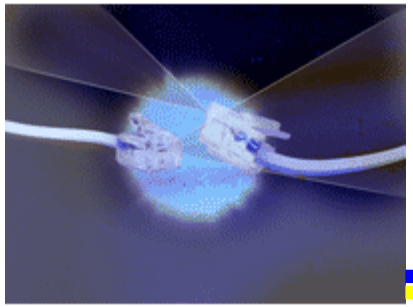
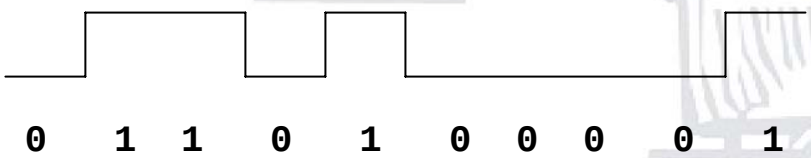
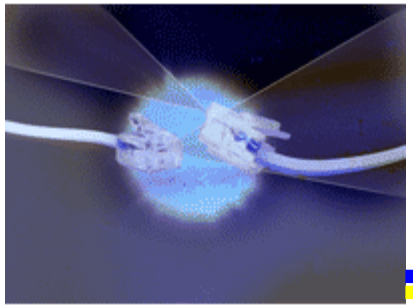


Encoding

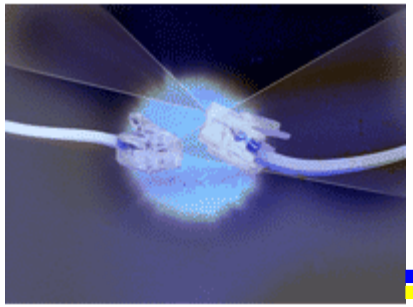
- Carrier signal is somehow modulated to have two discrete signals
- Need to encode bits onto these two signals. Easiest way is to assign one level to “0” and one level to “1”
  - This is *NRZ* encoding (Non-Return to Zero)





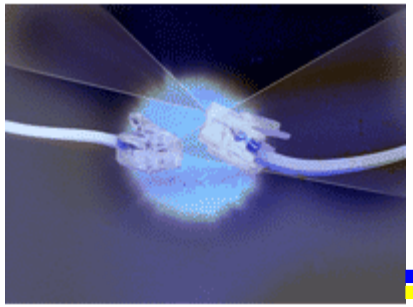
NRZ Encoding (cont.)

- Problem – too many consecutive bits with the same value causes average signal value to shift in that direction, making it harder for receiver to detect next bit.
- Also, lack of transitions make it difficult to recover clock from signal.
- One solution is to use *NRZI* encoding (Non-Return to Zero Inverted)

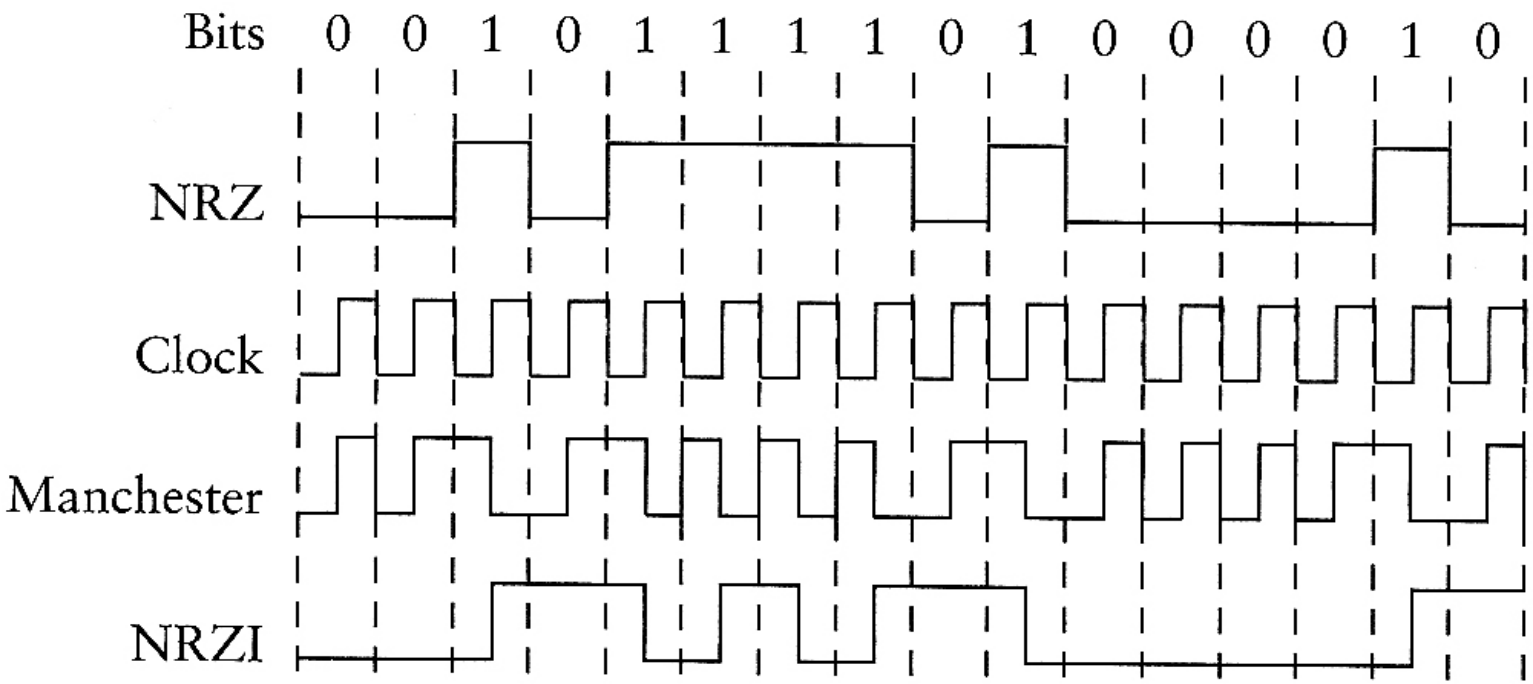


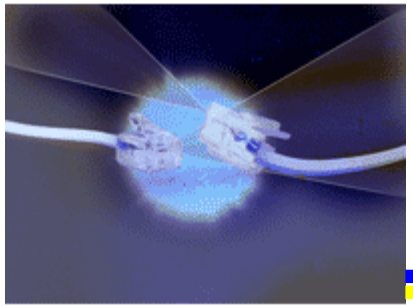
Alternative Encodings

- Encode 1 as a transition and 0 as no transition
  - Fixes problem with sequences of 1s, but not 0s
- Manchester encoding – XOR clock and NRZ-encoded data
  - 0 is low-to-high transition, 1 is high-to-low
  - Transition every bit, so clock recovery easy
  - Doubles the transition rate, or *baud rate*, so efficiency is only 50% (bit rate / baud rate).



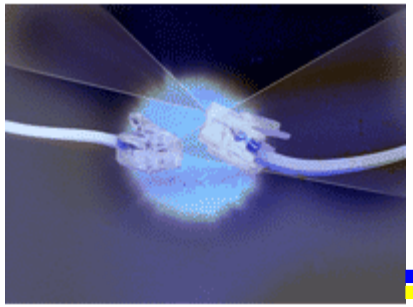
Encodings





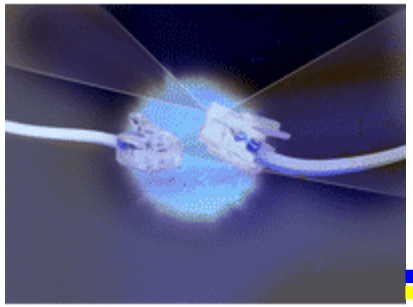
4B/5B Encoding

- Similar to NRZI – avoid the inefficiency of Manchester encoding by ensuring bit transitions
  - Transmit groups of 4 bits using 5 bits – ensures that there is no more than one leading zero and no more than two trailing zeroes
  - Back-to-back groups cannot cause a run of more than three zeroes
  - Only 80% efficiency, but better than 50%



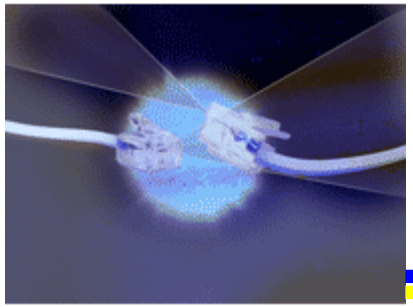
Bit Stuffing

- Another alternative is to insert bits only as required to break up long runs.
  - Use NRZI encoding, and if data contains five zero bits in a row, insert a one bit in the stream.
  - Receiver will remove all one bits following five zeroes
  - *Bit oriented* protocol, since data stream is no longer a fixed number of bits per byte



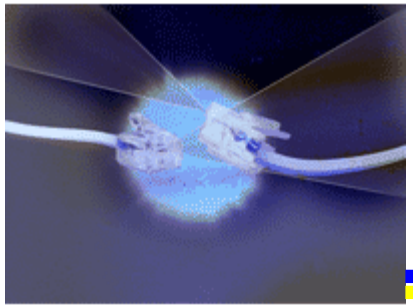
Sync vs. Async

- Asynchronous – no separate clock signal.
  - Each side needs to have an approximately correct clock to sample the data
  - Transmit start bits to sync clock with signal
  - Need stop bit or bits so you can recognize next start bit
- Synchronous – separate line carrying clock signal



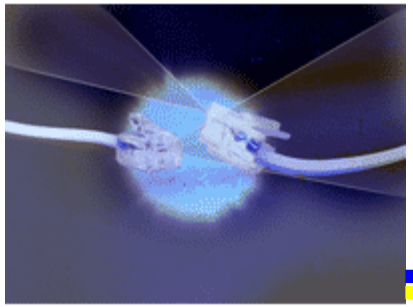
Framing

- Group bits of data into message blocks, called *frames*, for transmission over link
- The main problem is finding start and end of each frame. Need to make sure these points can be distinguished from data content of frame.
- Different ways to do this
  - Byte-oriented or character-oriented protocols
  - Bit-oriented protocols



Character-Oriented Protocols

- Sentinel values
  - BPS (Burroughs Poll Select), PPP (Point-to-Point Protocol - both async
  - BSC (Binary Synchronous Communication) – sync
- Byte counting
  - DDCMP (Digital Data Communication Message Protocol)



Bit-Oriented Protocols

- SDLC, HDLC, LAPB (all async)
  - All very similar, use a special flag sequence (01111110) to delimit frames
  - Stuff zero bits into data to prevent runs of more than five ones