

#### 4 Links

- ► At the lowest (logical) level, network links look like serial lines.
- Encoding determines how bits and bytes are represented on a serial line,
- Framing allows the receiver to identify the beginnings and endings of packets.
- Decoding retrieves the data from the frame

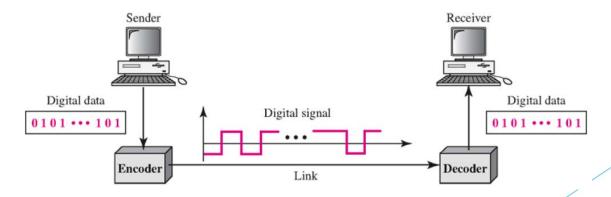
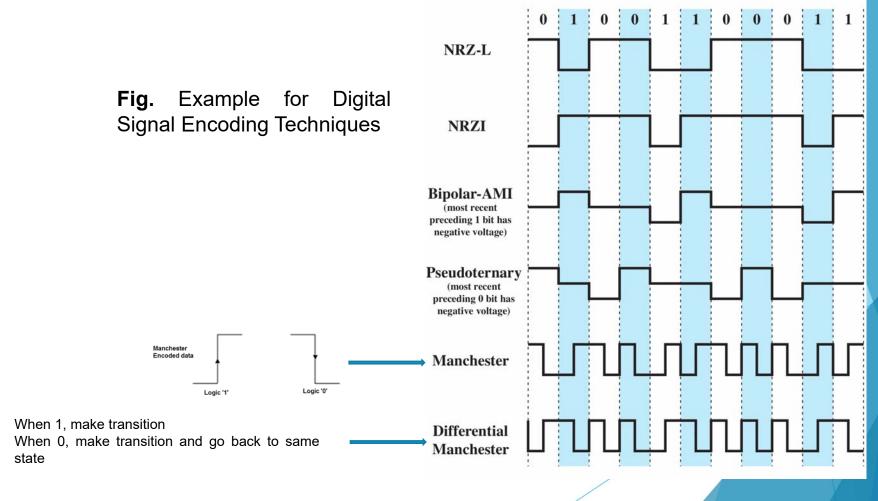


Figure. Digital data transfer

### 4.1 Encoding and Framing

- ➤ A typical serial line is ultimately a stream of bits, not bytes.
- ► Encoding is the process of converting data into a specific format for storage or transmission.
- For example: Image encoding(JPEG,PNG), Video Encoding (H.264, H.265). Audio Encoding (MP3, AAC) etc.
- ► Encoding: correctly recognizing all the bits in a stream.
- ► Framing: correctly recognizing packet boundaries

# **Digital Signal Encoding Formats**



#### 4.1.1 NRZ (Non-Return to Zero)

- NRZ (Non-Return to Zero) is perhaps the simplest encoding; it corresponds to direct bit-by-bit transmission of the 0's and 1's in the data. We have two signal levels, lo and hi, we set the signal to one or the other of these depending on whether the data bit is 0 or 1.
- Appears as raw binary bits without any coding.
- One drawback to NRZ is that we cannot distinguish between 0-bits and a signal that is simply idle. However, the more serious problem is the lack of synchronization: during long runs of 0's or long runs of 1's, the receiver can "lose count", eg if the receiver's clock is running a little fast or slow. The receiver's clock can and does resynchronize whenever there is a transition from one level to the other.
- It is a unipolar encoding technique.



NRZ Encoding: 1 = hi, 0 = lo

#### 4.1.2 NRZL (Non-Return to Zero Level)

- When the data bit is 0, encoded data is positive voltage level, which may vary according to the technology (+5V, +12V, + 3,3V etc.)
- When the data bit is 1, encoded data is negative positive level, which may vary according to the technology (-5V, -12V, -3,3V etc.)
- ▶ Polar encoding technique because it can have + and values

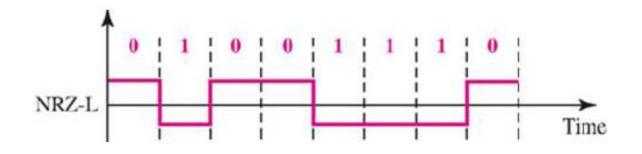
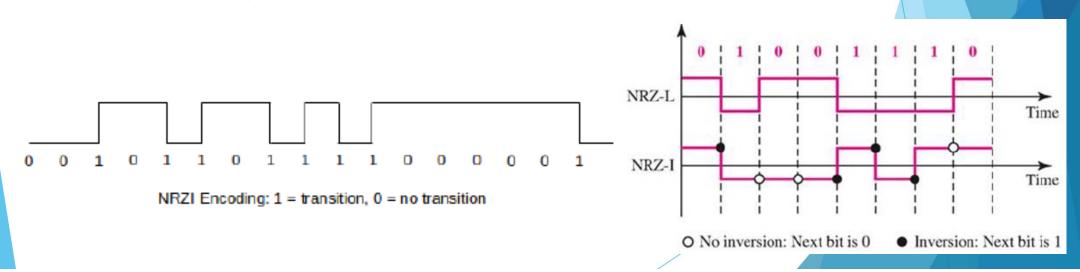


Figure. NRZL

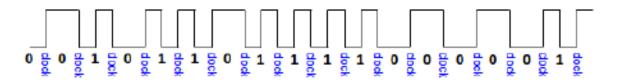
#### 4.1.3 NRZI (NRZ Inverted)

- An alternative that helps here (though not obviously at first) is NRZI, or NRZ Inverted. In this encoding, we represent a 0-bit as no change, and a 1-bit as a transition from lo to hi or hi to lo.
- Zero means "Do not change the previous state"
- One means "Change (complement) the previous state"
- ▶ Polar encoding technique because it can have + and values



#### 4.1.4 Manchester

Manchester encoding sends the data stream using NRZI, with the addition of a clock transition between each pair of consecutive data bits. This means that the signaling rate is now double the data rate, eg 20 MHz for 10Mbps Ethernet (which does use Manchester encoding). The signaling is as if we doubled the bandwidth and inserted a 1-bit between each pair of consecutive data bits, removing this extra bit at the receiver:



Manchester Encoding: NRZI alternating with clock transitions

# 4.1.4.1 Manchester Encoding

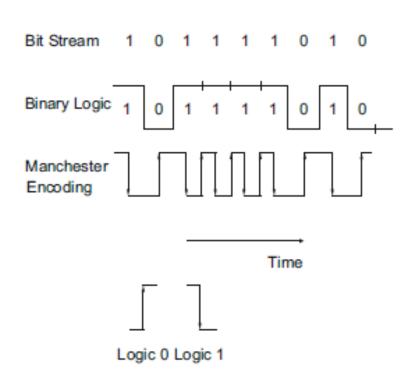


Figure 1: Manchester encoding without clock

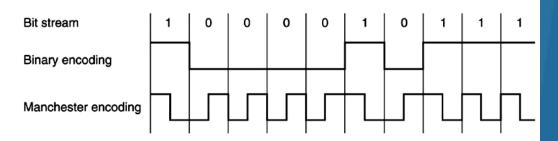


Figure 2: Manchester encoding without clock

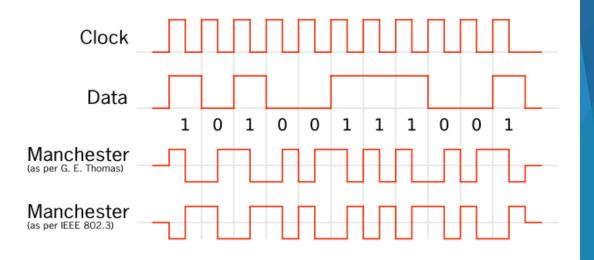


Figure 3: Manchester encoding with clock, triggered with negative edge of the clock signal

### 4.1.4.2 Differential Manchester Encoding

▶ It is also known as the Biphase mark code, etc. The presence and absence of the transition indicate the value. In Differential Manchester Encoding 0 should contain an edge but 1 should not contain any edge it should be continuous.

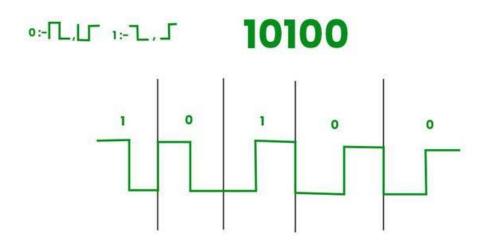
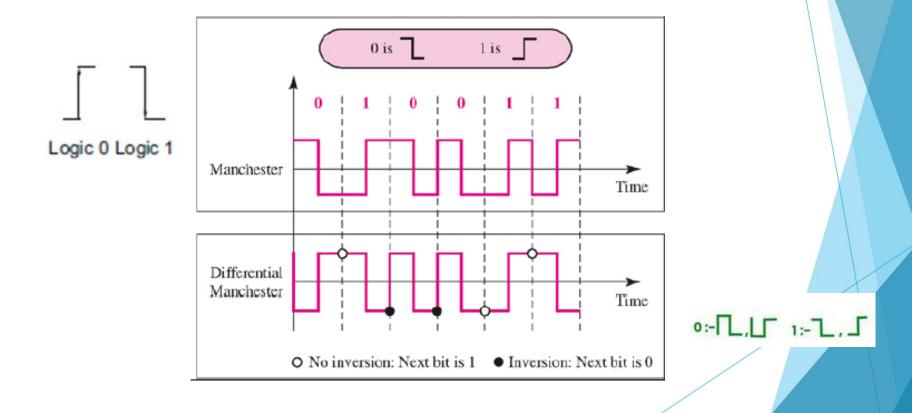


Figure 3: Manchester encoding with clock, triggered with negative edge of the clock signal

#### Manchester Encoding vs Differential Manchester Encoding



#### 4.1.5 4B/5B

In 4B/5B encoding, for each 4-bit "nybble" of data we actually transmit a designated 5-bit symbol, or code, selected to have "enough" 1-bits. A symbol in this sense is a digital or analog transmission unit that decodes to a set of data bits; the data bits are not transmitted individually. (The transmission of symbols rather than individual bits is nearly universal for high-performance links, including all forms of Ethernet faster than 10Mbps and all Wi-Fi links.)

The idea of 4B/5B is to insert extra bits into the bit stream so as to break up long sequences of 0s or 1s. Specifically, every 4 bits of actual data are encoded in a 5-bit code that is then transmitted to the receiver hence, the name 4B/5B. The 5-bit codes are selected in such a way that each one has no more than one leading 0 and no more than two trailing 0s. Thus, when sent back-to-back, no pair of 5-bit codes results in more than three

consecutive 0s being transmitted.

data	symbol	data	symbol	
0000	11110	1011	10111	
0001	01001	1100	11010	
0010	10100	1101	11011	
0011	10101	1110	11100	
0100	01010	1111	11101	
0101	01011	IDLE	11111	
0110	01110	HALT	00100	
0111	01111	START	10001	
1000	10010	END	01101	
1001	10011	RESET	00111	
1010	10110	DEAD	00000	

### 4.1.6 Framing

- How does a receiver tell when one packet stops and the next one begins, to keep them from running together?
- We have already seen the following techniques for addressing this framing problem: determining where packets end:
- Interpacket gaps (as in Ethernet)
- 4B/5B and special bit patterns
- Putting a length field in the header would also work, in principle, but seems not to be widely used. One problem with this technique is that restoring order after desynchronization can be difficult.

# 4.1.6.1 HDLC(High-level Data Link Control)

- ► HDLC (High-level Data Link Control) is a general link-level packet format used for a number of applications, including Point-to-Point Protocol (PPP) (which in turn is used for PPPoE PPP over Ethernet which is how a great many Internet subscribers connect to their ISP), and Frame Relay, still used as the low-level protocol for delivering IP packets to many sites via telecommunications lines. HDLC supports the following two methods for frame separation:
- HDLC over asynchronous links: byte stuffing
- HDLC over synchronous links: bit stuffing
- ▶ The basic encapsulation format for HDLC packets is to begin and end each frame with the byte 0x7E, or, in binary, 0111 1110. The problem is that this byte may occur in the data as well; we must make sure we don't misinterpret such a data byte as the end of the frame.

Flag (7E)	Address (8bits x n)	Control (8bits x n)	Data	FCS (16/32 bits)	Flag (7E)
	(00113 × 11)	(05/10 × 1)		(10/32 5/13)	

Figure. HDLC Frame Format

# 4.1.6.1 HDLC(High-level Data Link Control)

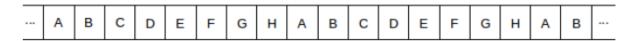
- Over synchronous serial lines (typically faster than asynchronous), HDLC generally uses bit stuffing. The underlying bit encoding involves, say, the reverse of NRZI, in which transitions denote 0-bits and lack of transitions denote 1-bits. This means that long runs of 1's are now the problem and runs of 0's are safe.
- ▶ Whenever five consecutive 1-bits appear in the data, eg 011111, a 0-bit is then inserted, or "stuffed", by the transmitting hardware (regardless of whether or not the next data bit is also a 1).
- The HDLC frame byte of 0x7E = 0111 1110 thus can never appear as encoded data, because it contains six 1-bits in a row. If we had 0x7E in the data, it would be transmitted as 0111 11010.

# 4.1.6.1 HDLC(High-level Data Link Control)

- **Example:**
- ▶ Data: 011110 0111110 01111110
- Sent as: 011110 011111100 0111111010 (stuffed bits in bold)
- Note that bit stuffing is used by HDLC to solve two unrelated problems: the synchronization problem where long runs of the same bit cause the receiver to lose count, and the framing problem, where the transmitted bit pattern 0111 1110 now represents a flag that can never be mistaken for a data byte.

### 4.2. Time Division Multiplexing

- There are other ways for multiple channels to share a single wire. One approach is frequency-division multiplexing, or putting each channel on a different carrier frequency. Analog cable television did this. Some fiber-optic protocols also do this, calling it wavelength-division multiplexing.
- But perhaps the most pervasive alternative to packets is the voice telephone system's time division multiplexing or TDM, sometimes prefixed with the adjective synchronous. The idea is that we decide on a number of channels, N, and the length of a timeslice, T, and allow each sender to send over the channel for time T, with the senders taking turns in round-robin style. Each sender gets to send for time T at regular intervals of NT, thus receiving 1/N of the total bandwidth.



Time-Division Multiplexing