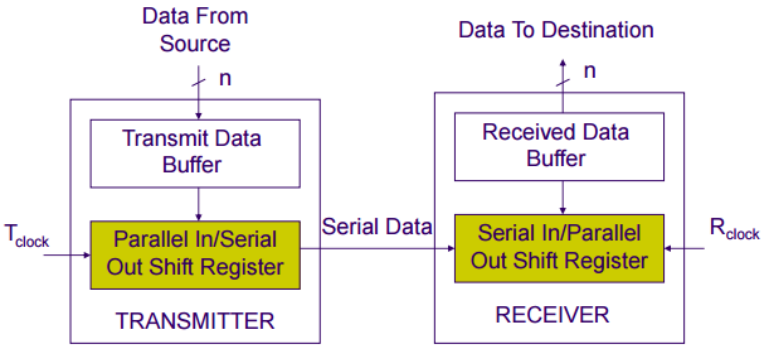
8. Serial I/O

COMP2121 • KC Notes

8.1 Introduction to Serial I/O

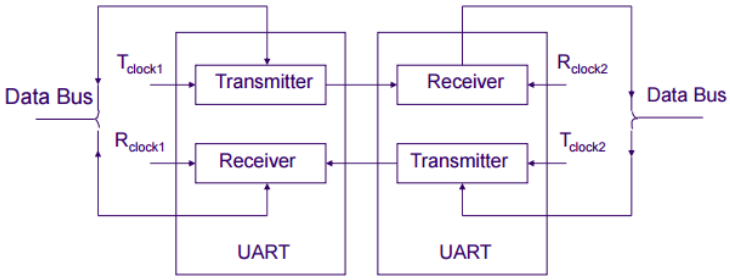
- Disadvantage of parallel I/O is that **a wire is needed for each bit** – cable is therefore bulky, expensive, and signals are susceptible to reflections and induced noises for long distances
    - **Serial I/O sends 1 bit at a time** – reduce costs, these effects



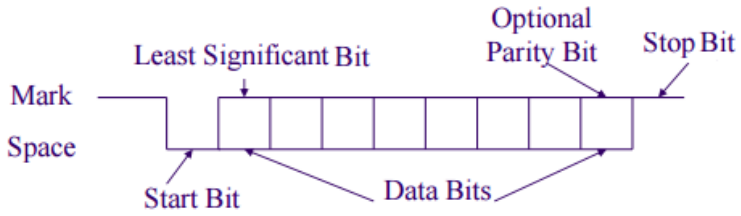
- At the **transmitter**/communication **source**:
    - Parallel interface **transfers data from source to transmit data buffer**
    - Data is loaded into **Parallel In/Serial Out (PISO)** Shift Register and **T_{clock} shifts the data bits out** to the receiver
- At the **receiver**/communication **destination**:
    - **R_{clock} shifts the data bits into the Serial In/Parallel Out (SIPO)** Shift Register
    - Transferred from the register into the **received data buffer** and read via the parallel interface

8.2 Serial Communication

- **Synchronous**: transmitter and receiver use synchronised clocks – faster transfer rate
- **Universal Asynchronous Reciever/Transmitter (UART)**: a device that implements both transmitter and receiver in a **single integrated circuit**
    - **Least significant bit** of data is transferred first
    - Data is transmitted asynchronously – **clocks are not synchronised**



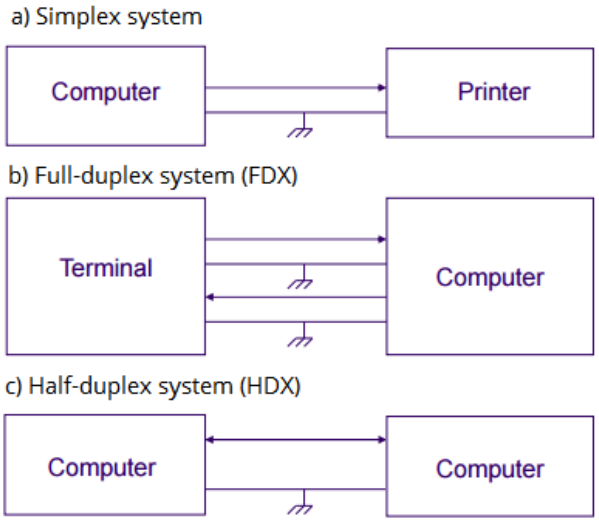
8.3 Data Encoding and Transmission



- Data is encoded in any method, e.g. ASCII
  - Encapsulated with two bits: **start bit, stop bit**
    - Start bit: line from mark to space for one bit-time – **synchronises receiver with transmitter** – tells the receiver to start clocking
    - Data bits: data, e.g. a character
    - Parity bit: used to detect errors in the data – the bit tells if there are an odd or even number of bits
    - Stop bit: Gives one bit-time between successive data
  - **Mark and space**: mark (logic one) and space (logic zero), when transmitter is not sending anything it **holds the line at mark level**
  - Data sent asynchronously, specified by a data rate in bits per second (baud rate)

8.4 Serial I/O System Types

- Three ways data can be sent in serial communication system:
- **Simplex system**: **data sent in one direction only**, e.g. from computer to printer
  - If computer sends data faster than printer can accept, **handshaking signals** required
  - Two signal wires needed
- **Full-Duplex (FDX) system**: data **sent in two directions**
  - Four wire system, two signal wires and a common ground
- **Half-Duplex (HDX) system**: data **sent in two directions** with one pair of signal
  - Needs additional hardware and handshaking signals



8.4 Serial I/O Devices

- **Data Communications Equipment (DCE)**: path for communication
    - E.g. two modems communicating via a phone line
- **Data Terminal Equipment (DTE)**: connects to a communication line
    - E.g. A PC and an internet provider
- **Modem**: modulator/demodulator, converts **logical levels into tones** sent over a telephone line
    - At the end of a telephone line, demodulator converts back to logic levels

8.5 Serial I/O Interface Standards

- Interface standards allow equipment to be interconnected
    - Standards include RS-232-C, RS-422, RS-423, RS-485
        - **RS-232-C used in most serial interfaces**
        - RS-422, 423, 485 used for transmitting signals farther or greater
- Various pins used for different purposes
    - **Ring indicator (RI)**: telecommunication transmits a special tone that rings phone
    - **Data set ready (DSR)**: DCE has made a connection on line and is ready to **receive data from the terminal** – DTE must see this asserted before transmitting data
    - **Data terminal ready (DTR)**: DTE is ready to send/receive data
    - **Data carrier detect (DCD)**: asserted when DCE detects the carrier on the line
- When two serial ports are connected, data rate, number of data bits, parity, number of stop bits must be **set properly and be identical on each UART**
- There are **several types of cable** depending on types of devices
    - Full DTE-DCE cable
    - Minimal DTE-DCE cable
    - DTE-DTE null modem cable
    - Minimal null modem cable

8.5.1 STANDARDS CONNECTIONS

- Logic levels for **RS-232-C** are **-25 to -3V** for mark, **+25 to +3V** for space
- **RS-423** is allows longer distance and higher data rates, and allows a driver to broadcast data to up to 10 receivers
- **RS-422** use a **differential amplifier** – eliminates noise with long transmission lines
- **RS-485**: Also uses differential drivers and receivers, allows up to 32 drivers and receivers

8.6 AVR USARTS

- AVR has two USART units that can be **configured for synchronous or asynchronous serial comms**. USART unit 0 is used to receive code from PC via USB
    - Has transmission error detection (odd/even parity error, framing error)
    - Has three interrupts: TX Complete, TX Data Register Empty, RX Complete
- Consists of:
    - **Clock generator**: synchronisation logic for external clock input
    - **Transmitter**: a single write buffer, serial Shift Register, Parity Generator, Control Logic for different frame formats
    - **Receiver**: includes a receive buffer (UDR), serial Shift Register, Parity Checker, Control Logic
        - Also has recovery units for async data
- **Frame Formats**: up to 30 formats that data can be sent in.
    - 1 start bit (St)
    - 5, 6, 7, 8, 9 data bits
    - Even, odd or no parity bit (P)
    - 1, 2 stop bits (Sp)
- **Control registers:**
    - **UCSRA**: status flags of USART, controls speed and use of multiple processors
    - **UCSRB**: enable interrupts, transmission operations, frame formats, bit extension
    - **UCSRC**: operation configuration

8.6.1 INITIALISATION, TRANSMISSION AND RECEPTION PROCESS

- **Initialisation** process: sets baud rate, frame format and enables transmitter/receiver
- **Transmission**: **TXEN** Transmit Enable in UCSRB
    - Load transmit buffer with data to be transmitted
    - CPU writes to the UDR I/O location, which moves to the Shift Register when ready
    - Status determined by:
        - **UDRE** USART **Data Register Empty** (when transmit buffer is empty)
        - **TXC Transmit Complete** when transmit shift register is all shifted out
- **Reception**: **RXEN** Receive Enable in UCSRB
    - **RXC Receive Complete** indicates whether there is any unread data in the receive buffer
    - **RXEN = 0**: receive buffer is flushed

8.6.2 ERROR DETECTION

- **Frame** error: check when first stop bit is correctly received
- **Parity** error: check whether data has odd/even number of 1's
- **Data OverRun** error: check if any data is yet read but is overwritten by another data frame
- AVR has **clock recovery, data recovery unit** when asynchronous