Title: Waveform and Time Difference Capturing with C2NF's TTP Manchester Encoding

Subject: C2NF design

Description:

The C2NF, which is an improved version of the C2, not only includes MFM support but also a Manchster codec for its TTP transmission. While MFM is well-suited for applications which require low bandwidth utilization, it is not DC-balanced and allows undetected bit flips with a only half-bit-time-long fault on the bus media. Manchester encoding on the other hand features a DC-free signal on the line while requiring twice the signal bandwidth of MFM while enhancing the ability do detect media errors.

When using MFM coding, the C2 (and C2NF) transmitter enables the line driver at the actiontime (AT) while transmitting the inactive signal level '1'. This is done to guarantee a correct reception of the upcoming start-of-frame pattern (SOF) which is both necessary for start-of-frame detection (one part of the frame validation) and time synchronization. The receiver of an MFM encoded TTP stream expects the first falling edge on the media to be the time synchronization event. Any transmitter must delay its SOF transmission in a way that allows any receiver (which may be within +/- PI synchronized to the transmitter) a correct reception of the frame. The receiver opens a so-called start window when this first falling edge is earliest expected and closes it after either receiving the awaited edge or the time when the first falling edge is latest expected. So the SOF is transmitted starting at the delayed actiontime (delayed AT).

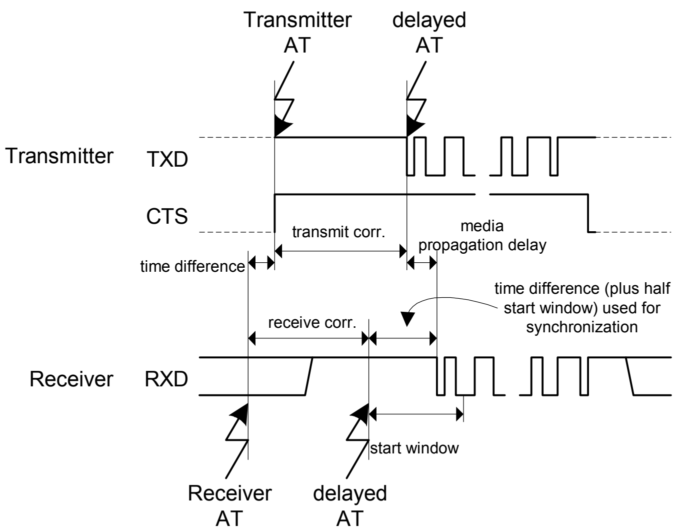

Figure 1 timestamping with MFM

When using Manchester coding, the C2NF transmitter enables the line driver at the delayed actiontime (delayed AT) while transmitting a preamble of configurable length. This preamble contains of repeated transmission of a "10" pattern. The purpose of the preamble is to retain DC balance on a per-bit basis, to allow any line driver to cut off some of the preamble due to its activation delay without destroying the start-of-frame (SOF), and to remove the saturation of a eventually-used line transformer coupler. With Manchester the SOF is equal to the pattern "1011". This means that the first three bits of the SOF can not be destinguished from three preamble bits, so the time difference capturing can not be based on the beginning of the SOF like in MFM coding but only on the detection of the last bit of the SOF. Note that using preamble bits usually doesn't have an impact on the line utilization bandwidth, only for long media propagation delays and for long preambles (needed only for line drivers with long activation delay) the receivers would have to move their startwindow to a later point of time. Note that frame reception works correct if the preamble reception starts either before or within the start window - only the SOF bit used for clock synchronization must be received within the start window.

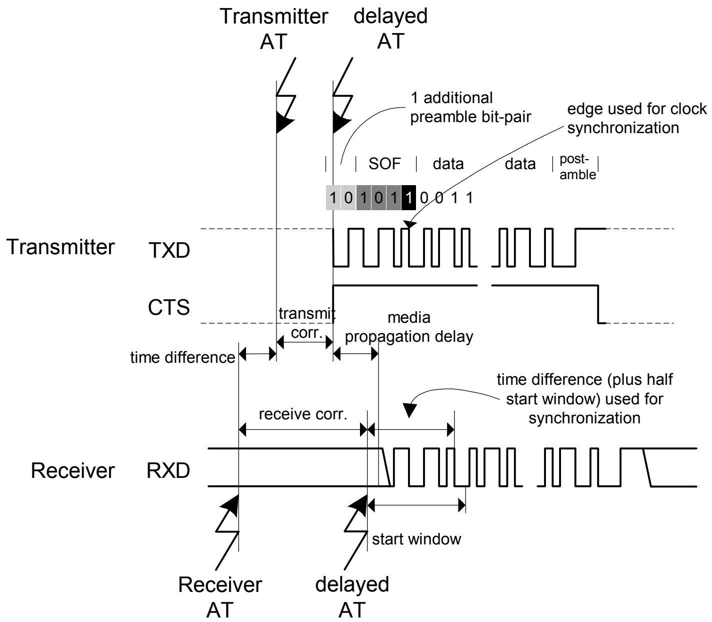

Figure 2 Timestamping with Manchester

The following formulas apply to the calculation of the transmit and receive correction values:

tx_corr	the transmitter correction value
rx_corr	the receiver correction value
tx_act_delay	the transmitter activation delay (implementation dependent)
rx_act_delay	the receiver activation delay (implementation dependent)
media_prop_delay	the propagation delay via tx line driver, line and rx line driver
pre_dibits_num	the number of preamble dibits added before the "1011" SOF, selectable via the MEDL transmitter configuration between 0 and 63
pre_halfbits_num	the number of preamble halfbits added before the "1011" SOF
halfbit_time	the length needed to transmit one halfbit, depends on the oversampling rate
startwin_half_time	half the width of the start window

MFM:

tx_corr + tx_act_delay + media_prop_delay = rx_corr + rx_act_delay + startwin_half_time

Manchester:

num_halfbits_per_bit = 2
num_bits_per_dibit = 2
pre_halfbits_num = num_bits_per_dibit * num_halfbits_per_bit * pre_dibits_num
pre_time = pre_halfbits_num * halfbit_time
sof_time = 3 * num_halfbits_per_bit * halfbit_time

tx_corr + tx_act_delay + media_prop_delay + pre_time + sof_time = rx_corr + rx_act_delay + startwin_half_time

Note: For MEDL generation, two passes of calculation are needed to find the optimum values for tx_corr and rx_corr:

1. Set rx_corr of all receivers of one frame to 0 and calculate the minimum (possibly negative) tx_corr = tx_corr_min of the transmitter of this frame.
2. Set tx_corr for this frame's transmitter to max(tx_corr_min,0) and re-calculate all receiver's rx_corr.

For further information about calculating the MEDL entries related to delay correction calculation, refer to the TTP/C specification.