Pulse Code Modulation

TECHNIQUES

with Applications in COMMUNICATIONS and DATA RECORDING

BILL WAGGENER

Pulse Code Modulation Techniques

with Applications in Communications and Data Recording

Bill Waggener

A Solomon Press Book

VAN NOSTRAND REINHOLD
I(T)P™ A Division of International Thomson Publishing Inc.

New York • Albany • Bonn • Boston • Detroit • London • Madrid • Melbourne
Mexico City • Paris • San Francisco • Singapore • Tokyo • Toronto

Editing and design: Solomon Press

Printed in the United States of America
For more information, contact:

Van Nostrand Reinhold
115 Fifth Avenue
New York, NY 10003

International Thomson Publishing GmbH
Königswinterer Strasse 418
53227 Bonn
Germany

International Thomson Publishing Europe
Berkshire House 168-173
High Holborn
London WCIV 7AA

International Thomson Publishing Asia
221 Henderson Road #05-10
Henderson Building
Singapore 0351

Thomas Nelson Australia
102 Dodds Street
South Melbourne, 3205
Victoria, Australia

International Thomson Publishing Japan
Hirakawacho Kyowa Building, 3F
2-2-1 Hirakawacho
Chiyoda-ku, 102 Tokyo
Japan

Nelson Canada
1120 Birchmount Road
Scarborough, Ontario
Canada M1K 5G4

International Thomson Editores
Campos Eliseos 385, Piso 7
Col. Polanco
11560 Mexico D.F. Mexico

1 2 3 4 5 6 7 8 9 10 HAM 01 00 99 98 97 96 95

Library of Congress Cataloging-in-Publication Data
Waggener, Bill.
Pulse code modulation techniques: with applications in communications and data recording / Bill Waggener.
p. cm.
Includes bibliographical references and index.
ISBN 0-442-01436-8
1. Digital communications. 2. Pulse-code modulation I. Title.
TK5103.7.W34 1995
621.382--dc20
95-16286
CIP

Contents

CHAPTER 1

Introduction to Pulse Code Modulation

The idea of communicating information from one place to another by the presence, or absence, of pulses is quite old with many historical examples. Pulse code modulation (PCM) embodies the basic concepts of transmitting a sequence of symbols, i.e., pulses, to represent information. Examples of early PCM systems are illustrated in Figure 1.1. In the telegraph system, for example, messages were transmitted from one station to the other as pulses of current with two widths, "dots" and "dashes," with groups of pulses representing letters of the alphabet.

Pulse code modulation applies two basic concepts; time-division multiplexing (TDM) and amplitude quantization. Continuously varying signals are sampled and quantized into discrete symbols for transmission. At the receiving end, the symbols are recovered and the original signal is recovered. Early theoretical work by Nyquist[1,2] established the principles of sampling while the work of Shannon[3] and others[4,5] laid the foundation for communication theory which revealed the advantages of PCM communication.

The practical beginning of PCM technology can probably be dated back to the late 1940's and work performed at Bell Laboratories on PCM telephony systems.[6,7,8] Since then, PCM has been applied to a variety of applications. This book is directed toward the underlying technologies of PCM including symbol encoding, symbol detection, symbol synchronization, format synchronization, modulation, demodulation, error detection and correction. Applications are taken from telecommunications, telemetry and magnetic recording.

A variety of applications such as telecommunications, telemetry and data storage share a common PCM technology. A number of data communi-

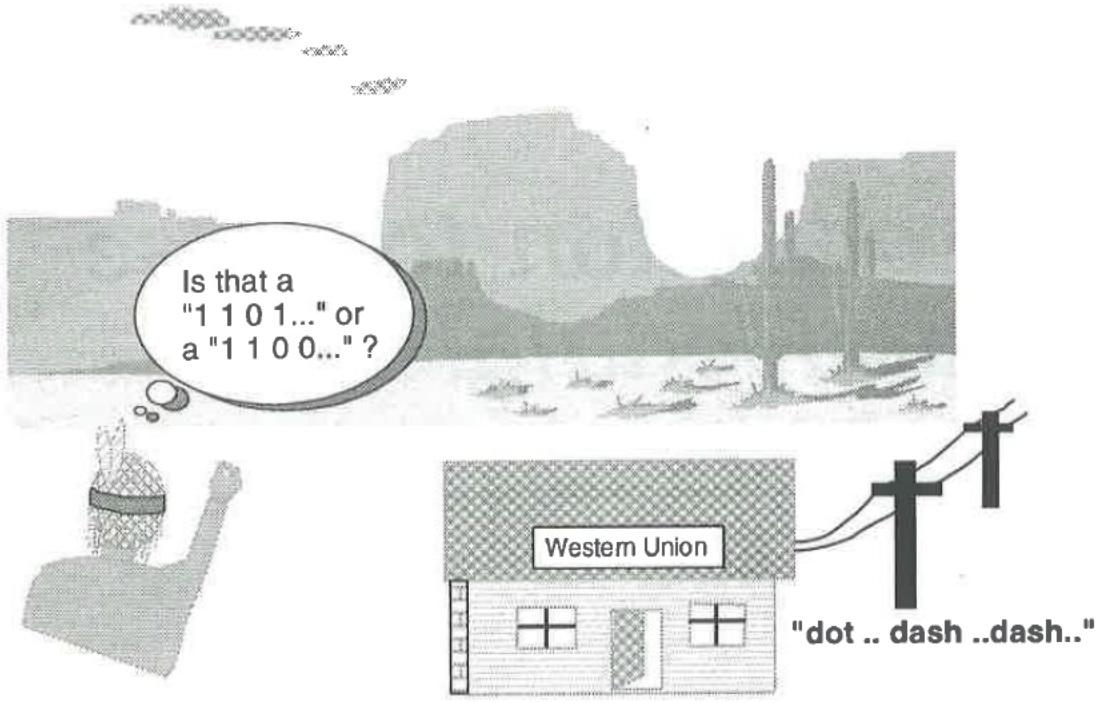


FIGURE 1.1 Early Pulse CodeModulation systems

cations applications share both the common PCM technology and communication media as illustrated in Figure 1.2. Telemetry applications, for example, tend to be a one-way communication between many data sources and one data sink. Telecommunications, on the other hand, deals with two-way communications between many sources and many sinks. When looked at in this fashion, it becomes obvious that one major distinction between the telemetry and telecommunication applications is an emphasis on switching technology in a telecommunication system. Indeed, many telecommunications organiza-

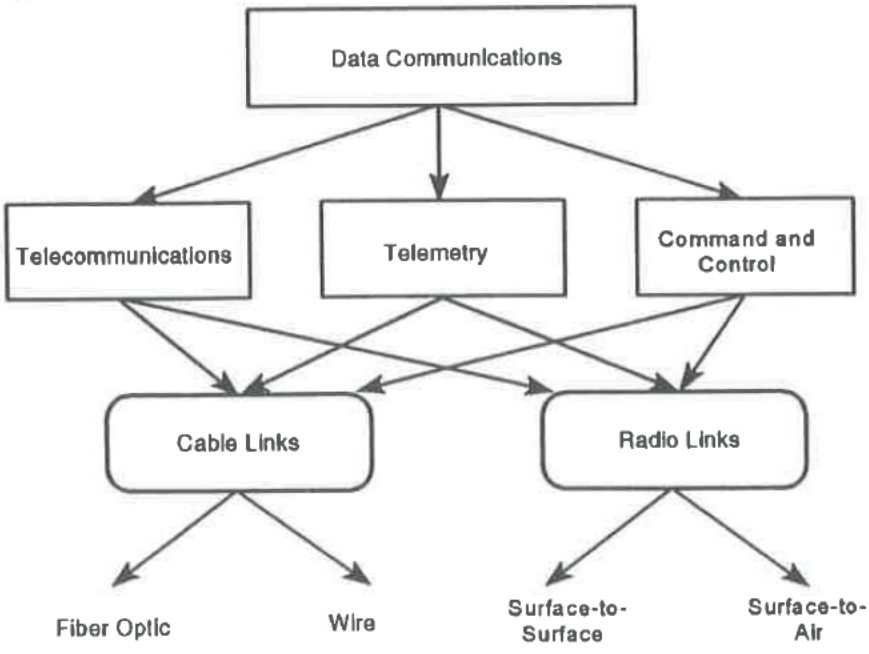


FIGURE 1.2 The data communicatons hierarchy

tions and technical references divide telecommunications into transmission and switching.

In the telemetry system, although there may be many users of the data, the data is usually combined and gathered at one location. Although switching and data distribution is implicit in most telemetry systems, the switching aspects are not a major focus of the system.

The transmission aspects of the telecommunication and telemetry systems clearly overlap using common communication technologies. The emergence of PCM as the dominant communication technology brings the two applications even closer as techniques developed in the larger telecommunications market find application in the more specialized telemetry market.

Within the last decade, data storage applications have used PCM technology for recording data on both magnetic and optical media. Indeed, data recording applications have been fertile areas for new developments in PCM symbol encoding and the use of complex error correction techniques.

The trend toward the use of PCM is pervasive, being driven by the rapidly advancing semiconductor technology and the principles of efficient communications. There are many facets to PCM communication and the emphasis placed in this book are on those aspects which are prime determinants of overall system performance. As a consequence of this philosophy, a considerable amount of attention is paid to the problems of synchronizing and detecting PCM signals in the presence of noise and interference. At the same time, more pragmatic issues must also be addressed such as the efficient use of allocated bandwidth and ease of implementation.

A measure of the technology trend in PCM systems is shown in Figure 1.3, showing the composite data rate of a variety of PCM systems over the past several decades. The trend line for PCM telemetry systems parallels the trend of PCM telecommunication systems increasing approximately an order of magnitude per decade. The ever increasing rates are driven by increasing numbers of data channels in both the telemetry and telecommunication areas as well as increasing sampling rates for wider bandwidth signals. This trend

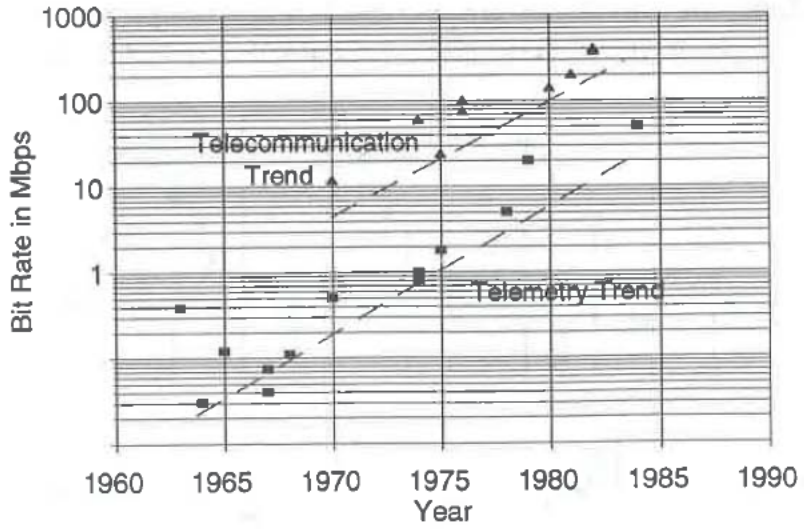


Figure 1.3 PCM Data Rate Trend.

is expected to continue unabated. Limitations on communication bandwidth are placing severe constraints on the ability to achieve the higher rates and are driving the PCM technology to more efficient signaling methods as well as toward fiber optic and wideband satellite communication channels.

It is impossible to cover all of the aspects of PCM in a single book without being superficial. I have elected to concentrate on key areas associated with the generation, transmission, synchronization and detection of PCM. At the risk of offending the mathematical purists, I use a "mathematics of modest rigor" approach, preferring to leave detailed proofs and developments to more specialized texts. Many excellent texts[9,10,11,12] devoted to statistical communications and specialized topics are available for the reader wishing to pursue in depth more advanced topics.

NYQUIST TO SHANNON

Many of the key aspects to a PCM system can be identified by looking at a simplistic example shown in Figure 1.4. A signal source producing a continuous time waveform is sampled and each sample quantized into a discrete number of levels by an analog-to-digital converter (ADC). Typically, the number of levels is a power of 2 and a given sample can be represented by a set of binary digits, i.e., "bits," such that:

$$L = 2^m \qquad [1.1]$$

where m = the number of bits and L = the number of levels in the sample.

Each sample can be considered to be a "word" and the individual bits in a word are sequentially transmitted from the data source via a transmission channel to the data sink, or user. The user identifies the bits in each word, reconstructs the word, and, if desired, converts the digital word back to analog form using a digital-to-analog converter (DAC).

In the telecommunications world, the data source could be the output of a caller's telephone and the data sink could be the telephone of the person being called or it could be one computer on a local area network (LAN) send-

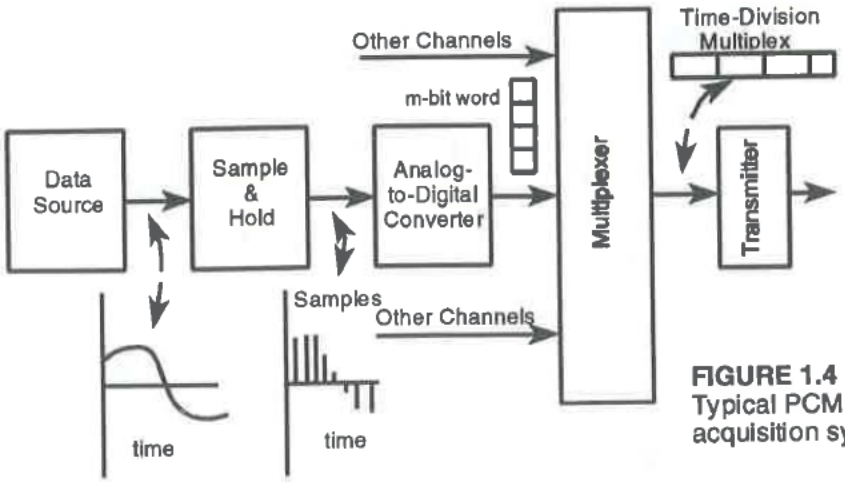


FIGURE 1.4
Typical PCM data acquisition system

ing files to another computer on another network. In the telemetry world, the data source could be the output of a strain gage and the data sink could be a strip chart display. In either case, once the signal has been sampled, other signals could be interleaved and transmitted with the original signal provided the transmission bandwidth will allow it. This is called multiplexing and the interleaving of samples in time leads to the notion of a time-division multiplexed (TDM) system.

Sampling is obviously a key feature of the PCM system. What sampling rate must be used for a given signal? The sampling theorem answers that question; the sampling rate must be at least twice the highest frequency of the input signal. If, in our example, the signal is bandlimited to 4 kilohertz, a minimum sampling rate of 8 kilosamples per second would be required. This is typical of a voice channel in a telecommunication channel. If the same signal sample is quantized to 8 bits ($2^8 = 128$ levels), the PCM bit rate would be 64 kilobits per second (8 bits per sample times 8 kilosamples per second).

Bits per Second, Symbols per Second and Bauds

Information rate is normally expressed in "bits per second" where a bit represents a binary choice, i.e., the information is either a "one" or a "zero." When the binary information is transmitted over a communication channel, one, or more, bits are encoded into "symbols" which are transmitted and the transmission rate is expressed in terms of "symbols per second." The unit "baud" is used interchangeably with symbol rate. It is incorrect to refer to "baud rate" since baud is defined as a measure of rate. Note that symbol rate is only equal to bit rate when only one bit is represented by a symbol. It is not uncommon for a transmission symbol to represent several bits. For example, a 4800 bit per second data modem typically transmits at 1200 baud, or 1200 symbols per second, each symbol representing 4 bits of information.

The quantization of the sample clearly limits the resolution of each sample to 2^{-m} of the maximum sample amplitude. While this is a consideration in the design of the overall system, this book will assume the data sources are encoded bit streams without regard for the individual data sampling rates or quantizations.

In order to appreciate some of the rationale for the use of PCM, compare the direct transmission of the signal in the presence of noise with transmission using PCM. Suppose the quantization of a signal sample is chosen to be comparable to the transmission channel signal-to- noise ratio so that

$$SNR \approx 2^m \qquad [1.2]$$

If the signal is sampled at the minimum rate, the PCM bit rate is then

$$R_b = 2mB \qquad [1.3]$$

where B = signal bandwidth in Hertz, m = number of quantization bits, and R_b = PCM bit rate in bits per second.

Intuitively, the PCM transmission bandwidth must be about $2m$ greater than the signal bandwidth in order to transmit the PCM bits with minimum distortion. Noise increases with the square root of the bandwidth so that the PCM signal-to-noise ratio (SNR) would be less than the direct transmission SNR by a factor of $\sqrt{(2m)}$. However, the PCM signal is composed only of binary pulses and the receiver need only decide if a "one" or a "zero" was transmitted. The SNR needed for a reliable decision is relatively small, typically about 4:1 is sufficient. If 8 bit quantization were used, a direct transmission channel SNR of greater than 100:1 would be required to maintain the 8 bit signal precision. Using PCM, a bandwidth increase of about 16:1 would be required, increasing the transmission noise by about 4:1, however, a channel SNR of only 4:1 or 5:1 would be required. Thus, the PCM transmission would require about a 20:1 SNR defined in a bandwidth equal to the signal bandwidth for the PCM system compared to over 100:1 SNR for direct transmission. Thus PCM has exchanged bandwidth for signal-to-noise ratio, a fundamental principle established by Shannon.

The sampling theorem establishes the minimum permissible rate required to sample a bandlimited signal. Conversely, Nyquist established the maximum signaling rate required for a bandlimited channel. Nyquist proves that signaling at a rate of twice the bandwidth of a bandlimited channel is possible without intersymbol interference (that is, energy from adjacent symbols overlapping). Nyquist further proved that the channel does not have to have "brickwall" bandlimiting to be free from intersymbol interference. Nyquist signaling will be an important concept in the analysis and design of the PCM system.

In 1959, C.E. Shannon[13] published a key development in communications theory in which he proposed a theoretical bound for communication over a noisy analog channel. Shannon developed the channel capacity bound:

$$C = W \log_2(1 + SNR) \qquad [1.4]$$

where C = the channel capacity in bits per second, W = the channel bandwidth in Hertz, and SNR = the channel signal-to-noise ratio.

Shannon's bound is very interesting because it asserts that there exist coding schemes which will allow the channel capacity to be reached with **arbitrarily small error rate**. Further, the theorem suggests that capacity can be traded for signal-to-noise ratio. Shannon's channel capacity theorem and the implicit trade-off between bandwidth and required signal-to-noise ratio will provide a valuable performance bound for comparing the performance of PCM systems. In Chapter 2 we will take a detailed look at Shannon's results and the implications to PCM systems.

Signal-to-Noise Ratio

Signal-to-noise ratio, or SNR, may be defined in a number of ways. Typically, SNR is defined as ratio of signal power to noise power for a given bandwidth, W. The noise power depends on the type of noise and

how it is measured. If the noise is "white" (that is, has a uniform distribution of power with frequency) the noise power is:

$$P_N = N_0 W$$

where N_0 = the noise spectral density, a constant, and W = the equivalent noise bandwidth.

If the equivalent noise bandwidth is expressed in terms of the symbol rate, SNR is frequently expressed in terms of the symbol energy-to-noise spectral density:

$$SNR = \frac{E_s}{N_0 k}$$

with, k = the ratio of the noise equivalent bandwidth to the symbol rate.

THE PCM LINK MODEL

A generalized PCM system can be represented as shown in Figure 1.5. An information source produces a sequence of information bits which are then encoded into channel symbols. The channel symbols, in turn, modulate a transmitter. The transmitted signal is propagated over a communication channel which may distort the signal and add noise. The receiver must recover the channel symbol sequence from the modulated signal. In order to recover the symbols, timing information must be extracted (synchronization) and symbol decisions (detection) made on the noisy signal. Finally, the recovered symbols are mapped back into information bits. In the practical PCM system, information sources may be multiplexed into a composite PCM stream. The composite data stream may have a complex format which must be demultiplexed at the receiving end.

A major effort has been expended over the last ten years, or so, defining

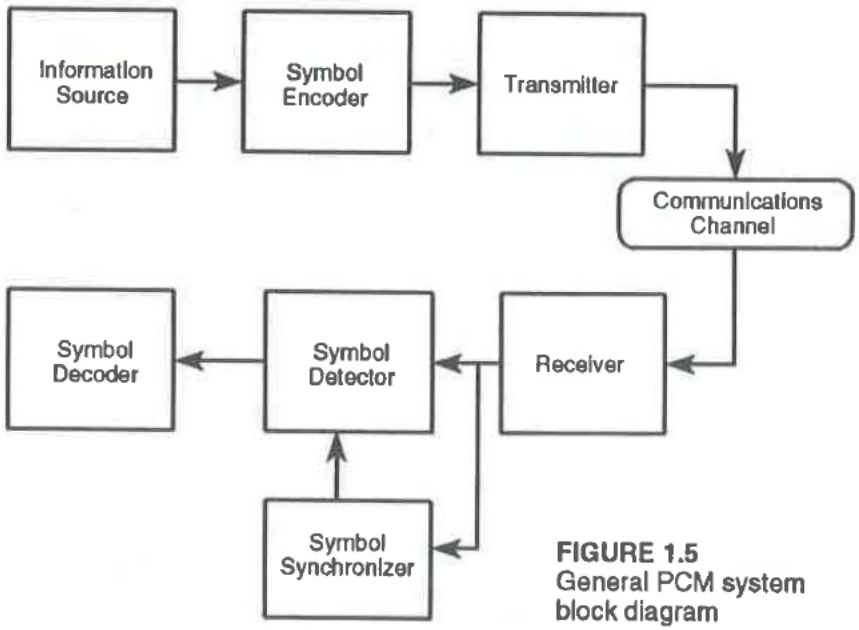


FIGURE 1.5
General PCM system block diagram

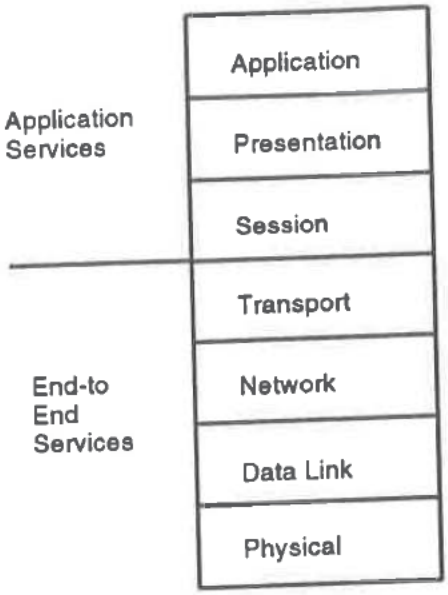


FIGURE 1.6
OSI reference model

communication protocol standards for data links and, specifically, local area networks. The Open Network Interconnect (OSI) standard describes communication systems in terms of a seven layer protocol "stack" as shown in Figure 1.6. The bottom of the stack is the physical layer (PHY) which is responsible for the electrical (or optical) interface to the communications media. The next higher layer in the stack, the Data Link Layer (DLL), is responsible for the transmission, framing (formatting) and error control. The Network Layer (NL) is responsible for network data transfer independent of the type of media and the network topology. The Transport Layer (TL) is responsible for reliability and multiplexing of data transfer over the network. These four layers are normally provided by the data service provider. The three layers above these are typically associated with the applications.

In OSI, the services provided by a layer are either connection-oriented or connection-less. A connection-oriented service establishes a connection between source and destination using a calling method. Connection-less services have only one mode, data transfer. Routing of data from source to destination must take place within the data transfer itself. Many data services are based on the OSI model although considerable liberty is taken with regard to the number, names and functions of the protocol stack. One view of the protocol stack is that it provides a logical "block diagram" of the system. Each layer sends and receives information from the layers above and below it. This book deals with the lowest levels in the data communications hierarchy - the physical layer which deals with the intimate details of the PCM signaling and reconstruction and, to a limited extent, the Data Link Layer which deals with the format and framing of data.

COMMUNICATION CHANNELS

PCM systems use a variety of communication channels ranging from radio systems to fiber optic cable systems. Each communication media has

it's own set of unique characteristics and the PCM system must be tailored to the link characteristics. Radio systems tend to be limited by noise, interference and multipath propagation distortion and signal-to-noise efficient communication techniques are essential. Systems using cable communications are typically limited by insufficient bandwidth and require techniques which maximize signaling efficiency. The recording media in a PCM data storage system may be regarded as a specialized communications channel requiring unique PCM symbol encoding methods.

Regardless of the type of media, the communications channel can, normally, be represented by a combination of an equivalent filter with an additive noise source as shown in Figure 1.7. In some cases, this simplistic model is inadequate and a more complex model must include nonlinear effects and multiplicative noise sources, as well as the additive, noise sources. For the purposes of this book, the simpler model will suffice. Communication system simulation programs[14,15] are now available which permit the simulation of more complex link models, if required.

Radio links are used in both telemetry and telecommunication systems. In many practical applications, radio telemetry links are used between a ground station and an airborne vehicle operating at significant ranges from the receiving site. Under these conditions, radio energy is reflected from the ground as well as traveling by a direct path causing a form of distortion known as "multipath." Transmission power is frequently limited and radio telemetry links operate with small communication link margins. Detection and synchronization performance is crucial for these applications.

Telecommunication systems use radio links both for ground microwave links and for satellite communication (SATCOM) systems. Although greater system signal-to-noise ratios are attainable in these systems through the use of higher signal power and more conservative design practices, fading and interfering signals are also cause for concern in the areas of detection and synchronization.

The use of wire cable for communications is, probably, still the major means of communications for telecommunication networks. Fiber optic cables are replacing both wire cable and microwave systems and will likely become the primary communication link for the telecommunication networks in the 1990s and beyond.

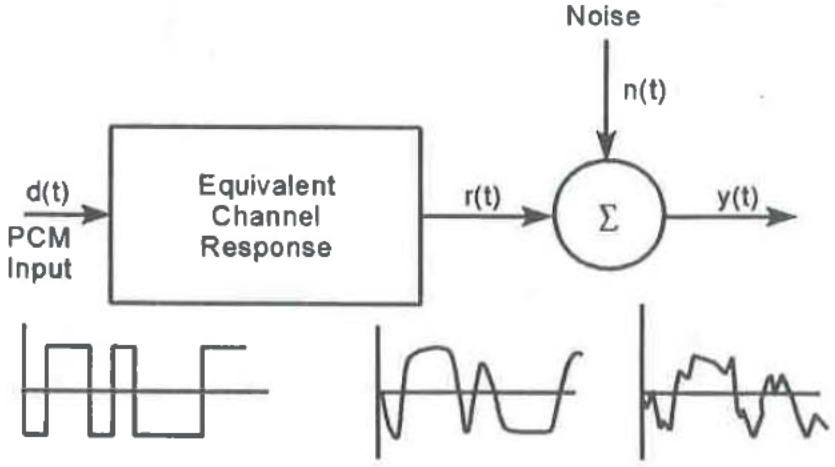


FIGURE 1.7
Communication channel model

The primary limitation of cables, either wire or fiber optic, is the signal distortion created by the transmission line behavior in which signals of differing frequencies are propagated with slightly differing velocities. This effect can be, at least partially, compensated by equalization networks. The distortion introduced by a cable communication system causes energy from one symbol to be spread over adjacent symbols causing "intersymbol interference." The intersymbol interference degrades system performance by reducing the symbol decision margin in the PCM receiver. Equalization techniques which reduce, or eliminate, intersymbol interference also tend to accentuate system noise, thereby making the system more sensitive to noise. Once again, detection and synchronization are key system functions. Since the characteristics of the typical cable communication channel change with time and network connection, equalization techniques should be adaptive to gain the greatest performance benefit. Although equalization is extremely important, the subject is complex and is left to several excellent books[16,17] which are devoted entirely to the subject. In this book, the effects of the equalizer on time (or frequency) domain performance and on equivalent noise are included in the equivalent communication link model.

AN OVERVIEW OF THE PCM SYSTEM

The typical PCM system is shown in Figure 1.8. The system begins with a data acquisition subsystem which acquires data from one, or more, sensors, samples, multiplexes and digitizes the signals. The digital data may then be multiplexed with other digital data and synchronization markers are embedded in the composite digital signal to permit proper data recovery. The digital data stream is serialized and fed to a symbol encoder which replaces one or

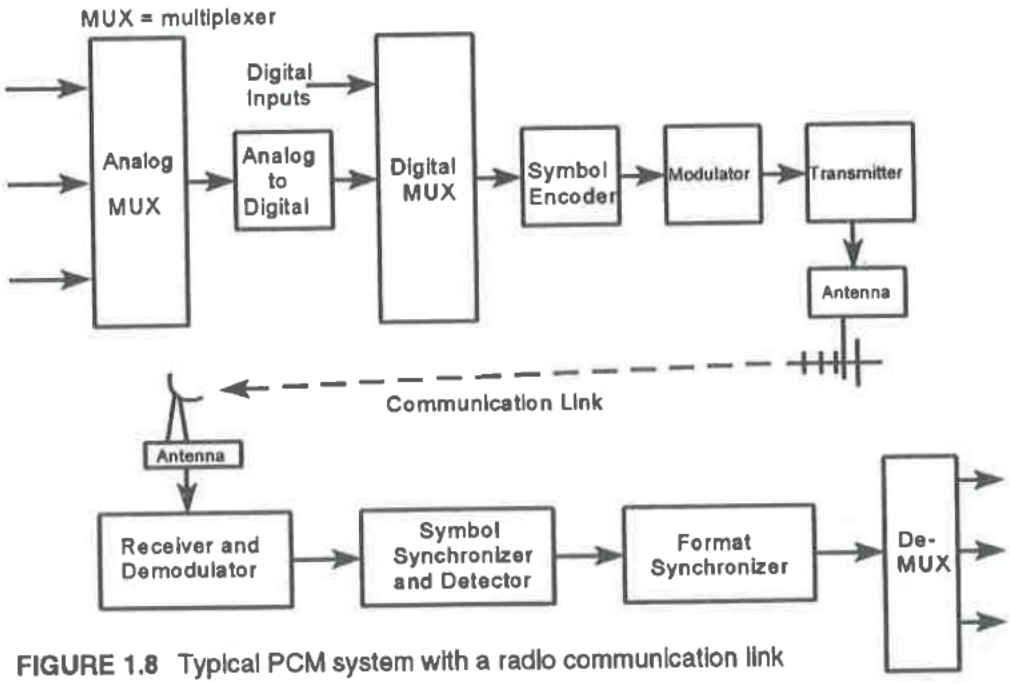


FIGURE 1.8 Typical PCM system with a radio communication link

more data stream bits with a symbol for transmission. Depending on the type of channel, the symbol stream may also modulate a carrier to translate the signal to a different frequency band. The baseband or modulated signal is then transmitted over the communication channel.

At the receiving end, the signal must be demodulated to recover the serial stream of symbols. The symbols must be detected to recover the original data bits and appropriate timing clocks must be extracted, both for optimum symbol detection and for format synchronization. The format synchronization process includes locating and synchronizing the embedded synchronization markers and establishing the position of the data words in the bit stream. Decommutation, or demultiplexing, is then required to split the data stream back into individual data channels corresponding to each individual sensor.

In the telecommunications system, the individual sensors may correspond to individual telephones. At the lowest level, 24 voice channels are digitized and multiplexed to form a single PCM data stream. The digital voice channels are multiplexed with data streams from other data sources. In addition to digitized voice, the telecommunications network now provides a variety of data services ranging from X.25 packet data to broadband multimedia information over ATM services. In the telemetry system, a wide variety of sensors may be used to measure everything from temperature to stress. The composite serial PCM data stream may contain hundreds or thousands of data channels.

In addition to the basic functions described, the PCM system may include error correcting coding and data encryption. As technology permits, the digitization of sensor data is moving toward sensors with digital outputs so that all multiplexing is performed digitally.

Data Acquisition

The PCM system begins with the acquisition of data which can take many forms. Traditionally, the telecommunication system handles voice and low rate data sources, although video and higher rate data are increasingly important. The sampling rates for data sources in the telecommunication system are typically limited to a relatively small range. Voice channels are sampled at 8 kilosamples per second with a nominal quantization of 8 bits resulting in each voice channel requiring 64 kilobits per second. Voice compression techniques reduce the rate per channel but the composite multiplex rate is usually kept constant, increasing the number of channels per multiplex. Similarly data in the telecommunication network is acquired in multiples of 75 baud ranging up to 56 kilobaud.

The telemetry system acquires data from a wide variety of sensors and sources ranging from individual sensors to data from aircraft buses. Whereas the telemetry instrumentation engineer, at one time had complete control over the sensor sampling rates, the PCM telemetry system must now contend with asynchronous data from on-board data buses and computers.

In general, the data acquisition process consists of signal conditioning electronics, analog and digital multiplexers, and the analog-to-digital con-

verter (ADC). The signal conditioning electronics are responsible for converting the wide variety of sensor and transducer outputs to a common electrical level. Analog signals are multiplexed and digitized using the ADC and combined with other digital signals to form the composite PCM signal.

The data acquisition subsystem is generally the domain of the instrumentation engineer who worries about things such as signal conditioning drift and noise, aliasing error and quantization errors. While these are all worth worrying about, this book will leave those topics to others and begin our story of PCM with the multiplexer.

Multiplexing, Formatting and Encoding

The PCM story really begins to get interesting when the individual data streams are multiplexed. While the multiplexing of the input signals may appear to be quite straight forward on the system block diagram, in fact, the multiplexing process can be quite complex. In multiplexing and sampling of the data, the sampling rate per sensor must be at least twice the highest signal frequency (Nyquist's theorem) in order to prevent the distortion known as aliasing. If the sensors all have an equal bandwidth, such as the voice channels in a telephone system, the multiplexing scheme is nearly trivial. On the other hand, if a system has a number of different sensors with widely varying bandwidth, it would be inefficient to sample all channels at twice the frequency of the highest bandwidth sensor. A great many channels could be highly oversampled with a composite data rate much higher than required. The multiplexing and sampling plan under these circumstances should attempt to minimize the composite data rate while satisfying the minimum sampling rate requirements for each channel. This approach leads to subcommutation and supercommutation techniques in the multiplexer design. The addition of asynchronous data sources further complicates the formatting design requiring data buffering and the insertion of fill words or patterns into the composite data stream.

Once a PCM data format is generated, the data is normally converted to a bit serial data stream. At this point, error correcting coding may be implemented to control transmission errors introduced in noisy communication channels and the encoded data is fed to the channel encoder. The role of the channel encoder is to generate the symbols used for transmission. As we will see in Chapter 5, the choice of the channel symbols can be dependent upon a number of factors including channel bandwidth and type of channel noise and interference. The PCM data recording system enters the picture, taking a composite PCM data stream and encoding it for recording on some media such as magnetic tape, magnetic disks or an optical media.

Modulation/Demodulation

PCM systems using radio communication links or communication channels with "bandpass" characteristics require the symbols data to be placed on carrier signals. The composite PCM stream modulates a carrier and the receiving system must, in turn, demodulate and recover the composite PCM

stream from the carrier. The modulation/demodulation technique used depends on many factors and has a significant impact on the overall system performance. Amplitude, frequency and phase modulation techniques are all used in PCM systems. The term "modem" is a contraction of **MOD**ulation/**DEM**odulation and several references discuss various aspects of modem design.[12,18].

The modulation type used is frequently dictated by the bandwidth of the communication channel. Both in the United States and internationally, governmental regulatory agencies limit the bandwidth which can be occupied by a given radio channel. Although two level frequency modulation (also called frequency shift keying, or FSK) and two level phase modulation (also called phase shift keying, or PSK) have been the most commonly used techniques in radio systems for a number of years, because of increasing bandwidth restrictions and increasing data rates more complex modulations are becoming attractive.

A chart illustrating the radio frequency (RF) spectrum bands used for telemetry and telecommunications is shown in Figure 1.9. The trend is to move toward the upper right hand corner of this chart as data rates increase and available bandwidth becomes limited. The telemetry spectrum allocation has moved from P-band (215 MHz) to L-band (1450 MHz) and S-band (2250 Mhz). Applications requiring higher data rates will undoubtedly move to even higher bands in the future.

PCM systems using cable communications also employ modulated carrier techniques in some applications and the restricted cable bandwidth requires even more complex modulation techniques to transmit the maximum

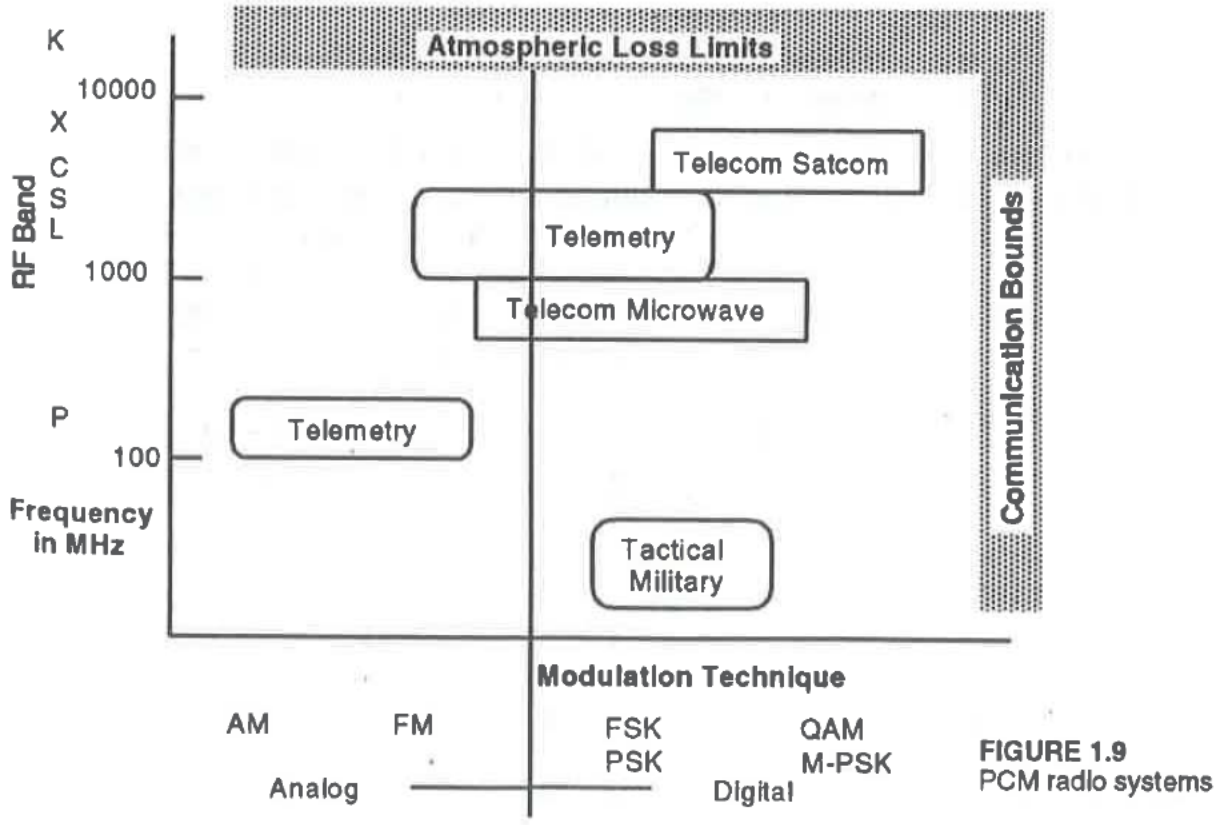


FIGURE 1.9
PCM radio systems

data rates possible. Quadrature modulation schemes such as quadrature PSK (QPSK) and quadrature amplitude modulation (QAM) are increasingly popular. In this book, the effects of the modulation/demodulation process will be included in a lowpass equivalent model of the communications process.

Detection and Synchronization

Once the composite PCM signal is recovered from a transmitted carrier, or is received directly, the digital data symbols must be reconstructed. In order to determine the value of individual symbols, the timing of the data stream must be established. Specifically, the time at which each symbol occurs must be determined. In some signaling techniques, the timing information is carried directly in the signal and can be readily recovered. When this method is used, a certain amount of signal power is devoted to the timing component and the error performance of the reconstructed data stream suffers accordingly. More commonly, the timing information is implicit in the data stream and no explicit timing signal is required. While it is more difficult to extract the timing information from these signals, no power is wasted on the timing information and the best possible error performance can be achieved.

Once symbol timing is established, the timing clock can be used to sample the input symbols to make a symbol decision. By appropriately filtering the signal to reject noise, maximum signal-to- noise can be achieved at the sampling time in order to minimize symbol decision errors. The symbol timing and detection circuits have traditionally been implemented in a functional unit known as the "bit synchronizer." The same functions are included in the digital data modem but are integrated with the demodulation function and are not separately identifiable.

Format Synchronization and Demultiplexing

Once symbol synchronization and detection has been accomplished, format synchronization must be established. Format synchronization implies the ability to locate individual data words within the data stream. The data stream format is created in the multiplexing and sampling plan and can be as simple as the 24 channel PCM voice multiplex used in the telephone industry or the extremely complex formats used in telemetry test programs.

The data format includes some form of imbedded symbol patterns (or markers) which identify the beginning (or end) of a data "frame." These markers (or frame sync words) are designed to be as distinct from the data sequences as possible. The format synchronizer includes a "frame synchronizer" which is designed to recognize and synchronize on the frame sync words in a data stream containing distributed errors. In addition, the format synchronizer includes functions to identify other substructures within the data format such as subcommutated or supercommutated data.

Once format synchronization is established, the data may be demultiplexed (sometimes referred to as decommutation) and distributed to individual data outputs. Once again, the complexity of the format affects the

demultiplexing task. PCM voice multiplexes used in telecommunication systems are designed to simplify the demultiplexing task using uniformly sampled data channels with distributed synchronization markers. Telemetry formats are typically complex with channels of varying word lengths, formats and sampling rates.

Error Detection and Correction

The quality of the communication link used in PCM systems can range from excellent to poor depending on the application. Even normally excellent links can occasionally degrade due to atmospheric effects or degradation of link components. As data rates increase, communication link signal-to-noise margins decrease and error rates increase. These factors lead to an increasing use of error correcting coding.

Three types of error correcting codes are in common use:

- Block codes
- Convolutional codes
- Concatenated codes

The basic principle used in all error correcting coding is to add redundancy to the data which can be used to correct transmission errors. Adding redundancy increases the composite PCM data rate but the code performance offsets the decrease in signal-to-noise ratio and improves overall performance.

Block codes add redundancy (or parity) symbols to a block of data symbols to form the encoded block of symbols. The simplest example of a block code is a repetition code which repeats the value of a data bit one or more times. Block coding is normally incorporated in the basic data format so that the PCM format synchronizer is used to establish block synchronization for the decoder.

Convolutional codes interleave data symbols and parity symbols to form the composite PCM stream and the convolutional decoder is inserted between the symbol synchronizer and the format synchronizer. Concatenated codes combine either a block code with a convolutional code or two block codes to form a more complex code structure. Concatenated codes are particularly well suited to the data recorder applications.

When error correcting coding is employed, the detection and synchronization performance of the PCM system is of particular concern since operation is in a high error rate environment. The error correction can only perform properly if the system can acquire and maintain synchronization.

KEY ISSUES IN PCM

PCM systems have been widely used since the 1970s and the technology is maturing. The demand for higher rates and performance near theoretical limits continues to provide challenging tasks for PCM designers. High perfor-

mance, very large scale integrated (VLSI) components provide the means to implement more features and functions in smaller packages.

Some of the key issues facing designers and users of PCM systems are outlined in the following paragraphs:

- Radio frequency management

  As data rates increase, wider bandwidth channel allocations will be required or more bandwidth efficient modulation techniques employed. Very wideband channels (> 100 MHz) can only be obtained in the 10 GHz and higher RF bands. Operation in these bands and at high bit rates poses challenging design problems for the PCM engineer. More bandwidth efficient modulation techniques require more sophisticated signal processing.

- Fiber optic communication links

  Fiber optic communication links are rapidly replacing wire cables as data rates continue to increase. The extremely wide bandwidth of fiber optic cable permit very high bit rates taking PCM designs into the realm of GaAs logic. Signaling and detection design techniques are affected by the characteristics of the incoherent optical transmission.

- Communication networks

  PCM systems are becoming more network oriented even for telemetry applications which have been point-to-point systems. The network aspects of the PCM system places greater emphasis on multiplexing and demultiplexing and the use of packet communications. Packet PCM systems place a premium on rapid symbol synchronization and error control.

- Complex formats

  Complex data formats require greater attention to the problems of acquiring and maintaining format synchronization. Embedded formats can change on a frame-to-frame basis, again focusing on the problems of rapid acquisition and maintenance of synchronization words.

- Error control and data compression

  The need for error correcting coding is expected to be increasingly important in PCM systems. Data compression as a means of reducing link bandwidth requirements has been studied and occasionally implemented, since the early 1960s. In the past there has been a widespread reluctance to use data compression for fear of losing important information or providing inadequate data quality. Channel bandwidth limitations as well as the use of information preserving compression techniques may well change the prevailing attitudes and greater transmission of compressed data is expected in the future. Although this book does not address data compression, it is important to recognize that error correction requirements in a PCM system may be driven by a need to minimize transmission errors because of the error sensitivity of compressed data.

- Digital signal processing

  Digital signal processing (DSP) is becoming a pervasive technology, sig-

nificantly changing the way in which many signal processing tasks are performed. There are many areas in PCM systems where DSP technology is replacing traditional analog designs. The use of DSP can be overdone, however, and some processing tasks will continue to be implemented in good old ASP (analog signal processing).

Many of these key issues are considered in this book. In the case of digital signal processing, the topics are directly addressed. In the other cases, the topics are addressed indirectly through discussions of modulation, demodulation, synchronization and detection.

REFERENCES

1. Nyquist, H. "Certain Factors Affecting Telegraph Speed" *Bell System Technical Journal* 3(1924):324–346.
2. Nyquist, H."Certain Topics in Telegraph Transmission" *AIEE Transactions* 47(1928): 617–644.
3. Shannon, C.E."A Mathematical Theory of Communication" *Bell System Technical Journal* 27(1948):July and October.
4. Hartley, R.V.L."Transmission of Information" *Bell System Technical Journal* 7(1928): 535–563.
5. Rice, S.O."Communication in the Presence of Noise - Probability of Error for Two Encoding Schemes" *Bell System Technical Journal* 29(1950): January.
6. Goodall, W.M. "Telephony By Pulse Code Modulation" *Bell System Technical Journal* 26(1947): July.
7. Meacham, L.A. and Peterson, E. "An Experimental Multichannel Pulse Code Modulation System of Toll Quality" *Bell System Technical Journal* 27(1948): January.
8. Oliver, B.M., Pierce, J.R. and Shannon, C.E. "The Philosophy of PCM" *Proceedings of the I.R.E.*(1948): November.
9. Helstrom, C. *Statistical Theory of Signal Detection*. New York: Pergamon Press, 1968.
10. Van Trees, H. L. *Detection, Estimation, and Modulation Theory*. New York: John Wiley & Sons, 1968.
11. Viterbi, A. J. *Principles of Coherent Communication*. New York: McGraw-Hill, 1966.
12. Lee, E.A, and Messerschmidt, D.G. *Digital Communication, Second Edition*. Boston: Kluwer Academic Publishers, 1994. Lee and Messerschmidt's book is one of the best references to be published in many years and is highly recommended.
13. Shannon, C.E. "Communication in the Presence of Noise" *Proceedings of the I.R.E.* 47(1959).
14. Saver, C. H. and MacNair, E.A. *Simulation of Computer Communications Systems*. New York: Prentice-Hall, 1983.
15. Frost, V.L., Shanmugan, K.S., and Tranter, W.H. "Computer-aided Analysis and Design of Communication Systems" Notes from Tutorial at MILCOM 86.Boston, MA., 1986 The MILCOM tutorial reviewed the following simulation programs; SYSTID, LINK, CHAMP, COMSIM, ICS, ICSSM, and BOSS.
16. Alexander, S.T. *Adaptive Signal Processing, Theory and Applications*. New York: Springer-Verlag, New York, 1986.
17. Honig, M.L. and Messerschmitt, D.G. *Adaptive Filters, Structures, Algorithms and Applications*. Boston: Kluwer Academic Publishers, 1984.
18. Proakis, J.G. *Digital Communications, Second Edition* New York: McGraw-Hill Book Co., 1989

CHAPTER 4

Multiplexing and Formatting

There seems to be a certain naturalness about discussing systems from the source of information to its final destination. This "source to sink" approach in a PCM system leads initially to the topics of multiplexing, formatting and encoding. The source of the digital data for the system may be digitized voice signals, sensor outputs, or arbitrary sequences of binary data. The PCM system, almost always, has a multiple number of data sources. In some cases, such as the telecommunication system, the inputs all have the same rate. In other more general systems, the input data sources may have a variety of rates.

The communication channel has a limited bandwidth, and according to Shannon, has a fixed channel capacity in bits per second. The PCM system must attempt to utilize the channel capacity as efficiently as possible. With multiple data sources, there are several ways of communicating the data over the channel, by frequency division multiplex (FDM), by time division multiplex (TDM), or a combination of these two methods. In FDM, individual data sources are separated by placing the signals on separate modulated carrier signals. The individual carriers may be amplitude, frequency or phase modulated. Each data source requires a modulator and demodulator. In TDM, the data sources are combined into one composite signal by placing data from the individual sources into different time "slots" and transmitting the composite signal with one modulator/demodulator set. This combining operation is commonly referred to as "multiplexing" with the inverse operation termed "demultiplexing."

The costs of the individual modulator/demodulator equipment required for FDM must be weighed against the TDM multiplexing/demultiplexing costs. In this book, the focus will be on TDM multiplexing and demulti-

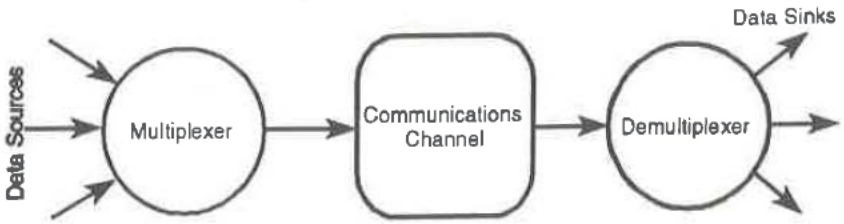


FIGURE 4.1
Time division multiplexing for PCM systems

plexing. TDM multiplexing will be used as a generic term to include a variety of multiplexing techniques. The multiplexer, as shown in Figure 4.1, combines a group of individual data sources into a composite signal for transmission over the communication channel. At the receiving end, the demultiplexer separates the data back into individual channels.

The process of multiplexing involves a formatting of the data into the composite signal such that the receiver can recover the individual data. When the characteristics of the data sources are identical, the formatting can be relatively simple. On the other hand, the multiplexing and formatting of a group of dissimilar data sources can be complex. In this chapter, the multiplexing and formatting problems are addressed. The problem of recovering the data from the multiplexed signal is considered in Chapter 7. The designer of the multiplexer must consider the function of demultiplexing otherwise the recovery of the data can be a serious problem.

Multiplexing and formatting are topics of concern at the link layer of the open systems protocol stack as well as the physical layer. Much of this book deals with the physical layer of the communications system and is concerned with synchronizing and detecting PCM signals. The problems of format synchronization (that is, identifying the boundaries of words and frames) require a knowledge of the data format. The data format is, in turn, intimately tied to the link layer protocol processing. Even though the format processing is often hidden from the user by an integrated circuit (such as an HDLC controller) understanding the multiplexing and format synchronization problems are important for the PCM designer. In telemetry systems, the format processing is generally exposed in the form of a module devoted to frame synchronization and decommutation. In telecommunications systems, these functions are embedded in the multiplexer/demultiplexer. An understanding of the format is crucial to the format synchronization problem which is discussed in Chapter 7.

MULTIPLEXING

There are a number of different classes of multiplexers which can be characterized as shown in Figure 4.2. The first level division classifies the multiplex as either synchronous or asynchronous. Synchronous is defined here as all data source clocks being related to a master clock, in both frequency and phase. In other words, the clock of any given data source can be derived from a common master clock. Asynchronous, on the other hand, means that the individual data clocks are independent of one another.

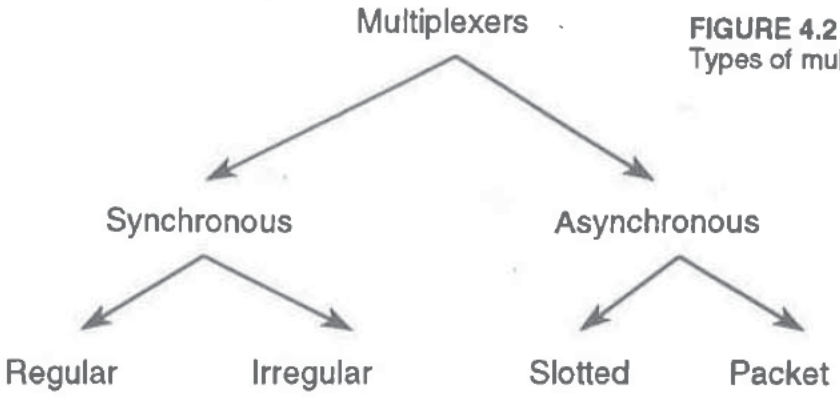


FIGURE 4.2
Types of multiplexers

Synchronous Multiplexes

The class of synchronous multiplexes is further subdivided into regular or irregular multiplexes. Regular is used (for want of a better word) to describe a multiplex in which all of the data sources are identical. A good example of a regular multiplex is the telecommunication DS-1 channel bank which multiplexes 24 voice channels, each of which is an 8-bit word, sampled at 8000 samples per second. The regular multiplex can be constructed using a data structure as illustrated in Figure 4.3. Each channel is assigned one time slot in a "frame." A synchronization marker (frequently referred to as a "sync pattern") is inserted in the frame to allow the receiver to identify each frame. The frame rate is then equal to the sampling rate for each channel. The synchronization marker can be assigned to specific time slots or distributed over the entire frame.

The composite output data rate of the regular multiplexer is

$$R_{output} = (N_c N_b + N_f) F_{rate} \qquad [4.1]$$

where N_c = the number of channels, N_b = the number of bits per channel, N_f = the number of sync marker bits, and F_{rate} = the frame rate in frames per second.

Apart from the overhead devoted to the synchronization marker, this multiplex has the maximum efficiency in terms of the ratio of the composite data rate compared to the total multiplex rate.

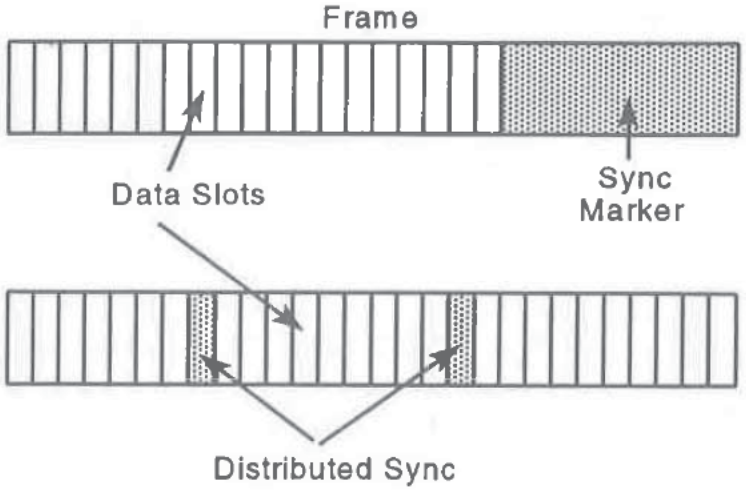


FIGURE 4.3
Common framing formats

T-1 Multiplex Frame

The standard North American T-1 multiplex combines 24, 8-bit channels into one frame with one additional bit assigned to frame synchronization. The basic frame rate is 8000 frames per second so that the composite T-1 multiplex rate is $(24 \cdot 8 + 1)\ 8000$, or 1.544 megabits per second.

The irregular multiplex includes the rest of the synchronous multiplex structures not considered regular. This includes most of the multiplexes encountered in data acquisition and telemetry systems. An irregular multiplex contains a mixture of data sources with differing rates. For example, an application may require a multiplex with the following mixture of data sources:

20, 8-bit channels sampled at 2000 samples per second,
40, 8-bit channels sampled at 200 samples per second,
10, 8-bit channels sampled at 10 samples per second.

How does one structure a multiplex so that the composite data rate is as low as possible? In the example, the minimum possible composite rate is 384800 bits per second. We could take a brute force approach and sample all channels at the highest rate of 2000 samples per second and create a regular multiplex with 70, 8-bit channels in a frame with a frame rate of 2000 frames per second. With this approach, the composite data rate (ignoring the synchronization marker overhead) is 1.12 megabits per second, almost three times the minimum rate. With this approach 50 of the 70 channels are grossly oversampled, wasting precious bandwidth.

Subcommutation

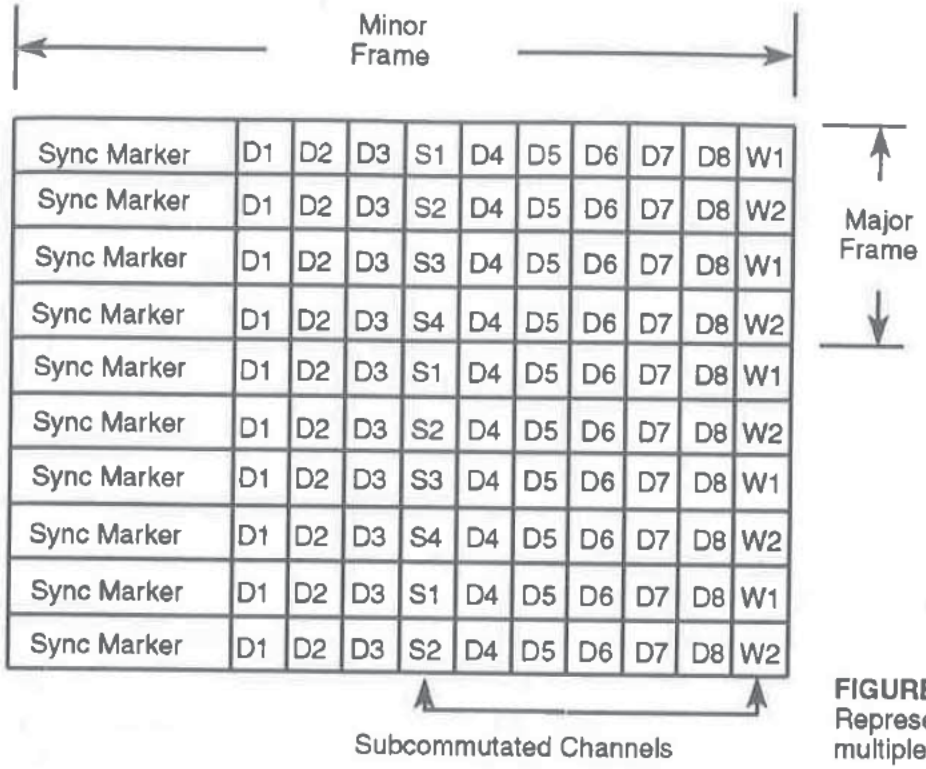
In the example given, many of the channels need to be sampled at only a fraction of the highest rate channels. This suggests that the multiplex be structured so that some of the time slots in the frame are shared by multiple channels. A new representation of the multiplex is needed. One possible representation of the multiplex as an array is shown in Figure 4.4. New definitions are required.

Minor frame: The length, in bits, of the number of columns in the array.

Major frame: The number of bits between samples of the lowest rate channel.

Given this array-like representation, the highest sampled channels can be assigned to time slots in the "minor" frame. Channels with lower sampling rates can be multiplexed into additional minor frame time slots. This submultiplexing is commonly called subcommutation.

Returning to the example multiplex, the 40 channels sampled at 200 samples per second can be multiplexed into four minor frame time slots which repeat every 10 minor frames as illustrated by the subcommutated channels labeled s1 to s40 in Figure 4.5. Thus a channel in one of these time slots is sampled at a 200 sample per second rate if the minor frame is sampled



Sync Marker	D1	D2	D3	S1	D4	D5	D6	D7	D8	W1
Sync Marker	D1	D2	D3	S2	D4	D5	D6	D7	D8	W2
Sync Marker	D1	D2	D3	S3	D4	D5	D6	D7	D8	W1
Sync Marker	D1	D2	D3	S4	D4	D5	D6	D7	D8	W2
Sync Marker	D1	D2	D3	S1	D4	D5	D6	D7	D8	W1
Sync Marker	D1	D2	D3	S2	D4	D5	D6	D7	D8	W2
Sync Marker	D1	D2	D3	S3	D4	D5	D6	D7	D8	W1
Sync Marker	D1	D2	D3	S4	D4	D5	D6	D7	D8	W2
Sync Marker	D1	D2	D3	S1	D4	D5	D6	D7	D8	W1
Sync Marker	D1	D2	D3	S2	D4	D5	D6	D7	D8	W2

FIGURE 4.4
Representing a multiplex as an array

at 2000 samples per second. What do we do with the 10 channels sampled at 10 samples per second? In a more general case, we could consider another level of subcommutation (cleverly called "sub-subcommutation") multiplexing the lower rate channels in a time slot associated with a subcommutation channel. In this case, let's just assign another minor frame time slot to the 10 channels and oversample them at 200 samples per second, like the

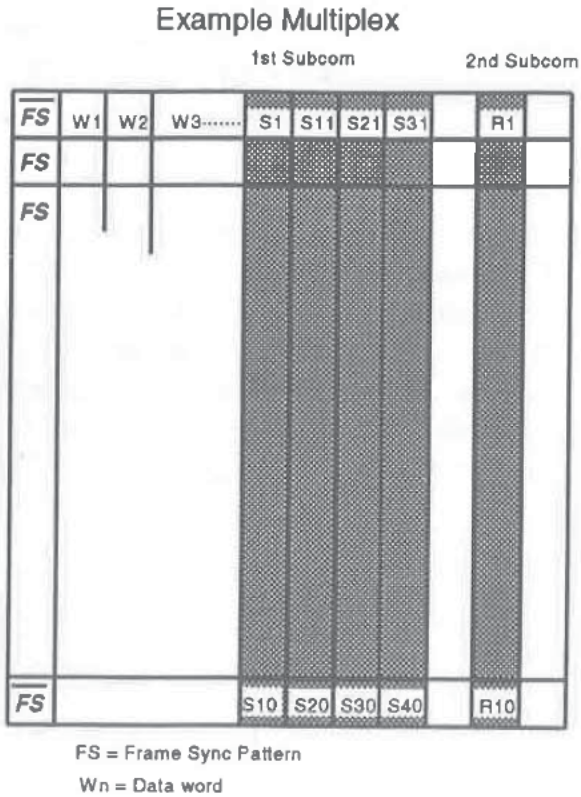


FIGURE 4.5
An example multiplex with subcommutated channels

other subcommutated channels. With this change in multiplex structure, what is the composite data rate, ignoring the sync marker overhead? We now have 25 minor frame channels, sampled at 2000 samples per second giving a composite data rate of 400000 bits per second, very close to the minimum achievable. The price we have paid is an increase in complexity of the multiplex which will have to be reflected in the design of the demultiplexer.

The deepest level of subcommutation defines the major frame length. *The major frame length is, in essence, the size of the array required to represent the subcommutation channels before the sampling sequence is repeated*. In the example, 10 minor frames are required to sample all of the multiplex channels so that the major frame size is 10 times the minor frame length.

Supercommutation

What do we do if a few channels are added to our example multiplex which require a sampling rate higher than the minor frame rate of 2000 frames per second? If lower sampling rates were accommodated with subcommutation, then the higher rate channels are included by "supercommutation." Supercommutation is accomplished by "strapping" several minor frame time slots together, assigning them to a single data channel. This is illustrated in Figure 4.6. Successive time samples of the higher rate data source are put into several minor frame time slots, effectively multiplying the minor frame sampling rate by the number of supercommutation time slots assigned to the channel.

Notice a difference between subcommutated channels and supercommutated channels. The subcommutated channels are uniformly sampled while the supercommutated channels are only uniformly sampled if the time slots can be uniformly distributed over the major frame. This may or may not be a problem. In many cases, the channel data can be buffered on both the input and output to hide the nonuniform sampling from the user.

Designing Irregular Multiplex Structures

By using subcommutation and supercommutation, complex multiplexes can be designed. In real life, very complex multiplexes can occur. Unlike our simple example, real applications can have channels with differing rates, word sizes and, even, changing numbers of channels with time. It is not unusual in telemetry applications to have within the same multiplex thousands of individual channels with differing rates and word sizes ranging from 4 bits to 32 bits. With the many combinations and permutations of

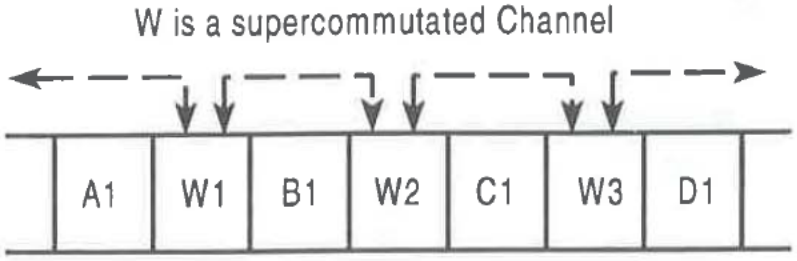


FIGURE 4.6
A multiplex with a supercommutated channel

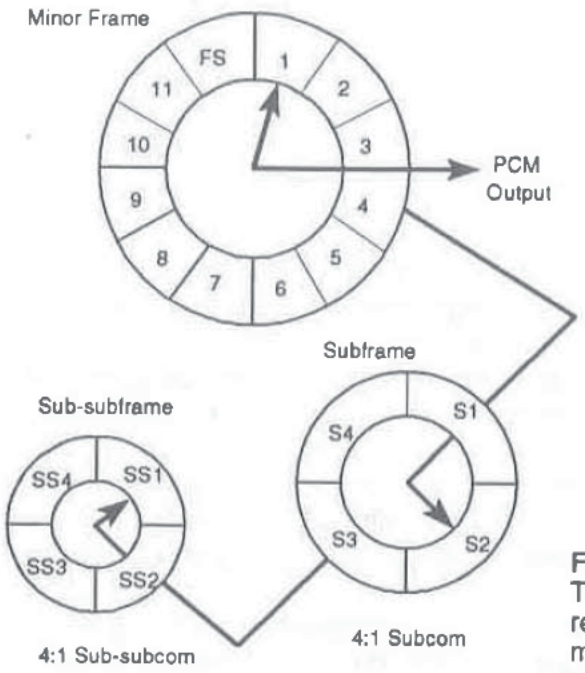


FIGURE 4.7
The wheel diagram representation of a multiplexed format

possible formats, it is not surprising that no algorithmic methods have been developed for designing complex multiplexes.

The array representation of the multiplex gives some insight into ad hoc design methods. An alternative representation of the multiplex structure, commonly known as a "wheel" diagram, may also be helpful in visualizing the multiplex. The wheel diagram for a multiplex with one supercommutated channel, one subcommutated channel and one subcommutated channel is shown in Figure 4.7. If we think of the wheels as gears, it is clear that an integer relationship is required between the minor frame rate and the subframe rates. Subframe words can be "strapped" to generate a large number of sampling rate ratios, however, all of the ratios must be rational fractions. It is extremely important for the multiplex designer to refrain from being exceptionally clever in forming the multiplex structure without considering the problems associated with decommutation of the data.

From the standpoint of the decommutation problem, there are several design guidelines which should be observed.

- Avoid more than two levels of subcommutation.
- Avoid supercommutation of subframe data.
- It is better to oversample than to overly complicate the multiplex structure.

A Multiplex Design Example

With these guidelines in mind, let's consider an ad hoc design procedure for structuring an irregular multiplex. Assume for this design that all of the data channels have the same word length, or are, at least, packed into a common word length.

Step 1 In a table sort the data source requirements according to sampling rate, highest rate first:

No. of channels	*Required sampling rate*
N_0	f_0
N_1	f_1
•	•
•	•

Step 2 Compute the composite data rate for each set of channels by multiplying the number of channels by the required sampling rate per channel.

Step 3 Select the minor frame rate as the sampling rate of the group of channels with the highest composite rate.

Step 4 Group the remaining channels into two groups with lesser sampling rates and two groups with higher rates (if any). The first group of channels will be candidates for subframes and the next group will be candidates for sub-subframes.

Step 5 Select an integer ratio between the sampling rate of each of these groups so that the group sampling rate is equal to, or greater than, the highest rate of the highest individual sampling rate.

Step 6 Select the group with sampling rate lower than the minor frame rate. Choose a sampling rate for this group that is an integer divisor of the minor frame rate, calling the ratio, R_1. Define a subframe which occupies minor frame slots as

$$N_{sub} = \textit{Integer part of} \left(\frac{N_1}{R_1} + 1 \right) \qquad [4.2]$$

Step 7 Repeat Step 6, for the possible subframe candidates. Repeat the procedure for the possible subsubframe candidates, adding additional subframe and (perhaps) minor frame slots as required.

Step 8 Take the remaining groups of channels and add sufficient minor frame channels to supercommutate these channels.

This procedure, best illustrated by example, is far from foolproof and has ignored the insertion of synchronization markers. The means for modifying the design for synchronization will be apparent once we consider the format synchronization problem. Let's try this method on another contrived example.

Assume the requirement is defined as follows:

2 channels sampled at 10 ksps
20 channels sampled at 2 ksps
5 channels sampled at 750 sps
15 channels sampled at 450 sps
20 channels sampled at 125 sps
10 channels sampled at 10 sps

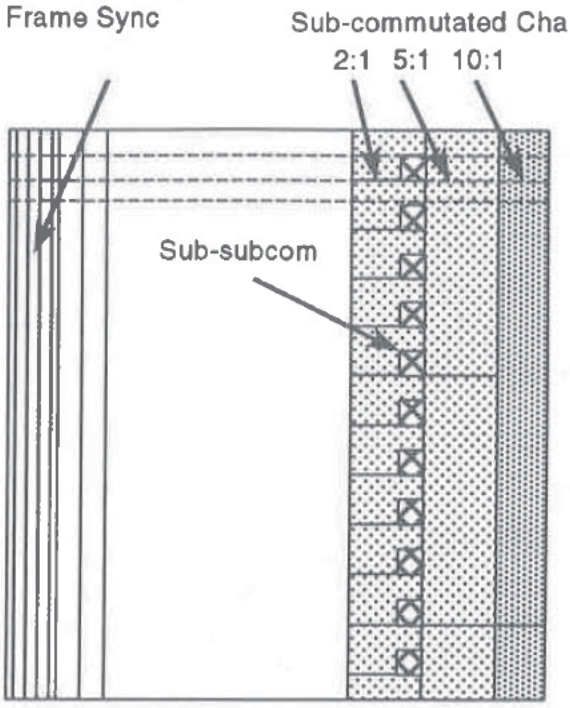


FIGURE 4.8
A multiplex example with 3 subcommutated channels

The multiplex structure designed by this method is shown in Figure 4.8 and can be used to follow the design steps as defined in the example above.

Step 1–2 Form a sorted table with the composite rate.

Channels	*Sampling rate*	*Composite rate*
2	10 ksps	20 ksps
20	2 ksps	40 ksps
5	750 sps	3.75 ksps
15	450 sps	6.75 ksps
20	125 sps	2.5 ksps
10	10 sps	0.1 ksps
Total rate		73.1 ksps

Step 3 Select a minor frame sampling rate. From the table select a minor frame rate of 2 ksps with 20 channels initially place in the minor frame.

Step 4 Group channels. Group the 5, 15 and 20 channels together as a candidates for the subframes and the last 10 channels as a subsubframe candidate.

Step 5–6 Subframe structures. The first 5 channels must be sampled at a minimum of 750 sps. To maintain an integer relationship with the minor frame rate, a sampling rate of 1 ksps (2:1 subcomm) should be used. Add three slots (5 channels divided by a subframe ratio of 2 increased to the next integer value) to the minor frame for these channels. Add the next 15 channels as a subframe with a sampling rate of 500 sps (5:1 subcomm). This requires three more minor frame slots. Sample the last

20 channels chosen as subframe candidates at 200 sps (10:1 subcomm). Two additional minor frame slots will be required for this subframe.

Step 7 Subsubcommutation. Finally, we have the remaining 10 channels as a candidate for a subsubframe. Notice that one slot in the first subframe is unused. This is a logical place for the subsubframe if the sampling rate requirement can be met. This subframe is sampled at 1 ksps. If all 10 channels are placed in this slot, the sub-subframe sampling rate will be 100 sps, well above the required sampling rate.

Step 8 Supercommutation. The remaining two channels are sampled at 10 ksps requiring a 5:1 supercom ratio. Thus, add an additional five minor frame slots for each of these channels to complete the multiplex structure.

How well did we do in structuring an efficient multiplex? The minimum possible rate from our table is 73.1 ksps. The multiplex structure as designed has 38 minor frame channels with a minor frame rate of 2 ksps for a composite rate of 76 ksps. Not bad! Can you do better?

In order to decommutate the multiplex we must know where the minor frame begins (or ends) and where the subframe and subsubframe channels begin. Some form of synchronization markers will have to be incorporated into our multiplex structure which not only will add overhead but may force us to modify the multiplex structure. The design method worked well in this example but may have problems with some requirements, particularly those with large spreads in sampling rate requirements. Unfortunately, there seems to be no easy algorithmic solution to this design problem and the problem seems more amenable to an artificial intelligence program which can reason in a manner similar to the human when the ad hoc method encounters problems.

ASYNCHRONOUS MULTIPLEXES

Synchronous multiplexes combine data channels which have a common clock relationship. Rational relationships exist between individual channels and it is possible to derive all timing from a common clock. When a channel's time slot comes up, the data value for that slot is available. Perhaps there are a few readers who remember a famous comedy routine where several workers remove cakes from a conveyor belt and pack them into boxes for shipment. When the workers are in synchronism with the flow of the cakes all is well, but when the cake flow increases faster than they are able to put them into the boxes . . . well, you get the picture.

In the asynchronous case, the clocks of the individual channels or groups of channels are independent. Even if the individual clock frequencies are nominally the same, small variations in the rate can ultimately cause synchronization problems. There are two general classes of asynchronous multiplex-

ers: slot multiplexers and packet multiplexers. The slot multiplexer is similar to the synchronous multiplexer in that channel values are placed in a time slot in a synchronous TDM format. On the other hand, the packet multiplexers gather up sequential values of a data channel, place them in a finite data format, termed a "packet," and transmit the packets when the communication channel is ready. Individual packets are treated independently, requiring symbol synchronization and packet format synchronization.

Asynchronous Slot Multiplexing

Taking the analogy of the cakes flowing on a conveyor belt with workers removing the cakes and packing them into boxes, there is an intuitive conservation of rate. The average rate at which the cakes are removed from the belt must equal the input cake rate (is the SI unit for cake rate, cps, or what?). If the average removal rate is higher than the input rate, there will be times when the workers are idle, waiting for cakes. If the removal rate is too slow, cakes will begin to fall from the belt before the workers can get them. As long as the average rates are equal, the problem of cakes falling from the belt can be solved by proving a "buffer" table for temporary storage when instantaneous rates are too high. But what happens when the input rate is so low that no cakes are available. As far as the workers are concerned this is great, but what goes in the shipping boxes which are being sent out at the average rate? The asynchronous multiplexer which uses a synchronous TDM output data stream is faced with similar problems.

A simplified block diagram of the asynchronous multiplexer is shown in Figure 4.9. Individual data streams are input to first-in/first-out (FIFO) buffers to smooth instantaneous variations in the input rates. The outputs of the buffers are placed into dedicated time slots in the synchronous output format. The output rate must insure that the average output rate always equals or exceeds the composite input rate. If not, the input data buffers will eventually overflow. The output rate can be compared to the composite input rate and the output clock can be adjusted to insure that the average rates are always equal. With this solution, however, an undesirable side effect is produced, the output data rate continually changes.

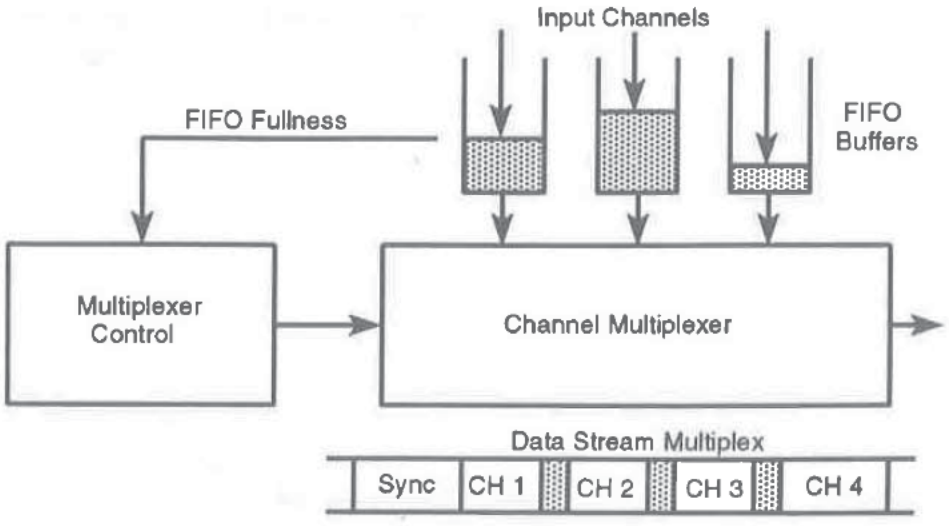


FIGURE 4.9 Asynchronous multiplexing

The output clock rate can be kept constant if the output rate is set to a rate greater than the maximum composite input rate. If this is the case, the input buffers will eventually be empty and no new data value will be available when the proper time slot comes along. We need to put something in the time slot, but what? The obvious answer is that we put in a "fill" word. For this to work, we must be able to recognize fill words and discard them at the receiving end . There are several ways in which the fill words can be recognized at the receiving end, provided there are no channel errors.

Fill words can be chosen to be bit patterns which will never be legal data values. If the data is limited to the printable ASCII character set, for example, a nonprintable character could be used as a fill word. More generally, however, all binary data patterns are possible and the only way to positively identify a fill word is to add additional information to the channel data. An additional bit, or bits, can be appended to each channel value identifing it as fill or data. Alternatively, the TDM format of the output data stream can contain a word, or words, which identify those time slots containing fill words. These approaches are illustrated in Figure 4.10.

In a noise-free environment, the fill words can always be identified and discarded at the receiver. When the channel is noisy, the fill bit identifiers must be protected from error, otherwise fill words may be mistaken for data and vice versa. Regardless of the method used for identifying fill words, a powerful error correction technique must be used. The fill identifiers may be embedded in an error correcting code or a brute force technique may be used such as repeating the identifiers and taking a majority vote.

Asynchronous Packet Multiplexing

An alternative method for handling asynchronous data sources uses packet multiplexing. There are many variations of the packet multiplexing concept and only a generic technique is discussed here.[1] The packet multiplexing scheme buffers a sequence of data values from an individual source, adds preamble and postamble information and transmits the packet independently of the other data packets. Each packet is handled independently and the receiver identifies the data source from the preamble information. A wide variety of packet systems have evolved from Ethernet (the local area network) to satellite systems (derived from the ALOHA system). In many cases, the

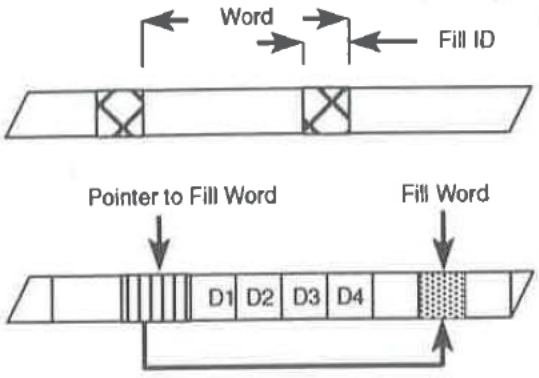


FIGURE 4.10
Identifying fill words using pointers

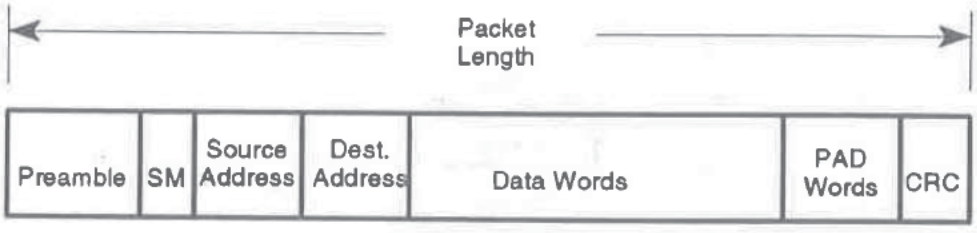
packet multiplexing system is called a "packet switch" since the packets can be directed to a number of locations.

The general packet multiplexer is similar to that multiplexer shown in Figure 4.9. The format of the packets varies from system to system. The packet format for the popular local area network, Ethernet, has been standardized by the IEEE[2] as IEEE Standard 802.3 and is shown in Figure 4.11. A 56 bit (seven octets) preamble of alternate "1" and "0" is used to aid packet synchronization. The preamble is followed by the synchronization marker which is used to identify the start of the data words. A cyclic redundancy code (see Chapter 5) is used to detect packet errors.

The packet multiplexer takes data from each input channel and buffers it in a first-in/first-out (FIFO) memory. When the buffer reaches a predetermined fullness, the controller determines if the communication channel is idle. If it is, the packet is transmitted. If the channel is busy, the input data packet must wait until the channel becomes free. In order to minimize collisions with other inputs, the waiting time is randomly varied as illustrated in the typical controller flow chart shown in Figure 4.12. The local area network protocol, Ethernet, uses the control strategy shown Figure 4.13. In this case, collisions on the communication channel are monitored and according to the number of collisions the random wait time, or "backoff time," is exponentially varied.

Generally speaking, the packet multiplexing technique is used in applications where there are a large number of users with low rate data compared to the communications channel rate and the user traffic is statistical in nature. Telephone traffic is a good example of a statistical data source. If we assume a PCM telephone channel requires 64 kilobits per second and there are 5000 users in a given central office exchange, a TDM multiplexer at the central office would require a 320 megabit per second data bandwidth to handle all users simultaneously. Intuitively, only a fraction of the users are likely to make simultaneous calls and the multiplexer design can take advantage of the statistical nature of the traffic. Data traffic is not so easily characterized as statistical, however, and a slotted TDM multiplexer may be more appropriate for some forms of data traffic.

Packet telemetry standards have been developed in the same manner as telecommunication standards for data transmission. The Consultative Com-



Preamble	=	Seven octets, 1010····
SM (Sync Marker)	=	10101011
Source Address	=	Two or six octets
Destination Address	=	Two or six octets
CRC	=	Cyclic redundancy check code

FIGURE 4.11
Ethernet (IEEE Standard 802.3) packet form

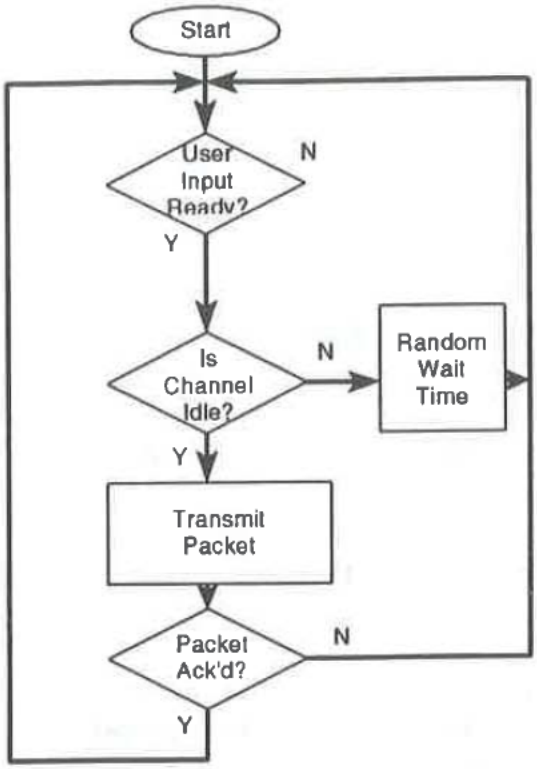


FIGURE 4.12
Packet controller flow chart for network packet collisions

mittee for Space Data Standards (CCSDS) has developed a set of standards for packet telemetry.[3,4,5] The CCSDS standards have been adopted by NASA and various other world organizations for scientific spacecraft applications. The architecture of the CCSDS is complex and contains several hierarchical levels of multiplexing. The network model is based on a spacecraft system with three general subsystems; CCSDS Onboard Network; CCSDS Space Link Subnetwork; and the CCSDS Ground Network. Users can transmit data from a spacecraft to the ground through a variety of services. A simplified model of

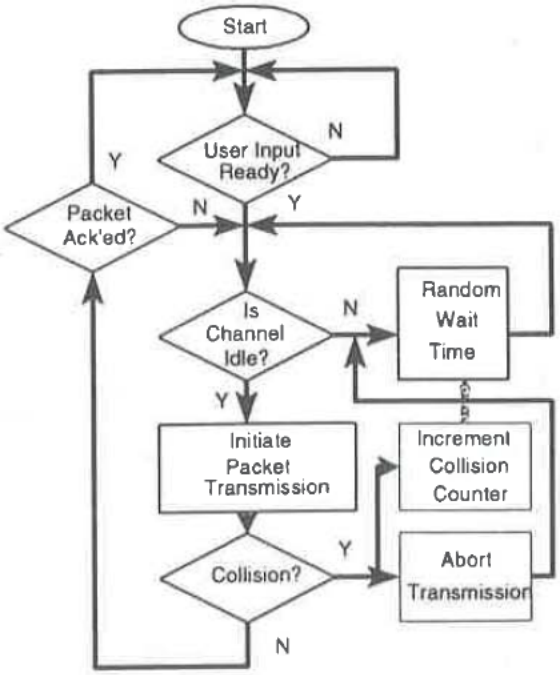


FIGURE 4.13
Ethernet packet control strategy for packet collisions

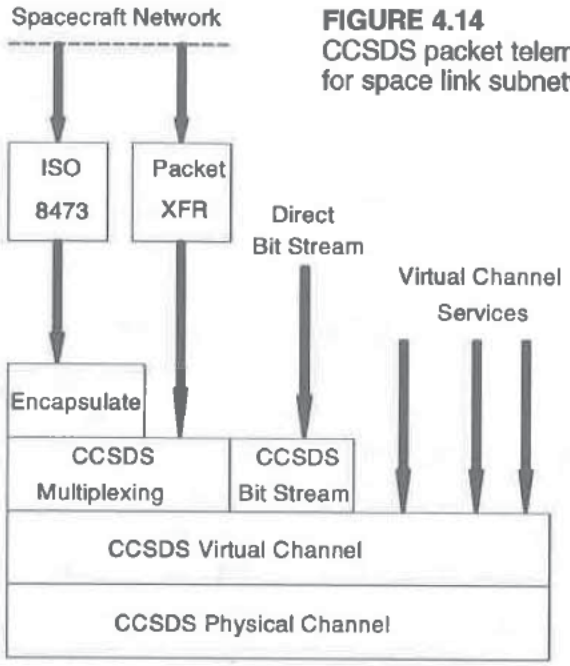


FIGURE 4.14
CCSDS packet telemetry model for space link subnetwork

the Space Link Subnetwork is shown in Figure 4.14. Within the Onboard and Ground Networks, data is transported using a standard packet structure. The Space Link Subnetwork uses a fixed length frame format within the Virtual Channel Data Unit (VCDU) for transmission over the spacecraft radio system. A powerful (optional) error correcting code may be appended to the VCDU data format to correct transmission errors. The packet and VCDU data formats are shown in Figure 4.15. The design of the NASA system is extensively documented in the literature and the interested reader can delve into the system in greater detail through the Chapter references.

FORMAT SYNCHRONIZATION MARKERS

The PCM data is transmitted from the source to the user as a sequence of symbols. The user must first identify the time at which the symbols occur,

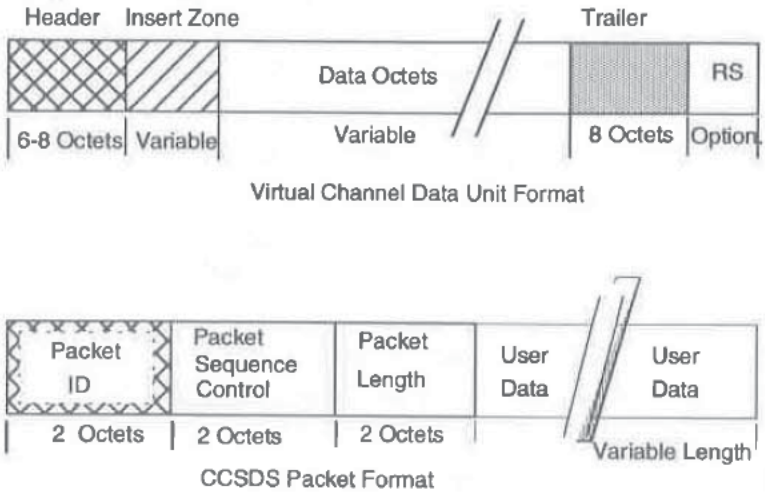


FIGURE 4.15
CCSDS packet format

decide on the symbol values and then identify the data words within the symbol stream. The problem of obtaining the symbol timing is known as symbol synchronization and is considered, in detail, in Chapter 6. Once the symbol timing is obtained, the user must determine the locations of the data words within the format. This is called format synchronization and will be discussed in Chapter 7. In order to obtain format synchronization, some type of synchronization "markers" will have to be imbedded within the data stream. The problem of format synchronization goes back to the early days of telephony. An examination of a table of ASCII values reveals remanents of early synchronization markers. Observe, for example, the ASCII characters SYN (16H), SOH (01H), STX (02H), ETX (03H), and EOT (04H). Could SYN have been intended for synchronization? Could STX and ETX been intended to mark the beginning and end of text? And could have SOH and EOT been intended to mark the beginning and end of a message (packet)? Clearly our forefathers were not dumb when it came to data transmission.

Synchronization markers for TDM formats and packet formats have distinctly different requirements. In the TDM format, the data has a repeating structure with a period equal to the duration of the major frame. This repetition can used to considerable advantage in the design of the format synchronization markers. In a packet system, on the other hand, the user only has one opportunity to acquire synchronization on the packet. If that opportunity is missed, the packet data will be lost.

The synchronization problem also depends on the availability of a priori knowledge of the time of occurrence of a data or synchronization word. The problem of selecting a synchronization marker is considerably different if a genie tells the receiver to expect the synchronization marker within the next few microseconds than if the receiver has no idea when to expect it. Initially we will consider the case in which we have to search for synchronization markers and then consider how to use our a priori knowledge.

Synchronization Marker Design

In order to design a synchronization marker, or "sync word," we need to consider the detection problem at the receiver. The received data, after symbol synchronization, is a serial stream of symbols with an appropriate clock. To establish format synchronization we have to embed sync words in the stream that can be recognized in the data. We have to assume the data to be random so that any pattern of symbols can be present. We will restrict our discussion to binary, i.e., "ones and zeros," symbols. In some cases we may be able to design our system so that a random data pattern can never be identical to a sync word providing that the transmission media does not introduce errors. The HDLC (*H*igh Level *D*ata *L*ink *C*ontrol) format[6] used for some local networks takes this approach, ensuring that no data word can have the same bit pattern as the sync word. In other cases, the sync word can be formed as an illegal transmission code which can be recognized. We will consider the more general case and insist that the sync word be recognizable within a totally random data stream and with possible data bit errors.

There are four possible outcomes when looking for the sync word in the received data stream. These possibilities are all-inclusive and the sum of the probabilities of these outcomes is unity.

- The sync pattern is sent and recognized.
- The sync pattern is sent but not recognized.
- The sync pattern is not sent but a data pattern is incorrectly identified as the sync pattern.
- The sync pattern is not sent and not recognized.

As a model for the sync pattern recognizer, assume the recognizer is a "window" which is as wide as the sync pattern and the received data scrolls past this window as illustrated in Figure 4.16. Each data bit in the window is compared to the known sync bit in the same position. If all of the bits match, the data pattern is identified as a sync marker. We might wish to allow some errors in the match and still identify the data pattern as sync.

Suppose the input data stream is entirely random data with the probability of a "one" equal to the probability of a "zero" equal to ½. Also assume we will not allow any pattern match errors. For this case it is easy to show the probability that an N-bit data pattern exactly matches an N-bit sync pattern is

$$P_{fs} = 2^{-N} \qquad [4.3]$$

With a 10 bit sync pattern, the probability of a random data pattern of 10 bits exactly matching the sync pattern is 2^{-10}, or about 10^{-3}. If the sync recognizer searches through L bits, sequentially, the probability of finding a data pattern which exactly matches the sync pattern is

$$P_{fs} = 1 - (1 - 2^{-N})^L \approx L2^{-N} \qquad [4.4]$$

If the recognizer searches 100 bits for the 10 bit sync pattern there is about 1 chance in 100 that a random group of data bits will be falsely recognized as a sync pattern. If the recognizer allows errors, the probability of data appearing to be sync is even higher. Two things are apparent—the sync pattern must be long to prevent false synchronization in random data and this result is independent of the actual sync pattern.

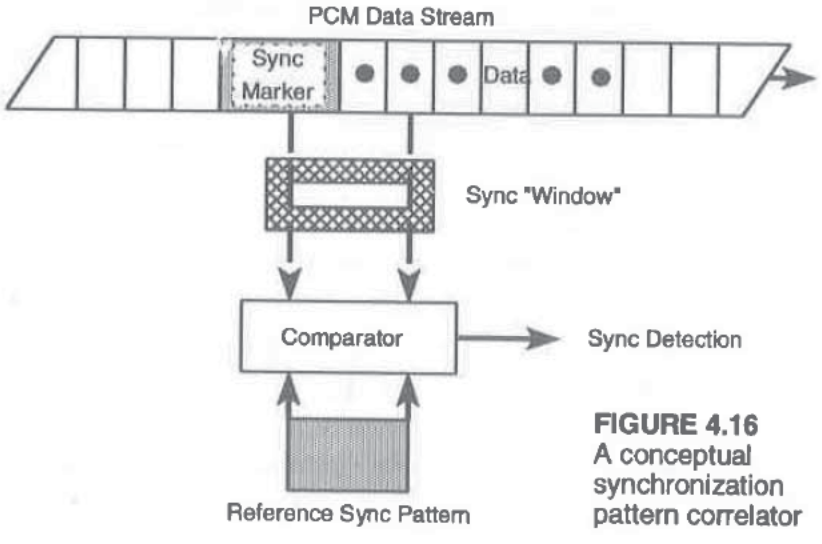


FIGURE 4.16
A conceptual synchronization pattern correlator

In the case of a TDM format, a sync pattern is inserted every minor frame. Since the false sync probability is a function of the search length, the minor frame cannot be too long without an excessive number of false syncs. At the same time, the longer the sync pattern, the lower the false sync probability but the higher the overhead devoted to synchronization. Some of these tradeoffs are deferred to Chapter 7 where synchronization strategies are investigated.

If the false sync probability in random data is not dependent on the actual sync pattern but only on its length, can we choose just any arbitrary pattern for the sync marker? If the answer is yes, a great many people have wasted a lot of time and money searching for "optimum" sync patterns. Obviously there is more to the sync problem than false synchronization in random data. If a sync marker is present, the recognizer must be able to identify it even in the presence of noise. As the sync pattern passes through the sync recognizer (if there are no errors) the pattern will perfectly match at some position in the window and sync will be declared at that point. Suppose the pattern is shifted one bit position to either side of the optimum match point. If pattern errors are allowed it may be possible for the shifted sync pattern to match the allowable pattern to within the error tolerance. This is clearly something that depends on the actual sync pattern. Suppose, for example, that the sync pattern is an all "ones" pattern. Even without errors a "one" bit position slip in either direction with a "one" data bit at either end of the pattern will be a perfect match. At this point, it is evident that the actual bit pattern for the sync marker does matter and the thing that seems to matter most is the correlation of the pattern with itself for various bit slips.

The idea of the "matched filter" was introduced in Chapter 2 as the optimum detector of a signal in random noise. The matched filter is a correlator which multiplies the received signal by a replica of the signal and integrates the result over the duration of the signal. Detection of the sync marker is a similar problem. The sync marker is embedded in a random data stream and we wish to detect it with minimum false alarm probability. Although the marker is embedded in a stream of "ones" and "zeros" rather than random noise, it seems reasonable to apply a detection technique similar to the matched filter. The received sequence is correlated with a replica of the sync marker and the marker is identified when the correlation is a maximum. The fact that the received signal is a binary stream as is the sync marker simplifies the sync detector to a binary correlator which measures the number of places in which the received sequence matches the sync pattern when the sequence is in a window equal to the sync marker length as shown in Figure 4.17. We would like to design the sync pattern so that the correlator output on either side of the peak output is as small as possible. In this way, a bit slip in either direction from the peak will not cause a false sync output.

The binary correlator output can be defined as

$$y_{output} = \frac{number\ of\ agreements\ -\ number\ disagreements}{sync\ marker\ length} \qquad [4.5]$$

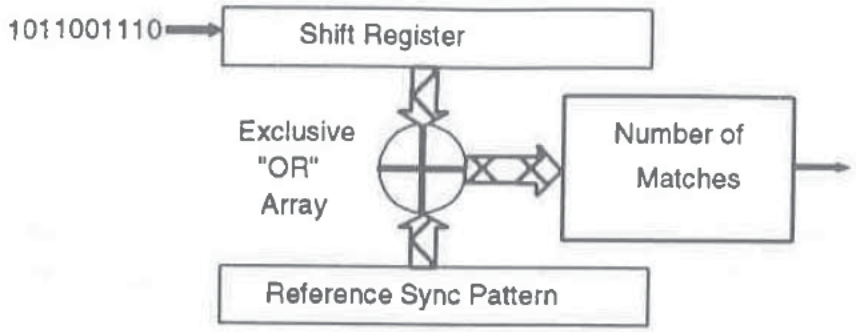


FIGURE 4.17
Digital implementation of synchronization pattern correlator

With this definition, the correlator output is normalized to the range of -1 to $+1$. When the pattern perfectly matches, the output is $+1$, when the pattern is the complement of the sync marker, the output is -1. If random data is in the sync window, we would expect about an equal number of agreements and disagreements with a correlator output of approximately zero. When the sync pattern is partially in the window with random data filling the remainder of the window the correlator output depends on the aperiodic autocorrelation coefficient, C_k, of the sync pattern defined by

$$C_k = \sum_{i=1}^{n-k} x_i x_{i+k} \qquad [4.6]$$

where x_i is either $+1$ or -1.

The x_i terms are the values of the sync pattern bits with a "one" represented by $+1$ and a "zero" by -1.

The design of good sync patterns is a search for patterns in which the autocorrelation coefficients, C_k, have minimum absolute value. Barker[7] was one of the first individuals to search for optimum sync patterns basing his search on patterns with minimum absolute C_k for $k > 0$. Barker found only three patterns in which the value of C_k was either 0 or -1 for $k > 0$. He was able to also able to show that any attempt to make a coefficient more negative would increase the value of some other coefficient. The three optimum Barker patterns are:

Barker Patterns

Pattern Length	*Pattern*
3	110
7	1110010
11	11100010010

Barker conjectured that no longer patterns exist with the minimum autocorrelation coefficients of -1 or 0 and despite exhaustive computer searches none have been found. Barker also suggested that it was possible to find nearly ideal patterns whose coefficients do not exceed $+1$ (except at the correlation peak) by combining the ideal patterns. One possibility is to replace each bit in a Barker sequence with a Barker sequence. For example, a new pattern of length 9 can be obtained by writing the 3 bit sequence down twice in the

same sequence followed by its complement, i.e., 110110001. This pattern has an autocorrelation sequence of:

$$1, 0, -3, 0, 1, 0, -3, 0, 9, 0, -3, 0, 1, 0, -3, 0, 1$$

Thus the maximum autocorrelation value is 9 with the maximum off-peak value of + 1. The off-peak values will be called the pattern "sidelobes." Other sequences with lengths 21, 33, 49, 77 and 121 can be generated in the same manner which exhibit a maximum sidelobe level of +1. Hollis[8] shows how to compute the aperiodic correlation functions of combined Barker sequences and presents other Barker codes which have sidelobes alternating between 0 and +1 and +1, 0, −1. Hollis also shows that the root mean square (rms) sidelobe level of an aperiodic Barker sequence is 0.576 regardless of the length of the pattern and, hence, the peak output to rms sidelobe level for the Barker sequences is 1.735 N, where N is the length of the pattern.

Since Barker's seminal paper, investigators[9,10,11] continue to search for optimum sync patterns. It should be noted that for any pattern, there are three other patterns which have the same autocorrelation characteristics:

- The complemented pattern.
- The reflected pattern.
- The complemented, reflected pattern.

Thus the following four forms of the 7 bit Barker pattern are equivalent:

Basic code	1110010
Complement	0001101
Reflected	0100111
Reflected/complement	1011000

During the early 1960s there was a furious search for "good" sync patterns. Willard,[12] Goode,[13] Magnin,[14] and others published tables of sync patterns (frequently called sync codes). Magnin and Codrington[14] found, in many cases, patterns based on the Legendre polynomials were nearly optimum. As a result of the numerous studies, the Inter-Range Instrumentation Group (IRIG) published[15] a recommended set of sync patterns for lengths from 7 bits to 33 bits. Table 4.1 lists many of the recommended sync patterns published by IRIG and others.

How do we select a sync pattern for a given application? The first and most obvious approach is to select a pattern from Table 4.1. Where more than one pattern is shown, the first choice would be an IRIG Standard pattern. If an IRIG pattern is not recommended, the pattern with the smallest rms sidelobe level is probably the next best choice if a detailed investigation is to be avoided. What do you do if you want a sync pattern of a length not listed? In this case you must design your own code pattern. The most direct starting point for designing your own pattern is to start with existing codes and form a trial code by combining them.

Table 4.1 Synchronization Marker Codes

N	Reference	Code in Hex (delete leading zeros)	N	Reference	Code in Hex (delete leading zeros)
3	Barker	6	23	IRIG	7AE680
4	Barker	D	24	IRIG	FAF320
5	Barker	1D	25	IRIG	1F2DC40
6	Barker	34	26	IRIG	3E9ACC0
7	IRIG	58	27	IRIG	7D69980
8	IRIG	B8	28	IRIG	F5E5980
9	IRIG	170	29	IRIG	1EBCCD00
10	IRIG	370	30	IRIG	3EBCCD00
11	Barker	712	31	IRIG	7F37D420
11	IRIG	5B8	32	IRIG	FE6B2840
12	IRIG	D60	33	IRIG	1F74E9498
13	Barker	1F35	34	Qui-Cheng	3E7B35600
13	Legendre	10DD	35	Qui-Cheng	7EDB18A80
13	IRIG	1D60	36	Qui-Cheng	FCED59600
14	IRIG	39A0	37	Qui-Cheng	1FD5A6C600
15	IRIG	7650	38	Qui-Cheng	3F9DAB2C00
16	IRIG	EB90	39	Qui-Cheng	7FAB92C600
17	Legendre	316D	40	Qui-Cheng	FEAD938C00
17	IRIG	1E6A0	41	Qui-Cheng	1FDB6351C00
18	IRIG	3CD40	42	Qui-Cheng	3FB66B51C00
19	IRIG	7CCA0	43	Qui-Cheng	7FCDA8E4A00
20	IRIG	1DE20	44	Qui-Cheng	FDA216549E0
21	IRIG	1DD2C0	45	Qui-Cheng	1FDAF4716400
22	IRIG	3CDA80	46	Qui-Cheng	3FD9AE2D0800

Synchronization Pattern Design Example

Suppose you need a 36 frame sync pattern. One approach would be to start with concatenating the Barker 3 bit sequence with the Barker 11 bit sequence to form the 33 bit pattern:

110110110001001001110001001110001

Now, add the original 3 bit Barker code to one end or the other of the 33 bit sequence to form the desired 36 bit pattern:

110110110001001001110001001110001110

At this point, compute the autocorrelation coefficients. Other combinations can be tried until the sidelobe levels of the pattern are acceptable.

The synchronization marker can be placed in the format as a contiguous sequence of symbols; i.e., a sync "word" or the pattern can be distributed over a number of frames. If the sync marker is embedded in the format as a word with L bits and is repeated every N bits, then a sync overhead of L/N is incurred. On the other hand, if only one bit of the pattern is inserted every N bits and the entire pattern takes LN bits, the sync overhead is reduced to

1/N but the sync acquisition time is increased by a factor of L. In telemetry applications, the most common practice is to embed a sync marker word in each minor frame. In telecommunications there is a fondness for distributing the sync marker over many frames.

The telemetry practice of using an embedded sync marker every minor frame trades off sync overhead for more rapid acquisition. This is important in data transmission applications where it is important to acquire as much data as possible. In telecommunications digital voice applications, rapid acquisition is not as important as using the communications bandwidth efficiently. In telemetry applications, from 1% to 3% of the bandwidth is devoted to format synchronization, while in telecommunications, the sync overhead is typically less than 1%.

In the DS-1 format[16] used for 24 voice channels, one framing bit is inserted every 192 bits (24, 8-bit channels). The framing bit has been used in several ways over the years. In one implementation, the framing pattern is a 12 bit sequence (110111001000) which is formed by interleaving an alternate "10" sequence with an alternate "111," "000" pattern. The alternating patterns can be used to identify odd and even frames used in some data channel units. The two repetitive sequences are a very poor choice for synchronization since tones in the data can produce bit sequences identical to the framing sequence and cause reframing errors. This was recognized and six bits in the 24th frame byte were allocated to frame synchronization. During sync acquisition, the six fixed framing bits are used with the 12 bit framing pattern used in the lock mode. Once the six-bit pattern has been located, four consecutive framing bits are required for sync verification. One frame has a period of 125 microseconds so that the minimum acquisition time is at least 0.5 milliseconds. Since the DS-1 format was designed for digital voice, this acquisition time is acceptable since little loss in intelligibility would be encountered if the system had to acquire sync during a call.

Subframe Synchronization

In the general definition of a multiplex format, what is commonly called a "frame" in telecommunications applications might be more appropriately called a minor frame in multiplexer applications. In the DS-1 multiplex, for example, the framing pattern repeats over 12 minor frames. In telemetry applications, most formats can contain one, or more, subframes and, in many cases, subsubframes. In both the telemetry and the telecommunications worlds, a means for identifying the beginning (or end) of the subframes is important. One obvious technique would be to start each subframe with a unique sync marker in the same manner as the minor frame sync word. In telemetry, this method has become known as a "recycle" sync since the unique pattern is placed in the minor frame at which the subframe "recycles." The sync marker can be located as the first word of the subframe or at some other location in the same minor frame. Although each subframe could have an independent sync marker, in most cases it is desirable to locate other subframes with common reference subframe whenever feasible. When several

subframes are present with differing lengths, it is difficult to decide if a given subframe requires a sync marker. The subframe with the longest length must have a sync marker.

A second subframe synchronization method used in telemetry applications is the "count-down," or "ID" (for identification) method. In this method, one word of the minor frame is allocated to a count which identifies the current sampled channel number in the subframe. This requires greater overhead but allows more rapid acquisition. Although this method is vulnerable to noise, the count can be locked to a local reference and counts perturbed by noise can be rejected.

In simple formats, the minor frame sync pattern can be complemented to identify the start of subframes. In the telecommunication formats, the equivalent of subframe information is normally conveyed with a distributed technique. The interleaving of two sync markers as used in DS-1 is an example of a method for identifying odd and even minor frames.

System Design Considerations

The acquisition of format synchronization is discussed in Chapter 7; however, we need to think about the synchronization problem when we design the format. The format synchronizer, commonly called the "frame synchronizer," has the task of finding the sync markers in the format in order to identify the frame and word boundaries. Assuming binary data, the synchronizer initially must search the incoming data stream for a pattern, or patterns, which match the known sync marker. Once a candidate sync location has been identified in the search, the location is then checked by looking for the same pattern at the same location in the next frame. Once the location has been verified, the synchronizer can go to a "lock" mode in which a small window is placed around the marker location and synchronization is maintained as long as the marker continues to occur within this window.

In the search or acquisition mode, the synchronizer is searching mostly through data for the marker. As indicated earlier, there is a false sync probability, approximated by Equation [4.4], that a pattern of data will be incorrectly mistaken for the sync marker. The longer the frame and the shorter the sync marker, the higher the probability of false sync. In this mode, the exact sync pattern is relatively unimportant and the length of the pattern and the frame length must be considered. It is also important to design the format so that data patterns which randomly match the sync pattern do not repeat every frame. One way to avoid synchronizing on a constant data pattern which exactly matches the sync marker each frame is to alternately complement the sync marker and then search for the pattern and its complement. In most data applications, the format words are multiples of 8 bits. So that choosing a 24 bit or 32 bit sync marker is a reasonable choice to minimize false sync probability.

Once initial synchronization has been achieved, the synchronizer continually maintains a window around the sync marker location to verify the lock condition. In the presence of noise or jitter, the most probable synchro-

nization error is for the symbol synchronizer to slip a clock cycle. If the frame synchronizer window is wide enough to accommodate a small clock slip in either direction about the initial sync location, clock slips can be detected and format synchronization can be re-established without reverting back to a search mode. In this mode the sync pattern is extremely important so that pattern slips of several bits can be clearly identified. The aperiodic correlation coefficients are important in this mode and patterns should be selected with C_1, C_2 and C_3 minimized. The recommended patterns in Table 4.1 have been chosen with this criteria in mind.

TELECOMMUNICATIONS MULTIPLEXING

PCM systems were introduced into the telecommunications network to replace analog voice circuits. Indeed, PCM to most telecommunications engineers is synonymous with 64 kilobit per second digitized voice. Prior to the PCM systems voice circuits were multiplexed using frequency division multiplexing (FDM), placing each voice channel on a separate amplitude modulated subcarrier. The North American PCM multiplex hierarchy is based on the AT&T network design. The lowest level PCM multiplexer is at the T-1 (AT&T nomenclature) or DS-1 (generic nomenclature) level. T-1 and DS-1 are frequently used interchangeably although, in this book, I will use the DS-x terminology. The DS-1 multiplexer is designed to combine 24 digital voice channels. The analog voice channels nominally have a bandwidth of about 3 kiloHertz and the digital sampling rate was chosen to be somewhat higher than the minimum Nyquist sampling rate. A sampling rate of 8 kilosamples per second was chosen as the base sampling rate and 8-bits per sample was chosen as the nominal sample size. Since all of the channels are identical, the 24 channel multiplex format required a basic frame size of $24 \cdot 8 = 192$ bits. A single, distributed frame sync bit was added to make a frame length of 193 bits with a frame rate of 8 kilosamples per second to produce a 1.544 Mbps composite rate. Within this basic structure, bits are periodically robbed from data channels to create an in-band signalling channel. Using the DS-1 multiplex as the lowest level, a hierarchy of multiplexes are formed as illustrated in Figure 4.18. Four DS-1 signals are multiplexed to form a DS-2 multiplex (6.3 Mbps) and seven DS-2 multiplexes are combined to form the DS-3 level (44.736 Mbps). Within AT&T, six DS-3's are combined into a DS-4 level at 274 Mbps. Outside of the common carriers, only the DS-1 and DS-3 multiplexes are normally used. Multiplexing levels above DS-3 vary widely and no common standards have been established.

All multiplex structures of the telecommunication hierarchy are based on bit level formats. The DS-1 format has been discussed earlier and is illustrated in Figure 4.19 as an array of 12, 193-bit minor frames, each with a 125 microsecond duration. At this level, the DS-1 multiplexer is normally considered to be synchronous, combining 24, 64 kilobit per second channels. When used for data transmission, the input channels may be asychronous

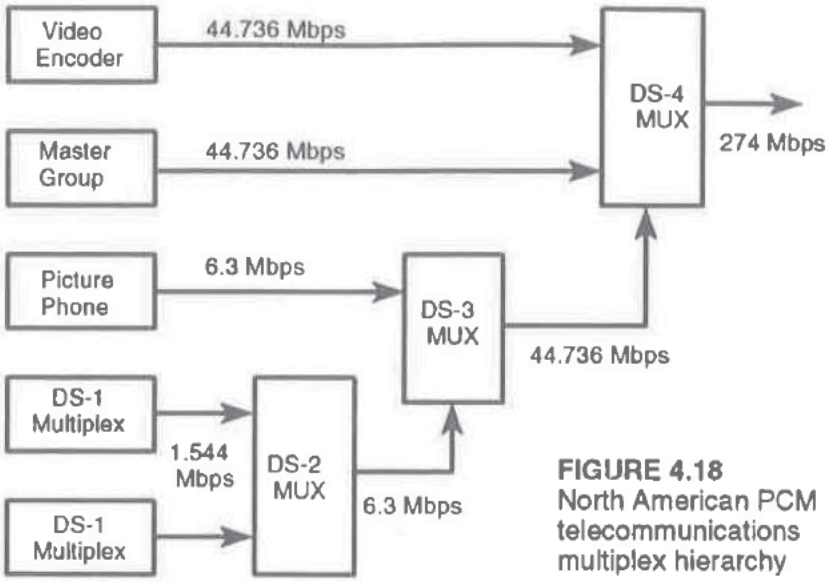


FIGURE 4.18
North American PCM telecommunications multiplex hierarchy

but the data terminal handles the data buffering and the DS-1 multiplexer combines the data channels from a common clock.

Above DS-1, there are several multiplex structures. Two DS-1 channels have been multiplexed into a DS-1C signal at 3.152 Mbps and four DS-1 channels are combined to form a DS-2 format at 6.3 Mbps. Two DS-2 signals have been combined to form a DS-A signal at 12.9 Mbps. Seven DS-2, or 28 DS-1 signals are multiplexed to form the DS-3 signal at 44.7 Mbps. Of these multiplexes, only DS-1 and DS-3 are of general interest outside of the common carriers, primarily because they are the only services which are tarriffed. The multiplexers associated with these formats are generally termed "M_{ij}" for a multiplexer that goes from level i to level j, i.e., a M_{12} multiplexer combines DS-1 signals into a DS-2 multiplex.

The multiplexes above DS-1 are asynchronous in the sense that the inputs do not have a common clock. Although a DS-2 multiplexer combines four DS-1 data streams, each with a nominal 1.544 Mbps rate, the clocks of the four streams may be slightly different in frequency due to drift, aging,

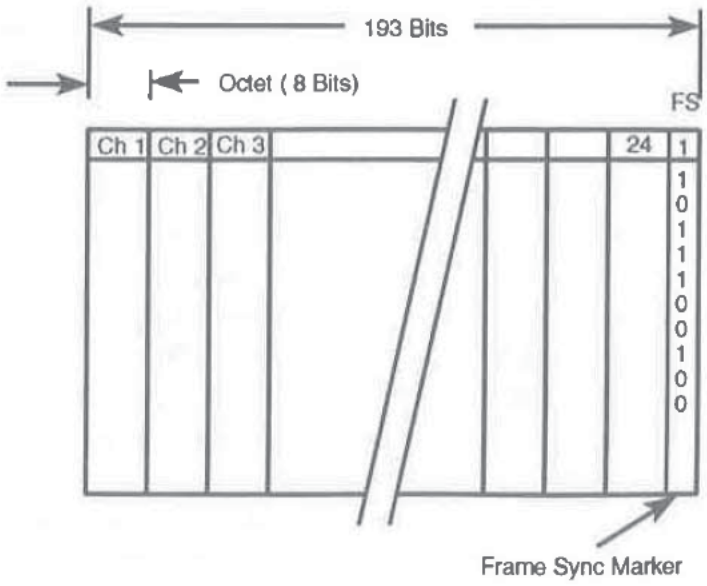


FIGURE 4.19
DS-1 PCM Format

environmental and other factors. Suppose two DS-1 data streams differ in frequency by

$$\delta f = f_1 - f_2 \text{ in Hertz} \quad [4.7]$$

The two data streams will slip in phase by one complete cycle in

$$t_{flip} = \frac{1}{\delta f} \text{ in seconds} \quad [4.8]$$

or

$$\begin{aligned} N &= f_b ts_{lip} \\ &= \frac{f_b}{\delta f} \text{ bits} \end{aligned} \quad [4.9]$$

Even if the two data streams are within 0.001% in frequency, the two data streams will have slipped a bit in 10^5 bits, or about every 65 milliseconds.

In order for the multiplexer to handle the asynchronous data streams without losing information bits, the data streams must be buffered and the output data rate must be greater than the average composite data rate. This implies that there will be times when the multiplexer is ready to accept a bit from an input stream but the input buffer is empty. In this case, the multiplexer must insert a "dummy" (or "stuff") bit into the output format. Having inserted the stuff bit, the multiplexer must insert a code into the format to identify to the demultiplexer that a stuff bit is present and must be removed. The telecommunication formats are based on a bit stuffing technique for the DS-1 through DS-3 multiplexers to handle the asynchronous data streams.

The DS-1C, DS-2 and DS-3 formats are shown in Figure 4.20 as array structures. The DS1C format, for example, has a 318 bit minor frame which is repeated every four frames. A stuffing indicator word identifies a frame with a stuffing bit. Four stuffing bit locations, one in each minor frame, are available for stuffing when necessary. A distributed, alternating "10" pattern is used for frame synchronization with a "011x" pattern used to identify the minor frame. The "x" bit in the frame alignment code is used to convey alarm information. The DS-2 format has a similar structure but with a 300 bit minor frame. Like the DS-1C format, a bit stuffing opportunity is presented each minor frame.

The DS-3 format is composed of seven minor frames with 8 blocks each. Each block contains 85 bits with 84 information bits and one bit shared between sync, status and control functions. The eighth block within each minor frame[17] is allocated to stuff bits. The control bits, $C_{i,j}$, are used to identify stuff bits. The framing pattern consists of a distributed "1001100110011001" pattern. The "subframes" are identified using a "010" pattern in the first bit of the subframes. The preoccupation with periodic sync sequences simplifies the synchronization design but does leave the design vulnerable to false sync problems. (The choice of seven minor frames in the DS-3 format is not simply because seven is a lucky number.) The format was optimized to multiplex

DS-1C Format

M	D		D	F	D		D				D	F	D
0	52	C11	52	0	52	C12	52	C13	4	I/S	47	1	52
1	52	C21	52	0	52	C22	52	C23	5	I/S	47	1	52
1	52	C11	52	0	52	C12	52	C13	4	I/S	47	1	52
x	52	C21	52	0	52	C22	52	C23	5	I/S	47	1	52

DS-12 Format

M	D		D	F	D		D		D	F		D
0	49	C11	49	0	49	C12	49	C13	49	1		I/S	48
1	49	C21	49	0	49	C22	49	C23	49	1	1	I/S	47
1	49	C11	49	0	49	C32	49	C33	49	1	2	I/S	46
x	49	C41	49	0	49	C42	49	C43	49	1	3	I/S	45

M = Major frame alignment bits
D = Data bits
F = Frame marker bits
I/S = Information/service bits
x

DS-3 M-Frame Format

X1	84	1	84	C	84	0	84	C	84	0	84	C	84	1	84
X2	84	1	84	C	84	0	84	C	84	0	84	C	84	1	84
P1	84	1	84	C	84	0	84	C	84	0	84	C	84	1	84
P2	84	1	84	C	84	0	84	C	84	0	84	C	84	1	84
0	84	1	84	C	84	0	84	C	84	0	84	C	84	1	84
1	84	1	84	C	84	0	84	C	84	0	84	C	84	1	84
0	84	1	84	C	84	0	84	C	84	0	84	C	84	1	84

1	2	3	4	5	6	7	8
							Stuff Blocks

FIGURE 4.20 DS-1C, DS-2 and DS-3M PCM formats

seven DS-2 signals to provide a total capacity of 28 equivalent DS-1 channels in DS-3. The common implementation of a DS-3 multiplexer internally combines groups of four DS-1 inputs into DS-2 data streams and then combines seven DS-2s into the final DS-3 stream.

The key to the higher level multiplexers lies in the synchronizer/desynchronizer (syndes) circuit which combines the asynchronous data streams by inserting and removing the stuffing bits. In the conventional telecommunication multiplexers, the syndes circuit operates on serial data streams. A simplified block diagram of the syndes circuit is shown in Figure 4.21. The transmitted data is sequentially written into an elastic buffer (typically 8 bits deep)

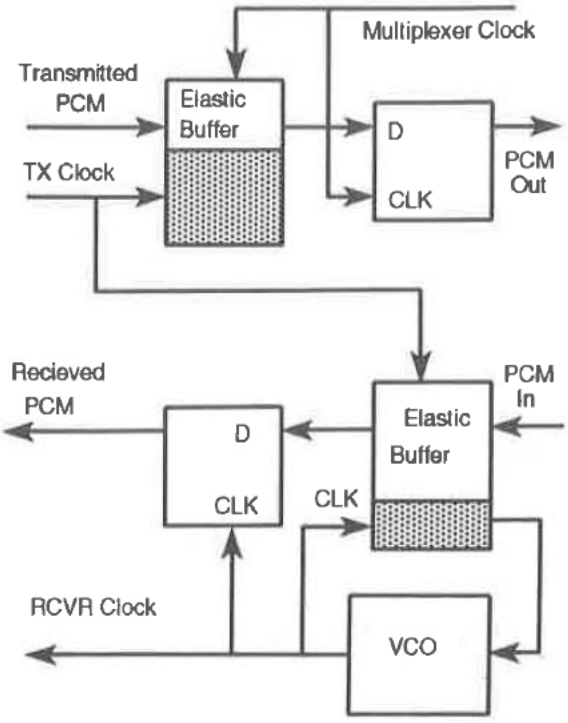


FIGURE 4.21
Synchronizer/Desynchronizer (SYNDES) circuit

by the data clock and read out by the multiplexer clock. When the elastic store is depleted, a stuff request is sent to the multiplexer where a stuff bit is inserted into the data format. At the receiving end, the received data is written into an elastic store by the destuffed clock and the buffer is readout by a clock which is phase-locked to the average destuffed clock rate.The bit stuffing technique works well with the standard formats when all inputs have approximately the same rate. In some applications, input rates can vary widely and more sophisticated multiplexers are required.

Elastic Stores

Elastic buffers are implemented by first-in/first-out memories with separate read and write clocks. Suppose the memory is one bit wide and serial data bits are written into the memory at one rate and read out at another rate. If the readout clock is stopped and data is written into the memory at the input, the buffer will begin to fill with bits. If the readout clock then begins to readout the memory at a rate faster than the input rate, the buffer will begin to deplete. The "fullness" of the buffer can be monitored and stuffing requests can be made when the buffer becomes depleted.

SONET

The DS-1 level communication link was initially implemented using wire cables. At the DS3 level, microwave radio and fibre optic cable is required. Fibre optic cable is rapidly replacing wire cable in the telecommunication networks. Until recently, each common carrier developed its own proprietary transmission formats for the fibre optic cables. The Synchronous Optical *Net-*

work (SONET) standards[18] were developed as an effort to standardize the fibre optic transmission formats. The SONET standards are being developed through the ANSI in the U.S. with compatible standards (SDH) concurrently developed under the CCITT. The basic multiplexing format for SONET was developed in 1989 with additional clarifications and enhancements in 1990. The electrical specifications for the first two interface levels, STS-1 and STS-3, were issued in 1991. In the future, the SONET standards are likely to be revised to conform to the CCITT standards.

At least part of the driving force behind the SONET standards can be attributed to local exchange carriers (LECs) who see significant advantages in compatible optical systems with a high degree of equipment integration. The SONET standards define a format with a common overhead structure supporting a variety of data structures, termed "payloads." SONET defines a hierarchy of transmission data rates in integer multiples of 51.84 Mbps with effective payload transmission rates in integer multiples of 50.112 Mbps. The SONET standards define both the electrical and the optical interfaces. The electrical interfaces are defined as STS-n with n currently defined for n = 1,3,9,12,18,24,36 and 48. The line rate for STS-1 equals 51.84 Mbps, STS-3 becomes 155.52 Mbps and so forth. Similarly, the optical interfaces are defined as Ocn where n = 1..255. All values of *n* are defined for the optical interface while only a few of the electrical interfaces will be defined and available.

The format of STS-1 is illustrated in Figure 4.22 as an array structure with 90 columns and 9 rows of octets (8-bit words). The basic array maintains the 125 microsecond frame time (8000 samples per second) so that the STS-1 has a composite rate of 51.84 Mbps. The first three columns of the array contain the transport overhead octets. The fourth column contains path overhead bytes leaving the remaining 86 columns for data. Thus the effective data rate available in STS-1 is

$$f_d = \frac{86 \cdot 9 \cdot 8}{125\,\mu s} = 49.536\ Mbps \qquad [4.10]$$

The SONET payload is generally asychronous with the SONET format. While the DS-x multiplexes handled asynchronous inputs by bit stuffing, the SONET format provides octet stuffing. Further, the payload can drift through the overall format. A pointer in the transport overhead structure points to

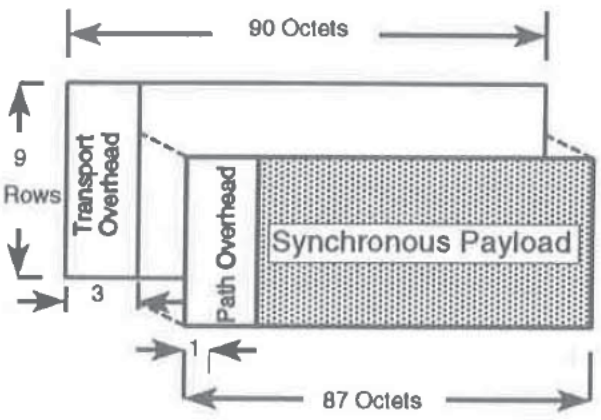


FIGURE 4.22
SONET STS-1 format

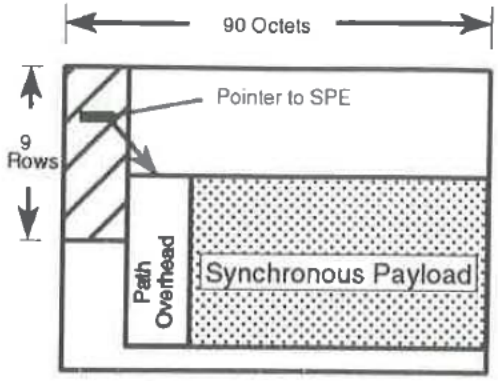


FIGURE 4.23
Synchronous payload pointer in STS-1 format

the starting octet of the payload format. The pointer is located in the fourth row of the STS-1 overhead as shown in Figure 4.23 and consists of two octets which point to the payload starting octet and a third octet which is shared among several functions. In SONET both stuffing and "destuffing" can be accomplished. If the payload is not ready to transmit an octet, a stuffing octet is inserted in the octet slot immediately following the third octet of the pointer and the pointer is incremented to accommodate the increase in the payload data. On the other hand, if the payload has an excess of octets available, an additional octet can be added by replacing the third pointer octet by the data octet and the pointer is decremented.

Within the payload envelope a variety of data formats have been defined to accommodate the DS-x signals. Data streams have been defined in terms of a hierarchy starting with "virtual tributaries" (VT) which can be grouped into "tributary groups" (VTG). The virtual tributaries include the DS-1, DS-1A, DS-1C and DS-2 signals according to the following table.

Types of Virtual Tributary Data Streams

Virtual Tributary	*Signal*	*Number per VTG*	*Number per Payload*
VT-1.5	DS-1	4	28
VT-2	DS-1A	3	21
VT-3	DS-1C	2	14
VT-4	DS-2	1	7

Note: VTG = Virtual Tributary Group

The SONET synchronous payload envelope (SPE) can be configured as a 9 row by 87 column array with either 86 columns for DS-3 or 84 columns for 7 virtual tributaries (Vts). Alternatively, the payload can contain a virtual tributary group (VTG) with 12 columns and 1 to 4 Vts. The higher levels of the SONET format are obtained by adding columns in increments of 90 columns as shown in Figure 4.24. At the present time there are several variations of the payload formats within the STS-n format. The SONET standard will define the telecommunication transmission format well into the 21st century and details of higher level formats will continue to evolve.

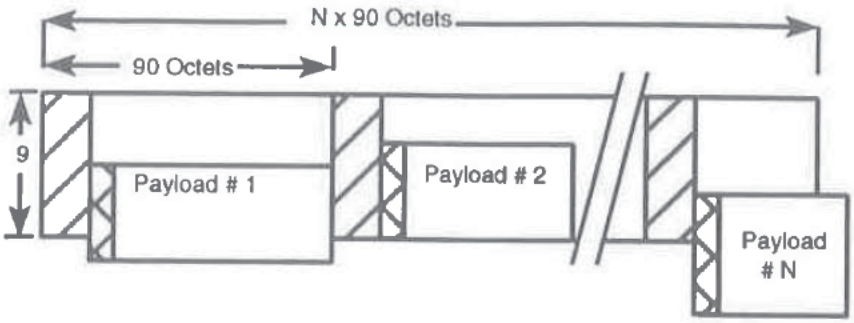


FIGURE 4.24
Higher level SONET formats

Broadband ISDN

A major standardization effort has been underway since the mid-1970s to define an *I*ntegrated *S*ervices *D*igital *N*etwork (ISDN) to provide a single communications network design capable of providing a single "wall socket" for both voice and data. It has been estimated that nearly 500 million dollars have been spent on ISDN and more than 5000 papers on ISDN have been published.[19],[20] It now seems clear that SONET can provide the physical communication layer for the ISDN concept and now the "buzzword" is Broadband-ISDN, or B-ISDN. The challenge is to be able to multiplex all kinds of signals, not just digital voice and slow speed data. Video data from a variety of sources, as well as much higher speed data, must be accommodated by B-ISDN. The multiplexing problem is embedded in the overall network concept.

In SONET, the various signaling levels have been defined and interfaces established for a subset of the possible levels. The data transport formats have been defined as indicated in earlier sections and the role of the multiplexer is to combine data sources, placing the digital data stream octets in the SONET payload. Stuffing and destuffing of the digital stream is handled through pointers to the payload data and the stuffing slots. This technique is an extension of the bit stuffing methods used in the DS-1 and DS-3 multiplexes.

A number of proposals have been made for B-ISDN and the CCITT currently recommends a multiplexing technique known as *A*synchronous *T*ransport *M*ode, or ATM, for multiplexing and switching data or digital voice. The ATM technique, also known as cell relay, combines input data streams into 53 octet packets with a 5 octet header and 48 octet information payload as illustrated in Figure 4.25. The CCITT recommendation defines two options for the ATM packets at the physical communication layer: a *S*ynchronous *D*igital *H*ierarchy (SDH) transport format or a packet based transport format. The SDH format is based on the SONET structure in which ATM cells are embedded in the basic SONET multiplex format using the existing SONET overhead structure. The cell transport option, also known as *A*synchronous *T*ime *D*ivision (ATD), transmits the ATM cells directly with any overhead information carried in ATM packets. The B-ISDN ATM protocol is defined as shown in Figure 4.26. The ATM protocol layer is directly over the physical communication layer and provides a common ATM interface to the communications network. Since two transport options are defined, the ATM streams can be exchanged between the two optional transport methods.

Within the B-ISDN concept, the customer premises equipment (CPE) is

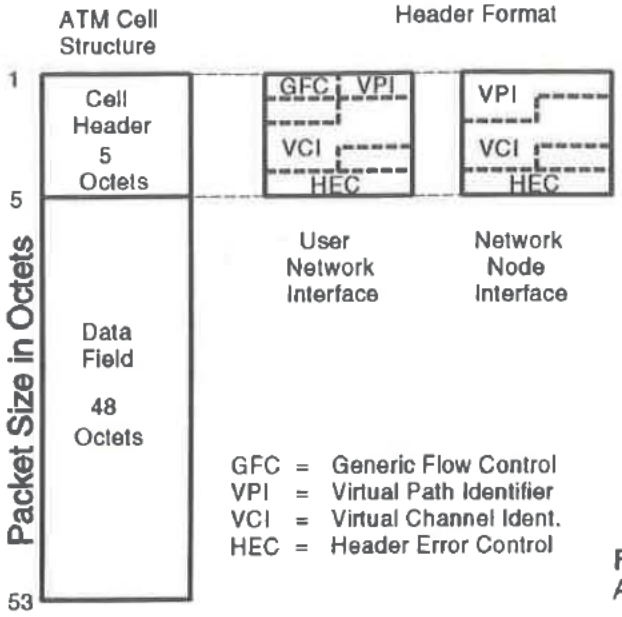


FIGURE 4.25
ATM cell format

connected to the central office via a digital subscriber line as shown in Figure 4.27. The ATM multiplexer, located in the central office, can interface to either the SONET SDH or directly to a cell-based transmission line. It is also conceivable that private networks may be implemented by using an ATM multiplexer at the customer premise connecting to a private line. In either case, the ATM multiplexer is a "statistical" multiplexer which is designed to take advantage of the burst nature of the source packets. Voice packets occur in bursts according to the speaker and have been modeled as modulated Markov processes. Interactive data sources have similar characteristics and even file transfers can appear bursty when viewed macroscopically. Recently, it has been suggested[21] that mixtures of TCP/IP traffic sources exhibit "scale similarity," a property of chaotic processes in which the process looks the same regardless of the time scale. The transmission network has a fixed bandwidth and the multiplexer must control the average data flow to match the transmission channel. Buffering the input cells in FIFOs can provide the statistical multiplexing capability up to the point at which the input buffers are full. Once the buffers are full, the multiplexer has no alternative but to discard

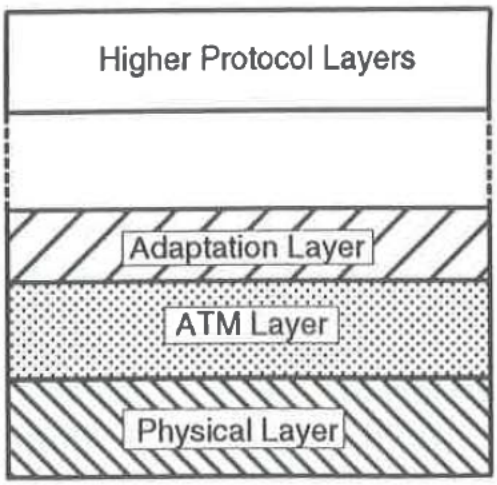


FIGURE 4.26
Broadband-ISDN (B-ISDN) ATM protocol reference model

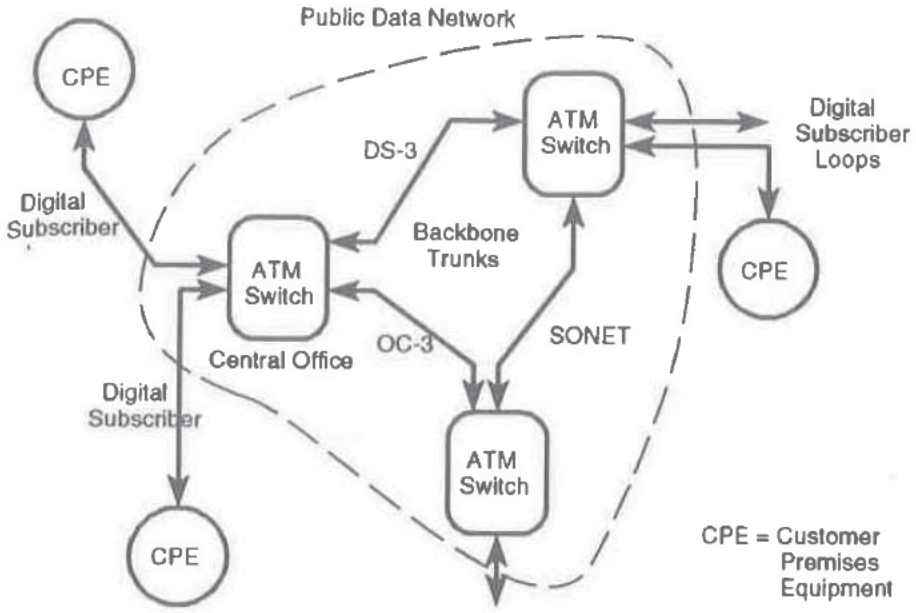


FIGURE 4.27
B-ISDN network using ATM switching

input cells until the buffers can accommodate them. With an error detection and correction capability the loss of cells is not catastrophic provided the data source has stored the cells until the receiver confirms that the cells have been received correctly. In the case of voice, or some images, the occasional loss of cells can probably be tolerated without correction.

The interest in ISDN and networked PCM systems is enormous and is beyond the scope of this book. The reader is directed to the article by Kano [18] and Stallings[19] book which provide good overviews of this complex subject.

Frame Relay, SMDS and ATM

Until recently, the bulk of data communication over the public networks used either modems, dedicated leased lines, or X.25 packet services. Starting in the late 1980s, some new data services have been announced and limited services offered for frame relay, Switched Multimegabit Data Service (SMDS) and ATM. All of these services are packet based. They have been developed independently of B-ISDN with the goal of reaching the marketplace as quickly as possible within the confines of technology.

Frame relay[22] was developed in response to the need for higher rate data transmission than could be provided by X.25 services. The X.25 data services are packet based but tend to be limited to rates in the order of 64 kilobits per second because of the nature of the X.25 protocol. In X.25 each data packet as it traverses a network is checked for errors at each node and if in error packet retransmission is requested. When X.25 was designed, the error checking was necessary for a robust service over the quality of communication links used. The price paid for the error protection is throughput. With the improvement in quality of communication links, the added overhead of the error checking has been deemed unnecessary. Frame relay packet communications was introduced in which error checking was deferred to the end user. With this change, throughput rates in excess of 2 Mbps are possible. Frame relay is based on a

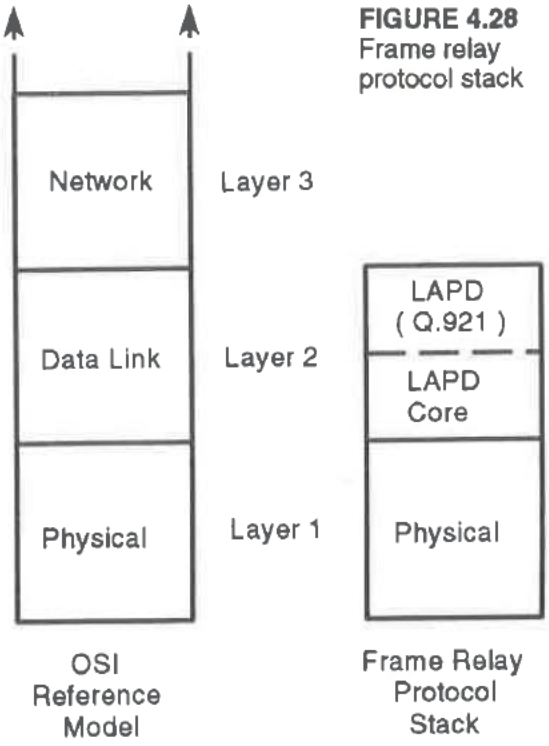


FIGURE 4.28
Frame relay protocol stack

three level protocol stack as illustrated in Figure 4.28. The frame relay level 3 packet format is shown in Figure 4.29 and is commonly referred to as the Level 3 Protocol Data Unit, or L3_PDU. Recovery of the data payload requires packet framing synchronization, a level 2 function. Although this book is focussed on the physical layer of the PCM system, level 2 framing recovery is considered in Chapter 7. In practice, the PCM system designer is not usually concerned with framing recovery of the L3_PDU since that function is buried in a standard silicon chip.

SMDS is a packet service developed by Bellcore[23] to provide 2Mbps to 45 Mbps data services to metropolitan areas. SMDS uses a Distributed Queue

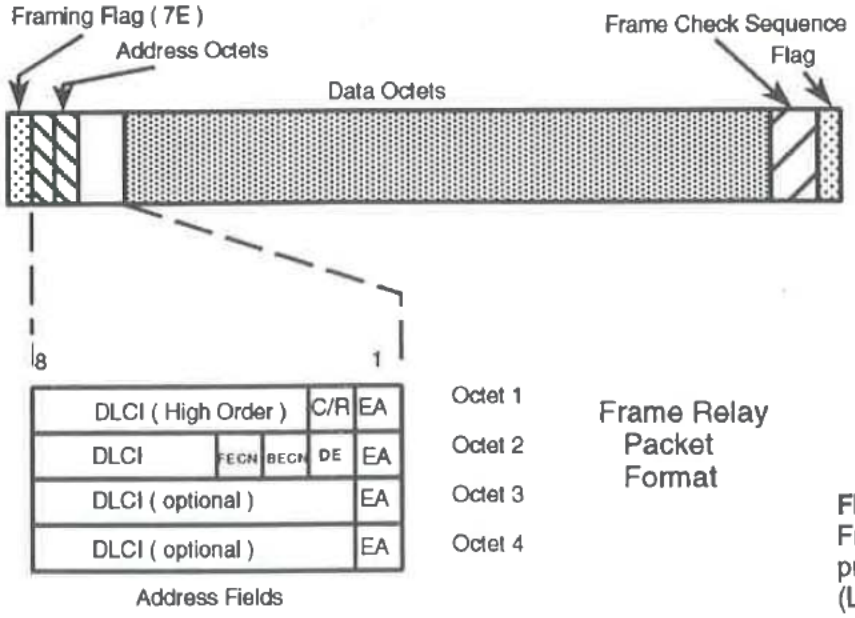


FIGURE 4.29
Frame relay level 3 protocol data unit (L3_PDU)

Dual Bus (DQDB) access method as described in IEEE 802.6 Standard. Like frame relay, SMDS uses a three level protocol stack with an L3__PDU format as shown in Figure 4.30. SMDS is a connectionless service as opposed to frame relay which is connection oriented. As a connectionless service, SMDS packets are dynamically routed within the network according to the packet address. In frame relay, the connection between source and destination is established at call connection time and the connection remains static for the duration of the data transmission. As a connectionless service, SMDS has many characteristics of local area networks (LANs) and SMDS is described as extending the LAN to metropolitan wide areas. SMDS packet formatting and framing synchronization is commonly handled with standard integrated circuits.

As discussed previously, the CCITT recommended the use of the so-called Asynchronous Transfer Mode, or ATM, for B-ISDN services. In B-ISDN, the ATM packets would be carried within the SONET payload. As with frame relay, users and vendors were anxious to implement a broadband data service in advance of the B-ISDN standards. To this end a nonprofit organization known as the ATM Forum was established with the express purpose of defining a broadband service based on ATM cells (packets) as the basic data unit. Because of the adoption of the 53 octet ATM cell format, the service has become known as "ATM" and ATM has become synomous with 53 octet cells no matter what the application. A simple ATM network is illustrated in Figure 4.31 with three ATM switch nodes. The initial implementation of ATM is a connection oriented service. The ATM Forum Standards define a "native" ATM service in which ATM cells from a variety of sources are multiplexed into virtual path connections from one node to another. ATM is a two level protocol requiring minimal processing at intermediate nodes in the network. In addition to the native ATM service, interoperability with frame relay and SMDS has been considered and frame relay services over an ATM network have been defined. Network switches are available which offer both native ATM services and frame relay over ATM.

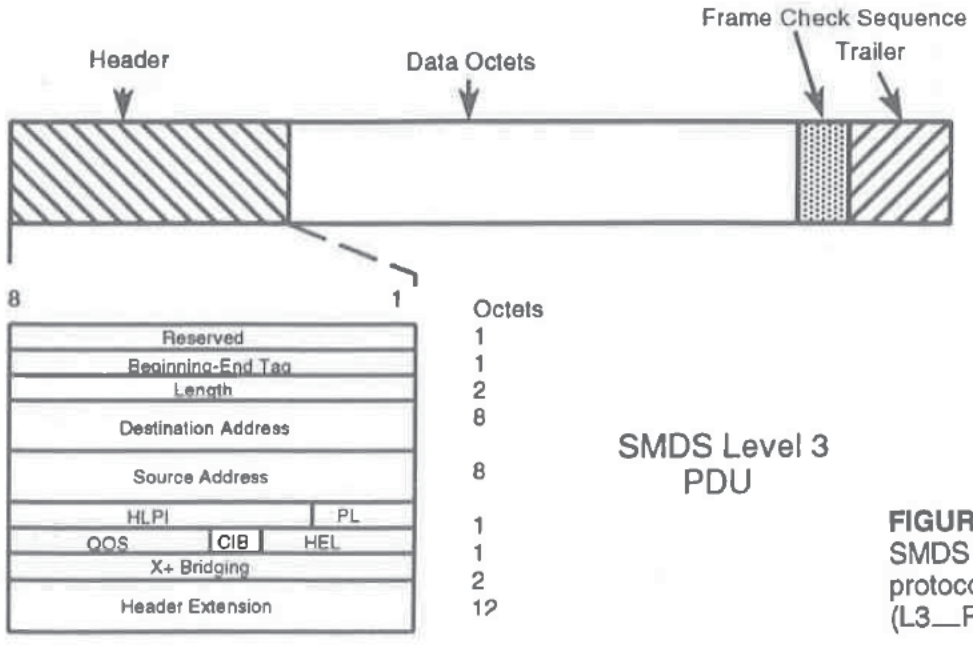


FIGURE 4.30
SMDS Level 3 protocol data unit (L3__PDU)

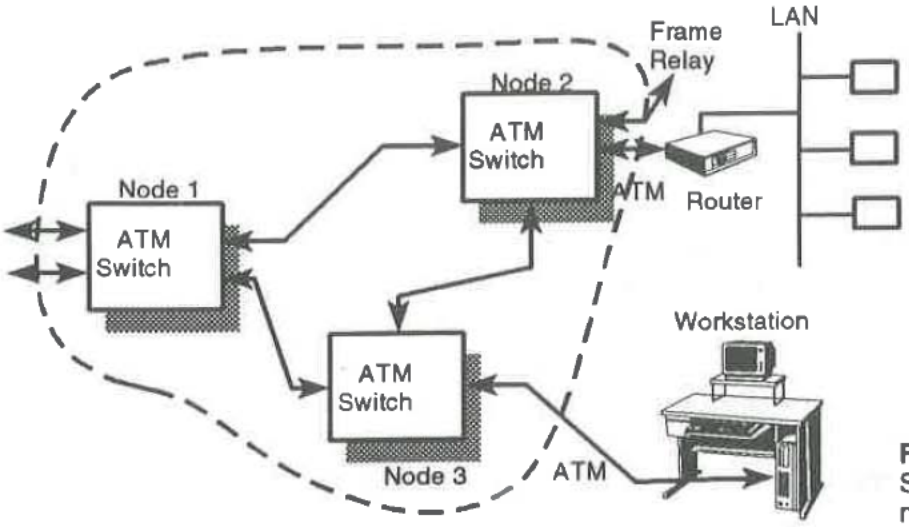


FIGURE 4.31
Simple ATM network

To provide very high data rate services, ATM uses a two level protocol stack to minimize processing at a switching node and eliminates error processing based on the assumption of communication circuits with very low error rates ($<10^{-9}$). An octet in the ATM cell header checks for errors in the cell header and the network discards cells with header errors to prevent cells from being delivered to the wrong address. Errors in the cell payload are the responsibility of the user. Standard integrated circuits are now available for processing ATM cells and for providing segmentation and reassembly of higher level PDUs into ATM cells.

TELEMETRY MULTIPLEXING

Multiplexing in telemetry systems has traditionally been synchronous with a frame structure as discussed previously. More recently, packet telemetry systems, particularly in space telemetry applications have been defined and implemented. Unlike the telecommunications industry, telemetry requirements have been so diverse that no format standards have been established. Frame lengths, number of subframes, number of superframes and synchronization markers have all been left to the individual user. Thus there are about as many telemetry formats as there are users. The space telemetry community has made an effort to standardize telemetry formats for space applications and have adopted a packet telemetry format and service model as previously shown in Figures 4.14 and 4.15.

The synchronous telemetry formats are governed to a large extent by the availability of the telemetry multiplexing equipment. A typical airborne telemetry multiplexer/encoder is shown in Figure 4.32. The multiplexer/encoder is a complete data acquisition subsystem comprised of signal conditioning modules, multiplexing modules, a control unit and the interface to the transmission system. The important features are associated with the format size limitations. A typical performance specification[24] is listed below:

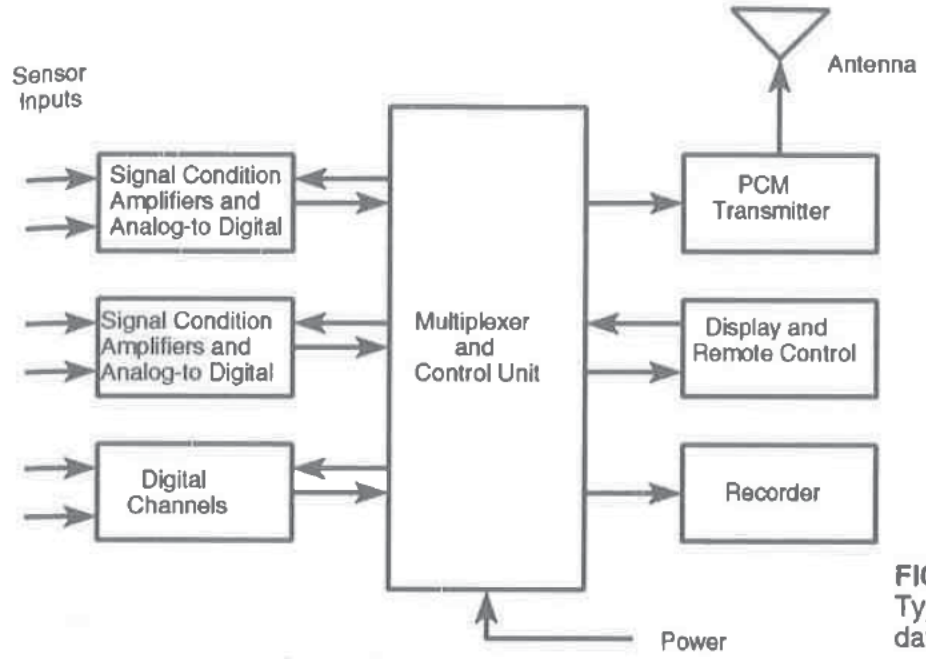


FIGURE 4.32
Typical airborne PCM data acquisition system

Input channels	to 2048
Words/frame	to 2046
Frames/subframe	256, maximum
Supercommutated channels	1023
Word length	8,10,12,14 or 16 bits
Word rate	to 200 kilowords/second

In designing a synchronous telemetry format, the capabilities of existing multiplexing equipment must be considered. In practice, most telemetry formats use less than 256 minor frame words and synchronization markers in the 24 bit to 32 bit range. Ground processing equipment is available to decommutate formats with over 8000 channels and include the capability to handle subframe, subsubframe and supercommutated channels. In more recent formats, some of the embedded data is asynchronous with the minor frame rates and special processing must be employed to handle the asynchronous data channels.

SUMMARY

Pulse code modulation systems rarely consist of one data source transmitting to one receiver. A part of the power of PCM lies in combining multiple data sources into a single *T*ime *D*ivision *M*ultiplex system. The combining of multiple input sources into one format can be surprisingly complex, particularly when the individual sources are sampled at differing rates. An ad hoc design approach is presented for configuring a PCM format including subframe and superframes which is efficient in the sense of minimizing format overhead.

The common telecommunication format structures have been described with an emphasis on the DS-1, DS-3 and SONET formats. The DS-x formats have been structured with digital voice channels in mind. The integration of

nontelecommunication data streams into the telecommunications network is an interesting design challenge.

A trend toward packet communication techniques both within telecommunications and telemetry is developing. The use of ATM cells for B-ISDN is recommended for future telecommunications networks. Packet telemetry is well established in space telemetry applications. Packet communication techniques impose unique constraints on symbol and format synchronization designs as well as the multiplexer design. The intense drive toward ATM as the standard telecommunications packet format for broadband applications has produced standard integrated circuits for processing ATM cells at level 2 and above.

REFERENCES

1. Lidinsky, W. and Vlack, D., ed, *Perspectives on Packetized Voice and Data Communications,* New York: IEEE Press, 1991. A recent collection of articles which discuss a number of packet networks for PCM voice and data.
2. *Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications,* ANSI/IEEE Std. 802.3-1985, published by IEEE, 1989. The IEEE has published several related standards including 802.4 and 802.5 for token ring networks.
3. CCSDS 701.00-R-3, "Recommendation for Space Data Standards, Advanced Orbiting Systems, Network and Data Links: Architectural Specification." CCSDS Red Book, Issue3, (June 1989).
4. CCSDS Recommendation for Space Data System Standards: "Packet Telemetry." Blue Book, (January 1987).
5. Hooke, Adrian "Advanced Orbiting Systems: A Standard Architecture for Space Data Communications." Proceedings of the ITC, 25(1989). This article gives a good overview of the space data communications architecture. This issue of the Proceedings of the ITC has a number of papers dealing with various aspects of the CCSDS recommendations.
6. *American National Standard for Advanced Data Communication Control Procedures (ADCCP),* X3S4/475, (13 February 1974). The ADCCP standard defines a format and protocol similar to HDLC and SDLC, the format originally defined by IBM. The HDLC, SDLC and ADCCP standards are often used interchangeably and integrated circuits are available to transmit and receive in any one of these formats.
7. Barker, R.H. "Group Synchronization of Binary Digital Systems," *Communication Theory*, ed by W. Jackson, London: Butterworth Scientific Publications, (1953)273-87. The seminal paper on synchronization codes.
8. Hollis, E.H. "Predicting the Truncated Autocorrelation Functions of Combined Barker Sequences of Any Length Without Use of a Computer," IEEE Transactions on Aerospace and Electronic Systems, AES-3, no. 2, (March 1967).
9. Qiu-Cheng, Xie and Ji-Qing, Ouyang "A New Method Searching Optimum Frame Synchronization Codes for PCM Telemetry Systems," *Proceedings of the ITC*, 22(1986).
10. Qie-Cheng, Xie and Jie, Chao "A New Searching Method of Optimum or Suboptimum Frame Synchronization Codes for PCM Telemetry Systems," *Proceedings of the ITC* (abstract only) 23(1987).
11. Qiu-Cheng, Xie and Jie, Chao "The Searching Method of Quasi-optimum Group Sync Codes on the Subset of PN Sequences," *Proceedings of the ITC* 26(1990).
12. Willard, M.W. "PCM Frame Synchronization," *Proceedings of the NTC* (1961).
13. Goode, G.E., and Phillips, J.L. "Optimum PCM Frame Synchronization Codes and Correlation," *Proceedings of the NTC* (1961).

14. Magnin, J.P. and Codrington, R.S. "Legendre PCM Synchronization Codes," National Symposium on Space Electronics and Telemetry (October, 1962).
15. *IRIG Standard 106-86 Telemetry Standards*, Telemetry Group Range Commanders Council, U.S. Army White Sands Missile Range, New Mexico 88002 (May 1986, revised September 1989). A copy of the IRIG Telemetry Standard can be obtained from Loral Test and Instrumentation Systems, Telemetry Marketing, Telemeter Editor, PO Box 3041, Sarasota, FL. 34230. The IRIG Standards are periodically updated and a later edition is likely to be available by the publication of this book.
16. Benowitz, P., et al "Digital Multiplexers," *The Bell System Technical Journal* 54(May-June 1975). Although this article is not the first to describe the format of DS-1 it does provide some insight into the DS-1 telecommunications multiplexers.
17. In the telecommunication industry the "minor" frame in the DS-3 multiplex is called a "subframe," which may cause confusion unless the reader is paying attention.
18. *ANSI Standards T1.105, T1.106, T1.105R1, T1.102A,* (1989-1991). SONET was originally proposed by Bellcore to ANSI in 1985 as a standard interface for fibre optic lines. ANSI, in turn, passed the proposal on to the CCITT in 1986. The ANSI standards referenced were issued starting and 1989 and continuing. The CCITT is also issuing similar standards under the name of SDH, *S*ynchronous *D*igital *H*ierarchy.
19. Kano, Sadahiko, et al "ISDN Standardization," *Proceedings of the IEEE* 79(February 1991). This article gives an excellent overview of the ISDN standardization process. The entire February 1991 issue of the Proceedings of the IEEE is devoted to ISDN.
20. Stallings, W. *ISDN: An Introduction* New YorK: Macmillan, 1989. Stallings book received favorable reviews in the *Proceedings of the IEEE*, February 1991 special issue on ISDN.
21. Leland, W.E. et al "On the Self-Similar Nature of Ethernet Traffic," *Proceedings of SIGCOMM* ACM (1993).
22. Smith, Philip *Frame Relay, Principles and Applications*. New York: Addison-Wesley, 1993.
23. "Generic System Requirements in Support of Switched Multi-megabit Data Service," TR-TSV-00072 Bell Communications Research, Inc. (May 1991).
24. *EMR 5000 Series Advanced Data Acquisition System Specification* Sarasota, FL: Loral Data Systems (April 1990).