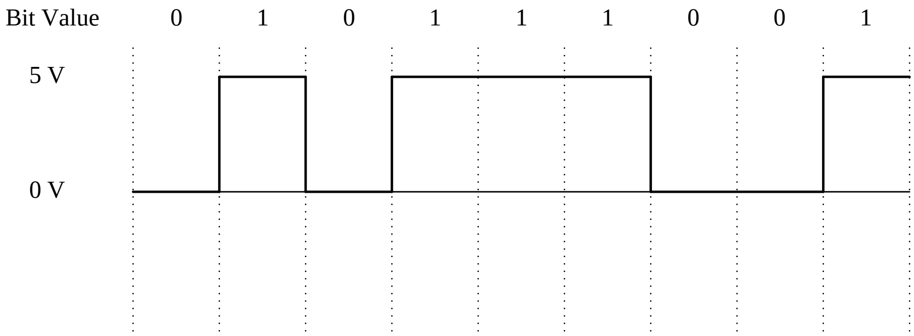
Contents

In digital communication systems, the information is always assumed to be generated in the form of binary data with values of 0 or 1. The origin of the digital binary information to be transmitted over digital communication systems may be an analog signal such as an audio signal or an analog picture that have been sampled and quantized and then converted to a PCM signal. The origin of the digital binary information may also be digital data in the form of text. In any case, the purpose of this chapter is to study the transmission of the information in the form of a digital binary signal.

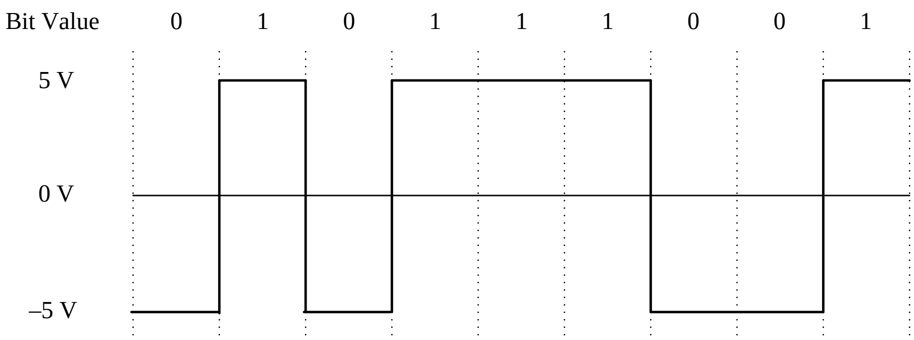
7.1 Line Coding

Given some binary information, the binary bits are not transmit through the channel as 1's and 0's but is used to generate a voltage signal that represents the information we would like to transmit. There are different forms of signals (called Line Codes) that can be used to represent the information. The terms Return to Zero (RZ) and Non–Return to Zero (NRZ) will be used in describing these signal.

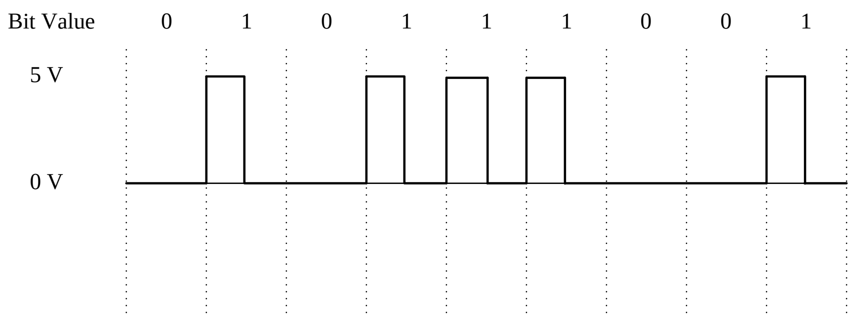
1. **<u>On–Off (NRZ):</u>** In this form of line codes, a bit of 1 is represented by some positive voltage (+5 volts for example) and a bit of 0 by 0 volts (justifying calling this signal On–Off). The pulses corresponding to binary 1 remain at the positive voltage for the whole duration of the bit period (it does not return to zero at any time during the bit period justifying calling this line code NRZ).



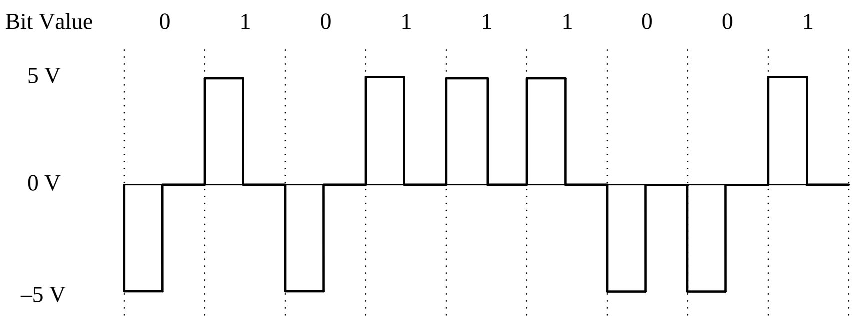
2. **<u>Polar (NRZ):</u>** In this line codes, a bit of 1 is represented by some positive voltage (+5 volts for example) and a bit of 0 is represented by negative of that voltage (so it would be –5 volts). The pulses corresponding to binary 1 and binary 0 remain at the positive and negative voltages, respectively, for the whole duration of the bit period (they do not return to zero). The advantage of this line code over the On–Off (NRZ) is that it has zero–DC value when the number of binary 1's is equal to the number of binary 0's. A line code with zero–DC is desired in some applications that require that the transmitted signal to have no DC.



3. **<u>On–Off (RZ):</u>** In this line codes, a bit of 1 is represented by some positive voltage (+5 volts for example) for half of the bit period and zero in the other half of the bit period and a bit of 0 is represented by zero for the whole bit period. This is why this line code is a return–to–zero line code (because any pulse corresponding to binary 1 always returns back to zero). The advantage of this line code over the previous line codes is that a long sequence of ones always has transitions at the center of each bit and therefore bit synchronization becomes easy for long sequences of ones. Long sequence of zeros is still difficult to be synchronized.

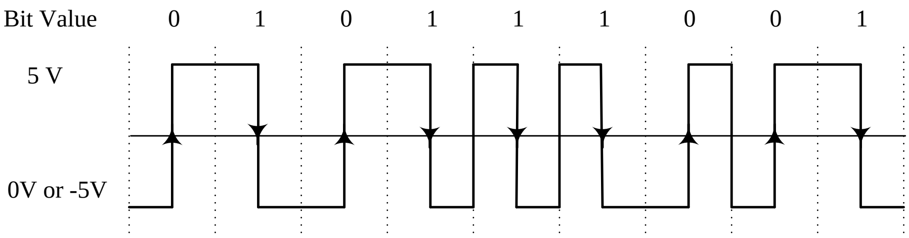


4. **Polar (RZ):** In this line code, a bit of 1 is represented by some positive voltage (+5 volts for example) for half of the bit period and zero in the other half of the bit period and a bit of 0 is represented by the negative of that voltage for half of the period and zero for the other half. The advantage of this line code over the previous ones is that long sequences of ones or zeros have transitions at the center of each bit and therefore bit synchronization becomes easy for long sequences of ones or zeros. Also, this line code has zero DC when the number of ones and zeros is the same.

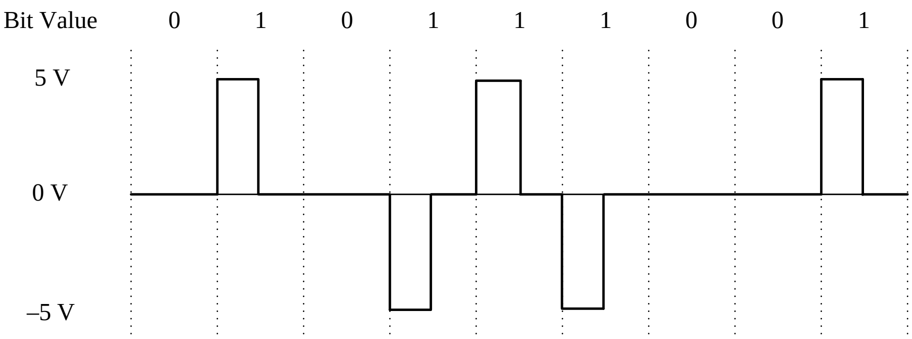


5. **Bi-Phase (Manchester):** The problem with the upper line codes is that they either have non-zero average or they do not provide sufficient information that allows the transmitter and receiver to synchronize with each other. So, if there is a long sequence of 0s or long sequence of 1s, the receiver and transmitter may loose synchronization. The Polar (RZ) solves these problems but it is a three level signal (the signal takes one of three levels (+5, 0, or –5). A transistor that is working a switch has only two states (On ➔ saturation, and Off ➔ cutoff). So, we can get all the advantages of the different line codes yet use a line code that has two levels only. The way the Biphase, or Manchester, code is constructed is that a bit of 0 is represented by a 0V signal for half the bit period and +5V for the second half, while a bit of 1 is represented by a +5V signal for half the bit period and 0V for the second half. So, while all the codes transmit the information in the level of the signal, this line code actually transmits the information in the transitions that occur in the middle of each bit. This is illustrated by the arrows in the middle of each bit period. A transition going up in this case means a bit of zero and transition going down is a bit of 1. To get an average of zero, we

can use the two levels of +5V and -5V. Note that while the average of some line codes is zero when the number of zeros is equal to ones, this is not a necessary condition for the Manchester line code to have zero average since each bit whither it is 0 or 1 has half of its duration at -5V and the other half at +5. It is most used by IEEE 802.3, baseband coax and twisted pair CSMA /CD bus LANs.



6. **<u>Bipolar (RZ):</u>** In this line code, a bit of 0 is represented by zero volts for the whole bit period. A bit of 1 is represented by some positive voltage (+5 volts for example) for half of the bit period and zero in the other half of the bit period. However, the next bit of one (wither it is the next bit or 1000 bits later is represented by the negative of the voltage for half of the bit period and zero for the second half. So, the bits of 1's are represented by alternating positive and negative pulses. This insures that the DC value of the signal is always zero even if we have non–equal number of ones and zeros.



Performance Criteria of Line Codes

- Zero DC value
- Inherent Bit-Synchronization: rich in transitions
- Average Transmitted Power for a given Bit Error Rate (BER): power efficiency.
- Spectral Efficiency (Bandwidth): inversely proportional to pulse width.
- Error detection and correction capabilities.
- Insensitivity to signal inversion.

Transparent Codes: a line code is said to be transparent if the bit pattern does not affect the accuracy of the timing information.

Advanced Examples in Line Coding: High Density Bipolar (HDBN)

Because the AMI is not transparent other methods are used to prevent long strings of zeros. HDBN also does not have any dc value and have the same data rate.

In this case when a run of N+1 zeros happens, they will be replaced by a code of length N+1 containing AMI violation.

The most popular form of HDBN is HDB3; which uses to special sequences: 000V and B00V.

B00V is used when there is an even number of ones following the last special sequence and 000V is used when there are an odd number of ones following the last special sequence. Consecutive V pulses alternate in sign to avoid dc wander.

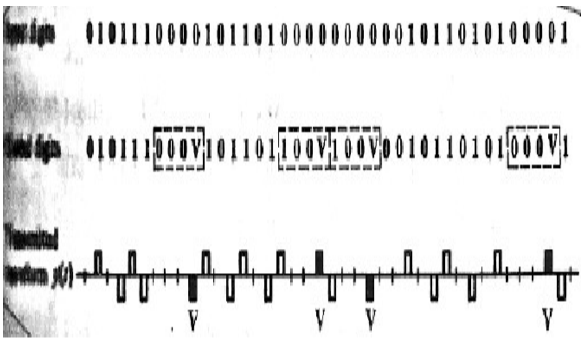
Because violation just happens at the fourth bit of the special code, it can be easily detected and will be replaced by a zero at the receiver.

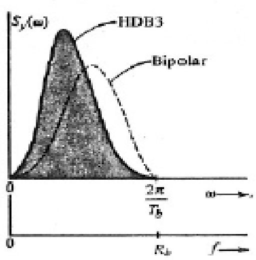
It is also capable of error detecting because a sign error would make the number of bipolar pulses between violations even instead of odd.

Another way to avoid long string of zeros or ones is using the BNZS code which is similar to HDBN.

For example in B8ZS a string of 8 zeros will be replaced by **000VB0VB** where V's are bipolar violation and B's are valid bipolar signals.

B8ZS is most used in DS1 signals and in North America.

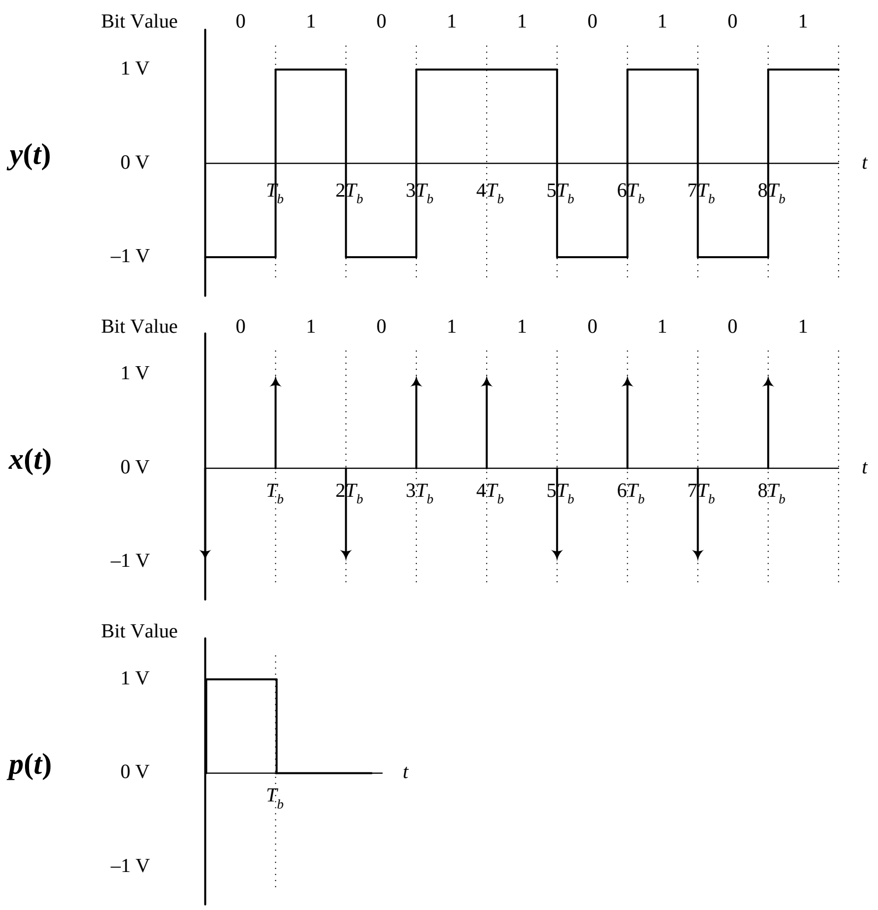




7.2 Power Spectral Density of Line Codes

The line codes discussed in the previous lecture generally not the best line codes to be used because all of these line codes have the form of pulses. As it is known, pulses have spectrums of the form of sinc functions. So, in theory, channels with infinite bandwidths are required to transmit any of the line codes discussed previously. To study the performance of a line code we need to consider the Power Spectral Density (PSD) of line codes. **The reason for not being able to use the Fourier transform to find the spectrum of a line code is that the information signals that generate a line code is a stochastic (non–deterministic) signal**, and the Fourier transform cannot be applied for non–deterministic signals. To study the spectrum of stochastic signals, we use the PSD, which shows the distribution of the signal power versus the frequency.

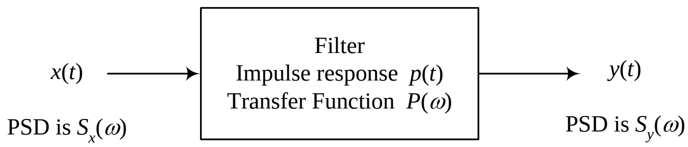
Consider the following Polar NRZ line code $y(t)$. This signal can be decomposed into two signals, the information signal $x(t)$ represented by a sequence of delta functions that have positive or negative areas depending on the corresponding bits (0's or 1's) that is convolved by a pulse signal $p(t)$.



So,

$$y(t) = p(t) * x(t).$$

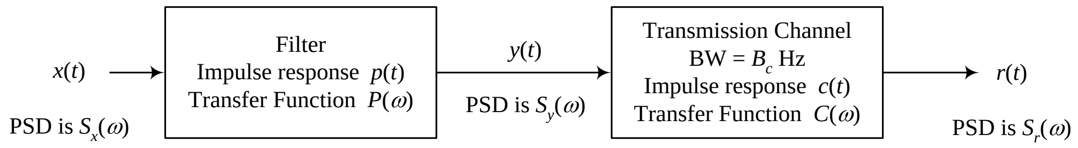
The signal $y(t)$ in the above relation is similar to the output that we obtain when we pass a signal $x(t)$ through a filter with impulse response $p(t)$ and frequency transform $P(\omega)$. So, the we can obtain $y(t)$ using the following block diagram.



A property of the PSD of a signal that passes through a filter is given as

$$S_y(\omega) = |P(\omega)|^2 S_x(\omega).$$

Using the same concept of the PSD, transmitting the signal $y(t)$ through a channel with impulse response $c(t)$ and transfer function $C(\omega)$ as shown below

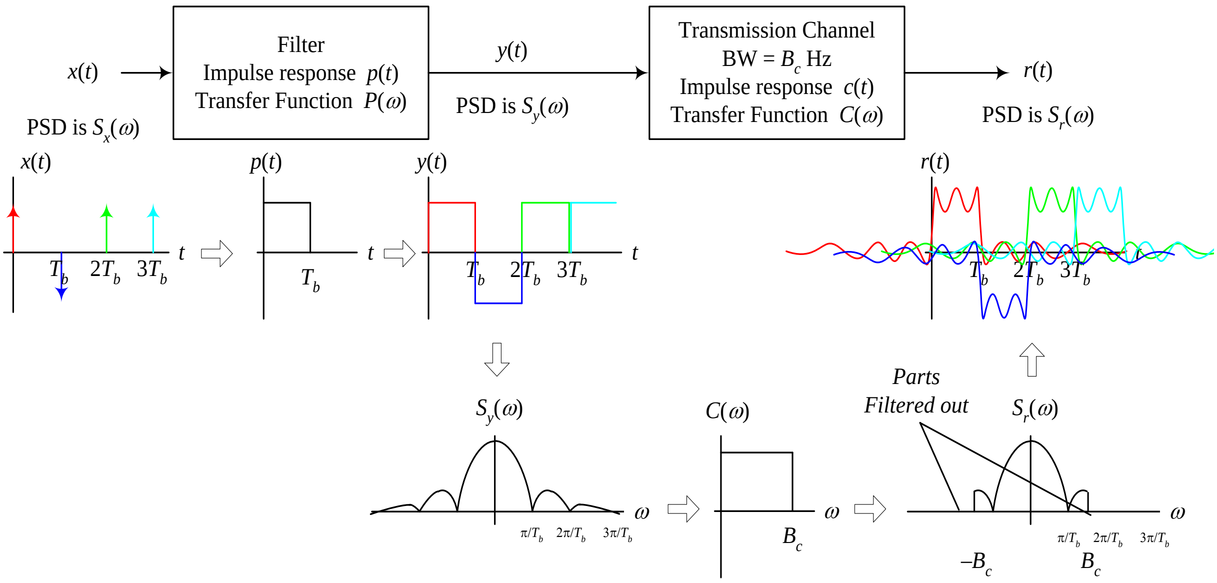


we see that the PSD of the received signal $r(t)$ is

$$S_r(\omega) = |C(\omega)|^2 S_y(\omega).$$

Knowing that the PSD gives the spectrum of random signals (so the maximum frequency at which the PSD of a signal is non–zero can be considered as the bandwidth of that signal), we see that for a signal $y(t)$ to be transmitted properly through a channel, the bandwidth of the channel B_c must at least be as much as the bandwidth of the transmitted signal $y(t)$ (or the bandwidth of $y(t)$ must be less than or equal to B_c).

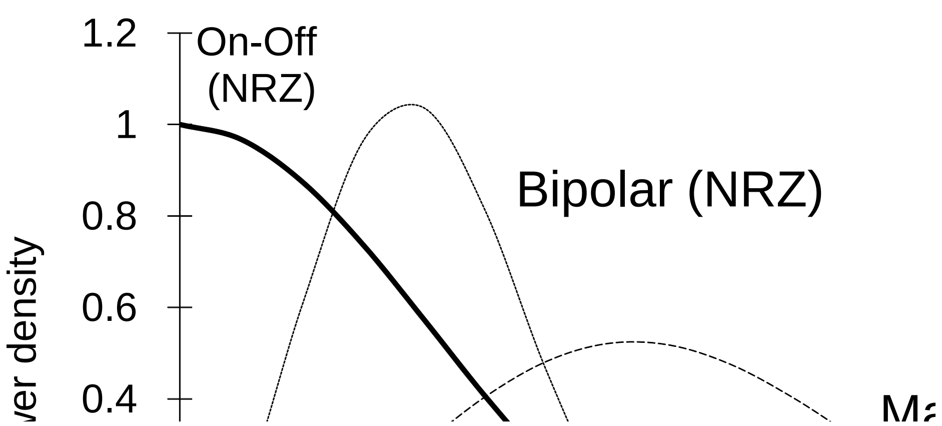
This means that the line codes discussed in the previous lecture are in theory not suitable for most channels (and in practice they are not). The reason is that these line codes are built on pulses (similar to the rect function) so they have very high bandwidth. Unless the channel has a very high bandwidth to accommodate this wideband signal, part of the transmitted signal will be cutoff by the channel and the received signal will be different from the transmitted signal. When part of the spectrum of a transmitted signal is cutoff by the channel, a phenomenon known as **Inter–Symbol Interference (ISI)** occurs. When square pulses (which have high bandwidth) are transmitted, channels with limited bandwidth remove the high frequency components of the transmitted signal. This causes the pulse of every bit to extend beyond its borders (instead of the pulse being confined to a bit period T_b, the pulse depending on how much was cut from the spectrum of the transmitted signal will extend its boundaries to several bit symbols. For some applications, this elongation of the pulse of a specific bit may effectively extend over 100 bits on each side. Clearly if every bit extends on each side over many bit periods, interference between the different bits will make it very difficult to detect the received bits. Therefore, the effect of ISI is that it will make it very difficult for the receiver to detect the transmitted bits. The extension of a pulse over many bit periods is shown in the next figure



It is clear from the above figure that the square pulses in $y(t)$ got spread out in the received signal $r(t)$. If the spreading is so severe, it may result in bit detection errors.

To reduce ISI in a signal, we need to change the PSD of the transmitted signal $y(t)$. As seen above, $S_y(\omega)$ can be modified by either modifying $S_x(\omega)$ or modifying $P(\omega)$. Since we have no control over $S_x(\omega)$ because it is the PSD of the original information bits, we only can modify $S_y(\omega)$ by modifying $P(\omega)$, and hence modifying $p(t)$, which is the pulse used in the line code.

It is clear that if we use a pulse that looks like a sinc function, then its spectrum will be similar to a rect function. In fact, the sync function is in theory the best signal to be used in terms of the required bandwidth since its spectrum occupies the minimum band. The problem with sinc pulses is that they extend in time theoretically from $t = -\infty$ to ∞ (in practice the sinc extends over a range around $t = -100T_b$ to $100T_b$ depending on the requirements), so we would have to start generating a sinc pulse in theory at $t = -\infty$ to transmit a bit at $t = 0$ and then continue generating this pulse until $t = \infty$. This is not practical, so we will search for other pulses that are time–limited and do not cause ISI.

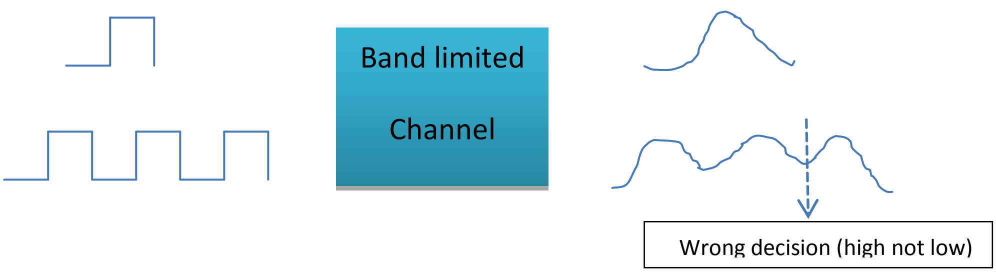


7.3 Pulse shaping and ISI

After choosing the line code, we need to choose the pulse shape for good PSD (power spectral density)

Channels are bandlimited (For example some twisted pair cables are limited to 1MHZ). The limited bandwidth results in time unlimited pulse (dispersion) which results in ISI (Inter-symbol Interference).

ISI which is spreading of a pulse beyond its interval Tb ia the major limiting factor in digital communications. As we increase the data rate, the pulses get closer to each other and ISI become higher. ISI is cause by bandlimitation.



So how to get ZERO ISI?

Solution #1 Pulse shaping

Nyquist Criterion for ZERO ISI : Nyquist achieves zero ISI by choosing a pulse shape that has a nonzero amplitude at its center and zero amplitude at $t=\pm nT_b$ (n=1,2,3,……).

T_b is the separation between successive pulses. $T_b=1/R_b$

$$p(nT_b) = \begin{cases} 1 & n = 0 \\ 0 & for\ all\ other\ n \end{cases}$$

Which pulse satisfies that with minimum bandwidth requirements?

$$p(t) = sinc(\pi R_b t)$$

$$P(\omega) = \frac{1}{R_b} rect\left(\frac{\omega}{2\pi R_b}\right)$$

Problems with sinc?

1) Starts at -∞ and ends at +∞. We have to truncate but the bandwidth will be greater than $R_b/2$.
2) Decays very slowly $1/t$ rate. In real life things will not be perfect. There is always time jitter (deviation in time). There will be a time error $\sum \frac{1}{n}$ (does not converge. Add up to a large value) .

The solution to the sinc problem is that we may choose a pulse that decay at a faster rate and it will require more bandwidth $rR_b/2$ $\quad 0 \le r \le 1$. To find this pulse, let us examine Nyquist criteria for zero ISI in the frequency domain.

$\bar{p}(t) = p(t)\delta_{T_b}(t) = \delta(t)$ Nyquist criteria for zero ISI.

Take Fourier transform of both sides

$\frac{1}{T_b}\sum_{n=-\infty}^{\infty} P(\omega - n\omega_b) = 1 \qquad\qquad \omega_b = \frac{2\pi}{T_b} = 2\pi R_b$

This means the pulse that have zero ISI, should have a spectrum if shifted to the multiple value of the rate should result in a constant.

Nyquist proposed a condition for pulses $p(t)$ to have zero–ISI when transmitted through a channel with sufficient bandwidth to allow the spectrum of all the transmitted signal to pass. Nyquist proposed that a zero–ISI pulse $p(t)$ must satisfy the condition

$$p(t) = \begin{cases} 1 & t = 0 \\ 0 & t = \pm T_b, \pm 2T_b, \pm 3T_b, \ldots \end{cases}$$

A pulse that satisfies the above condition at multiples of the bit period T_b will result in zero–ISI if the whole spectrum of that signal is received. The reason for which these zero–ISI pulses (also called Nyquist–criterion pulses) cause no ISI is that each of these pulses at the sampling periods is either equal to 1 at the center of pulse and zero the points other pulses are centered.

In fact, there are many pulses that satisfy these conditions. For example, any square pulse that occurs in the time period $-T_b$ to T_b or any part of it (it must be zero at $-T_b$ and T_b) will satisfy the above condition. Also, any triangular waveform (Δ function) with a width that is less than $2T_b$ will also satisfy the condition. A sinc function that has zeros at $t = \pm T_b, \pm 2T_b, \pm 3T_b, \ldots$ will also satisfy this condition. The problem with the sinc function is that it extends over a very long period of time resulting in a lot of processing to generate it. The square pulse required a lot of bandwidth to be transmitted. The triangular pulse is restricted in time but has relatively large bandwidth.

There is a set of pulses known as raised–cosine pulses that satisfy the Nyquist criterion and require slightly larger bandwidth than what a sinc pulse (which requires the minimum bandwidth ever) requires.

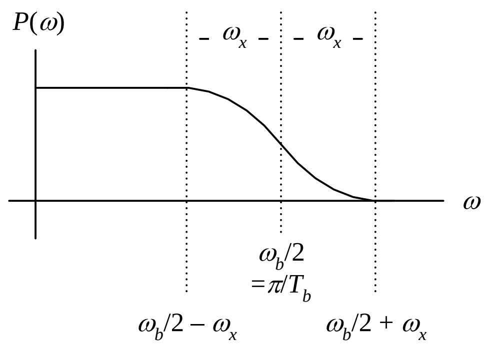
The spectrum of these pulses is given by

$$P(\omega)=\begin{cases}\frac{1}{2}\left[1-\sin\left(\frac{\pi\{\omega-(\omega_b/2)\}}{2\omega_x}\right)\right] & \left|\omega-\frac{\omega_b}{2}\right|<\omega_x \\ 0 & |\omega|>\frac{\omega_b}{2}+\omega_x, \\ 1 & |\omega|<\frac{\omega_b}{2}-\omega_x\end{cases}$$

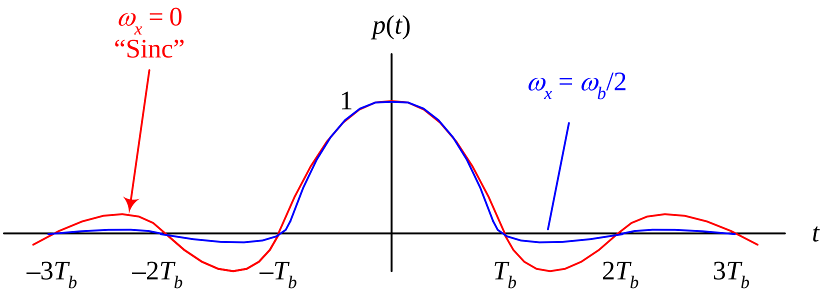
where ω_b is the frequency of bits in rad/s ($\omega_b = 2\pi/T_b$), and ω_x is called the excess bandwidth and it defines how much bandwidth would be required above the minimum bandwidth that is required when using a sinc pulse. The excess bandwidth ω_x for this type of pulses is restricted between

$$0 \le \omega_x \le \frac{\omega_b}{2}.$$

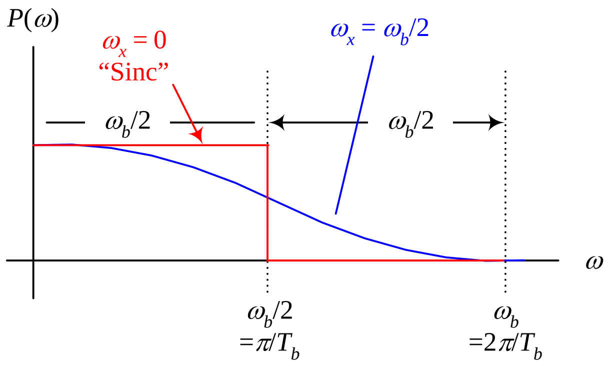
Sketching the spectrum of these pulses we get



We can easily verify that when $\omega_x = 0$, the above spectrum becomes a rect function, and therefore the pulse $p(t)$ becomes the usual sinc function. For $\omega_x = \omega_b/2$, the waveform is similar to a sinc function but decays (drops to zero) much faster than the sinc (it extends over 2 or 3 bit periods on each side). The expense for having a pulse that is short in time is that it requires a larger bandwidth than the sinc function (twice as much for $\omega_x = \omega_b/2$). Sketch of the pulses and their spectrum for the two extreme cases of $\omega_x = \omega_b/2$ and $\omega_x = 0$ are shown below.



The above blue figure should have double zero crossing



We can define a factor r called the roll–off factor to be

$$r = \frac{\text{Excess Bandwidth}}{\text{Minimum Bandwidth}} = \frac{\omega_x}{\omega_b/2} = \frac{2\omega_x}{\omega_b}.$$

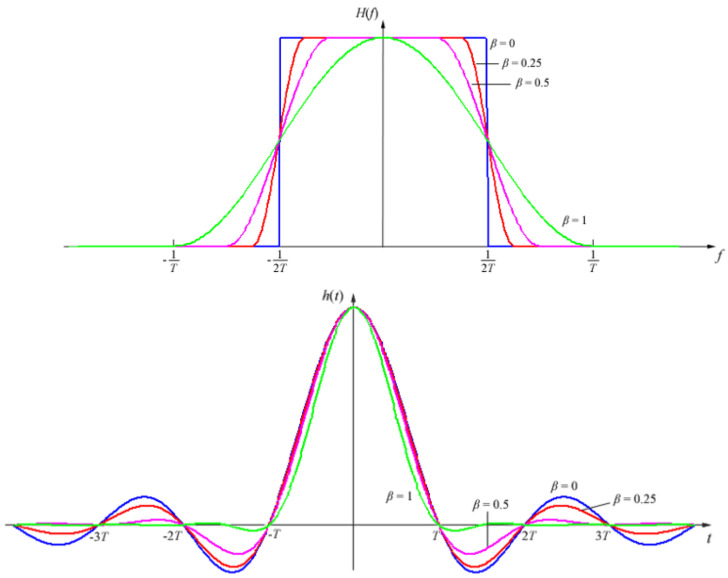
The roll–off factor r specifies the ratio of extra bandwidth required for these pulses compared to the minimum bandwidth required by the sinc function.

For r=1, (full roll-off factor). The above equations for raise d cosine becomes

$$P(\omega) = cos^2\left(\frac{\omega}{4R_b}\right) rect\left(\frac{\omega}{4\pi R_b}\right)$$

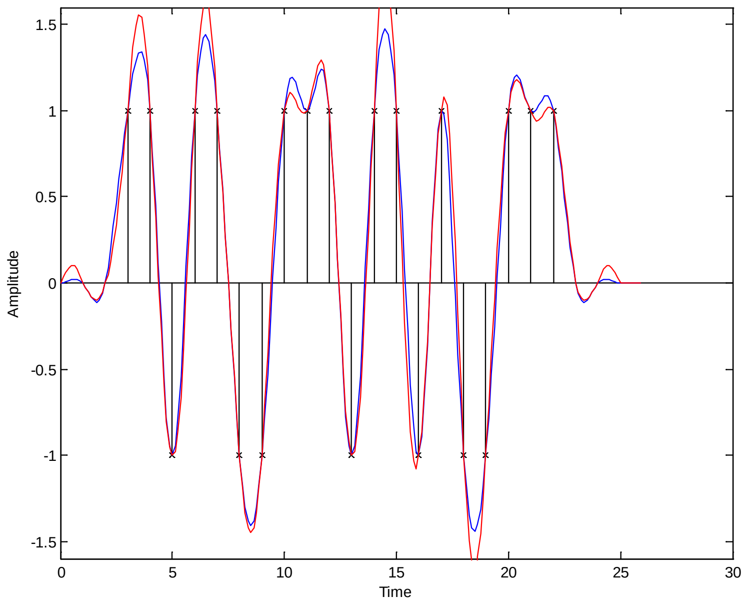
And $$p(t) = R_b \frac{cos4\pi R_b}{1-4R_b^2t^2} sinc(\pi R_b t)$$

From http://en.wikipedia.org/wiki/Raised-cosine_filter



Using Matlab/Toolbox/Communications/Raised Cosine Filtering

This step demonstrates the effect that changing the rolloff factor from .5 (blue curve) to .2 (red curve) has on the resulting filtered output. The lower value for rolloff causes the filter to have a narrower transition band causing the filtered signal overshoot to be greater for the red curve than for the blue curve.



Solution #2 Controlled ISI (Duobinary Pulses)

Signaling with controlled ISI : Partial Response Signals. Pulse shaping achieves zero ISI at the cost of reduced the rate (-ve) or by increased bandwidth.

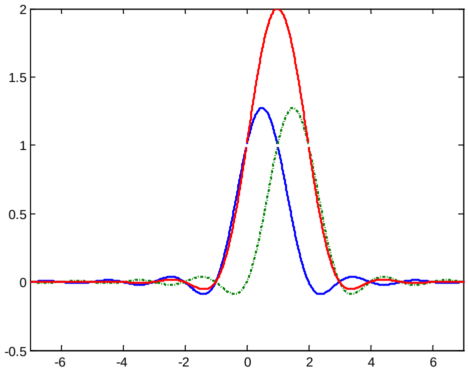
In controlled ISI we allow the pulse to expand more which reduces the needed bandwidth. Consider the **duobinary pulse.**

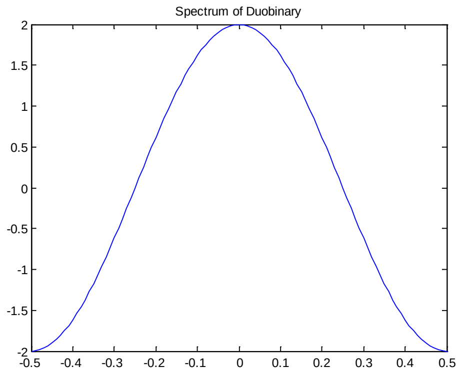
$$p(nT_b) = \begin{cases} 1 & n = 0,1 \\ 0 & for\ all\ other\ n \end{cases}$$

If two consecutive bits are 1's the received voltage will be 2. If two consecutive bits are 0's the received voltage will be -2. If the two consecutive bits are opposite the voltage will be 0.

Error detection capability: we cannot have +2 -2 or -2 +2

Full value of the same polarity have even number of zeros in between






```
Rb=1; % Assume the rate Rb=1
t=-10:0.01:10;
p=sin(pi*Rb*t)./(pi*Rb*t)./(1-Rb*t);
p2=sin(pi*Rb*(t-1))./(pi*Rb*(t-1))./(1-Rb*(t-1));

plot (t,p ,t,p2,t,p+p2)
axis([-7 7 -0.5 2])
```

Full values of the opposite polarity have odd number of zeros in between

The pulse that does that

$$p(t) = \frac{\sin(\pi R_b t)}{\pi R_b t(1 - R_b t)}$$

$$P(\omega) = \frac{2}{R_b} cos\left(\frac{\omega}{2R_b}\right) rect\left(\frac{\omega}{2\pi R_b}\right) e^{-j\frac{\omega}{2R_b}}$$

+ve: minimum required bandwidth $R_b/2$ Hz.

+ve: rate of decay $1/t^2$.

+ve: still start at -∞ but easier to approximate compared with sinc.

Example: duobinary signaling

Data	1	1	0	1	1	0	0	0	1	0	1	1	1
	+	+	-	-	+	+	-	-	+	+	+	+	+
		+	+	+	+	-	-	-	-	-	-	+	+
Duobinary	1	2	0	0	2	0	-2	-2	0	0	0	2	2
Detected	1	1	0	1	1	0	0	0	1	0	1	1	1

The decoding rule is the relation between the received Duobinary and the detected sequence (0= change, +2 or -2 means no change).

Differential Precoding

The problem with the above example is that the decoding is differential and if there is an error it will propagate.

The problem of error propagation is solved using differential encoding (precoding).

Precoding means

1 sends identical like previous

0 change previous transmission

The precoded sequence will be insensitive to polarity change (flipping does not affect)

Example : Differential coding (pre-coding)+ Duobinary pulse

Data	1	1	0	1	1	0	0	0	1	0	1	1	1
Precoded	1	1	0	0	0	1	0	1	1	0	0	0	0
	+	+	-	-	-	-	-	-	+	+	-	-	-
		+	+	-	-	+	+	+	+	-	-	-	-
Duobinary	1	2	0	-2	-2	0	0	0	+2	0	-2	-2	-2
Detected	1	1	0	1	1	0	0	0	1	0	1	1	1

The decoding rule become easier (advantage)

0 for 0

-2 or +2 for 1

Simplified decision (independent of previous bit) and no error propagation

Note: designing for a specific pulse shape might have practical limitations, especially that the designed pulse should be at the receiver side.

7.7 M-ary Communications

New applications call for high data rate but we are limited!

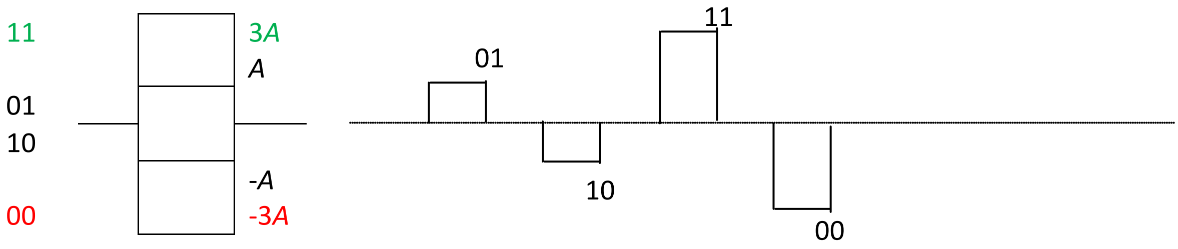
Baseband Channel Capacity= 2*BW(Symbols/sec) (**not** bits/sec)

M-ary Communications means communication using ***M*** symbols.

Multi-amplitude

Example M=4 (4-ary or quaternary).

Two binary digits can be transmitted by one 4-ary symbol (RZ)



$I_M = \log_2 M$ binary digits/symbol

Notice to make the above quaternary system equivalent to the binary system in terms of BER, the spacing between the level is kept $2A$

$P_{2\text{-ARY}} = (A^2 + A^2)/2 = A^2$

$P_{4\text{-ARY}} = (A^2 + 9A^2 + A^2 + 9A^2)/4 = 5A^2 = A^2 + 4A^2$

Practice: show that the power required for 8-ary multi-amplitude case will be

$P_{8\text{-ARY}}$=…………………… $=A^2+20A^2$

Example How many symbols are required to send 3000 bits using 2-ary (binary system), 8-ary system, 64-ary ?

Ans. 3000, 1000, 500 symbols

For multi-amplitude communications

+ve the transmission rate, R_b , increase by a factor of I_M.

-ve (the cost) to maintain the same bit error rate (BER), power increase is proportional to M^2.

Bandwidth is independent of M (the amplitude changes and not the duration)

Alternatively rather than increasing the rate, we can reduce the bandwidth by a factor of I_M at the cost of increased duration and power

BER	BW	Power	Rate
fixed	fixed	increase	increase

Another way to increase the signaling is

Orthogonal Signaling

$$\int_0^{T_b} \varphi_i(t)\varphi_j(t)\,dt = \begin{cases} c & i = j \\ 0 & i \neq j \end{cases}$$

Example $\varphi_k(t) = sin\left(\frac{2\pi kt}{T_b}\right)$ $0 < t < T_b$ k=1,2,……..M

Highest pulse frequency =M/T_b Bandwidth =M/T_b

The power is independent M

BER	BW	Power	Rate
fixed	increase	fixed	increase
increase	fixed	fixed	Increase

Compare Multi-amplitude with Orthogonal M-ary signaling

	BW	Power	When to use
Multi-amplitude	Independent of M	Proportional to M^2	BW at premium like telephone lines
Orthogonal	Proportional to M	Independent of M	Power at premium like space communications

Binary is the single most important way for signaling in practice (simplicity)
M-ary is used in many applications like dial up modems and satellite communications.

Example

A satellite transponder has a bandwidth of 36 MHz. Earth stations use raised cosine filters with M-ary PSK modulation to transmit 256 Mbps. What is the minimum possible value for M and the associated *roll-off factor? Hint: satellite communication indicate it is Passband communications*

Solutions

BW=(1+ r)R_s=(1+ r)R_b/$\log_2 M$

(1+ r)=(BW*log2 M)/R_b . Minimum $\log_2 M$ =8 => M=256, related r=0.125 .

7.8 Digital Carrier Systems

So far we have been dealing with baseband transmission which is good for many applications over (twisted pair cables, co-axial cables, fiber….)

Passband communication or carrier communication requires modulation (shifting the frequency continent). Passband communications is needed for frequency division multiplexing (FDM), or to control the propagation characteristics and antenna size.

	Maximum symbol rate
Baseband B Hz	2B

Passband B Hz	B

Types of digital carrier systems

1. Amplitude shift keying (ASK) : on-off keying (OOK) is one example
2. Phase shift keying (PSK)
3. Frequency shift keying (FSK): can be viewed as sum of two ASK(OOK) with f_1 and f_2. FSK is in principle two ASK with two different frequency. Hence the spectrum of FSK is the sum of two ASK spectra at the two different frequencies. It can be shown that by properly choosing the two frequencies the discrete component can be eliminated. Also that the bandwidth of FSK is higher than that of ASK and PSK.

See the three figures below:

Note that the use of rectangular pulses is for illustration: we need to use pulse shaping to eliminate (ISI) for passband systems too.

Comparing the above digital carrier systems, PSK has some advantage in terms of power efficiency compared with FSK and ASK. PSK is based on the polar representation (code) which is more efficient than the on-off signaling. This is why PSK requires 3dB less power to achieve the same BER.

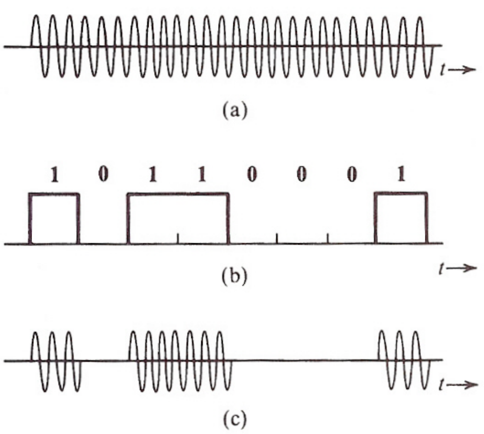


Figure 7.27 (a) Carrier $\cos \omega_c t$. (b) Modulating signal $m(t)$. (c) ASK: modulated signal $m(t) \cos \omega_c t$.

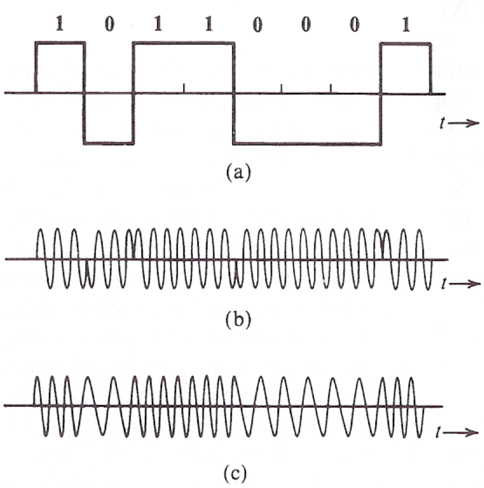
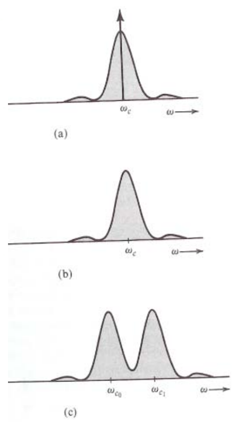


Figure 7.28 (a) Modulating signal $m(t)$. (b) PSK: modulated signal $m(t) \cos \omega_c t$. (c) FSK: modulated signal.

Figure 7.29 PSD of: (a) ASK. (b) PSK. (c) FSK.



De-modulation

- Coherent detection (relatively complicated, good at low SNR, excellent performance)
- Non-coherent

ASK

Non-coherent otherwise we lose simplicity which the main reason for using ASK.

PSK

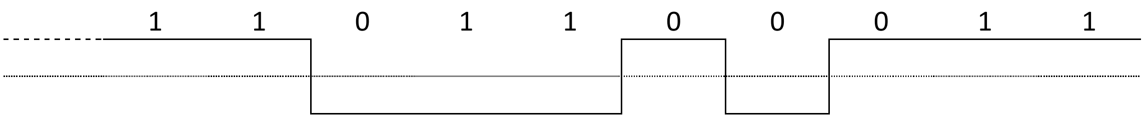
- Coherent (similar to analog)
- Non-coherent: using ingenuous method: We need to differentially encode the data, the resultant is known as differential PSK (DPSK)

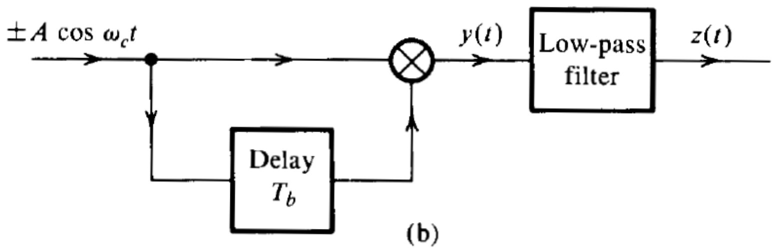
Differential encoding means,

- if the data is 1=> same encoding as before
- If the data is 0=> negative of the previous

<u>Figure 7.30b</u>

Example





(b)

If identical like previous

$y(t) = \frac{A^2}{2}(1 + \cos{(2\omega_c t)}$ and $z(t) = \frac{A^2}{2}$ => bit was 1

If different than previous

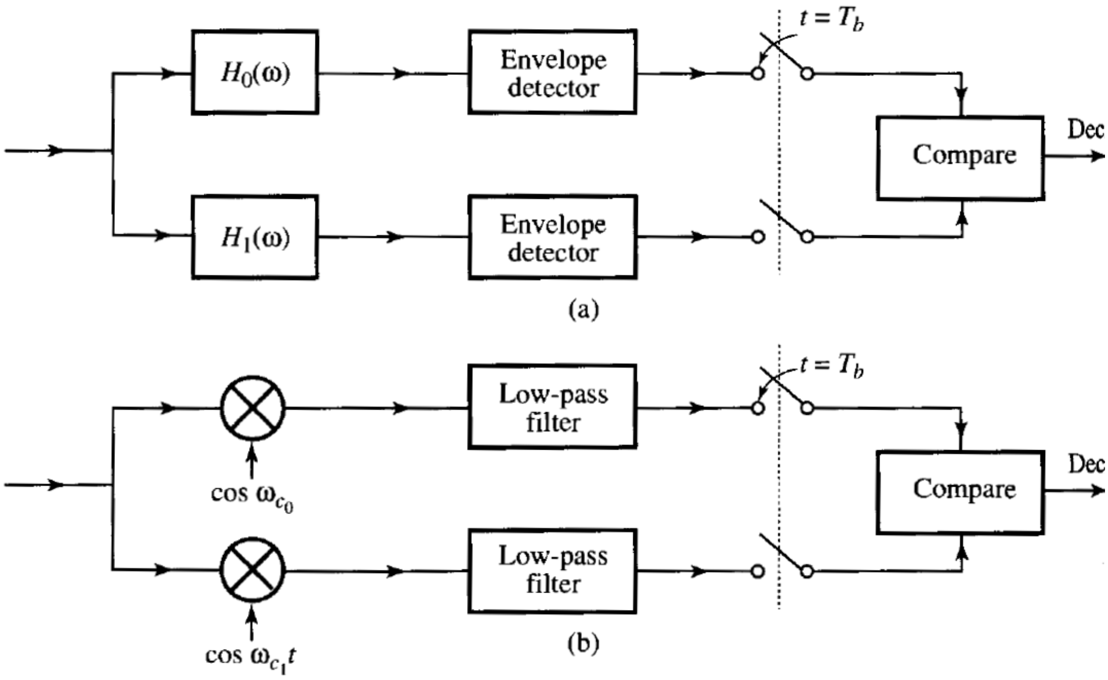
$y(t) = -\frac{A^2}{2}(1 + \cos{(2\omega_c t)}$ and $z(t) = -\frac{A^2}{2}$ => bit was 0

As an example decode the above sequence and demonstrate that you can recover the original data.

FSK

Can be demodulated either coherently or non-coherently:

Build Figures 7.31 yourself!



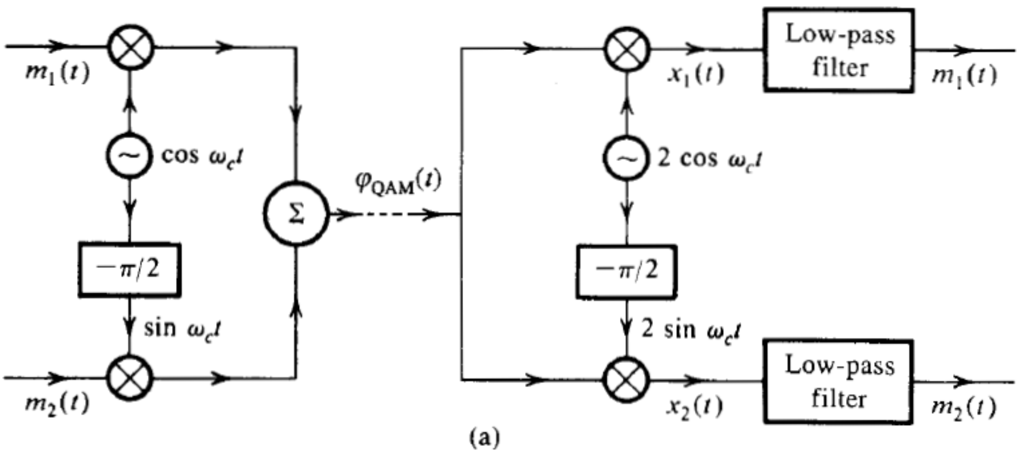
Digital Signal Transmission Using QAM

We may use M-ary QAM with different amplitudes and phases (Figure 7.32)

QAM is used for telephone line modems. **Mod**ulate-**dem**odulate=(modem)

$m_1(t)$ and $m_2(t)$ are binary polar pulse sequences.

When M=4, QAM is equivalent to QPSK.

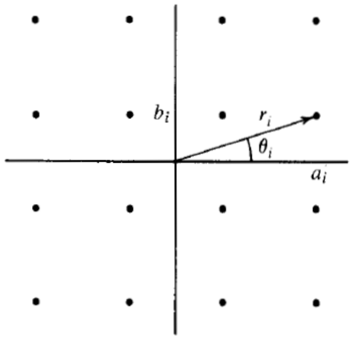


We can use M=16

$p_i(t) = a_i p(t) cos\omega_c t + b_i p(t) sin\omega_c t = r_i p(t) cos(\omega_c t - \theta_i)$

i=1,2,……..,16

where $r_i = \sqrt{a_i^2 + b_i^2}$ and $\theta_i = tan^{-1}\frac{b_i}{a_i}$



7.9 Digital Multiplexing

Several low-bit-rate singles can be multiplexed to form a one high-bit rate signal to be transmitted over a high-frequency medium. We use TDM with overhead bits to identify the beginning of the frame.

Types of Time-division multiplexing of digital signals

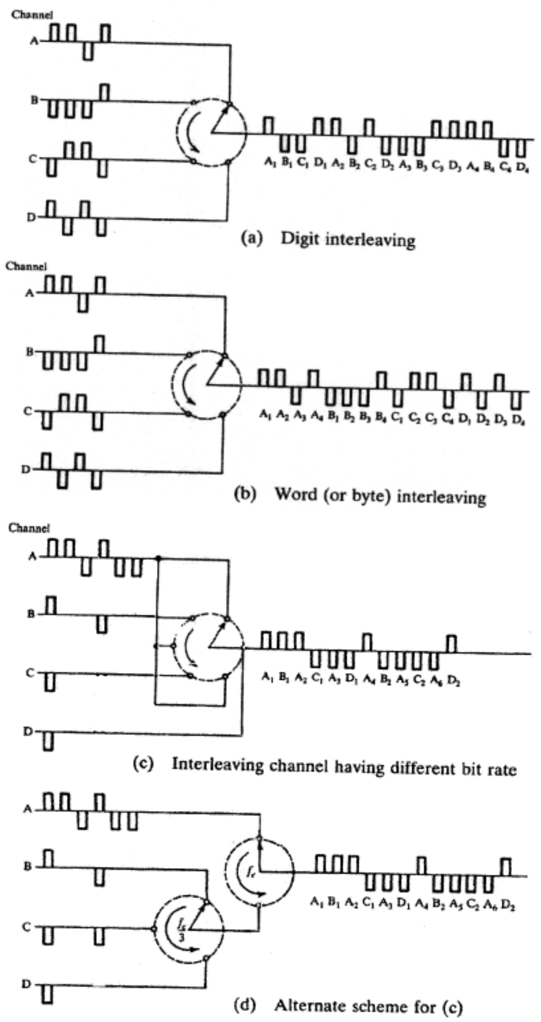


Figure 7.33 Time-division multiplexing of digital signals.

In the majority of cases not all incoming channels are active all the time. Some of them are idle. We can accept more inputs assuming that with a very low probability the system will be overloaded. Example (TDMA for satellite)

Asynchronous Channels and Bit Stuffing

Example: a 1000km coaxial cable carrying $2*10^8$ pulses per second. Assuming a nominal propagation speed $2*10^8$ m/s it takes 1000km/$2*10^8$ m/s =1/200 sec of transit time and 1 million pulses will be in transit. If the cable temperature increased by 1°F the propagation velocity will increase by 0.01%. The transit pulses will arrive sooner…. We need to control the rate. it is a synchronous . empty slots need to be filled with dummy digits **(pulse stuffing)** and we also need **elastic store** (**justification buffer**)

Another source of asynchronous-ness is imperfect clocking.

What is positive and negative pulse stuffing? p345-346

Digital Hierarchy

DS=Digital signal level

Signals with appropriate format need not be voice signals

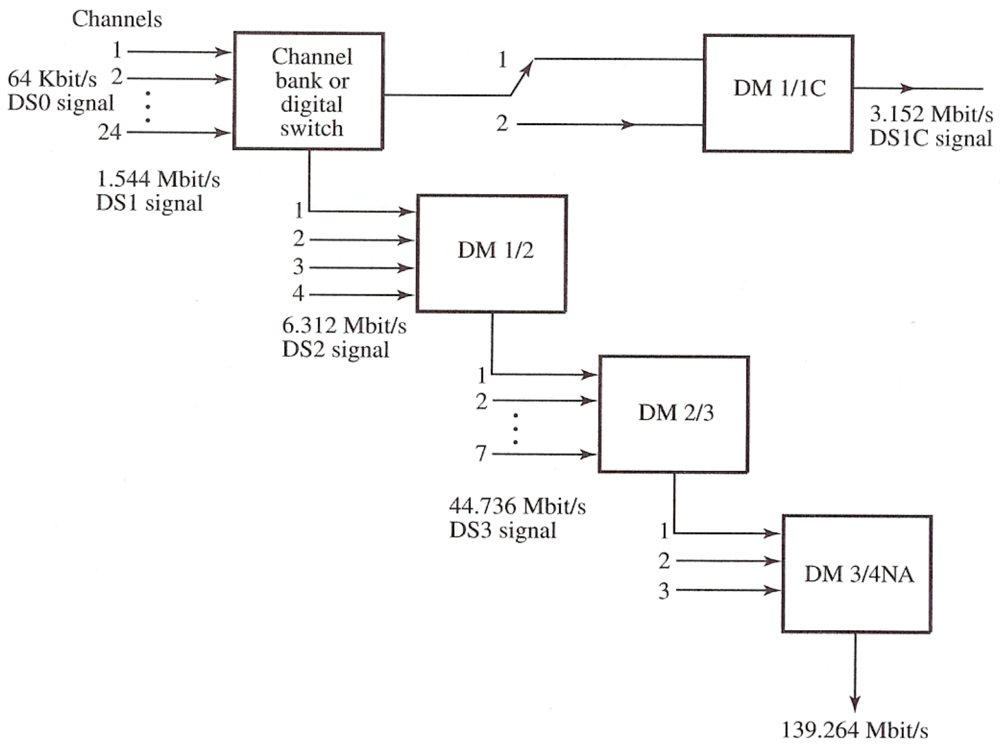


Figure 7.36 North American digital hierarchy (AT&T system).

CCITT= Consultative Committee on International Telephony and Telegraphy (now ITU: International Telecommunication Union)

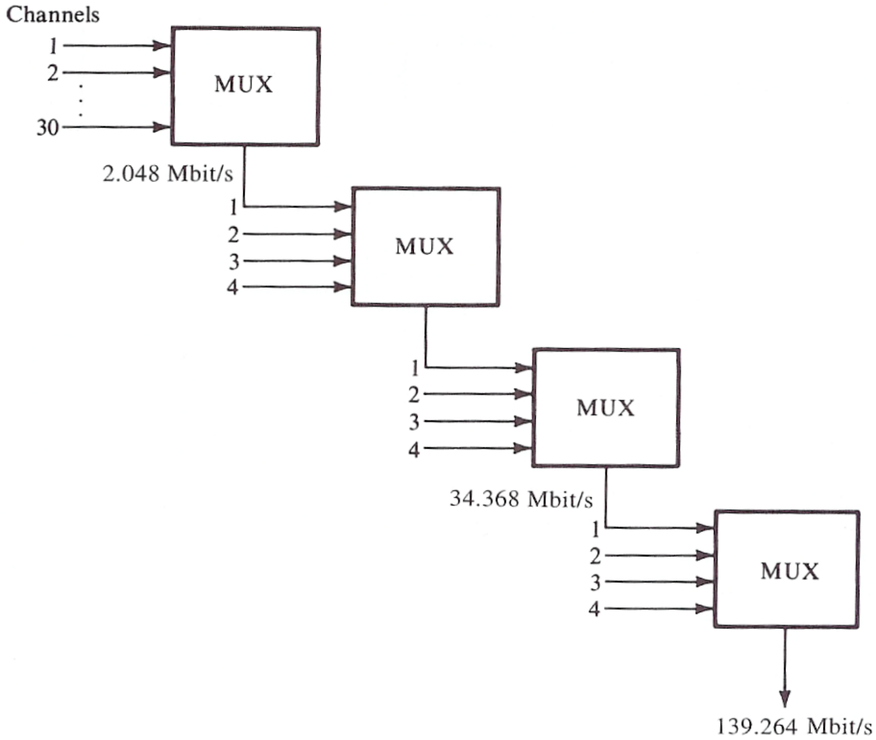


Figure 7.37 Digital hierarchy, CCITT recommendation.