Lecture 6A: Line Coding

John M Pauly

February 10, 2026

Line Coding for Digital Communication

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

Line coding refers to how the sequence of pulses that encode bits are sent over a channel.

In the next few classes we will look more at the pulses themselves.

Most of this was developed for baseband systems (phones, wired ethernet), but the same techniques show up in carrier based systems.

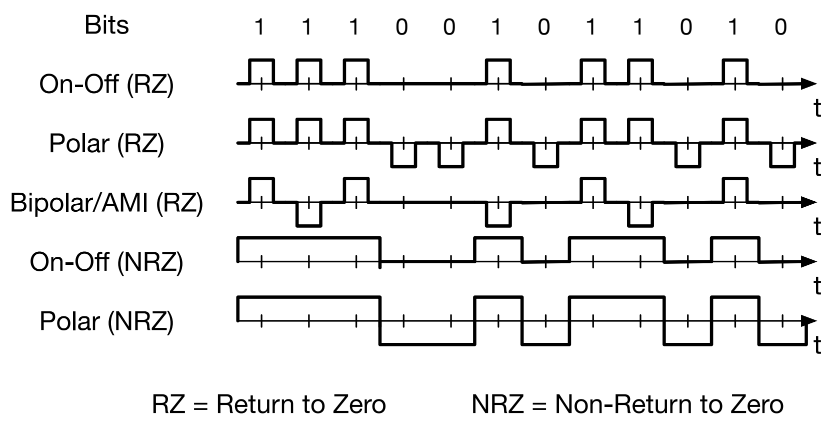
Line Coding Requirements

Some things we'd like when we send a stream of bits:

- Small transmission bandwidth
- Power efficiency: want power as small as possible for required data rate and error probability
- Error detection/correction
- Limited 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

Some of the most common methods for transmitting bits:

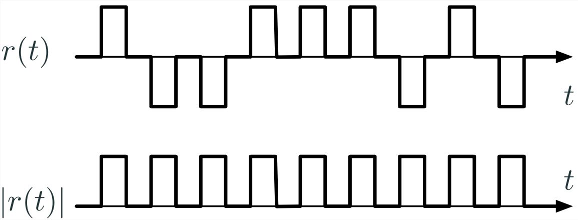


We'd like it to be easy to extract timing, and also have zero average value.

Timing Signal

We'd like to be able to extract the timing information from the signal easily.

Consider the polar RZ $r(t)$ waveform. If we take the absolute value

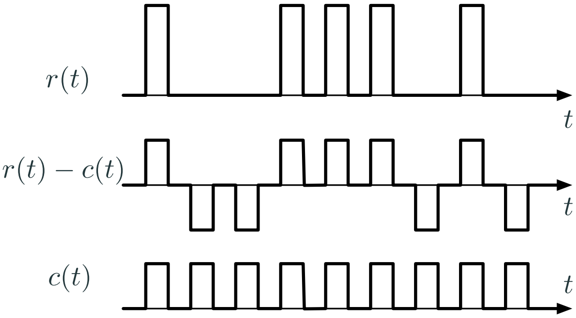


we get a timing signal. A line code where it is easy to extract the timing signal is called a *transparent code*. This is the reason many codes are designed the way they are.

We will look at extracting the timing signal later. This is an important separate topic.

Timing Signal

Another is on-off keying (OOK), with either RZ or NRZ codes. The RZ case is shown here

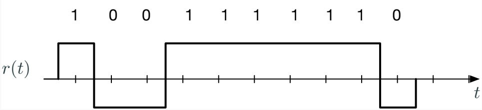


The RZ OOK signal is:

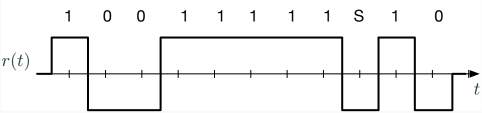
- a RZ binary signal, plus
- a RZ clock signal.

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 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.



When decoding the waveform we 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

This can be combined with the pulse shaping that we will talk about in two classes.

Power Spectral Density (review)

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

$$R_{yy}(t) = \int_{-\infty}^{\infty} y(u)y(u+t)du \implies |Y(f)|^2 = \mathcal{F}\{R_{yy}(t)\}$$

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,

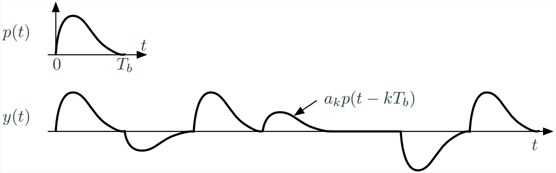
$$y_T(t) = \Pi(t/T)y(t)$$

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

$$R_{yy}(t) = \lim_{T\to\infty} \frac{R_{y_T y_T}(t)}{T} \implies S_y(f) = \lim_{T\to\infty} \frac{|Y_T(f)|^2}{T}$$

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.

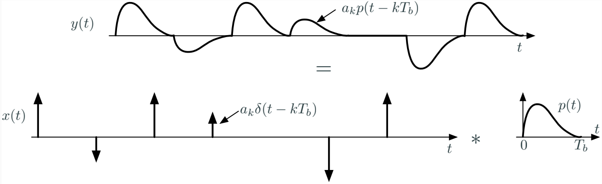
$$y(t) = \sum_{k=-\infty}^{\infty} a_k p(t - kT_b)$$

where T_b is spacing between pulses.

Pulses may be wider than T_b, which leads to inter-symbol interference (ISI), as we will talk about in two classes.

PSD of Line Codes (cont.)

PSD depends on pulse shape, rate, and values $\{a_k\}$. We can represent $y(t)$ as an amplitude weighted impulse train $x(t)$ convolved with a pulse $p(t)$



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

$$\boxed{S_y(f) = |P(f)|^2 S_x(f)}$$

$P(f)$ 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

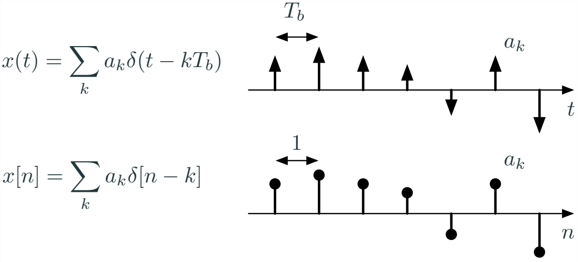
We'll start by finding the autocorrelation of the impulse train

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

In discrete time the signal is

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

This is illustrated below



PSD of Impulse Train (cont.)

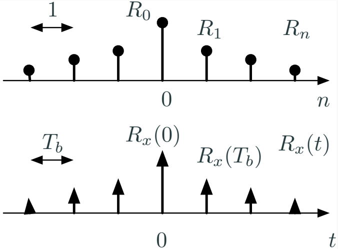
The autocorrelation in discrete time is

$$R_{xx}[n] = \lim_{N\to\infty} \frac{1}{2N} \sum_{k=-N}^{N} a_k a_{k-n}$$

The continuous time autocorrelation is then

$$R_{xx}(t) = \frac{1}{T_b} \sum_{n=-\infty}^{\infty} R_{xx}[n]\delta(t - nT_b)$$

where each discrete time sample is represented by an impulse



PSD of Line Codes (cont.)

The PSD is then

$$\mathcal{F}\left\{R_{xx}(t)\right\} = S_x(f) = \frac{1}{T_b}\sum_{n=-\infty}^{\infty} R_{xx}[n]e^{-jn2\pi f T_b}$$

Hence, if we know the discrete time autocorrelation of the transmitted bits, we know the continuous time power spectral density of the pulse train.

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

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

Example Line Code Power Spectra

We'll look at the power spectra of several common line coding methods.

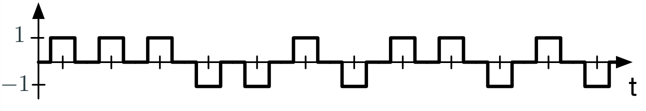
- Polar signaling
- On-off keying
- Split-phase, or Manchester, encoding
- Bipolar, or alternate mark inversion, signaling

Each of these trade off the problems of synchronization, DC value, and spectral efficiency in different ways.

For now we'll assume the pulses are square pulses for simplicity. We look at more sophiscated pulses in a couple of classes.

PSD of Polar Signaling

The polar signaling waveform looks like this:



a_k and a_{k+n} are independent and equally likely.

With zero shift

$$R_{xx}[0] = \lim_{N\to\infty} \frac{1}{2N} \sum_{k=-N}^{N} a_k^2 = \lim_{N\to\infty} \frac{1}{2N} \sum_{k=-N}^{N} 1 = 1$$

With a shift $a_k a_{k+n}$ is equally likely to be ± 1, and

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

PSD of Polar Signaling, (cont.)

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

$$S_y(f) = |P(f)|^2 \left(\frac{1}{T_b} R_{xx}[0] \right) = \frac{1}{T_b} |P(f)|^2$$

This reduces to the power spectrum of a single pulse!

This makes sense. An individual pulse of length T_b has a wider bandwidth than a signal sampled at a time T_b.

The actual power spectrum of the sequence of bits will just multiply the pulse power spectrum, $|P(f)|^2$

Polar Signaling PSD Examples:

NRZ (100% pulse)

$$\begin{aligned} p(t) &= \Pi\left(\frac{t}{T_b}\right) \\ P(f) &= T_b \text{sinc}(T_b f) \\ |P(f)|^2 &= T_b^2 \text{sinc}^2(T_b f) \end{aligned}$$

RZ half-width:

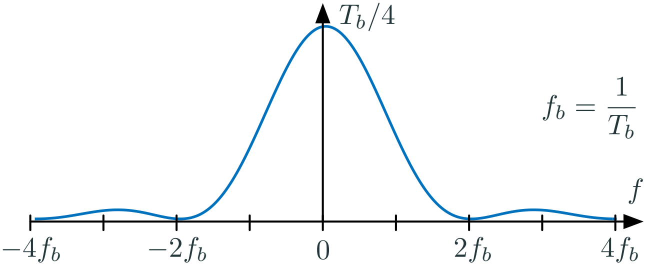
$$\begin{aligned} p(t) &= \Pi\left(\frac{t}{T_b/2}\right) \\ P(f) &= \frac{T_b}{2} \text{sinc}\left(\frac{T_b}{2} f\right) \\ |P(f)|^2 &= \frac{T_b^2}{4} \text{sinc}^2\left(\frac{T_b}{2} f\right) \end{aligned}$$

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

PSD of Polar Signaling (Half-Width Pulse)

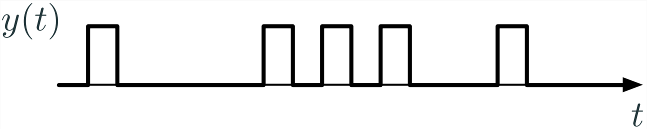
For the RZ pulse,

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



PSD of On-Off Keying

OOK looks like



As we saw earlier, OOK is shifted polar signaling:

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

$R_{xx}[0]$ is $\frac{1}{2}$ because half the time the signals are 1, and half the time they are zero,

$$R_{xx}[0] = \left(\frac{1}{2}\right)1 + \left(\frac{1}{2}\right)0 = \frac{1}{2}$$

PSD of On-Off Keying, (cont.)

The issue is with all the higher order terms.

If we look at $R_{xx}[n]$,

- $1/4$ 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

$$R_{xx}[n] = \left(\frac{1}{4}\right)1 + \left(\frac{1}{2}\right)0 + \left(\frac{1}{4}\right)0 = \frac{1}{4}$$

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

PSD of On-Off Keying, (cont.)

The expression for the PSD is

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

Since $R_{yy}[0] = \frac{1}{2}$ and $R_{yy}[n] = \frac{1}{4}$ 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.

PSD of On-Off Keying, (cont.)

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 will be a net current.

As we we see shortly, we can eliminate impulses by using a pulse $p(t)$ with

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

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 or Manchester Encoding

Line codes with a DC value lower performance, because a DC component with no information.

Recall that the power spectrum of the transmit waveform is

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

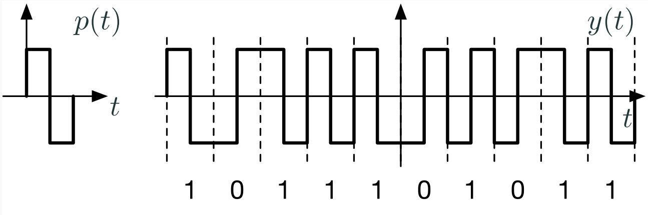
We can eliminate the DC value with either the spectrum of the pulse $P(f)$, or with the the power spectrum of the line code $S_x(f)$.

Our options are

- 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 or Manchester Encoding (cont.)

Split phase encoding looks like this:



This was first introduced for magnetic disk drives in the 1950's. Read heads were only sensitive to magnetization transitions. This guaranteed at least one transition per bit.

This is widely used in wired ethernet and RF, particularly in low power near field RF (NFRF) devices. If you see an binary waveform, it is probably Manchester encoded.

Split Phase or Manchester Encoding, (cont.)

By the same reasoning as we used for polar signaling, $R_{xx}[0] = 1$, and $R_{xx}[n] = 0$ for $n \neq 0$, since offset pulses are independent, and their product is just as like to be ± 1.

The PSD is then

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

where $P(f)$ is the spectrum of the Manchester pulse.

In addition, $P(0) = 0$, which we'll see in a few slides

Note that it very easy to get the timing signal from the Manchester encoded waveform.

Manchester Pulse Power Spectrum

For convenience, we center the Manchester pulse at zero

$$p(t) = \Pi\left(\frac{t + T_b/4}{T_b/2}\right) - \Pi\left(\frac{t - T_b/4}{T_b/2}\right)$$

The spectrum then is

$$\begin{aligned} P(f) &= \frac{T_b}{2}\text{sinc}\left(\frac{T_b f}{2}\right)\left(e^{+j2\pi f T_b/4} - e^{-j2\pi f T_b/4}\right) \\ &= \frac{T_b}{2}\text{sinc}\left(\frac{T_b f}{2}\right)\left(2j\sin\left(\frac{2\pi T_b f}{4}\right)\right) \\ &= jT_b\,\text{sinc}\left(\frac{T_b f}{2}\right)\sin\left(\frac{\pi T_b f}{2}\right) \end{aligned}$$

and the power spectrum is

$$|P(f)|^2 = T_b^2\,\text{sinc}^2\left(\frac{T_b f}{2}\right)\sin^2\left(\frac{\pi T_b f}{2}\right)$$

Manchester Line Code Power Spectrum

Recall that $S_x(f)$ is the same as for polar codes, since each Manchester pulse is multiplied by ± 1, so

$$S_y(f) = |P(f)|^2 \left(\frac{1}{T_b} R_{xx}[0] \right) = \frac{1}{T_b} |P(f)|^2$$

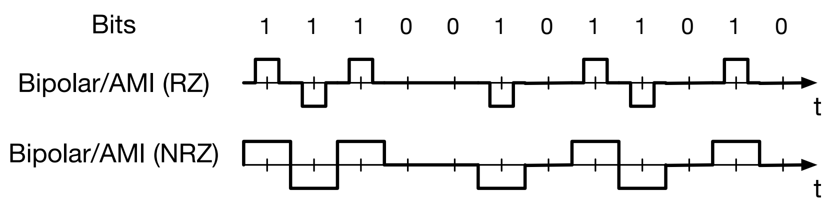
The line code power spectrum is then

$$\begin{aligned} S_y(f) &= |P(f)|^2 S_x(f) \\ &= \frac{1}{T_b} \left(T_b^2 \operatorname{sinc}^2 \left(\frac{T_b f}{2} \right) \sin^2 \left(\frac{\pi T_b f}{2} \right) \right) \\ &= T_b \operatorname{sinc}^2 \left(\frac{T_b f}{2} \right) \sin^2 \left(\frac{\pi T_b f}{2} \right) \end{aligned}$$

The sinc term is the same as for the RZ polar case. The sine squared term zeros out the DC value of the transmitted power spectrum.

Alternate Mark Inversion (Bipolar) Signaling

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



AMI was used in early PCM (digital phone) 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 $R_{xx}[0]$ is

$$R_{xx}[0] = \left(\frac{1}{2}\right) 1 + \left(\frac{1}{2}\right) 0 = \frac{1}{2}$$

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,

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

For $n = 2$, the various permutations of 1's and 0's are either zero, or cancel out and give $R_{xx}[2] = 0$. This continues for $n > 2$.

PSD of AMI Signaling, (cont.)

Therefore

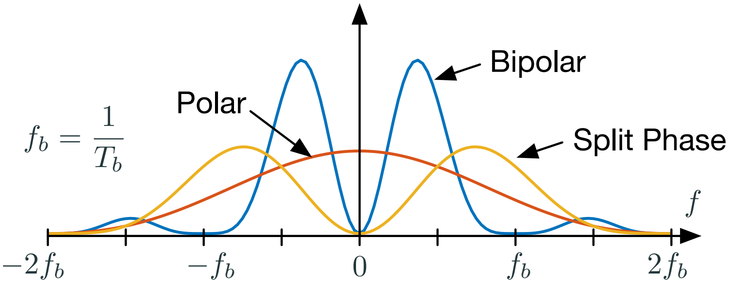
$$\begin{aligned} S_y(f) &= \frac{|P(f)|^2}{T_b} \sum_n R_{xx}[n] e^{j2\pi n f T_b} \\ &= \frac{|P(f)|^2}{T_b} \left(R_{xx}[0] + R_{xx}[-1] e^{j2\pi f T_b} + R_{xx}[1] e^{-j2\pi f T_b} \right) \\ &= \frac{|P(f)|^2}{T_b} \left(\frac{1}{2} - \left(\frac{1}{4} \right) 2\cos(2\pi f T_b) \right) \\ &= \frac{|P(f)|^2}{2T_b} \left(1 - \cos(2\pi f T_b)\right) \\ &= \frac{|P(f)|^2}{T_b} \sin^2(\pi T_b f) \end{aligned}$$

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

It also has a null at DC, which is desirable.

PSD Comparison

The PSD's a couple of line codes that allow easy clock recovery are plotted below:



These are RZ polar, split phase, and NRZ bipolar (AMI)

OOK has the same shape as polar, but with an added impulse at the origin.

Differential Line Encoding

So far we've assumed we send a "1" with one pulse, and a "0" with another. Pulses map directly onto bits.

As we will see, another very common option is transitmit the *change* in bits

- one pulse to indicate that this bit is the same as the previous one
- another pulse to indicate that the bit flipped.

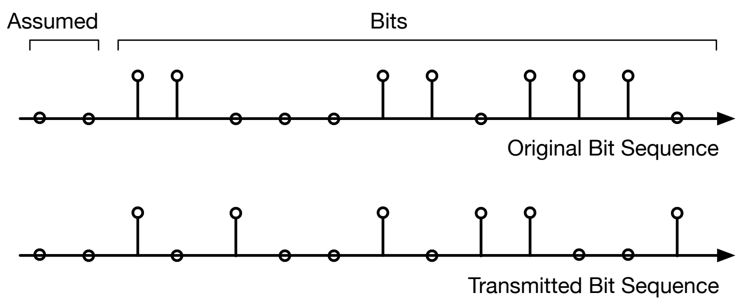
Then, when decoding the bits, an initial state is assumed.

This is called *differential line encoding*, because the difference in bits is transmitted.

It can be combined with most of the previous methods.

Differential Encoding Example

This looks like this



There are many ways this can be transmitted. Which symbol is a transition (mark), which symbol means stay the same (space)?

And then sometimes the logic is inverted. Reverse engineering a signal can be a challenge!

Line Coding Conclusion

Line coding defines the way pulses are used to transmit information.

Many different options, each with their own tradeoffs.

It can be confusing initially, but you will get used to the terminology

We will see many variations on these ideas in the rest of the class.

In the next classes we will look at

- How do I optimally detect a pulse in noise?
- How can a design $p(t)$ to minimize the bandwidth required?
- How do I characterize how difficult a signal is to decode?
- How can I send many bits per symbol?