**Byte ordering**

**Byte ordering** (also known as **endianess**) refers to the sequence in which bytes are arranged into larger numerical values when stored in memory or transmitted over digital communication links.

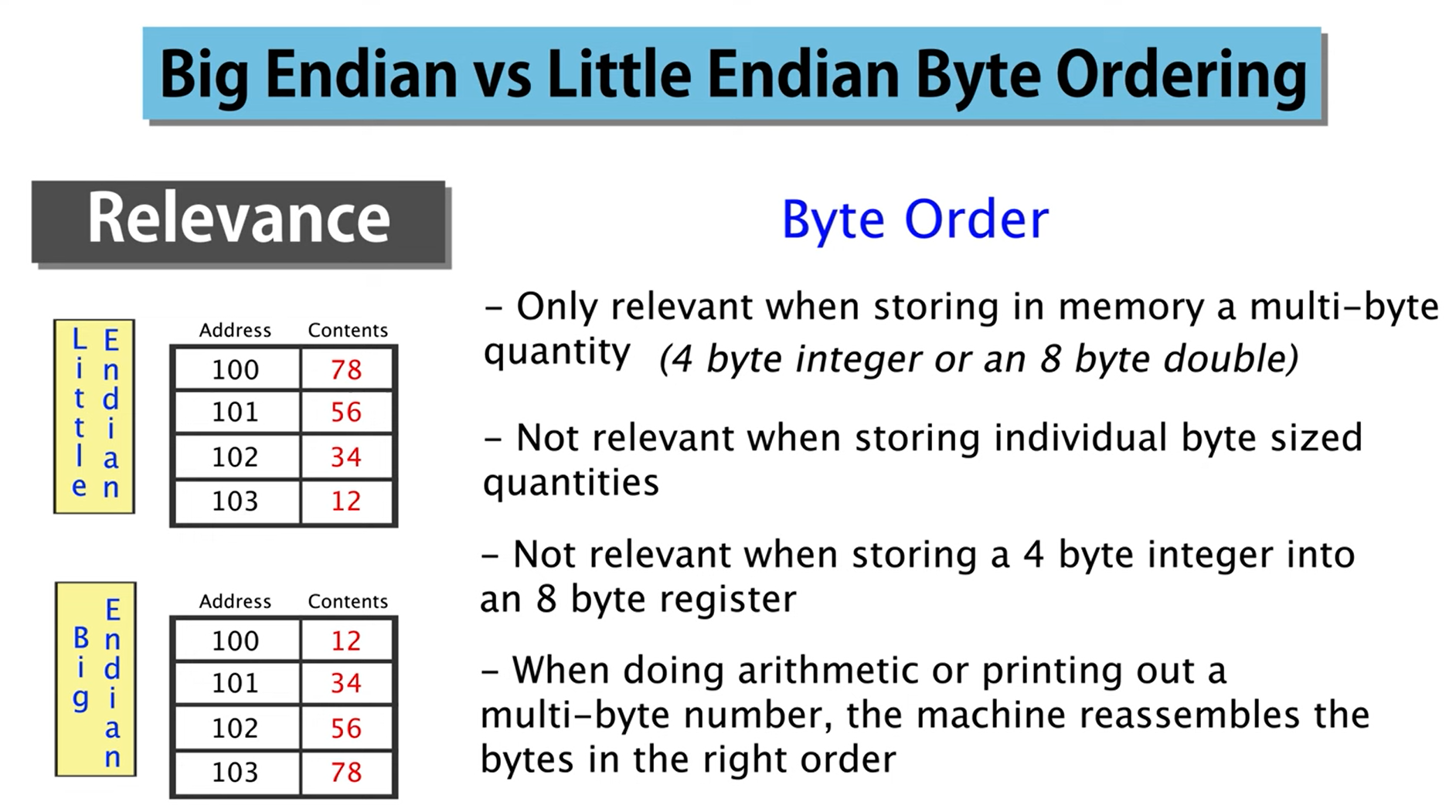
There are two main types of byte ordering:

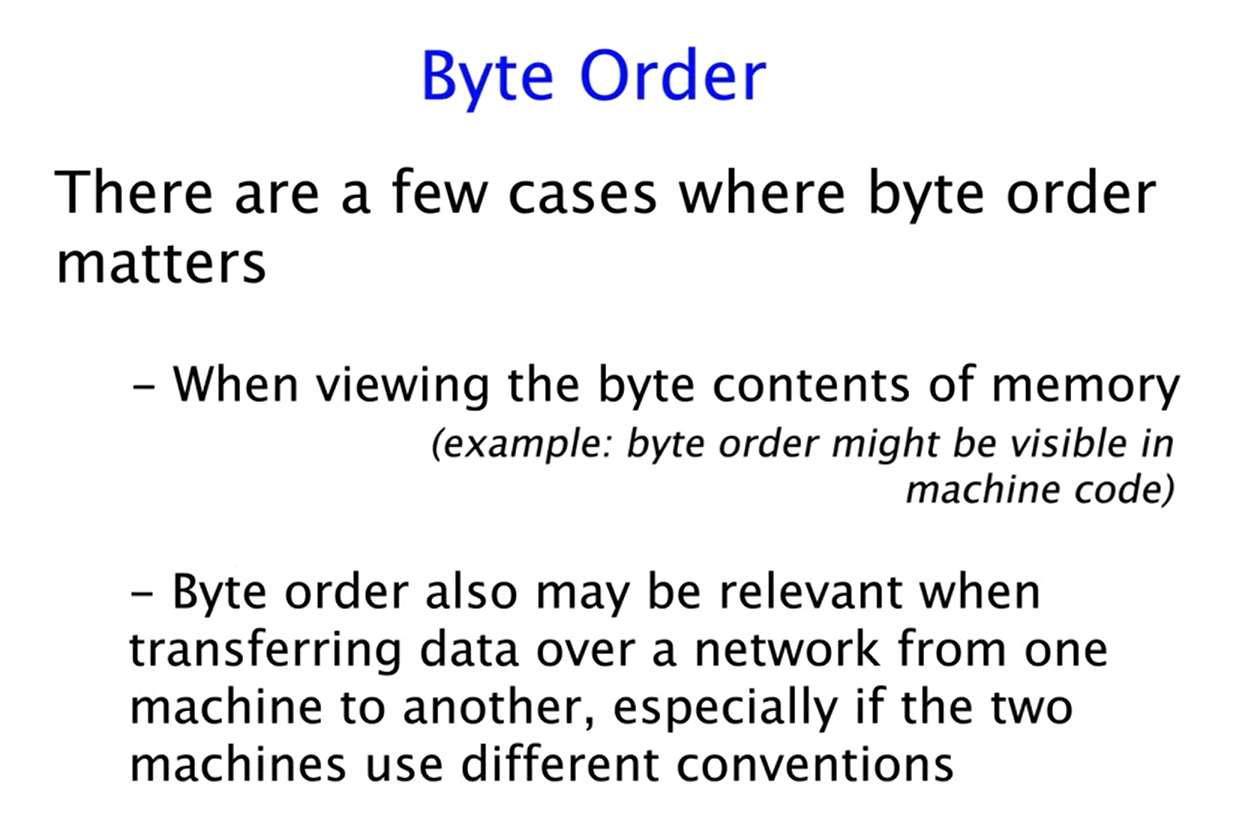
1. **Little-Endian**:
   * In this ordering, the least significant byte (LSB) is stored at the lowest memory address, and the most significant byte (MSB) is stored at the highest address.
   * Example: The 32-bit hexadecimal number 0x12345678 would be stored in memory as 78 56 34 12 in little-endian format.
2. **Big-Endian**:
   * In this ordering, the most significant byte (MSB) is stored at the lowest memory address, and the least significant byte (LSB) is stored at the highest address.
   * Example: The 32-bit hexadecimal number 0x12345678 would be stored in memory as 12 34 56 78 in big-endian format.

**Example in Detail**

Consider storing the 16-bit value 0x1234 in memory:

* **Little-Endian**:
  + Memory Address: 1000 1001
  + Value Stored: 34 12
* **Big-Endian**:
  + Memory Address: 1000 1001
  + Value Stored: 12 34





**Intel 8086 Microprocessor Format**

The Intel 8086 microprocessor uses **little-endian** format for storing multi-byte data in memory.

**Little-Endian Format Explained:**

* **Little-endian** means that the least significant byte (LSB) of a multi-byte value is stored at the lowest memory address, and the most significant byte (MSB) is stored at the highest memory address.

**Example:**

Suppose you want to store the 16-bit hexadecimal number 0x1234 in memory.

* In **little-endian** format:
  + The byte 0x34 (LSB) is stored first at the lower memory address.
  + The byte 0x12 (MSB) is stored next at the higher memory address.