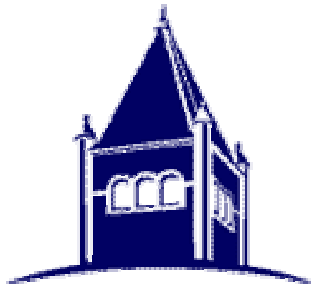


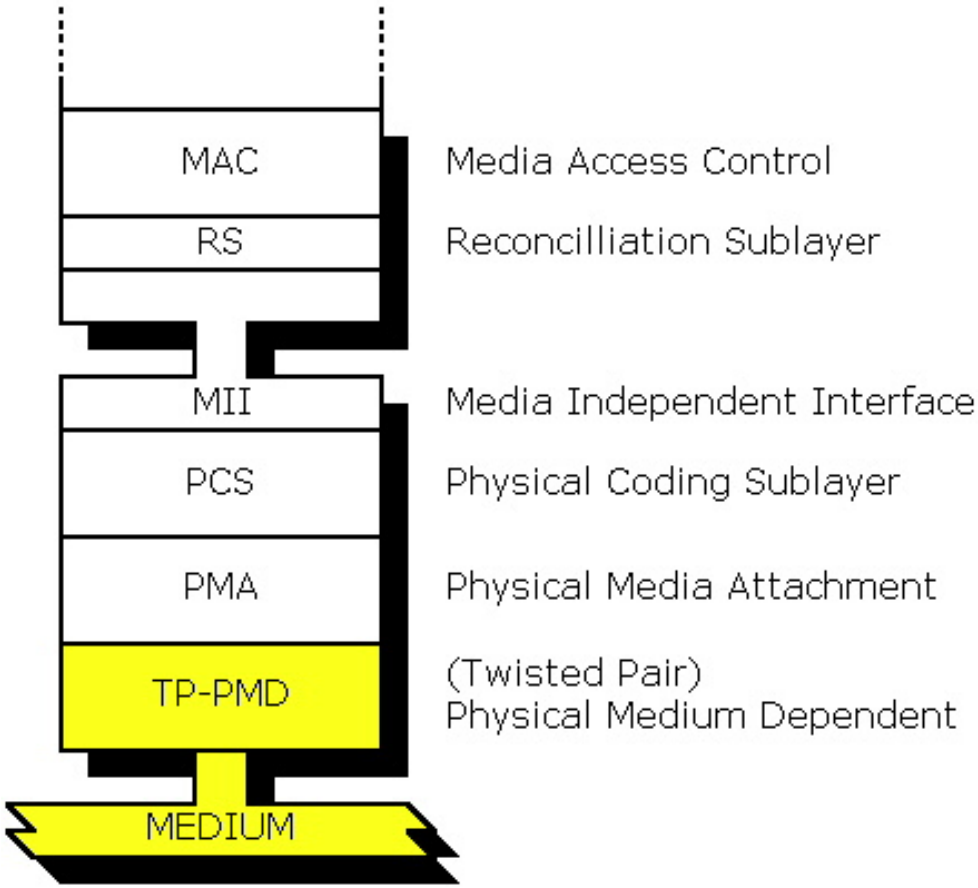
100BASE-TX

Physical Medium Dependent

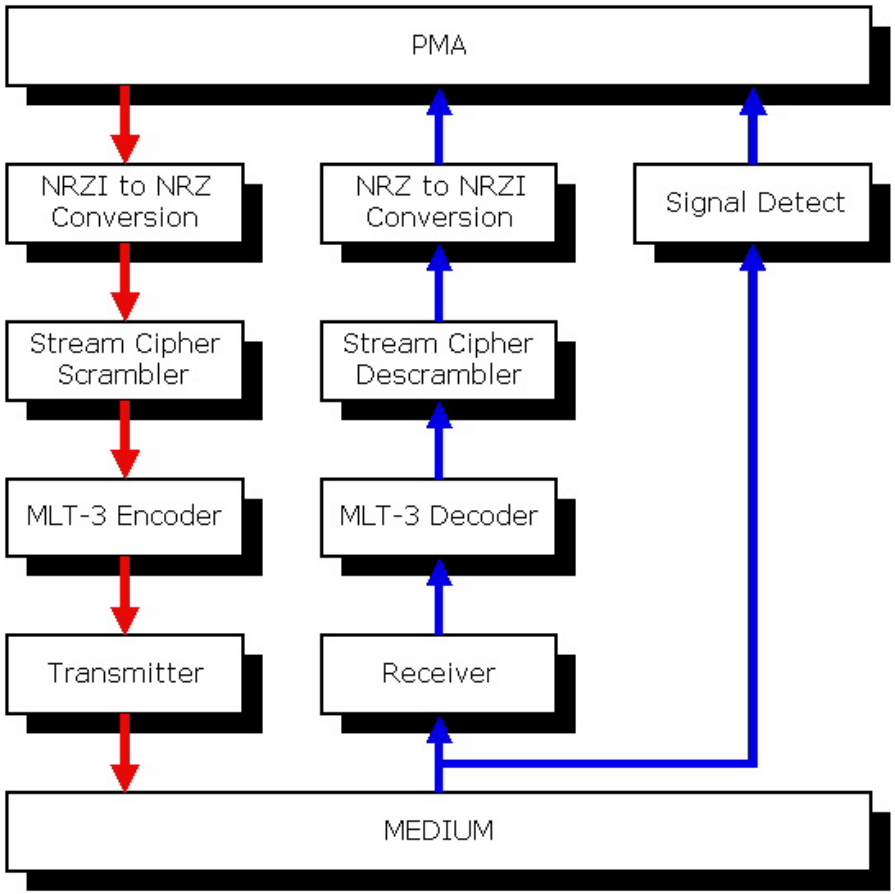
Presentation Overview:

- Location in the OSI Stack
- Interface with the Physical Medium Attachment (PMA)
- Interface with the medium (twisted pair link segment)
- PMD sublayer functionality

Stack Model


MAC
Media Access Control
RS
Reconcilliation Sublayer
MII
Media Independent Interface
PCS
Physical Coding Sublayer
PMA
Physical Media Attachment
TP-PMD
(Twisted Pair)
Physical Medium Dependent
MEDIUM

TP-PMD Block Diagram



Interface with the Physical Medium Attachment (PMA)

- The PMD converts NRZ bits to NRZI bits for presentation to the PMA
- The process is reversed for signaling coming from the PMA
- The PMD signal detect lets the PMA know if the signal being received is satisfactory
- Link_control from auto-negotiation can disable the PHY in order to negotiate a link and then enable the PHY once complete

Interface with the Medium

- The PMD converts MLT-3 symbols into electrical signals which are then sent out over the medium
- Electrical signals are received by the PMD from the medium and converted to MLT-3 symbols

Understanding the 100BASE-TX PMD

- Review
- Transmitter Functions
- Receiver Functions
- Signal Detect

Review

- 4B5B Block Code
- Non Return to Zero, (NRZ code)
- Non Return to Zero, Invert on one (NRZI code)
- Multi Level Transmit, 3 levels (MLT-3 code)

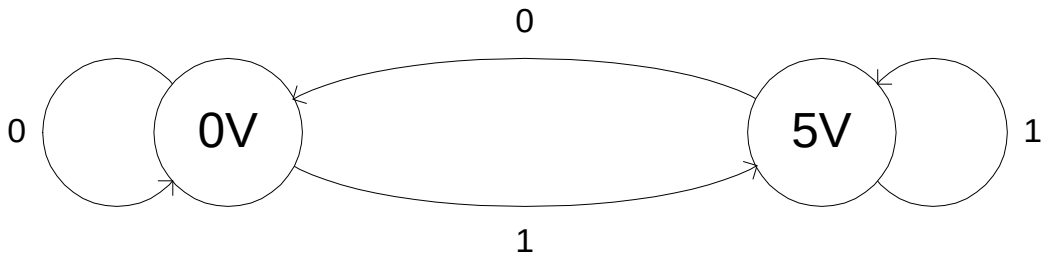
The 4B5B Block Code

- Implemented in the PCS
- Maps a block of 4 bits (a nibble) to a block of 5 bits (a code-group)
- Notice there are 2^5 code-groups and only 2^4 nibbles.
- Only use code-groups with a sufficient transition density to facilitate timing recovery.
- Notice that the 4B5B block code requires a 5/4 symbol rate increase.

Control Codes

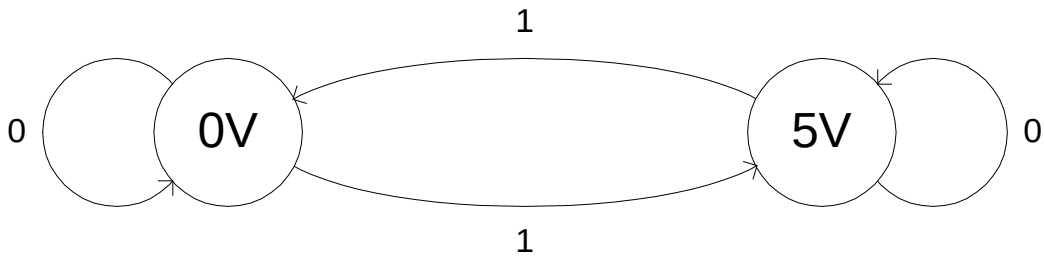
- Several code-groups are reserved for control purposes.
- The /I/ or *idle* code-group (11111) is sent when there is no data to send.
- The /J/K/ code-group pair (1100010001) indicates where idle ends and data begins.
- The /T/R/ code-group pair (0110100111) indicates where data ends and idle begins.
- The reception of any code-group other than the defined data and control code-groups is indicative of an error.

NRZ Code



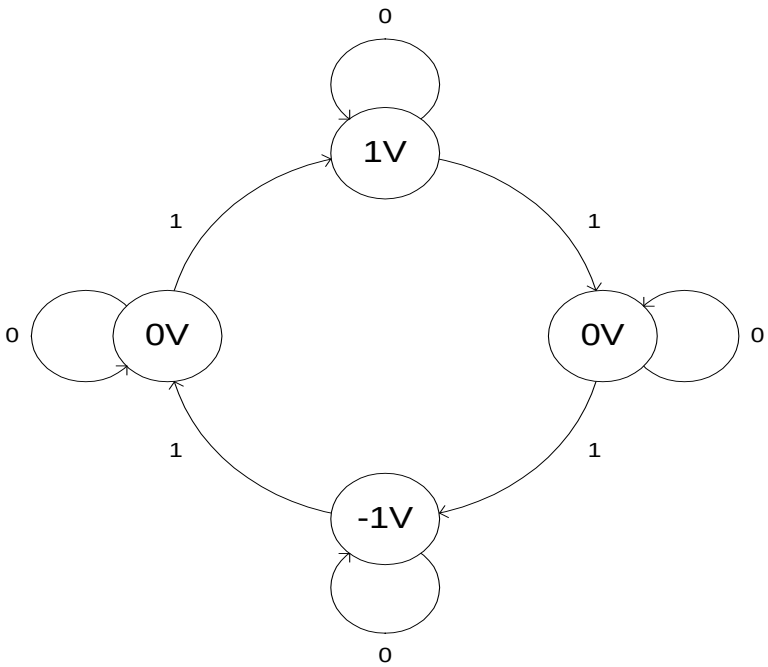
- Long runs of 0's or 1's can cause problems for timing recovery

NRZI Code



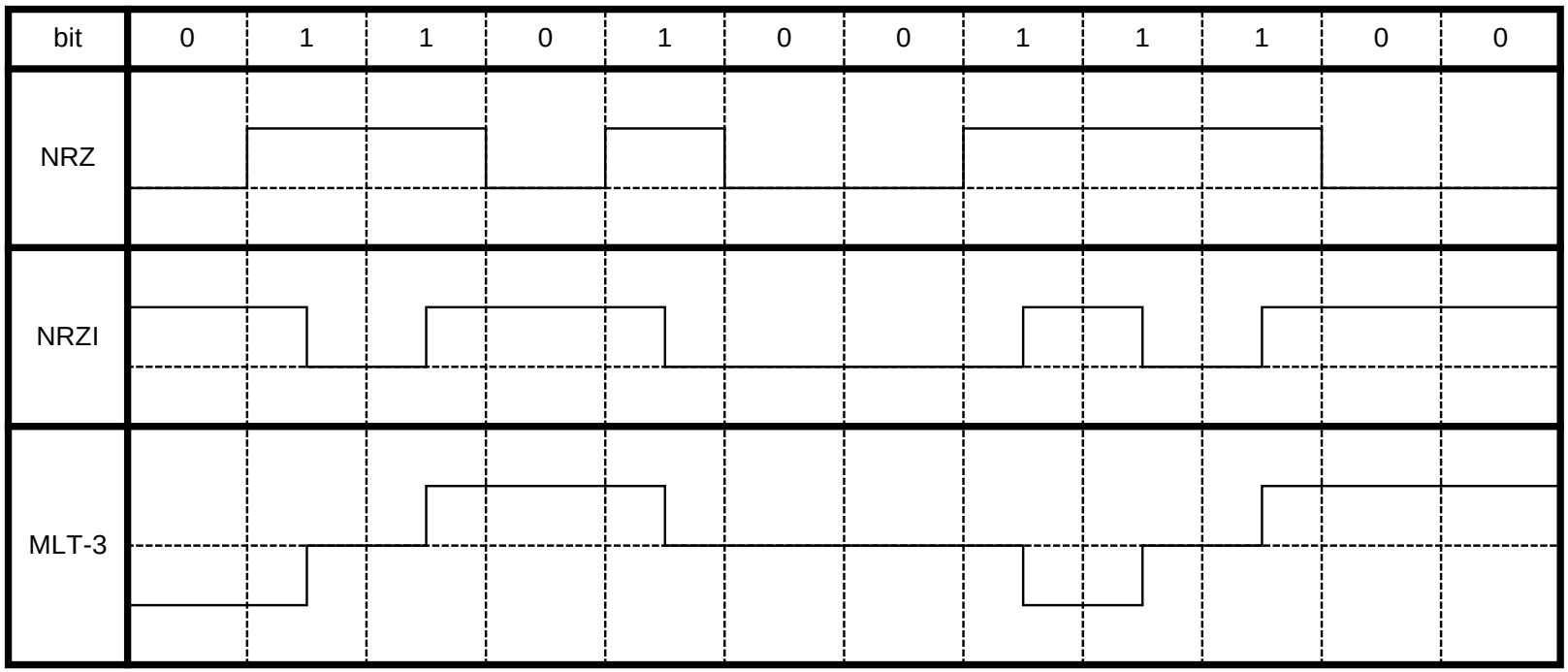
- Long runs of 1's are actually good for timing recovery
- Long runs of 0's are still a problem for timing recovery

MLT-3 Code



- Same as NRZI with the added bonus of reduced bandwidth

Summary of Coding Techniques



Transmit Functions

- NRZI-to-NRZ conversion
- Scrambler
- Encoder
- Transmitter

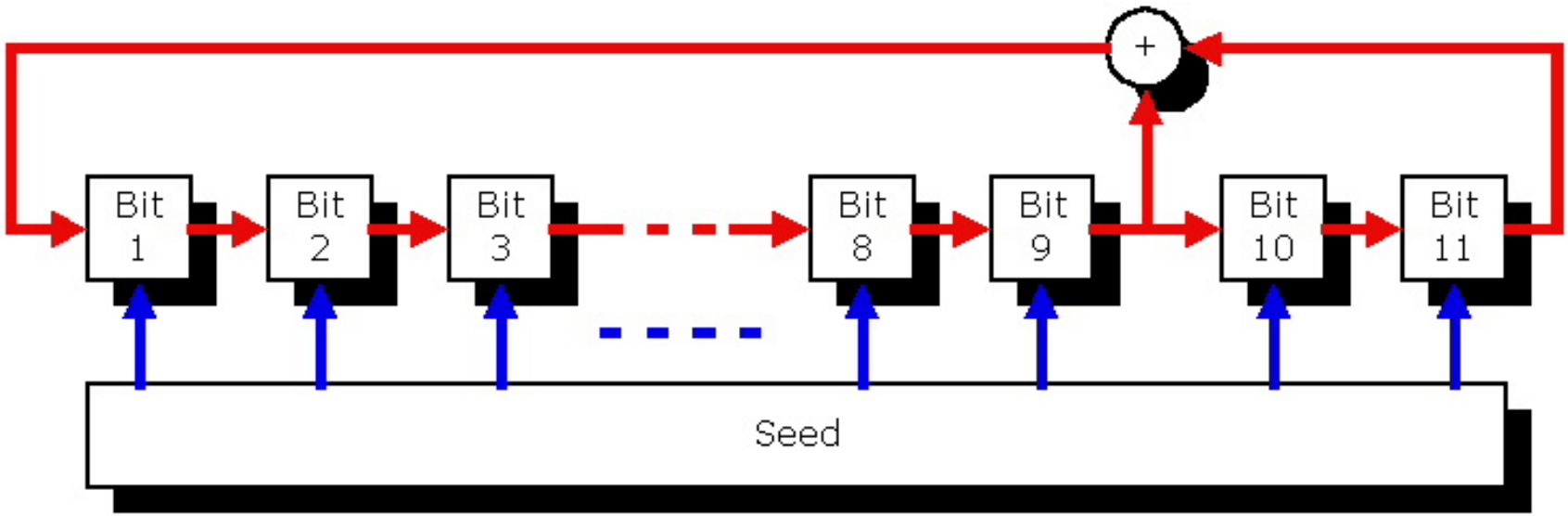
NRZI to NRZ Conversion

- The PMD receives NRZI encoded bits from the PMA
- The PMD then converts these NRZI encoded bits into NRZ encoded bits
- It does this because...

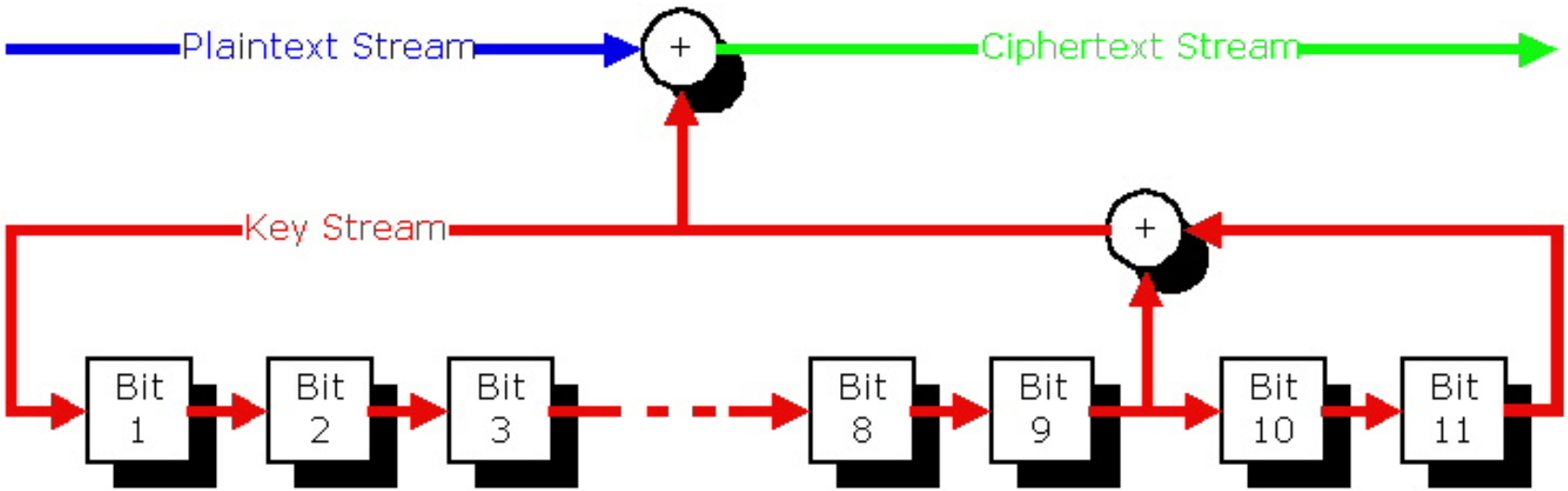
Stream Cipher Scrambler

- The purpose of the scrambler is to *whiten* the data and evenly distribute the spectral energy.
- The input to the scrambler is referred to as the *plaintext* stream.
- The scrambler adds (modulo-2) the plaintext stream to the *key* stream to produce the *ciphertext* stream.
- A linear feedback shift register (LFSR) is used to generate the key stream.

Line Feed Shift Register



Scrambler Block Diagram



Boolean analysis of stream cipher scrambler

- Let *p[n]* be the n^{th} bit of the plaintext stream.
- Let *k[n]* be the n^{th} bit of the key stream.
- Let *c[n]* be the n^{th} bit the ciphertext stream.
- ‘^‘ denotes the exclusive-or operation.
- ‘*!*’ denote the complement operation.
- *k[n] = k[n–9] ^ k[n –11]*
- *c[n] = p[n] ^ k[n]*
- Notice that for idle, *p[n] = 1*, therefore *c[n] = !k[n]*.

Scrambler synchronization

- *h[n] = c[n] ^ c[n – 9] ^ c[n – 11]*
- Remember, for idle *c[n] = !k[n]*.
- *h[n] = !k[n] ^ !k[n –9] ^ !k[n –11]*
- *h[n] = !k[n] ^ k[n]*
- Thus, for idle, *h[n] = 1*.
- The descrambler considers itself synchronized (*locked*) when it is certain idle is being sent.
- When it first transitions from *!locked* to *locked*, it loads *k'[1:11]* with *!c[1:11]*.
- Once locked, if the descrambler fails to see idle for a long period, it will go *!locked* and begin a new search for idle.

MLT-3 Encoding

- Redistributes spectral energy to lower frequencies
- Serial encoding system with three possible output symbols: +1, 0, -1
- Encoder changes state with each input one
- Direct transition between the +1 and –1 symbols are not allowed

Active Output Interface (AOI)

- Complicated expression for *transmitter*.
- Converts MLT-3 symbols to electrical signals.
- Employs differential signaling.

Receiver Functions

- Receiver
- Decoder
- Descrambler
- NRZ-to-NRZI conversion

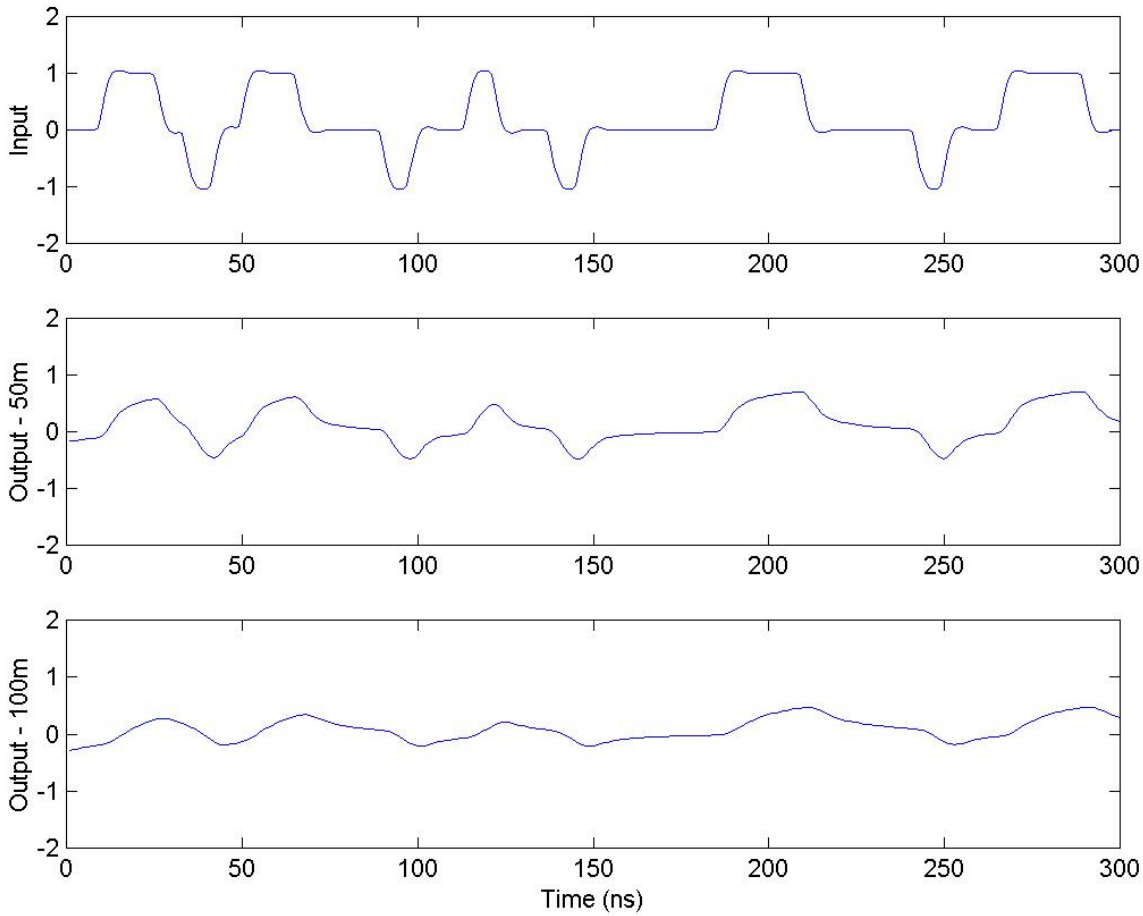
Active Input Interface (AII)

- Complicated expression for *receiver*.
- Converts electrical signals to MLT-3 symbols.
- To perform the conversion reliably, the receiver should incorporate adaptive equalization and baseline wander compensation.
- The performance of the AII is evaluated in terms of the bit error rate.

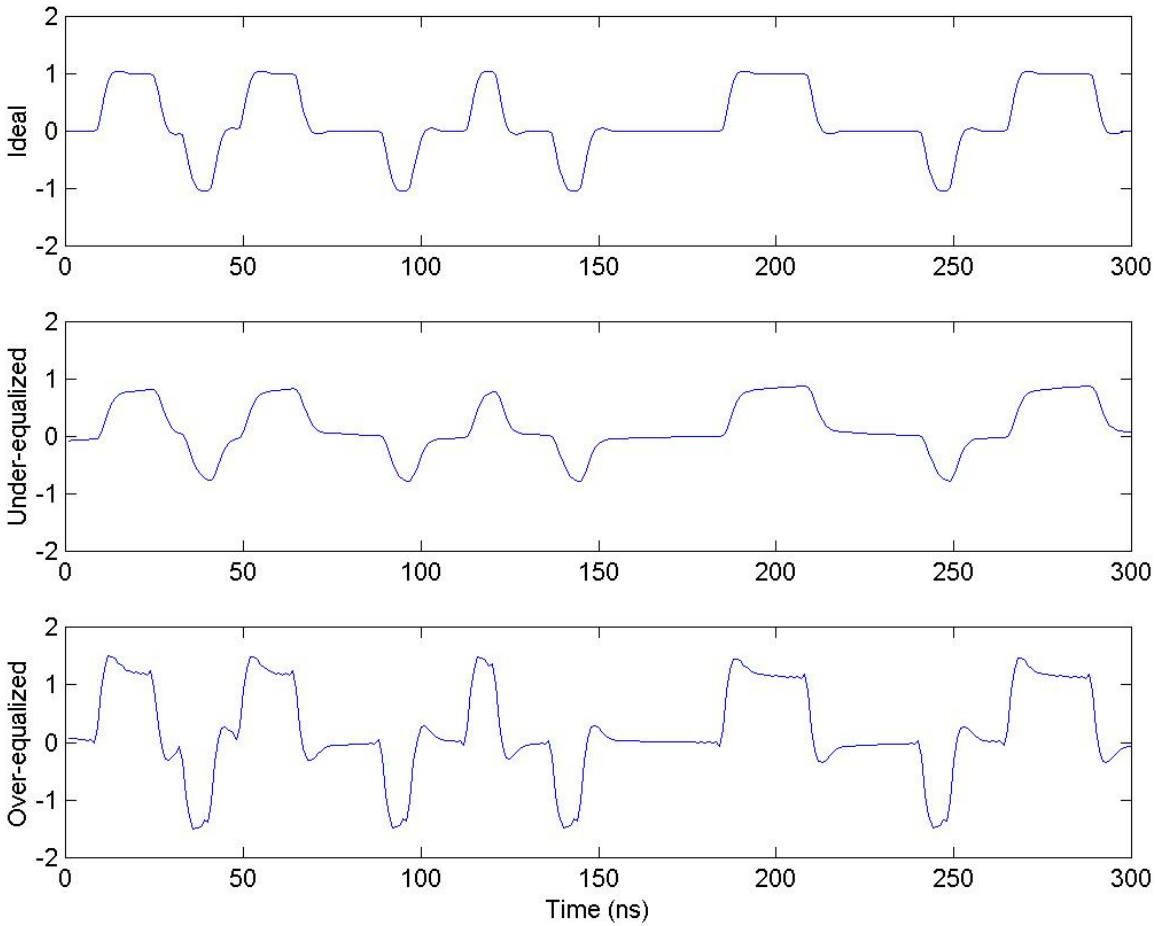
Adaptive Equalization

- Cable attenuates and distorts the signal, making it difficult for the receiver to understand.
- An *equalizer* is used to cancel the cable distortion. It implements the inverse function of the cable.
- Cable attenuation and distortion is a function of cable length.
- There are penalties for over- or under-equalizing the signal.
- An equalizer must accurately determine cable length in order to achieve the best results.

Cable Distortion



Under- and Over-Equalization Examples



Baseline Wander

- Data dependent variations in the low frequency component of a signal.
- The magnetics and cable define the low frequency response of the channel (invariably high pass).
- Any information that a signal carries at DC will be removed by the channel. This may make it more difficult for the receiver to understand.
- Packets that, after encoding, contain a significant amount of DC energy are called *killer* packets.

MLT-3 Decoder

- Observe the input for the duration of one symbol, the unit interval
- If the symbol changes during the unit interval, a logic 1 is sent
- If the symbol does not change during the unit interval, a logic 0 is sent

Stream Cipher Descrambler

- The plaintext stream may be recovered from the ciphertext stream by subtracting the key stream used by the source scrambler.
- Modulo-2 subtraction is the same as modulo-2 addition (exclusive-OR).
- The key stream is uniquely determined by the LFSR.
- The only remaining challenge is *synchronizing* the descrambler with the source scrambler.

Descrambler Synchronization

- The descrambler can achieve synchronization with 60 or fewer idle bits.
- If the input to the source scrambler is idle, then the output is the complement of the key stream.
- The descrambler simply needs a test to determine that idle is being sent.

Loss of Synchronization

- The descrambler will lose synchronization if it does not see 20 consecutive idle bits in a 1.5ms period.
- Many implementation will also lose synchronization if 58 consecutive idle bits do not occur in a 722μs period.

NRZ to NRZI Conversion

- The last function of the receiver is converting the NRZ encoded bits coming from the scrambler into NRZI encoded bits for presentation to the PMA.

Signal Detect

- A function that returns ON or OFF based on the level and quality of the signal that is received.
- Assertion of signal detect implies that the bit error rate (BER) in the receiver output is less than 10^{-2}.

In Summary

- NRZI-to-NRZ conversion
- Scrambler evenly distributes spectral energy
- Encoder redistributes to lower frequencies
- Transmitter converts encoded symbols to electrical signals