Serial communication
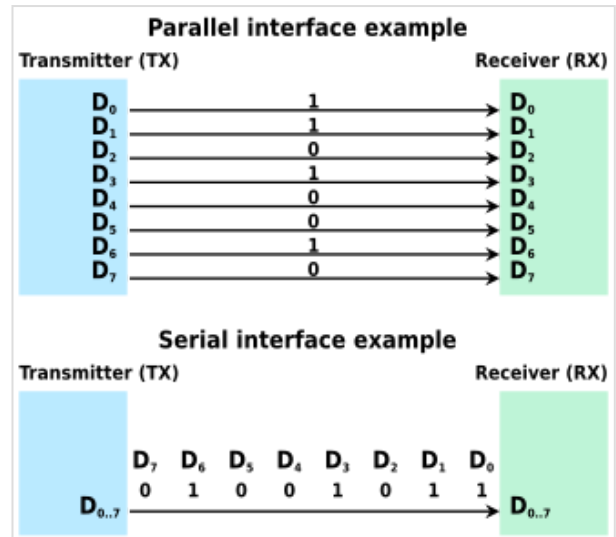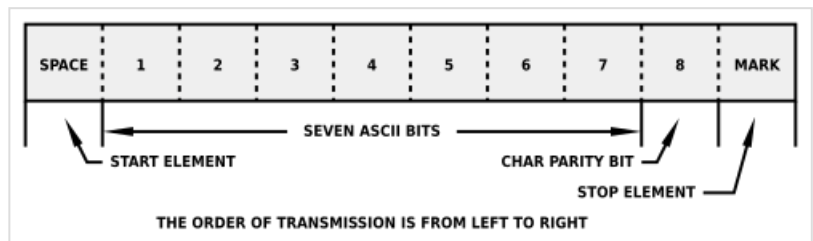
In telecommunication and data transmission, **serial communication** is the process of sending data one bit at a time, sequentially, over a communication channel or computer bus. This is in contrast to parallel communication, where several bits are sent as a whole, on a link with several parallel channels.

Serial communication is used for all long-haul communication and most computer networks, where the cost of cable and difficulty of synchronization make parallel communication impractical. Serial computer buses have become more common even at shorter distances, as improved signal integrity and transmission speeds in newer serial technologies have begun to outweigh the parallel bus's advantage of simplicity (no need for serializer and deserializer, or SerDes) and to outstrip its disadvantages (clock skew, interconnect density). The migration from PCI to PCI Express (PCIe) is an example.

Serial and parallel data transmission of 01001011_2. Standard bit sequence is least-significant-bit-first (D_0 to D_7 in ascending order).[1] D_0 is received first via serial transmission. All bits are received simultaneously via parallel transmission.

Modern high speed serial interfaces such as PCIe[2][3][4] send data several bits at a time using modulation/encoding techniques such as PAM4 which groups 2 bits at a time into a single symbol, and several symbols are still sent one at a time. This replaces PAM2 or non return to zero (NRZ) which only sends one bit at a time, or in other words one bit per symbol.[5][6][7][8][9][10][11][12] The symbols are sent at a speed known as the symbol rate or the baud rate.[13][14][15][16]

Standard character structure for asynchronous data communication consisting of 10 elements for a 7-bit ASCII character

Cables

Many serial communication systems were originally designed to transfer data over relatively large distances through some sort of data cable.

Practically all long-distance communication transmits data one bit at a time, rather than in parallel, because it reduces the cost of the cable. The cables that carry this data (other than "the" serial cable) and the computer ports they plug into are usually referred to with a more specific name, to reduce confusion.

Keyboard and mouse cables and ports are almost invariably serial—such as PS/2 port, Apple Desktop Bus and USB.

The cables that carry digital video are also mostly serial—such as coax cable plugged into a HD-SDI port, a webcam plugged into a USB port or FireWire port, Ethernet cable connecting an IP camera to a Power over Ethernet port, FPD-Link, digital telephone lines (ex. ISDN), etc.

Other such cables and ports, transmitting data one bit at a time, include Serial ATA, Serial SCSI, Ethernet cable plugged into Ethernet ports, the Display Data Channel using previously reserved pins of the VGA connector or the DVI port or the HDMI port.

Serial buses

Many communication systems were generally designed to connect two integrated circuits on the same printed circuit board, connected by signal traces on that board (rather than external cables).
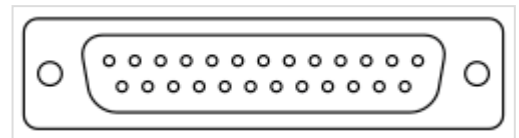
RS-232 connector (D-Sub DB-25 variant)

Integrated circuits are more expensive when they have more pins. To reduce the number of pins in a package, many ICs use a serial bus to transfer data when speed is not important. Some examples of such low-cost lower-speed serial buses include RS-232, DALI, SPI, CAN bus, I²C, UNI/O, and 1-Wire. Higher-speed serial buses include USB, SATA and PCI Express.

Serial versus parallel

The communication links, across which computers (or parts of computers) talk to one another, may be either serial or parallel. A parallel link transmits several streams of data simultaneously along multiple channels (e.g., wires, printed circuit tracks, or optical fibers); whereas, a serial link transmits only a single stream of data. The rationale for parallel communication was the added benefit of having Direct Memory Access to the 8-bit or 16-bit registry addresses at a time where mapping direct data lanes was more convenient and faster than synchronizing data serially.

Although a serial link may seem inferior to a parallel one, since it can transmit less data per clock cycle, it is often the case that serial links can be clocked considerably faster than parallel links in order to achieve a higher data rate. Several factors allow serial to be clocked at a higher rate:

- Clock skew between different channels is not an issue (for unclocked asynchronous serial communication links). This can be caused by mismatched wire or conductor lengths.[17][18]
- 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 less of an issue, because there are fewer conductors in proximity.[17]
- Budgets for power use, power dissipation, cable cost, component cost, IC die area, PC board area, ESD protection, etc. can be focused on a single link.

The transition from parallel to serial buses was allowed by Moore's law which allowed for the incorporation of SerDes in integrated circuits.[19] An electrical serial link only requires a pair of wires, whereas a parallel link requires several. Thus serial links can save on costs (also known as the Bill of Materials). Differential signalling uses length-matched wires or conductors and are used in high speed serial links.[20] Length-matching is easier to perform on serial links as they require fewer conductors.

In many cases, serial is cheaper to implement than parallel. Many ICs have serial interfaces, as opposed to parallel ones, so that they have fewer pins and are therefore less expensive.

Examples of architectures

- ARINC 818 Avionics Digital Video Bus
- Atari SIO (Joe Decuir credits his work on Atari SIO as the basis of USB)
- Binary Synchronous Communications BSC - Binary Synchronous Communications
- CAN Control Area Network Vehicle Bus
- ccTalk Used in the money transaction and point-of-sale industry
- CoaXPress industrial camera protocol over Coax
- DMX512 control of theatrical lighting
- Ethernet
- Fiber Distributed Data Interface
- Fibre Channel (high-speed, for connecting computers to mass storage devices)
- FireWire
- HDMI
- HyperTransport
- InfiniBand (very high speed, broadly comparable in scope to PCI)
- I²C multidrop serial bus
- MIDI control of electronic musical instruments
- MIL-STD-1553A/B
- Morse code telegraphy
- PCI Express
- Profibus
- RS-232 (low-speed, implemented by serial ports)
- RS-422 multidrop serial bus
- RS-423
- RS-485 multidrop multimaster serial bus
- SDI-12 industrial sensor protocol
- SERCOM
- Serial ATA
- Serial Attached SCSI
- Shift Register with serial-in and serial-out configuration
- SONET and SDH (high speed telecommunication over optical fibers)
- SpaceWire Spacecraft communication network

- [S/PDIF](#) and [AES3](#) audio communication protocols
- [SPI](#)
- [T-1](#), [E-1](#) and variants (high speed telecommunication over copper pairs)
- [Universal Serial Bus](#) (for connecting peripherals to computers)
- [UNI/O](#) multidrop serial bus
- [1-Wire](#) multidrop serial bus

See also

- [8N1](#)
- [Asynchronous serial communication](#)
- [Comparison of synchronous and asynchronous signalling](#)
- [Computer bus](#)
- [Data transmission](#)
- [Eye pattern](#)
- [Federal Standard 1037C](#)
- [High-Level Data Link Control](#) (HDLC)
- [List of device bandwidths](#)
- [MIL-STD-188](#)
- [Serial Peripheral Interface Bus](#)
- [Serial port](#)
- [Synchronous serial communication](#)
- [Universal asynchronous receiver/transmitter](#) (UART)

References

1. Mackenzie, Charles E. (1980). *Coded Character Sets, History and Development* (https://text files.meulie.net/bitsaved/Books/Mackenzie_CodedCharSets.pdf) (PDF). The Systems Programming Series (1 ed.). Addison-Wesley Publishing Company, Inc. pp. 247–253. ISBN 978-0-201-14460-4. LCCN 77-90165 (https://lccn.loc.gov/77-90165). Archived (https://web.archive.org/web/20160526172151/https://textfiles.meulie.net/bitsaved/Books/Mackenzie_CodedCharSets.pdf) (PDF) from the original on May 26, 2016. Retrieved December 29, 2022.
2. Robinson, Dan (12 January 2022). "Final PCIe 6.0 specs unleashed: 64 GTps link speed incoming... with products to follow in 2023" (https://www.theregister.com/2022/01/12/final_pcie_60_specs_released/). *The Register*.
3. "PCIe 7.0 Draft 0.5 Spec Available: 512 GB/S over PCIe x16 on Track for 2025" (https://web.archive.org/web/20240404125350/https://www.anandtech.com/show/21335/full-draft-of-pcie-70-spec-available-512-gbs-over-pcie-x16-incoming). Archived from the original (https://www.anandtech.com/show/21335/full-draft-of-pcie-70-spec-available-512-gbs-over-pcie-x16-incoming) on April 4, 2024.
4. "PCIe 5.0 is just beginning to come to new PCS, but version 6.0 is already here" (https://arstechnica.com/gadgets/2022/01/pci-express-6-0-spec-is-finalized-doubling-bandwidth-for-ssds-gpus-and-more/). 12 January 2022.
5. *Handbook of Serial Communications Interfaces: A Comprehensive Compendium of Serial Digital Input/Output (I/O) Standards* (https://books.google.com/books?id=wnGDBAAAQBAJ&dq=serial+pam4+signals&pg=PA9). Newnes. 21 August 2015. ISBN 978-0-12-800671-9.

6. "PAM4: For Better and Worse | 2019-02-26 | Signal Integrity Journal" (https://www.signalintegrityjournal.com/articles/1151-pam4-for-better-and-worse).
7. "PAM-4 Signaling" (https://semiengineering.com/knowledge_centers/communications-io/off-chip-communications/pam-4-signaling/).
8. Zhang, Hongtao; Jiao, Brandon; Liao, Yu; Zhang, Geoff. *PAM4 Signaling for 56G Serial Link Applications − A Tutorial* (https://www.xilinx.com/publications/events/designcon/2016/slides-pam4signalingfor56gserial-zhang-designcon.pdf) (PDF). DesignCon 2016.
9. "PAM4 Signaling in High-Speed Serial Technology: Test, Analysis, and Debug" (https://download.tek.com/document/PAM4-Signaling-in-High-Speed-Serial-Technology_55W-60273.pdf) (PDF) (application note). Tektronix.
10. Pan, Zhongqi; Yue, Yang (3 December 2019). *Advanced DSP Techniques for High-Capacity and Energy-Efficient Optical Fiber Communications* (https://books.google.com/books?id=JljBDwAAQBAJ&dq=pam4+symbol&pg=PA54). ISBN 978-3-03921-792-2.
11. *Essentials of Modern Communications* (https://books.google.com/books?id=MlruDwAAQBAJ&dq=pam4+symbol&pg=PA944). John Wiley & Sons. 4 August 2020. ISBN 978-1-119-52149-5.
12. Kim, Gain (January 2022). "Design Space Exploration of Single-Lane OFDM-Based Serial Links for High-Speed Wireline Communications" (https://www.researchgate.net/figure/Eye-diagrams-of-PAM-2-4-8-with-normalized-vertical-full-swing-level-Peak-to-peak-swings_fig2_361960252). *IEEE Open Journal of Circuits and Systems*. **3** (1). Figure 2. doi:10.1109/OJCAS.2022.3189550 (https://doi.org/10.1109%2FOJCAS.2022.3189550). ISSN 2644-1225 (https://search.worldcat.org/issn/2644-1225).
13. "Eye diagrams: The tool for serial data analysis" (https://www.edn.com/eye-diagrams-the-tool-for-serial-data-analysis/). 4 June 2019.
14. "Advantages of PAM4 modulation | Disadvantages PAM4 signaling" (https://www.rfwireless-world.com/Terminology/Advantages-and-disadvantages-of-PAM4-modulation.html).
15. "Generate PAM4 signals for receiver compliance testing" (https://www.edn.com/generate-pam4-signals-for-receiver-compliance-testing/). 20 September 2016.
16. *Complex Digital Hardware Design* (https://books.google.com/books?id=L1YIEQAAQBAJ&dq=pam4+symbol+rate&pg=PA16). CRC Press. 9 May 2024. ISBN 978-1-040-01179-9.
17. "Lecture 24" (https://courses.cs.washington.edu/courses/cse378/11wi/lectures/lec24.pdf) (PDF). *CSE378: Machine Organization & Assembly Language*.
18. Ledin, Jim; Farley, Dave (4 May 2022). *Modern Computer Architecture and Organization: Learn x86, ARM, and RISC-V architectures and the design of smartphones, PCS, and cloud servers* (https://books.google.com/books?id=BxptEAAAQBAJ). Packt Publishing. ISBN 978-1-80323-823-4.
19. *The Boundary — Scan Handbook* (https://books.google.com/books?id=aUCgNOpyUbgC&dq=parallel++serial++serdes+moore%27s+law&pg=PA275). Springer. 30 June 2003. ISBN 978-1-4020-7496-7.
20. Ledin, Jim; Farley, Dave (4 May 2022). *Modern Computer Architecture and Organization: Learn x86, ARM, and RISC-V architectures and the design of smartphones, PCS, and cloud servers* (https://books.google.com/books?id=BxptEAAAQBAJ). Packt Publishing. ISBN 978-1-80323-823-4.

External links

- Serial Interface Tutorial for Robotics (http://www.societyofrobots.com/microcontroller_uart.shtml) (contains many practical examples)
- Serial interfaces listing (with pinouts) (http://pinouts.ru/pin_SerialPorts.shtml)
- Wiki: Serial Ports (http://c2.com/cgi/wiki?SerialPorts)

- Visual studio 2008 coding for Serial communication (https://web.archive.org/web/200902261 95855/http://www.thaiio.com/prog-cgi/VBnetSerialPort.htm)
- Introduction to I²C and SPI protocols (http://www.byteparadigm.com/kb/article/AA-00255) Archived (https://web.archive.org/web/20160308185505/http://byteparadigm.com/kb/article/ aa-00255) 2016-03-08 at the Wayback Machine
- Serial communication introduction (https://learn.sparkfun.com/tutorials/serial-communicatio n/all)
- Serial Port Programming in Linux (https://training-kits.appspot.com/serial-linux.html) Archived (https://web.archive.org/web/20180702014758/http://training-kits.appspot.com/seri al-linux.html) 2018-07-02 at the Wayback Machine