Communication

EE5182 Microcontrollers and Embedded Systems

Introduction

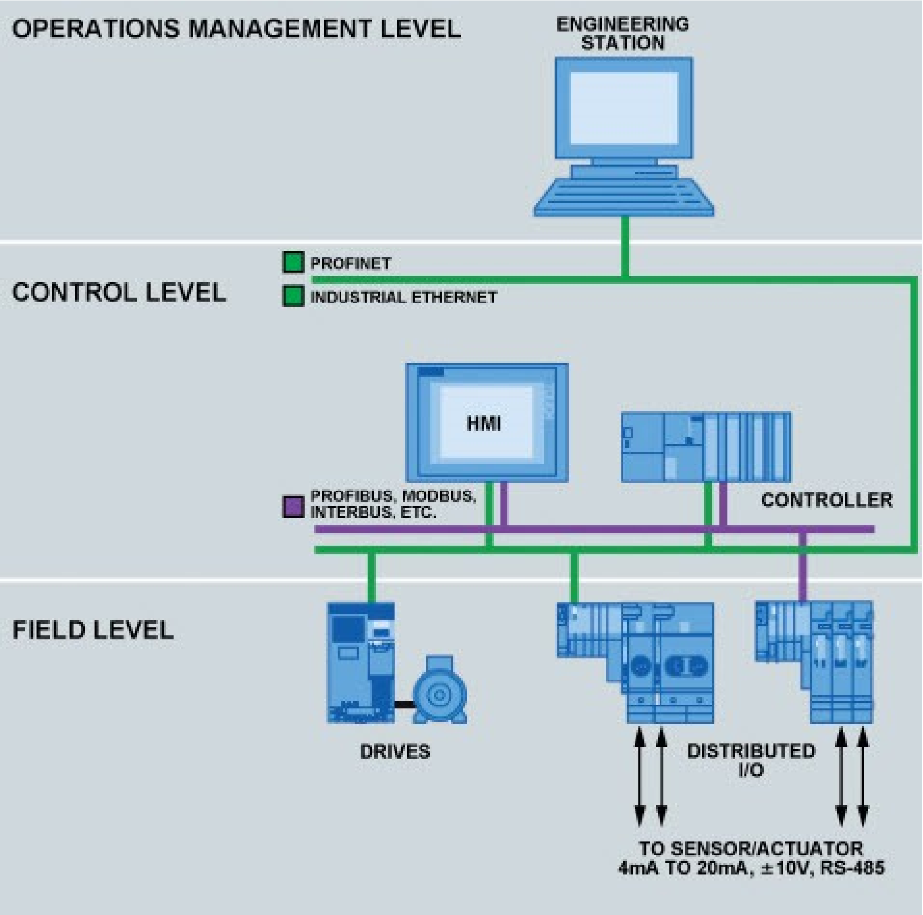
- Embedded & real-time systems could be standalone or connected
- A real-time system is often composed from a number of periodic (time triggered) and sporadic (event triggered) tasks which communicate their result by passing messages.
- In distributed real-time systems these messages are sometimes sent between processors across a communication device.

Introduction

- To guarantee that the timing requirements of all tasks are met, the communications delay between a sending task and a receiving task being able to access that message must be bounded.
- For examples
  - Control systems: between sensors and actuators via central computer
  - Multiprocessors: between processors, tasks communicating

Hierarchical levels of communication

- **Device Level:** This lowest level consists of field devices such as sensors and actuators of processes and machines.
- **Control Level:** This level consists of controllers, distributed control units, and computer systems. The tasks of this level include configuring automation devices, loading of program data and process variables data, supervising control, historical archiving, etc.
- **Information Level:** This is the top level of the industrial automation system which gathers the information from its lower level i.e., control level.

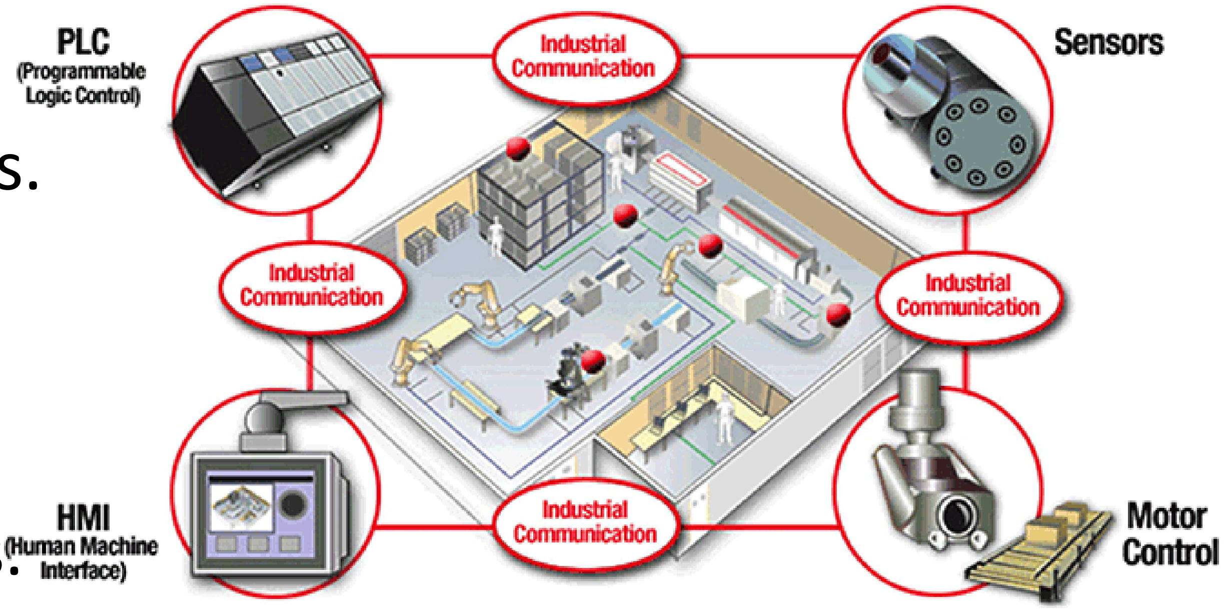


Data communication in control systems

- Control systems need data communication to communicate:
  - Between controllers and plants (controlled devices)
  - Between controllers and sensors
  - Between controllers and other related controllers
  - Between controllers and systems managers/monitors

Industrial communication networks

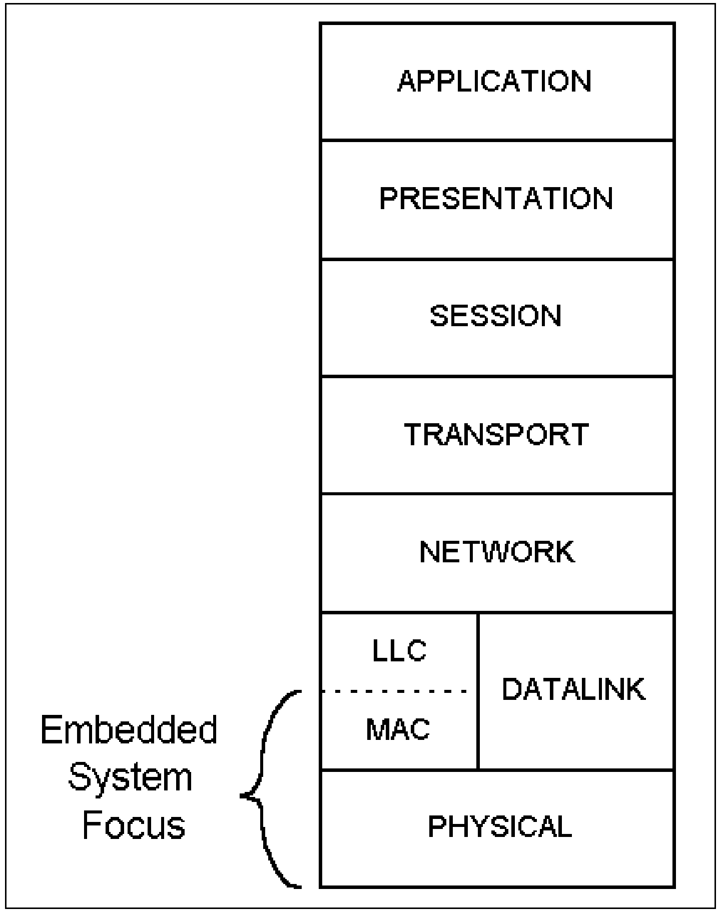
- Industrial control broadly defines the wide range of electronic equipment used in factories, process control plants, and automated facilities for monitoring and controlling manufacturing and other operations.
- It involves machines like robots, computers, machine tools, programmable logic controllers (PLCs), sensors, relays, valves, motors, and measuring instruments.



Industrial communication networks

- There exist many different communication networks designed to interconnect digital control systems to field devices and I/O modules.
- Some common and popular industrial communication standards are:
  - Serial Communication standards such as RS232, RS422 and RS485
  - Highway Addressable Remote Transducer (HART)
  - DeviceNet
  - ControlNet
  - Profibus
  - Foundation Field Bus
- https://www.youtube.com/watch?v=MNmzbMEzkdk

Open System Interconnection



- Intended for computers
- Designed to solve compatibility problem
- Layers provide standard interface and services
- Embedded systems use some standardisation ideas
- Higher layers require lower layers to work

Embedded systems communication concepts

- Point-to-point networks
  - Each node connected to every node
  - Simple and reliable
  - Dedicated links make it easy to meet real-time deadlines
  - Costly due to many wires required
- Shared media networks
  - Nodes are connected via bus or other topologies
  - Less wiring and hence cheaper
  - Easily extendable by adding new nodes to network
  - Complex network protocol

Embedded systems communication concepts

- Event based communication
  - E.g. alarm, user inputs, requests for data from other systems

- State based communication
  - E.g. regular sensor readings
  - Predictability

Event based system

- Efficient use of network resource
- Needs high reliability (event based data comes once in a while)
- May need acknowledgement
- Hard to predict delay in case of overloading (e.g. alarm)

State based system

- Messages sent at predefined, regular intervals.
- Less efficient due to regular occupation of communication channel by nodes.
- More tolerance. Missed message may be ok, since the next one will be coming.
- Transient data problem. Sending node has to keep data long enough for other to see. E.g. button pressed may need to be repeated.

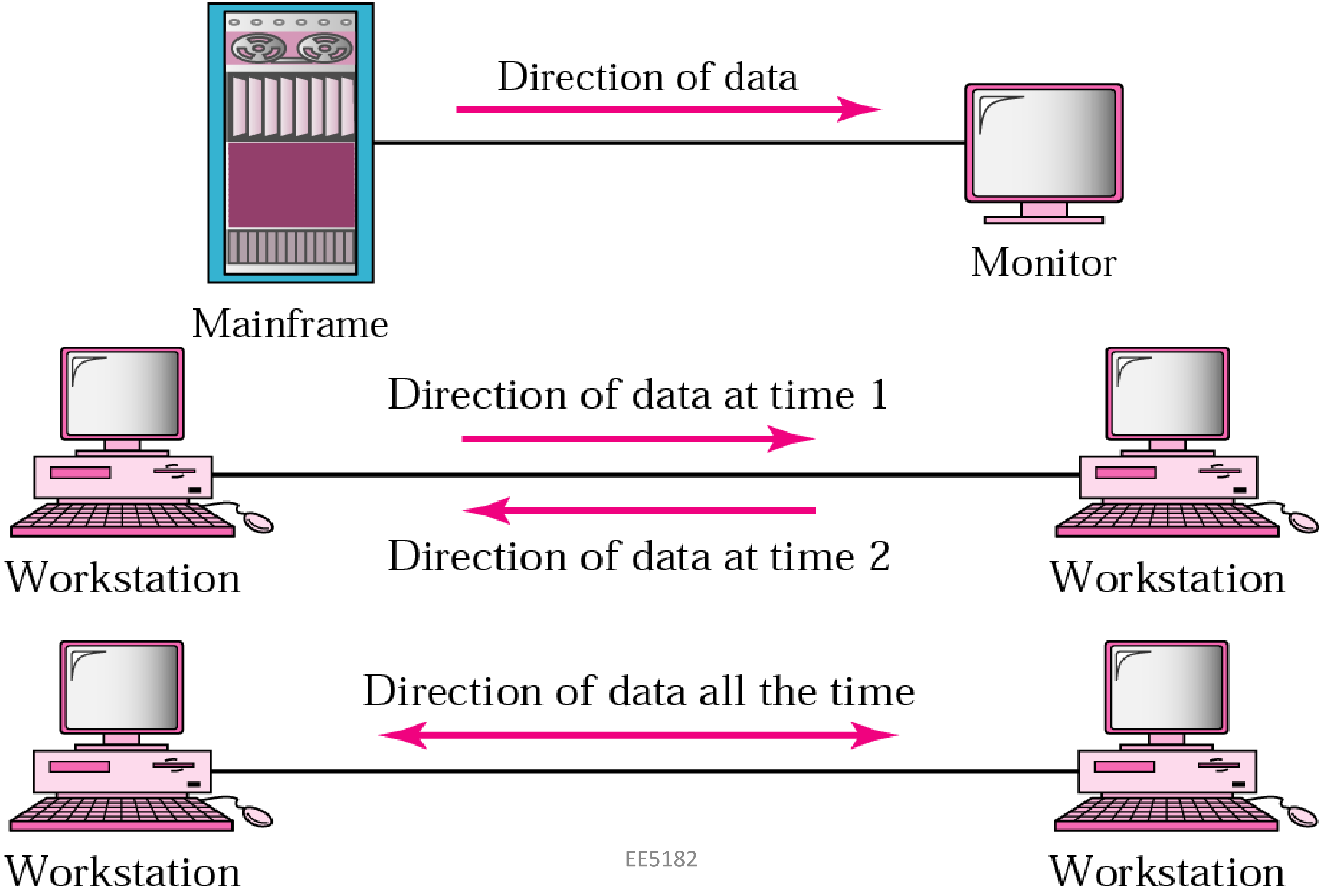
Advanced communication principles

- Layering
  - Break complexity of communication protocol into pieces easier to design and understand
  - Lower levels provide services to higher level
    - Lower level might work with bits while higher level might work with packets of data
  - Physical layer
    - Lowest level in hierarchy
    - Medium to carry data from one actor (device or node) to another
- Parallel communication
  - Physical layer capable of transporting multiple bits of data
- Serial communication
  - Physical layer transports one bit of data at a time
- Wireless communication
  - No physical connection needed for transport at physical layer

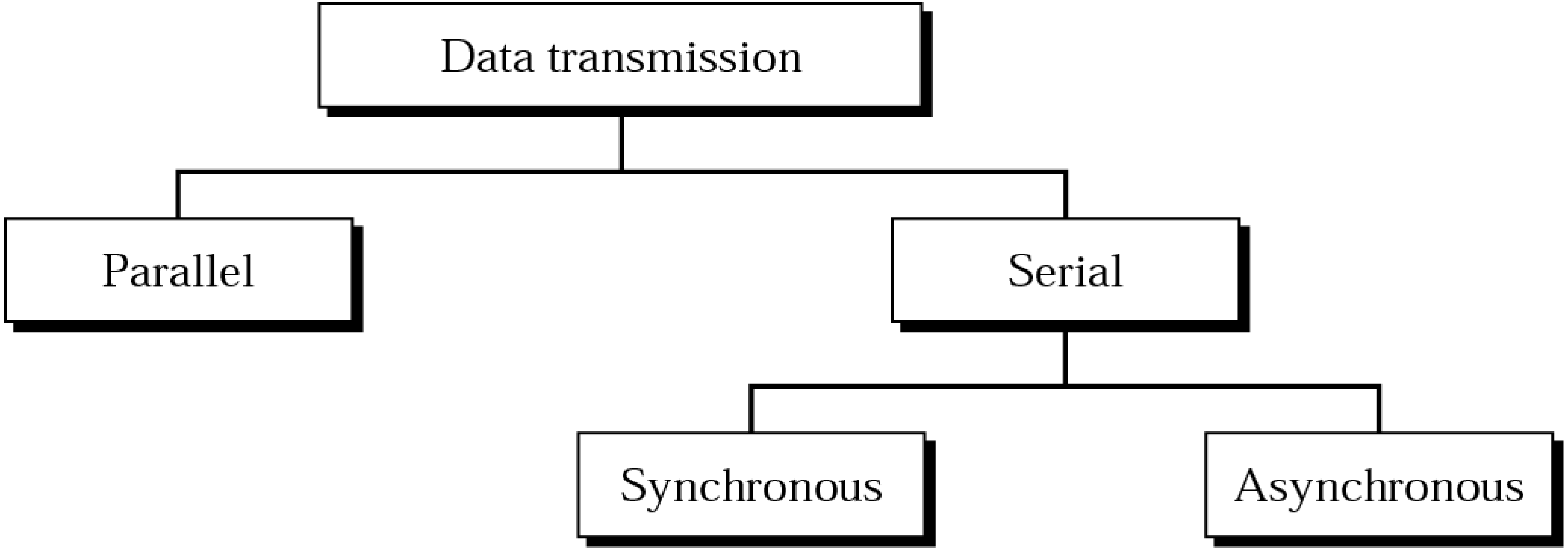
Basic communication methods

- It is essential to understand some of the basic communication methods that can be used to interconnect control systems.
- Characteristics of communication methods:
  - Simplex, Duplex & Semi Duplex
  - Serial Vs Parallel
  - Synchronous Vs Asynchronous
  - Data Throughput

Simplex, Half-duplex, Full-duplex

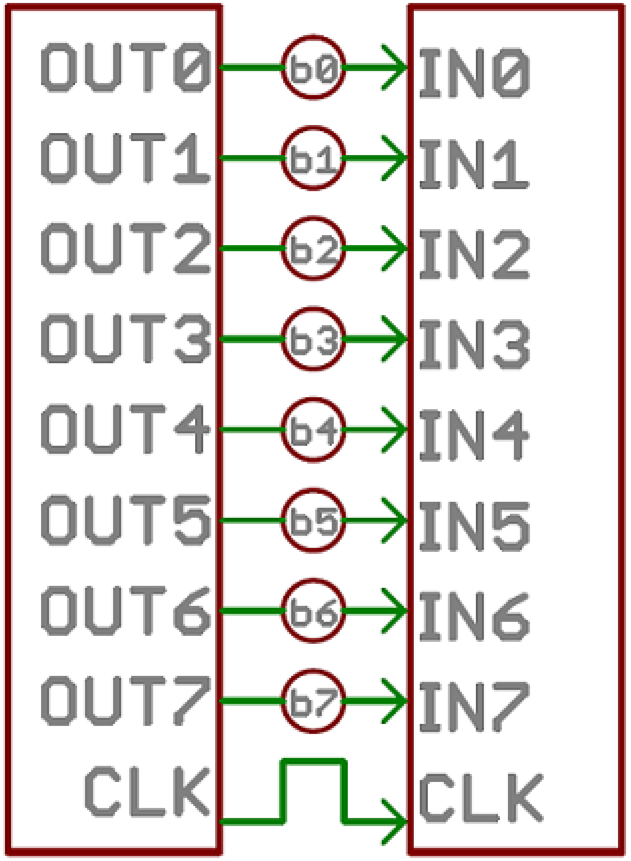

Direction of data
Mainframe
Monitor
Direction of data at time 1
Direction of data at time 2
Workstation
Workstation
Direction of data all the time
Workstation
Workstation

Types of data transfer



Parallel Communication

- Parallel interfaces transfer multiple bits at the same time.
- They usually require **buses** of data - transmitting across eight, sixteen, or more wires.
- It’s fast, straightforward, and relatively easy to implement.
- But it requires many more input/output (I/O) lines.

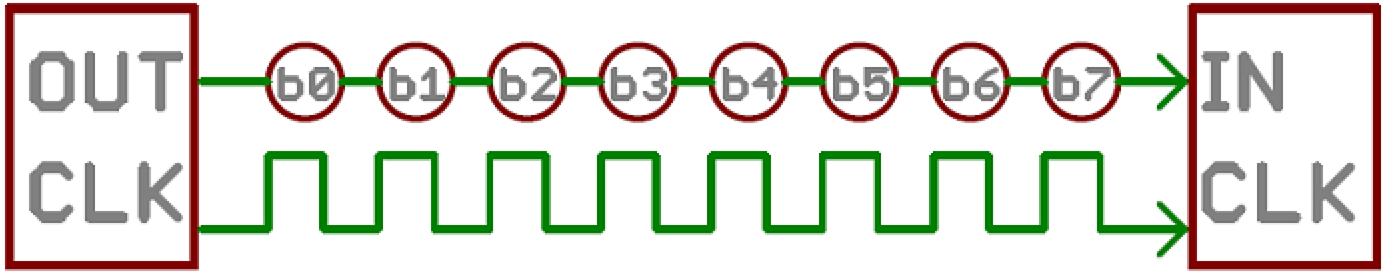


Parallel communication

- Multiple data, control, and possibly power wires
  - One bit per wire
- High data throughput with short distances
- Typically used when connecting devices on same IC or same circuit board
  - Bus must be kept short
    - long parallel wires result in high capacitance values which requires more time to charge/discharge
    - Data misalignment between wires increases as length increases
- Higher cost, bulky

Serial Communication

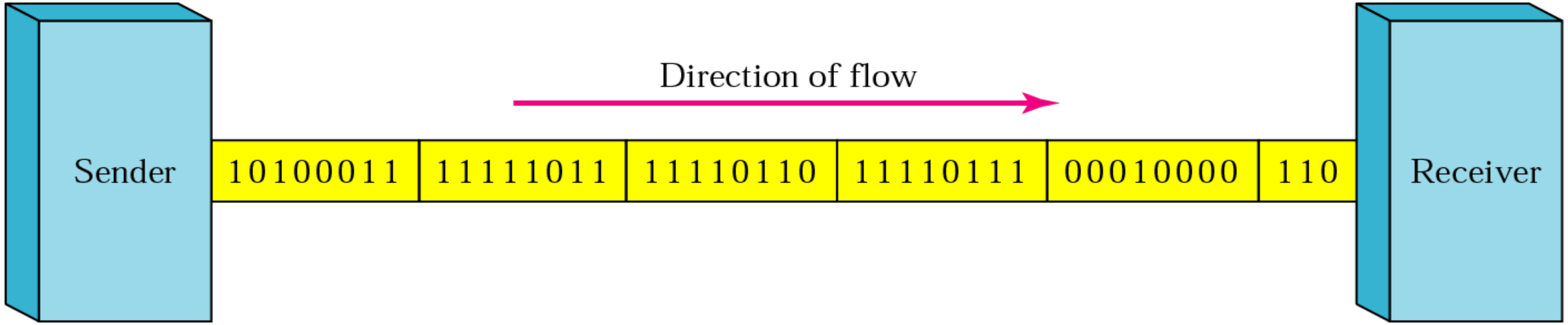
- Serial communication is the process of sending/receiving data in one bit at a time.
- Serial interfaces stream their data, one single bit at a time.
- These interfaces can operate on as little as one wire, usually never more than four.



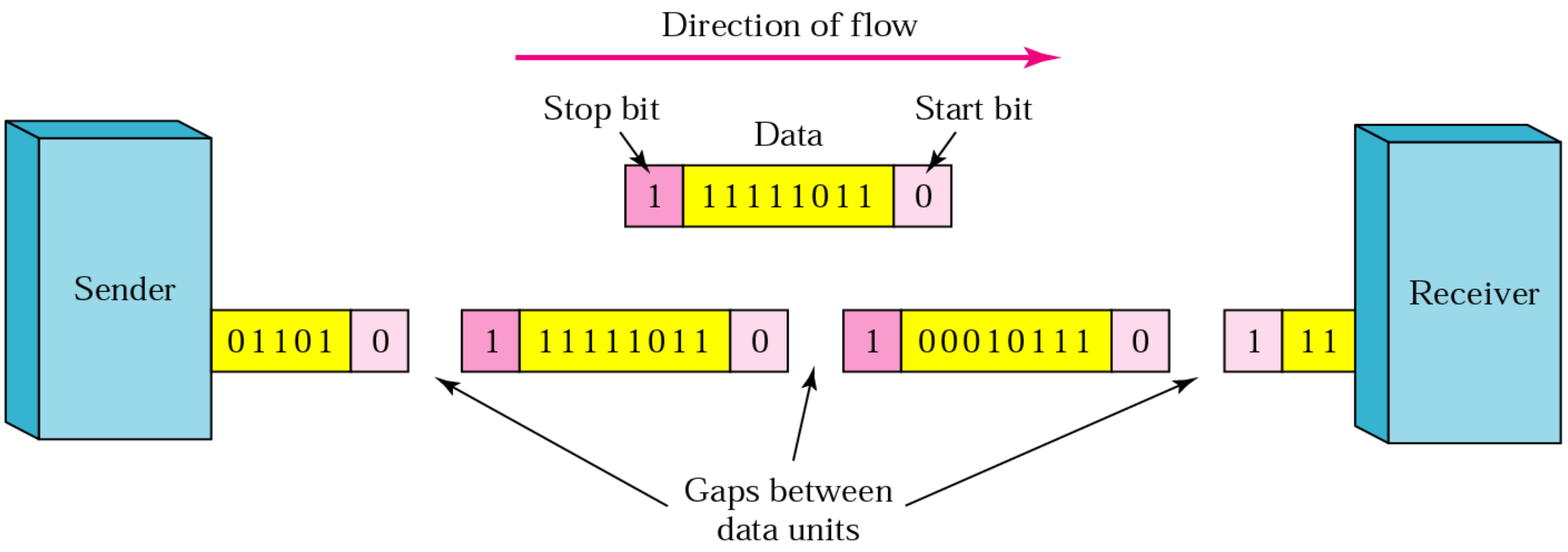
Serial communication

- Single data wire, possibly also control and power wires
- Words transmitted one bit at a time
- Higher data throughput with long distances
  - Less average capacitance, so more bits per unit of time
- Cheaper, less bulky
- More complex interfacing logic and communication protocol
  - Sender needs to decompose word into bits
  - Receiver needs to recompose bits into word
  - Control signals often sent on same wire as data increasing protocol complexity

Synchronous serial transfer



Asynchronous serial transfer



Advantages of Serial over Parallel

- A serial connection requires fewer interconnecting cables (e.g. wires/fibers) and hence occupies less space.
  - The extra space allows for better isolation of the channel from its surroundings.
  - Crosstalk is not a much significant issue, because there are fewer conductors in proximity.
- In many cases, serial is a better option because it is cheaper to implement.
  - Many devices and sensors relevant to control systems have serial interfaces, as opposed to parallel ones, so that they have fewer pins and are therefore less expensive.

Serial Communication Protocols

- There are various protocols that can be used with digital control systems for serial communication.
  - UART (Universal Asynchronous Receiver/Transmitter)
  - SPI (Serial Peripheral Interface)
  - I2C (Inter-Integrated Circuits)
  - CAN (Controller Area Network)
  - USB (Universal Serial Bus)
  - 1-wire
- https://www.deviceplus.com/how-tos/arduino-guide/arduino-communication-protocols-tutorial/
- https://www.embedded.com/design/connectivity/4023975/Serial-Protocols-Compared

Assignment: *SPI and* I^2C

Discuss the differences and advantages between the serial communication protocols, SPI and I2C.

File name: StudentIndexNo_Name.pdf

Upload to:
https://script.google.com/macros/s/AKfycbz96T6CJyCjUiaUCZC2WQkkD4Kk_XgEwAhWLXiQLDETdL9wHkM4/exec