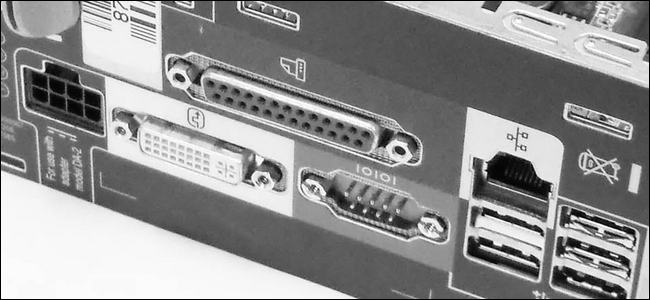
[**Why Is Serial Data Transmission Faster Than Parallel Data Transmission?**](http://www.howtogeek.com/171947/why-is-serial-data-transmission-faster-than-parallel-data-transmission/)



SATA hard drive connections are faster than older PATA hard drive connections and the same can be said for external cabling standards, but this is counter-intuitive: why wouldn’t the parallel transmission be faster?

*Today’s Question & Answer session comes to us courtesy of SuperUser—a subdivision of Stack Exchange, a community-driven grouping of Q&A web sites.*

**The Question**

SuperUser reader Modest is curious about the data transfer rates of parallel and serial connections:

Intuitively, you would think that parallel data transmission should be faster than serial data transmission; in parallel you are transferring many bits at the same time, whereas in serial you are doing one bit at a time.

So what makes SATA interfaces faster than PATA, PCI-e devices faster than PCI, and serial ports faster than parallel?

While it’s easy to fall into the reasoning that SATA is newer than PATA, there must be a more concrete mechanism at work than just age.

**The Answer**

SuperUser contributor Mpy offers some insight into the nature of the transmission types:

You cannot formulate it this way.

**Serial transmission is *slower* than parallel transmission given the *same signal frequency*.** With a parallel transmission you can transfer one word per cycle (e.g. 1 byte = 8 bits) but with a serial transmission only a fraction of it (e.g. 1 bit).

The reason modern devices use serial transmission is the following:

* You cannot increase the signal frequency for a parallel transmission without limit, because, by design, all signals from the transmitter need to arrive at the receiver at *the same time*. This cannot be guaranteed for high frequencies, as you cannot guarantee that the **signal transit time** is equal for all signal lines (think of different paths on the mainboard). The higher the frequency, the more tiny differences matter. Hence the receiver has to wait until all signal lines are settled — obviously, waiting lowers the transfer rate.
* Another good point (from [this post](http://only-vlsi.blogspot.de/2008/04/parallel-vs-serial-data-transmssion.html)) is that one needs to consider **crosstalk** with parallel signal lines. The higher the frequency, the more pronounced crosstalk gets and with it the higher the probability of a corrupted word and the need to retransmit it. [1]

**So, even if you transfer less data per cycle with a serial transmission, you can go to much higher frequencies which results in a higher net transfer rate.**

[1] This also explains why [UDMA-Cables](http://en.wikipedia.org/wiki/File%3aPATA-cable.jpg) (Parallel ATA with increased transfer speed) had twice as many wires as pins. Every second wire was grounded to reduce crosstalk.

Scott Chamberlain echoes Myp’s answer and expands upon the economics of design:

The problem is synchronization.

When you send in parallel you must measure all of the lines at the exact same moment, as you go faster the size of the window for that moment gets smaller and smaller, eventually it can get so small that some of the wires may still be stabilizing while others are finished before you ran out of time.

By sending in serial you no longer need to worry about all of the lines stabilizing, just one line. And it is more cost efficient to make one line stabilize 10 times faster than to add 10 lines at the same speed.

Some things like PCI Express do the best of both worlds, they do a parallel set of serial connections (the 16x port on your motherboard has 16 serial connections). By doing that each line does not need to be in perfect sync with the other lines, just as long as the controller at the other end can reorder the “packets” of data as they come in using the correct order.

The [How Stuff Works page for PCI-Express](http://computer.howstuffworks.com/pci-express.htm) does a very good explination in depth on how PCI Express in serial can be faster than PCI or PCI-X in parallel.

**TL;DR Version:** It is easier to make a single connection go 16 times faster than 8 connections go 2 times faster once you get to very high frequencies.

In a digital communications system, there are 2 methods for data transfer: parallel and serial. Parallel connections have multiple wires running parallel to each other (hence the name), and can transmit data on all the wires simultaneously. Serial, on the other hand, uses a single wire to transfer the data bits one at a time.

Contents

1 Parallel Data

2 Serial Data

3 Which is Better?

4 UART, USART

Parallel Data

The parallel port on modern computer systems is an example of a parallel communications connection. The parallel port has 8 data wires, and a large series of ground wires and control wires. IDE harddisk connectors and PCI expansion ports are another good example of parallel connections in a computer system.

Serial Data

The serial port on modern computers is a good example of serial communications. Serial ports have either a single data wire, or a single differential pair, and the remainder of the wires are either ground or control signals. USB, FireWire, SATA and PCI Express are good examples of other serial communications standards in modern computers.

Which is Better?

It is a natural question to ask which one of the two transmission methods is better. At first glance, it would seem that parallel ports should be able to send data much faster than serial ports. Let's say we have a parallel connection with 8 data wires, and a serial connection with a single data wire. Simple arithmetic seems to show that the parallel system can transmit 8 times as fast as the serial system.

However, parallel ports suffer extremely from inter-symbol interference (ISI) and noise, and therefore the data can be corrupted over long distances. Also, because the wires in a parallel system have small amounts of capacitance and mutual inductance, the bandwidth of parallel wires is much lower than the bandwidth of serial wires. We all know by now that an increased bandwidth leads to a better bit rate. We also know that less noise in the channel means we can successfully transmit data reliably with a lower Signal-to-Noise Ratio, SNR.

If, however, we bump up the power in a serial connection by using a differential signal with 2 wires (one with a positive voltage, and one with a negative voltage), we can use the same amount of power, have twice the SNR, and reach an even higher bitrate without suffering the effects of noise. USB cables, for instance, use shielded, differential serial communications, and the USB 2.0 standard is capable of data transmission rates of 480Mbits/sec!

In addition, because of the increased potential for noise and interference, parallel wires need to be far shorter than serial wires. Consider the standard parallel port wire to connect the PC to a printer: those wires are between 3 and 4 feet long, and the longest commercially available is typically 25 feet. Now consider ethernet wires (which are serial, and typically unshielded twisted pair): they can be bought in lengths of 100 feet, and a 300 foot run is not uncommon!