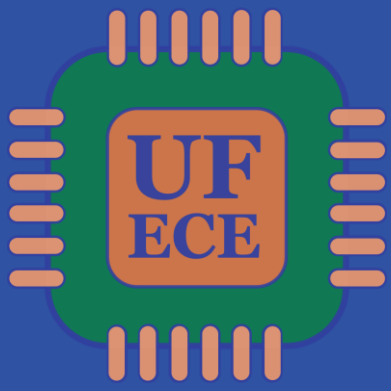
Microprocessor Applications

Overview of Serial Communication



UF UNIVERSITY of FLORIDA

Dr. Eric Schwartz
Christopher Crary
Wesley Piard

1

What is Serial Communication?

- Serial communication is a very broad form of communication and can be loosely defined as the sequential transmission of bits, one at a time, via some shared medium.
- Serial communication is mostly digital in the modern age, but it started out in physical or mechanical forms, e.g., binary smoke signaling, Morse code, etc.

2

Serial vs. Parallel Communication

- Serial communication usually consists of a single bit being transmitted at a time.
- Parallel communication always consists of multiple bits being transmitted at a time.

Serial

Pros and Cons of Serial Communication

- Pros
  - Fewer wires and connections are required when compared to parallel communication.
  - It allows for smaller devices with lower pin counts and less energy consumption to be employed.
  - Generally, serial busses are not affected by noise as much as parallel busses. Crosstalk between signals in a parallel bus is a critical issue at higher data rates.
- Cons
  - Serial communication typically slower data rates than parallel communication because only one bit at a time is transmitted.

Different Types of Serial Communication Protocols

There are two main types of serial protocols: a**synchronous** and **synchronous**.

- *Asynchronous* serial protocols use a common data rate that is shared between two entities that wish to communicate. In addition, start and stop signals are often used to indicate the beginning or end of a transmission.
- *Synchronous* serial protocols use a dedicated signal for synchronization that is shared between two entities and controls the rate at which data is transmitted and received. This signal is often called a **clock** signal.

Asynchronous Communication Protocols

- Unlike synchronous protocols, asynchronous protocols do not use a shared clock signal for communication synchronization. Instead, they ensure that the entities trying to communicate transmit and receive at the same data rate.
- As long as the transmitter sends data at the agreed-upon rate, the receiver can properly interpret the data.
- This shared rate is called the **baud rate** which is typically expressed in bits per second.

Examples of Asynchronous Communication Protocols

- 20mA current loop
  - Logical 1: Represented by a pulse of 20 mA
  - Logical 0: Represented by 0 mA, or no current.
- RS-232 (EIA-232)
  - Logical 1: Negative voltage (-3V to -25V)
  - Logical 0: Positive voltage (+3V to +25V)
  - Large voltages used to allow for better signal integrity at longer distances.

Examples of Asynchronous Communication Protocols

- Universal Asynchronous Receiver/Transmitter (**UART**)
  - UARTs are *universal* serial components that can be used to implement some different protocols.
  - They typically use non-negative DC voltages.
  - Logical 1: Positive DC voltage (commonly 3.3V, 5V, etc.)
  - Logical 0: Zero DC voltage (0V, ground, etc.)
  - UARTs don't have as many strict guidelines and are easier to setup and get working compared to other asynchronous protocols.

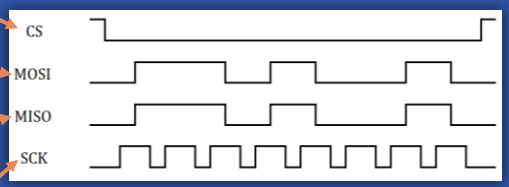
Synchronous Communication Protocols

- Synchronous protocols use a **clock** signal shared between two entities that controls the rate at which data is transmitted and received.
- The rising or falling edges of the clock signal dictate when data should be "clocked" into the receiving entity.
- Since there is a dedicated signal for synchronization, synchronous protocols tend to have higher data rates than asynchronous protocols.

Examples of Synchronous Communication Protocols

Serial Peripheral Interface (**SPI**)

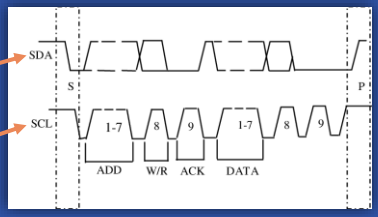
- An SPI port is usually comprised of four signals:
  - Chip Select (**CS**) – Tells a receiving device when it should begin to interpret data
  - Master-out Slave-in (**MOSI**) – Data transmitted from the master to a slave device
  - Master-in Slave-out (**MISO**) – Data received by a slave device
  - Clock (**SCK**) – Dictates when each data bit is valid and can be read by the receiving device



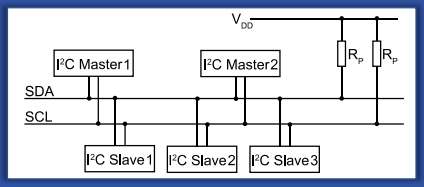
Examples of Synchronous Communication Protocols

Inter-Integrated Circuit (I²C)

- An I²C port is usually comprised of two signals:
  - Serial Data (**SDA**) – Bidirectional signal for data to propagate
  - Serial Clock (**SCL**) – Dictates when data is valid and can be read by the receiving device



- Multiple masters and slaves can share the same bus, which is a powerful feature, but it requires a more complicated protocol to make sure there are no bus contentions.
- Each slave device has a slave *address* associated with it which is what allows the bus to be shared.



Conclusion

Serial communication is used nearly everywhere in embedded systems. Here are some examples:

- Microcontrollers interfacing with each other
- Microcontrollers interfacing with a PC serial port
- Sensors (ADC, IMU, Magnetometer, Temperature, etc.)
- External memory devices
- LCDs
- Wireless communication