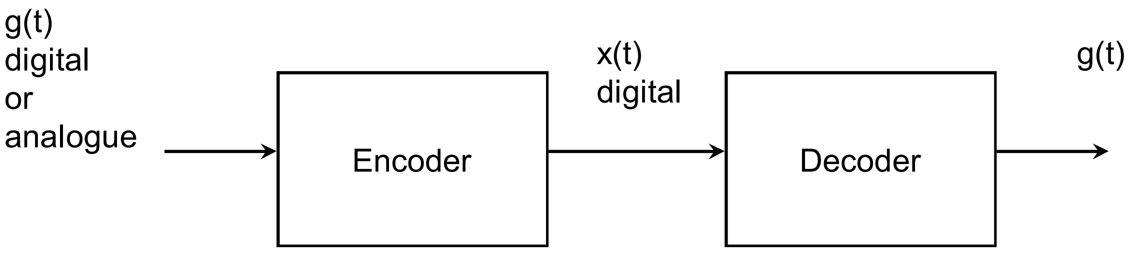
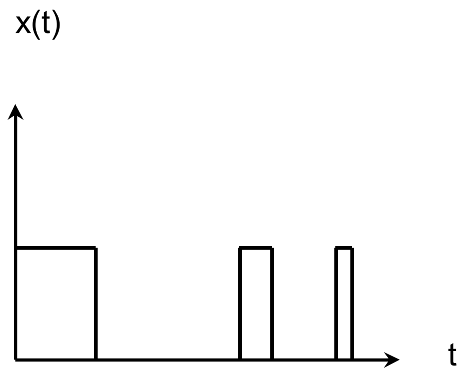
Data Encoding

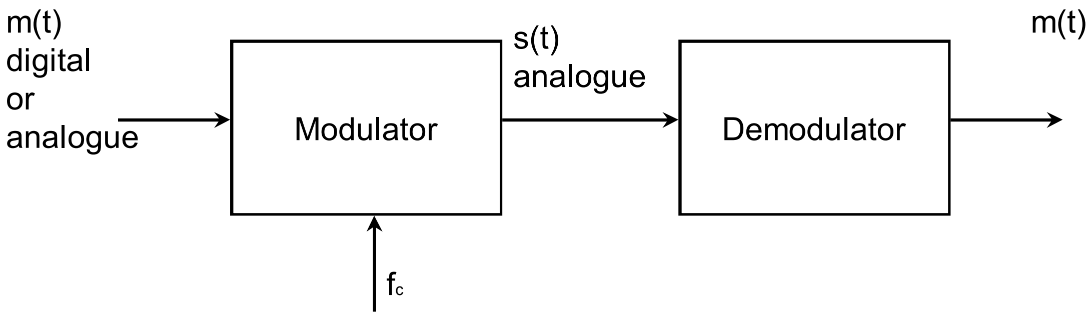
- Data are propagated from point to point by encoding data into signals
- The data may be *analogue* or *digital*
- The signals may be analogue or digital
  - Two devices used for producing the signals:
    - CODEC produces digital signals
    - MODEM produces analogue signals

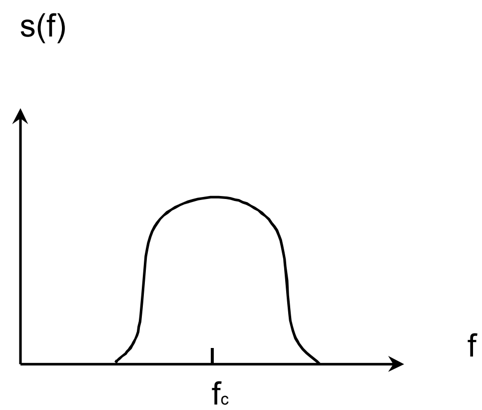
Data Encoding Devices





Encoding onto a digital signal





Modulation onto an analogue signal

Options for data encoding

- Digital Signal Encoding involves the use of a CODEC:
  - Excellent for transporting computer data as it's already in digital form i.e. Digital data
  - Analogue data can also be carried
- Analogue Signal Encoding involves the use of a MODEM:
  - Excellent for transporting voice and video data as they are already in analogue form i.e. Analogue data
  - Allows for transmission of Digital Data across transmission systems that can only deal with analogue signals
- Each of the above will be explored in turn
  - Firstly, the transport (encoding) of Digital Data onto a Digital Signal

Digital Signal Encoding Schemes

Nonreturn to Zero-Level (NRZ-L)
- 0 = high level
- 1 = low level

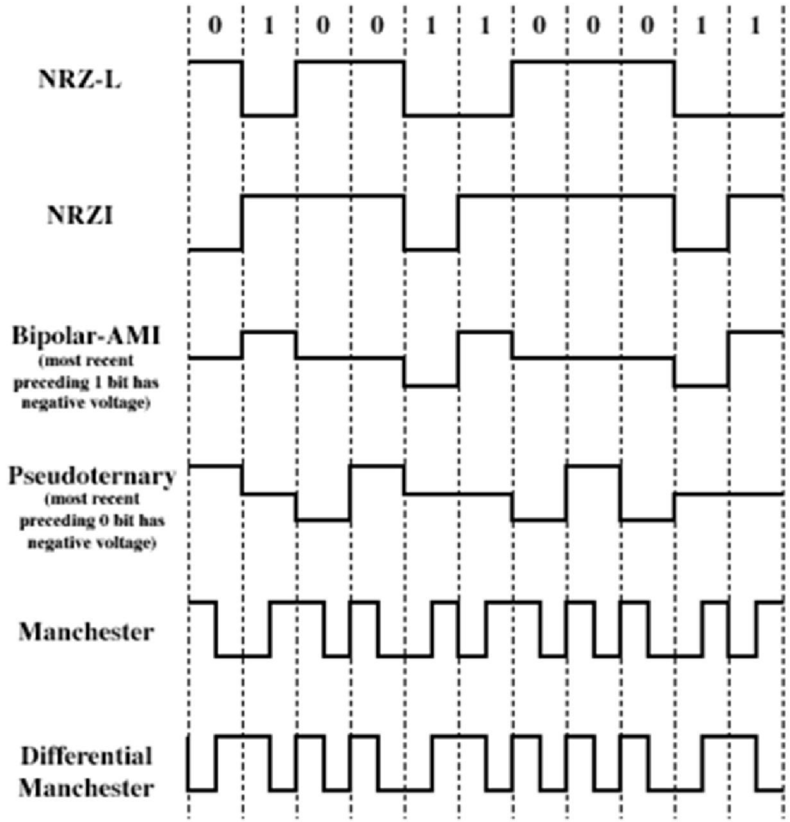
Nonreturn to Zero Inverted (NRZI)
- 0 = no transition at beginning of interval (one bit time)
- 1 = transition at beginning of interval

Bipolar-AMI
- 0 = no line signal
- 1 = positive or negative level, alternating for successive ones

Pseudoternary
- 0 = positive or negative level, alternating for successive zeros
- 1 = no line signal

Manchester
- 0 = transition from high to low in middle of interval
- 1 = transition from low to high in middle of interval

Differential Manchester
- Always a transition in middle of interval
- 0 = transition at beginning of interval
- 1 = no transition at beginning of interval



Digital Data onto *Digital Signals*

- Here the binary data are transmitted by encoding each data bit into *signal elements*
- Each *signal element* may represent one or more data bits

Some terminology:

- ***Signal Element***
  - ***Analogue*****:** a pulse of constant frequency, phase and amplitude
  - ***Digital*****:** a voltage pulse of constant amplitude
- ***Unipolar*** – all signal elements have the same algebraic sign i.e. all +*ve* or all –*ve*
- ***Polar*** – signal elements can have different algebraic sign
- ***Mark/Space*** – refers to binary one/zero

Digital Data onto *Digital Signals* - Terminology

- ***Data element*** - the bits
- ***Data rate*** - the rate at which data are transmitted
  - Measured in *bits per second*
- ***Modulation rate*** - the rate at which the *signal level* is changed per second
  - Represents the number of *signal elements* transmitted per second and is expressed in *baud*
  - *Bit rate* and *baud* are equal if, <u>and only if</u>, one signal element represents one data bit
- ***Bit duration*** – the time to transmit one bit

NRZ Encoding Schemes

- Non-Return to Zero (NRZ) techniques
- Two NRZ techniques to consider:
  - NRZ-L:
    - The easiest way to encode data
    - It is used by computer terminals and other devices
    - Easy to implement
    - Level (amplitude) and polarity are important
  - NRZI:
    - An example of *differential encoding*
    - Relies on the voltage *transitions* rather than actual *levels*
    - Easier to detect a *signal transition* in presence of noise
    - Polarity is not important

NRZ Encoding Schemes

- NRZ techniques make good use of bandwidth
  - On average one bit per signal element
- Limitations of NRZ techniques:
  - Presence of a *dc component*
  - Poor *synchronisation*

Multilevel Binary Encoding Schemes

- Multilevel Binary techniques, two to consider:
  - *Bipolar AMI*
  - *Pseudoternary*
  - Both techniques use three signal/voltage levels
- Bipolar AMI
  - *Space* represented by **<u>no</u>** line signal i.e. zero volts
  - *Mark* represented by alternating +ve and –ve voltages i.e. *bipolar*
- Pseudoternary: very similar to Bipolar AMI

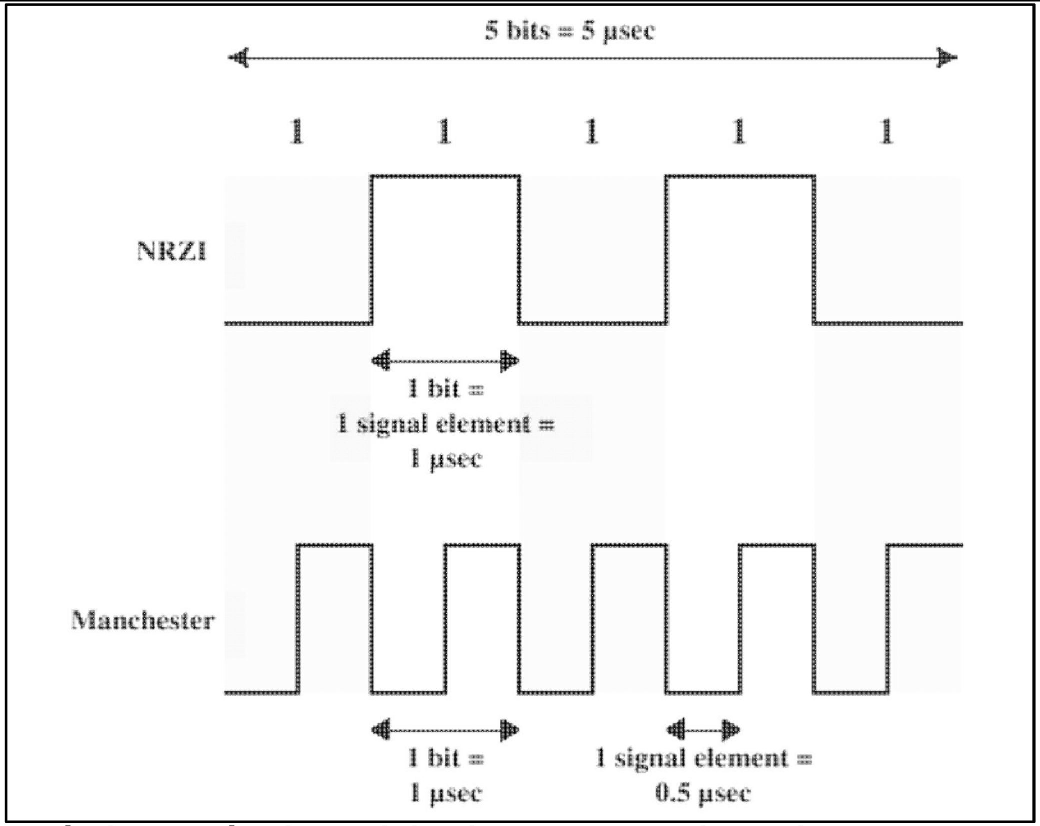
Multilevel Binary Encoding Schemes

- Comparison with NRZ techniques:
- **Advantages** of Multilevel Binary techniques over NRZ techniques:
  - No DC component
  - Synchronisation is easier
  - Provides simplistic means of error detection
- **Disadvantages** over NRZ techniques:
  - Less efficient in terms of number of encoded bits per signal level
  - More voltage changes for the same number of bits
  - Forces the receiver to work harder to interpret bits

Biphase Encoding Schemes

- Biphase, two techniques to consider:
  - *Manchester*
  - *Differential Manchester*
- Comparison with NRZ techniques:
  - Modulation rate is twice that for NRZ, implies greater signal BW
  - Synchronisation is provided within each bit interval
  - <u>No</u> DC component
  - Better low level error detection
  - Good performance in the presence of noise

Modulation Rate V Bit Rate



- In general the relationship between the *data rate* and *baud* can be expressed as follows:

$$\mathbf{D = R/b}$$

D = Mod. Rate, R = Data Rate and b = bits per sig. element

Comparison of Digital Signal Encoding Schemes

	NRZ	*Multilevel Binary*	*Biphase*
Bandwidth	Smallest	Same as NRZ	Double of NRZ or Multilevel Binary
DC component	Presence of DC component leads to power wastage	Zero DC component	Zero DC component
Synchronization	String of continuous 0s (and 1s) leads to loss in synchronization	String of continuous 0s (or 1s) leads to loss of synchronization	Transition at middle of pulse allows synchronization
Error Detection	No capability	No capability	Built in capability because of transition
Maximum Modulation Rate	Same as data rate. NRZL: for 1010… NRZI: for 1111…	Same as data rate. Bipolar: for 1111… Pseudo: for 0000…	Double of data rate. For 000…

Reading Digital Data *off* a Digital Signal

- To interpret a digital signal the receiver must:
  - Be able to read the *signal elements* e.g. voltage level
  - Know when a bit starts and ends (i.e. bit duration - timing)
- Successful reception of data depends on
  - Adequate Signal-to-noise ratio – already covered
  - Sufficient bandwidth – already covered
  - Sufficiently slow data rate – already covered