

## Line Codes

Line codes are the waveforms used to transmit data over baseband channels. Many different line codes have been developed. After this chapter you should be able to: encode/decode data to/from the line codes described below and identify the characteristics of a line code including: number of levels, unipolar vs bi-polar, maximum time between transitions, use of differential encoding, block vs bit-by-bit encoding, bandwidth and DC balance.

### Lines Codes

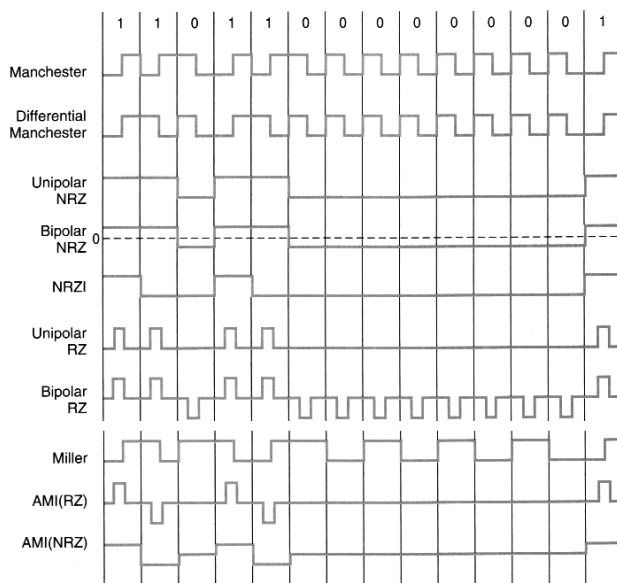
Line codes encode data bits into waveforms that are suitable for transmission over baseband channels.

Dozens of different line codes have been developed to match the characteristics of different channels, different data rates, different implementation technologies and different cost/performance requirements. The choice of line code will influence the signal's bandwidth, DC content, transition density and implementation complexity.

This chapter describes some typical line codes and their characteristics.

### Examples of Line Codes

This section describes some common or representative codes. The figure above shows some of these codes and is from the text by [Hioki](#):



### NRZ - Non-Return to Zero

The NRZ line code is often used over baseband channels.

There waveforms consists of one of two different voltage levels with durations equal to the bit period.

### Differential NRZ

NRZ can use differential encoding so that a 1 is encoded as a change in level and 0 is encoded by no change.

For example, the 12 Mb/s USB protocol uses differential NRZ. A change in voltage represents a '1', no change represents a '0'. To avoid long periods without transitions a '1' bit is bit-stuffed after 5 consecutive 0's.

### RZ

The waveforms in the RZ (return to zero) line code return to zero in the second half of each bit interval.

The first part of each waveform (1 or 0) can be encoded as unipolar (only one voltage is non-zero) or bipolar (both voltages are non-zero). The first part may be encoded differentially if desired.

RZ ensures a transition in every bit if bipolar encoding is used. The disadvantage is the increase in bandwidth.

A variant of RZ called AMI (alternate mark inversion) was used in the earliest digital telephony transmission systems called "T1".

### Manchester

A '1' is transmitted as a waveform that goes from low to high half-way into the bit period. A '0' is transmitted as a waveform that goes from high to low half-way into the bit period<sup>1</sup>.

<sup>1</sup>The opposite convention is sometimes used.

Manchester coding can also be transmitted differentially. A '1' is encoded by using a different waveform than was used for the previous bit and a '0' is encoded by the same waveform as the previous bit.

Manchester coding is used by 10Mb/s Ethernet (10BaseT).

### Delay (Miller) Coding

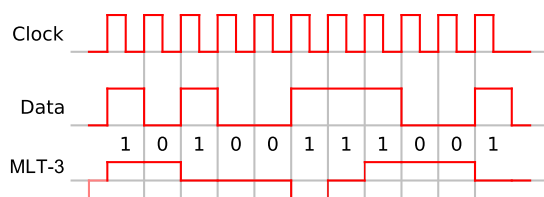
Delay coding encodes a '1' as a transition in the middle of the bit period. The encoding of the transition for a '0' depends on the previous bit. This results in less variation in time between transitions and a more concentrated power spectrum.

This line code was commonly used when writing data to disk drives.

### MLT-3

This line code uses three voltage levels. Each '1' bit switches to the next level. When the highest or lowest level is reached the direction reverses. This produces something that looks like a sine wave and has a compact spectrum.

This is the line code used for 100 Mb/s Ethernet (100BaseTX). The following example is from Wikipedia:



### Zero Substitution

Some line codes recognize specific sequences of data and substitute other signal patterns. These substitutions can be recognized because they include line coding violations.

An example is the B8ZS variant of AMI. This line code is similar to AMI: there are 3 voltage levels, and successive marks (1's) used alternate polarities (first positive, then negative, then positive, etc.) to obtain zero DC balance. The pulse width is 50% of the bit duration.

However, if the transmitter detects a sequence of 8 consecutive zeros it substitutes a special sequence.

These special sequences: 000-+0+- if previous pulse was a - or opposite if previous pulse was a +, include coding violations so they can be recognized by the receiver which then replaces them with zeros.

**Exercise 1:** Convert the sequence 0100 0000 0000 0100 to a B8ZS waveform assuming the previous mark was transmitted as a positive pulse.

Note that the sequence remains balanced because it includes two coding violations (one initial and one embedded) of opposite polarities.

The HDB3 line code is used by E1 (European variant of T1). It is similar but substitutes for sequences of four consecutive zeros and thus has a higher guaranteed transition density.

### 4B5B

This line code code converts a sequence of 4 bits into a sequence of 5 bits. It is an example of a block code.

Since there are 16 input sequences and 32 output sequences, certain problematic output sequences can be omitted (not used). Those sequences include those that do not have transitions and those that have more than a certain number of consecutive 1's or 0's.

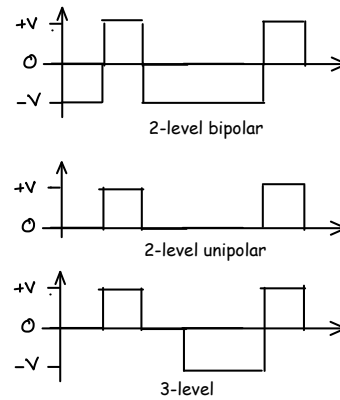
The advantage of 4B5B is that only a relatively small increase in bandwidth is required while still ensuring low DC content and a minimum transition density.

This code is used for 100 Mb/s Ethernet (100BaseTX) (before MLT-3 encoding).

The table below, taken from the IEEE 802.3 (Ethernet) standard, shows the 4B5B encoding:

Table 24-1—4B/5B code-groups

	PCS code-group [4:0] 4 3 2 1 0	Name	MII (TXD/RXD) <3:0> 3 2 1 0	Interpretation
DATA	1 1 1 1 0	0	0 0 0 0	Data 0
	0 1 0 0 1	1	0 0 0 1	Data 1
	1 0 1 0 0	2	0 0 1 0	Data 2
	1 0 1 0 1	3	0 0 1 1	Data 3
	0 1 0 1 0	4	0 1 0 0	Data 4
	0 1 0 1 1	5	0 1 0 1	Data 5
	0 1 1 1 0	6	0 1 1 0	Data 6
	0 1 1 1 1	7	0 1 1 1	Data 7
	1 0 0 1 0	8	1 0 0 0	Data 8
	1 0 0 1 1	9	1 0 0 1	Data 9
	1 0 1 1 0	A	1 0 1 0	Data A
	1 0 1 1 1	B	1 0 1 1	Data B
	1 1 0 1 0	C	1 1 0 0	Data C
	1 1 0 1 1	D	1 1 0 1	Data D
	1 1 1 0 0	E	1 1 1 0	Data E
	1 1 1 0 1	F	1 1 1 1	Data F
	1 1 1 1 1	I	undefined	IDLE; used as inter-stream fill code
	0 0 0 0 0	P	0 0 0 1	SLEEP; LPI code only for the EEE capability. Otherwise, invalid code; refer to Table 22-1 and Table 22-2
CONTROL	1 1 0 0 0	J	0 1 0 1	Start-of-Stream Delimiter, Part 1 of 2; always used in pairs with K
	1 0 0 0 1	K	0 1 0 1	Start-of-Stream Delimiter, Part 2 of 2; always used in pairs with J
	0 1 1 0 1	T	undefined	End-of-Stream Delimiter, Part 1 of 2; always used in pairs with R
	0 0 1 1 1	R	undefined	End-of-Stream Delimiter, Part 2 of 2; always used in pairs with T
INVALID	0 0 1 0 0	H	Undefined	Transmit Error; used to force signaling errors
	0 0 0 0 0	V	Undefined	Invalid code
	0 0 0 0 1	V	Undefined	Invalid code
	0 0 0 1 0	V	Undefined	Invalid code
	0 0 0 1 1	V	Undefined	Invalid code
	0 0 1 0 1	V	Undefined	Invalid code
	0 0 1 1 0	V	Undefined	Invalid code
	0 1 0 0 0	V	Undefined	Invalid code
	0 1 1 0 0	V	Undefined	Invalid code
	1 0 0 0 0	V	Undefined	Invalid code
	1 1 0 0 1	V	Undefined	Invalid code



Four levels (often called PAM4) can encode two bits in each symbol time. Five levels are used by 1000Base-T Ethernet (1 Gb/s over “twisted pair”).

### Differential Coding

It is possible to encode each bit as a waveform that is either different or the same as the previous waveform. For example, with two voltage levels a data value of ‘1’ could be transmitted as a change from the previous level and ‘0’ as no change.

This “differential” encoding makes the data insensitive to the channel polarity. This is convenient since it allows cable pairs to be connected without regards to polarity. Differential line codes may or may not use differential voltages and balanced transmission lines – these two meanings of “differential” are different.

One disadvantage of differential encoding is that errors always occur in pairs which doubles the bit error rate.

**Exercise 4:** Assume a 1 is transmitted as 5V and 0 as 0V. Draw the waveform for the bit sequence 1011. Draw the waveform if the bits are transmitted differentially with a 1 encoded as a change in level. Assume the initial value of the waveform is 0. Invert the waveform and decode it.

### Gray Coding

Line codes with multiple levels typically uses Gray coding. This ensures that there is only one bit in error if the receiver mistakes one level for an adjacent one.

**Exercise 5:** Show the binary and Gray-coded encodings for PAM4. What is the average number of bits in error if the only errors are between adjacent levels?

**Exercise 2:** How would the bit sequence 0110 be encoded using 4B5B followed by MLT3 assuming the starting level is 0V?

### Characteristics of Line Codes

#### Voltage Levels

Most line codes use only two voltage levels. These are the simplest to implement and have the largest noise margin for a given signal power.

**Exercise 3:** What noise level is required to cause an error when using a bipolar line code with levels of  $\pm 1$  V? What are the voltage levels for a unipolar line code with the same noise margin? What are the RMS voltages of these two line codes when transmitting a dotting sequence (alternating 1’s and 0’s)? Why might you use unipolar line codes anyways?

The two voltages may be balanced about zero (sometimes called a bipolar code) or one of the levels may be zero (sometimes called unipolar).

Multi-level codes are also common.

For example, a bipolar line code that uses three levels. It alternates a zero value and two polarities (negative and positive voltages of equal magnitude). This makes the average (DC) voltage zero.

## Maximum Time Between Transitions

Digital systems use a periodic signal called a clock to synchronize the transfer of data between devices. Few communications systems transmit a clock signal along with their data because it would require an additional channel (an extra twisted pair, for example). The receiver thus has to recover the clock from the received data waveform.

Both the transmitter and receiver know the approximate clock frequency so the clock recovery problem is limited to correcting for an accumulated time difference between the clocks due to a relatively small frequency difference.

Receivers look at the transitions between signal levels to help estimate the phase (or timing) error between their clocks. They typically average these estimates to correct the point at which the receiver samples the received signal. More signal transitions provide more information for the receiver to resynchronize its clock.

## Transition Density

Some line codes have a maximum of one transition per bit, while others have two.

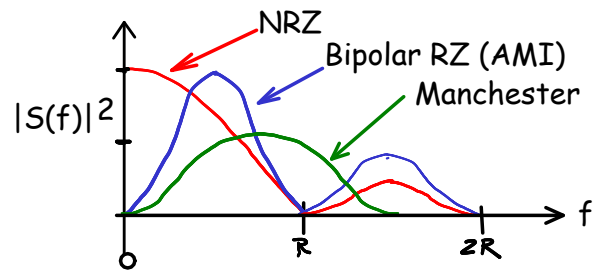
Some codes guarantee a minimum number of transitions in a given time interval. This can help the receiver synchronize itself to the received waveform so that it can accurately recover the data.

More transitions per symbol typically means a higher bandwidth for a given symbol rate although the actual spectrum can be difficult to calculate and depends on the statistics of the transitions which in turn depends on the line code.

## Power Spectrum

A power spectral density (PSD,  $|S(f)|^2$ ) is a measurement, in W/Hz of the amount of power contained at different frequencies. The following figure shows the approximate PSD of three different line codes. They are drawn relative to the symbol rate,  $R$ .

The power of an NRZ-encoded signal is maximum at DC and the first null is at the bit (or symbol) rate,  $R$ . The power of a bipolar RZ signal using AMI is zero at DC and peaks at half the symbol rate. A Manchester-coded signal also has no DC content and little power at low frequencies but has a wider first-null bandwidth than NRZ.



## High-Frequency Content

Channels attenuate signals above a certain frequency (the “bandwidth” of the channel). We can maximize the symbol rate that can be transmitted over such a channel by minimizing the high-frequency content of the signal relative to the symbol rate.

Even when bandwidth is not a concern, limiting the bandwidth reduces the power at higher frequencies. Power at these frequencies radiates more than low-frequency signals and are more likely to cause interference and crosstalk (interference to signals on adjacent cables).

## Low-Frequency Content

In many cases transformers or capacitors are used by baseband channels for impedance matching or protection purposes. These channels do not pass DC and will attenuate lower frequencies. Line codes can be designed so that they have no DC content (average voltage is zero) and little power at low frequencies. These line codes pass through AC-coupled channels with less distortion.

## DC Balance

Line codes can be designed so that the average voltage of the waveform (its DC value) is zero. This is different than having reduced low-frequency content. For example an NRZ signal with voltages balanced about zero will have significant low frequency content but zero average voltage.

## Block Line Codes

Some line codes operate on blocks of bits. They read a number of data bits from the source and output a larger number of bits (or symbols). For example, a transmitter can read a block of 3 bits and output 4 bits. This called a 3B4B block code (B stands for binary).

**Exercise 6:** How many combinations are there of 3 bits? Of 4 bits? How many bits might be input and output by an 8B10B code? What might a 4B3T code mean?

The advantage of a block code is that the possible output patterns do not have to include those with undesirable properties. For example, the all-high or all-low waveforms might be omitted because the average DC content is too high or too low. Other output waveforms might be excluded because there is too long an interval between transitions.

**Exercise 7:** Design your own 2B3B line code by choosing the output waveforms that have the lowest average DC value and giving preference to those that start and end at different levels (assume bipolar signalling).

Block codes can be more efficient because the ratio of output to input bits can be low (10/8, for example). But this comes at the cost of additional complexity for encoding, decoding and detecting the boundaries of the blocks. Synchronization is usually done by using output bit patterns that do not correspond to any possible input data sequence.

## Violations

Some line codes allow for the encoding algorithm to be violated to indicate infrequent events such as the start of a frame. A typical example is a bipolar code that transmits two adjacent bipolar pulses using the same (instead of alternating) polarity. Another example is substituting a specific sequence that include a violation (e.g. the B8ZS variant of AMI substitutes  $000+-0-+$  or  $000-+0+-$  for eight consecutive zeros).

**Exercise 8:** What (minimum) delay does a B8ZS encoder add? Receiver? Why is there a zero between successive same-polarity pulses?

In other cases violation of the line coding rules indicates an error. This can be useful for detecting problems at the physical layer (we will study other techniques that are used at higher layers).