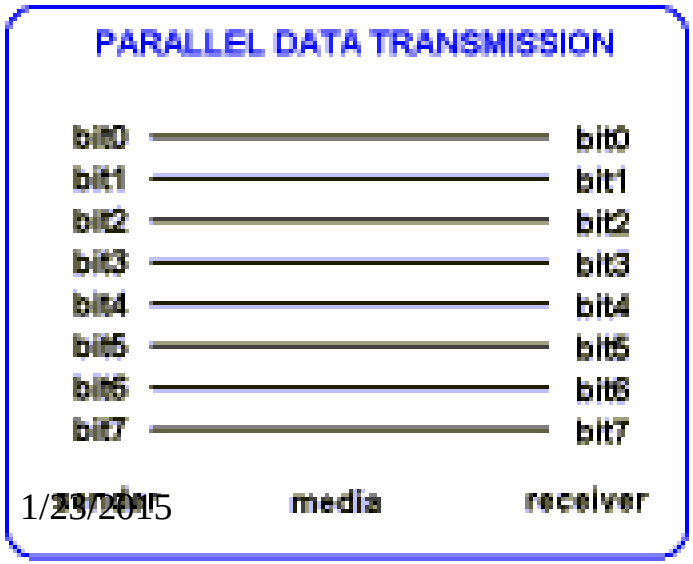
PC-PC Communication

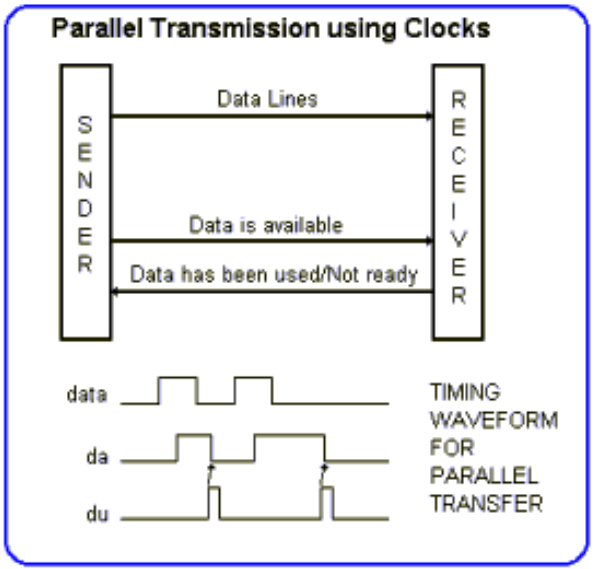
Data Transmission

- **Data transmission** is the transfer of data from point-to-point often represented as an electromagnetic signal over a physical point-to-point or point-to-multipoint communication channel
- A communication channel refers to the medium used to convey information from a sender (or transmitter) to a receiver, and it can use fully or partially the medium.
- Examples of channels: copper wires, optical fibbers or wireless communication channels.

Parallel and Serial Data

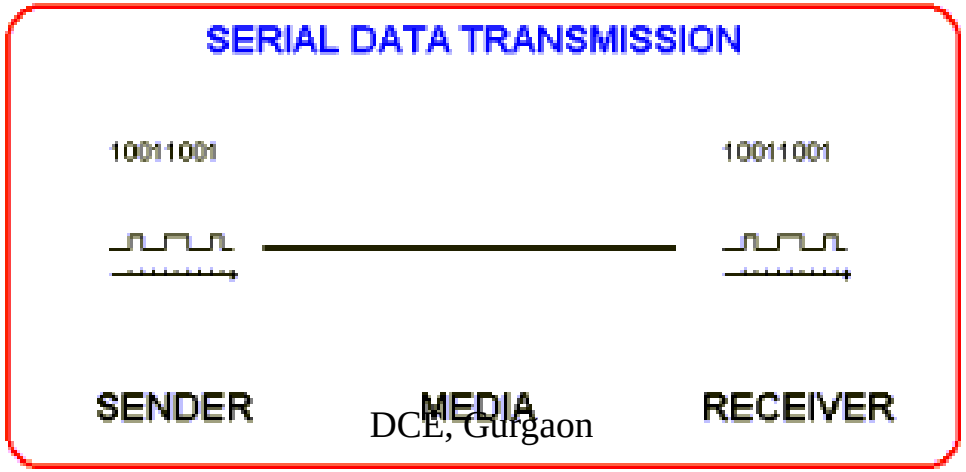
- Data may be transmitted between two points in two different ways. Lets consider sending 8 bits of digital data (1 byte)
- Parallel transmission
  - Each bit uses a separate wire
  - To transfer data on a parallel link, a separate line is used as a clock signal. This serves to inform the receiver when data is available. In addition, another line may be used by the receiver to inform the sender that the data has been used, and its ready for the next data.





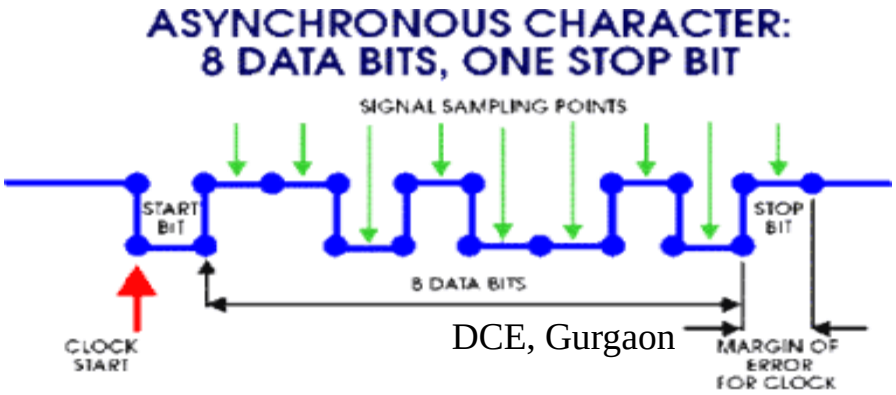
Parallel and Serial Data

- Serial
  - Each bit is sent over a single wire, one after the other
  - Usually no signal lines are used to convey clock (timing information)
  - There are two types of serial transmission, essentially having to do with how the clock is embedded into the serial data
    - Asynchronous serial transmission
    - Synchronous serial transmission
- If no clock information was sent, the receiver would misinterpret the arriving data (due to bits being lost, going too slow).
- Parallel transmission is obviously faster, in that all bits are sent at the same time, whereas serial transmission is slower, because only one bit can be sent at a time. Parallel transmission is very costly for anything except short links.



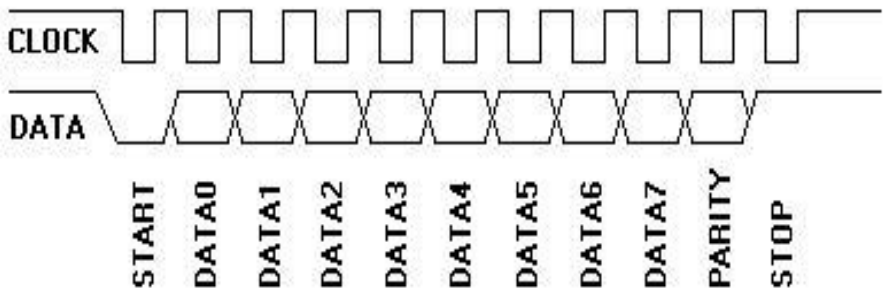
Asynchronous Serial Transmission (RS232 Example)

- Because no signal lines are used to convey clock (timing) information, this method groups data together into a sequence of bits (five to eight), then prefixes them with a start bit and a stop bit. This is the method most widely used for PC or simple terminal serial communications.
- In asynchronous serial communication, the electrical interface is held in the **mark** position between characters. The start of transmission of a character is signaled by a drop in signal level to the **space** level. At this point, the receiver starts its clock. After one bit time (the start bit) come 8 bits of true data followed by one or more stop bits at the mark level.
- The receiver tries to sample the signal in the middle of each bit time. The byte will be read correctly if the line is still in the intended state when the last stop bit is read.
- Thus the transmitter and receiver only have to have **approximately the same clock rate**. A little arithmetic will show that for a 10 bit sequence, the last bit will be interpreted correctly even if the sender and receiver clocks differ by as much as 5%.
- It is **relatively simple**, and therefore inexpensive. However, it has a **high overhead**, in that each byte carries at least two extra bits: a 20% loss of line bandwidth.



Synchronous Serial Transmission (PS2 Example)

- The PS/2 mouse and keyboard implement a bidirectional synchronous serial protocol.
- The bus is "**idle**" when both lines are high (open-collector). This is the only state where the keyboard/mouse is allowed begin transmitting data. The host has ultimate control over the bus and may inhibit communication at any time by pulling the Clock line low.
- The device (slave) always generates the clock signal. If the host wants to send data, it must first inhibit communication from the device by pulling Clock low. The host then pulls Data low and releases Clock. This is the "Request-to-Send" state and signals the device to start generating clock pulses.
  - Summary: Bus States
    Data = high, Clock = high: *Idle state.*
    Data = high, Clock = low: *Communication Inhibited.*
    Data = low, Clock = high: *Host Request-to-Send*



Data is transmited 1 byte at a time:
•1 start bit. This is always 0.
•8 data bits, least significant bit first.
•1 parity bit (odd parity - The number of 1's in the data bits plus the parity bit always add up to an odd number. This is used for error detection.).
•1 stop bit. This is always 1.
•1 acknowledge bit (host-to-device communication only)

Serial Communication

Name	Sync /Async	Type	Duplex	Max devices	Max speed (Kbps)	Max distance (feet)	Pin count (not including ground)
RS-232	async	peer	full	2	115.2	30	2 (or 4 with HW handshake)
RS-422	async	multi-drop	half	10	10000	4,000	1 (unidirectional only, additional pins for each bidirectional comm.)
RS-485	async	multi-point	half	32	10000	4,000	2
I^2C	sync	multi-master	half	Limitation based on bus capacitance and bit rate	3400	<10	2
SPI	sync	multi-master	full	Limitation based on bus capacitance and bit rate	>1000	<10	3+1(Additional pins needed for every slave if slave count is more than one)
Microwire	sync	master/slave	full	Limitation based on bus capacitance and bit rate	>625	<10	3+1(Additional pins needed for every slave if slave count is more than one)
1-Wire	async	master/slave	half	Limitation based on bus capacitance and bit rate	16	1,000	1

Data Communication Terminology

- **Channel**
  - A channel is a portion of the communications medium allocated to the sender and receiver for conveying information between them. The communications medium is often subdivided into a number of separate paths, each of which is used by a sender and receiver for communication purposes.
- **Baud Rate**
  - Baud rate is the same as symbol rate and is a measure of the number of line changes which occur every second. Each symbol can represent or convey one (binary encoded signal) or several bits of data. For a binary signal of 20Hz, this is equivalent to 20 baud (there are 20 changes per second).
- **Bits Per Second**
  - This is an expression of the number of data bits per second. Where a binary signal is being used, this is the same as the baud rate. When the signal is changed to another form, it will not be equal to the baud rate, as each line change can represent more than one bit (either two or four bits).
- **Bandwidth**
  - Bandwidth is the frequency range of a channel, measured as the difference between the highest and lowest frequencies that the channel supports. The maximum transmission speed is dependant upon the available bandwidth. The larger the bandwidth, the higher the transmission speed.

Protocols and Synchronization

- **Protocols**
  - A protocol is a set of rules which governs how data is sent from one point to another. In data communications, there are widely accepted protocols for sending data. Both the sender and receiver must use the same protocol when communicating.
  - BY CONVENTION, THE LEAST SIGNIFICANT BIT IS TRANSMITTED FIRST (Network order)
- **ASYNCHRONOUS PROTOCOLS**
  - Asynchronous systems send data bytes between the sender and receiver. Each data byte is preceded with a start bit, and suffixed with a stop bit. These extra bits serve to synchronize the receiver with the sender.
  - Transmission of these extra bits (2 per byte) reduce data throughput. Synchronization is achieved for each character only. When the sender has no data to transmit, the line is idle and the sender and receiver are NOT in synchronization. Asynchronous protocols are suited for low speed data communications.
- **SYNCHRONOUS PROTOCOLS**
  - Synchronous protocols involve sending timing information along with the data bytes, so that the receiver can remain in synchronization with the sender. When the sender has no data to transmit, the sender transmits idle flags (a sequence of alternating 0's and 1's) to maintain sender/receiver synchronization. Data bytes are packaged into chunks called packets, with address fields being added at the front (header) and checksums at the rear of the packet.