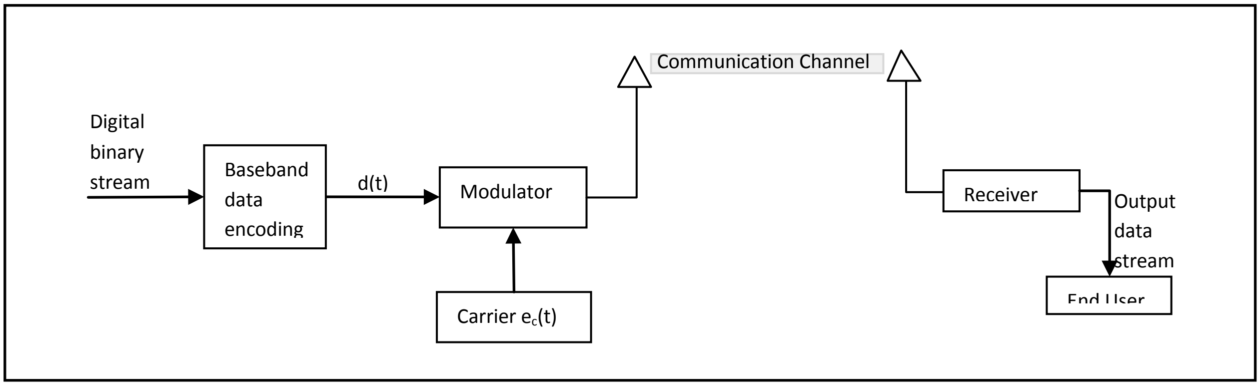
Digital Baseband Transmission
(Line Coding Techniques)

A communication system requires a transmitter to convey information from source to the destination. This is done by modulating a signal on a carrier as shown below.



The signal d(t) is an electrical voltage (or current) waveform which is obtained by mapping the binary bits of input binary data stream to signal waveform. This mapping of digital binary bits to signal waveforms is called **Line Coding.** Thus Line coding are the techniques of representing a binary digital signal by waveforms suitable for transmission over various media. Binary bits '0' and '1' are mapped to two distinct waveforms (signal elements) S_1 and S_2 for transmission over the communication channel. Each signal element lasts for a fixed interval T which is decided by the clock signal and this in-turn decides the data rate and the band width requirements of the communication channel.

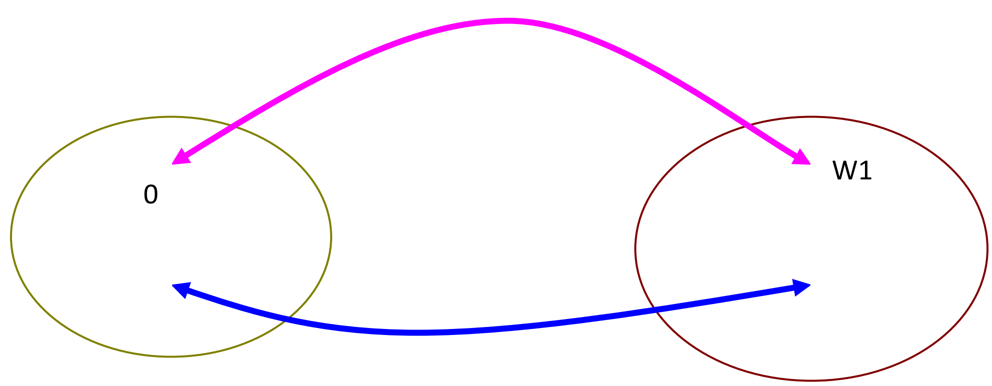


Fig: Mapping of Binary bits to waveforms

Desirable Features of a good line code:

A line code must have the following properties;

- *Transmission Bandwidth*

Transmission bandwidth of the code should be as small as possible

- *Adequate timing information for synchronization*
  It should be possible to extract timing or clock information from the line code so that synchronization between transmitter and the receiver is maintained

- *No DC component*
  It should not contain any DC component as AC coupling is used in the system

- *Error detection and correction capability*
  A line code should help in error detection and correction

- *Transparency*
  It should be possible to transmit a digital signal correctly regardless of the pattern of 1's and 0's.

Line Codes

A number of line codes have been proposed, studied and examined. Each has its own advantages and disadvantages. Some common line codes used are;

- Non Return Zero unipolar (NRZ-UP)
- Non Return Zero Bipolar (NRZ-BP)
- Non Return Zero-Mark (NRZ-M)
- Non Return Zero-Space (NRZ-S)
- Return Zero Unipolar and Bipolar Codes (RZ-UP, RZ-BP)
- Manchester Code
- Alternate Mark Inversion (AMI)
- Bipolar with 8-zeros substation (B8ZS)
- High Density Bipolar-n
- Block Line cods

Non Return Zero-unipolar

- Here 1 is transmitted as +V for entire clock cycle and 0 is transmitted as 0V for the entire clock period
- The code has maximum bandwidth which is half of the bit rate
- It is a transparent line code
- The code develops a DC component which gets lost due to ac-coupling.
- No clock information is available if a long sequence of 1s or 0s occur in the data
- The code does not have any error detection capability.

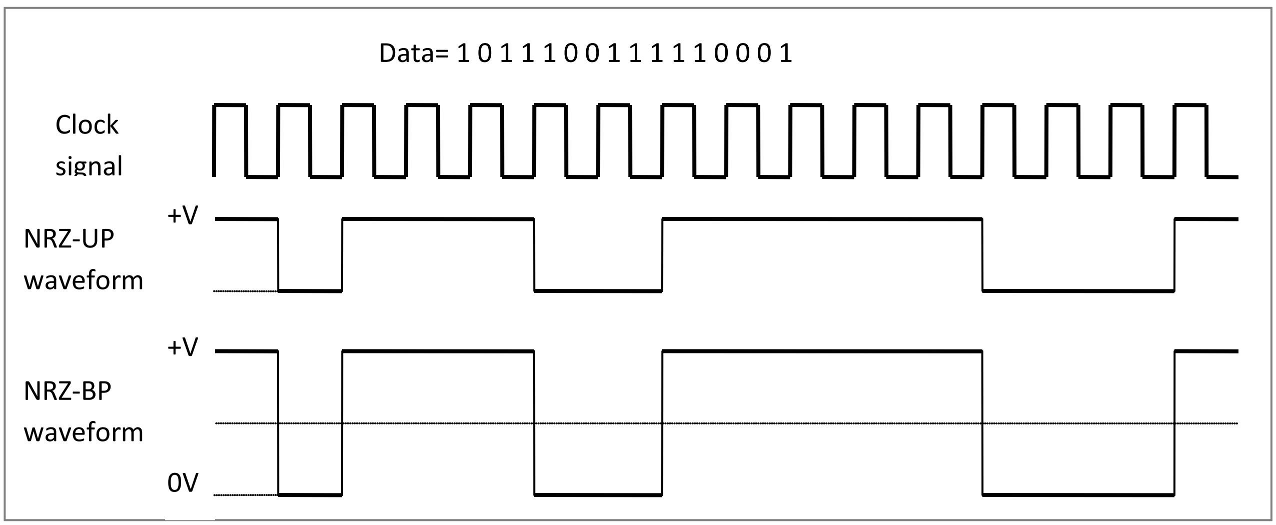
Non-Return Zero- Bipolar

- Binary 1 is transmitted as +V and binary 0 is represented as -V

- Bandwidth of the code is half of the bit rate
- No DC component is developed
- No clock information is available if a long sequence of 1s or 0s appears in the data stream
- It is a transparent code
- The code does not have any error detection and correction capability.

Non-Return Zero-Mark (NRZ-M)

- In this encoding scheme '1' is represented by change in voltage and '0' is represented by no change in voltage
- The bandwidth of the code is half the bit rate
- DC component is developed
- No clock information is available if a long sequence of 0s appears in the data stream
- The code does not possess error detection or correction capabilities



Non-Return Zero-Space (NRZ-S)

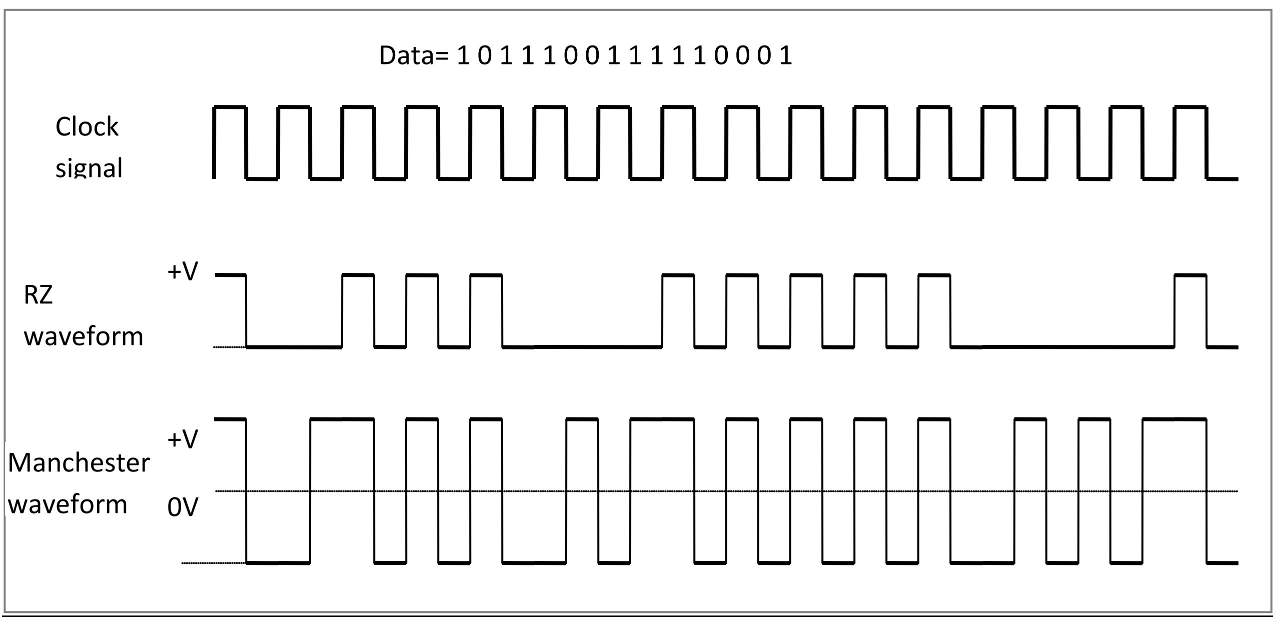
- In this technique of encoding '0' is represented by transition and '1' is represented by no transition in voltage
- The bandwidth of the code is half the bit rate
- DC component is developed
- No clock information is available if a long sequence of 1s appears in the data stream
- The code does not possess error detection or correction capabilities

Return Zero Code

- Binary 1 is transmitted as a transition from +V to 0V for unipolar and +V to –V for bipolar RZ line code. 0 is transmitted as 0V. Transition from high to low takes place usually in the middle of bit interval 1 and no transition during bit 0
- Bandwidth of the code is equal to the bit rate
- RZ- unipolar suffers from DC component
- RZ unipolar and bipolar donot provide adequate timing information if a long sequence of 0s is transmitted.

Manchester Code

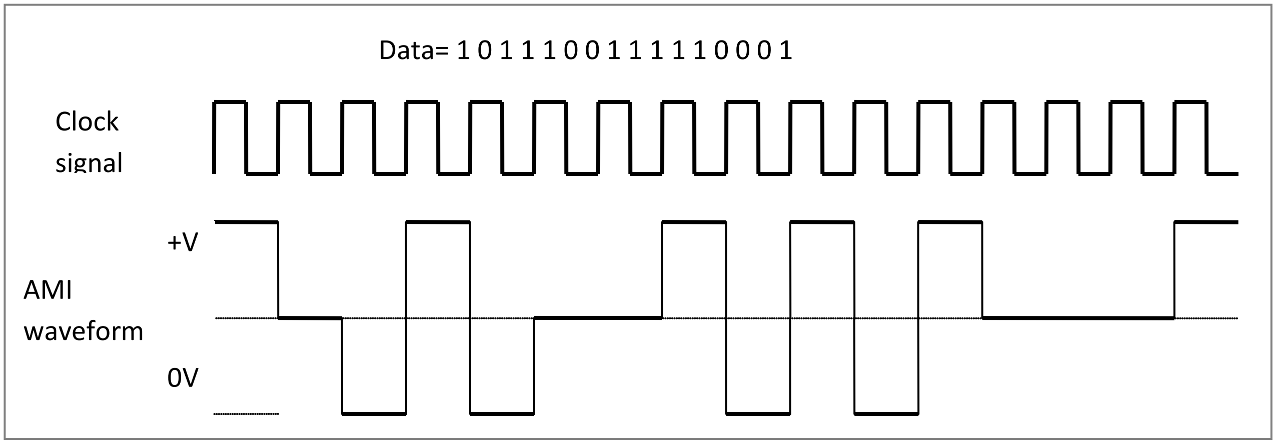
- Binary 1 is encoded as a transition from **HI** to **Low** and binary 0 is encoded as a transition from **Low** to **HI**. Transition occurs during the bit interval.
- Band width of the code is equal to bit rate
- No DC component is present
- Adequate timing information is available for synchronization
- It is a transparent code



Alternate Mark Inversion (AMI)

- Binary 1 is alternatively encoded as +V and –V while Binary 0 is transmitted as 0V
- Bandwidth of the code is one-half of data rate.
- DC component is abscent
- It is transparent
- The code possess error detection and correction capabilities because any code violation due to additive noise can be detected and in some cases corrected.

- The system loses timing synchronization if a long sequence of zeros appears in the data stream.



Bipolar with 8-zeros substation (B8ZS)

- The scheme is used by telephone companies in North America and Canada
- It is similar to AMI except that for maintaining timing synchronization if a continuous octet of zeros occurs in the data then the 8-zeros sequence is replaced by a waveform sequence forcing two code violations.
- If an octet of all zeros occurs and the last voltage pulse preceding the octet of 8 zeros was positive then 8-zero sequence is encoded as 000+−0−+
- If an octet of all zeros occurs and the voltage pulse preceding this octet was negative then 8-zero sequence is encoded as 000−+0+− (See table below)
- Bandwidth of code is one-half of bit rate
- DC component is not present
- It facilitates timing synchronization when long sequences of zeros occurs
- It has error detection and correction capability

Table depicting the encoded waveforms for octet of 8 zeros

Preceding pulse Polarity	Encoded waveform for octet of 8 zeros
+ve	0 0 0 + − 0 − +
-ve	0 0 0 − + 0 + −

High Density Bipolar-n

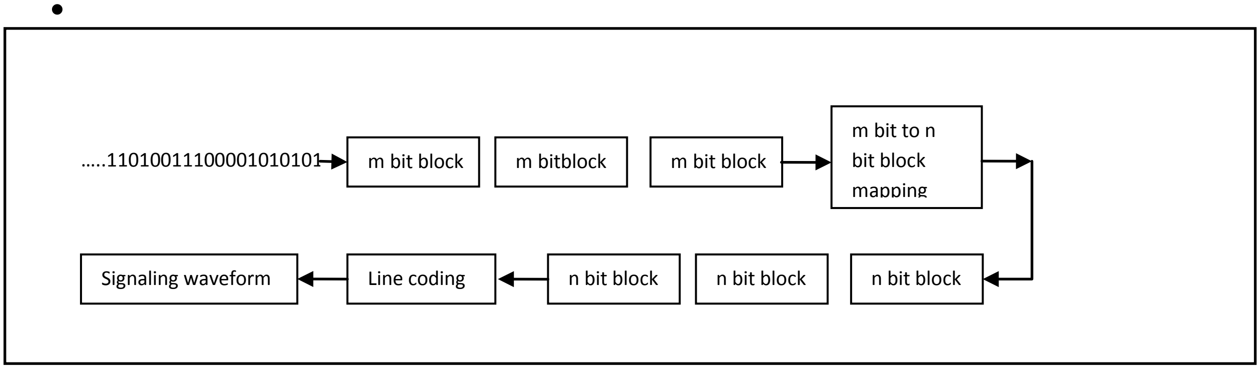
- This line code is used by telephone companies in Europe , Japan and other developing countries
- The scheme replaces a sequence of 4 zeros by code violation
- If a sequence of 4 zeros occurs and the last pulse preceding it is positive then a sequence **0000** is encoded as **000+** provided the number of 1s occurring since last such substitution is odd. If the number of 1s occurring since the last such substitution are even then **0000** sequence is replaced as **−00−**

- If a sequence of 4 zeros occurs and the last pulse preceding it is negative then a sequence **0000** is encoded as **000−** provided the number of 1s since last substitution is odd. If the number of 1s are even then **0000** sequence is replaced as **+00+**
- Bandwidth of code is one-half of bit rate and DC component is absent
- It facilitates timing synchronization when long sequences of zeros occurs
- It has error detection and correction capability

Preceding pulse Polarity	No of 1s since last substitution **Odd**	No of 1s since last substitution **Even**
+ve	0 0 0 +	− 0 0 −
-ve	0 0 0 −	+ 0 0 +

Block Line Coding Schemes

- Block line codes differ from above coding schemes because these operate on a block of incoming information bits.
- To ensure synchronization and error detection redundant bits are added to each block of incoming bits.
- In these schemes a pattern of m binary bits is encoded into a pattern of n signal elements, where m < n.
- These line codes are called mB/nT line codes
- With m bits there can be 2^m data patterns and with n signal elements there can be 2^n signal patterns, thus 2^m data patterns can be mapped to a subset of 2^n signal patterns
- Important block codes used are 4B/5T, 8B/10T etc
- The disadvantage of Block line codes is that these increase the signal rate.



Assignment

(Q.1) Explain the Manchester coding and AMI coding Schemes.

(Q.2) Sketch neatly the NRZ(BP), RZ(BP), Manchester and AMI wave forms for following binary data streams

(i) 1 0 0 1 1 1 0 1 0 0 1 1 1 1 0 0 1
(ii) 0 0 1 1 1 0 0 0 0 1 0 1 0 1 1 1 0
(iii) 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1

(Q.3) ketch neatly B8ZS and HDB3 wave forms for following data stream
0 0 1 1 1 0 0 0 0 1 0 1 0 1 1 1 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1

(Q.4) Write a detailed note on block codes and their applications.

Course Title: Digital Communications

Course Content:

(1) Advantages of Digital communication; Block diagram of digital comm. system, Signal and its characteristics, Bandwidth, Band limited signal, power spectral density of signal, sampling, sampling theorem, signal reconstruction, quantization, quantization-noise and signal-to-noise ratio due to quantization.
(2) Pulse code modulation, Uniform and non-uniform quantization, voice companding, A-law, μ-law, Time Division Multiplexing, AT&T standard, European standard, Differential PCM, ADPCM,
(3) Error control coding; parity check, error detection codes, error correcting codes, linear block codes, polynomial codes, hardware implementation through shift registers, Convolution codes
(4) Waveform encoding (Line coding) Techniques
(5) Digital modulation techniques; ASK, FSK, M-ary FSK, PSK, BPSK, Differential- PSK, M-ary PSK, QAM
(6) Performance of Digital modulation techniques
(7) Spread spectrum communication

References:

(i) Digital Communication by J G Proakis Tata McGraw Hill
(ii) Digital communications Fundamentals & Applications by Bernard Sklar ; pearson education publication
(iii) Communication system Analog & Digital by Sanjay sharma
(iv) Communication principles by Taub & Schilling Tata McGraw Hill Publication
(v) Analog and Digital Communication by Shanumugam

<u>Evaluation Process:</u>

Test I:	20
Test II:	20
Test III:	50
Attendance:	05
Assignments	05

Digital Communication:

In digital communication, the information bearing signal has to be in the form of a sequence of binary coded numbers. The signal is represented by long sequences of 1s and 0s. The information to be conveyed may already be available in digital form e.g. text or it may be available in analog form e.g. audio signal or video signal. An analog signal is converted to digital signal through sampling and quantization processes. This digital signal is used to modulate the carrier and then transmitted to the destination.

Advantages:

(i) Better signal to noise ratio and low probability of bit error.
(ii) Signals can be reconstructed in original form on the way through repeaters.
(iii) Signals can be easily stored in the field and then can be easily transported to point of transmission.
(iv) Digital signal processing techniques can be used for signal conditioning.
(v) VLSI technology can be effectively utilized
(vi) Digital computers can be used for signal storage and processing
(vii) Off the field signal processing is possible.
(viii) Error detection and correction techniques can be employed
(ix) Signal time compression and expansion is easily done
(x) Time division multiplexing and demultiplexing can be done for transmitting signals of different bandwidths simultaneously on same wide band width channel
(xi) Encryption and decryption techniques can be employed to enhance the security of the signal
(xii) Spread Spectrum Techniques can be used to combat jamming. This also facilitates code division multiplexing access.
(xiii) Voice, Text, Video, data and Graphics signals can be combined together. This is called ISDN and multimedia services.

Block Diagram of a Digital Communication System

General block diagram of a digital communication system is given in fig (1) and (2) below.

The various blocks of a digital Transmitter are;

(a) Information source
(b) Sensor/ Transducer
(c) Signal conditioning Block
(d) Sampling
(e) Quantization and encoding
(f) Data Compression
(g) Error encoding
(h) Waveform encoding
(i) Modulation
(j) Transmission

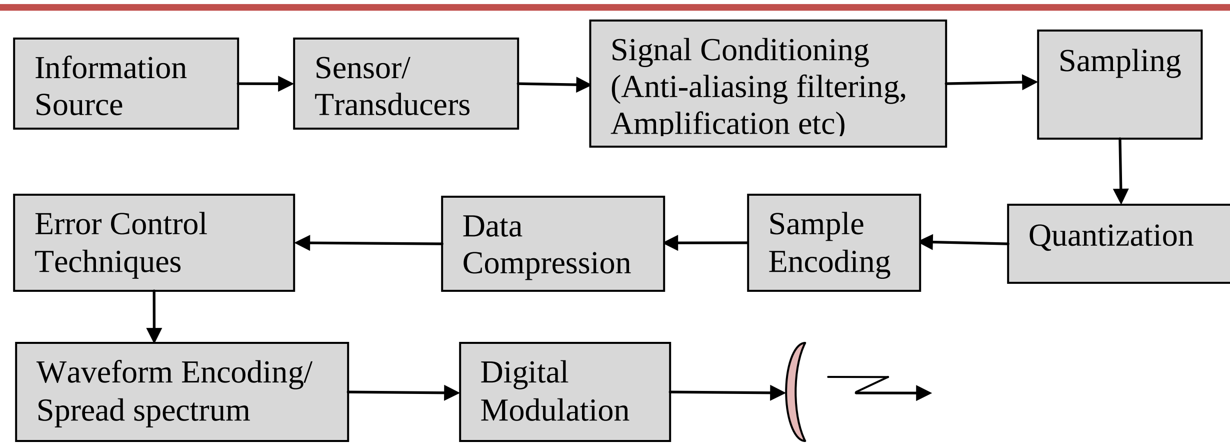


Fig.1: Digital Communication Transmitter System

The various blocks of a receiver are

(a) Antenna
(b) Band pass filter (Tuner)
(c) Demodulator
(d) Low pass filter
(e) Bit recovery system
(f) Error decoding
(g) Frame alignment
(h) Signal reconstruction

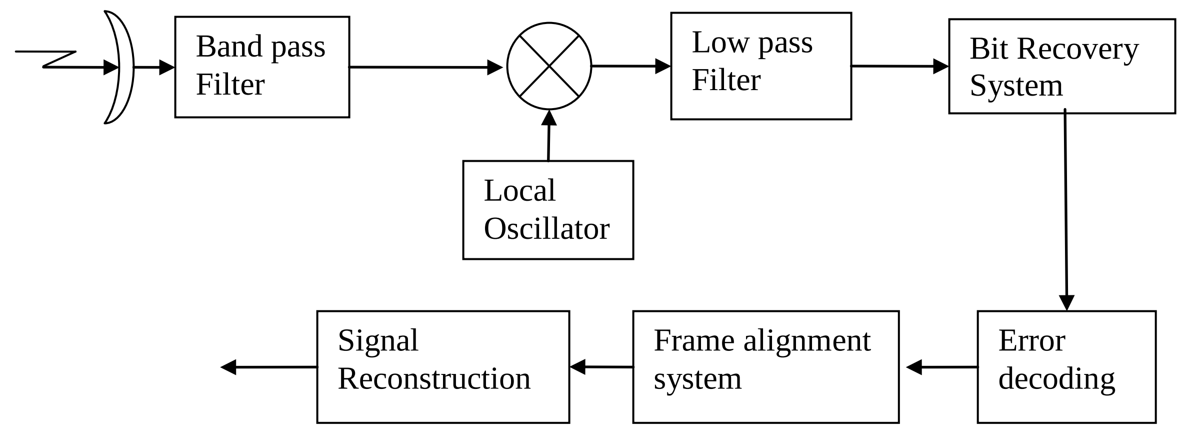


Fig.2: Digital Communication Receiver

Signal: A signal is a waveform which varies randomly with time (Fig.3). The information carried by signal is embedded in its characteristic features.

<u>Characteristic features of signal:</u>

An analog signal is random in nature. The information content of any signal is embedded in

(a) Amplitude Spectrum or power Spectrum

(b) Phase spectrum

A signal consists of number of frequency components having different amplitudes and phases. The relative amplitude (or power) of various spectral components of a signal is called Amplitude (or Power) Spectrum and Relative phase of various spectral components is called phase spectrum. For faithful reproduction of the signal both, (i) Amplitude (Power) Spectrum and (ii) Phase spectrum have to be preserved during transmission. The recovered signal is distorted if any change or loss takes place in the amplitude (Power) spectrum or phase spectrum or both during transmission.

Bandwidth of a signal: A random signal has infinite frequency components and energy of the signal is relatively distributed among these frequency components. Fourier transform X(f) of time domain signal x(t) gives the frequency spectrum of a signal. The range of frequencies i.e. $f_h - f_l$ is known as

bandwidth of a signal where f_h and f_l are highest and lowest frequency components respectively present in the spectrum of a signal.
Actually bandwidth of any random signal is infinite. However in communication application the bandwidth of the signal is restricted to those frequency components which carry 99% of the total signal power.

Band limited Signal: When a signal is passed through a filter, some of the frequency components are filtered out. The resulting signal is known as band limited signal.

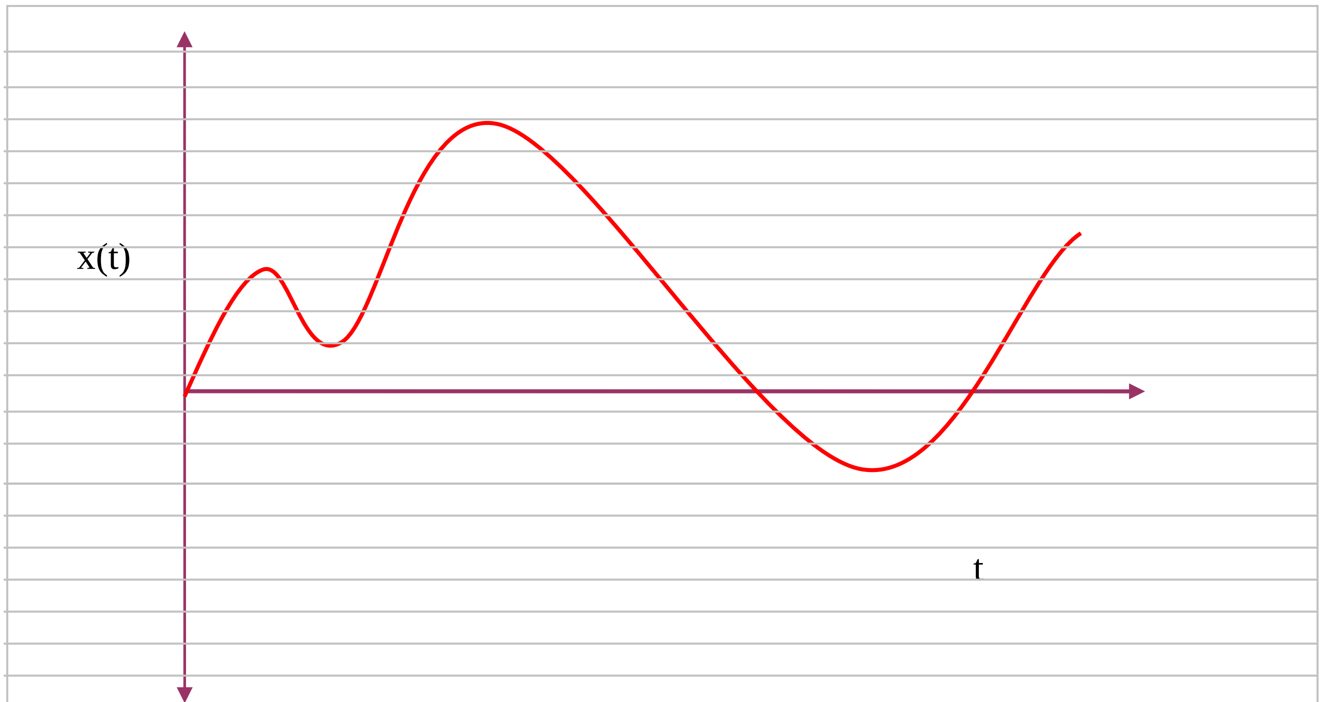


Fig 3: A time varying signal

Power spectral density of signal:
The total power or energy of a signal is spread over entire bandwidth of the signal. The relative power distribution among various spectral components of a signal is called power spectral density function. It is denoted by G(f) or G(ω). In Communication applications two sided power spectral density is considered. The characteristics of power spectral density function of a signal are;

1. It is an even function of f (or ω)
2. It is a positive function of f

3. The average power carried by a signal is given by

$$p_{av} = \int_{-\infty}^{\infty} G(f)df$$

4. The power spectral density function and auto correlation function for a Wide Sense Stationery-Random Process constitute Fourier transform pair.

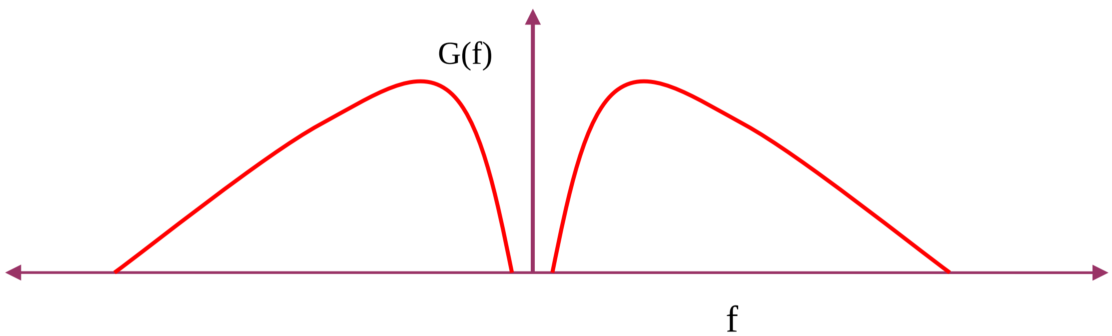


Fig 4: Two sided Power spectral Density of x(t)

Sampling:

It is the process of observing the amplitude of analog signal x(t) at discrete instants of time. The observed instantaneous amplitude value of signal at that instant of time is called sample. The time between two successive samples is called sampling period T. Sampling period T can be variable or fixed. Usually it is fixed. The inverse of T is called sampling rate F_s. The sampling converts a continuous time signal into a discrete time signal.

Sampling operation is defined by equation

$$x(n) = x(t)\ s(t)$$

Where x(n) is sampled signal

x(t) is analog signal and

s(t) is the sampling waveform

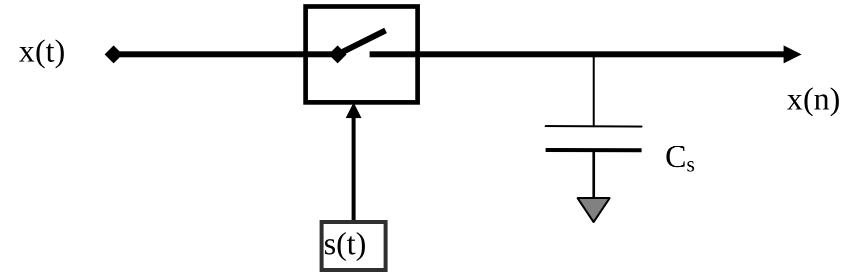


Fig.5 : Block Schematic of sampling process

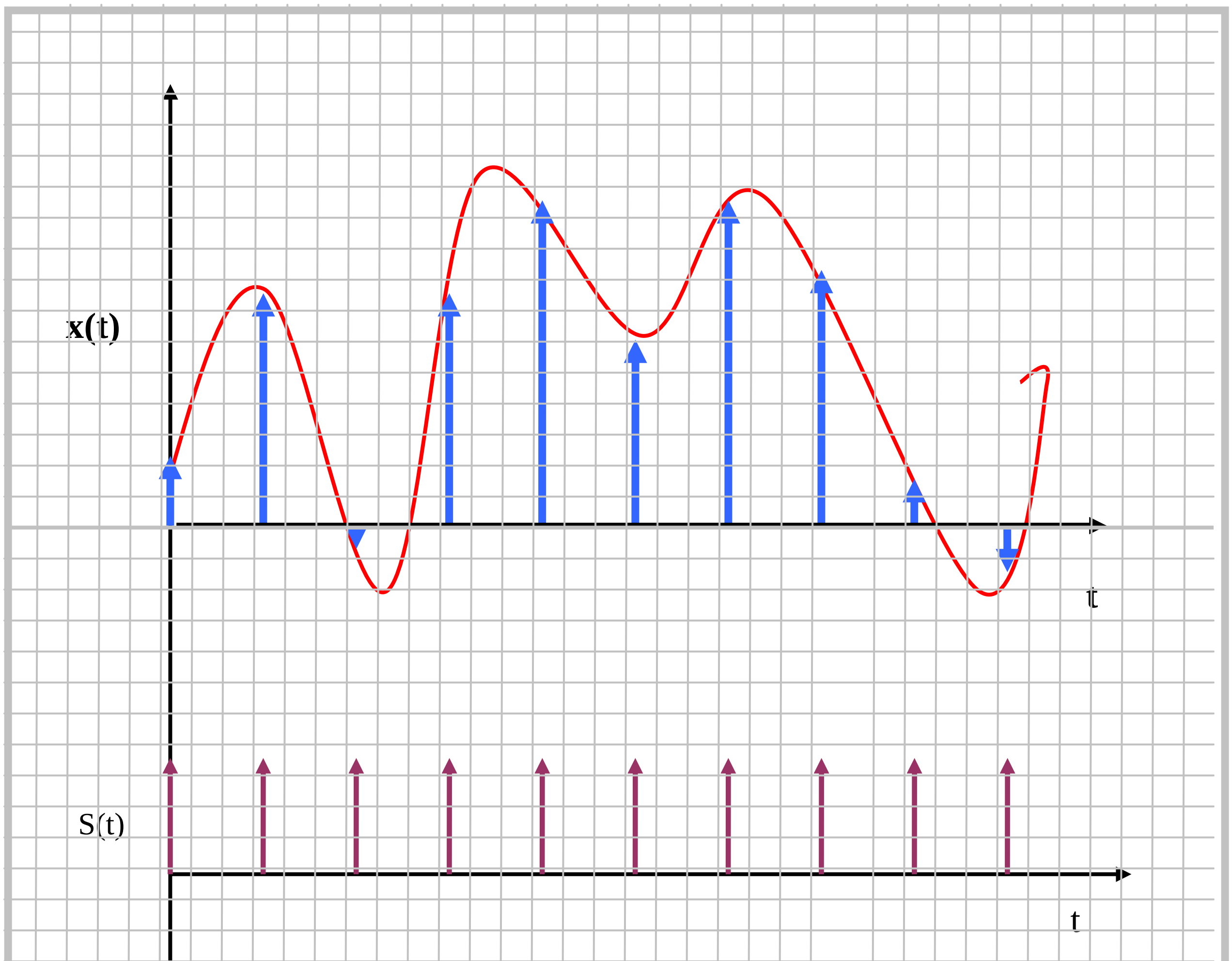


Fig.6 : Sampling operation

Sampling Theorem:

The theorem is due to Nyquist and is known a Nyquist sampling theorem. It puts a minimum limit on the sampling rate of a band pass signal. It states that in order to preserve the information and recover the signal faithfully from its samples, the sampling rate should be at least twice the bandwidth of the the analog signal.

Mathematically $F_s \geq 2$ B

Where B is the Bandwidth of the signal

For base-band signal B = f_h

where f_h is the highest frequency component of the base-band signal

Thus for base-band signals $F_s \geq f_h$

Proof: Mathematically the sampling operation can be expressed as

$$x(n) = x(t)s(t)$$

Where x(n) is sampled signal

x(t) is analog signal and

s(t) is the sampling waveform

s(t) is a periodic impulse train and it can be expressed in Fourier series as

$$s(t) = \frac{1}{T_s}(1 + 2\cos 2\pi n f_s t) = \frac{1}{T_s}\sum_{n=-\infty}^{\infty} e^{j2\pi n f_s t}$$

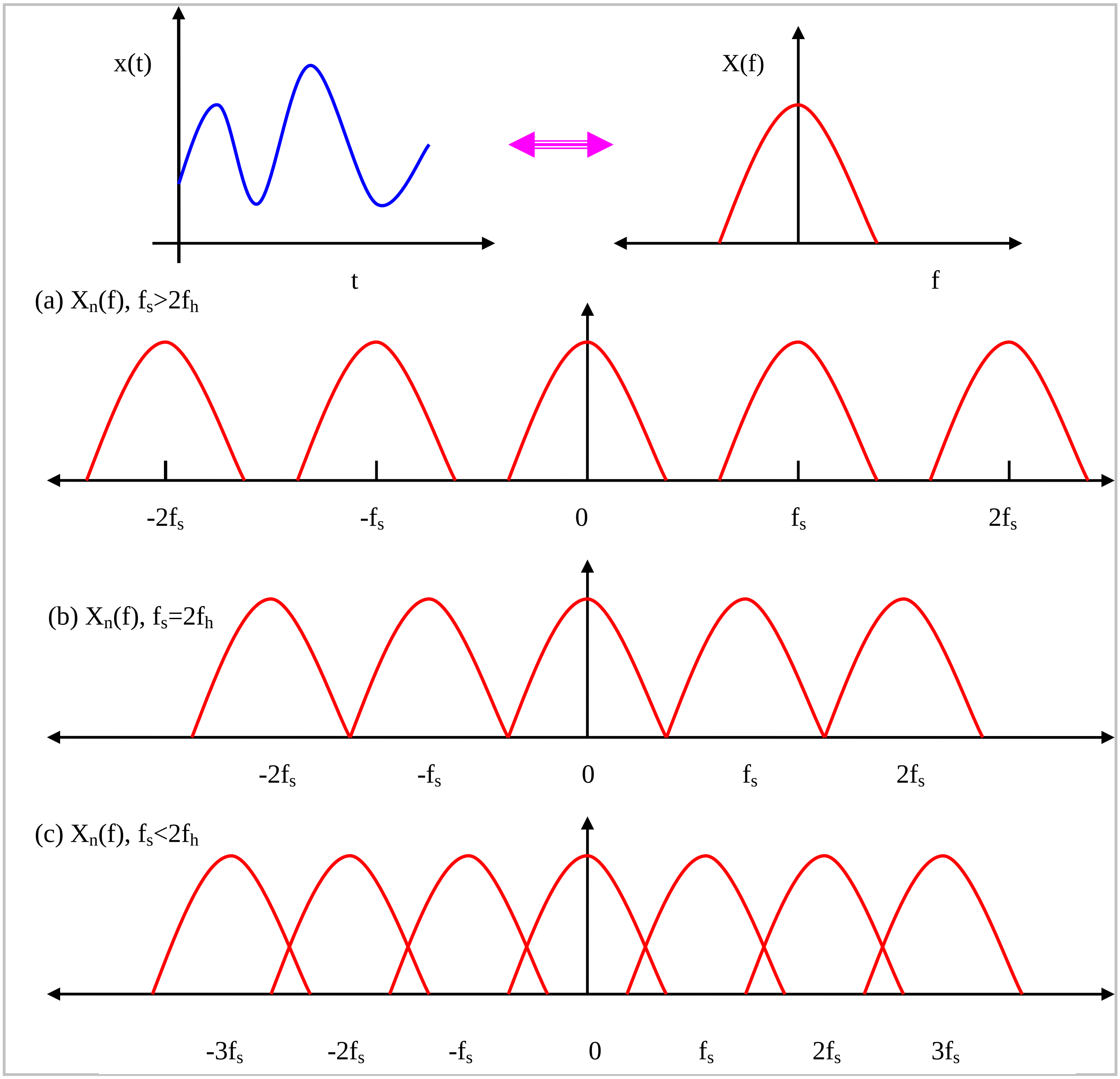
Thus in frequency domain x(n) can be expressed as

$$X_n(f) = \frac{1}{T_s}\sum_{n=-\infty}^{\infty} X(f - nf_s)$$

Where X_n(f) is Fourier transform of x(n)

X(f) is Fourier transform of x(t)

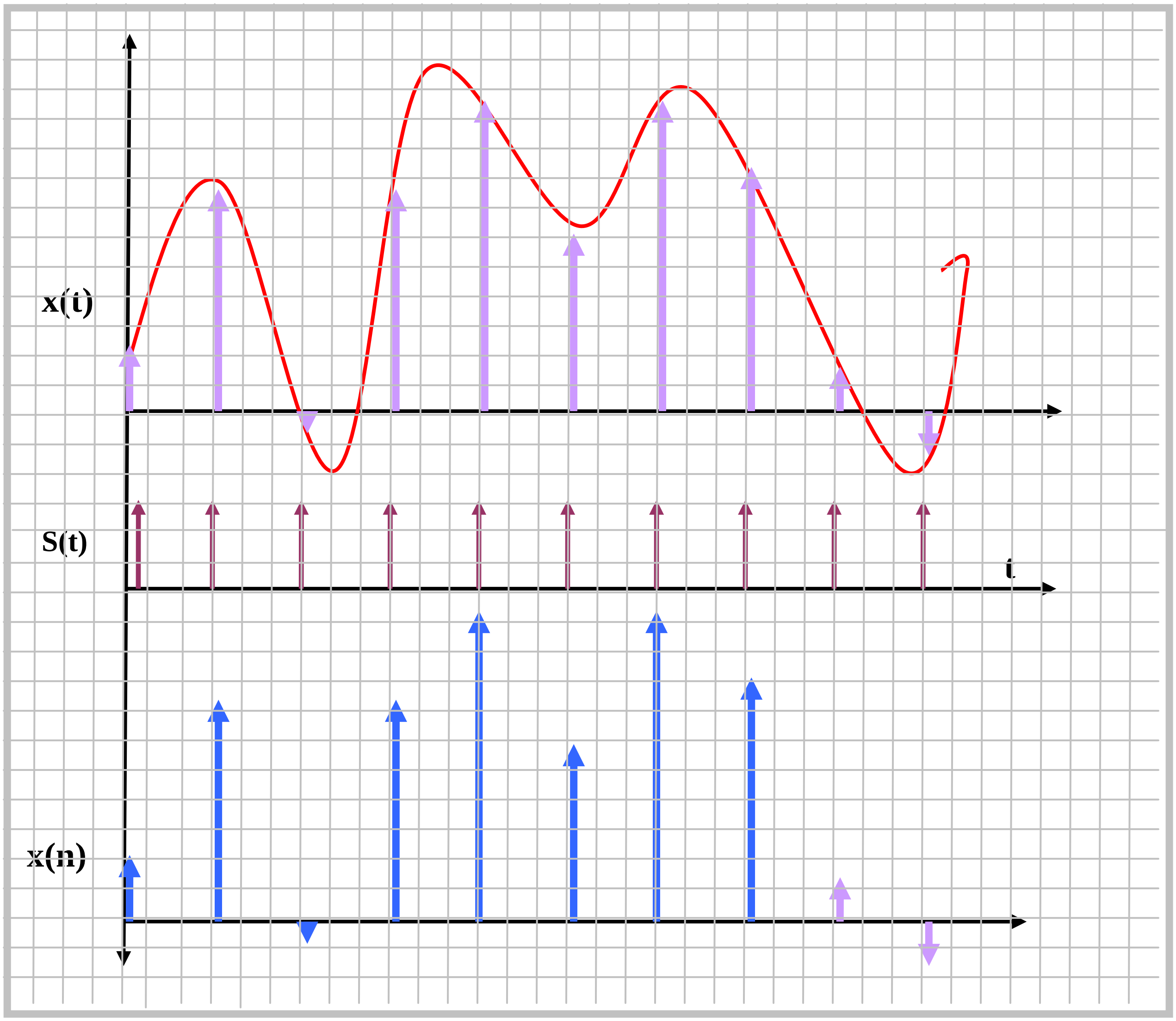
Following figure explains the effect of sampling in frequency domain under different conditions of sampling rate f_s.



x(t)
t
X(f)
f
(a) $X_n(f)$, $f_s>2f_h$
$-2f_s$
$-f_s$
0
f_s
$2f_s$
(b) $X_n(f)$, $f_s=2f_h$
$-2f_s$
$-f_s$
0
f_s
$2f_s$
(c) $X_n(f)$, $f_s<2f_h$
$-3f_s$
$-2f_s$
$-f_s$
0
f_s
$2f_s$
$3f_s$

Quantization:

It is the process of discretizing the amplitude of the signal samples. The peak-to-peak signal amplitude range is divided into number of levels which are called quantization levels. The amplitude difference between two successive levels is called step size. The step size can be uniform or non-uniform. During quantization process amplitude of the each sample is approximated to the nearest quantization level which is represented in binary form. In actual practice amplitude value of all those samples whose amplitude ranges between (k-½)S to (k+½)S are reprsented as Ks. If x(i) is the ith sample and has amplitude Ks. Then binary equivalent of K represents the sample value.



Sample x(n)≡x(nT)	Actual sample value in terms of no of steps s	Quantized Value	Binary Representation	Remarks
x(0)	+2.33	2	00010	-S is Step Size -MSB represents Sign; 0 = +ve and 1 = -ve
x(1)	+7.60	8	01000	
x(2)	-0.50	-1	10001	
x(3)	+7.60	8	01000	
x(4)	+10.5	11	01011	
x(5)	+6.33	6	00110	
x(6)	+10.5	11	01011	
x(7)	+8.30	8	01000	
x(8)	+1.60	2	00010	
x(9)	-1.30	-1	10001	

Fig: Pulse Code Modulation

Quantization Error and Noise:

During the process of quantization the amplitude of the signal sample is approximated to the nearest quantization level. This process introduces an error in the signal value and this error is known as quantization error. The distortion introduced in the signal due to quantization error is called quantization noise. If n_q is quantization error and S is step size then

$$n_q = \varepsilon \qquad \text{where } -S/2 \le \varepsilon \le S/2$$
$$= 0 \qquad \text{else where}$$

Assuming that the error ε is uniformly distributed over the interval [-S/2, S/2] then quantization noise N_q is given by

$$N_q = E[\varepsilon^2] = \int_{-\frac{s}{2}}^{\frac{s}{2}} \frac{1}{S}(\varepsilon^2)d\varepsilon = \frac{1}{S}\int_{-\frac{s}{2}}^{\frac{s}{2}} \varepsilon^2 d\varepsilon = \frac{S^2}{12}$$

Thus N_q can be reduced by reducing the step size S. However step size S can not be reduced indefinitely as this will result in increase in the code length and hence data rate shall increase.

Signal-to-noise ratio due to quantization:

Three types of SNR can be defined

(i) Peak signal power to Quantization Noise ratio

(ii) RMS signal power to quantization noise ratio

(iii) Average signal power to quantization noise ratio

(a) Peak signal power to quantization noise ratio:

If V_q is the peak signal value then $(SNQ)_q$ is defined as

$$(SNQ)_q = \frac{V_p{}^2}{N_q} = \frac{\left(\frac{MS}{2}\right)^2}{\frac{S^2}{12}} = 3M^2$$

Since $M=2^N$ where N is the no of bits in the equivalent binary code.

$$(SNQ)_q = 3.2^{2N} = 10\log_{10}(3.2^{2N}) = 10\log_{10}3 + 10\log_{10}(2^{2N})$$
$$(SNQ)_q = 6N + K$$

(b) RMS signal power to quantization noise:

If V is peak signal value then RMS signal value then (SNR)q is defined as

$$(SNQ)_q = \frac{\frac{V^2}{2}}{N_q} = \frac{\frac{1}{2}\left(\frac{MS}{2}\right)^2}{\frac{S^2}{12}} = 1.5M^2 = 1.5(2^{2N})$$

$$(SNQ)_q = 10\log_{10}(1.5\ 2^{2N}) = 10l0g_{10}(1.5) + 10\log_{10}(2^{2N}) = 6N + k$$

(c) Average Signal Power to quantization noise:

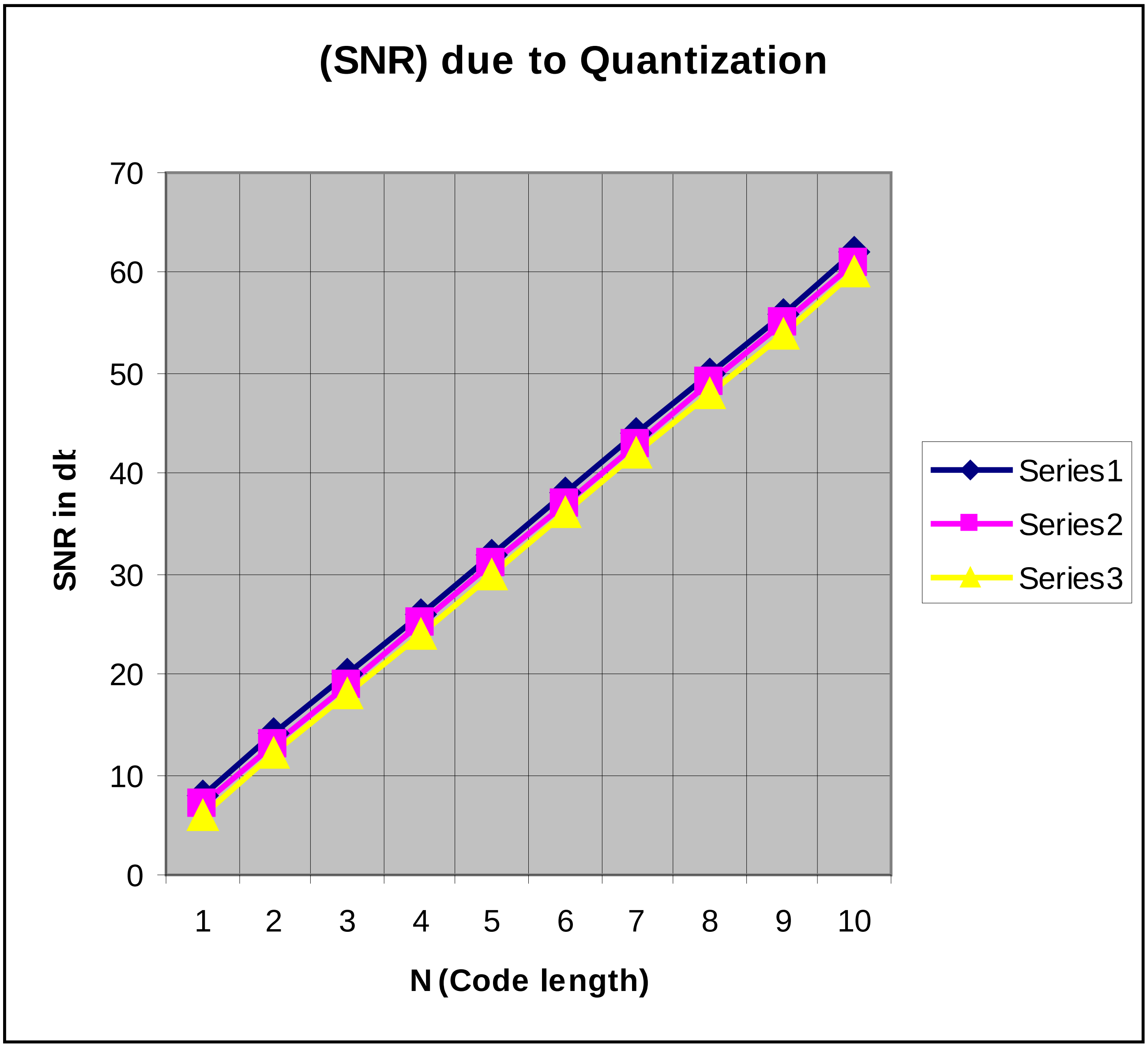
Assuming all quantization levels are equally likely then average signal power P is given by

$$p = \frac{2}{M}\left\{\left(\frac{S}{2}\right)^2 + \left(\frac{3S}{2}\right)^2 + \left(\frac{5S}{2}\right)^2 + \dots\dots\dots\dots + \left(\frac{(M-1)S}{2}\right)^2\right\} = \left(\frac{(M^2-1)S^2}{12}\right)$$

Therefore $$(SNR)_q = \frac{\frac{(M^2-1)S^2}{12}}{\frac{S^2}{12}} = (M^2-1) = (2^{2N}-1)$$

$$(SNR)_q = 10\log_{10}(2^{2N}-1) = 10\log_{10}(2^{2N}) = 6N$$

Thus in all the three cases $(SNR)_q$ in dbs is directly proportional to N where N is the width of code word. If code length is increased by one bit the $(SNR)_q$ improves by 6dbs.



Exercise for practice

(1) A base band signal has frequency range from 10Hz to 1350Hz and peak amplitude 4.096V. (a) What should be the sampling rate? (b) The code length of the quantized sample is 10 bits Calculate the total number of quantization levels used, step size and quantization noise. (c) What is the SNR due to quantization.

(2) In exercise 1 Calculate the number of quantization levels required so that the quantization noise is improved to 96dBs. Also calculate the step size.

Program Exercise: An analog signal x(t)= 0.2Cos 100πt + 0.5Sin 300πt + 0.4Sin500πt + 0.1 Cos600π. (a) Plot x(t). (b) What should by the Nyquist sampling rate (c) If sampling rate used is 20% more than Nyquiste rate, plot the x(n)

Waveform Coding Techniques

The techniques of representing an analog signal by a sequence of numbers are called waveform coding techniques. The most common technique used is Pulse Code Modulation (PCM). This is described now.

Pulse Code Modulation (PCM):

In PCM an analog signal is first conditioned, then sampled slightly at a higher rate than Nyquist rate. The signal samples are quantized and the amplitudes of the quantized signal samples are represented as binary numbers called codes. The block diagram showing various sub-system of PCM system has been given below (Fig.1)

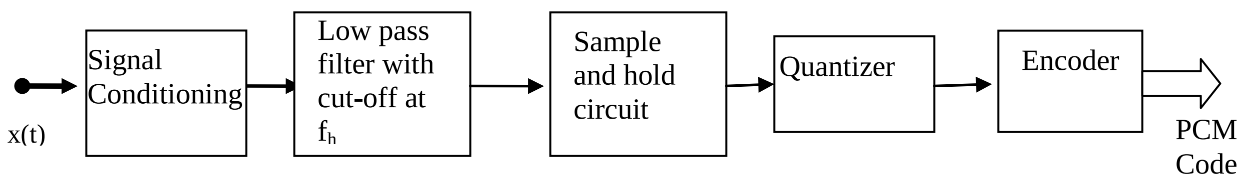


Fig. 1: Elements of a Basic PCM System

The analog signal x(t) is first preprocessed, it is then bandlimited to f_h Hz. The signal is then sampled and samples are fed to quantizer. The quantized signal samples are encoded into binary code with the help of an encoder. In practical systems quantizer and coder are together replaced by an ADC. Fig. 2 shows a practical PCM generator.

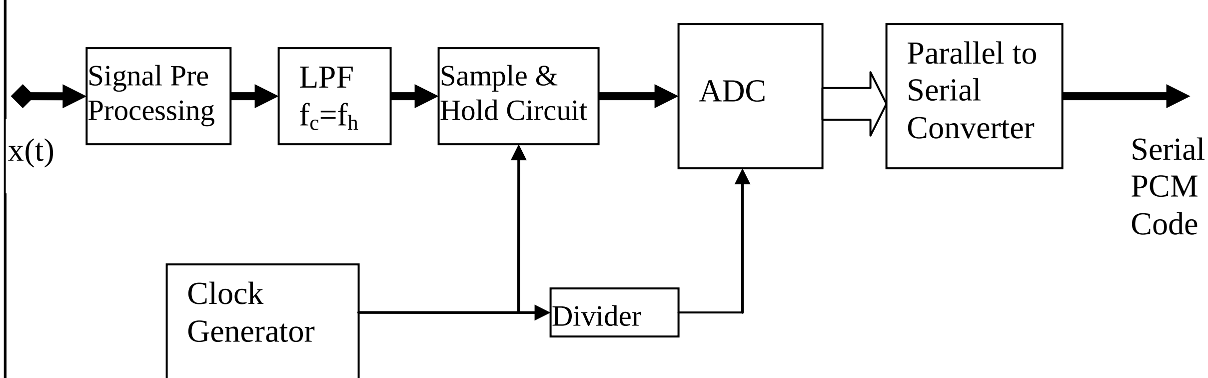
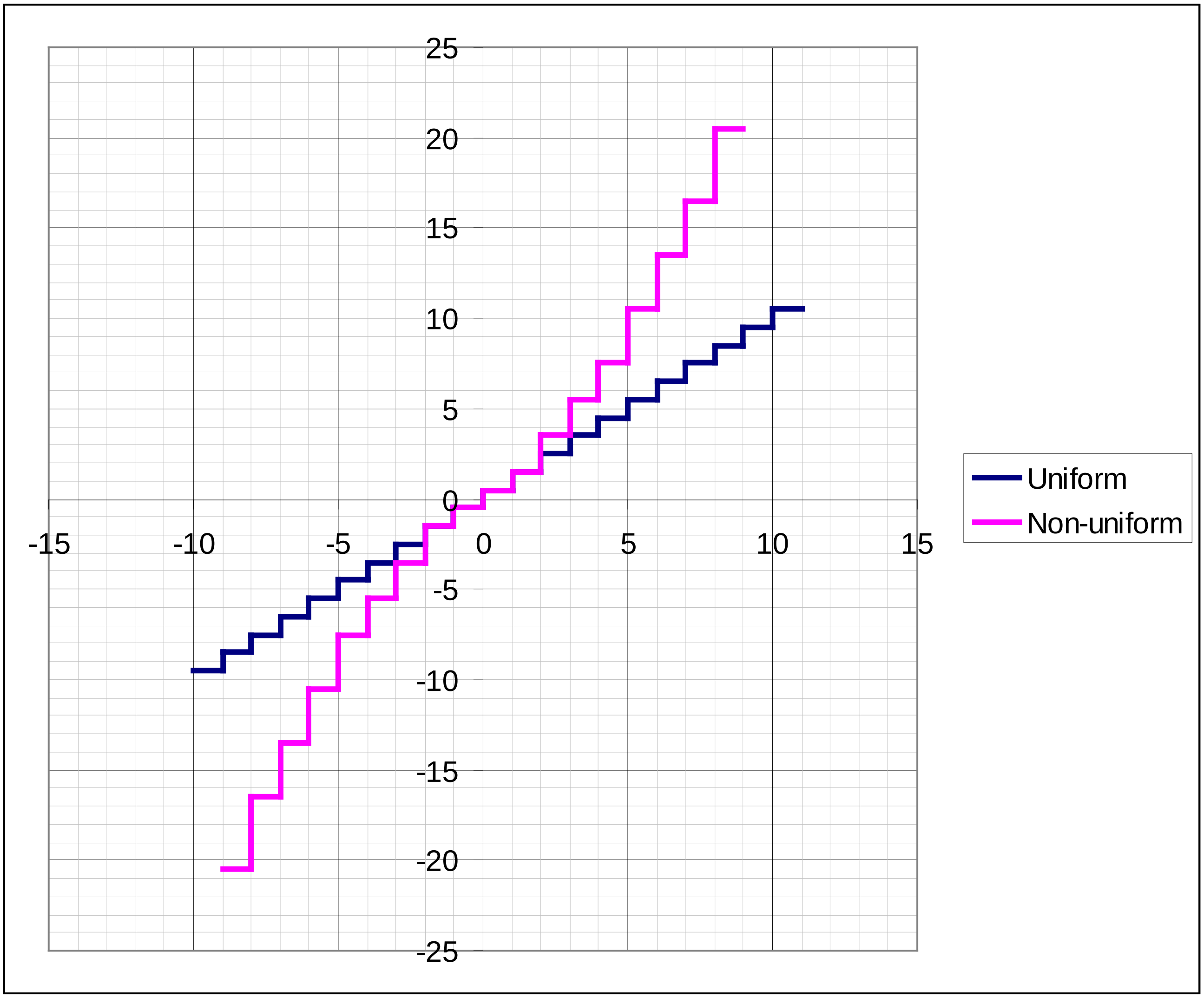


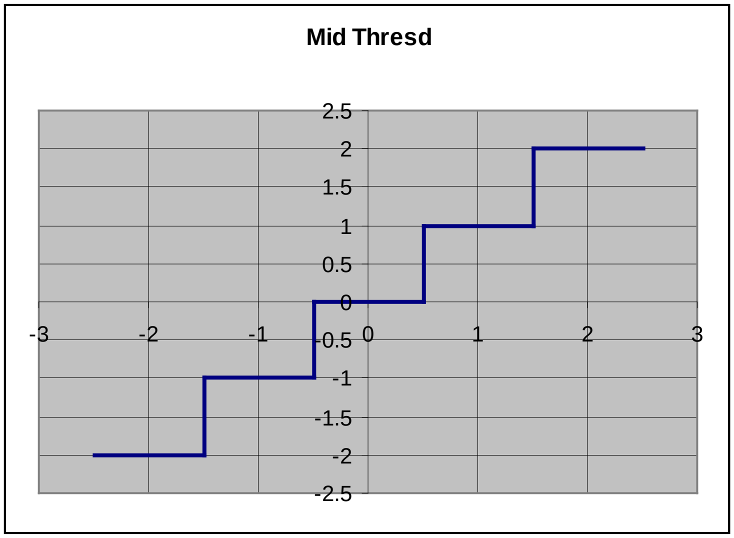
Fig.2: A Practical PCM Generator

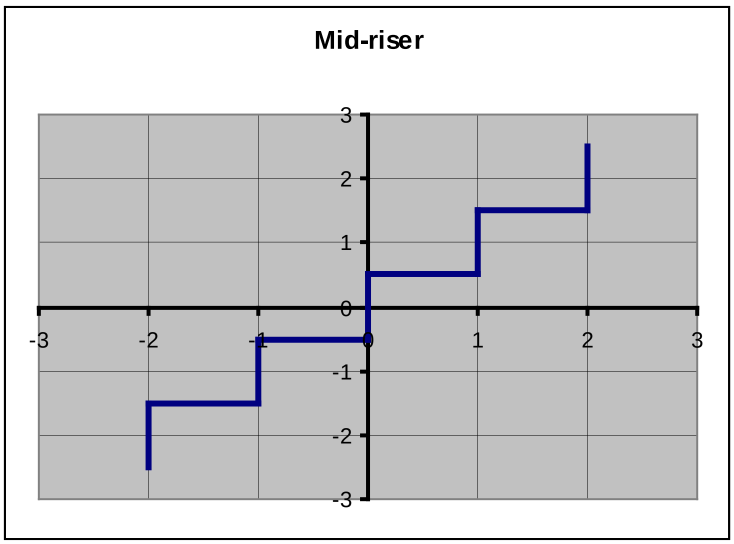
Companding of Speech Signals:

The statistical dynamic range of speech signal used for telephone communication is very large. To cover entire dynamic range 12 bit code is required. This shall result in a data rate of 96Kbps per speech signal. To reduce the bit rate of the coded signal non-uniform quantization is used for speech signals. In non-uniform quantization the step size between the successive quantized levels increases from lower signal values to higher signal values.



In practice Signal is compressed at transmitter end and after compression uniform quantization is carried out. To overcome the distortion, the signal is expanded back on the receiver side. The process of signal compression at transmitter and expansion at receiver is known as companding of speech signals. The characteristics of the expander are inverse of the characteristics of the compressor.





The compressor characteristics chosen are logarithmic and are such that for low signal values no compression takes place but magnitude of compression increases progressively for higher values of the signal.

Two laws are being followed world wide as standards for companding speech signal in telephone systems.

(1) μ- Law:

This is followed in USA, Canada and Japan.

This law uses mid thread characteristic logarithmic compander for encoding.

According to this law the input signal is compressed according to the compression

$$|V| = \frac{\log(1+\mu\left(\frac{|x(t)|}{x_{\max}}\right))}{Log(1+\mu)}$$

Where V is output signal value and x(t) is input signal value.

x_{max} is maximum value of input signal

μ is constant and its value has been taken as 255.

μ = 0 refers to linear quantization

(2) A-Law:

This is followed in Europe and rest of the world. This law follows mid-rise characteristic logarithmic compander for encoding.

According to this law the compressor characteristics are given by

$$|V| = \begin{cases} \dfrac{A\left(\dfrac{|x(t)|}{x_{\max}}\right)}{1+\log A} \ldots\ldots\ldots \quad for \quad 0 \le |x(t)| \le \dfrac{1}{A} \end{cases}$$

$$|V| = \begin{cases} \dfrac{1+\log\left(\dfrac{|x(t)|}{x_{\max}}\right)}{1+\log A} \ldots\ldots \quad for \quad \dfrac{1}{A} \le |x(t)| \le 1 \end{cases}$$

The value of A has been chosen 87.56 as standard

<u>Compression through code Converter</u>

The signal samples are coded into 12 bit code using uniform quantization. This 12 bit code is applied as address to the code converter (ROM). The output of code converter is 8 bit. The mapping from 12 bit to 8 bit is as shown

+1024	Step Size = 32ΔV	Every 32 successive input code words one code word is generated	512	16
+512	Step Size = 16ΔV	Every 16 successive input code words one code word is generated	256	16
+256	Step Size = 8ΔV	Every 8 successive input code words one code word is generated	128	16
+128	Step Size = 4ΔV	Every 4 successive input code words one code word is generated	64	16
+64	Step Size = 2ΔV	Every 2 successive input code words one code word is generated	32	16
+32	Step Size = ΔV	One-to-one correspondence between input and output	32	32
-32	Step Size = ΔV	One-to-one correspondence between input and output	-32	32
-64	Step Size = 2ΔV	Every 2 successive input code words one code word is generated	32	16
-128	Step Size = 4ΔV	Every 4 successive input code words one code word is generated	64	16
-256	Step Size =8ΔV	Every 8 successive input code words one code word is generated	128	16
-512	Step Size = 16ΔV	Every 16 successive input code words one code word is generated	256	16
-1024	Step Size = 32ΔV	Every 32 successive input code words one code word is generated	512	16

Time Division Multiplexing:

In PCM voice signal is first band limited to 3400Hz and it is sampled at 8000 samples per second and each sample is encoded as 8 bit PCM code. Thus 8 bits of voice data is generated every 125µsec. If this data is transmitted in fraction of this time then the idle period of communication channel can be utilized for transmission of data from other voice channel. This is known as time division multiplexing (TDM). Here time available on channel is divided among number of signals. A TDM frame is formed.

One Frame period of 125 µsec				
Source 1	Source 2	Source 3	-------	Source n
8 bit	8 bit	8 bit		8 bit

Requirements of TDM system:

(1) Frame length 125 μs
(2) Frame synchronization signal: Required to identify the individual signals at the receiver
(3) Signaling information for conveying call initiation, call termination, address of called party and other supervisory and control operations.

Two TDM frame formats have been accepted as standards.

AT&T T1 carrier

-This is a standard frame format used in USA.
-24 PCM voice channels are multiplexed together to constitute a frame
-1 bit is added extra with each frame for frame synchronization
-12bit framing code is used for frame synchronization. Its format is 110111001000
-Framing code is repeated every 12 frames i.e. after every 1.5ms
-193 bits (24x8 + 1) per frame of 125 μs are transmitted
-Data rate is 1.544MBPS
-signaling information for supervisory operations like call initiation, call termination, address of called party etc is transmitted during the 8 bit period of each voice code after every six frames. This type of signaling is called Channel-associated signaling.
-Signaling data rate per voice channel per second is 8000/6=1333

One Frame period of 125 μsec					
	Voice 1	Voice 2	Voice 1	------	Voice 24
F					

CCITT DS1 TDM Frame format

-This standard is followed in Europe and rest of the world.
-30 PCM voice channels are multiplexed together to constitute a frame
-8 bits are added extra with each frame for frame synchronization
-8 bit framing code has format is 00100111
-Framing code is repeated with every frame
-8 bits are added for signaling information with every frame.
- Thus the CCITT frame has 32 channels i.e. 30voice +1 sync + 1 Signaling
- Data rate is (32x8) 256bits per frame of 125μs i.e. 2.048 MBPS
-Sync signal is transmitted 8000 times per sec and signaling data rate is 64KBPS

CCITT Frame Format

One Frame period of 125 µs						
SYNCHRONIZATION	**Voice 1**	**Voice 2**	**---------**	**Signaling Channel**	**---------**	**Voice 30**
0 0 1 0 0 1 1 1			---------		---------	

Comparison between T1 and DS1 TDM frames

S.No	Parameters	T1	DS1
1	Total no of channels	24	32
2	No. of Voice channels	24	30
3	Frame sync information	1 bit per frame	8 bits per frame; 1 dedicated channel
4	Sync code length	12 bits	8 bits
5	Sync repetition rate	Once every 1.5 ms	Once every 125 µs
6	Signaling bits	1 bit per voice channel after every six fames	8 bits per frame; 1 dedicated channel
7	Signaling data rate	32 KBPS	64KBPS
8	Signaling type	Channel associated	Separate channel
9	No of bits per frame	193	256
10	Data rate	1.544MBPS	2.048MBPS
11	Sync performance	Takes longer time	fast

TDM/PCM Digital Hierarchy

The TDM multiplexing of voice signals into a higher bit rate signal as discussed above is called first level of Digital hierarchy. In telecommunication systems world over first level multiplexed signals are again multiplexed into second level and second level multiplexed signals are multiplexed into third level and so on. Following table gives the details of two main TDM/PCM digital hierarchies followed world over

Digital hierarchy level	AT&T Standard (USA, Canada and Japan)	CCITT standard (Europe and rest of the world)
DS-0	64KBPS Single voice channel	64 KBPS Single voice channel
DS-1	24 voice channels Data rate 1.544 MBPS	30 voice channel Data rate 2.048MBPS
DS-II	96 Voice channels 4 DS-1 signals are multiplexed Data rate 6.312 MBPS	120 Voice channel 4 level-1 signals are multiplexed Data rate 8.448 MBPS
DS-III	672 voice channels 7 level-II signals are multiplexed Data rate 44.736MBPS	480 voice channels 4 level-II signals are multiplexed Data rate 34.368MBPS
DS-IV	4032 voice channels 6 level-III signals are multiplexed Data rate 274.176MBPS	1920 voice channels 4 level-III signals are multiplexed Data rate 139.364MBPS
DS-V	8064 voice channels 2 level-IV signals are multiplexed Data rate 560.160MBPS	7680 voice channels 4 level-IV signals are multiplexed Data rate 565.148MBPS

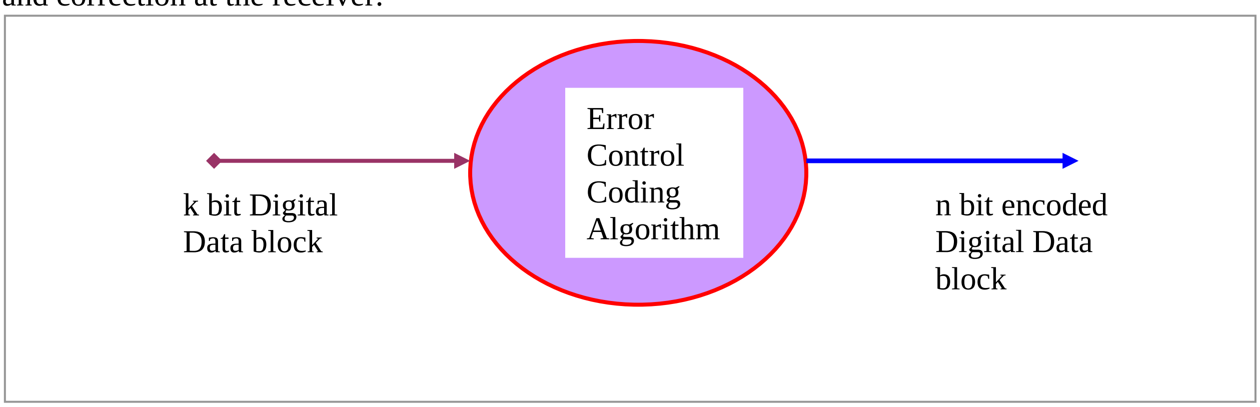
Errors in Digital Data Communication

Errors occur in the digital transmission due to number of reasons like
- (i) Additive White Gaussian Noise (AWGN)
- (ii) Impulse noise
- (iii) Inter Symbol Interference (ISI)
- (iv) Cross talk.

Errors are of two types
- **(i) Isolated or Single bit errors**
  A single bit is in error in a group of bits. These errors are caused by AWGN and ISI.
- **(ii) Burst errors**
  A string of bits are corrupted. These errors occur due to impulse noise or cross talk. In a burst error it is not necessary that all bits should be in error, instead a burst error of k bits is said to have occurred if only first and last bits of the burst are received in error.

Two main schemes are being used in digital communication to control the errors.
1. (1) Forward Error Detection and Correction (FEC) Scheme
2. (2) Error Detection followed by Retransmission

Forward Error correction Scheme is used when channel is simplex or transmission is in real time and there is no possibility of retransmission e.g. public broadcasting
Error Detection followed by Retransmission Scheme is used in all data communication networks where transmission is off line and retransmissions can be employed.
In both the schemes error control coding is applied which helps in error detection at the receiver and also in error correction in some times. In error control coding, additional parity redundancy check bits are appended with each data block and these parity check bits help in error detection and correction at the receiver.



Properties of Error Control Codes

Two Important characteristics of an error control code are

(i) Hamming Distance (d_m)

(ii) Code Rate (Rc)

Hamming Distance (d_m)

The number of bit positions at which two code words differ is known as distance (d_{ij}) between two code words C_i and C_j.

Hamming distance d_m = Min (d_{ij}) where $i \neq j$, $0 \leq I \leq (L-1)$, $0 \leq j \leq (L-1)$.

It gives a measure of the error detection and error correcting capability of an error control code.

The error detection capability of a code is (d_m-1)

The error correcting capability of the code is ½(d_m-1)

Code Rate (R_c):

It gives the measure of data transmission efficiency of an error control code. It is given by

$$R_c = \frac{k}{n} \times 100\%$$

Where k: number of message bits in the code word

n: code length(message bits + check bits)

Types of Error Control Codes

Various types of error control codes are

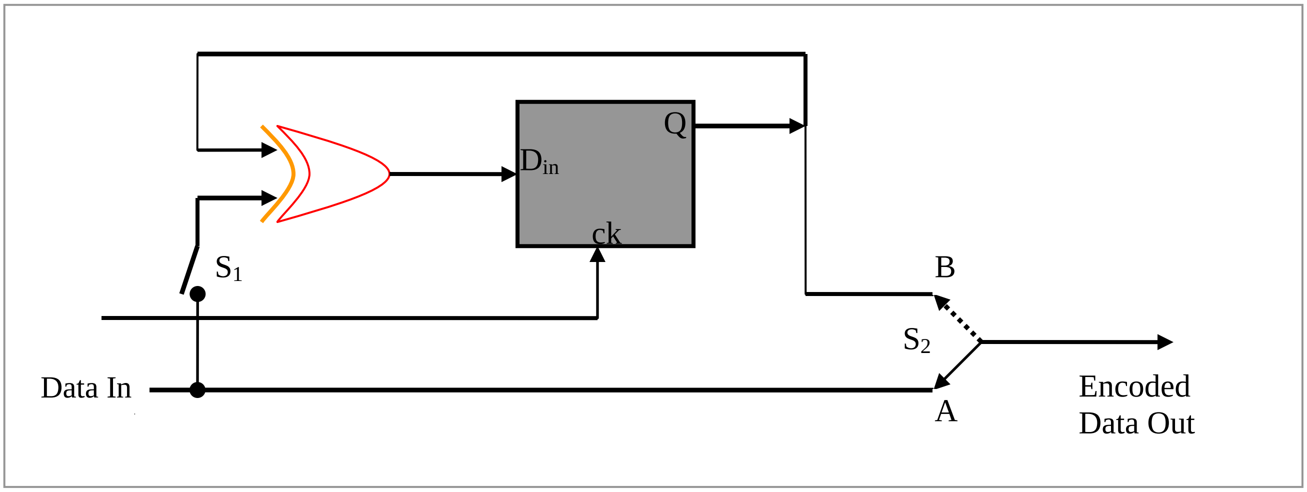
(1) Single bit Parity Check code
(2) Polynomial codes.
(3) Repeated Codes
(4) Block Codes
(5) Burst error correction using Hamming code
(6) Convolution Codes

Single Bit Parity Check Code

This code is useful for detection of errors. An extra bit is appended with the k bit data sequence to make the number of 1s in the data block even or odd. Accordingly it is known as even or odd parity check. Usually even parity is used.

This code is used while storing data on digital storage media like floppy disks, hard disks, CD-ROMs, DVD etc

The single bit parity check code detects all odd bit errors but fails to detect if even number of bits are in error. The code does not possess any error correcting capability. The circuit diagram for generating single bit parity check code has been shown.



Polynomial Error Detection:

This is also known as Cyclic Redundancy Check (CRC).

The technique is used for error detection in data and computer communication network.

In polynomial technique each message sequence is treated as a message polynomial M(x) of degree k

$$M(x) = \sum_{i=o}^{k} a_{n-i} x^{k-i}$$

Where a_i's either 1 or 0

There is a generator polynomial G(x) of order r (r = n-k).

$$G(x) = \sum_{j=0}^{r} g_{r-j} x^{r-j}$$

The G(x) is known both to the transmitter and the receiver.

At the transmitter end G(x) is used to generate transmission polynomial T(x) from M(x) as follows

(i) First r zeros are appended to M(x) to get a modified message polynomial $M^*(x)$.

$$M^*(x) = M(x)x^{n-k}$$

(ii) The modified M(x) is divided by G(x) using modulo-2 division.

(iii) A remainder polynomial R(x) is obtained.

(iv) The transmitted polynomial T(x) is obtained by adding R(x) to $M^*(x)$.

(v) T(x) is transmitted.

(vi) The receiver receives $T^*(x)$. At the receiver $T^*(x)$ is divided by G(x) using modulo-2 division. If remainder is zero, then there are no errors in the $T^*(x)$. Last r bits are discarded and remaining bits are accepted as error free.

(vii) If after division at the receiver remainder is non-zero, then there are errors in the received message. It is rejected and retransmission is requested.

The modulo-2 division at receiver and at the transmitter is easily carried out by using following hardware arrangement. It consists of shift register stages connected through Ex-OR gates as shown in the figure. In the figure g represents a connection corresponding to generator polynomial. If a term is absent then there is no connection.

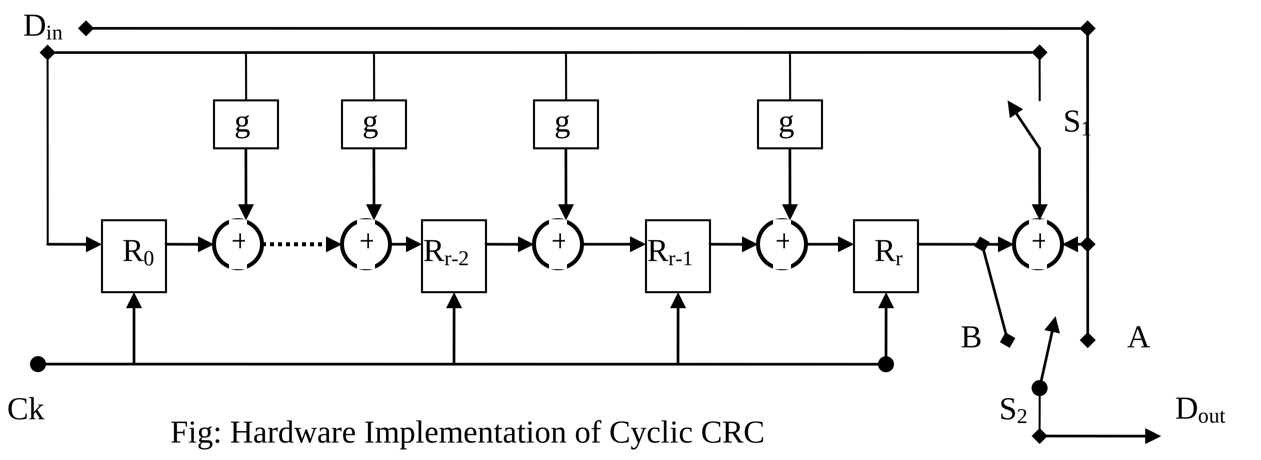


Fig: Hardware Implementation of Cyclic CRC

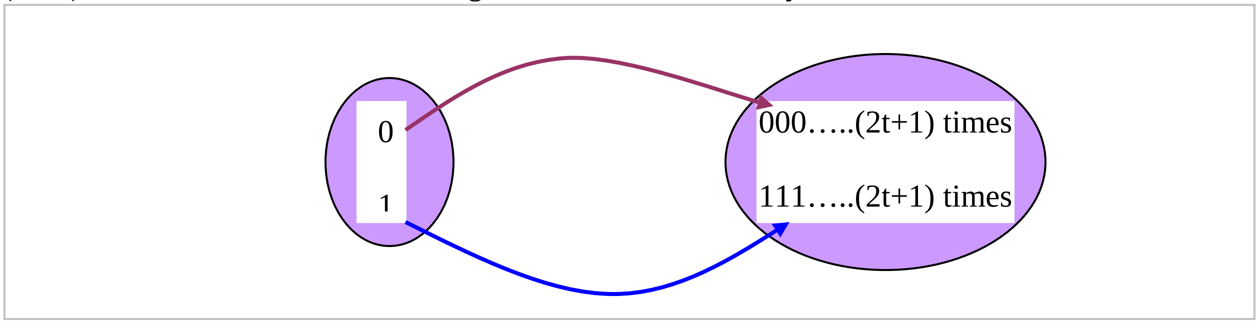
The operation of the hardware for computing CRC is now explained.
Initially shift register is cleared.
Switch S_1 is closed and switch S_2 is placed to position A.
Data is supplied. It is simultaneously taken out and also circulated through the shift register.
Once the data block is over, switch S_1 is opened and switch S_2 is placed to position B and then contents of shift register are taken out immediately after data.

Standard Important CRC generator Polynomials are

CRC-12	$X^{12} + X^{11} + X^3 + X^2 + X + 1$
CRC-16	$X^{16} + X^{15} + X^2 + 1$
CRC-CCITT	$X^{16} + X^{12} + X^5 + 1$
CRC-32	$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

Repeated Code

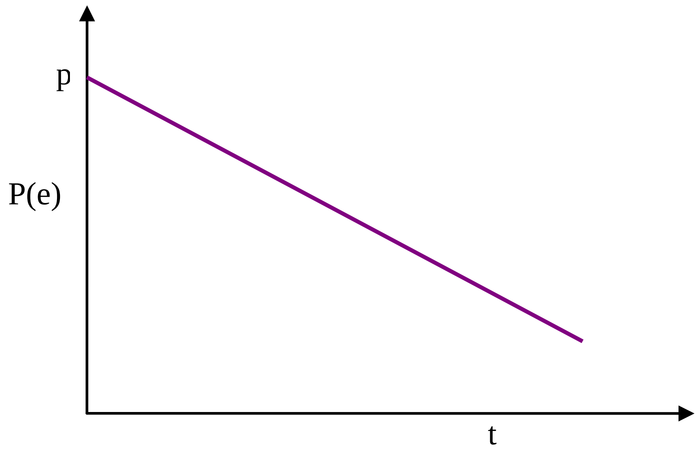
It is an error correcting code. In this code each data bit '0' or '1' is encoded as s sequence of (2t+1) similar bits, where t is an integer. The code set has only two code words



The code can be used for error detection and correction. This code is capable of correcting t errors in a code word.

Assuming that t is the number of bit errors that has to be corrected then code word length is (2t+1) bits. If p is the probability that a bit is received in error then P(e) the probability of receiving code word in error is given by

$$P(e) = \sum_{i=t+1}^{2t+1} C(2t+1, i)\, p^{i} (1-p)^{2t+1-i}$$



The disadvantage of the repeated code is that it is highly inefficient. The effective code rate is

$$R_c = \frac{1}{2t+1} \times 100\%$$

Block Codes

These are also called (n, k) codes.
These are used for error detection and Correction
Each message of k bits is encoded into an n bit code where n > k.
(n-k) parity bits are appended to the message of k bits.
The commonly used block codes used for error detection/correction are the Hamming Codes.
These codes have the capability of correcting 1 bit error and these can detect two bit errors.
The various Hamming codes are (3,1), (7,4), (15,11), (31,26) and so on.

The general formula for all Hamming codes is $(2^i-1,\ 2^i-i-1)$

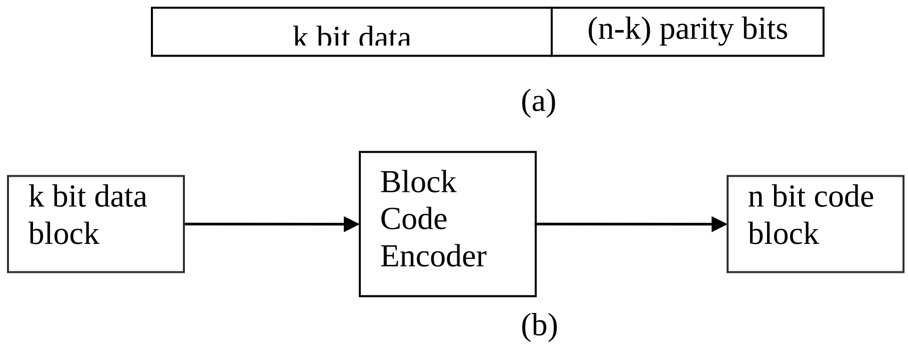


Fig: (a) Format of Block Code (b) Generation

Generation of Hamming codes

- The bit positions in the code word of n bits are numbered from 1 to n
- Bit positions 1, 2, 4, 8, 16….. i.e. positions which are power of 2 are reserved for check bits and remaining bits are filled with k bits of message.
- As an example consider the generation of (7,4) code. Here bit positions 1, 2 and 4 are reserved for parity check bits p_0, p_1, and p_2 bits and remaining positions viz., 3, 5, 6 and 7 are filled with 4 message bits m_0, m_1, m_2 and m_3 respectively.

7	6	5	4	3	2	1
m_3	m_2	m_1	p_2	m_0	p_1	p_0

- Parity check bits are generated from message bits using ex-or operation as follows
    - o $P_0 = m_0 + m_1 + m_3$
    - o $P_1 = m_0 + m_2 + m_3$
    - o $P_2 = m_1 + m_2 + m_3$
- The 7 bit code is then transmitted.
- At the receiver following relations are used for check
    - o $P_0 + m_0 + m_1 + m_3$
    - o $P_1 + m_0 + m_2 + m_3$
    - o $P_2 + m_1 + m_2 + m_3$
- If result is 000 , the code word is correct, no correction is needed
- If one parity bit shows error, then that bit is alone inverted in the received code word
- If two parity bits are in error, then the message bit common to both is inverted.
- If all three parity bits show error , bit m_3 is inverted.
- After performing the check and correction parity bits are discarded and message bits accepted

Mathematical Treatment of Block codes

A general method of generating the hamming code is now discussed.
Assuming B_j is the jth data block of k bits. The corresponding Code word is obtained as

$$C_j = B_j\, G$$

Where C_j = Code word n bit long. It is row matrix of order (1x n)
B_j = Data word of k bits long. It is expressed as a row matrix of order (1x k)
G = It is a Generator matrix of order (k x n)
Modulo-2 addition is performed while computing C_j.

$$G = [\ I_k : C\ ]$$

Where I_k is an identity matrix of order (k x k)
C is parity check matrix of order **(**k x (n-k) **)**

The generated code word C_j is transmitted. It is received at the receiver as R_j. The validity of the received code vector is checked by computing error syndrome S as follows

$$S = R_j\, H^T$$

Where S is error syndrome row matrix of order (1 x (n-k))
R_j is received code vector
H^T is transpose of H matrix and is given as

$$H = [\ C^T : I_{n-k}\ ]$$

Where C^T is transpose of C matrix
I_{n-k} is identity matrix of order {(n-k) x(n-k) }
Modulo-2 addition is performed while computing S.

If error syndrome S = 0 , the received code vector has no errors. Parity check bits are discarded and remaining bits are accepted as valid data
If error syndrome S ≠ 0, the received code vector R_j contains errors. Error correction is attempted. Afterwards parity bits are discarded and remaining bits are accepted as valid data.
The above procedure is described for (7, 4) Hamming code.
The parity check matrix of (7, 4) code is as

$$C = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

Therefore $$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

If $$B_j = \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix}$$

Then $$C_j = \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

The code word C_j = 0 1 1 0 0 1 1 is transmitted. It is received at the receiver as R_j.
At the receiver R_j is multiplied with H^T.

Where
$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Two Cases arises here

(a) Received code vector R_j has no errors i.e $R_j \equiv C_j$

$$S = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$$

The S is zero. Parity bits are discarded and remaining bits i.e. 0 1 1 0 are accepted as valid data.

(b) Received code vector R_j has one bit in error. Let Rj= [0 1 1 1 0 1 1] i.e (Assuming C_2 in error). In this case

$$S = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}$$

The $S \neq 0$ implies that the received code vector R_j has errors. Parity bits P_1 and P_0 show error. These two bits are generated by message bit B_0. Thus by inverting the B_0 error is corrected. The corrected code vector is R^* = [0 1 1 0 0 1 1]

Single bit Error Correction using Parity Bits

- A single bit error can be corrected in a block of data bits using parity bit approach.
- The data block is stored into a two dimensional array row wise.
- Parity is computed along rows as well as along columns.
- The entire data block along with parity is transmitted and at receiver it is again stored into a two dimensional array.
- Parity check is performed row wise and column wise. If only one bit is in error then two parity bits one along row and one along column shows error. The bit can be corrected by inverting it.

m	m	m	m	**p**
m	m	m	m	**p**
m	m	m	m	**p**
m	m	m	m	**p**
p	**p**	**p**	**p**	**p**

Burst Error Correction using Hamming Code

- Hamming code can be used to correct burst errors of L bits long.
- At transmitter the data block is arranged into a (k x L) two dimensional array. Loading is done row wise
- (n-k) Parity bits are appended with each row and resulting array is (n x L)
- Data is transmitted column wise and at receiver it is loaded into an (n x L) memory column wise.
- Parity check is performed row wise and errors if any is corrected.
- In this scheme if during transmission an entire column of L bits gets corrupted, it is dispersed as single bit per row. The same can be corrected through Hamming code.

m	m	m	m	**p**	**p**	**p**
m	m	m	m	**p**	**p**	**p**
m	m	m	m	**p**	**p**	**p**
m	m	m	m	**p**	**p**	**p**
m	m	m	m	**p**	**p**	**p**
m	m	m	m	**p**	**p**	**p**
m	m	m	m	**p**	**p**	**p**
m	m	m	m	**p**	**p**	**p**

Convolution Codes

These are very powerful codes used for error correction.
These codes are suited for use in very noisy channels with high probability of error.
These codes are used for deep space communication applications.
Like block codes the Convolution codes are encoded by using shift register encoders
The convolution codes differ from block codes in structural form. In block codes a frame of k bits is taken and encoded into a code word of n bits long. Each data block is taken independently. In convolution code entire data stream, regardless of its length, is converted into a single code word.
The code rate of block codes is much higher than convolution codes.
In convolution code source data is segmented into frames of k bits per frame.
L+1 frames of data are encoded into an n bit code frame, where L is the memory of the shift register.
Each time a new data frame of k bits is shifted into the shift register while the old frame is shifted out. A new code word is computed and transmitted.
The convolution code is also known as (n, k, L) code; where
n is code length
k is data frame length
L is constraint length
For most of the convolution codes k=1

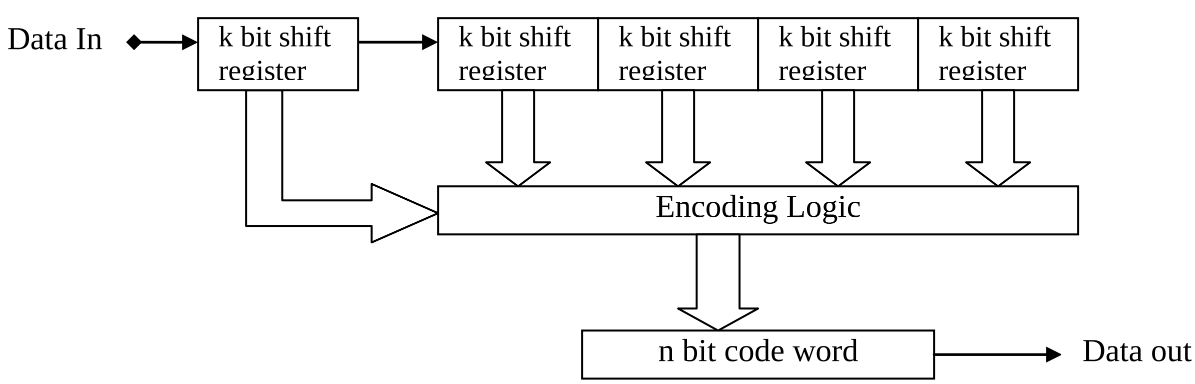


Fig: Hardware Schematic of Convolution

To illustrate the convolution code we assume n=3, k=1 and L=2. The corresponding convolution encoder is as shown.

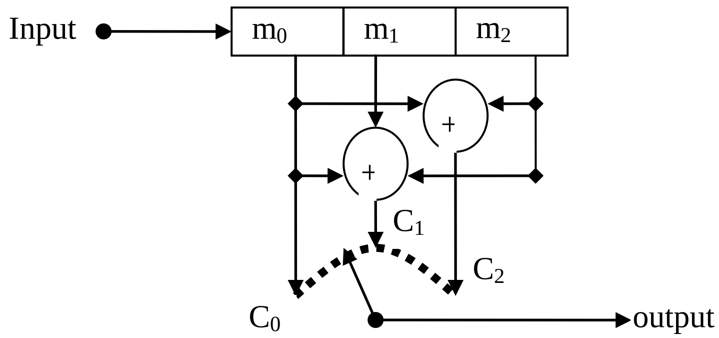


Fig: Showing the convolution encoder with rate 1/3

The three check bits C_0, C_1 and C_2 are generated as

$C_0 = m_0$

$C_1 = m_0 + m_2$

$C_2 = m_0 + m_1 + m_2$

The shift register is initially cleared
On bit of the data is shifted into the shift register
All the check bits C_0, C_1 and C_2 are sampled and transmitted
A second bit is shifted into register and again three bits C_0, C_1 and C_2 are transmitted
The process continues till complete data stream is transmitted.
All the bits are shifted out before feeding in next data stream.
Corresponding to each input bit 3 bits are transmitted, thus code rate is 1/3.
The operations of Convolution encoder can be described in three ways

(i) State Diagram
(ii) Tree Diagram
(iii) Trellis Diagram

<u>State Diagram:</u>

A convolution encoder belongs to a class of devices which are commonly known as is a ***finite state machine*** because it possesses the memory of the past inputs and its future state along with the outcome can be predicted with current input.

A state of a finite machine consists of the smallest information that together with the current input to the machine can predict the output of the machine. It provides the knowledge of the past inputs and the set of possible outputs.

A state diagram gives a compact way of completely characterizing of a convolution encoder. It depicts all the current states of the encoder and all the possible state transitions with the incoming input. It also shows the possible outcomes accompanying the state transition.

The state of the convolution encoder shown above is given by the information present in m_0m_1 stages of the shift register. Thus it can have four possible states 00, 01, 10 and 11 denoted as a, b, c and d respectively.

Transition from one state to another occurs depending on the next input. However from any current state there are only two possible state transitions depending on input state.

A solid line shows the transition from current state to next when input is at 0 state. A dashed lines shows the transition from current state to next when input is at 1 state.

A three bit output is also shown alongside the state transition line.

Table showing the state transitions and encoder outputs

S.No.	Current State	Current input	Future state	Encoder output
1	00	0	00	000
	00	1	10	111
2	01	0	00	011
	01	1	10	100
3	10	0	01	001
	10	1	11	110
4	11	0	01	010
	11	1	11	101

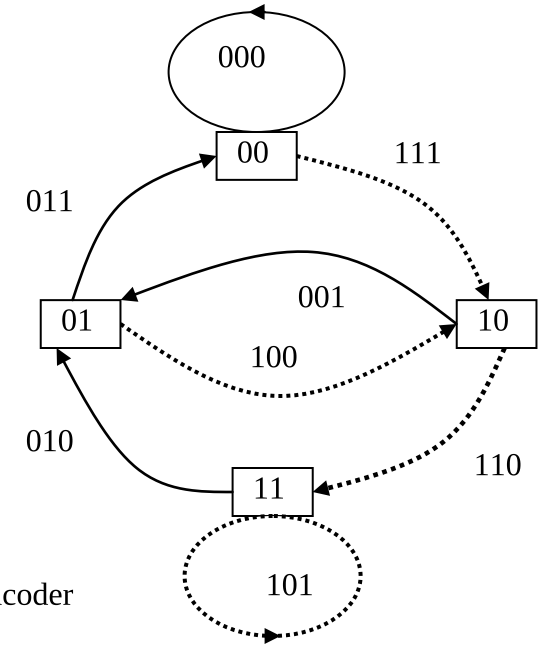


Fig: State diagram of the convolution encoder

Trellis Diagram

In the tree diagram the structure repeats itself after the third stage. This is because the three bit output sequence at each stage is determined by the input bit and the two previous input bits i.e. the four states a=00, b=01, c=10 and d=11 of the encoder described by m_0m_1 data.
The state of the encoder changes at the input of each input bit.
This can be depicted by trellis diagram where in solid line denotes the output when input bit is 0 and dashed line denotes the output when input bit is 1.

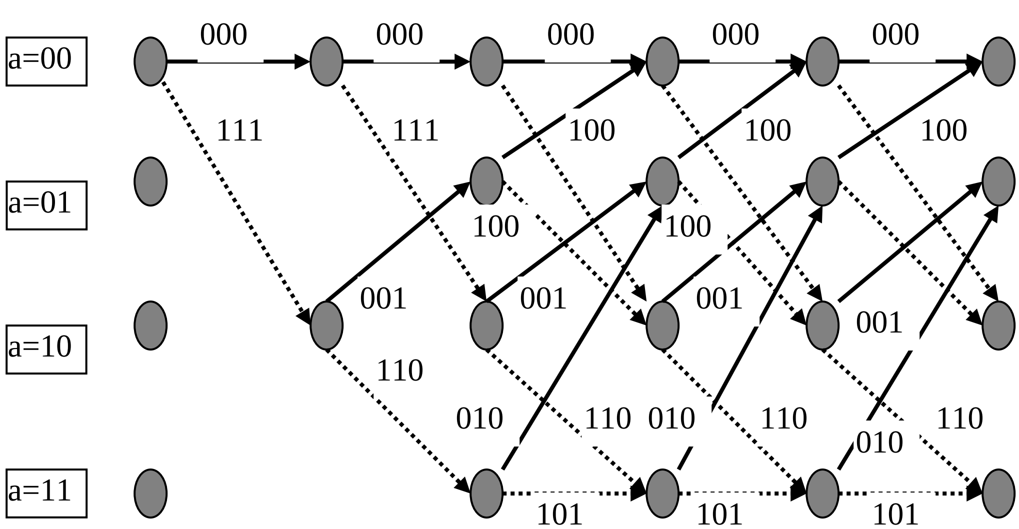


Fig: Trellis Diagram of Convolution encoder

Tree Diagram

A sate diagram can not be used for tracking the encoder transitions as a function of time. A tree diagram adds the dimension of time to the state diagram and can be conveniently used to track the encoder transitions with time.
The tree diagram of the encoder shown above is given below. At each successive input bit time the encoding procedure can be described by traversing the diagram from left to right, each tree branch describing an output branch word.
The branching rule for determining the code word sequence is as follows;
If the input bit is ‘0’, its associated branch word is found by moving to the next rightmost branch in the upward direction.
If the input bit is ‘1’, its associated branch word is found by moving to the next rightmost branch in the downward direction.
Assuming that the initial contents of the encoder are all zeros, the diagram shows that;

(i) if the first input bit is zero, the output branch word is 000
(ii) if the first input bit is 1, the output branch word is 111

(iii) if the first input is 0 and the second input bit is 0, the output branch word is again 000
(iv) if the first input is 0 and the second input bit is 1, the output branch word is again 111
(v) if the first input is 1 and the second input bit is 0, the output branch word is again 001
(vi) if the first input is 1 and the second input bit is 1, the output branch word is again 110

(vii) if first two bits are 00 and third bit is 0, the output branch word is 000
(viii) if first two bits are 00 and third bit is 1, the output branch word is 111
(ix) if first two bits are 01 and third bit is 0, the output branch word is 011
(x) if first two bits are 01 and third bit is 1, the output branch word is 100
(xi) if first two bits are 10 and third bit is 0, the output branch word is 001
(xii) if first two bits are 10 and third bit is 1, the output branch word is 110
(xiii) if first two bits are 11 and third bit is 0, the output branch word is 010
(xiv) if first two bits are 11 and third bit is 1, the output branch word is 101

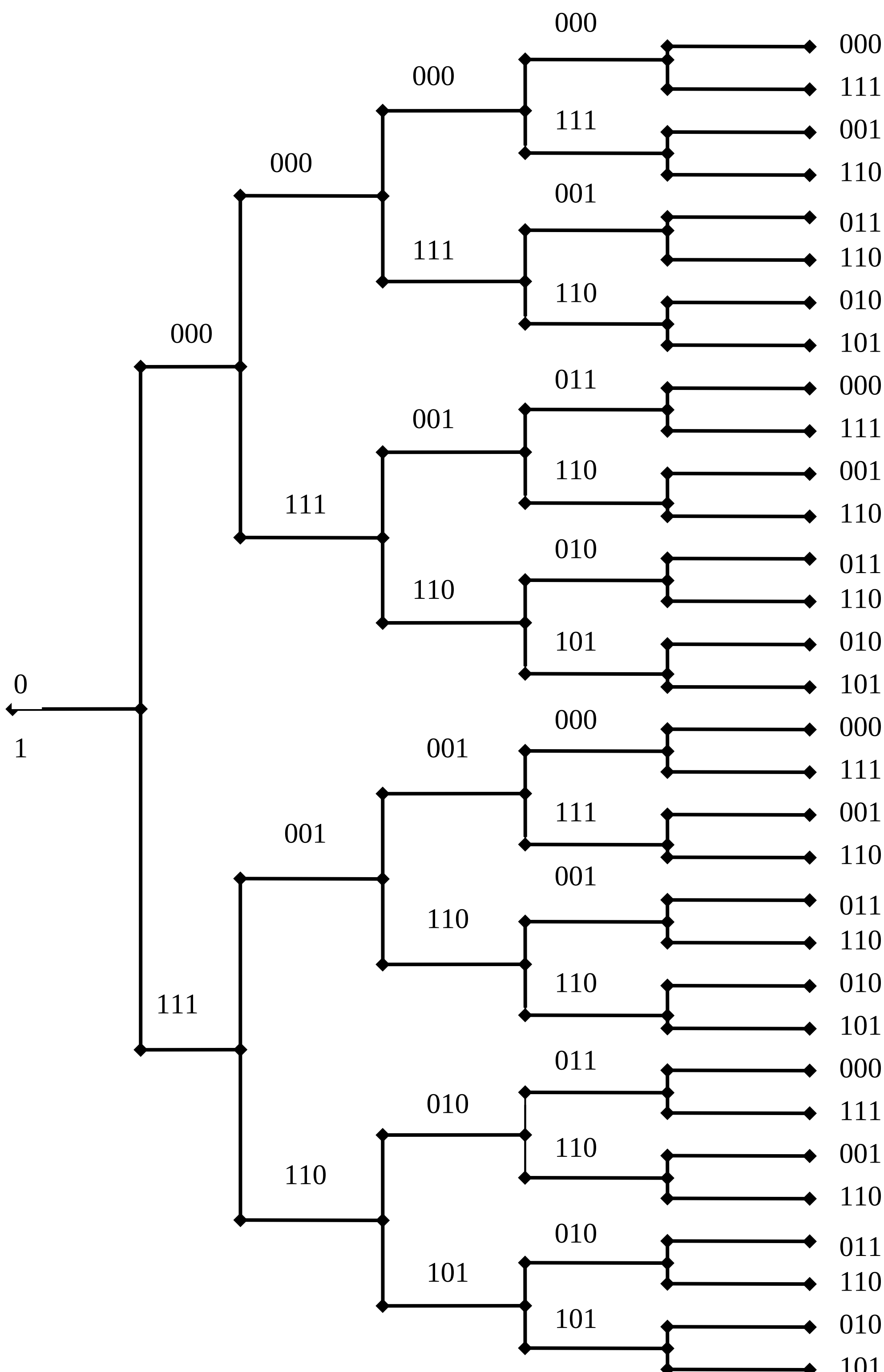


Fig: Tree Diagram of Convolution Encoder

Low BIT Rate Speech coding techniques

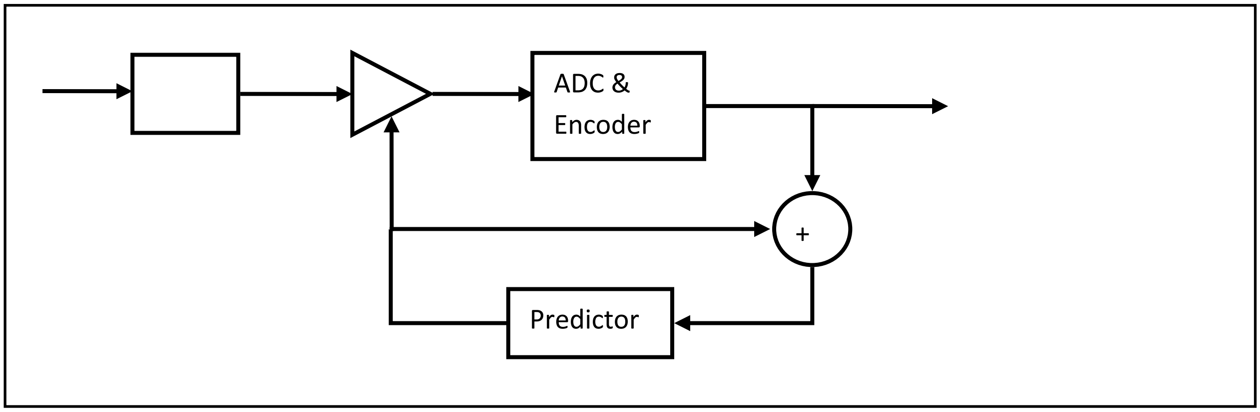
The telephone speech signal has baseband signal bandwidth 3400Hz. The sampling rate used is 8000sa/s. The conventional PCM technique of speech coding generates the data rate of 64Kbps per voice channel using logarithmic waveform companding. Since the sampling rate used is 8000sa/s and at this sampling rate the successive speech signal samples generated have very high correlation which results in lot of data redundancy in the output and hence unnecessary usage of excessive transmission channel resources. Various techniques of telephone speech signal coding have been proposed, investigated. These include

(i) Differential Pulse code modulation (DPCM)
(ii) Delta modulation (DM)
(iii) Adaptive delta modulation (ADM)
(iv) Adaptive Differential PCM(ADPCM)
(v) Sub band coding
(vi) Vector coding

Significant data rate reductions have been achieved through these techniques. For example GSM mobile standard uses a technique which generates a data rate of 13.78Kbps per voice channel only. And IS-95 USA mobile standard uses 7.8Kbps per voice channel. Some of the techniques listed above are discussed briefly here.

Differential Pulse Code Modulation (DPCM)

In PCM each signal sample is coded independently and the generated code is transmitted to the receiver where the original signal is reconstructed from these coded samples through process of reconstruction. In DPCM the difference e(t) between estimated sample value and the actual sample value is computed and coded. The binary code value proportional to the difference signal e(t) is transmitted to the receiver. The receiver constructs back the original sample by adding the received error signal code to the estimated sample value. From these reconstructed samples original speech signal is reconstructed back. The block diagram of the transmitter and the receiver is as shown.



In one of these techniques the estimated sample value is taken equivalent to the previous sample value. Thus it is the difference between current sample value and previous sample value which is coded and transmitted. At the receiver the current sample is constructed by adding received difference signal code to the immediate previous sample value. The predictor is a simple one sample memory element.

In order to have better estimate of the current sample value, number of previous samples are used for prediction. This helps to reduce the error signal e(t) magnitude and hence less bits can be used to encode it and hence data rate is reduced. The hardware element used is called a predictor. The predictors are of different orders. Higher the order of the predictor better is the estimate of current sample. But it increases the complexity of the system.

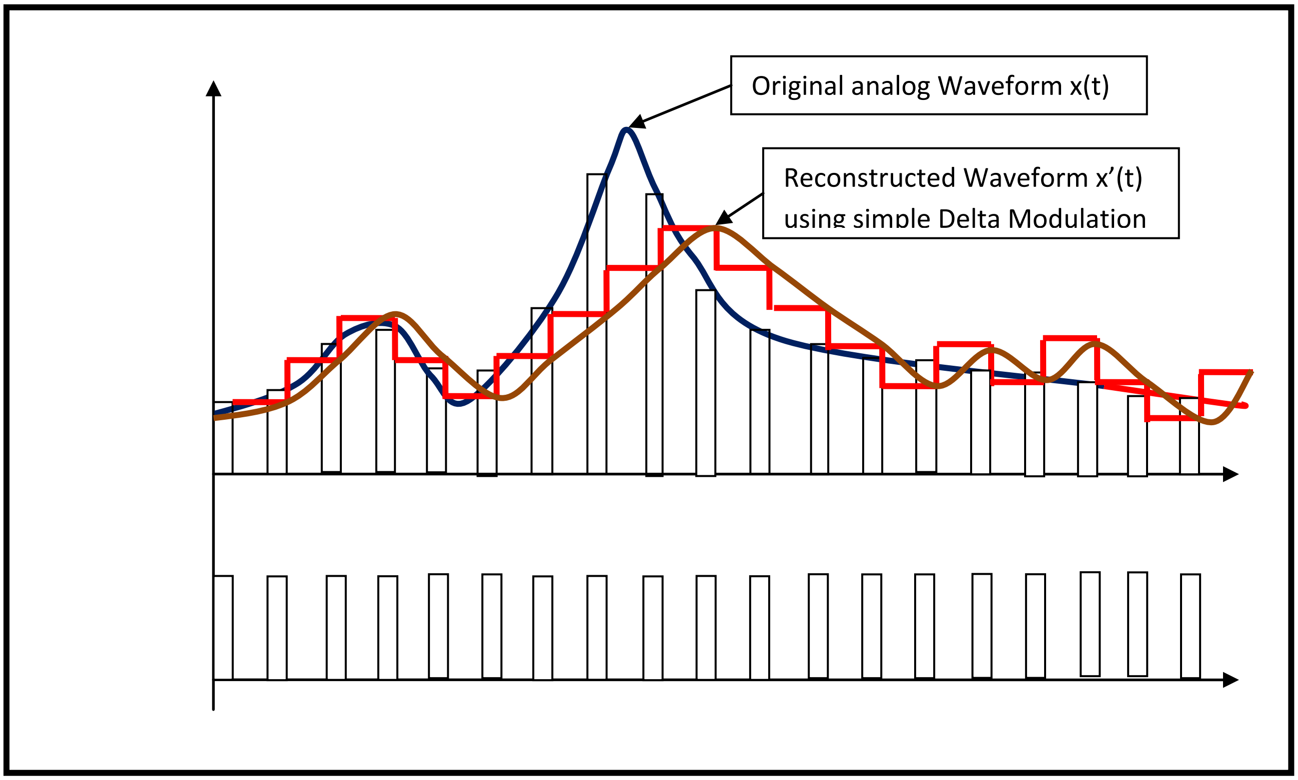
Delta Modulation (DM) : This is the simplest form of DPCM technique. In this technique only one bit is used to encode the difference between predicted sample value and the current sample value. Depending on the incoming bit status a constant quantity or step size (Δv) is either added or subtracted from the previous sample value at the receiver end. This technique essentially encodes each sample by a single bit, thereby date rate effectively becomes equal to sample rate. Delta modulation performs satisfactorily if the sampling rate is very high.

This technique is affected by following two disadvantages

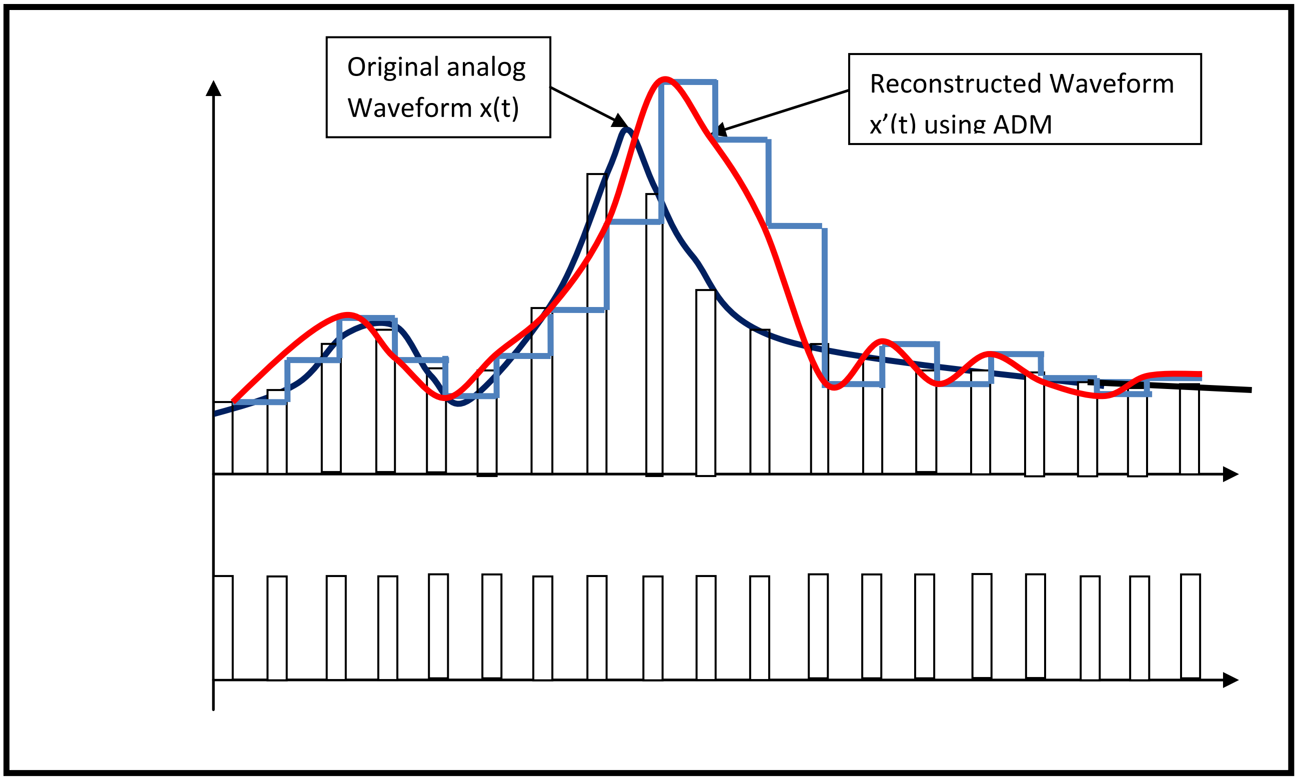
(i) Slope overload
(ii) Granular Noise

Slope Overload effect: Slope overload is a situation which occurs when delta modulation is used to encode a fast changing waveform. A fixed size step size Δv fails to keep pace with the fast rising or falling waveform. Slope overload effect can be controlled by using step size Δv of larger magnitude.

Granular Noise: This occurs when a waveform is changing very slowly and sign of the difference signal changes alternatively for every incoming sample. Following figure explains both these effects. The granular noise can be controlled by using step size of smaller magnitude. However smaller step size increases slope overload effect.



Adaptive Delta Modulation (ADM): ADM is a technique which overcomes the drawbacks, like slope overloading and granular noise, of simple DM. In this technique the magnitude of step size is increased when slope of a incoming waveform changes fast. This is visible if the sign of difference does not change for several incoming samples, accordingly the step size Δv is progressively increased. If the sign of the difference signal changes for every incoming sample this implies that the waveform is changing very slowly and when this happens, the step size Δv is progressively reduced.



Adaptive Differential Pulse Code Modulation (ADPCM) :

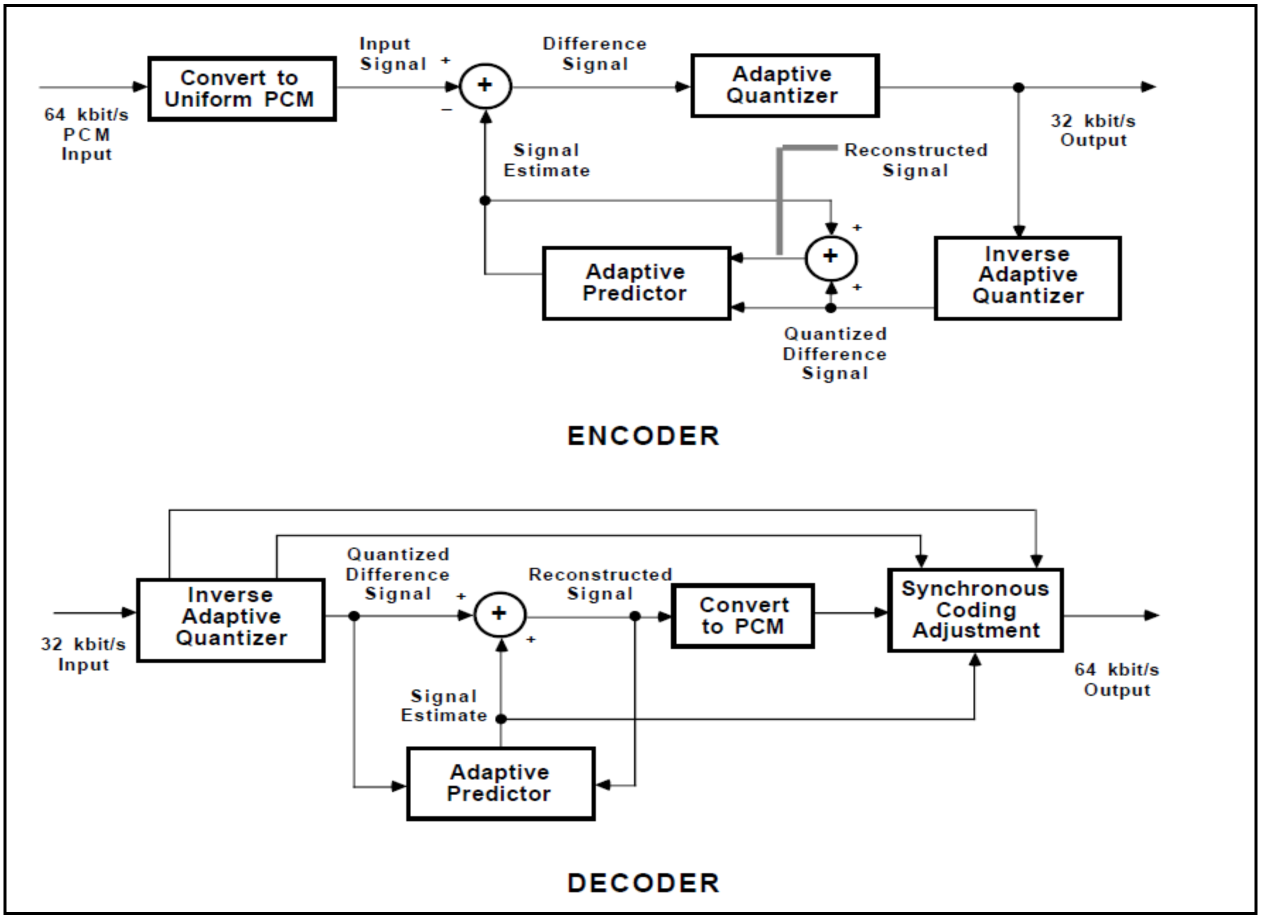
This is an another Technique for speech coding. Pulse code modulation (PCM) samples an input signal using a fixed quantizer to produce a digital representation. This technique, although simple to implement, does not take advantage of any of the redundancies in speech signals. The value of the current input sample does not have an effect on the coding of future samples. Adaptive differential PCM (ADPCM), on the other hand, uses an adaptive predictor, one that adjusts according to the value of each input sample, and thereby reduces the number of bits required to represent the data sample from eight (nonadaptive PCM) to four.

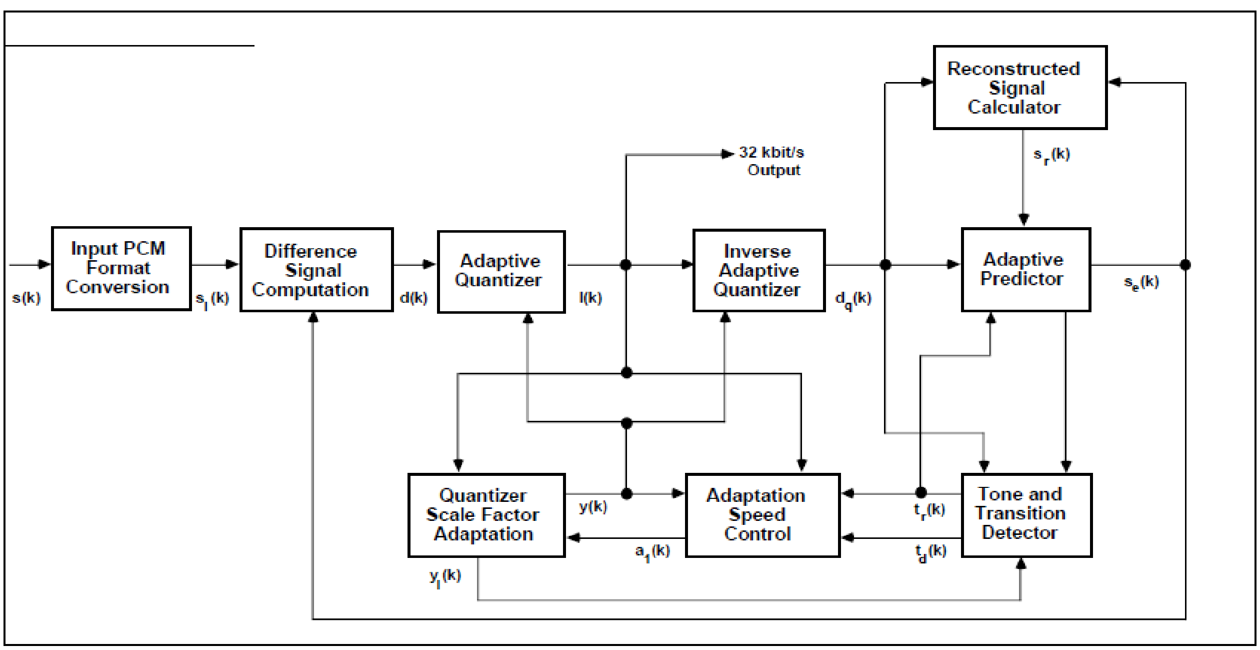
ADPCM does not transmit the value of the speech sample, but rather the difference between a predicted value and the actual sample value. Typically, an ADPCM transcoder is inserted into a PCM system to increase its voice channel capacity. Therefore, the ADPCM encoder accepts PCM values as input, and the ADPCM decoder outputs PCM values.

ADPCM ALGORITHM

ADPCM presented here is based on the CCITT recommendation G.721. Figure 1, on the following page, shows a block diagram of the ADPCM algorithm. An 8-bit PCM value is input and converted to a 14-bit linear format. The predicted value is subtracted from this linear value to generate a difference signal. Adaptive quantization is performed on this difference, producing the 4-bit ADPCM value to be transmitted.

Both the encoder and decoder update their internal variables based on only the generated ADPCM value. This ensures that the encoder and decoder operate in synchronization without the need to send any additional or sideband data. A full decoder is embedded within the encoder to ensure that all variables are updated based on the same data. In the receiving decoder as well as the decoder embedded in the encoder, the transmitted ADPCM value is used to update the inverse adaptive quantizer, which produces a dequantized version of the difference signal. This dequantized value is added to the value generated by the adaptive predictor to produce the reconstructed speech sample. This value is the output of the decoder. The adaptive predictor computes a weighted average of the last six dequantized difference values and the last two predicted values. The coefficients of the filter are updated based on their previous values, the current difference value, and other derived values.


Convert to Uniform PCM
64 kbit/s PCM Input
Input Signal
Difference Signal
Adaptive Quantizer
32 kbit/s Output
Signal Estimate
Reconstructed Signal
Adaptive Predictor
Inverse Adaptive Quantizer
Quantized Difference Signal
ENCODER
Inverse Adaptive Quantizer
32 kbit/s Input
Quantized Difference Signal
Reconstructed Signal
Convert to PCM
Synchronous Coding Adjustment
64 kbit/s Output
Signal Estimate
Adaptive Predictor
DECODER


Reconstructed Signal Calculator
32 kbit/s Output
s_r(k)
Input PCM Format Conversion
s(k)
s_l(k)
Difference Signal Computation
d(k)
Adaptive Quantizer
I(k)
Inverse Adaptive Quantizer
d_q(k)
Adaptive Predictor
s_e(k)
Quantizer Scale Factor Adaptation
y(k)
Adaptation Speed Control
t_r(k)
Tone and Transition Detector
a_1(k)
t_d(k)
y_l(k)