USB 2.0 – It Just Works…

by Jeff Chang - Marketing Manager, Personal Communications Division
Cypress Semiconductor

Let's rewind. In 1995, a fairly unknown specification called the Universal Serial Bus (or USB) began circulating among a handful of companies. With it came the promise of a truly easy-to-use universal interface for the PC. Since then, USB has emerged as the most successful interface in the history of PCs. From the user perspective, the benefits of USB are obvious: universal plug-and-play and revolutionary ease-of-use. When a USB peripheral is plugged-in to a USB port on the PC, the system will auto-detect and auto-configure the device. In most cases, there is zero user intervention required. The days of opening up the PC are gone. Furthermore, USB eliminates the need for multiple I/O standards, thereby simplifying PC connectivity for the consumer (as well as simplifying manufacturing for the PC OEMs.)

The original USB spec has evolved over time to meet the needs of the industry, giving us the two flavors which are available today:

USB 1.1
The architects of the original USB specification had a single focus – make computing easy for everyone. It's safe to say that they were very successful in achieving this goal. USB was the perfect interface for simple PC peripherals – mice, keyboards, joysticks, and the occasional mass storage device. However, USB 1.1 had one drawback – lack of bandwidth. With a top speed of 12 Mbit/s, bandwidth sharing has long been a major issue with USB. For example, when a USB video camera and USB scanner were run simultaneously on the same PC, the system performance would become noticeably sluggish to the end user.

USB 2.0
In 1999, a Promoter Group was formed to address the bandwidth issue. Within 1 year, a final specification was released: USB 2.0. The USB specification was updated to improve the performance and usability of PC peripherals, opening the door to a world of high-performance/high-bandwidth applications such as mass storage, digital video, and broadband access…and all can run simultaneously! The speed of USB 2.0 has been increased to 480 Mbit/s, a 40x improvement over its predecessor. This improvement in the USB specification will ensure its place as the de-facto standard for PC connectivity.

It Just Works…
Perhaps the most important feature of USB 2.0 is the fact that it is both forward and backward compatible with existing USB 1.1 devices and systems. To the consumer, the connectors, software, cables and other external product aspects are all the same. Existing USB 1.1 peripherals will continue to work with PCs equipped with USB 2.0, and vice versa. So even for the novice computer user who does not know the differences between USB 2.0 and USB 1.1, the system will still work. The migration is seamless.

So how does it work?
USB uses a master/slave (or host/peripheral) protocol. The PC (or host) is in charge of all communication on the bus. Peripherals connected to the PC do not 'speak' unless they are spoken to. This type of architecture leaves much of the system intelligence on the PC, allowing the USB peripherals to be designed and implemented in a very cost-effective manner – a very important feature for the cost-sensitive PC market.

At the physical level, USB is a 4-wire serial interface – Power, Ground, and 2 data lines for differential signaling. There are three speeds supported by the USB specification:

- Low-Speed – 1.5 Mbit/s
- Full-Speed – 12 Mbit/s
- Hi-Speed – 480 Mbit/s

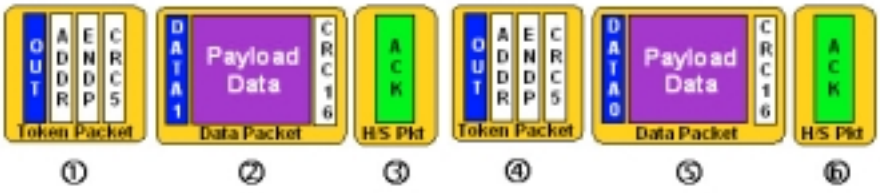
Low-Speed and Full-Speed (also called 'Classic Mode') were defined in the original USB 1.1 specification, while Hi-Speed signaling was introduced with the release of USB 2.0. Classic Mode electrical signaling occurs at 3.3 V, whereas Hi-Speed occurs at 400 mV. Having three different speeds allows developers the flexibility to choose between multiple price/performance levels when implementing their design.

A Few Important Concepts
When a device is first plugged into a USB host system, an auto-configuration process is initiated known as ***enumeration***. The device itself will have a unique ***Vendor ID (VID)***, ***Product ID (PID)***, and ***Device ID (DID)***, as well as some other custom configuration settings. The host will bind the device to a driver (depending on the configuration IDs) which is typically already resident on the host system (meaning no user intervention required.) The host will also assign a unique device address, so that there is no confusion amongst other peripherals on the bus (up to 127 devices can be connected to a single host.) The host will communicate with devices on the bus through addressable ***endpoints***. Endpoints are logical data units in memory that reside within the device architecture. The default endpoint (endpoint 0) is bi-directional, and typically only used for configuration communication. Other data endpoints may be set-up as required for a specific application, but these are unidirectional only (send or receive.)

Prior to being configured, a device is only allowed to draw 100 mA from the bus. During the enumeration process, the device may request a higher power mode (up to 500 mA max.) Depending on whether or not the requested power is available from the host, the device may or may not be allowed to enumerate. These power constraints will have an impact on a designer's decision to create a ***bus-powered***, ***self-powered***, or ***hybrid-powered*** application.

A typical USB transaction has three phases:

- Token Packet
- Data Packet
- Handshake Packet



- Packet ① is an OUT token
  - Signifies that data from the host is about to be transmitted over the bus (to a specific endpoint of a specific device)
- Packet ② contains the data
- Packet ③ is a handshake packet
  - Signifies to the host that the device received the data error-free
- The process is repeated in ④ through ⑥

Fig. 1: A Graphical Representation Of A Typical Transaction

The packets consist of encoded bit fields, and devices will respond to packets only addressed to them. The host also sends out a ***Start-Of-Frame (SOF)*** packet every 1 ms (in classic mode) or a micro-SOF every 125 µs (in hi-speed mode) in order to keep the entire bus synchronized. The host will access multiple devices within a single (micro)frame in order to optimize bandwidth allocation.

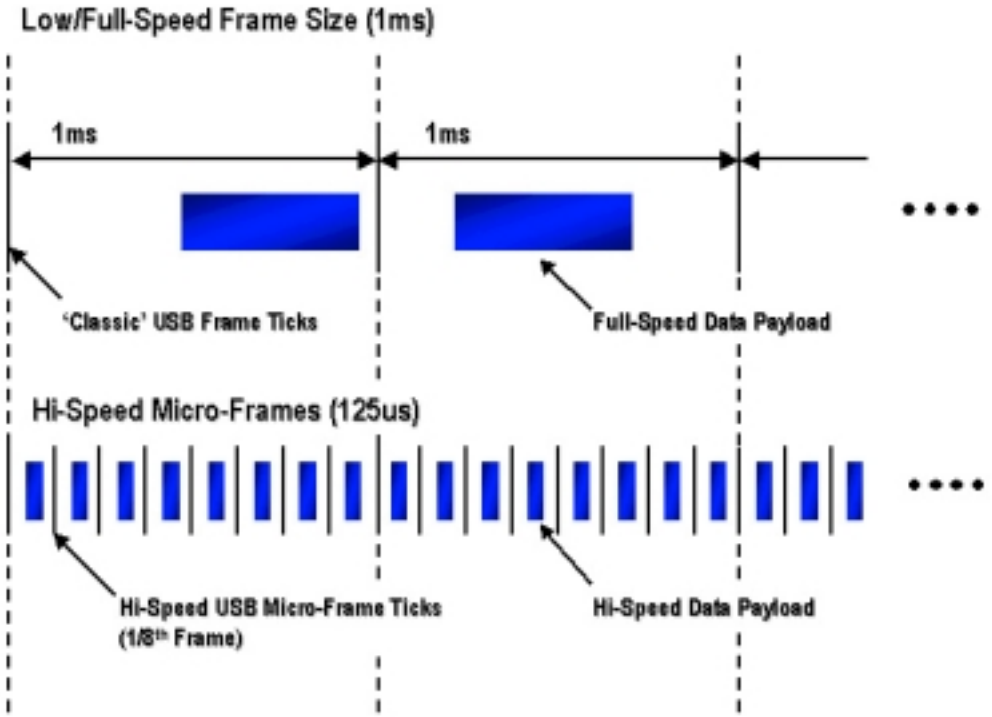


Fig. 2: The Bus Timing For Classic And Hi-Speed Modes

The USB specification supports four different data transfer types:

- ***Control*** – bursty transactions typically used for configuration when a device is first attached to the bus.
- ***Interrupt*** – polled scheme, used for timely, reliable delivery of data (mainly used in low-speed Human Interface Device applications.)
- ***Bulk*** – large, bursty transactions that provide guaranteed data integrity; reliable exchange of data is ensured at the hardware level by using error detection in hardware and invoking a limited number of retries in hardware (sample application – mass storage.)
- ***Isochronous*** – streaming real-time transfers; guaranteed timing, but does not support retries (sample application – web video camera.)

In order to minimize the impact of forward/backward compatibility requirements, USB 2.0 introduced extensions to the existing protocol, and a new hardware component for hubs – the ***Transaction Translator (TT)***. Transaction Translators isolate and buffer transactions to/from full and low-speed devices connected directly to a hub. Between the host and the hub (both USB 2.0 enabled), the transactions are transmitted at 480 Mbit/s, so the hub's buffering of the slower speed transactions virtually eliminates the overhead involved with supporting the slower transaction speeds.

USB 2.0 Silicon Design Challenges

Although the architects of the USB specification went to great lengths to ensure a "positive user experience", the changes introduced with USB 2.0 pose some difficult design challenges for peripheral silicon providers. The complexity of designing analog and digital blocks that are capable of keeping up with USB's top speed of 480 Mbit/s increases greatly over 'classic' implementations. The additional analog design considerations involved with USB 2.0 design can include:

- 480 MHz Phase-Locked Loop (PLL)
- 480 MHz Clock Recovery
- Squelch Circuitry
- Current Mode Drivers
- Precision Output Resistance
- Receive and Transmit Eye Pattern Characteristics

A compliant USB 2.0 transceiver must meet stringent electrical criteria.

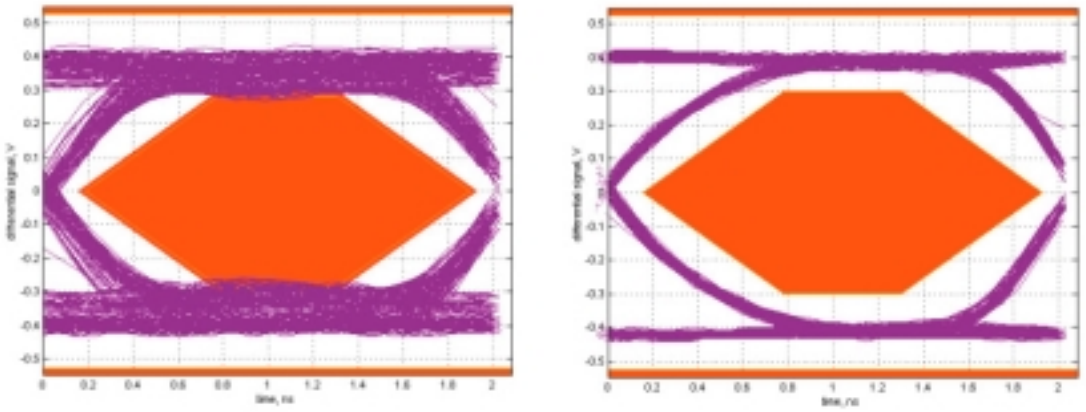


Fig. 3: Electrical Eye Patterns Of Two Different Transceivers.

The signals must stay outside of the red area in order for a device to be compliant. As one can see, the eye pattern on the left has been generated from a non-compliant device, while the one on the right has been generated from one that was compliant.

Taking into account the digital aspect, novel design approaches are required in order to move data on and off the chip at the much higher speeds. Digital design issues to consider include:

- High-speed control and decode logic
- High-speed data buffering for larger packet sizes
- Serializing/deserializing of data
- Forward/backward compatibility between hi-speed and classic modes
- Extensive test modes

Perhaps the greatest design challenge for USB 2.0 silicon building block providers is implementing both the analog and digital portions into a complete solution. Timing and power constraints play a large role in the full system simulation/validation plan.

USB 2.0 Silicon Implementation Options

When developing a hi-speed USB peripheral device, designers must first determine a suitable approach dependent on the desired level of integration. From a high-level perspective, a USB application can be broken down into four main functional blocks (see Fig. 4):

- USB 2.0 Transceiver
- Serial Interface Engine (SIE)
- Protocol Controller
- Peripheral Application

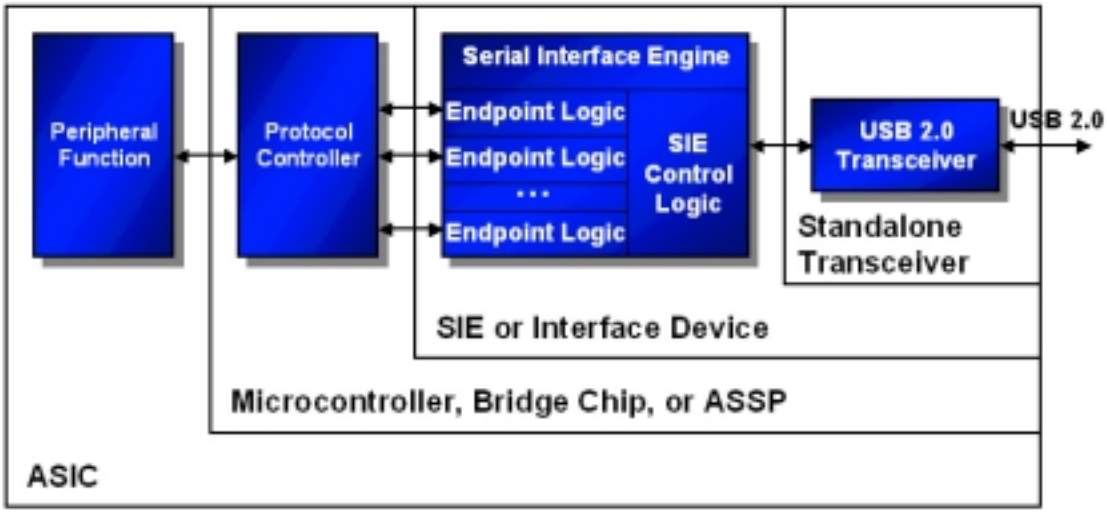


Fig.4: Functional Blocks In USB 2.0 Solution

Four design options present themselves (Fig. 4, again), each with distinct characteristics that must be considered when determining the 'best fit' approach to producing an end product:

- Standalone Transceiver – external components required for peripheral function. This design approach is attractive when an ASIC is involved, but the designer does not want to risk integration of the USB 2.0 transceiver (or the process technology is not compatible to the analog macrocell.) USB protocol also needs to be handled externally (developed in-house or bought.) This approach may also be attractive for prototyping with an FPGA or CPLD.
- SIE or Interface Device – low-level protocol is handled by this device, but external intelligence will need to handle higher level USB transactions. The USB 2.0 transceiver is also integrated in this device. This design approach is attractive when an external microprocessor or DSP is already in the design, and firmware can be modified to handle USB traffic.
- Microcontroller, Bridge Chip, or ASSP – all USB transactions are handled by this device. Depending on the application, this device may also be a single-chip solution for the entire design. This design approach is attractive when the peripheral designer does not want to learn the nuances of USB protocol.
- ASIC – highest level of integration. USB design expertise must be in-house (or bought.) Provides the lowest cost solution.

USB 2.0 Design Considerations

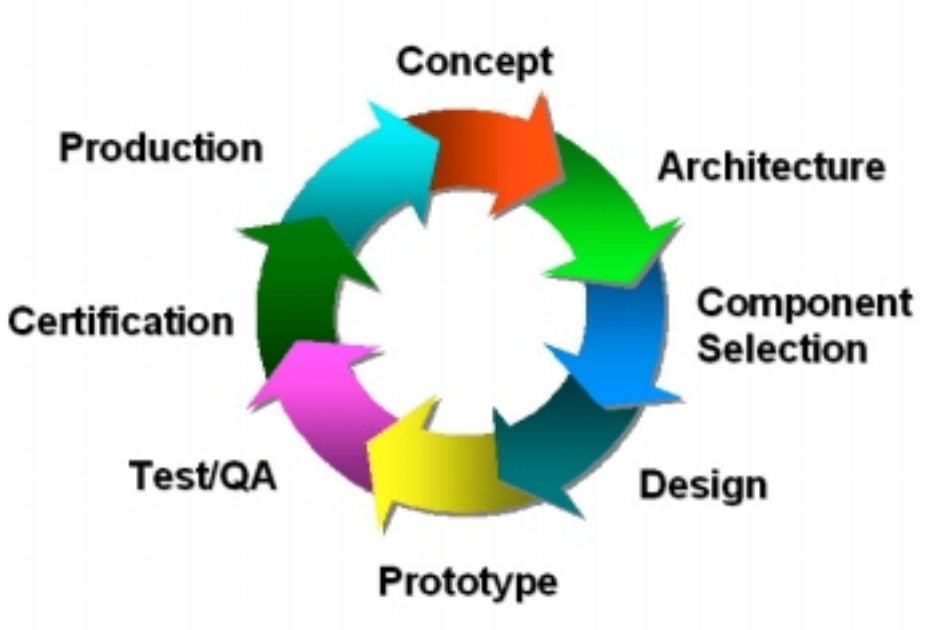


Fig. 5: Design Cycle For A USB Product

A typical USB 2.0 design cycle will follow the stages depicted in Fig. 5. When designing a USB 2.0 end product, there are many considerations that one must take into account. These issues tend to be common across all types of designs, although the means by which they are dealt with may vary from product to product. The issues can be segmented into 4 different categories:

- System Level Considerations
- Software Considerations
- Hardware Considerations
- USB-Compliance Considerations

System Level Considerations

System level issues (Table below) tend to have a heavy influence on the product development cycle during the Concept, Architecture, and Component Selection stages. These are issues that have a 'global' impact to the overall design.

Table 1: System Level Issues

System Issue	Questions to Ask	Design Impact	Real-World Example
Speed/Bandwidth	"What are my performance goals?" "Do I need to 'touch' the data?"	Device I/O – real-world interface needs to be faster than performance goals Processing Power Memory Architecture – FIFOs must be adequate to buffer data efficiently	Hi-Speed Video – requires high-performance core for image processing, and large buffers for streaming video endpoints

Power	"Bus-powered, self-powered, or hybrid-powered?"	Component Selection – silicon must meet USB power requirements Operating Voltage(s) Current budgeting for LDO & switch-mode regulators, passive components, electro-mechanicals	Flash Card Readers – must be bus-powered, so low-power components are required
Board Space	"Is my product under any form factor constraints?"	Device packaging, pin-count Total component count	NAND Flash Drive ("Thumb" Drive) – height and area of devices used are critical
Cabling	"Do I need to ship my product with a cable?" "Is the cable tethered or removable?"	Bill-of-Materials (BOM) – cables increase cost Risk – always use certified cables; extensive testing has been conducted	Retail cables – certified cables (identified with a USB logo) are guaranteed to work.
Data Transfer Types	"Do I need to stream data or just make sure it gets there reliably?"	Component Selection – device and endpoints must support appropriate data transfer types	Hard Disk Drive – must use bulk transfers due to data integrity issues
Cost	"What will the consumer be willing to pay for my product?"	Everything adds cost – total BOM, Manufacturing, Test, Q/A, Software, Packaging, etc… Cost of ownership – support costs are inevitable	The PC peripheral market is extremely cost-sensitive (regardless of the application). Retail/distributor mark-up drives margins even further down.
Heat Dissipation	"Does heat affect my performance and/or reliability?"	Board Layout/Airflow Component Selection – lower power devices not as susceptible to heat issues	Video – imaging components (CMOS arrays) degrade in performance at higher temperatures
ESD/EFTB/EMI	"Do I need to pass agency testing?"	Component Selection and Board Layout have the largest influence over these tests	Shielding may be required to suppress EMI as found in many applications today (even keyboards!)

Testability	"Can I test my product reliably and cost effectively?"	Vendor selection – some vendors offer manufacturing utilities System design – special 'hooks' for test may be required	Many OEMs have had to create custom test fixtures
Design Re-Use	"Is this design a platform?"	Component selection – programmable solutions offer greater flexibility Board design – one PCB for multiple products reduces inventory cost	A family of products from a single company (printers for example) tend to have a common design with various 'stuffing' options
User Experience (Human Factors)	"How good does it look?" "How easy is it to use?" "How loud is it?"	Industrial design	Apple iMac peripherals tend to have the same look and feel

Software Considerations

Software related issues will influence the design early as well, but mainly during the Design and Prototype phases.

Table 2: Software Considerations

Software Component	**Questions to Ask**	**Design Impact**	**Real-World Example**
Host Driver Stack	"Does the target Operating System have native drivers?"	If not native, drivers need to be developed in-house or bought/licensed from a 3rd party	USB 2.0 host drivers are available from Microsoft for Windows XP, but not Windows 98
Device Drivers	"Does my product conform to an established USB device class?"	If not, drivers need to be developed in-house or bought/licensed from a 3rd party	HID-class drivers are native to Windows, and support standard mice and keyboards
Peripheral Firmware	"Is my design programmable?"	Firmware development may be needed for programmable platforms	Many designs can be changed/upgraded without changing the hardware
Application Software	"Do I need to ship software with my product?"	Software adds value to the product. Development costs may be large.	CD publishing software often ships with CD-R/W drives

Hardware Considerations
As mentioned previously, there are various silicon architecture options that designers can choose from. Each of these architectures address a number of product design issues in different ways. Here is a comparison between the different design options.

Table 3: Hardware Considerations

Product Design Issue	Standalone Transceiver	SIE or Interface Device	Microcontroller, Bridge Chip, or ASSP	ASIC
USB Learning Curve				
Cost				
NRE Costs				
Time to Market				
Board Space				
Power				
Flexibility				

Key:
Best-Fit Solution (i.e. ASSPs have a low USB learning curve associated with them)
Medium-Fit Solution
Poor-Fit Solution
Worst-Fit Solution

USB Compliance Considerations
All USB products that wish to display the USB logo are required by the USB Implementers' Forum to pass all compliance tests. Compliance testing typically occurs at a USB-IF sponsored "PlugFest" (which are regularly scheduled once per quarter) or at certified 3rd party test labs. There are three test categories in which the products are stressed:

- Electrical Tests – ensures that the electrical parameters are within USB specs.
- Software Tests – USBCV (Command Verifier) tests protocol compatibility of the device
- Interoperability Test (a.k.a. "Gold-Tree" Test) – ensures that the device works correctly in a real-world environment with known-good products

Prior to testing, USB checklists must be filled out and submitted. It is also recommended that pre-testing be conducted in-house. All software and test procedures can be downloaded from the USB-IF website (www.usb.org). Once a device passes all compliance tests, it will be added to the USB Integrators' List (also accessible on the USB website.) If a device does fail one of the compliance tests, a Request For Waiver (RFW) may be submitted to the USB Compliance Review Board, who has final say on all compliance related issues.

The User Experience

A good user experience is critical to the success of USB. USB not only needs to bring perceivable value to the end user, it needs to do so without hassle. The ultimate goal is for the PC consumer to recognize and buy USB systems and peripherals off their store shelves because of the “ease-of-use” connectivity that has become synonymous with them.

The USB Implementers’ Forum (or USB-IF) has gone to great lengths to ensure that this goal is met. As mentioned before, new stricter testing methods and certification procedures have been introduced to ensure seamless interoperability. A new logo has also been introduced that will signify to users that a specific device has passed these very strict compliance tests:

Fig. 6: New USB 2.0 Logo

All USB-enabled systems and peripherals must pass the compliance tests before being allowed to use the new logo, and the manufacturer is required to sign a licensing agreement with the USB-IF. A red banner has also been added to signify whether or not the USB product is Hi-Speed capable.

Performance, Performance, Performance

So what can the user expect? The Table below shows a performance comparison of a typical PC peripheral (CD-R/W drive) with various interface options. As you can see, the end-user will see significant performance improvements with USB 2.0 over USB 1.1, and even over the native IDE connection.

Table 4: Time Comparison for Reading and Writing an Audio CD

Connection	**'Rip' an Audio CD**	**Write an Audio CD**
USB 2.0 (Hi-Speed)	3 minutes 15 seconds	4 minutes 20 seconds
USB 1.1 (Full-Speed)	15 minutes	24 minutes
IDE	6 minutes 40 seconds	6 minutes 10 seconds

Note: Benchmark data taken on a 1.3-GHz system with 128 Mbyte of RAM. CD-R/W manufactured by TDK

Likewise, the following Table shows the performance improvements of a hi-speed USB 2.0 hard drive.

Table 5: Performance Comparison for Reading From and Writing To a Hard Disk Drive

Connection	**Read (Mbyte/s)**	**Write (Mbyte/s)**
USB 2.0 (Hi-Speed)	28.0	17.6
USB 1.1 (Full-Speed)	0.98	0.86

Note: Benchmark data taken on 1.7-GHz system with 128 Mbyte of RAM using HD Tach 2.61. HDD manufactured by IBM

Other performance benchmarks that have been reported by the USB-IF:

Table 6: USB Scanner Performance Comparison

Connection	**Single-Page Scan Time**
USB 2.0 (Hi-Speed)	1 minute 45 seconds
USB 1.1 (Full-Speed)	31 seconds

Note: Elapsed time to digitize an 8.5x11 in. page, 4.56 Mbit file at 600 dpi

Table 7: USB Video Performance Comparison

Connection	Frames Per Second
USB 2.0 (Hi-Speed)	30
USB 1.1 (Full-Speed)	20

Note: XGA resolutions

State of the Market
The success of the market is dependent on four pieces to a puzzle – Host Hardware, Host Software, Peripheral Hardware, and Technology Infrastructure:

With full support available in each category today, USB 2.0 is slated to continue its dominance as the PC interface of choice. While some would argue that USB 2.0 competes directly with another interface technology, 1394 (FireWire), we do not agree. While the volume of USB 2.0 peripherals will clearly surpass that of 1394 peripherals in PC-centric applications, these standards are different enough that both will retain their own niche. USB 2.0 as a master/slave architecture has some inherent advantages over 1394, and likewise 1394 over USB 2.0 as a peer-to-peer architecture. USB 2.0 will emerge as the de-facto standard for PC peripheral connectivity, while 1394 will emerge as the interface of choice for consumer electronics. True, there will be some crossover in a few high-end markets, but the majority of the business for each respective standard will weigh heavily on one side or the other of the digital divide.

So What's Next? Will There Be A USB 3.0?
USB 2.0 has been successful in its goal – to provide an easy-to-use interface for the PC, while providing high performance for bandwidth hungry applications. Although there have been some side discussions regarding a USB 3.0, the market has yet to identify a real need for an evolution of the spec.

There has been, however, work done on a recently released supplement to the USB 2.0 specification:

- USB On-The-Go (OTG). In simple terms, this supplement allows for peripheral-to-peripheral communication without the need for a host PC as an intermediary. This is accomplished by defining a new class of devices that have limited host capability. OTG is still a very new concept, and products are not expected to hit the mass market

until late 2003. There is, however, a lot of industry effort being conducted today to roll-out the infrastructure to support this new technology.

For more information about USB and USB On-The-Go, please visit the official USB Implementers' Forum website at www.usb.org.