Serial Communication Interface

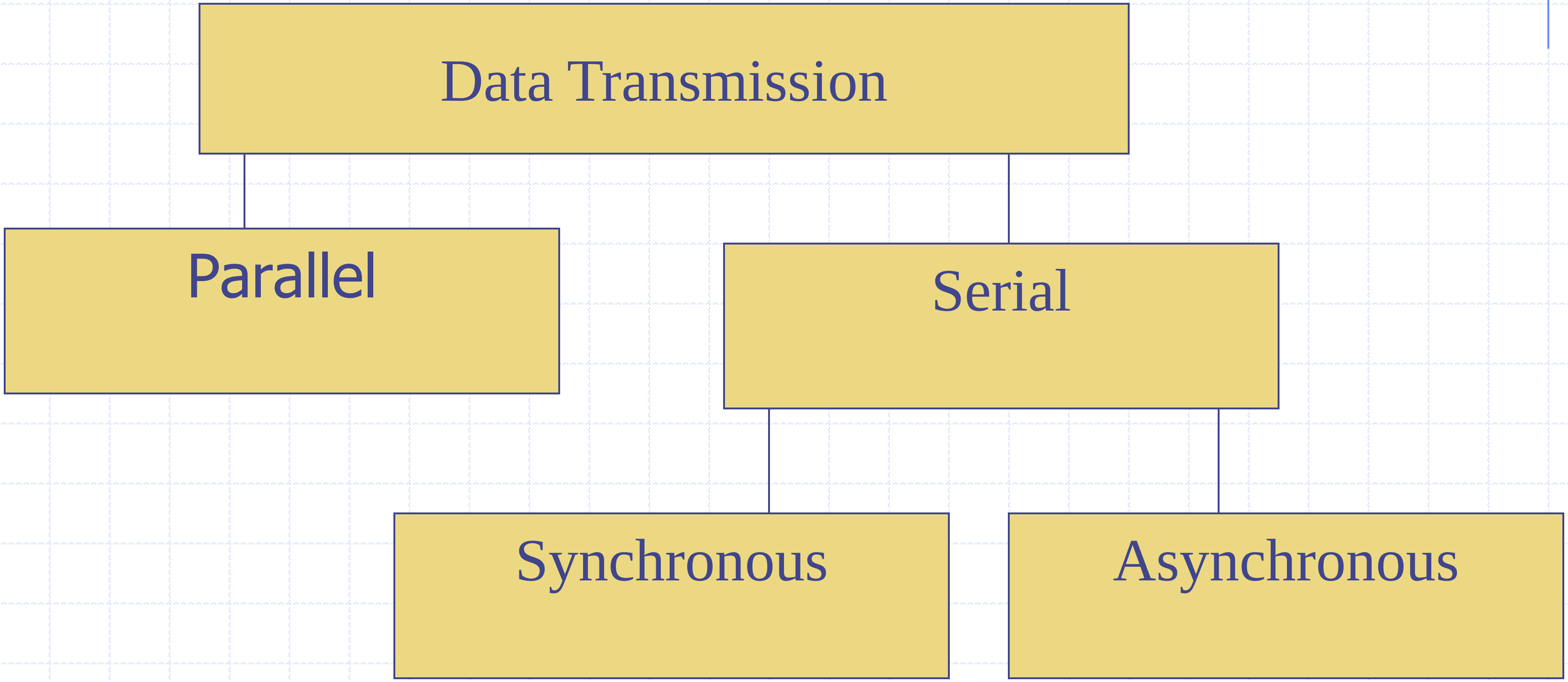
Agenda

- Basic Definitions
- Detailed Information
- Concrete Examples

Learning Objectives

- Describe the Difference Between Serial and Parallel Communication
- Explain Asynchronous Communication
- Determine Time Needed to Transmit a Block of Data
- Describe a Common Error Detection Mechanism
- Serial Communication with Arduino

Data Transmission Tree



Data Transmission

Parallel

Serial

Synchronous

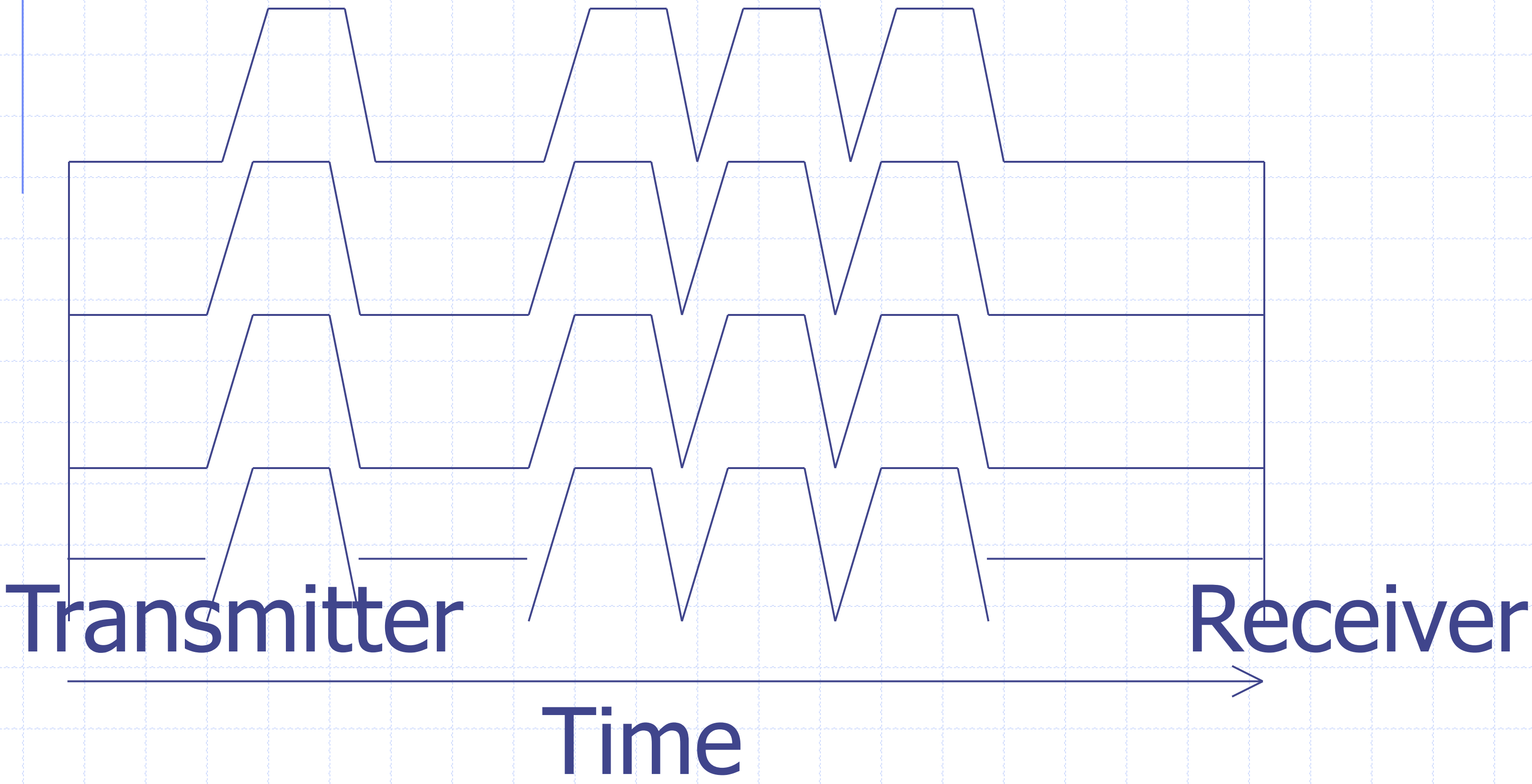
Asynchronous

Definition: Parallel

- Data is sent and received more than one bit at a time
- Transmission on multiple wires

Parallel Communication

- Many lines of communication, synchronized bursts of data



Endianness, how it relates to communication

- Big Endian- MSB first, less significant bytes in descending order
- Little Endian- MSB last, data in ascending order

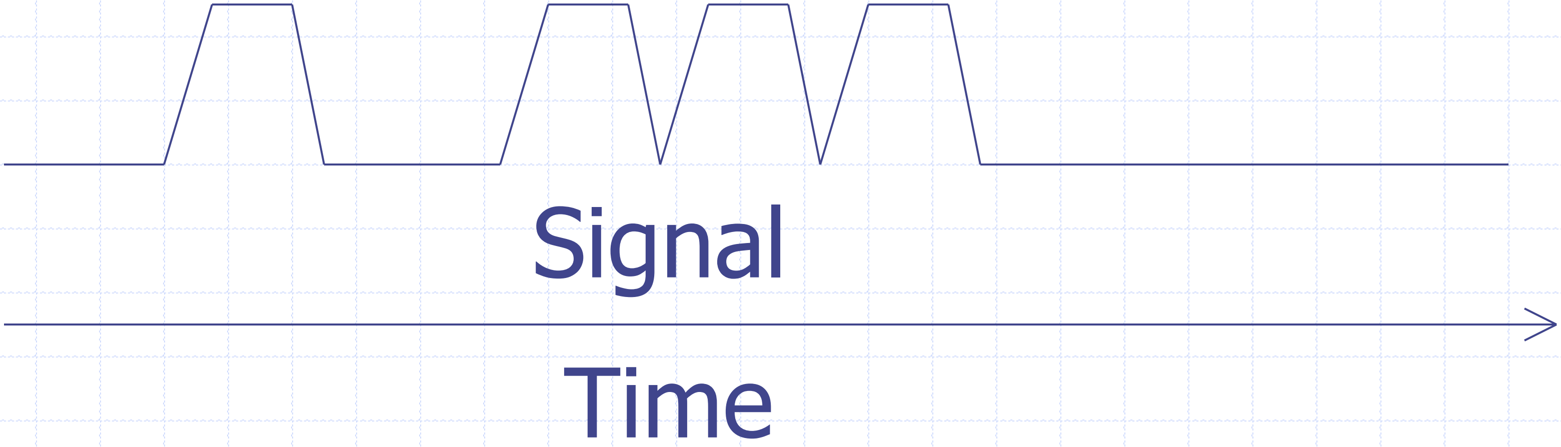
- Endian type determines how the data is interpreted, and how it should be sent in both serial and parallel communication.

Definition: Serial

- Data is sent and received one bit at a time
- Transmission on single wire

Serial Communication

- One line of communication, long string of data



RS232, SCI, and SPI

- RS232- Typical computer COM port
- SCI- Serial Communication interface, uses the universal asynchronous receiver/transmitter or UART
- SPI Serial peripheral interface, part of Port D.

Why Serial?

- Fewer wires translates to
    - Lower cost
    - Simpler set-up

Definition: Synchronous

- Sender and receiver have their clocks synchronized
- Transmissions occur at specified intervals

- Advantage:
  - Faster

Definition: Asynchronous

- Devices are not synchronized
- Transmissions happen at unpredicted intervals

- Advantages:
  - Simpler
  - More robust

Please Note:

- Both synchronous and asynchronous must have agreed upon bit transfer rate

Why Asynchronous?

- Disadvantage:
  - Slower due to overhead
- Advantages:
  - Simpler
  - Cheaper
  - Information can be sent when ready

FYI Term: “UART”

- Universal
- Asynchronous
- Receiver-
- Transmitter

“…a computer component that handles asynchronous serial communication.”

www.webopedia.com

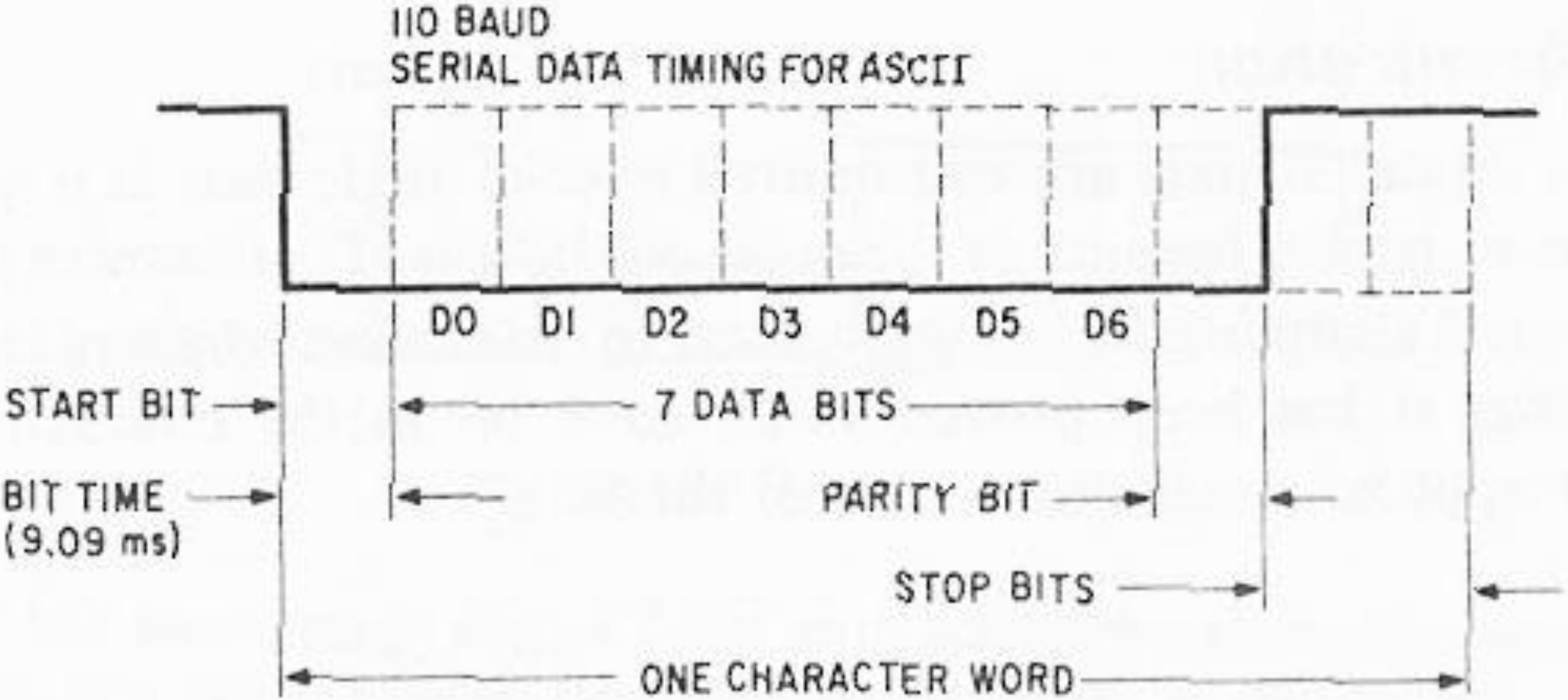


Fig. 9.23 Role of stop, start, and parity bits

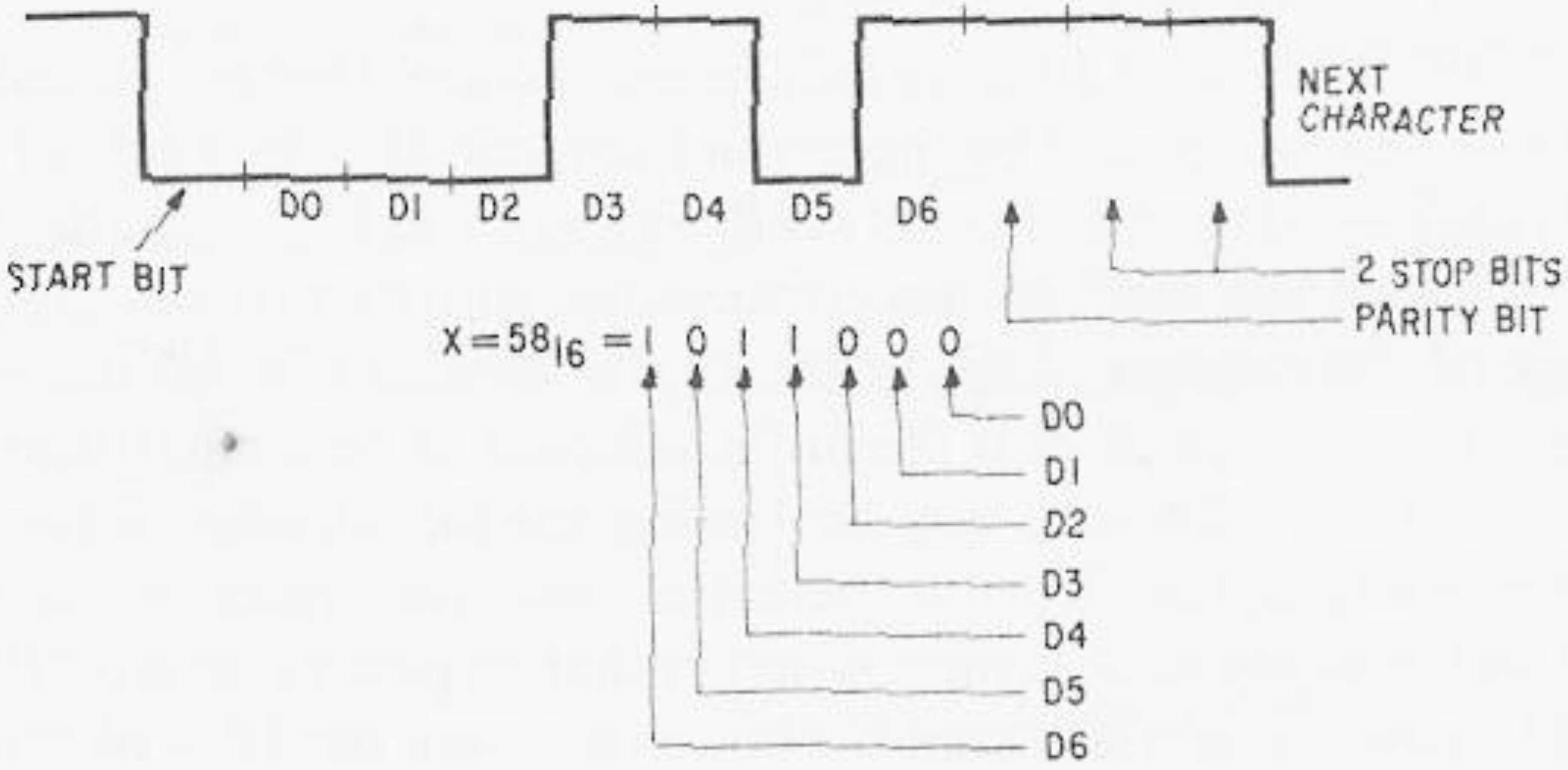


Fig. 9.24 Pulse train

Definitions

- Start Bit
  - Signals the beginning of the data word
  - A low bit after a series of high bits
- Data Bits
  - The meat of the transmission
  - Usually 7 or 8 bits

Definitions Continued

- Parity Bit
  - An error check bit placed after the data bits
  - Can be high or low depending on whether odd parity or even parity is specified
- Stop Bit/s
  - One or two high bits that signal the end or the data word
- Data Word
  - Start Bit, Data Bits, Parity Bit, & Stop Bit/s

BAUD RATE

$\neq$

BIT RATE

Baud Rate

- Baud Rate = bits transferred/second
- baud rate INCLUDES start, stop, and parity
- “bit rate” refers to JUST data bits transferred per second (may include parity)
- baud rate > bit rate