Data Communications

Synchronization

Motivation

- Whenever a device transmits digital data to another device, there must be a certain rhythm established between the two devices.
- The receiving device must have some way of knowing where each unit of data begins and where it ends
- The signal must be synchronized in a way that the receiver can distinguish the bits and bytes.

Synchronous and Asynchronous Systems

- The synchronous transmission methods combine data with clock in the created signal.
- The asynchronous transmission methods send only data bits. The receiver uses transitions on that signal to figure out the transmitter bit rate and timing.

Synchronization Levels

- Synchronization is carried out in three levels:
  - Bit level
  - Byte (Character) level
  - Frame level

Synchronous Transmission

- Clock information is embedded into the data
- Manchester encoding for instance, has a transition at the center of each bit cell.
- This transition provides a periodic pulse (clock)

Asynchronous Transmission with Digital Phase Lock Loop (DPLL)

- Sender encodes data with a clock rate of TxC using NRZI
- Receiver uses a clock (RxC) with a rate of N time the transmitter clock RxC = N . TxC
- Receiver uses a counter which counts down from N to zero

Digital Phase Lock Loop (DPLL)

- Detecting the first transition, counter in initialized to N/2
- When the counter value is zero, N is loaded into it again.
- Zero values of the counter indicate bit cell centers

Bit Cell Center Deviation Correction

- Receiver expects a bit transition at the beginning of the bit cell when the counter has a value of N/2
- If the receiver detects a transition when the counter is more or less than N/2, it makes a correction by adding to or subtracting from the counter.

Maximum Bit Cell Center Deviation

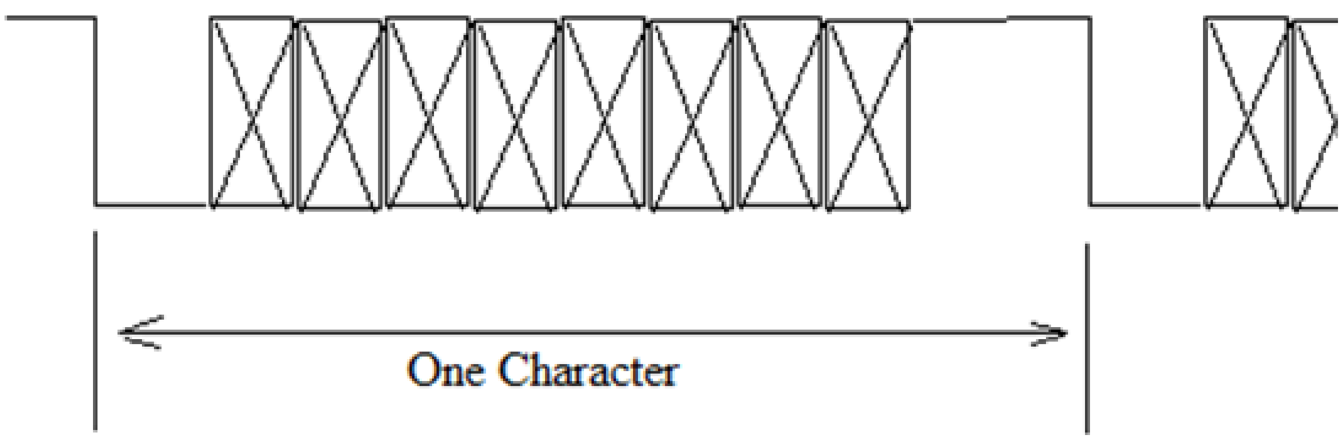
- Maximum bit cell center deviation is equal to one pulse of the receiver clock
- Maximum deviation percentage is TxC/RxC x 100%
  - E.g. RxC = 2400 Hz, TxC = 600 Hz
  - Max Deviation Percentage = 600/2400 x 100 % = 25%

Character Level Synchronization

- Character level synchronization is achieved simply by counting the bits.
- For instance: If one start bit, 8 data bits, and two stop bits are used to define a character, every 11 bits will give a character.

Character Level Synchronization with Start and Stop Bits

- Adds a ‘0’ to the beginning of the character, and two ‘1’ bits to the end.
- Always a character starts with a transition from ‘1’ to ‘0’



Frame Level Synchronization

- Frame:
  - The data is divided into units called **Frame** before transmission.
  - Each frame include a header which provides information about data such as sender, receiver, length, etc.
  - A frame may also include a trailer at the end. Trailer include error checking code such as CRC.
  - A frame uses special codes or bit patterns to define the frame boundaries. (**Delimiter**)

Frame Level Synchronization

- Character Stuffing
  - A character is used to define the beginning or end of the frame (**delimiter**)
  - If delimiter happens in the frame data, an ESC character is inserted before it
  - If ESC occurs in data, a second ESC is inserted before it.

Frame Level Synchronization

- Bit Stuffing
  - To define the frame boundaries, a bit pattern is used (**delimiter**).
  - In case of occurring of the pattern in the frame data (**Delimiter collision**), an extra bit is inserted

Bit Stuffing Example

- Delimiter : 01111110
- Sender: Insert a 0 after 5 consecutive 1s
- Receiver: If the bit after 5 consecutive 1 is a 0 then remove it