# LINE CODING FOR DIGITAL COMMUNICATION

Prof. Dr. Qaysar Salih Mahdi
Mechatronics Engineering Department
Faculty of Engineering
Tishk International University

Digital Communications Systems

Week 6

Date : 3-7/3/2024

# Line Coding for Digital Communication

How do you transmit bits over a wire, RF, fiber?

- Line codes, many options
- Power spectrum of line codes, how much bandwidth do they take
- Clock signal and synchronization
- Common serial communications systems

Based on lecture notes from John Gill

# Line Coding for Digital Communication

- Goal is to transmit binary data (e.g., PCM encoded voice, MPEG encoded video, financial information)
- Transmission distance is large enough that communication link bandwidth is comparable to signal bandwidth.

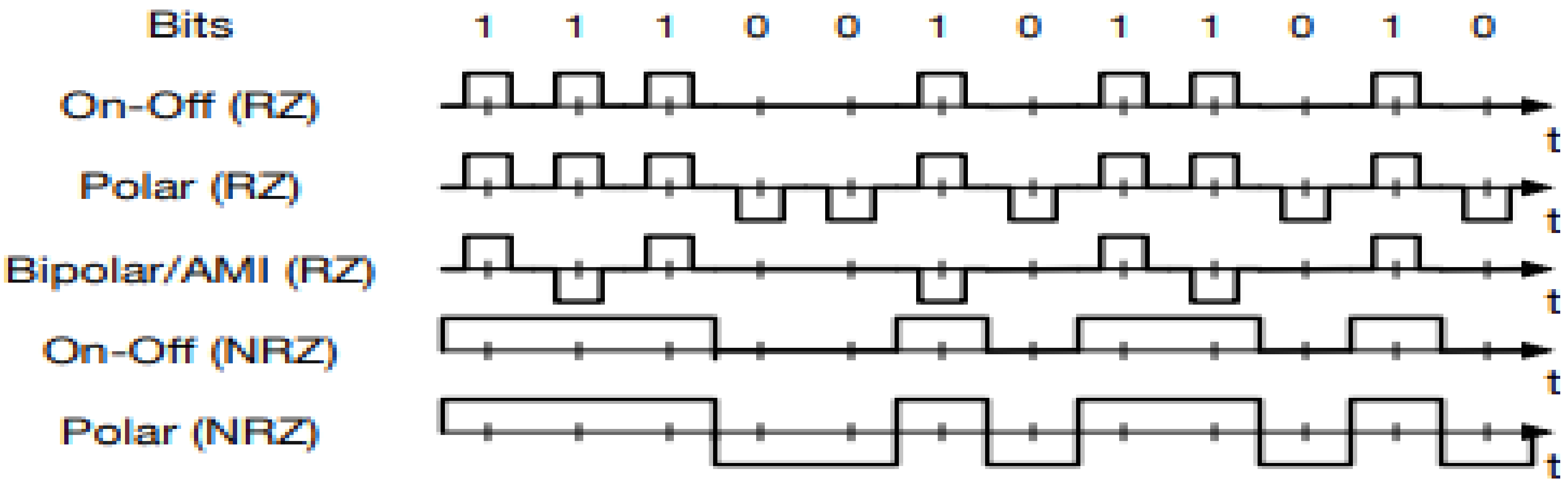
  In comparison connections between nearby logic gates have bandwidth greater than switching speed, so no line coding is needed. But longer connections use pulse shaping.
- Multiple links may be used, with regenerative repeaters

# Line Coding Requirements

- Small transmission bandwidth
- Power efficiency: as small as possible for required data rate and error probability
- Error detection/correction
- Suitable power spectral density, e.g., little low frequency content
- Timing information: clock must be extracted from data
- Transparency: all possible binary sequences can be transmitted

# Line Code Examples

Many different waveforms are used to transmit bits



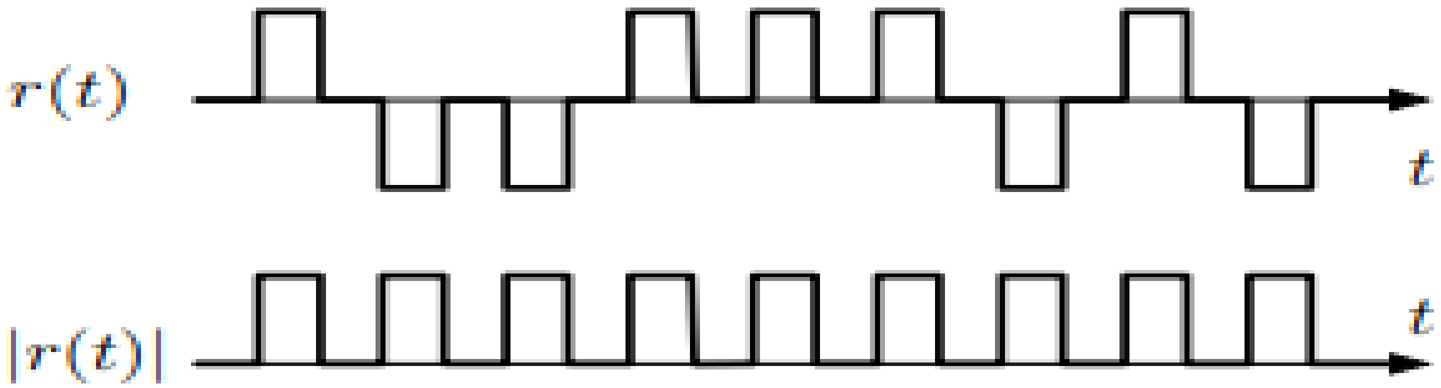

RZ = Return to Zero NRZ = Non-Return to Zero

Some of the features we would like are

- Easy to extract timing
- Zero average value

# Timing Signal

- We'd like to be able to extract the timing information from the signal easily.
- Consider the polar RZ $r(t)$ waveform shown below. If we take the absolute value

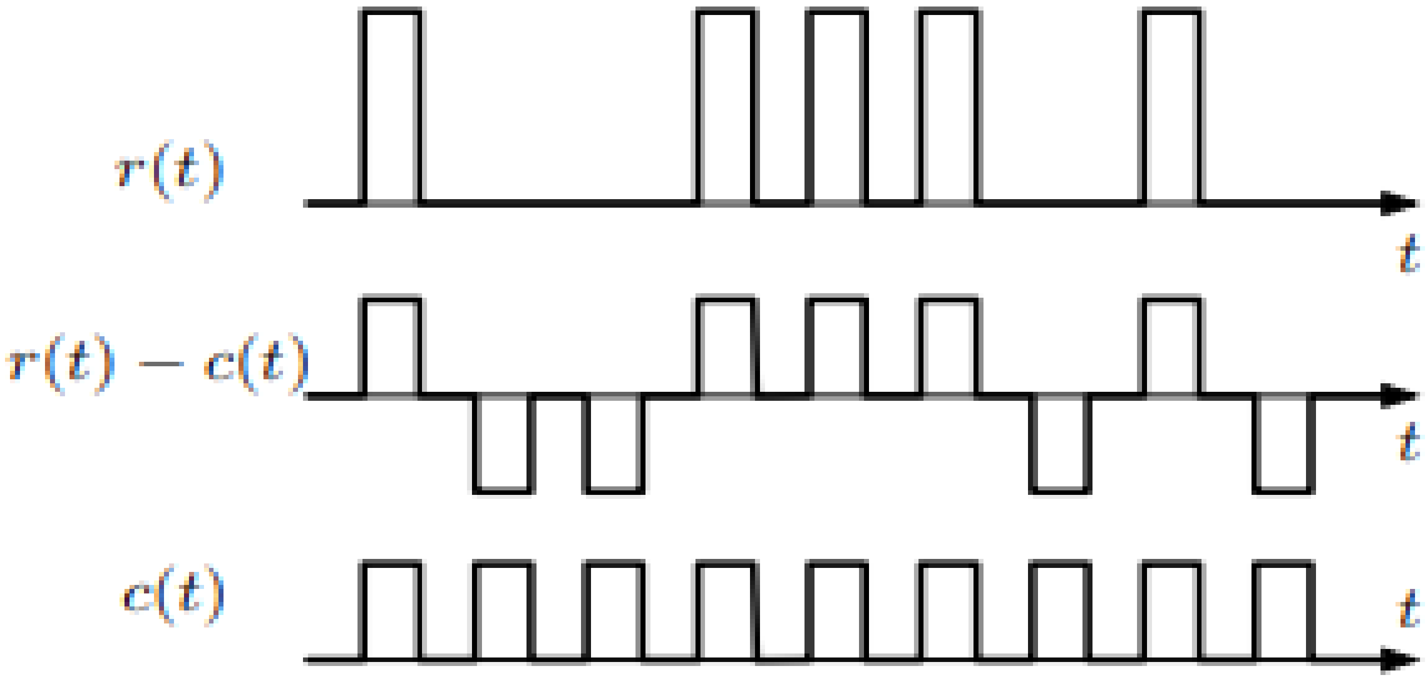



we get a timing signal.

- A line code where it is easy to extract the timing signal is called a *transparent code*. This is the reason many codes are designed the way they are.
- In practice the timing signal will be cleaned up by a narrowband bandpass filter before it is used to extract the bits in the input signal.

# Timing Signal

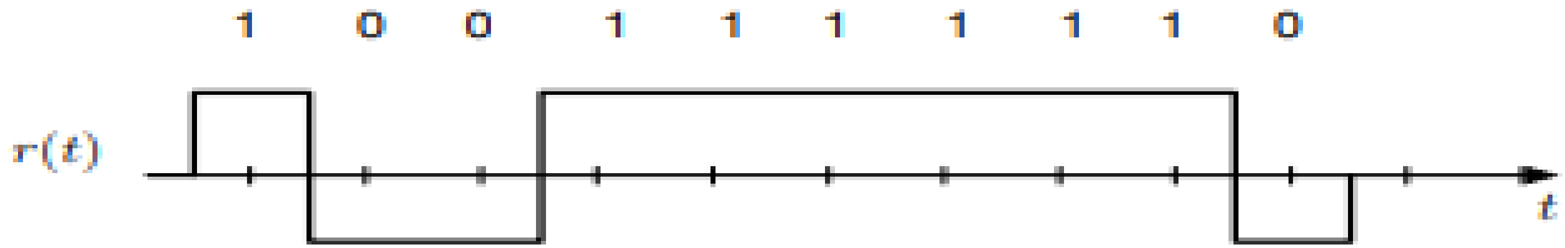
- Another is OOK, with either RZ or NRZ codes. The RZ case is shown here



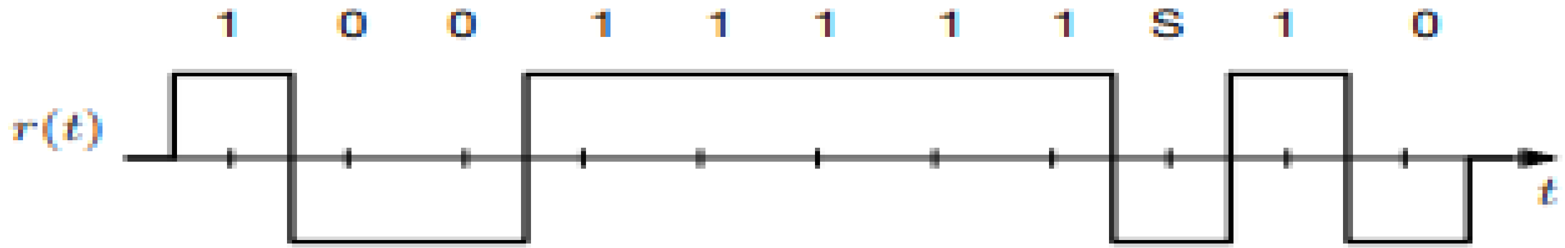

- The RZ OOK signal is a RZ binary signal plus a RZ clock signal.

# Timing and Bit Stuffing

- The NRZ codes can be more problematic. Long strings of 1's or 0's can cause the loss of synchronization.




- Many codes limit the number of consecutive runs of 1's or 0's, and force bit changes after a given number of bits.
- The example we'll look at in the next lab forbids 6 1's in a row, and adds an extra zero bit (bit stuff) after 5 1's. We'd transmit




When we decode the waveform, we'll have to watch for this, and throw the extra bits away.

# Power Spectral Density (PSD) of Line Codes

- We'd like to get as many bits/s across the channel as we can for a given channel bandwidth
- That will be limited by the power spectrum of the signal
- This will depend on
  - Pulse Rate (spectrum widens with pulse rate)
  - Pulse Shape (smoother or longer pulses have narrower PSD)
  - Pulse Distribution (line code)
- Today we'll look at the effect of the line code and several simple pulses
- Next time will look more carefully at pulse shaping

# Power Spectral Density (review)

- For an energy signal $g(t)$ the energy spectral density is the Fourier transform of the autocorrelation:

$$\psi_g(t) = R_g(t) = \int_{-\infty}^{\infty} g(u)g(u+t)\,du \;\Rightarrow\; |G(f)|^2 = \mathcal{F}\{R_g(t)\}$$

- For a power signal, autocorrelation and PSD are average over time.
- If we take a signal $g(t)$, and extract a segment of length $T$,

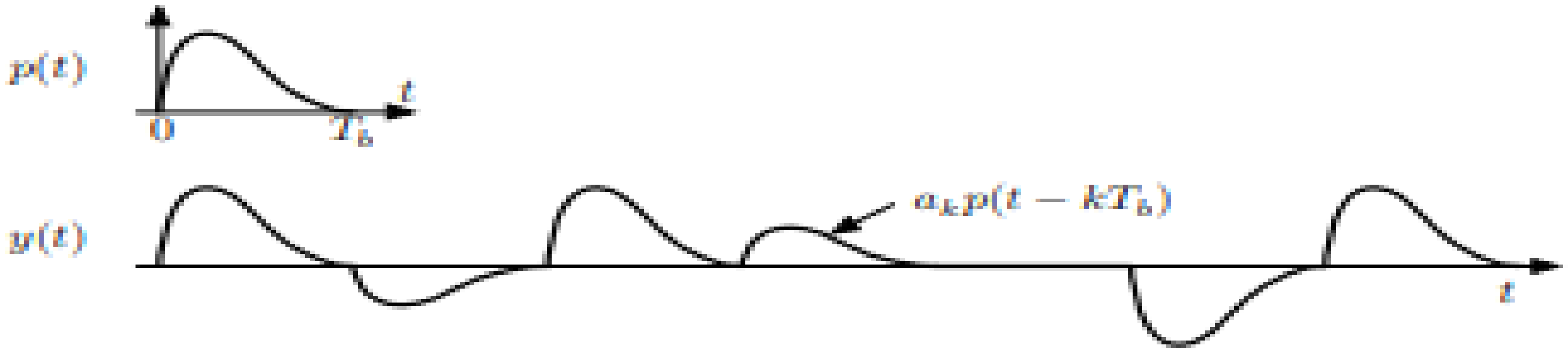
$$g_T(t) = \Pi(t/T)g(t) = \begin{cases} g(t) & |t| < T/2 \\ 0 & |t| > T/2 \end{cases}$$

- Then, if we let $T$ get large, the autocorrelation and the power spectral density are

$$R_g(t) = \lim_{T\to\infty} \frac{R_{g_T}(t)}{T} \;\Rightarrow\; S_g(f) = \lim_{T\to\infty} \frac{|G_T(f)|^2}{T}$$

# PSD of Line Codes

- The PSD of a line code depends on the shapes of the pulses that correspond to digital values. Assume the pulses $p(t)$ are amplitude modulated (PAM).




- The transmitted signal is the sum of weighted, shifted pulses.

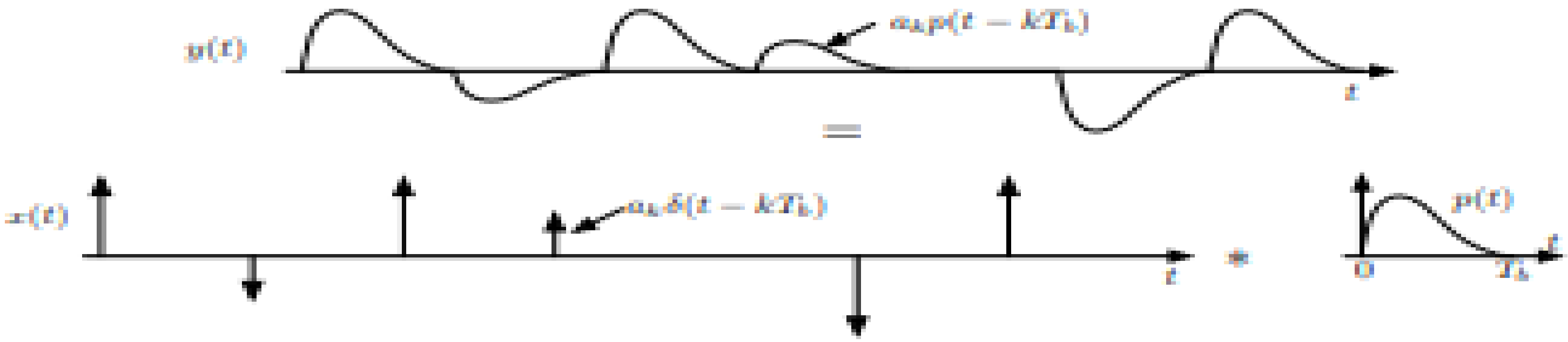
$$y(t) = \sum_{k=-\infty}^{\infty} a_k p(t - kT_b)$$

where $T_b$ is spacing between pulses.

- Pulse may be wider than $T_b$, which leads to inter-symbol interference (ISI), which we will look at next class.

# PSD of Line Codes (cont.)

- PSD depends on pulse shape, rate, and digital values $\{a_k\}$.
- We can simplify analysis by representing $y(t)$ as impulse train convolved with $p(t)$




- Then $Y(f) = P(f)X(f)$, and the PSD of $y(t)$ is

$$S_y(f) = |P(f)|^2 S_x(f)$$

- $P(f)$ depends only on the pulse, independent of digital values or rate.
- $S_x(f)$ increases linearly with rate $1/T_b$ and depends on distribution of values of $\{a_k\}$. E.g., $a_k = 1$ for all $k$ has narrower PSD.

# PSD of Impulse Train

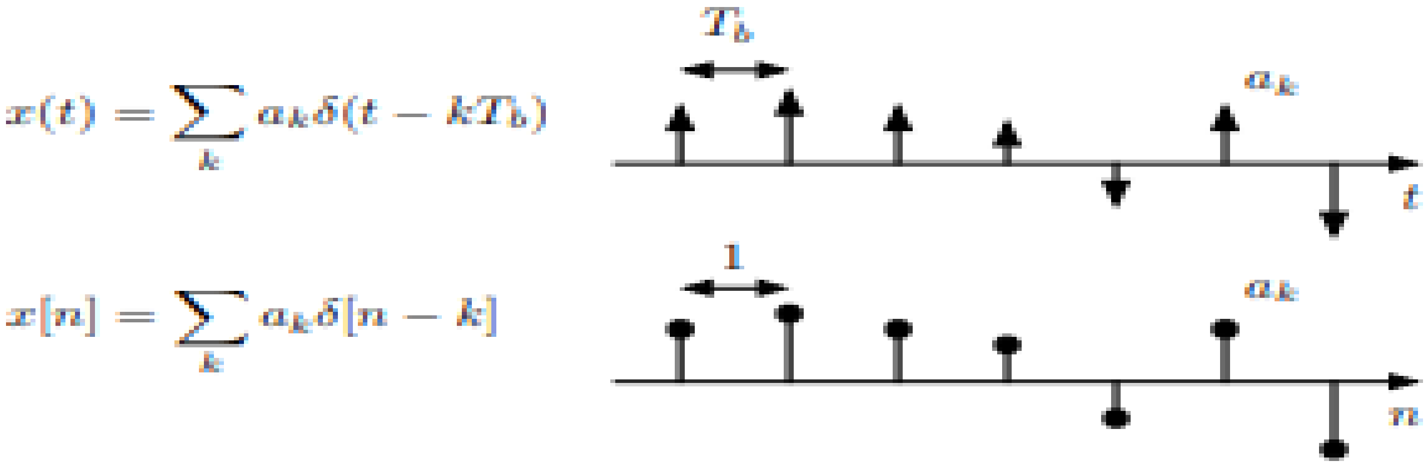
We like to find the autocorrelation of an impulse train

$$x(t) = \sum_{k=-\infty}^{\infty} a_k \delta(t - kT_b)$$

In discrete time the signal is

$$x[n] = \sum_{k=-\infty}^{\infty} a_k \delta[n - k]$$

This is illustrated below

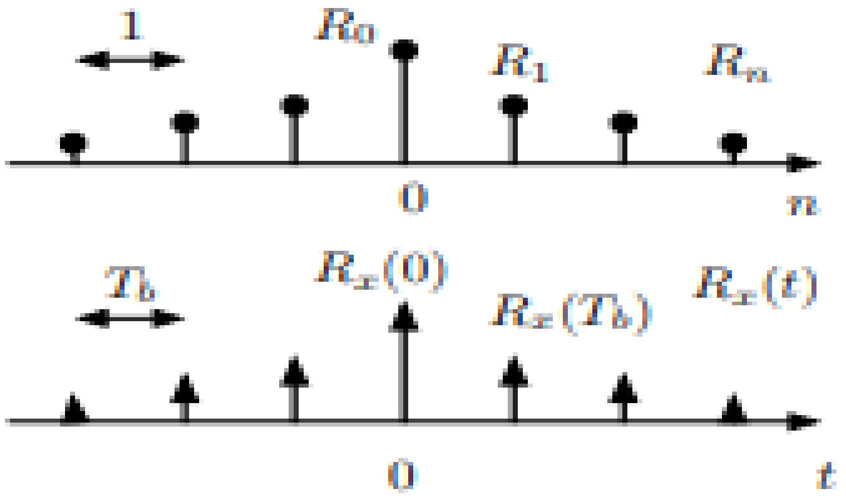


The autocorrelation in discrete time is

$$R_n = \lim_{N\to\infty} \frac{1}{2N} \sum_{k=-N}^{N} a_k a_{k-n}$$

The continuous time autocorrelation is then

$$R_x(t) = \frac{1}{T_b} \sum_{n=-\infty}^{\infty} R_n \delta(t - nT_b)$$



The PSD is then

$$\mathcal{F}\{R_x(t)\} = S_x(f) = \frac{1}{T_b}\sum_{n=-\infty}^{\infty} R_n e^{-jn2\pi f T_b}$$

Hence, if we know the discrete time autocorrelation of the transmitted bits, we know the continuous time power spectral density.
Then, given a PAM pulse sequence

$$y(t) = \sum_{k=-\infty}^{\infty} a_k p(t - kT_b)$$

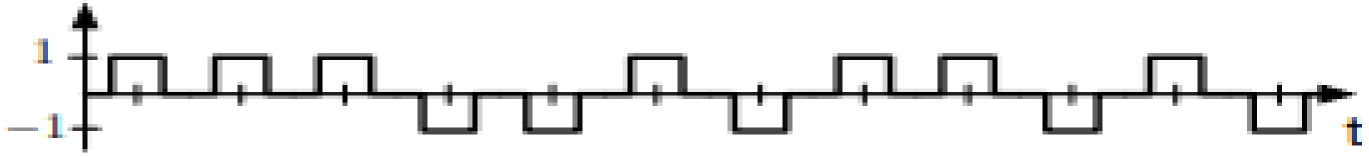
the PSD of the encoded signal is

$$S_y(f) = |P(f)|^2 \left(\frac{1}{T_b}\sum_{n=-\infty}^{\infty} R_n e^{-jn2\pi f T_b}\right)$$

We will look at the effect of each of the terms on the right.

# PSD of Polar Signaling

- The polar signally waveform looks like this:




- $a_k$ and $a_{k+n}$ are independent and equally likely for $n \neq 0$
- With zero shift

$$R_0 = \lim_{N\to\infty} \frac{1}{2N} \sum_{k=-N}^{N} a_k^2 = \lim_{N\to\infty} \frac{1}{2N} \sum_{k=-N}^{N} 1 = 1$$

- If there is a shift, the $a_k a_{k+n}$ is equally likely to be $\pm 1$, and

$$R_n = \lim_{N\to\infty} \frac{1}{2N} \sum_{k=-N}^{N} a_k a_{k+n} = 0$$

- As a result, only the zero shift term comes through, and

$$S_y(f) = \frac{|P(f)|^2}{T_b} R_0 = \frac{|P(f)|^2}{T_b}$$

# Polar DIgnalling PSD Examples:

- NRZ (100% pulse)

$$\begin{aligned} p(t) &= \Pi(t/T_b) \\ P(f) &= T_b \,\text{sinc}(\pi T_b f) \\ |P(f)|^2 &= T_b^2 \,\text{sinc}^2(\pi T_b f) \end{aligned}$$

- RZ half-width:

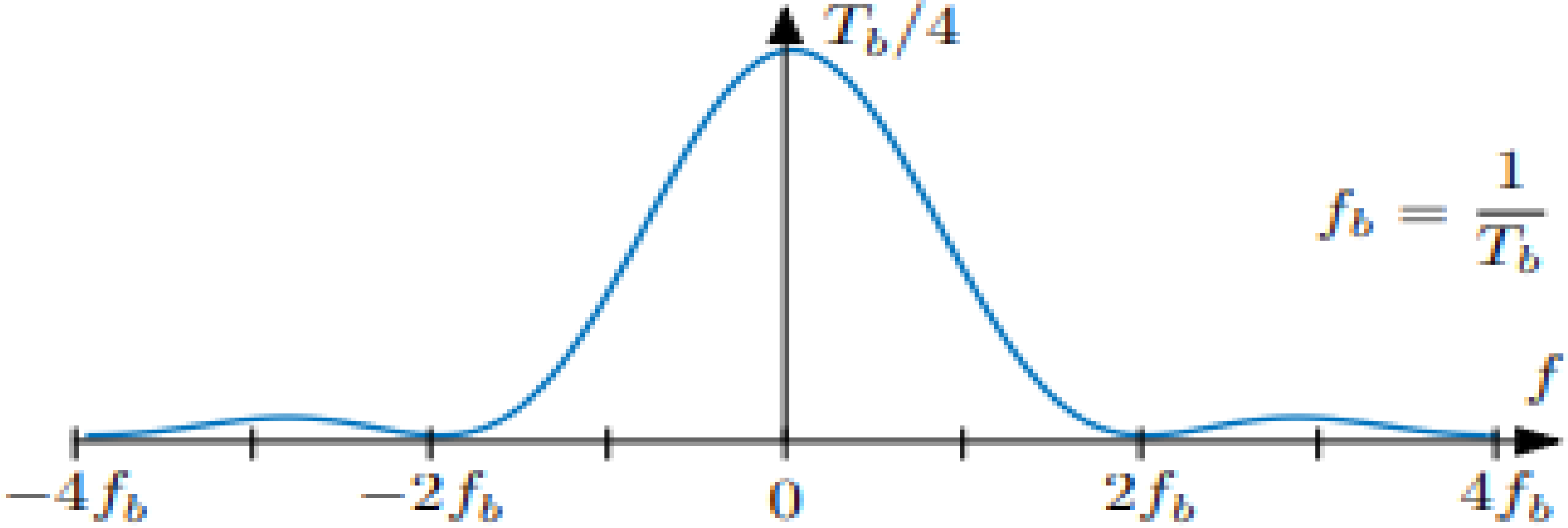
$$\begin{aligned} p(t) &= \Pi(t/(T_b/2)) \\ P(f) &= \tfrac{1}{2} T_b \,\text{sinc}(\tfrac{1}{2}\pi T_b f) \\ |P(f)|^2 &= \tfrac{1}{4} T_b^2 \,\text{sinc}^2(\tfrac{1}{2}\pi T_b f) \end{aligned}$$

- RZ half-width has twice the spectral width, as expected

# PSD of Polar Signaling (Half-Width Pulse)

For the RZ pulse,

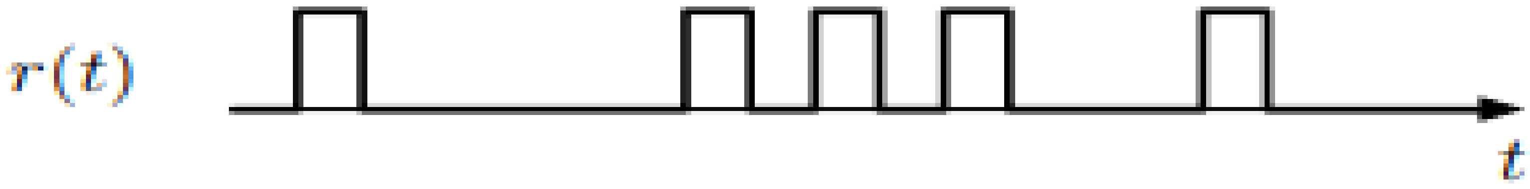
$$S_y(f) = \frac{|P(f)|^2}{T_b} = \frac{\frac{1}{4}T_b^2 \operatorname{sinc}^2(\frac{1}{2}\pi T_b f)}{T_b} = \frac{T_b}{4}\operatorname{sinc}^2\left(\frac{\pi T_b f}{2}\right)$$




The bandwidth $2f_b$ is $4\times$ theoretical minimum of 2 bits/Hz/sec.

# PSD of On-Off Keying

- OOK looks like




- As we saw earlier, OOK is shifted polar signaling:

$$y_{\text{on-off}}(t) = \tfrac{1}{2}(1 + y_{\text{polar}}(t))$$

- $R_0$ is $\frac{1}{2}$ because half the time the signals are 1, and half the time they are zero,

$$R_0 = \left(\frac{1}{2}\right) 1 + \left(\frac{1}{2}\right) 0 = \frac{1}{2}$$

- The issue is with all the higher order terms. If we look at $R_n$, 1/4 of the time two bits separated by $n$ are both 1, 1/2 the time one is one and one is zero, and 1/4 the time they are both zero. The autocorrelation is then

$$R_n = \left(\frac{1}{4}\right) 1 + \left(\frac{1}{2}\right) 0 + \left(\frac{1}{4}\right) 0 = \frac{1}{4}$$

  This contributes a constant term of 1/4 for any $n \neq 0$

The expression for the PSD is

$$S_y(f) = |P(f)|^2 \left( \frac{1}{T_b} \sum_n R_n e^{j2\pi f n T_b} \right)$$

Since $R_0 = \frac{1}{2}$ and $R_n = \frac{1}{4}$ for $n \neq 0$,

$$\begin{aligned} S_y(t) &= \frac{|P(f)|^2}{T_b} \left( \frac{1}{2} + \frac{1}{4} \sum_{n \neq 0} e^{j2\pi f n T_b} \right) \\ &= \frac{|P(f)|^2}{T_b} \left( \frac{1}{4} + \frac{1}{4} \sum_{n} e^{j2\pi f n T_b} \right) \\ &= \frac{|P(f)|^2}{4T_b} \left( 1 + \frac{1}{T_b} \sum_{n} \delta(f - n/T_b) \right) \end{aligned}$$

Hence the constant term in the autocorrelation leads to impulses in the power spectrum.

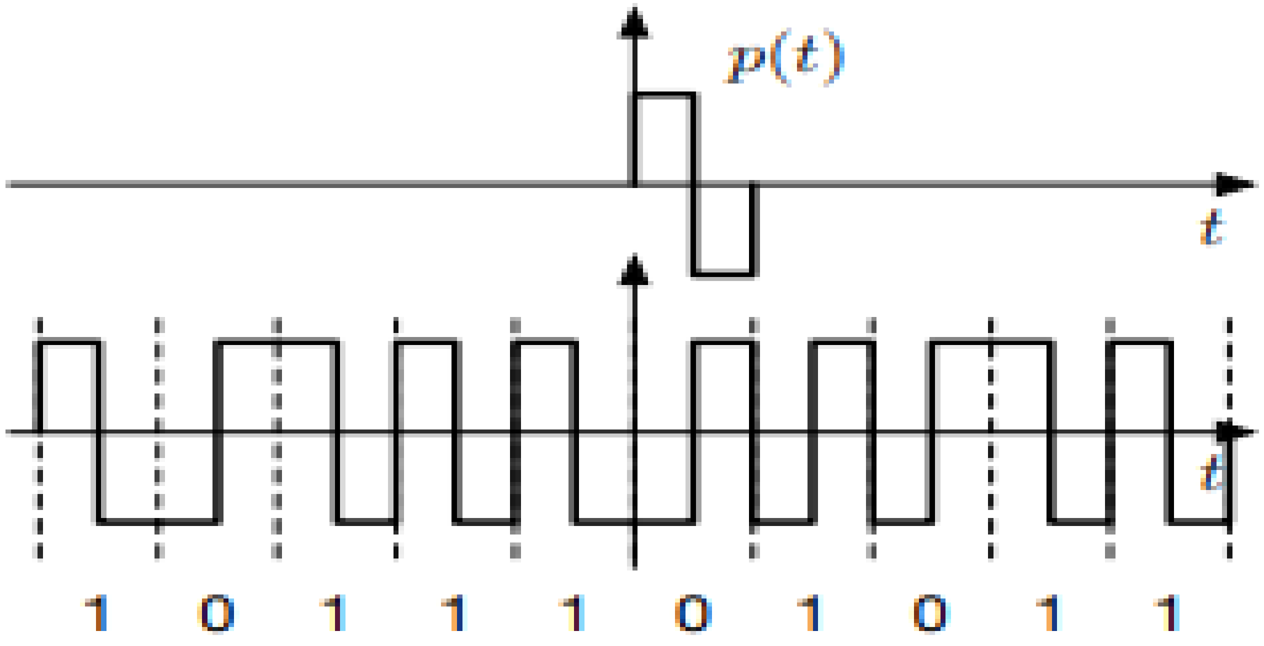
- The impulses in the power spectrum are extra frequency components that don't carry information.
- This is a DC current that just heats up the wires! This makes sense, this is a sequence of positive or zero pulses. There willl be a net current.
- We can eliminate impulses by using a pulse $p(t)$ with

$$P\left(\frac{n}{T_b}\right) = 0 , \quad n = 0, \pm 1, \pm 2, \ldots$$

- Overall, on-off is inferior to polar. For a given average power, noise immunity is less than for bipolar signaling.
- However, OOK is very simple (you just have to gate an oscillator on and off), so it shows up widely in lower power systems (like key fobs) or very high frequency systems (where modulation can be difficult).

# Split Phase (Manchester) Encoding

- As we saw with OOK, line codes with a DC value lower performance, because a DC component with no information heats up the wires.
- There are two alternatives
  - Use pulses $p(t)$ that have zero average value (split phase, or Manchester encoding, now)
  - Use sequences of pulses that have average values that go to zero (bipolar signaling, next)
- Split phase encoding looks like this:



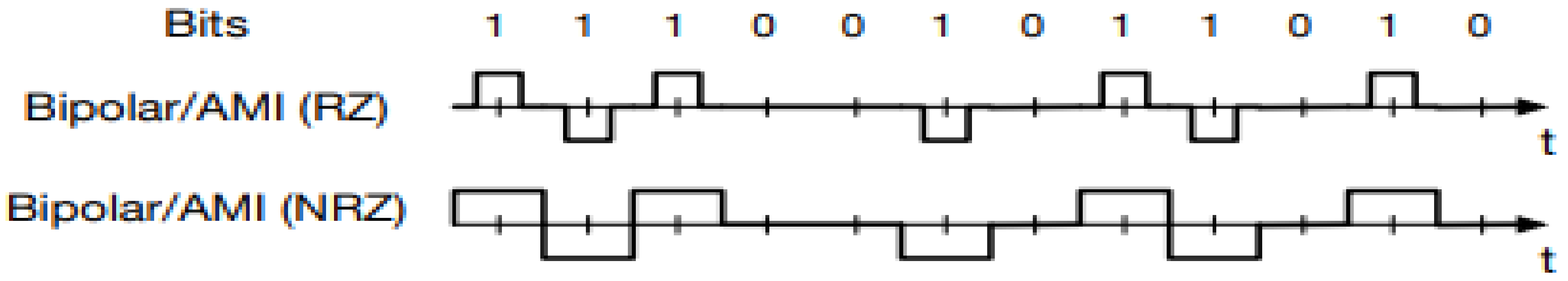
- By the same reasoning as we used for polar signalling, $R_0 = 1$, and $R_n = 0$ for $n \neq 0$, since offset pulses are independent, and their product is just as like to be $\pm 1$.
- The PSD is then

$$
\begin{aligned}
S_y(f) &= |P(f)|^2 \left( \frac{1}{T_b} \sum_{n=-\infty}^{\infty} R_n e^{-jn2\pi f T_b} \right) \\
&= |P(f)|^2 \frac{1}{T_b}
\end{aligned}
$$

- In addition, $P(0) = 0$, which we'll see in a few slides
- It is also very easy to get the timing signal from the coding waveform.
- This was first introduced with the development of magnetic disk drives in the late 1940's and early 1950's. Read heads were only sensitive to transitions in magnetization, so this approach guaranteed at least one transition per bit.
- This is widely used in wired ethernet. Also common in RF, particularly in low power near field RF (NFRF) devices.

# Alternate Mark Inversion (Bipolar) Signaling

AMI encodes $0$ as 0 V and $1$ as $+V$ or $-V$, with alternating signs.




AMI was used in early PCM systems.

- Eliminates DC build up on cable.
- NRZ bipolar reduces bandwidth compared to polar RZ.
- Guarantees transitions for timing recovery with long runs of ones.
- Provides error detecting; every bit error results in bipolar violation.

AMI is also called *bipolar* and *pseuodternary*.

# PSD of AMI Signaling

- If the data sequence $\{a_k\}$ consists of equally likely and independent 0s and 1s, then the autocorrelation function of the sequence is for $R_0$ is

$$R_0 = \left(\frac{1}{2}\right) 1 + \left(\frac{1}{2}\right) 0 = \frac{1}{2}$$

- For $R_{\pm 1}$ there are four possibilities, 11, 01, 10, and 00. Since the signs change for successive 1's, and all the others have autocorrelations of zero,

$$R_{\pm 1} = \left(\frac{1}{4}\right)(-1) + \left(\frac{3}{4}\right) 0 = -\frac{1}{4}$$

- For $n = 2$, the various permutations of 1's and 0's are either zero, or cancel out and give $R_2 = 0$. This continues for $n > 2$.

- Therefore

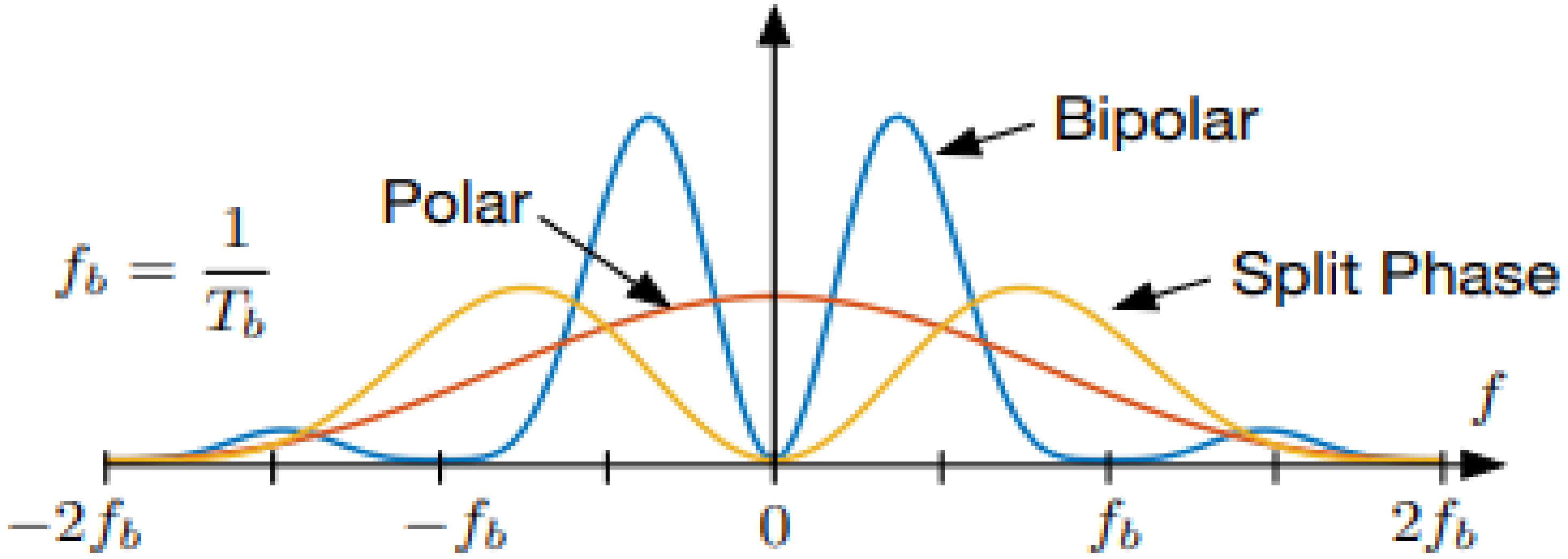
$$
\begin{aligned}
S_y(f) &= \frac{|P(f)|^2}{T_b} \sum_n R_n e^{j2\pi n f T_b} \\
&= \frac{|P(f)|^2}{T_b} \left( R_0 + R_{-1} e^{j2\pi f T_b} + R_1 e^{-j2\pi f T_b} \right) \\
&= \frac{|P(f)|^2}{T_b} \left( \frac{1}{2} - \left( \frac{1}{4} \right) 2\cos(2\pi f T_b) \right) \\
&= \frac{|P(f)|^2}{2T_b} (1 - \cos(2\pi f T_b)) \\
&= \frac{|P(f)|^2}{T_b} \sin^2(\pi T_b f)
\end{aligned}
$$

This PSD falls off faster than $\mathrm{sinc}(\pi T_b f)$.

- The PSD has a null at DC, which aids in transformer coupling.

# PSD Comparison

The PSDs of RZ polar, split phase, and NRZ bipolar (AMI) line codes look like this:




These all allow easy clock recovery.

# Data Transfer in Digital System

- In a synchronous digital system, a common clock signal is used by all devices.

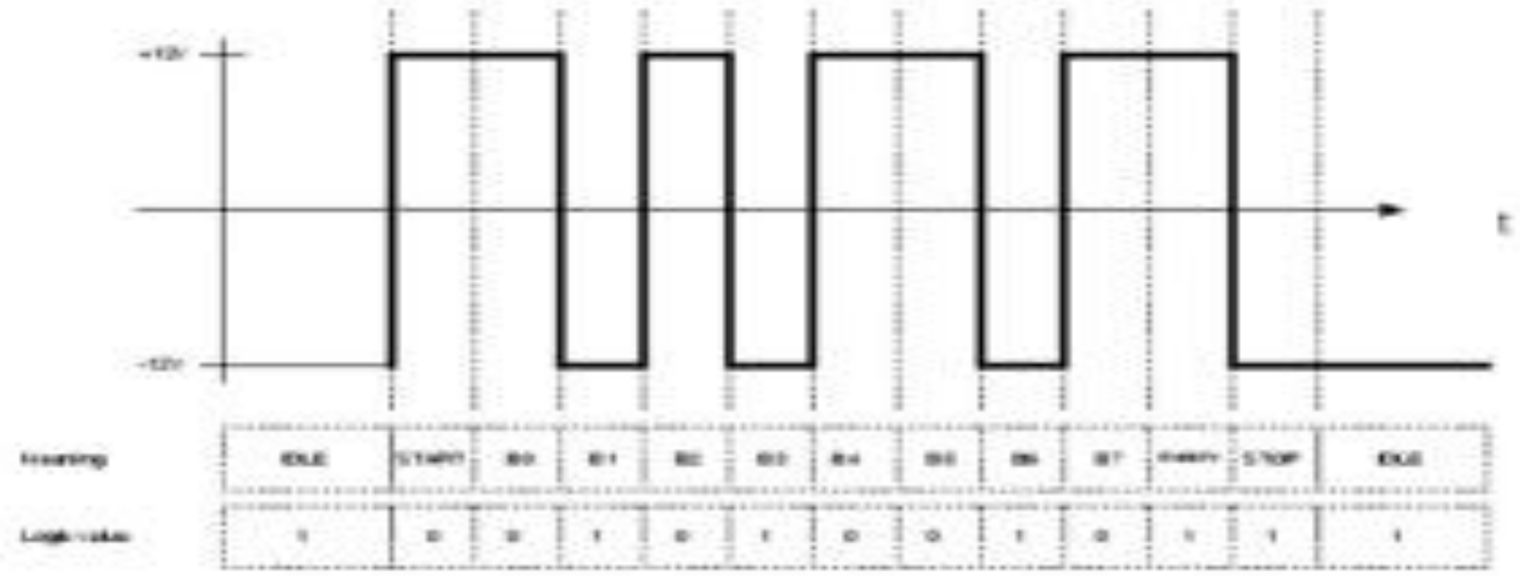
  $$\text{data} + \text{clock}$$

  Multiple data signals can be transmitted in parallel using a single clock signal.
- Serial peripheral communication schemes (RS-232, USB, FireWire) use various clock extraction methods
  - RS-232 is asynchronous with (up to) 8 data bits preceded by a start bit (0) and followed by optional parity bit and stop bit (1); clock recovery by "digital phase-locked loop"
  - USB needs a real phase-locked loop and uses *bit stuffing* to ensure enough transitions
  - FireWire has differential data and clock pairs; clock transitions only when data does not

# Serial Communication: RS-232 Signaling

RS-232 is a standard for *asynchronous* serial communication.

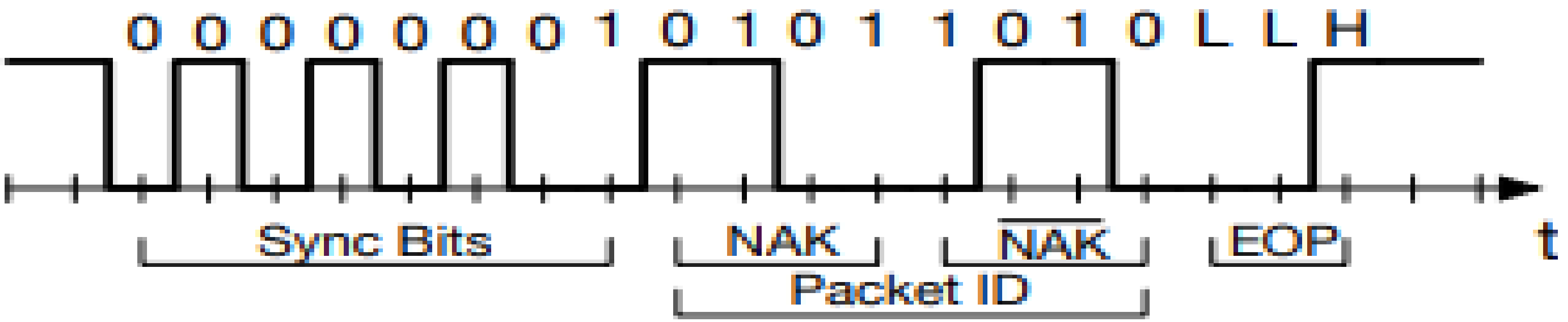
RS232 Transmission of the letter 'J'




- 1 is -12V, 0 is 12V
- Parity bit to detect errors
- Stop and start bits to help synchronization
- Each transition resynchronizes the receiver's bit clock.

# Serial Communications: USB

- For USB 1.1 a data packet consists of
  - NRZI Encoding: 0 is a transition, 1 stays the same
  - Synchronization bits 00000001
  - The 4 bit packet ID followed by its complement (little endian)
  - Payload, if any, with CRC
  - End of packet (EOP) sequence LLH
- For a NAK packet (1010) this looks like




- Bit stuffing is used after 6 bits in a row.
- USB is actually differential pair (plus on one line, minus on the other).
- We're only showing the positive line.

# Next Classes

- Pulse shaping and intersymbol interference (ISI)
- Simple FSK Packet Radio (APRS)
- Digital encoding, M-ary digital encoding schemes
- : Error Correction and CRC codes
- Final project discussion
-
- : Radar, Doppler, Ultrasound
- Spread Spectrum, CDMA, GPS