Ministry of Higher Education and Scientific Researches
Al-Mansour University College
Department of Computer Technology
3rd class

Serial Interfacing

Instructor: Noor Kareem

Serial Interface

Introduction

A serial interface is a device that enables the serial (one bit at a time) exchange of data between a digital device like microprocessor and external devices (peripherals) such as printers, drives, scanners, camera or another pc.

We can send the bits one at a time over a single data line. Because the bits are now sent in series, we call this serial interfacing.
The transmits data as a single stream of bits, typically using a wire-plus-ground cable, a single wireless channel or a wire-pair usually never more than four wire.
The serial interface acts as a communication interface between two digital systems that exchange the data as a series of voltage pulses over a wire.
Some devices that use the serial interface include the Universal Serial Bus (USB) Standard No. RS – 232 ,RS – 422 ,RS – 423 ,RS – 485 and I2C.

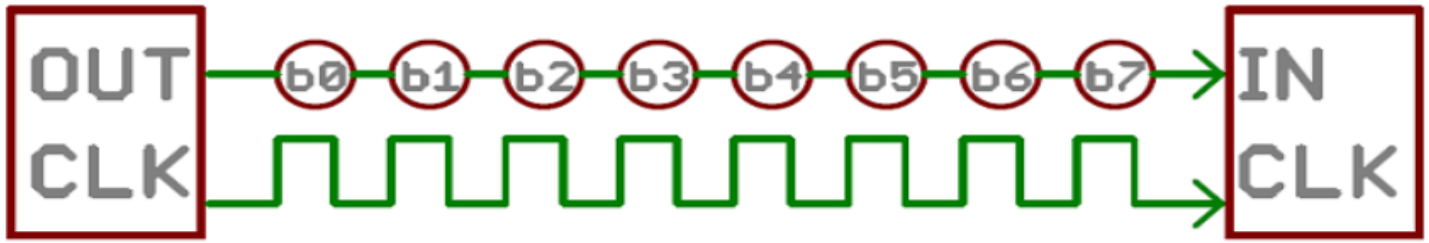
types of serial interface:

There are two types of serial interface:

1) Synchronous serial interfaces (SPI)
2) Asynchronous serial interfaces (SCI).

Synchronous serial interfaces (SPI)

- A ***synchronous serial interface*** always pairs its data line(s) with a clock signal, so all devices on a synchronous serial bus share a common clock.
- This makes for a more straightforward, often faster serial transfer, but it also requires at least one extra wire between communicating devices.
- In this type the data rates depend on clock rates.
- Constant transmission of data.
- Clocks of Transmitter and Receiver must be synchronized.
- Flexible to communication with peripheral devices LCD drivers, A/D converter, other microprocessors.
- Simultaneously transmits and receives data Transmission line, Receiving line, and Ground.
- In this type of interfacing the data is sent from the Least Significant Bit (LSB = smallest binary bit value)

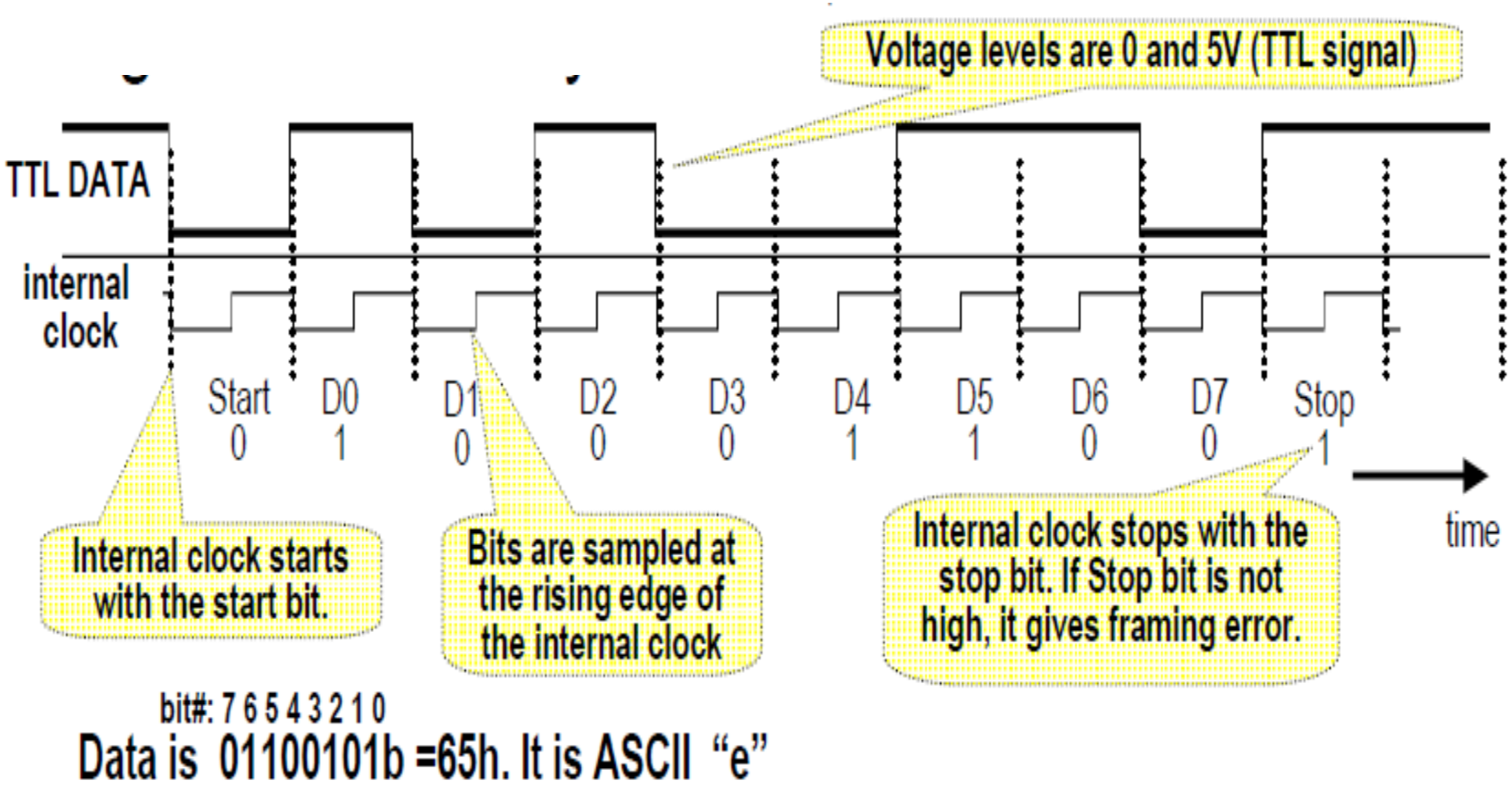


Asynchronous serial interfaces (SCI)

Asynchronous means that data is transferred without support from an external clock signal. This transmission method is perfect for minimizing the required wires and I/O pins, but it does mean we need to put some extra effort (serial protocol) to synchronous to transferring and receiving data.

- Transmission of data through “words”
- Continuous transmission unnecessary
- Built-in safeguards against noise and error
- Transmitter and Receiver operate independently
- Requires start and stop bit for each byte of data
    - Sends constant ‘1’ for idle
    - Sends a ‘0’ for start and “1” for stop bits

- Start bit is required to synchronize the internal clock of receiver.
- Stop bit is required to test the clock frequency.
- Only DATA is transmitted, internal clock is generated locally.



Framing the data

Each block (usually a byte) of data transmitted is actually sent in a packet or frame of bits. Frames are created by appending synchronization and parity bits to our data.

Data chunk

The important part of every serial packet is the data it carries. We ambiguously call this block of data a chunk, because its size isn't specifically stated. The amount of data in each packet can be set to anything from 5 to 9 bits. Certainly, the standard data size is your basic 8-bit byte, but other sizes have their uses. A 7-bit data chunk can be more efficient than 8, especially if you're just transferring 7-bit ASCII characters.

Details Frame Bits

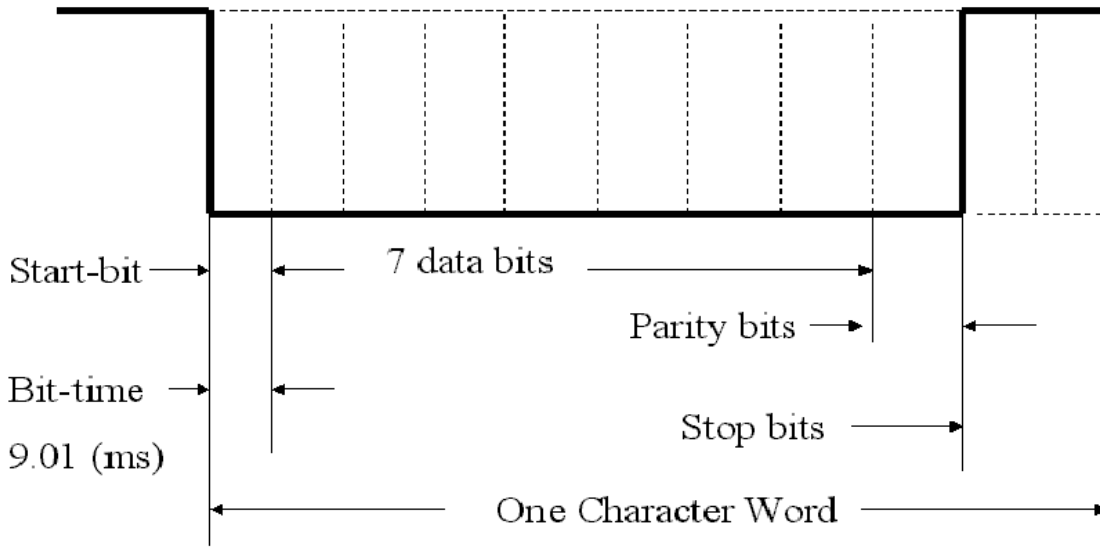
The frame consist of the following:

Start bit (1 – bit).

Data bit (5 – 9 bit).

Parity bit (0 – 1 bit)

Stop bit (1 – 2 bit).



Start Bit –

- Signals the transmission of a word.
- Transition from “1” to “0”. (“Mark-to-space”)
- First bit to be transmitted.

Stop Bits –

- Bit at the end of a data word to indicates the end of a word. .
- Bit set to high “1”.

Data bits –

- Data bits to be transmitted.
- Sender and receiver have to agree in the number of data bits. (Usually 8 or 9)
- Least significant bit is sent first can be low or high.

Parity bit –

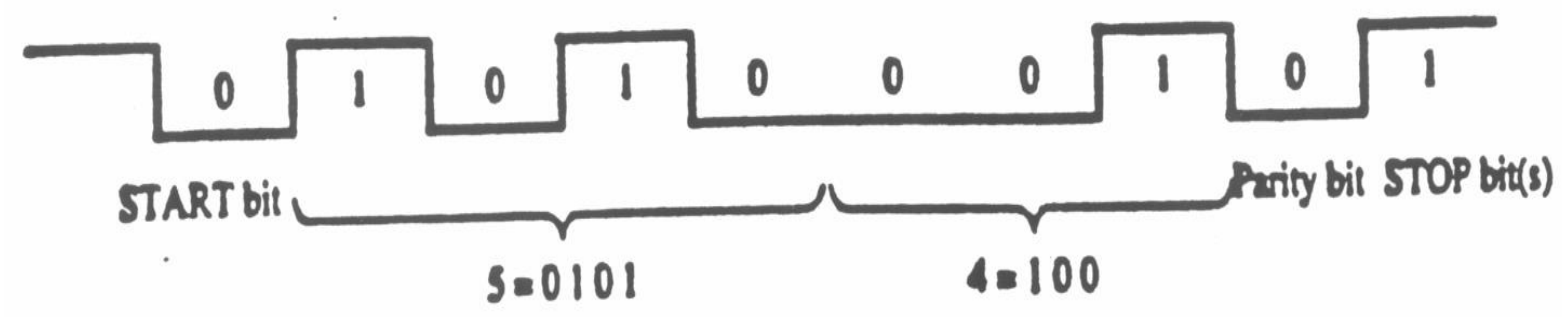
- Works as an error check.
- There are two types: odd and even
- Even: if number of 1’s in the data word is even. Parity bit set to one.
- Odd: if number of 1’s in the data word is odd. Parity bit set to zero.
- Bit after the data bits and before the stop bit.
- Can prevent noise signal, but does not recognize when two bits are altered by noise.

Example:

Sketch the output when you suppose the number “45” in hex is to be transmitted in an 8-bit asynchronous format with odd parity.

Solution:

- The number “45 in hex” equal to (01000101) in binary.
- Keep in mind that the data is sent least significant bit (LSB) first, the result is displayed below.
- Since there are already three 1’s in the data word, the parity bit would be set to zero to mean the parity odd.

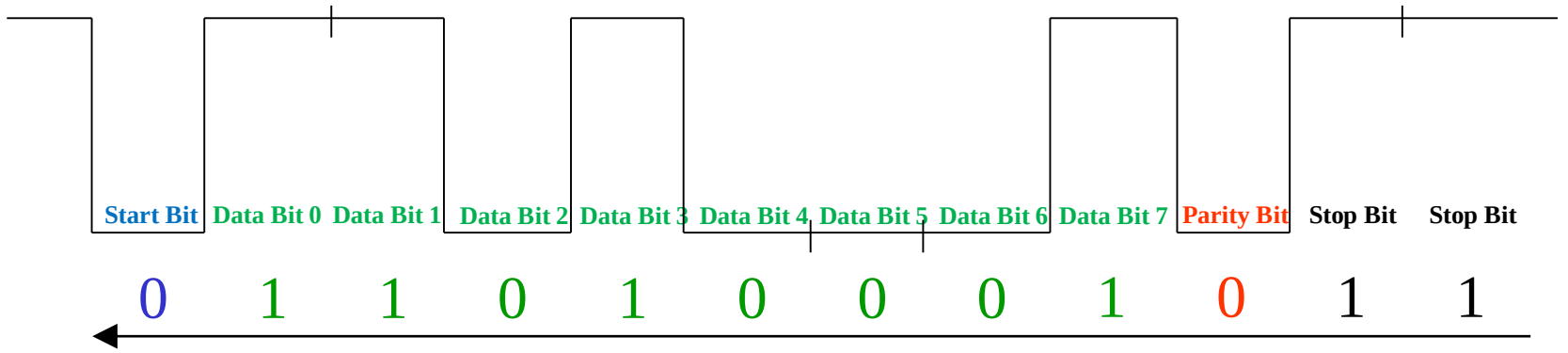


Example

Draw the serial output Send $8B_{16}$ with one start bit, 8 data bits, even parity, and two stop bits.

Solution:

8B hex = 10001011 binary



Baud Rate

- Baud rate: number of total information bits transmitted per second (includes start, data, parity and stop bits).
- The baud rate specifies how fast data is sent over a serial line. It's usually expressed in units of bits per second (bps) ***Baud = bit/second .***
- Baud rates can be just about any value within reason. The only requirement is that both devices operate at the same baud rate (transmitter and receiver) . One of the more common baud rates is 9600 bps. Other "standard" baud are 1200, 2400, 4800, 19200, 38400, 57600, and 115200.
- If each bit takes T seconds the baud rate is B=1/T.
- Baud rate tolerance for a 10 – bit frame is 5%.

Bit rate:

The number of data bits only transmitted per second so for us,

Baud rate > Bit rate

Baud Rate Example

What is the bit rate for a 2400 baud rate using one start bit, one parity bit and two stop bits per 8-bit /data word?

Solution:

$$\text{bit rate} = \text{baud rate} \frac{8 \text{ data bits/word}}{12 \text{ total bits/word}}$$

$$\text{bit rate} = 2400(2/3) = 1600 \text{ bps}$$

Example:

What is the If a system send a packet of 50 bytes at 1200 Baud, using 8-bit data, no party, one stop bits, what is the transmission time of the whole packet

Solution:

1 – byte frame is 1- start + 8 – data + 1 – stop bit = 10 bits/byte.

packet is transferred by 50 x 10 bits= 500 bits.

T_{packet} = 500 bits / 1200 Baud = 0.417 sec. = 417 ms.