Computer Organization and Architecture

Module 7

Input-Output Systems

Prof. Indranil Sengupta

Dr. Sarani Bhattacharya

Department of Computer Science and Engineering

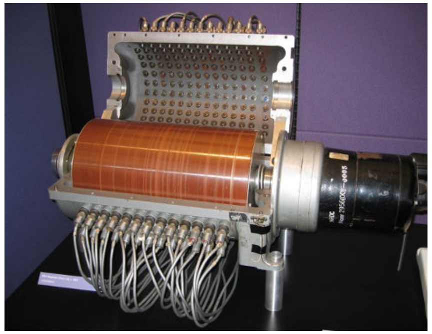
IIT Kharagpur

Secondary Storage Devices

2

Magnetic Disk (Hard Disk)

- Magnetic disks constitute a traditional method for non-volatile storage of information using magnetic technology.
- Broadly three types of devices appeared:
  1) *Floppy disk* : made of bendable plastic
  2) *Magnetic drum* : made of solid metal
  3) *Hard disk* : made of metal or glass
- All of these rely on a rotating platter (metal or glass or plastic) coated with a thin magnetic material, and use a moveable read/write head to read and write data from / to the disk.
  - Data stored as tiny magnets.

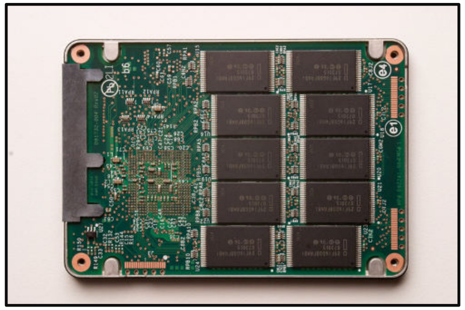


Magnetic drum
(62.5 Kbytes)

8” floppy disk
(360 Kbytes)

3.5” floppy disk
(1.2 Mbytes)

3.5” magnetic disk
(1 Tbytes)

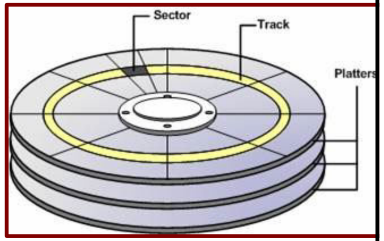


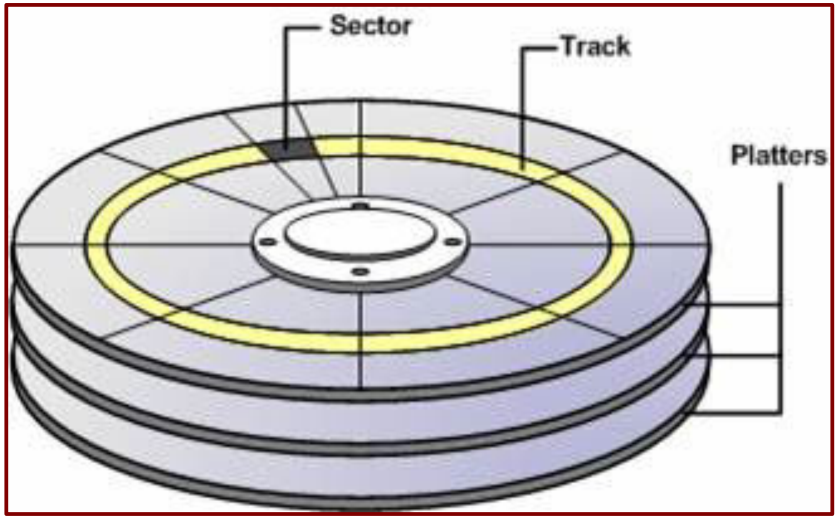
1.8” solid-state disk
(512 Gbytes)

- Since the platters in a hard disk are made of rigid metal or glass, they provide several advantages over floppy disks:
  - They can be larger.
  - Can have higher density since they can be controlled more precisely.
  - Has a higher data rate because it spins faster.
  - No physical contact with read/write head as it spins faster.
    - The read/write head floats on a cushion of air (few microns separation).
    - Requires dustless environment.
    - Results in higher reliability.
  - More than one platters can be incorporated in the same unit.

Organization of Data on a Hard Disk

- The hard disks consists of a collection of *platters* (typically, 1 to 5), which are connected together and can spin in unison.
  - Each platter has two recording surfaces, and comes in various sizes (1 – 8 inches).
  - The stack of platter typically rotates at a speed of 5,400 to 10,000 rpm.
  - Each disk surface is divided into concentric circles called *tracks*.
    - The number of tracks per surface can vary from 1000 to 5000.
  - Each track is divided into a number of *sectors* (64 – 200 sectors/track).
    - Typical sector size: 512 – 2048 bytes.
    - Sector is the smallest unit that can be read or written.
  - The disk heads for all the surfaces are connected and move together.
  - All the tracks under the heads at a given time on all surfaces is called a *cylinder*.




Sector
Track
Platters

Disk Access Time

- There are three components to the access time in hard disk:
  - **a) Seek time:**
    - The time required to move the head to the desired track.
    - Average seek times are in the range 8 – 20 msec.
    - Actual average can be 25 – 30% less than this number, since accesses to disks are often localized.

b) Rotational delay:

- Once the head is on the correct track, we must wait for the desired sector to rotate under the head.
- The average delay or latency is the time for half the rotation.
- Examples:
  - For 3600 rpm, average rotational delay = 0.5 rotation / 3600 rpm = 8.30 msec
  - For 5400 rpm, average rotational delay = 0.5 rotation / 5400 rpm = 5.53 msec
  - For 7200 rpm, average rotational delay = 0.5 rotation / 7200 rpm = 4.15 msec

c) **Transfer time**:

- The total time to transfer a block of data (typically, a sector).
- Transfer rates are typically 15 MB/sec or more.
- Transfer time depends on:
  - ➢ Sector size
  - ➢ Rotation speed of the disk
  - ➢ Recording density on the tracks

Example 1

- Consider a disk with sector size 512 bytes, 2000 tracks per surface, 64 sectors per track, three double-sided platters, and average seek time of 10 msec.
  a) What is the capacity of the disk?
  b) If the disk platters rotate at 7200 rpm, and one track of data can be transferred per revolution, what is the transfer rate?

Bytes/track = 512 x 64 = 32K

Bytes/surface = 32K x 2000 = 64,000K

Bytes/disk = 64,000K x 3 x 2 = 384,000K

Transfer rate = Capacity of a track / rotational delay

= 32K / 8.30 msec = 3,855 Kbytes/sec

Some Recent Advancements

- Most of the modern-day disk units include a *high-speed cache* directly in the disk unit.
  - Allows fast access of data that was recently read between transfers requested by the CPU.
- In conventional disks, each track contains the same number of bits.
  - Outer tracks record data at a lower density than inner tracks (circumference of a circle is proportional to its radius).
  - An alternate scheme uses *constant bit density*, where the outer tracks store more bits than the inner tracks.

Solid State Drives

- Also referred to as flash drives.
- Very popular today as removable storage devices, and also as solid-state storage devices in computer system as a replacement of hard disk.
- Some features:
  - Non-volatile
  - Low power consumption
  - Faster than hard disk
  - Random access
  - Data typically written block-wise (erase followed by write)

Input / Output Organization

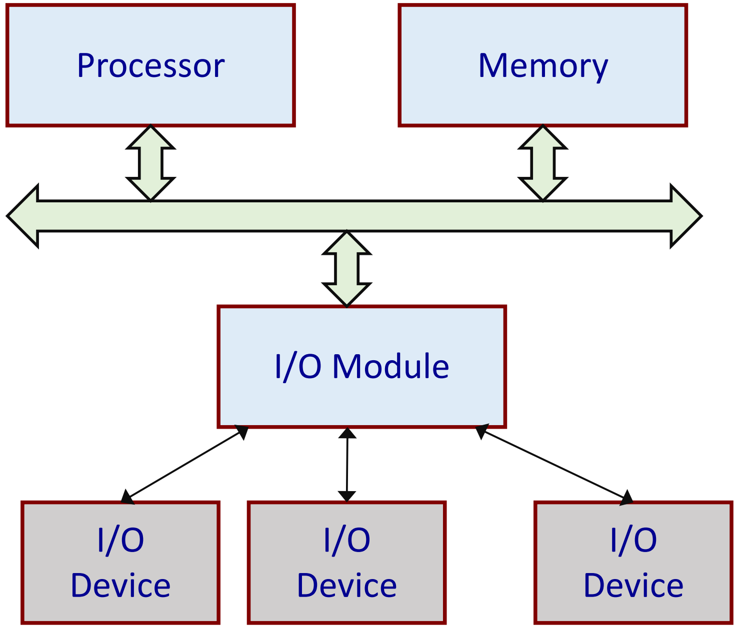
14

Introduction

- Interfacing input/output devices is more complex as compared to interfacing memory systems.
- Why?
  - Wide variety of peripherals (keyboard, mouse, disk, camera, printer, scanner, etc.).
  - Widely varying speeds.
  - Data transfer rate can be regular or irregular.
  - Sizes of data blocks transferred at a time varies widely (few bytes to Kbytes).
- Slower than processor and memory.

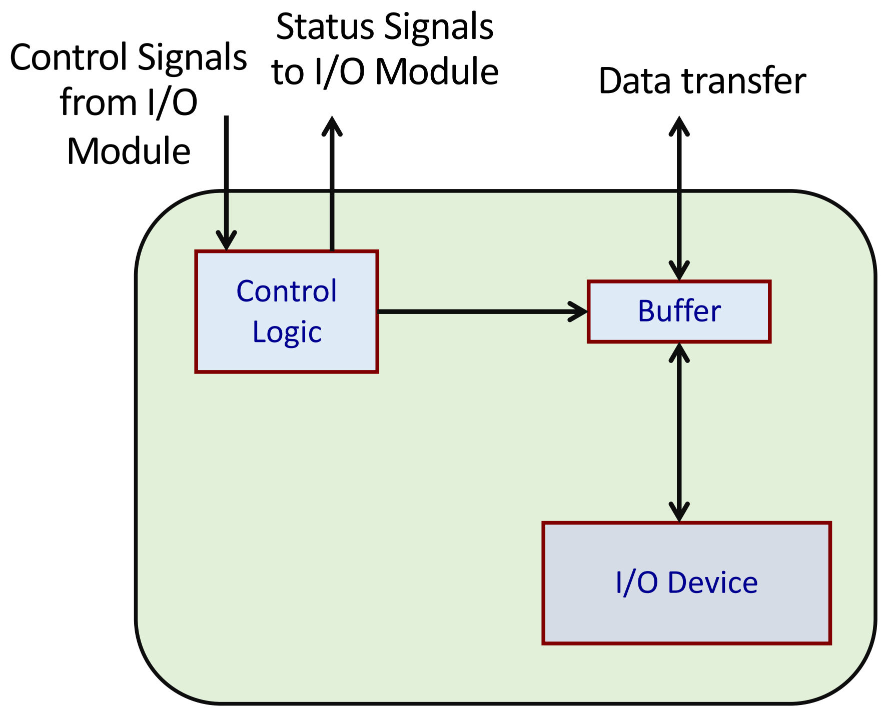
Input / Output Interface

- To handle widely different types of I/O devices, we need a programmable I/O interface or *I/O module*.
  - Interfaces to processor and memory on one side.
  - Interfaces to one or more peripheral devices on the other side.

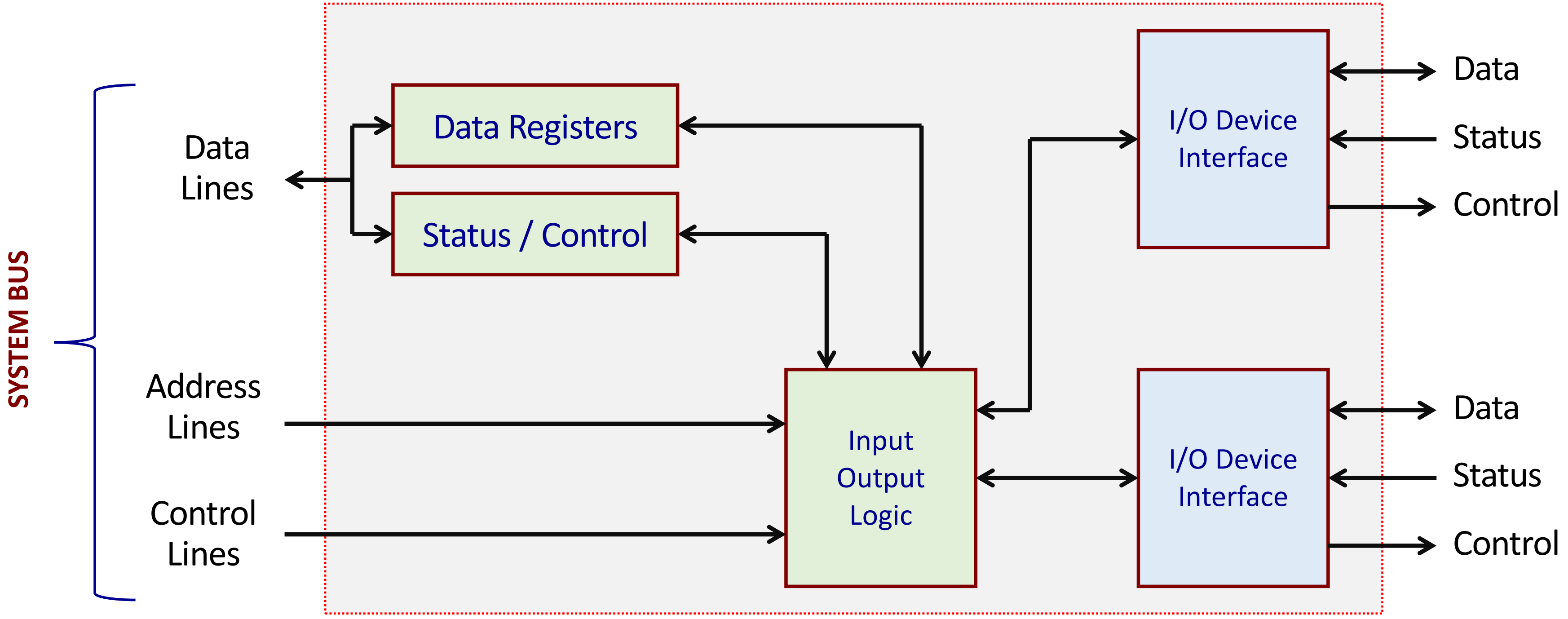


16

Typical I/O Device Interface



I/O Module Schematic



18

Typical Steps During I/O

a) Processor requests the I/O Module for device status.

b) I/O Module returns the status to the processor.

c) If the device is ready, processor requests data transfer.

d) I/O Module gets data from device (say, input device).

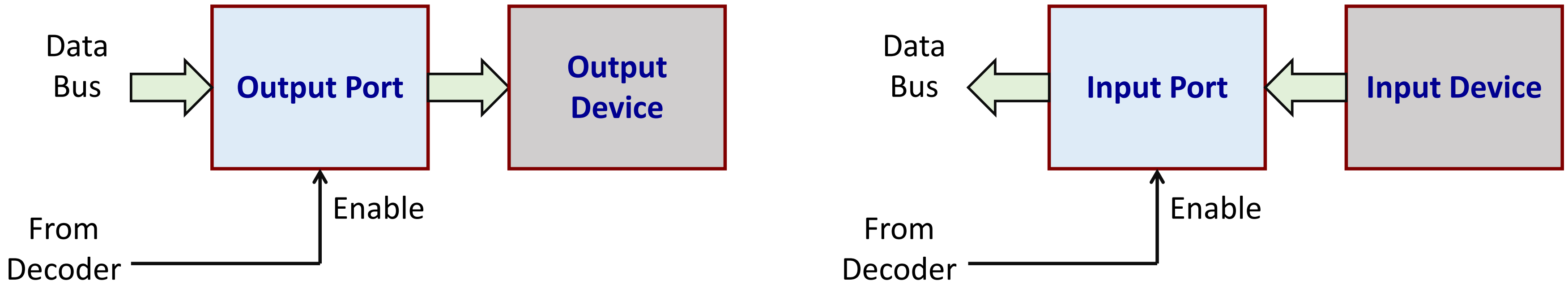
e) I/O Module transfers data to the processor.

f) Processor stores the data in memory.

How are I/O devices typically interfaced?

- Through input and output ports.
- *Output port*:
  - Basically a PIPO register that is enabled when a particular output device address is given.
  - The register inputs are connected to the data bus, and the register outputs are connected to the output device.
- *Input port*:
  - Basically a parallel tristate bus driver that is enabled when a particular Input device address is given.
  - The driver outputs are connected to the data bus, while the inputs are connected to the input device.

Example of output and input ports



Data Transfer Techniques

22

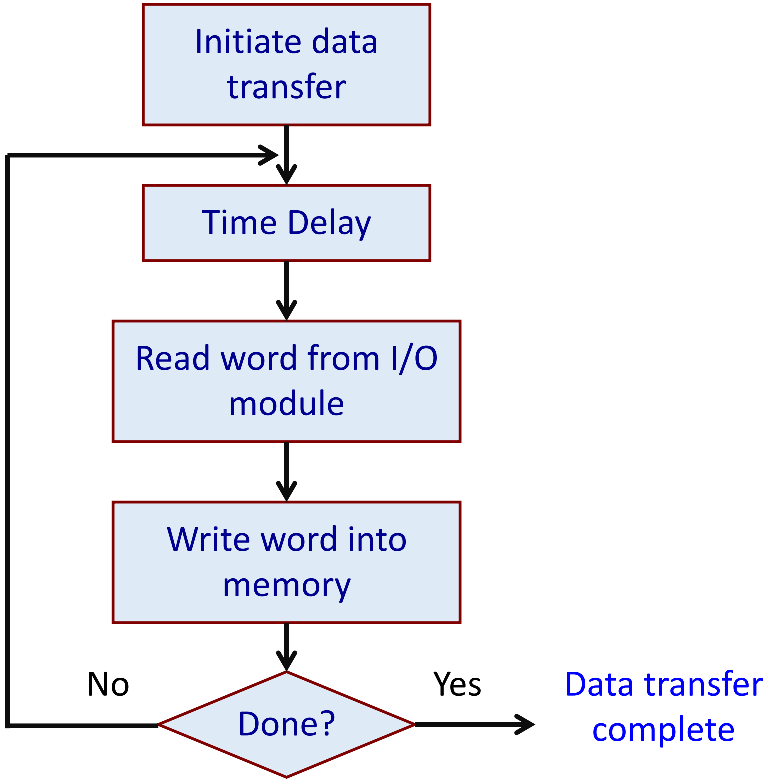
Data Transfer Techniques

1) **Programmed**: CPU executes a program that transfers data between I/O device and memory.
    a) Synchronous
    b) Asynchronous
    c) Interrupt-driven
2) **Direct Memory Access (DMA)**: An external controller directly transfers data between I/O device and memory without CPU intervention.

(a) Synchronous Data Transfer

- The I/O device transfers data at a *fixed rate* that is known to the CPU.
- The CPU initiates the I/O operation and transfers successive bytes/words after giving fixed time delays.
- Characteristics:
  - During the time delay, CPU lies idle.
  - Not many I/O devices have strictly synchronous data transfer characteristics.
- A flowchart for synchronous data transfer from an input device is shown on the next slide.

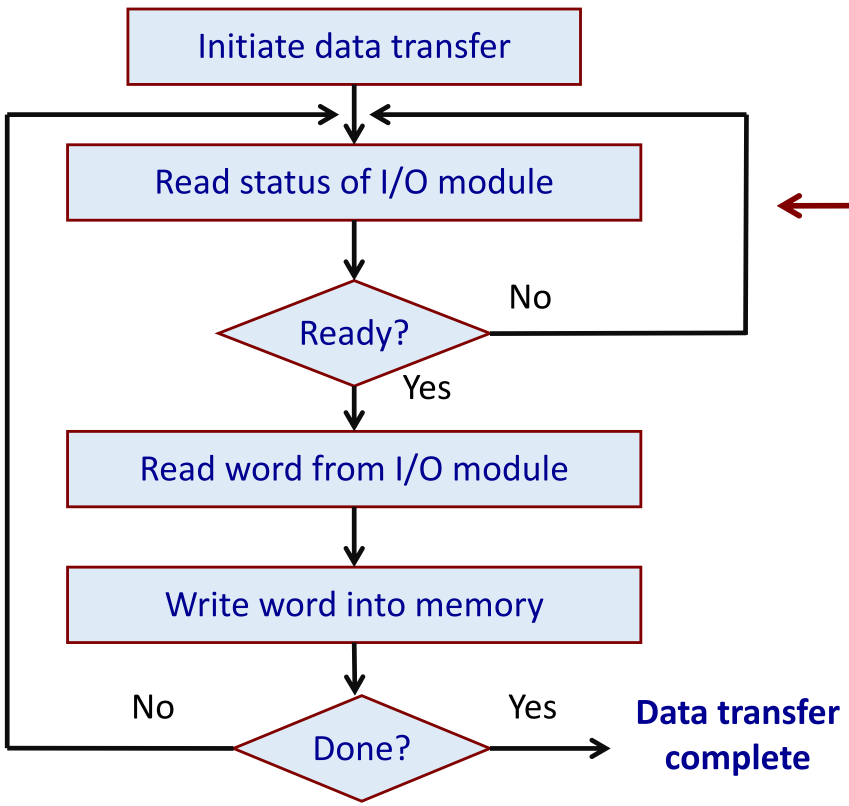
24



- Error may occur if the input device and the processor get out of synchronization.
- Large number of words cannot be transferred in one go.
- Speed of data transfer depends not only on the speed of I/O device and memory, but also on the execution time of the code.

(b) Asynchronous Data Transfer

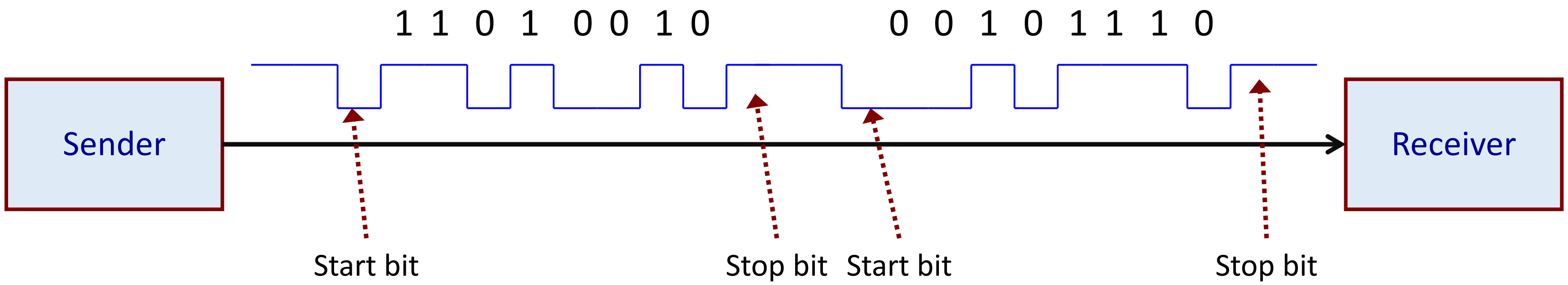
- The CPU does not know when the I/O module will be ready to transfer the next word.
- CPU has to check the status of the I/O module to know when the device is ready to transfer the next word.
  - Called *handshaking*.
- Characteristics:
  - While the CPU is checking whether the I/O module is ready, it cannot do anything else.
  - Wasteful of CPU time for slow devices like keyboard or mouse.



Lot of CPU time is wasted in this loop

- Cannot be used for high-speed devices.
- Speed of data transfer depends not only on the speed of I/O device and memory, but also on the execution time of the code.

- An example of asynchronous data transfer:
  - Serial data transfer between two devices using start and stop bits.
  - The devices are asynchronous at the level of bytes, but are synchronous at the level of bits within the bytes.



- Just like asynchronous data transfer, receiver waits for the next *START* bit, which indicates the beginning of a new byte transfer.
- After the *START* bit is received, receiver gives (known) bit delays and reads out the 8 bits of the byte.
- The *STOP* bits between the bytes serves to synchronize the data transmission.