceding bit. The transition stays the same as the preceding one, if the current bit is 0, but switches when the bit is 1. Once the start bit is known, all the following bits are obtained, which gives this code an advantage. For example, this code ensures no problems if wires are mistakenly reversed in a connector at the time of installation.
2.5 ALTERNATE MARK INVERSION ENCODING

Alternate mark inversion (AMI) encoding is also known as bipolar return to zero (BPRZ) and is a three-level line code used in the 30-channel TDM PCM E1 system. Return to zero–alternate mark inversion RZ AMI signal has every alternate “one” polarity reversed, with zeros represented as a zero DC level. The alternating characteristic is an advantage because it is easy to recognize a line code violation. AMI production is shown in Fig. 2.19.

The data stimulus for this circuit is provided by a STIM1 part called Datagen and is the same as that used previously where a digital signal pattern is created by looping the signal three times. The alternating mark inversion signal is clearly observed in Fig. 2.20.

The FFT icon, when selected, demonstrates that no DC is present in the AMI spectrum in Fig. 2.21.

2.5.1 AMI Decoding

Fig. 2.22 shows the format for the AMI text file inputted using a VPWL_F.RE_FOREVER part. The text file is located at C:\Pspice\Circuits\signalsources\data\ami.txt and was created by copying the AMI output signal from the PROBE output from the last simulation results.

The schematic in Fig. 2.23 is for recovering data from an AMI encoded signal. This AMI decoder contains a monostable multivibrator with Schmitt-trigger inputs. The components $R_2$ and $C_2$, connected to the supply voltage, set the pulse width in a real circuit but are not

FIGURE 2.18: Differential Manchester signals.

FIGURE 2.19: AMI coder.
FIGURE 2.20: AMI waveforms.

FIGURE 2.21: Spectrum of AMI signal.

FIGURE 2.22: AMI signal parameters.

FIGURE 2.23: AMI decoding.
modeled in PSpice. We must set the pulse width by selecting the 74121 IC, Relick and select Edit Properties. In the spreadsheet, enter the pulse width in the PULSE box (the default value is 30 ns). The ABS part converts the bipolar signal into a polar signal.

The recovered NRZ is now shown in Fig. 2.24.

2.6 DUO-BINARY BASEBAND SIGNALING

Duo-binary or partial response signaling is a technique where intersymbol interference (ISI) is added in a controlled manner to the transmitted stream for the purposes of reducing the filter design and the need for Nyquist filtering requirements [ref: 3]. Fig. 2.25 shows a duo-binary
or partial response signaling baseband technique, with a pulse-shaping filter to overcome ISI by introducing a controlled amount of ISI.

The NRZ-B input data is applied using a STIM1 part called Data with parameters as shown in Fig. 2.26.

This data stream is then delayed by an amount equal to the bit period, $T_b$. The delay is achieved using a correctly-terminated transmission line called a $T$ part [ref: 9 Appendix A]. Correctly-terminating a line means placing a resistance across the input and output terminals whose value is the same as the characteristic impedance. The composite signal from the output of the SUM part is then filtered using an ideal pulse-shaping low-pass filter using a LOPASS part with a cut-off frequency equal to $0.5 \times 1/T_b$ as shown in Fig. 2.25. The three-level signal in Fig. 2.27 has positive and negative amplitudes that are twice the NRZ-B amplitude.

The transfer function for the duo-binary system, assuming an ideal low-pass filter, is

$$H(f) = H(f)_{\text{ideal}}(1 + e^{-jT_b}) = H(f)_{\text{ideal}}(1 + e^{-j2\pi f T_b})$$

$$= H(f)_{\text{ideal}}(e^{j\pi f T_b} + e^{-j\pi f T_b})e^{-j\pi f T_b}. \quad (2.1)$$

FIGURE 2.27: Duo-binary waveforms.
We may write the overall transfer function by applying Euler’s expression to (2.1):

\[ H(f) = \begin{cases} 
2 \cos(\pi f T_b) e^{-j \pi f T_b} & \text{for } |f| \leq 0.5 x 1/T_b \\
0 & \text{elsewhere.} 
\end{cases} \tag{2.2} \]

The duo-binary frequency response in Fig. 2.28 was obtained by renaming the input wire segment to sine and changing the simulation profile to AC.

Applying the inverse Fourier transform to equation (2.2), yields \( h(t) \) as

\[ h(t) = \frac{\sin(\pi t/T_b)}{\pi t/T_b} + \frac{\sin(\pi (t - T_b)/T_b)}{(t - T_b)/T_b}. \tag{2.3} \]

The impulse response is obtained by changing the input signal wire segment to impulse. This results in the display shown in Fig. 2.29.

### 2.6.1 Use of a Precoder in Duo-Binary Signaling

Fig. 2.30 shows a scheme for utilizing a precoder to overcome the problem of error propagation.

The flip-flops must be initialized to a certain state. Select the Simulation Setting menu and then select the Options tab. In the Category box select Gate-Level Simulation. This will then show the Timing Mode - select Typical. In the Timing Mode change Digital Setup from Typical to Maximum. The Initialize all flip flops is set to All 0. Failing to do this will
result in the flip-flop being in an undetermined state, i.e., All X, and shows up in the plot as two red lines. The waveforms for this circuit are shown in Fig. 2.31.

### 2.7 INTEGRATE AND DUMP MATCHED FILTER BASEBAND RECEIVER

The Integrate and Dump matched filter in Fig. 2.32 applies an NRZ-B signal and noise to an integrator. Each bit period is integrated, and bit recovery is possible, provided the noise is random.
FIGURE 2.31: Precoder waveforms.

FIGURE 2.32: Integrate and Dump matched filter.
FIGURE 2.33: NRZ-B data.

The input NRZ-B data was created by entering values into a text editor (or, alternatively, use the file created by a Vector part), as shown in Fig. 2.33. The signal has a period of 100 µs, and the ASCII file is then applied to the circuit using a FileStim part. Noise picked up in the transmission path is simulated by applying another ASCII signal using a VPWL_F_RE_FOREVER generator part located at the directory C:\signalsources\noise\noise_info2.txt. We vary the noise amplitude using a GAIN part, or alternatively, we can use the voltage-scaling factor (VSF—one of the generator parameters). Thus, doubling the VSF value doubles the noise amplitude.

The time constant, \( \tau = CR = 2\text{ms} \), is much greater than the 100 µs symbol period. The baseband waveforms in Figs. 2.34 and 2.35 show the NRZ-B signal integrated over a symbol period. At the end of each 100 µs period, switch S1 discharges the capacitor. S2 is a sampler switch that operates in the middle of the ramp for 1 µs, thus allowing only 1 µs of the ramp through. The decision threshold circuit is simple using an ABM1 part and an IF THEN ELSE statement, e.g., If(V(vin)) > = v0max, 4, 0). This states that if the input sampled signal is greater than the variable defined in the PARAM part, \( v0max = 100 \text{mV} \), then the output is 4 V, otherwise it is 0 V.

The D-type flip-flop operates to give a pulse when clocked. Repeat the simulation for different noise amplitudes. Repeat the above exercise but include a passband filter having a 1000 Hz bandwidth.

2.7.1 Example
A 100 baud 1 V NRZ signal contains additive white Gaussian noise (AWGN) and is band-limited to 1000 Hz. This is applied to an Integrate and Dump matched filter and the error probability is estimated by assuming equal probability of a 1 or 0 occurring. The threshold
is determined by assuming zero mean noise and with variance $\sigma^2 = 0.125$. The noise variance is

$$
\sigma_0^2 = \frac{\sigma^2 T}{2B} = \frac{0.125 \times 0.01}{1000} = 1.25 \times 10^{-6} \Rightarrow \sigma_0 = 1.12 \times 10^{-3}.
$$

After integration, the 1 V signal becomes 0.01 (VT) and the decision threshold becomes 0.005 V. Assuming a base probability for 0 and 1 as 0.5, the error probability is

$$
0.5 Q\left(\frac{0.005}{\sigma_0}\right) + 0.5 Q\left(\frac{0.005}{\sigma_0}\right) = Q\left(\frac{0.005}{1.12 \times 10^{-3}}\right) = Q(4.46) = 3.5 \times 10^{-6}. \quad (2.4)
$$

The decision threshold is 0.5 V for a 1 V pulse and noise variance $\sigma^2 = 0.125$ (without the Integrate and Dump circuit). The error probability of a noisy sample with standard deviation of 0.35 exceeding 0.5 V when a zero is transmitted is less than $-0.5$ V. The bit error rate for a
transmitted “1” is calculated as $Q(0.5/0.35) = Q(1.42) \sim 8 \times 10^{-2}$ etc., which is a poor BER rate.

2.8 EXERCISES

1. A modification to the DB system in Fig. 2.35 produces a response with a null at DC, and is useful for channels having a poor low-frequency response.

The modified Duo-binary frequency response is shown in Fig. 2.36. The modified Duo-binary signals are shown in Figure 2.37.

**FIGURE 2.35:** Modified duo-binary transmitter and receiver.

**FIGURE 2.36:** Modified duo-binary frequency response.
CHAPTER 3

Sampling and Pulse Code Modulation

3.1 SINGLE-CHANNEL PULSE CODE MODULATION

Telephone transmission lines have a bandwidth of 300–3400 Hz (bandwidth of 3.1 kHz). To transmit multiple voice signals simultaneously on the same transmission line requires the signals to be time- or frequency-division multiplexed. In this chapter, we investigate time-division multiplexed pulse code modulation (TDM PCM—Alec Reeves: 1902–1971), which requires sampling the speech signals at a minimum rate of twice the highest frequency contained in the speech. The speech is filtered with a low-pass filter that has a cut-off frequency of 3400 Hz. Filtering prevents a phenomenon called aliasing where extra frequencies are produced if the sampling is not at the Nyquist rate, i.e., 6800 samples/s (Nyquist sampling theorem) [ref: 8 Appendix A]. This rate is increased to 8000 Hz sampling rate to allow for nonideal filtering in order to recover the signal completely in the receiver. Fig. 3.1 shows a single-channel PCM block diagram.

To limit the frequency spectrum of the speech to a value below half the sampling rate, and prevent aliasing, the analog speech signal is low-pass filtered with a cut-off frequency of 3.4 kHz. This filter is called an antialiasing filter because it prevents aliasing frequencies appearing in the output. The filtered speech is then sampled and companded. Companding is formed from the words compressing/expanding where the signal is first compressed to improve the signal to noise ratio and expanded in the receiver. In PCM systems, the initial signal processing occurs in an integrated circuit called Codec, where the 12-bit digital code is compressed to 8 bits. Each sampled value is assigned one of 256 ($2^8 = 256$) discrete levels for the maximum amplitude range of $-1$ V to $1$ V or $2$ V, where $V$ is the largest signal applied and each quantized sample is represented by an 8-bit code.

Encoding a quantization level into 8-bit words produces a single-channel transmission bit rate equal to 8 bits $\times$ 8 kHz = 64 kbps. The European E1 primary multiplexing format is
the 30-channel 2.048 Mb/s bit rate, whilst the T1 system used in Japan and North America has a 1.5 Mb/s bit rate. The E1 system has 8-bit 32 time slots to accommodate 30 voice signals plus two other time slots for framing, alarm, and signals other than voice information (i.e., ringing tone, engaged tone, call forward, etc.). Time slot 16 accommodates two signaling channels, and hence we need a multiframe comprising 16 frames with a period of 2ms containing 4096 bits.

3.2 COMPANDING CHARACTERISTICS

Linear quantizing is where an equal number of quantized levels are allotted for low- and high-level signals and results in a very poor signal to quantization noise ratio. A nonlinear system increases the number of decision levels for small amplitude signal levels and yields an overall improved quality. In the European E1 system, prior to transmission, the 12-bit digital signal is compressed to 8 bits using a 15-segmented $A$-law, where the compression parameter is $A = 87.6$. The compressor characteristic is defined for two regions as

$$F_A(x) = \text{sgn}(x) \left[ \frac{A |x|}{1 + \ln(A)} \right] \quad \text{for} \quad 0 \leq |x| \leq \frac{1}{A}$$

for $0 \leq |x| \leq 0.0114$ \hspace{1cm} \text{(3.1)}

$$F_A(x) = \text{sgn}(x) \left[ \frac{1 + \ln(A |x|)}{1 + \ln(A)} \right] \quad \text{for} \quad \frac{1}{A} \leq |x| \leq 1$$

$$= \text{sgn}(x) \left[ \frac{1 + \ln(87.6 |x|)}{1 + \ln(87.6)} \right] \quad \text{for} \quad 0.0114 \leq |x| \leq 1. \hspace{1cm} \text{(3.2)}$$

The signum function is defined as

$$\text{sgn}(x) = \begin{cases} -1 & \text{for } x < 0 \\ 1 & \text{for } x > 1, \end{cases}$$

and is zero when $x$ is zero. The LIMIT part defines the signal constraints in (3.1) and (3.2). Note that \textbf{LOG()} in PROBE is natural log, whereas \textbf{LOG10()} is log to the base 10. The continuous $A$-law companding function is simulated by entering the following expression into the \textbf{ABM1} part using an IF THEN ELSE statement to manage the input level constraints as:

$$\text{if}(V(%IN) < = 1/A, \text{sgn}(V(%IN)) \times A \times \text{abs}(V(%IN))/(1 + \log(A)), \text{sgn}(V(%IN)) \times (1 + \log(A \times \text{abs}(V(%IN))))/(1 + \log(A))).$$
The $\mu$-law characteristic is defined as
\[ F_\mu(x) = \text{sgn}(x) \left( \frac{1 + \ln(255 |x|)}{\ln(1 + 255)} \right) \quad \text{for} \quad 0 \leq |x| \leq 1, \]
where $u = 255$. This is entered in the ABM1 part as $\text{sgn}(V(%IN))^*(\log(1+u*\text{abs}(V(%IN))))/\log(u)$. Sweep the input voltage as shown in Fig. 3.3. Set Analysis Type: DC Sweep, Sweep Variable = Voltage source, Name: vin, Linear, Start Value = 0, End Value = 1, Increment = 0.0001. The $A$-law characteristic shows higher amplitude input signals compressed.
This nonlinear companded PCM (CPCM) uses a fixed number of quantized levels, but with a much higher percentage of the levels assigned to smaller amplitude input voltages. A larger signal will have a smaller number of quantizing levels, but this is OK since statistically these signals are not significant. At the receiver, there is an identical nonlinear expander. “Compression and expansion” is known as companding. There are two segmented nonlinear instantaneous companding systems: \( A \)-law for the European PCM system and \( \mu \)-law for the American system. Replace the VDC source with a VSIN part and observe the effect on the signal in Fig. 3.4.

The complete \( \mu \)-law companding characteristic is simulated with the following transient parameters: \textbf{Analysis Type}: DC Sweep, \textbf{Sweep Variable} = Voltage source, \textbf{Name}: vin, \textbf{Linear}, \textbf{Start Value} = -1, \textbf{End Value} = 1, \textbf{Increment} = 0.0001. The complete \( \mu \)-law characteristic is shown in Fig. 3.5. In practice, however, the compression is not done at the analog level but on a 12-bit digital signal, where it is compressed to an 8-bit format using a segmented \( A \)-law characteristic. We may use a Table part or the \textbf{Value List} in the \textbf{DC Sweep} menu to achieve the segmentation or chords. In the latter case, tick \textbf{Value List} in the \textbf{Sweep Type} menu and enter in the \textbf{Values} box the following values: 0, 0.0156, 0.0313, 0.0625, 0.125, 0.25, 0.51. The segmented \( A \)- and \( \mu \)-law characteristics are shown in Fig. 3.5 with the complete \( \mu \)-law characteristic shown in the left panel along with swept input signal.

The first bit of the eight bits represents the signal sign. The next three bits \((2^3 = 8)\) represent the segment the signal lies in, and the last four bits are for each of the \(2^4 = 16\) decision levels where bits are transmitted serially, sign bit first. The dynamic range for the
$A$-law is $20\log_{10}(4096/15) = 48.7$ dB, where 0–15 spans the first chord. The dynamic range for the $\mu$-law is $20\log_{10}(8159/31) = 48.4$ dB, 0–31 spans the first chord.

3.3 SAMPLING

To recover an analog signal from a sampled signal, and with no aliasing (extra frequencies generated), requires that the signal be sampled in the first place by a sampling frequency that is greater than twice the highest frequency component in the analog signal. A 1 kHz analog signal must be sampled at a rate at least equal to 2 kHz (the work of C. Shannon, 1948), to preserve and recover the waveform exactly. Sampling a signal at a rate below twice its highest frequency produces aliasing components, which are extra aliasing frequencies in the recovered signal. Low-pass filtering the signal prior to sampling prevents aliasing in a fixed sampling system. The sampling rate in PCM telecommunication systems is 8 kHz (period $T_P = 125 \mu s$), which means a 4 kHz theoretical maximum input signal frequency.

An ideal brick-wall filter that cuts off abruptly at 4 kHz is not practical, so the input signal frequency is limited to 3.4 kHz with a finite filter transition region width. A square wave using a \texttt{VPULSE} part is applied to a second-order Sallen and Key active low-pass filter in Fig. 3.6. The square wave simulates a complex signal such as speech because the square wave contains harmonics from DC to infinity. The filter output is then sampled using a dual sampling switch IC CD 4016.
Design a second-order antialiasing Sallen and Key LPF filter to extract the fundamental harmonic of the square wave, and attenuate higher order harmonics. The filter ensures that frequencies greater than twice the sampling frequencies are sufficiently attenuated and thus eliminates alias components.

### 3.3.1 Sallen and Key Antialiasing Active Filter

The antialiasing second-order Butterworth Sallen and Key active filter is designed to meet the specification $A_{\text{max}}$, $A_{\text{min}}$, $\omega_c$, and $\omega_s$. These are the passband and stopband gains, passband and stopband edge frequencies, respectively. The passband cut-off frequency is $\omega_c = \frac{1}{C_1 R_1}$ and the gain in the passband region is $20 \log(1 + R_a / R_b)$ with the roll-off rate of $-40$ dB per decade. We need to extract the fundamental harmonic (the 1 kHz fundamental harmonic component) from the square wave. The following analysis yields a low-pass transfer function, which produces maximum attenuation of 1 dB at the fundamental component of 1 kHz. The 3 kHz, or third harmonic, should be attenuated by 12 dB [ref: 1 Appendix A]. The frequency correction factor is calculated:

$$\epsilon = \sqrt{(10^{0.1 A_{\text{max}} - 1}) (10^{0.1 A_{\text{min}} - 1})} = 0.508.$$  

(3.3)

The filter order is calculated using the following expression:

$$n = \frac{\log_{10} \left[ \frac{(10^{0.1 A_{\text{min}} - 1})}{(10^{0.1 A_{\text{max}} - 1})} \right]}{2 \log_{10} \left( \frac{\omega_s}{\omega_p} \right)} = \frac{\log_{10} \left[ \frac{(10^{0.1 1.2 - 1})}{(10^{0.1 1.1 - 1})} \right]}{2 \log_{10} \left( \frac{2 \pi 3000}{2 \pi 1000} \right)} = 1.84 \approx 2.$$  

(3.4)

A normalized second-order Butterworth approximation loss function is:

$$A(\$) = \$^2 + 1.414\$ + 1.$$  

(3.5)
This function has a normalized cut-off frequency of 1 \( \text{r/s} \), that is denormalized by replacing $ with:

\[
$ = \left( \frac{\varepsilon^{1/n}}{\omega_c} \right) = \frac{s}{2\pi 1000} = \frac{s}{6283}. \tag{3.6}
\]

Substitute this value into (3.5) and invert to yield the transfer function:

\[
H(s) = \frac{1}{s^2 + \omega_p/Q + \omega_p^2} = \frac{1}{(6283/0.712)^2} \times \left( \frac{6283/0.712}{s} \right)^2 + 1.414 \times \left( \frac{6283/0.712}{s} \right) + 1)
\]

\[
= \frac{7.787 \times 10^7}{s^2 + 1.247 \times 10^4 s + 7.787 \times 10^7}. \tag{3.7}
\]

We are now in a position to obtain component values by comparing the denominator transfer function coefficients of (3.7) to the standard second-order transfer function denominator \( s^2 + \omega_p/Q + \omega_p^2 \) and the transfer function is

\[
E_0 \over E_i = \frac{k / C^2 R^2}{s^2 + s(3 - k)/CR + 1/C^2 R^2}. \tag{3.8}
\]

Consider the non-s coefficient term

\[
\omega_p^2 = 7.787 \times 10^7 = \frac{1}{(CR)^2} \Rightarrow \omega_p = 8824.4 = \frac{1}{CR}. \tag{3.9}
\]

The capacitance is calculated for \( R = 100 \, \text{k\Omega} \) as

\[
C = \frac{1}{8824.4 \times 10^5} = 1.139 \, \text{nF}. \tag{3.10}
\]

Compare the s coefficient terms in the denominator of (3.8) and (3.7), and substitute the values for \( C \) and \( R \):

\[
\frac{\omega_p}{Q} = \frac{3 - k}{CR} = 1.247 \times 10^3 \Rightarrow 3 - k = 1.247 \times 10^4 \times 1.139 \times 10^{-9} \times 10^5 = 1.42. \tag{3.11}
\]

The passband gain is

\[
k = 3 - 1.42 = 1.58 = 1 + R_e/R_o \Rightarrow R_e/R_o = 0.58 \, \text{or} \, R_e = 0.58 R_o. \tag{3.12}
\]

If \( R_e = 10 \, \text{k\Omega} \), then \( R_o = 5.8 \, \text{k\Omega} \) and a gain of 1.58. Set Output File Options/Print values in the output file to 1 ms, Run to time to 10 ms, Maximum step size = 10 u, and press the F11 key to simulate. The frequency spectrum for a 1 kHz square wave input signal and other signals are shown in Fig. 3.7.
3.3.2 Speech Signals

A speech signal is applied using the VPWL.F.RE_FOREVER generator as shown in Fig. 3.8. This generator reads in a speech file “speech.txt,” created and stored on the hard drive in the “signalsources” directory. Sampling uses the Sbreak switch part operated by a DigClock part. Set Output File Options/Print values in the output file to 0.1s, Run to time to 1s, and Maximum step size = 100 u and press the F11 key to simulate. Fig. 3.9 shows the input and
output speech signals in the time domain, but to see the effect of filtering we need to look at
the signals in the frequency domain. Note how the sampled speech sidebands are now located
at the sampling frequency, and at multiples of the sampling frequency. The reduced frequency
range is evident from the middle trace, which means that the aliasing components are reduced
to such a small value that effectively, they are zero.

### 3.3.3 Sample and Hold

The schematic in Fig. 3.10 investigates how a signal sample is held at a certain level in between
sampling times. An ideal opamp part is used instead of a ua741 operational amplifier in order

![](image)
46 PSPICE FOR DIGITAL COMMUNICATIONS ENGINEERING

to satisfy the evaluation version limits. Set **Output File Options/Print values in the output file** to 100 ns, **Run to time** to 10 ms, and **Maximum step size** = 10 u and press the F11 key to simulate and observe, in the right pane of Figure 3.13, the effect of not buffering the sample and hold circuit components.

A square wave signal has a spectrum that extends to infinity and may be used to simulate complex signals such as speech. The sampling frequency in modern telecommunication systems is 8 kHz, which is a sampling period of $T = 125 \mu s$. This means that the maximum frequency of the input signal should not be greater than 4 kHz. This is achieved by band-limiting the input analog signal to 4 kHz using an active low-pass filter. The antialiasing filter specification defines $A_{\text{max}}$, $A_{\text{min}}$, $\omega_c$, and $\omega_s$ as the passband gain, stopband gain, the passband edge frequency, and the stopband edge frequency, respectively. The passband cut-off frequency is $\omega_c = \frac{1}{C_1 R}$ and the passband gain is $20 \log (1 + R_a / R_b)$. We may investigate aliasing by reducing the sampling rate below the minimum rate required, which for a 3.4 kHz modulating signal is 6.8 kHz and lower.

The undersampled signal spectrum is compared to the correct sample spectrum. Undersampling the input signal will produce aliasing components in the recovered spectrum and contains extra aliasing frequency components. This manifests itself as distortion in the recovered signal. Increase the sampling rate to 20 kHz and observe any changes in the recovered signal. Oversampling reduces the “burden” on the antialiasing filter and thus a simpler filter design would suffice. In the sample and hold circuit, the hold capacitor, $C_h$, across the hold resistance, $R_h$, “holds” the instantaneous amplitude of the sampled signal over the sampling period until the next sample and thus produces less distortion in the recovered signal. We may observe the effect of not using a hold circuit by observing the frequency spectrum and measuring the distortion. Sample and hold may introduce aperture error, however, depending on how much the signal changes during the period of sampling. We need to calculate the slope of the 1 kHz sinusoidal signal and multiply it by the sampling period, $T_s = 125 \mu s$, to calculate the maximum amount of error introduced. Consider the instantaneous value of a sinusoidal to be sampled:

$$v_c(t) = V_c \sin \omega t = 1 \sin 2\pi 10^3 t \, V.$$  \hspace{1cm} (3.13)

We determine the maximum voltage error from the maximum rate of change, or slope, which occurs at the zero crossover point for the sine wave shown in Fig. 3.11. Differentiating (3.13) yields

$$\frac{dv_c(t)}{dt} = \omega V_c \cos \omega t = 2\pi f \cos \omega t = \frac{\Delta V_c}{\Delta t}.$$  \hspace{1cm} (3.14)
FIGURE 3.11: Maximum slope occurs at the zero crossover point.

The maximum slope occurs at time $t = 0$ s:

$$\frac{dv(t)}{dt} \bigg|_{t=0} = \omega V_c = 2\pi f V_c = \frac{\Delta V_c}{\Delta t} \Rightarrow \Delta V_c = 2\pi f V_c \Delta t.$$  \hspace{1cm} (3.15)

Thus, the maximum voltage error is $\Delta V_c = 2\pi f V_c \Delta t$, where

$$2\Delta t = T_{\text{clock}} = \frac{1}{f_{\text{clock}}} \Rightarrow \Delta t = \frac{1}{2f_{\text{clock}}} = \frac{1}{2 \times 8000 \times 10^3} = 62.5 \times 10^{-6} \text{ s.}$$  \hspace{1cm} (3.16)

Substituting this value into (3.15) yields the maximum voltage error as

$$\Delta V_c = 2\pi \times 10^3 \times 62.5 \times 10^{-6} = 392 \times 10^{-3} \text{ V.}$$  \hspace{1cm} (3.17)

Draw the sample and hold schematic shown in Fig. 3.12. A reconstruction filter is included and has the same cut-off frequency as the input LPF in the sampler circuit considered previously.

The recovered signals in the right panel in Fig. 3.13 show the effect of not buffering the hold circuit.

The sample and hold pulse is the weighted inverted delayed unit step subtracted from a unit step also weighted by $V$:

$$b(t) = V u(t) - V u(t - T).$$  \hspace{1cm} (3.18)
FIGURE 3.12: Sample and hold and recovery schematic.

FIGURE 3.13: Sample and hold time signals.

From the inverse Laplace Transform tables, we get an expression in time for the signals in the charging period:

\[ F(s) = \int_0^T e^{-st} \cdot t \cdot e^{-st} \, dt = F(s^2 + 1) = \frac{-V}{s} \left[ e^{-st} \right]_0^T = V \left[ 1 - e^{-\frac{T}{s}} \right]. \]  

The sampling and hold signals are shown in Fig. 3.14 but note the charging and discharging periods.

A correct choice for the hold circuit components must be chosen; otherwise we get distortion in the final output signal.
3.4 QUANTIZATION NOISE

Quantization noise, \( q \), is an unwanted product produced in the conversion from analog signals into a digital format. Analog to digital converters (ADC) produce this noise and it is reduced by increasing the number of levels, \( M \), in the ADC process. The signal to noise ratio (SNR) in analog circuits is defined as the ratio of the signal amplitude to the noise amplitude present. In sampled systems, however, the signal to quantization noise ratio (SQNR) is the ratio of signal power to quantizing noise power. The easiest way to examine SQNR is to apply a sinusoidal signal \( V \cos \omega t \) that occupies the full range of the quantizer. The full-scale value is \( V_{FS} = 2^n q/2 \), where \( n \) is the number of bits used. The average signal power for full-scale voltage \( V_{FS} \) is

\[
P_{\text{max}} = \frac{V_{\text{rms}}}{R} = \frac{(V_{FS}/\sqrt{2})^2}{R} = \frac{(q M/\sqrt{2})^2}{R} = \frac{q^2 (2^n/2 \sqrt{2})^2}{8 R} = \frac{q^2 2^{2n}}{8 R}.
\]

(3.20)

The quantizing noise voltage is limited to \( \pm q/2 \) and, except at the signal peak level, the voltage has all values within this range occurring with equal probability. The averaged normalized noise power is determined by integrating the quantization noise voltage (saw-toothed in shape and with a negative slope \(-qt/T_0\) over a period \(T_0\)):

\[
N_q = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} \left[ -\frac{q}{T_0} t \right]^2 dt = \frac{q^2}{3 T_0^3} \left[ \frac{T_0^3}{8} + \frac{T_0^3}{8} \right] = \frac{q^2}{12}.
\]

(3.21)
The ratio of the signal power to noise power \( SQNR \) for a signal that occupies the full scale is

\[
SQNR = 10 \log_{10} \frac{P_{\text{max}}}{P_N} = 10 \log_{10} \left[ \frac{2^n q^2 / 8}{q^2 / 12} \right] = 10 \log_{10} \left[ \frac{3}{2} 2^n \right] = 10 \log_{10} \left[ \frac{3}{2} M^2 \right] \quad (3.22)
\]

\[
SQNR = 10 \log_{10} 2^{2n} + 10 \log_{10} 1.5 = 20n \log_{10} 2 + 10 \log_{10} 1.5 = 6.02n + 1.77 \text{ dB}.
\]

However, the SQNR decreases when the input sinusoid is below the full scale of the ADC. Signal power is a function of waveform shape but noise power is independent of the signal shape. The signal power for a sinusoid with amplitude \( \pm A \) is \( A^2 / 2 \) and \( A^2 \) for a square wave and \( A^2 / 3 \) for a triangular wave.

### 3.5 ANALOG TO DIGITAL CONVERSION

An 8-bit analog to digital converter (ADC) and digital to analog conversion (DAC) is shown in Fig. 3.15. The low-pass filter on the output reconstructs the original input signal. To measure the quantization noise we use a notch filter to remove the desired signal leaving the quantization noise. The 8-bit ADC binary output, set to the nearest integer, in given by

\[
\frac{V_{\text{in}}}{V_{\text{ref}}} 2^{\text{bits}}.
\]

The digital to analog converter (DAC) output produces a rough version of the original input signal. A low-pass filter attenuates the high-frequency components leaving the original signal. Consider the input and reference voltages for an 8-bit system. The \( V_{\text{SIN}} \) part parameters are as follows: \( V_{\text{OFF}} = 10 \text{ V} \) and \( V_{\text{AMPL}} = 10 \text{ V} \), giving a peak input of \( V_{\text{in}} = 20 \text{ V} \). We set the reference to 256 V in order to read the output in binary directly. The binary equivalent for the ADC reference voltage, \( V_{\text{REF}} \), is equal to 256 V and is \((20 \times 2^8 / 256)\) equal to 10100. From the PROBE plot read the most significant bit to the least significant as b4 b3 b2 b1 b0.

![Figure 3.15: PCM coder and decoder system.](image-url)
FIGURE 3.16: 20 V input voltage.

The signal in Fig. 3.16 shows the input 10 V offset raising the 10 V AC input. The binary signal is read vertically downward using the cursor located at the maximum input signal of 20 V.

3.5.1 DAC Resolution

The DAC output is calculated as

\[
V_{\text{out}} = V_{\text{ref}} \left[ \frac{D_7}{2} + \frac{D_6}{4} + \frac{D_5}{8} + \frac{D_4}{16} + \frac{D_3}{32} + \frac{D_2}{64} + \frac{D_1}{128} + \frac{D_0}{256} \right] V. \quad (3.25)
\]

The output voltage for a binary DAC output 00001001 and 256 V reference voltage is

\[
V_{\text{out}} = V_{\text{ref}} \left[ \frac{0}{2} + \frac{0}{4} + \frac{0}{8} + \frac{1}{16} + \frac{0}{32} + \frac{0}{64} + \frac{1}{128} + \frac{1}{256} \right] = 256 \left( \frac{1}{32} + \frac{1}{256} \right) = 9 \text{ } V. \quad (3.26)
\]

The DigClock conversion clock has on/off times equal to 20 µs. Set the transient parameters as follows: Output File Options/Print values in the output file = 1ms, Run to time = 20ms, Start saving data after = 2ms. Making the Output File Options/Print values in the output file larger than the 20 ns default value will speed up the simulation time in most instances. However, it must be set to a value smaller than the Run to time. If all eight bits are used, i.e., full scale, then we will have 256 levels available \(2^8 = 256\). Table 3.1 shows the relationships between the ADC resolution and the voltage reference Vref.
TABLE 3.1: The ADC and Reference Relationship

<table>
<thead>
<tr>
<th>VIN (V)</th>
<th>VREF (V)</th>
<th>NO. OF BITS</th>
<th>NO. OF LEVELS</th>
<th>RESOLUTION (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>16</td>
<td>8</td>
<td>$2^8 = 256$</td>
<td>0.0625</td>
</tr>
<tr>
<td>9</td>
<td>32</td>
<td>7</td>
<td>$2^7 = 128$</td>
<td>0.125 V</td>
</tr>
<tr>
<td>9</td>
<td>64</td>
<td>6</td>
<td>$2^6 = 64$</td>
<td>0.25</td>
</tr>
<tr>
<td>9</td>
<td>128</td>
<td>5</td>
<td>$2^5 = 32$</td>
<td>0.5</td>
</tr>
<tr>
<td>9</td>
<td>256</td>
<td>4</td>
<td>$2^4 = 16$</td>
<td>1</td>
</tr>
</tbody>
</table>

The input signal VSIN generator parameters are VOFF = 5 V, VAMPL = 4 V, and FREQUENCY = 1 kHz. To measure the quantization noise, we must attenuate the 1 kHz input signal using a band-stop notch filter.

3.5.2 Band-Stop Filter
The band-stop (notch) filter is implemented using an ABM part called BANDREJ. The filter is characterized by six parameters: four cut-off frequencies and two attenuation values—RIPPLE defines the maximum allowable ripple in the passband and STOP is the minimum attenuation in the stopband. Set the Analysis tab to Analysis type: AC Sweep/Noise, AC Sweep Type to Linear/Logarithmic, Start Frequency = 100, End Frequency = 10 k, Total Points/Decade = 10001. Enter the lowest frequency F0 to highest frequency F3 for the band-stop filter parameters as shown in Fig. 3.17. A value for either of the passband frequencies, say $f_{p1}$ (or stop frequencies), is calculated by assuming a value for one of the passband frequencies and knowing the center frequency to yield for $f_{p2}$

$$f_0 = \sqrt{(f_{p1}f_{p2})} = \sqrt{(f_{p1}f_{p2})} \Rightarrow f_{p2} = f_0^2 / f_{p1}.$$  \hfill (3.27)

FIGURE 3.17: The band-stop part called BANDREJ.
The band-stop filter response is shown in Fig. 3.18.

We are now in a position to investigate the relationship between the SQNR and the number of bits used. Delete all traces in the PROBE screen and insert a trace, which is the RMS of the output signal from the band-stop filter. Measure the noise RMS value and repeat for a smaller number of bits, by deleting the lines b0 to b7 from the input of the DAC and in turn measuring the SQNR. Alternatively, you may place switches on each bit line, and then open each one in succession.

3.6 PULSE CODE MODULATION

It is not economical to send signals in parallel form over a telephone cable network, so the output from the analog to digital converter (ADC) is multiplexed into a serial bit stream. The multiplexing and demultiplexing process in Fig. 3.19 is an important part of the 8-bit 30-channel TDM PCM system and requires 8-bit parallel to serial conversion. There are many multiplexing IC devices but here we use the 74151 IC where the IC pins S0, S1, and S2, control the sequence of conversion. The STIM4 generator part has four parallel output lines, of which one is unused but is terminated with a line segment name called unused. Command line 1 starts the sequence as 0ms and the parallel bit pattern is 0000. The four parallel signals repeat a number of times using a repeat function on command line 2 and the sequence is terminated on command line 4 with endrepeat. Command line 3 shows the pattern being incremented by 1 every 125 µs.

The generator output is terminated with a BUS line using the Place Bus icon (Short-cut key B). Dlclick the bus and type in s[3-0] where each line connected to the bus is named s0,
FIGURE 3.19: Multiplexing and demultiplexing.

s1, s2, and s3. Placing a marker on a bus produces a PROBE display in hexadecimal format. The clock frequency is twice the speed of the STIM4 part increment (i.e. half the time period). In an actual system, the final output s3 could synchronize the system. The serial data is applied using a STIM1 part.

Set the Analysis tab to Analysis type: **Time Domain** (Transient), **Run to time** = 10ms, and **Maximum step size** = (left blank), Press F11 to simulate. The waveforms for the mux–demux devices are shown in Fig. 3.20.

### 3.6.1 Universal Shift Register

74194 LS is a 4-bit bidirectional universal shift register where all data and mode control inputs are edge-triggered and respond to LOW to HIGH clock transitions only. The mode control
TABLE 3.2: 74194 Universal Shift Register

<table>
<thead>
<tr>
<th>OPERATING MODE</th>
<th>CLR (1)</th>
<th>S1 (10)</th>
<th>S0 (9)</th>
<th>SR (2)</th>
<th>SL (7)</th>
<th>PN (3–6)</th>
<th>QA (15)</th>
<th>QB (14)</th>
<th>QC (13)</th>
<th>QD (12)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reset</td>
<td>L</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td></td>
</tr>
<tr>
<td>Hold</td>
<td>H</td>
<td>L</td>
<td>L</td>
<td>X</td>
<td>X</td>
<td>Q0</td>
<td>Q1</td>
<td>Q2</td>
<td>Q3</td>
<td></td>
</tr>
<tr>
<td>Shift left</td>
<td>H</td>
<td>H</td>
<td>L</td>
<td>X</td>
<td>L</td>
<td>Q1</td>
<td>Q2</td>
<td>Q3</td>
<td>L</td>
<td></td>
</tr>
<tr>
<td>Shift right</td>
<td>H</td>
<td>L</td>
<td>H</td>
<td>X</td>
<td>X</td>
<td>Q1</td>
<td>Q2</td>
<td>Q3</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>Parallel load</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>X</td>
<td>X</td>
<td>Pn</td>
<td>P0</td>
<td>P1</td>
<td>P2</td>
<td>P3</td>
</tr>
</tbody>
</table>

and selected data inputs must be stable one setup time prior to the positive transition of the clock pulse. The four parallel data inputs, P0, P1, P2, P3, are D-type inputs (also called A, B, C, and D). When both S0 and S1 are HIGH, signals presented to P0, P1, P2, and P3 inputs are transferred to Q0, Q1, Q2, and Q3 outputs, respectively, following the next LOW to HIGH transition of the clock. This is a parallel-to-parallel conversion. The mode control inputs, S0 and S1, determine the synchronous operation of the device. Table 3.2 shows how data is shifted from left to right (Q0 → Q1), or right to left (Q3 → Q2). Parallel data is entered, by loading all four bits of the register simultaneously.

When both S0 and S1 are LOW, the existing data is retained in a “do-nothing” mode without restricting the HIGH to LOW clock transition. D-type serial data inputs (SR, SL) allow multistage shift right or shift left data transfers without interfering with parallel load operation, and will appear on the QA and QC outputs. The asynchronous clear (CLR), when LOW, overrides all other input conditions and forces the Q outputs LOW.

In the table, L = low level, H = high level, X = do not care, l = LOW voltage level one setup time prior to the LOW to HIGH clock transition; h = high voltage level one setup time prior to the LOW to HIGH clock transition; P(n) and Q(n), where n indicates the state of the input or output one setup time prior to the LOW to HIGH clock transition.

3.6.2 74194 Universal Shift Register
Draw the schematic in Fig. 3.21. What is the output for the parallel input 0110? Set the Analysis tab to Analysis type: Time Domain (Transient), Run to time = 10ms, and Maximum step size = (left blank), press F11 to simulate.
FIGURE 3.21: Universal shift register.

FIGURE 3.22: Universal shift register signals.

3.7 SINGLE-CHANNEL 4-BIT PCM TRANSMITTER

The simple PCM system in Fig. 3.23 uses only four of the 8 bits available, but it may be extended to an 8-bit capability using additional IC circuitry (see exercise at the end of the chapter). Here, we apply a 1 kHz sinusoidal signal and sample it at 8000 Hz, which is the rate used in the public switched telephone network (PSTN) and converts the ADC parallel output to a serial bit stream. To test that the circuit is working correctly, we need to reverse the procedure and attach a serial to parallel converter, a latch for temporary store, and a DAC.

The DAC output is then low-pass filtered. The registers must be initialized to a certain state. Select the Simulation Setting menu and then select the Options tab. In the Category box select Gate-Level Simulation. This will then show the Timing Mode - select Typical. In the Timing Mode change Digital Setup from Typical to Maximum. The Initialize all flip flops is set to All 1. Failing to do this will result in the flip-flop being in an undetermined state, i.e., All X, and shows up in the plot as two red lines. Set the Analysis tab to Analysis type: Time Domain (Transient), Run to time = 2ms, and Maximum step size = (left blank), Press F11 to simulate.
FIGURE 3.23: Single-channel 4-bit PCM transmitter and receiver.
3.8 TIME-DIVISION MULTIPLEXING AND DEMULTIPLEXING

In the European E1 PCM telephone systems, 30 voice signals are sampled, coded, and multiplexed, to produce a 2.048 Mb/s digital signal that is transmitted over a high-bandwidth line, coaxial cable, microwave line-of-sight link, optical fiber, etc. Sample binary values are multiplexed using byte interleaving so every 125 µs a framing bit is followed by a coded sample for each of the 30 voice signals. Other housekeeping bits are allocated for different purposes. Also super frames take care of the signaling requirements by 16 frames with the center time slot used for two channel signaling of 4 bits each. The transmission rate is 257 bits every 125 µs (2.048 Mb/s = 8000 × 8 × 32). The U.S. and Japanese 24-channel time-division multiplexed (TDM) PCM system generates a composite bit stream for a total of 24 digitized voice channels and results in a frame structure consisting of 193 bits in each 125 µs time interval. Secondary multiplexed systems combine primary multiplexing digital streams into a higher rate output bit stream called the plesiochronous digital hierarchical structure. A TDM transmitter and receiver, using ABM library parts, are shown in Fig. 3.25. Set the Analysis tab to Analysis type: Time Domain (Transient), Run to time = 4 ms, and Maximum step size = (left blank), Press F11 to simulate.

Fig. 3.26 shows the TDM and recovered signals.

3.8.1 Time-division Multiplexing of Two PAM Signals

The 4066 IC in Fig. 3.27 multiplexes two signals. One of the electronic switches may be used (see exercise 12 at the end of the chapter) to invert the clock so that the second signal is sampled with a delay of one clock period.