Interfacing Data Transfer Models

Course Teacher:

Md. Obaidur Rahman, Ph.D.
Professor
Department of Computer Science and Engineering (CSE),
Dhaka University of Engineering & Technology (DUET), Gazipur.

Course ID: CSE - 4619
Course Title: Peripherals, Interfacing and Embedded Systems
Department of Computer Science and Engineering (CSE),
Islamic University of Technology (IUT), Gazipur.

Lecture References:

- **Book:**
  - *Microprocessor Architecture, Programming and Applications with 8085 (Part-III)*, **Author:** Ramesh Gaonkor

- **Lecture Materials:**
  - *Ramesh Gaonkar*

Peripherals (I/Os) and Interfacing

- The primary functions of the microprocessor/microcontroller are:
  - to accept data from input devices such as – keyboards and A/D converters
  - read instructions from the memory
  - process data according to the instructions and
  - send the results to output devices such as – LEDs, printers and monitors.
- The input and output devices are called either **peripherals or I/O**.
- Designing logic circuits (hardware) and writing instructions (software) to enable microprocessor to communicate with I/Os is called **interfacing** and logic circuits are called **I/O ports** or **interfacing devices**.

Models of Data Transfer

- Transmit or receive data occurs either in
- **Parallel Mode:**
  - The entire data (i.e., 8-bit or16-bit) is transferred at one time
  - In 8085, an 8-bit data is transferred simultaneously over the 8-data lines (i.e., data bus)
  - Connected via direct cable that has **one wire** for each bit in a character of data code being used by the terminal
  - With multiple wires, all the bits of a characters can be transmitted between the terminals and computer at once
  - **Disadv:** Very expensive & no practice over long distance
  - Devices use parallel mode: *Keyboards*, *7-segment Display, Data Converters and Memory*.

Models of Data Transfer

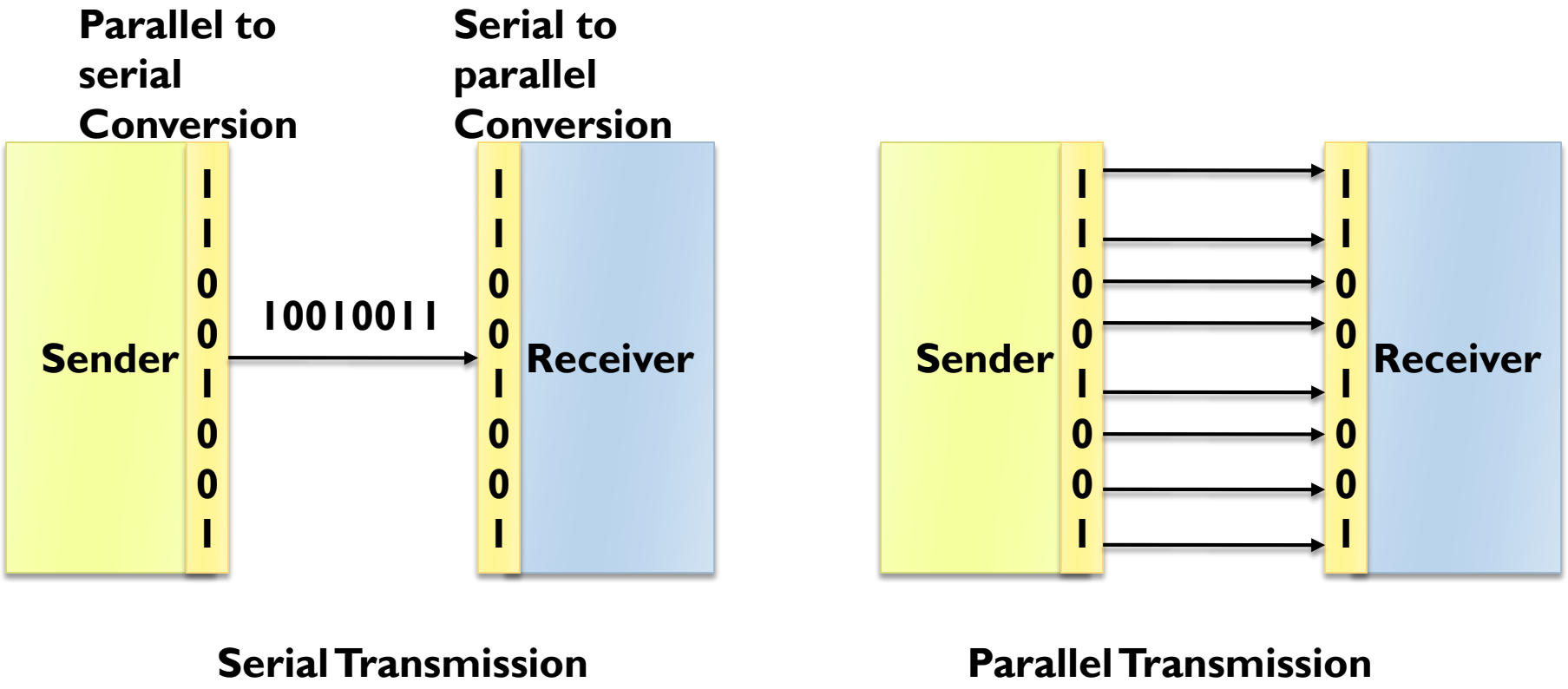
- **Serial Mode:**
  - Bits of each character are sent down to a line **one by one**
  - Complicated process because machine needs to know how to **decompose** and to **reconstruct** of bits at each respective end
  - Data are transferred one bit at a time over a single line between the MP and a peripheral.
  - A data is converted to a stream of bits
  - Devices use serial mode: *CRT, Printers, USB, Modems, Telephone lines*.

Models of Data Transfer

- **Serial Mode:**
  - RS232- Typical computer COM port
  - SCI- Serial Communication interface, uses the universal asynchronous receiver/transmitter (USART) or UART
  - SPI Serial peripheral interface

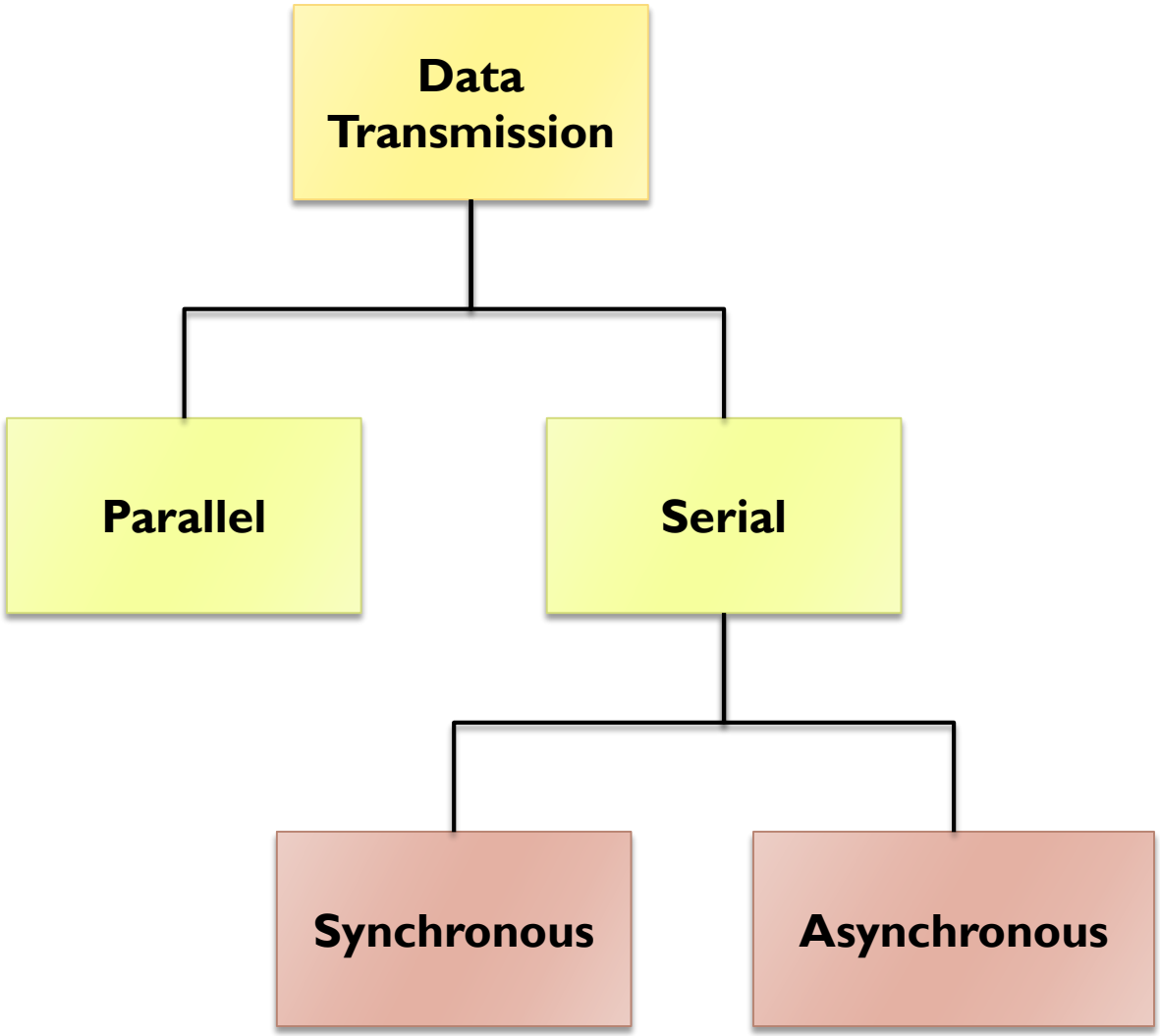
Models of Data Transfer

Two basic modes of data transmission



Models of Data Transfer

- **Serial**
  - Cheaper
  - Slower
- **Parallel**
  - Faster
  - Limited to small distances
  - Simultaneous transmission
  - Requires separate data lines
  - Bits must stay synchronized
  - Expensive

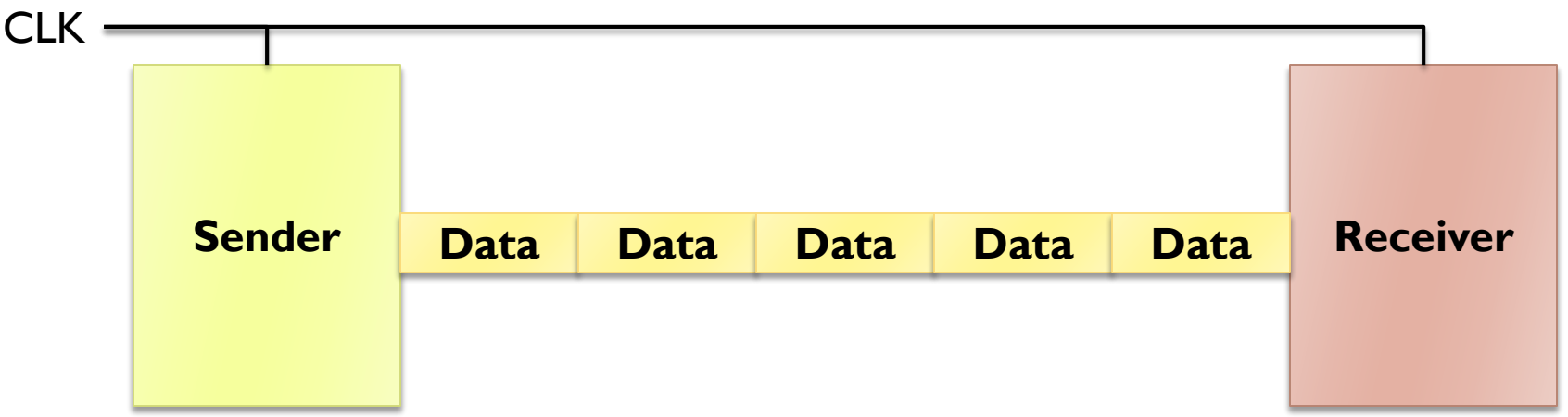


Type of Serial Communication

- Microprocessor/microcontroller communicates with peripherals in either of two transmission formats:
- **Synchronous Transmission:**
  - Synchronous means at the same time
  - The transmit and receive are synchronized with same clock
  - Block of data can be sent
  - It is used for high-speed data transmission

Type of Serial Communication

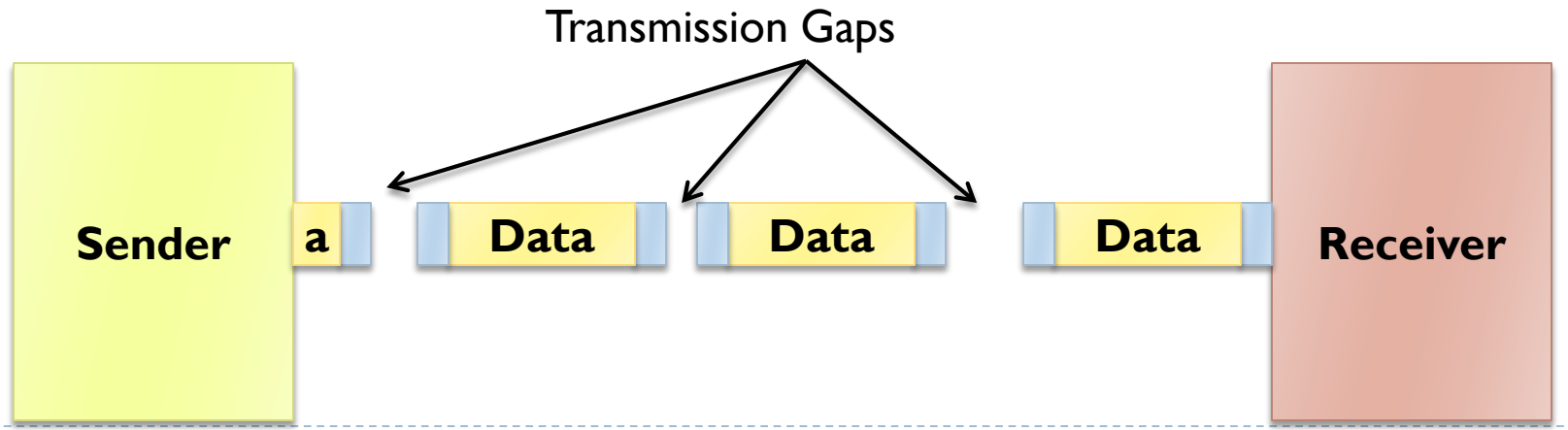
- **Synchronous Transmission:**
  - Data rates are dependent on clock rates
  - Continuously transmitting characters to remain in sync.
  - More efficient : Less overhead than asynchronous transmission



Synchronous transmission

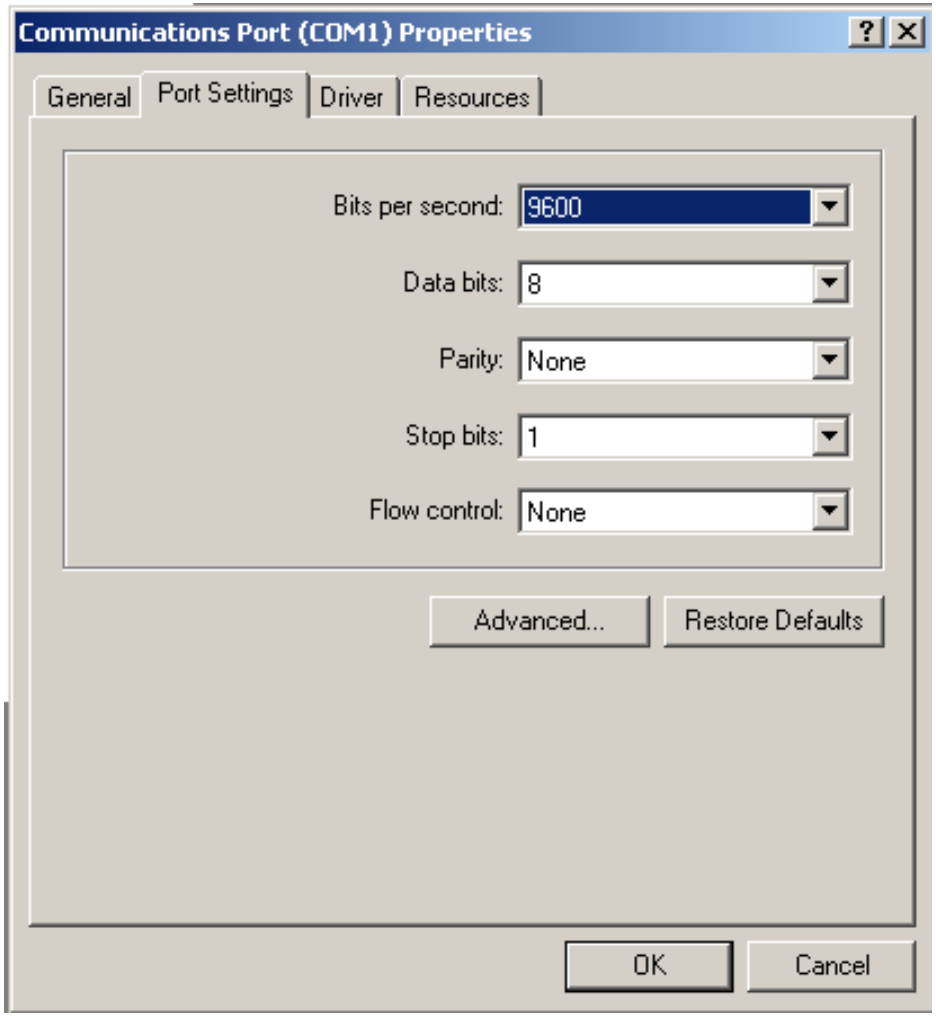
Type of Serial Communication

- **Asynchronous Transmission:**
  - Asynchronous means irregular intervals
  - Each byte is encoded for transmission
    - Start and stop bits
  - No need for sender and receiver synchronization
  - It is used for low-speed data transmission
- Data transfer between microprocessor and peripherals are primarily ***asynchronous***.



Asynchronous Serial Transmission

- Bits are transmitted in a specified format
- Defined by settings on transmitter and receiver:
  - Start Bit
  - Data Bits
  - Parity Bits
  - Stop Bits

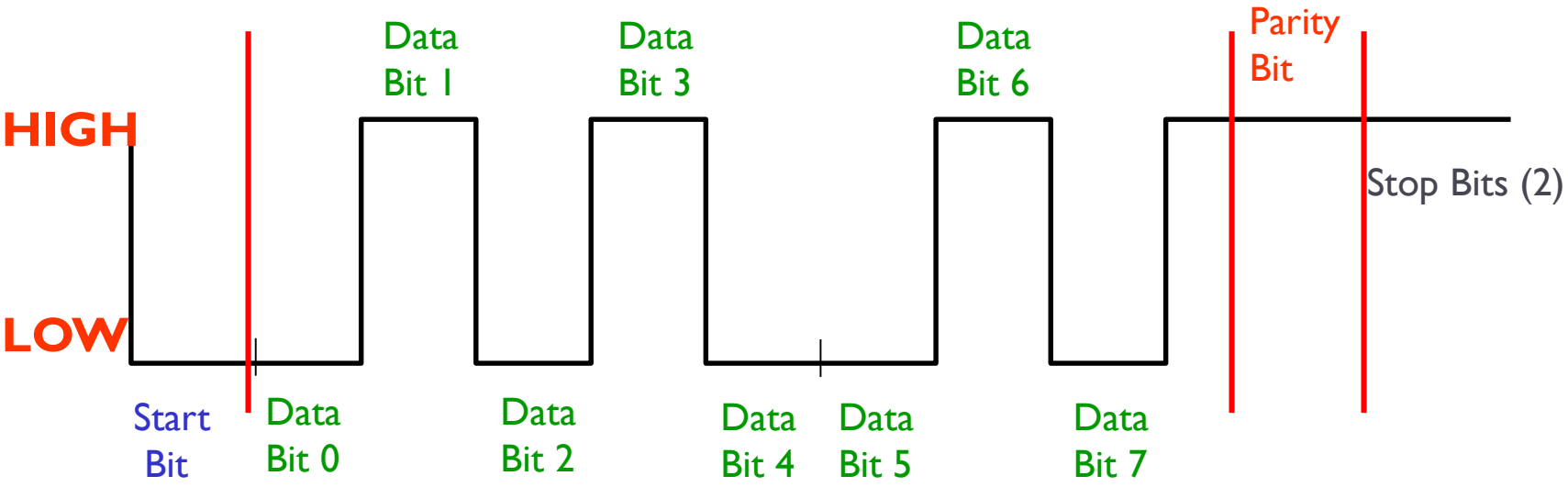


Example of Windows setting

Asynchronous Serial Transmission

One Data Package

- Four parts per package



Performance Comparison

- **Example Scenario**: Consider a character consists of
  01001011
- **Asynch**: 250 char x (8 data + 2 start/stop) = 2500 bits
- Assuming 1 ***Synch character*** after each 50 Character Transmission **Synch**: (250 + 5 synch char) x 8 bits) = 2040 bits
- Thus, for the scenario Synch is approximately 20% more efficient than Aysnch
- **Note**: Mostly, host computers adopt Synch transmission.

Conditions for Data Transfer

- **Microprocessor Controlled Data Transfer**
  - ***Unconditional Data Transfer***
    - MP thinks of that a peripheral is always available
  - ***Data Transfer with Polling***
    - Polling a peripheral or set of peripherals regularly
  - ***Data Transfer with Interrupt***
    - A peripheral is ready to transfer data and sends an INT signal
    - MP stops current execution and send the peripheral an INTA signal
  - ***Data Transfer with READY Signal***
    - If peripherals response time is slower than the execution time of MP then T states are extended to complete data transfer
  - ***Data Transfer with Handshake Signals***
    - Signal are exchanged between MP and peripheral prior to data transfer

Conditions for Data Transfer

- **Peripheral Controlled Data Transfer**
  - This type is applied when a peripheral is much faster than the MP
  - In case of Direct Memory Access (DMA), the DMA controller sends HOLD signal to the MP and MP releases its Data and Address Bus to the DMA controller
  - Thereupon, data transferred at high-speed without the intervention of MP

Thank You !!