

Non-return to zero encoding



**ASSIGN
BUSTER**

Non-Return to Zero (NRZ) Encoding

Non-return to zero is one of the encoding formats used in digital signals. It is commonly used in slow speed communications interfaces for both synchronous and asynchronous transmission. Using NRZ, a logic 1 bit is sent as a high value and a logic 0 bit is sent as a low value (the line driver chip used to connect the cable may subsequently invert these signals).

Some problem arises while using NRZ to encode a synchronous link which may have long runs of consecutive bits with the same value. Below is the problem that would arise if NRZ encoding were used with a DPLL recovered clock signal. In Ethernet for example, there is no control over the number of 1's or 0's which may be sent consecutively. There could potentially be thousands of 1's or 0's in sequence. If the encoded data contains long 'runs' of logic 1's or 0's, this does not result in any bit transitions. The lack of transitions prevents the receiver DPLL from reliably regenerating the clock making it impossible to detect the boundaries of the received bits at the receiver. This is the reason why Manchester coding is used in Ethernet LANs.

A long run of bits with the same value results in no transitions on the cable when NRZ encoding is used

Non-return-to-zero

The binary signal is encoded using rectangular pulse amplitude modulation with polar non-return-to-zero code

Non-return-to-zero (NRZ) line code is a binary code in which 1's are represented by a positive voltage and 0's are represented by a negative voltage, with no other neutral or rest condition. The pulses have more

energy than a RZ code. Unlike RZ, NRZ does not have a rest state. NRZ is not inherently a self-synchronizing code, so some additional synchronization technique . For a given data signaling rate, i. e., bit rate, the NRZ code requires only half the bandwidth required by the Manchester code.

When used to represent data in an asynchronous communication scheme, the absence of a neutral state requires other mechanisms for data recovery, to replace methods used for error detection when using synchronization information when a separate clock signal is available.

NRZ-Level is not a synchronous system but rather an encoding that can be used in either a synchronous or asynchronous transmission environment, that is, with or without an explicit clock signal involved. Because of this, it is not strictly necessary to discuss how the NRZ-Level encoding acts “ on a clock edge” or “ during a clock cycle” since all transitions happen in the given amount of time representing the actual or implied integral clock cycle. The real question is that of sampling—the high or low state will be received correctly provided the transmission line has stabilized for that bit when the physical line level is sampled at the receiving end.

It is easier to see NRZ transitions as happening on the falling clock edge in order to compare NRZ-Level to other encoding methods, such as the mentioned Manchester code, which requires clock edge information (is the XOR of the clock and NRZ, actually) and to see the difference between NRZ-Mark and NRZ-Inverted

Unipolar Non-Return-to-Zero Level

“ One” is represented by one physical level by a negative voltage.

“ Zero” is represented by a positive voltage.

In clock language, “ one” transitions or remains high on the trailing clock edge of the previous bit and “ zero” transitions or remains low on the trailing clock edge of the previous bit, or just the opposite. This allows for long series without change, which makes synchronization difficult. One solution is to not send bytes without transitions, see the diagram shows a line representing the physical zero below the biased logical zero—showing the less usual case of “ one” being a high voltage.

Bipolar Non-Return-to-Zero Level

“ One” is represented by a negative voltage.

“ Zero” is represented by a positive voltage.

In bipolar NRZ-Level the voltage “ swings” from positive to negative on the trailing edge of the previous bit clock cycle.

Non-Return-to-Zero Space

Non-Return-to-Zero Space no change in physical level represents one.

change in physical level represents zero.

In clock language, the level transitions on the trailing clock edge of the previous bit to represent a “ zero.”

This “ change-on-zero” is used by High-Level Data Link Control and USB.

They both avoid long periods of no transitions (even when the data contains long sequences of 1 bits) by using zero-bit insertion. HDLC transmitters insert a 0 bit after five contiguous 1 bits (except when transmitting the frame

delimiter '01111110'). USB transmitters insert a 0 bit after six consecutive 1 bits. The receiver at the far end uses every transition — both from 0 bits in the data and these extra non-data 0 bits — to maintain clock synchronization. The receiver otherwise ignores these non-data 0 bits.

Non return to zero, inverted (NRZI)

It is a method of mapping a binary signal to a physical signal for transmission over some transmission media. The two level NRZI signal has a transition at a clock boundary if the bit being transmitted is a logical one, and does not have a transition if the bit being transmitted is a logical zero.

NRZI might take the opposite convention, as in Universal Serial Bus (USB) signalling, when in Mode 1 (transition when signalling zero and steady level when signalling one). The transition occurs on the leading edge of the clock for the given bit. This distinguishes NRZI from NRZ-Mark.

However, even NRZI can have long series of zeros (or ones if transitioning on “zero”), so clock recovery can be difficult unless some form of run length limited coding is used on top. Magnetic disk and tape generally uses fixed-rate RLL codes, while USB uses bit stuffing, which is efficient, but results in a variable data rate: it takes slightly longer to send a long string of 1 bits over USB than it does to send a long string of 0 bits. (USB inserts an additional 0 bit after 6 consecutive 1 bits.)

Differential Manchester encoding

It is a method of encoding data in which data and clock signals are combined to form a single self-synchronizing data stream. It is a differential encoding,

using the presence or absence of transitions to indicate logical value. This gives it several advantages over standard Manchester encoding:

Detecting transitions is often less error-prone than comparing against a threshold in a noisy environment.

Because only the presence of a transition is important, polarity is not. Differential coding schemes will work exactly the same if the signal is inverted (wires swapped). (Other line codes with this property include NRZI, bipolar encoding, biphas mark code, coded mark inversion, and MLT-3 encoding).

A '1' bit is indicated by making the first half of the signal equal to the last half of the previous bit's signal i. e. no transition at the start of the bit-time. A '0' bit is indicated by making the first half of the signal opposite to the last half of the previous bit's signal i. e. a zero bit is indicated by a transition at the beginning of the bit-time. In the middle of the bit-time there is always a transition, whether from high to low, or low to high. A reversed scheme is possible, and no advantage is given by using either scheme.

Example of Differential Manchester encoding

A related method is Manchester encoding in which the meaningful transitions are the mid-bit ones, and these encode data by their direction (positive-negative is one value, negative-positive is the other).

Differential Manchester is specified in the IEEE 802.5 standard for token ring LANs, and is used for many other applications, including magnetic and optical storage.

In differential Manchester encoding,

if a “ 1” is represented by one transition, then a “ 0” is represented by two transitions and vice versa

Manchester code

Manchester code is a line code in which the encoding of each data bit has at least one transition and occupies the same time. It is, therefore, self-clocking, which means that a clock signal can be recovered from the encoded data.

Manchester code is widely-used e. g. in Ethernet. There are more complex codes e. g. 8B/10B encoding which use less bandwidth to achieve the same data rate (but which may be less tolerant of frequency errors and jitter in the transmitter and receiver reference clocks).

Features

Manchester code ensures frequent line voltage transitions, directly proportional to the clock rate. This helps clock recovery.

The DC component of the encoded signal is not dependent on the data and therefore carries no information, allowing the signal to be conveyed conveniently by media (e. g. Ethernet) which usually do not convey a DC component.

Description

Example of Manchester encoding showing both conventions

Extracting the original data from the received encoded bit (from Manchester as per 802. 3):

original data = clock XOR Manchester value

0 0 0

0 1 1

1 0 1

1 1 0

Summary:

- * Each bit is transmitted in a fixed time (the “ period”).
- * A 0 is expressed by a low-to-high transition, a 1 by high-to-low transition (according to G. E. Thomas’ convention — in the IEEE 802. 3 convention, the reverse is true).
- * The transitions which signify 0 or 1 occur at the midpoint of a period.
- * Transitions at the start of a period are overhead and don’t signify data.

Manchester code always has a transition at the middle of each bit period and may (depending on the information to be transmitted) have a transition at the start of the period also. The direction of the mid-bit transition indicates the data. Transitions at the period boundaries do not carry information. They exist only to place the signal in the correct state to allow the mid-bit transition. Although this allows the signal to be self-clocking, it doubles the bandwidth requirement compared to NRZ coding schemes (or see also NRZI).

In the Thomas convention, the result is that the first half of a bit period matches the information bit and the second half is its complement.

Manchester encoding as phase-shift keying

Manchester encoding is a special case of binary phase-shift keying (BPSK), where the data controls the phase of a square wave carrier whose frequency is the data rate. Such a signal is easy to generate.

To control the bandwidth used, a filter can reduce the bandwidth to as low as 1Hz per bit/second without loss of information in transmission.[citation needed] In radio transmission, the encoded signal may also be modulated with a carrier wave; however, the property of 1Hz per bit/second is preserved.

Conventions for representation of data

Encoding of 11011000100 in Manchester code (as per G. E. Thomas)

There are two opposing conventions for the representations of data.

The first convention specifies that for a 0 bit the signal levels will be a low level in the first half of the bit period, and a high level in the second half. For a 1 bit the signal levels will be High-Low.

The second convention is also followed by numerous authors (e. g., Stallings) as well as by IEEE 802. 4 (token bus) and lower speed versions of IEEE 802. 3 (Ethernet) standards. It states that a logic 0 is represented by a High-Low signal sequence and a logic 1 is represented by a Low-High signal sequence.

Shortcomings

Manchester code needs twice the bandwidth of asynchronous communications, and the signal spectrum is much wider. Most high-speed communication now uses encoding schemes with better coding performance.

One consideration is synchronization of the receiver to the transmitter. It might appear that a half bit period error would give an inverted output at the receiver, but for typical data this leads to code violations. The receiver can detect these violations and use this information to synchronise accurately.

Return-to-zero

it uses a line code in which the signal drops (returns) to zero between each pulse. This takes place even if a number of consecutive 0's or 1's occur in the signal. The signal is self-clocking. This means that a separate clock does not need to be sent alongside the signal, but suffers from using twice the bandwidth to achieve the same data-rate as compared to non-return-to-zero format.

The “ zero” between each bit is a neutral or rest condition, such as a zero amplitude in pulse amplitude modulation (PAM), zero phase shift in phase-shift keying (PSK), or mid-frequency in frequency-shift keying (FSK). That “ zero” condition is typically halfway between the significant condition representing a 1 bit and the other significant condition representing a 0 bit.

Although return-to-zero (RZ) contains a provision for synchronization, it still has a DC component resulting in “ baseline wander” during long strings of 0 or 1 bits, just like the line code non-return-to-zero.

Return-to-zero in optical communication

Return to pulse 3/16 of a bit long) by the IrDA serial infrared (SIR) physical layer specification. zero, inverted

Return-to-zero, inverted (RZI) is a method of mapping for transmission. The two-level RZI signal has a pulse (shorter than a clock cycle) if the binary signal is 0, and no pulse if the binary signal is 1. It is used.