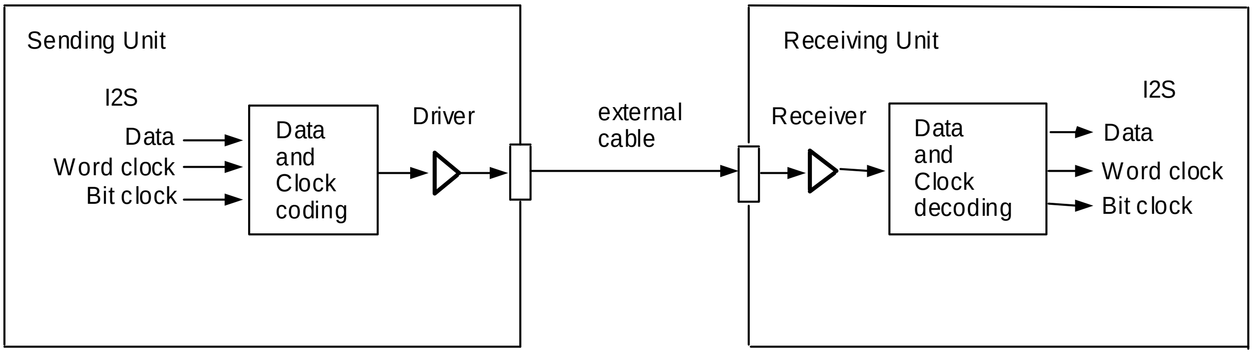
I am often asked "what is the best digital interface?" I have a long answer and a short answer: let's start with the long answer, which is just a discussion of digital interfaces.

Before we get to specific interfaces, it is important to understand two things.

First is that implementation is paramount. By this I mean that you cannot talk about a generic best interface – a well implemented SPDIF interface may function better than a poorly implemented USB interface and vice versa. We'll talk more about this later, but for now we will assume equally-well-implemented interfaces.

The second important thing to understand is that an interface is not just something that you plug a cable into – it is a complex system, incorporating circuitry in two devices and an interconnection cable. So it is meaningless to talk about the quality of one part of the system without taking into consideration the rest of the system. I guess that the exception to this is that the performance of the system is limited by its weakest link.



This picture shows a ubiquitous I2S internal interface on each end with the generic interface system in the middle. The Philips engineers developed the inter-IC sound (I2S) interface many years ago as a standard way to interconnect parts of a digital audio system within an enclosure. By that I mean no outside connection to another enclosure. It is a very simple interface with one data and two clock signals. The physical implementation is not defined, being left to the product engineer.

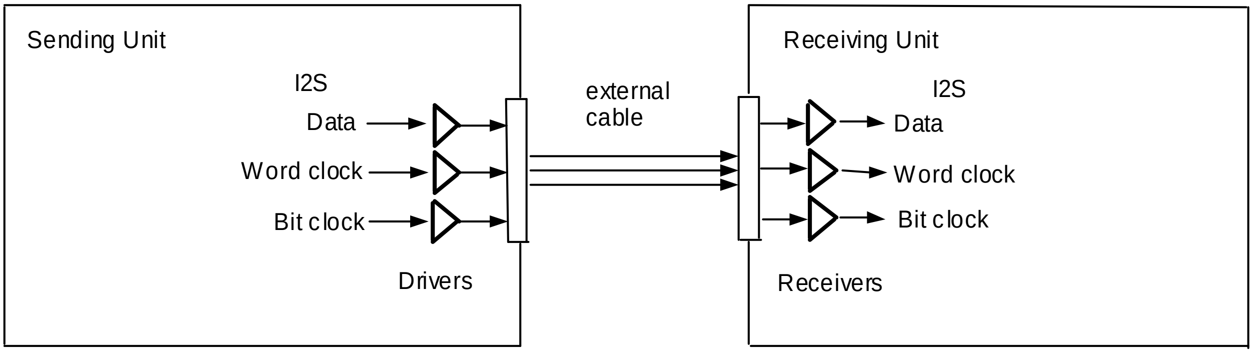
The requirements statement for internal interfaces is different than that for external interfaces. The cost for a couple or three or ten signals on an internal interface is miniscule, however multi-pin connectors are expensive so it is important to minimize cost by minimizing the number of pins on the external connector and it is worth adding some circuitry to do that. The engineers at Philips teamed up with the engineers from Sony and came up with the Sony Philips Digital InterFace (SPDIF) which uses bi-phase mark encoding to combine both the clock and the data onto a single pair of wires – you can't do any better than that for meeting the minimum pin-count requirement.

And of course, as Grace said, "The wonderful thing about standards is that there are so many to choose from." So we have the AES/EBU variations as well as TOSLINK to choose from. Each of these has pros and cons – in the end it all comes down to implementation. I think that the AES/EBU is the easiest to implement well, so perhaps it has an edge here.

Enter the computer, with all of its various interfaces – I am sure that someone has used each of these to transfer audio data – and if it is sufficiently unique and you are the only one that offers it, then you have an opportunity to hype it up and claim that it has some magic qualities because, as Arthur said, "Any sufficiently advanced technology is indistinguishable from magic." In some cases the interface used is application driven, for example, Ethernet may be the interface of choice for networked audio – something entirely different from standard point-to-point audio applications we are discussing here. The point is, you work with what you are given, not because it is better but is all you have.

Such is the case with the ubiquitous Universal Serial Bus (USB) – every computer has one so why not use it for digital audio? Not because it is inherently better, but because in some cases it is all we have. The USB encoding is much more complex than the SPDIF encoding, but then the quality comes down to implementation.

Which brings me to my short answer to the question “what is the best digital interface?” What if you dropped the requirement for a minimum pin-count on the external interface? What if you just sent the signals as they were across the interface? This idea has been around for a while and a few years ago PS Audio did just this in the I2S interface between their PerfectWave Transport and their PerfectWave DAC.



This simply side-steps all of the implementation problems associated with coding the data and clock on one end and retrieving the data and clock on the other end. I know that I haven’t talked about what the problems are and ways to best implement each of the interfaces, but then, that isn’t the topic of this discussion.

Since PS Audio released their I2S interface, many other manufacturers, including RAKK dac (more on the RAKK dac I2S interface,) got on the band wagon and now support the I2S interface, which in my assessment is the best digital interface.

Of course this is moot if the device that you want to attach to doesn’t support the I2S interface. Perhaps then, rather than “what is the best digital interface?” a more appropriate question would be “what device do I want to attach?’ and then “what interfaces does that device support?”