Line Codes

Line codes are the waveforms used to transmit data over baseband channels.
After this chapter you should be able to: encode and decode data using the line codes described below and identify the advantages and disadvantages of each.

Line Codes

Line codes encode data into waveforms that are suitable for transmission over baseband[1] channels. Some common line codes are described below.

Exercise 1: Encode the bit sequence 1101 1001 using NRZ, RZ, AMI and Manchester line codes described below.

NRZ

NRZ (non return to zero) transmits a 1 as a positive voltage for one bit (symbol) duration and a 0 as a zero voltage. The opposite (active-low) convention could also be used. This line code has two disadvantages:

- A long sequence of bits of the same value will have no transitions between voltages and the receiver may lose track of the position of boundaries between bits. In other words, it may lose synchronisation.
- The average (DC) voltage may not be zero. This means NRZ signals cannot be reliably transmitted over AC-coupled transmission lines such as those using transformers.

RZ

Synchronisation can be aided by transmitting a positive pulse for a 1 and a negative pulse for a 0 and having the waveform return to zero in-between pulses.

This is a "self-clocking" line code – each symbol includes an edge.

However, the average (DC) voltage will not be zero if we transmit a different number of 1's and 0's.

[1]Baseband channels are those that are centered around DC (zero frequency). Passband channels are centered around a carrier frequency.

AMI

AMI (alternate mark inversion) ensures a zero DC voltage by transmitting RZ pulses only for 1's but alternating between positive and negative pulses.

However, 1 bits must be inserted in long runs of 0 bits to ensure transitions. This can be done either periodically or using zero-substitution as described below.

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

Manchester

The Manchester line code achieves both zero average voltage and is self-clocking. It encodes each bit as two different voltage levels. A 1 is transmitted as a low level followed by a high level (a rising edge in the middle of the bit interval) an a 0 is transmitting as a high level followed by a low level (a falling edge)[2].

Unlike AMI and RZ, Manchester is unipolar which simplifies its implementation.

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

Block Codes

Block codes gather bits into a word and use that word to select an output waveform.

An example is 4B5B which inputs 4 bits at a time and uses those four bits word to select one of 16 5-bit sequences. Because only 16 of the 32 possible output sequences are used, the transmitted signal can be guaranteed to have transitions.

This code is used for 100 Mb/s Ethernet over twisted-pair transmission lines (100BaseTX).

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

[2]The opposite convention is sometimes used.

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

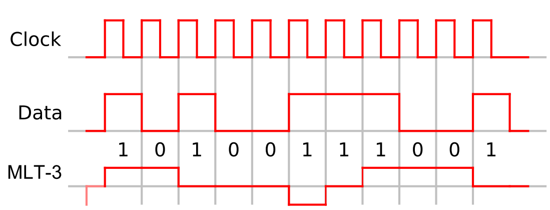
Other examples include 4B3T that selects one of 16 3-level waveforms, and 8B10B (used in USB3 and HDMI) in which 8 bits are used to select one of 256 10-bit outputs.

MLT-3

MLT-3 uses three levels and encodes a 1 bit as a transition to the adjacent level and a 0 bit as no change.

MLT-3 reduces the frequency and amplitude of transitions which reduces the high-frequency content of the signal. This is desirable because transmission lines have more attenuation and radiate more at higher frequencies.

This is the line code used, together with 4B5B, for 100 Mb/s Ethernet (100BaseTX). The following example is from Wikipedia:



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

Differential Coding

Differential coding makes data insensitive to polarity inversion by encoding data as the difference between successive symbols. For example, a 1 can be encoded as transmitting a different symbol than the previous one and a 0 as the same symbol.

Differential coding is often used with NRZ (then often called NRZI) and Manchester.

A disadvantage of differential coding is that each error introduces two errors which doubles the bit error rate.

Exercise 3: Why?

Exercise 4: Encode the bit sequence 1011 using NRZ, NRZI and Manchester. Invert the waveforms. Decode them. Assume the initial value of the waveform is 0.

Note that differential *coding*is not the same as differential *signalling* . Differential *signalling* transmits signals as the voltage difference between two wires and will be described later.

Bit Stuffing

Bit stuffing is used to introduce transitions into line codes such as NRZ and NRZI.

To avoid long periods without transitions a '1' bit is bit-stuffed after a certain number of consecutive 0's.

To implement bit-stuffing the transmitter and receiver both track the number of consecutive 0 bits and insert or delete a 1 bit after a certain number of 0 bits.

For example, the 12 Mb/s USB protocol uses NRZI with bit-stuffing after 5 zero (no change) bits.

A disadvantage of bit-stuffing is that bit stuffing increases the bit rate by a variable amount – it depends on the data.

Exercise 5: What is worst-case increase in bit rate?

Exercise 6: Encode the bit sequence 1101 0000 0001 using NRZI with bit-stuffing after 5 zero bits.

Zero Substitution

It's possible for the transmitter to replace long sequences of 0 bits with a pattern that includes line coding violations. These patterns can be recognized by the receiver and the 0 bits reinserted.

An example is the B8ZS (bipolar with 8 zero substitution), a variant of AMI. As with AMI there are 3 voltage levels, and successive marks (1's) use alternate polarities (alternating positive and negative) to ensure zero average voltage.

When the transmitter detects a sequence of 8 consecutive zeros it substitutes the sequence 000-+0+- (where + and - indicate the pulse polarity) if the previous pulse was negative or the opposite sequence if the previous pulse was positive. These patterns include coding violations – two successive pulses of the same polarity – so they can be recognized by the receiver which then replaces the whole 8-bit sequence with zeros.

Exercise 7: Convert the sequence 0100 0000 0000 0100 to a B8ZS waveform assuming the first 1 is 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 used by E1 (the European variant of T1) is similar but substitutes for sequences of four consecutive zeros and thus has a higher guaranteed transition density.

Multi-Level Line Codes

Four levels (often called PAM4) can encode two bits in each symbol time.

This halves the bandwidth but increases the likelihood of errors due to noise because of the smaller voltage difference between levels.

Line codes with multiple levels typically use Gray coding. This ensures that there is only one bit error in the most common situation – where the receiver mistakes one level for an adjacent one.

Exercise 8: Show the binary and Gray-coded encodings for PAM4. What is the average number of bits in error in each case if the only errors are between adjacent levels?

Other number of levels can be used. For example, PAM5, PAM8, and PAM16 are used by various Ethernet standards.

Line Code Violations

Some systems use violations of the line coding rules to indicate infrequent events. A typical example is a system using AMI that transmits two adjacent pulses of the same polarity to indicate the start of a frame.

In other cases violation of the line coding rules indicates an error. Counts of such violations can be a rough measure of signal quality.