MICROPROCESSOR REPORT

THE INSIDERS' GUIDE TO MICROPROCESSOR HARDWARE

VOLUME 9 NUMBER 5 APRIL 17, 1995

Universal Serial Bus to Simplify PC I/O

New Interface to Service Keyboards, Mice, Telecom, and More

by Michael Slater

Following in the footsteps of the successful revamping of the PC bus architecture with PCI, a group led by Intel and Microsoft is now tackling the external I/O architecture with a new peripheral bus. The 12-Mbps Universal Serial Bus (USB) is designed to provide a single interface to connect both user I/O devices, such as keyboards and mice, and telecommunications devices, such as modems and PBX interfaces.

USB can be used for analog peripherals, such as speakers and microphones, by including A/D or D/A converters in those devices. The data rate is sufficient for MPEG-2 video and CD-ROM interfaces. The availability of a standard, low-cost, multidevice connection could spur a proliferation of peripherals, from joysticks and digitizers to data gloves.

Computer-telephony integration is expected to be a major growth area for PCs, and USB will facilitate its development by providing a standard, low-cost interface that can be used not only for modems but also for ISDN and PBX interfaces. PCs have been handicapped by conventional serial ports, which provide inadequate bandwidth for modern telecom applications, are notoriously difficult to configure, can't connect to more than one device at a time, and don't provide power.

Deep Backing from Intel and Others

USB is not the first bus to tackle the peripheral attachment problem, but its technical capabilities and, even more important, its backing make it destined to dominate in x86 PCs—though it won't become pervasive until 1997. Driven by the dominance of the Intel/Microsoft PC standard, USB is likely to have a particularly chilling effect on efforts to widen PC industry support for Apple's GeoPort and Philips' Access.bus.

Intel has promised to integrate a USB interface into all its system-logic chip sets, with the first to be delivered by the end of the year, and also to provide embedded controllers with USB interfaces for peripheral devices. So far, Digital Semiconductor is the only other chip maker to announce plans for USB interface chips.

Microsoft will provide USB drivers and other software support for Windows 95 (though not as part of the initial release) and Windows NT, and IBM will provide support in OS/2. Microsoft is also the world's leading supplier of mice, and it will build USB versions of its mice and keyboards. Other peripheral vendors are likely to follow suit, but none has announced plans.

Other members of the initial development group are PC makers Compaq, Digital, IBM PC Co., and NEC, plus telephony vendors Northern Telecom and NEC. The group was kept small to keep the specification development process moving quickly, and this restriction limited the number of supporters ready to announce their plans at the bus's formal debut during Microsoft's WinHEC event. Given Intel's strong role in system-logic chip sets, as well as its dominance as a supplier of Pentium motherboards, the appearance of USB ports on a wide range of PCs is all but guaranteed.

The specification is still in development, with some details remaining to be resolved, but a great deal of work has been completed. The 0.9 version of the specification has just been released, and it is available on request *(see **090501.PDF**)*. After collecting industry feedback, the USB group plans to deliver version 1.0 in June, enabling implementations to begin. The bus's promoters expect systems with USB ports to debut at fall Comdex and begin shipping in the first quarter of 1996.

Low-Speed Subchannel Cuts Peripheral Cost

Early USB design work assumed a data rate of 5 Mbps, representing a compromise between cost and performance. Even at this speed, however, the cost of shielding for FCC approval and signal quality is significant for a low-cost device like a mouse. At the same time, some high-end applications—notably MPEG-2 video and other CD-ROM software—would benefit from a higher data rate.

To solve both problems, the USB specification was changed to raise the basic data rate to 12 Mbps while adding support for a low-speed subchannel running at approximately 1 Mbps. Devices running at the slow speed can use unshielded wire that need not be a twisted pair (twists cost about 4 cents per foot!) and can also use cheaper semiconductor technology for the interface chip. Low-speed devices can also use single-layer PC boards without requiring shielded enclosures. Total cost savings per device for using the slower rate is estimated to be $1–$2, which is a lot for a mouse.

Tiered Star Topology Supports 63 Devices

Figure 1 shows a typical USB configuration. Logically, USB is a bus, but physically, it is a tiered star—that is, a series of stars connected in series. It is a master/slave architecture, with one host and up to 63 peripheral devices.

Because each USB link is a point-to-point connection, a hub is required at each point where multiple connections are needed. PCs will integrate both the USB host and a hub to provide two or more USB connectors.

Peripheral devices may incorporate hubs to allow other peripherals to be connected to them. The figure shows an example in which the monitor and keyboard each include a hub.

Hubs act as repeaters, re-driving the signals in each direction and providing termination for each line. They do not process the data as it is passing through, so little intelligence is required. Hubs do include control and status registers that enable the host to enable or disable each port and to determine whether a device is connected to each port.

Power is distributed in the USB cable, so USB devices do not need their own power supplies. The host must supply 500 mA (at a nominal 5 V) on each port. Hubs can be either powered or unpowered; a hub in a monitor will typically be powered, while one in a keyboard typically will not. A powered hub provides the full 500 mA for each port, while an unpowered hub is limited to redistributing its single 500-mA feed.

Unpowered hubs cannot be tiered; only peripheral devices or powered hubs can be connected to an unpowered hub. The current plan is that only keyboards will be allowed to be unpowered hubs, largely eliminating the possibility of illegal configurations.

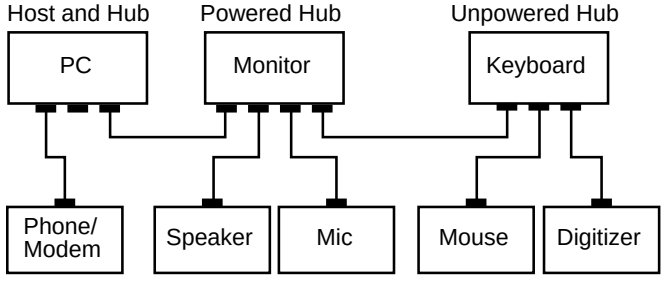

Figure 1. A typical USB configuration with hubs built into the monitor and keyboard.

Hubs Enable Geographic Configuration

Each hub has a status register that can be read by the host to determine which ports have devices connected. The hub does not need any intelligence to make this determination; the differential pair used for USB signaling is connected to a biased termination at the receiver, so the hub can tell if a device is connected simply by detecting the voltage level on each of the wires.

When the system is initialized, the host traverses the tiered star one level at a time, building a map of what devices are connected. The exact mechanism for handling the low-speed subchannel is not defined in the 0.9 specification, but the configuration mechanism will ensure that high-speed signals never appear on the cable of a low-speed device, eliminating the need for shielding.

Because the host can address each device geographically by enabling each port of each hub individually, unique addresses are not required in each device at startup, so no setup switches are needed. After querying each device to determine its type, the host sends a command to the device to set its logical address, which is used for all addressing after configuration is complete.

The host periodically polls the status registers of all hubs to look for newly attached (or detached) devices. The system supports hot attach and detach, so devices can be added or taken away at any time.

Physical Layer Uses Four Wires

USB cables consist of four wires: one pair for power and a second pair for data. USB distributes 5-V power, but because of voltage drops in the cables and connectors, a device may see as little as 4.4 V on the plug at the end of its cable, and even less by the time the voltage reaches the device's circuit board. The intent is that devices will use 3.3-V logic, so this scheme leaves room to regulate the voltage down.

Signalling is performed at 3.3-V CMOS levels, driven with an impedance of 45 ohms. The receiver must have a sensitivity of at least 200 mV. The system allows cables of up to 5 meters for every segment.

The clock and data are combined into a single half-duplex signal using NRZI encoding, which represents a zero bit with a transition. Bit stuffing (inserting a zero after a long string of ones) ensures that signal transitions are frequent enough for reliable clock recovery. Each USB device must include clock-recovery circuitry.

Protocols Support Isochronous Data

The USB hardware/software system provides com-

munication between the host and a peripheral through an abstraction called a pipe. When a device is configured, its software indicates its requirements for latency and bandwidth, and the system provides—if possible—a pipe meeting these specifications. It is possible, of course, to assemble a system whose bandwidth needs exceed the bus's capability, and in this case the software will refuse to establish one or more of the pipes.

Both asynchronous and isochronous pipes are supported. An isochronous pipe guarantees a continuous data stream; it is used for data types, such as audio or video, where the timing of the data is significant. Asynchronous pipes can be used to transfer either short messages, such as a set of mouse coordinates, or block data streams—such as for a printer—that want as much bandwidth as possible but can wait as needed. Isochronous pipes get priority, and asynchronous transfers get whatever bandwidth remains.

The host provides guaranteed bandwidth and latency by allocating each peripheral a given number of slots within a 1-ms frame, which is the basic unit of scheduling. This provides a maximum latency of 1 ms, which the designers believe is adequate for all target applications. USB is not truly isochronous, in that it maintains an average data rate but not a constant interval between samples. As a result, some buffering is required in peripherals. This buffer may be as small as one byte for a mouse, or as much as several hundred bytes for a high-speed device.

USB uses a polled master/slave protocol. The host initiates every communication with a three-byte "token," which includes the address of the target peripheral and the type of transaction. For a write to a peripheral, the host then sends a data packet, and the peripheral responds with an acknowledgment packet; for a read, the opposite occurs. Broadcast packets are supported, but peer-to-peer transfers are not.

Data packets include from 1 to 256 bytes of data, plus 3 bytes of protocol overhead. Each packet (including the token) includes a CRC for error detection. The single-byte acknowledgement handshake can be used to request retransmission or for flow control, depending on the needs of the device. For example, in an audio data stream, flow control will typically be used to manage the receiver's buffer, but retries on errors may be skipped to maintain isochrony. When a block of data is moved to a printer, on the other hand, retries would be used to ensure successful transmission of all the bits, even though transmission may take longer.

Because the polled protocol has some overhead, the full 12-Mbps bandwidth is not available for data transmission. USB's designers believe that 8-Mbps compressed video (such as an MPEG-2 stream) could be supported while still leaving enough bandwidth for low-speed peripherals. A 128-kbps ISDN adapter should be no problem, and even a T1-rate (1.5 Mbps) connection would use less than 15% of the bandwidth.

Hardware Implementations

USB is an asymmetrical system; the host is much more complex than the peripherals. Intel estimates that a USB peripheral interface requires about 1,500 gates—roughly equivalent to a UART. A five-port hub would add about another 500 gates. Because of the data rate and the need for clock extraction, however, the interface cannot be implemented in firmware using a standard I/O port (like a "bit banged" UART).

Intel plans to offer 8051-family microcontrollers with USB interfaces, and other microcontroller vendors are likely to do so as well. Intel plans to make available at little or no cost the logic design for a basic USB peripheral interface (but not the host interface), although the details of such an offering have not been worked out. So far, no makers of low-cost microcontrollers (used in mice) have announced plans for USB support.

The host interface must manage the bus and maintain the communication protocol, so it is much more complex. It may also require special logic to support emulation of the PC-standard keyboard interface at boot time, when OS software has not yet been loaded. Intel estimates that a host will require about 10,000 gates, including a five-port hub. This is a nontrivial amount of logic to add to a chip set, but Intel's leadership will make it hard for other vendors to leave it out. Standalone PCI-to-USB interface chips are also likely to become available.

The USB specification should provide sufficient information for other chip-set makers to implement the controller, but as in the early days of PCI, there may be concerns about the specification's completeness and accuracy. Intel's physical implementation is likely to set the standard, and other implementations may need to conform to Intel's initial implementation in areas where there may be gaps in the standard.

The USB standard is open and royalty-free, so it will not be tied to Intel or even x86 systems. All members of the initial USB group have agreed to enter into a patent nonassertion agreement designed to sidestep any patent claims; Microsoft, Intel, and Compaq have already signed the agreement. To benefit from this grant, companies building USB devices must similarly agree not to assert any patents that might apply to USB.

Implementation of USB in a PC requires a substantial amount of software as well. The USB driver model, following the approach used for PCMCIA interfaces, eliminates the need for a register-level standard. One software layer, called Host Controller Services (HCS), communicates directly with the controller hardware. The combination of the hardware and HCS must deliver a standard set of functions—a system-level programming interface—but the dividing line between what is in

	Data Rate	Host Complexity	Peripheral Complexity	Topology	Key Backers	Max. Cable Length	Connector Pins
USB	12 Mbps	10,000 gates	1,500–2,000 gates	Tiered star	Intel, Microsoft, Digital, Compaq, IBM PC Co., NEC, Northern Telecom	5 m per segment	4
Access.bus	100 kbps	Simple HW or SW UART	Simple HW or SW UART	Bus	Philips, Computer Access Technology Corp., Microchip	10 m total	4
GeoPort	2.048 Mbps	SCC USART	SCC USART	Point-to-point	Apple, IBM, AT&T, Siemens Rolm	1.2 m (10 m w/14-pin connector)	9 (today) or 14 (for longer cables)
1394 FireWire	100 Mbps (200 soon, 400 in future)	12,000–20,000 gates	5,000–7,000 gates	Tree	Apple, TI, NCR, Adaptec, Microsoft, IBM, National, Fuji, Philips, Sony	4.5 m per segment (longer versions planned)	6

Table 1. USB covers the performance range of GeoPort and Access.bus but stops well short of 1394 (FireWire).

hardware and what is in HCS is up to the implementer.

A second software layer, called Serial Bus Services (SBS), communicates with HCS—not with the hardware interface—and provides services to the OS, configuration software, and device-specific software. The SBS will be provided by the OS vendor.

After the initial release of Windows 95, Microsoft plans to provide USB drivers that presumably will support Intel's hardware implementation. As with graphics cards, Microsoft may provide USB drivers for the most popular controllers, but chip-set makers will probably have to invest in some software development to ensure support for their hardware (unless they maintain register-level compatibility with Intel's implementation). Note that the most complex parts of the USB software are in the SBS, which need not concern the hardware vendor.

Trouble for Other Serial Buses

At the top of the priority list for Intel and Microsoft has been that USB be ubiquitous, which is essential if it is to fulfill the goal of simplifying and enhancing the I/O system in mainstream PCs. This is a major reason why USB is designed to support both user-interface peripherals and telecommunications devices; USB's backers were concerned that a bus that serviced only one area would fail to reach critical mass. This is one factor that led to the rejection of existing standards—notably, GeoPort and Access.bus.

The steamroller effect of USB is likely to flatten efforts to establish two these buses as widespread PC standards. As Table 1 shows, USB overlaps the capabilities of both, and the ubiquity of USB will make it difficult for these alternatives to gain broad adoption in the PC market.

Backers of Access.bus, which evolved from Philips' I^2C chip-to-chip interface, have been trying for years to establish this 100-kbps bus as a mainstream alternative to the standard PC keyboard and mouse interfaces. Access.bus is very inexpensive to implement because of its sluggish data rate, so it will continue to be successful in some areas (such as VESA's Display Data Channel for CRT control and the System Management Bus for smart battery control). But USB appears to have ended its chances to become a mainstream PC interface.

GeoPort is Apple's 2-Mbps telecom interface. From a hardware point of view, there is little to it; Apple simply added a power pin to its existing RS-422 serial port. Apple has built a rich software environment around GeoPort, supported by the Macintosh Telephony Architecture and by Microsoft's TAPI for Windows. Apple's first GeoPort device is the Macintosh Telecom Adapter, which provides a telephone and phone-line interface with all modem processing performed in the host. GeoPort is supported on the Quadra 660 AV and 840 AV, which have an AT&T DSP3210 on the motherboard, and on all the Power Macs, which perform the DSP tasks on the host processor. (Note that despite Intel's high-profile promotion of native signal processing, Apple was actually the first to implement it.)

Through the Versit coalition, founded by Apple, AT&T, IBM, and Siemens Rolm, Apple hopes to establish the GeoPort interface as an industry-wide standard. Although GeoPort has a significant head start over USB, the likely ubiquity of USB in x86 PCs will make it hard for GeoPort to hold much ground there.

GeoPort is entirely focused on providing a low-cost isochronous data link for telecommunications devices. Though it satisfies this need very well, USB will subsume this capability and others. For Apple, which already had ADB as its solution for connecting low-speed peripherals, it made sense to focus GeoPort on telecom, but the PC needs a solution to both problems—and one bus is better than two if it can do the job.

GeoPort interface boards, in the form of PCMCIA or PCI cards, will be available for PCs, but it is unlikely that many PC makers will put a GeoPort interface on the motherboard. GeoPort should have at least a short-term advantage in the number of telecom devices avail-

able to connect to it; AT&T and Rolm are potentially powerful backers.

Indeed, GeoPort's survival in the non-Macintosh PC market is dependent upon there being more—or cheaper, or better—telecom devices with GeoPort interfaces than with USB interfaces. GeoPort devices may be less expensive, because the bus's low latency eliminates the need for buffering. A conventional USART can be used, eliminating the need for special silicon, as required by USB.

One serial bus that is unlikely to be affected much by USB is IEEE 1394, which Apple calls FireWire *(see **080304.PDF**)*. Running at 100 Mbps today with plans for scaling to 200 Mbps very soon and 400 Mbps in the future, 1394 is much faster but more expensive to implement than USB. The 1394 bus appears poised to dominate as the digital interface for consumer video electronics and could move into disk drives, but it is in a different market space than USB. The only point of overlap is compressed digital video; this is at the low end of 1394's range and at the high end of USB's capability.

Like GeoPort, 1394 is truly isochronous, eliminating the need for buffers in peripheral devices, and it does not use a polled protocol. The 1394 bus has a very different software model than USB; devices on the bus are mapped into the host memory space.

Opinions are divided on how low 1394 costs can go. Today, TI's interface chip sells for about $25. Some proponents believe this cost will fall to $5–$10 in high volume. Clearly the cost must not be too high, or consumer electronics makers would not be showing such strong interest. As with GeoPort, 1394 is more likely to be supported on an add-in card than on the motherboard.

Cleaning Up PC I/O

Today's PCs have several I/O ports that add to system cost, use back-panel space (especially precious on notebooks), and still fail to deliver either plug-and-play ease of use or the ability to connect to all devices a user might want. Many peripherals are most practical as add-in cards because of the lack of a suitable interface for connecting them outside the box. Add-in cards are often difficult to configure, however, and the need to open the box is a handicap.

Ultimately, USB promises to clean up this mess and encourage a shift from add-in boards to external add-ons. This is an important step on the PC's path to becoming a true consumer device. In the near term, however, the USB port will become yet another interface on the back of the PC. Until a wide range of USB devices is available, PC makers will be unwilling to drop the standard ports. Keyboard and mouse ports should be the first to disappear, probably in 1997. Serial and parallel ports ultimately can be replaced by USB, but their widespread use is likely to keep them around for a long time.

USB also ties in with Intel's Native Signal Processing initiative. One of the goals of NSP is to reduce the cost of communication peripherals, from modems to video capture devices, by cutting I/O device complexity to the bone and moving as much processing as practical to the host CPU. The USB structure will provide a low-cost vehicle for building NSP peripherals. Inexpensive ISDN and PBX interfaces should be among the first fruits of the USB architecture.

USB's backers clearly have grand ambitions for the bus, and it is tempting (particularly for competitors of the bus!) to discount it as all paper, with not even a complete specification—much less any actual device implementations—available. It is indeed possible, though unlikely, that significant problems will be found in turning the design into reality.

It would be a mistake, however, to underestimate the ability of USB's backers to drive this standard to dominance. Intel has enormous influence over the direction of the PC industry, not only because of its role as the leading microprocessor supplier but also because of its position as the top supplier of Pentium chip sets and motherboards. PCI demonstrated how rapidly Intel could push a new technology into the market; with Microsoft's backing as well, USB looks unstoppable. ♦

Obtaining the USB Spec

Version 0.9 of the USB specification is available now in either electronic or printed form. It can be accessed on the Web at *http://www.teleport.com/~USB*. Printed copies are available for $35 by calling 800.433.3652.

