

INTELLIGENT I/O:

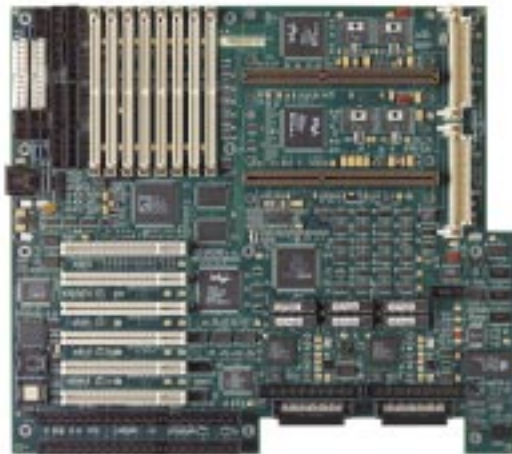
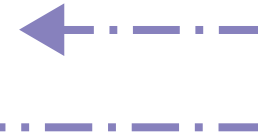


PHOTO COURTESY: ADAPTEC

DOES I₂O HOLD H₂O?



Maury Wright, Technical Editor

The Intelligent I/O (I₂O) initiative, which Intel launched publicly almost two years ago, has finally yielded products ready to ship. Only time will tell if these products deliver on the initiative's promises. Shepherded by the I₂O special interest group (SIG), www.i2osig.org, the initiative attempts to create an abstracted, hardware- and OS-independent I/O subsystem standard.

By adding an abstraction layer, the ini-

By offloading interrupt and bus-traffic

overhead caused by standard I/O operations,

intelligent I/O subsystems allow host CPUs to support more

users or transactions. Now, the I₂O initiative seeks to make

such subsystems both OS- and hardware-independent, leading

to widespread deployment.

tiative presumably allows multiple vendors to develop interoperable intelligent I/O products that solve traditional driver-incompatibility problems and lower the cost of deploying intelligent I/O subsystems. By moving away from systems that rely on the host CPU to control minute details of I/O operation, server designers can produce systems that make more efficient use of the CPU and serve more users. A review of I₂O concepts and available products should help you decide if the standard will deliver as promised and if it can help you design more efficiently, whether your application is a real-time data-acquisition system or a multiprocessing server.

The concept of intelligent I/O or distributed intelligence is decidedly simple as simple as relying on a dedicated μ C to handle PC keyboard activity. In fact, you can look throughout the computer industry and find examples of a type of processor that proves to be an ideal match for an application in power

consumption and computational resources relative to cost. For example, DSPs typically handle the modem-modulation function in PCs. Although a Pentium can today replace a DSP in the modem role, the DSP proves to be a superior modem vehicle when measuring implementation cost. PC designers must choose between using the Pentium CPU to handle user tasks and the modem function and supplementing the

CPU with a DSP leaving more Pentium cycles available for user tasks. Moreover, smart users will choose a system/modem combination that meets their needs. If users want a low-cost system and don't need to run computationally intensive applications, a system with a host-based modem may be the best option.

For decades, mainframe designers

have deployed specialized processors to handle data-storage and networking I/O tasks. I/O processors increased the overall system performance and proved cheaper than adding a more powerful CPU. Looking back 10 years to the 68020/68030 era in the VMEbus arena, you'll note a similar trend. Back then, all performance-sensitive systems used a dedicated processor on the LAN board to run the TCP/IP stack. Over time, the host CPU became more capable of handling TCP/IP software, and LAN ICs integrated functions that further simplified the task. In today's multiprocessor VMEbus systems, every CPU hosts TCP/IP, among many other tasks.

In the original PC, IBM placed many demands on the CPU because the target price didn't allow for more processors. For example, the CPU in the original IBM PC handled memory-refresh operations, thereby eliminating the need for a dedicated DRAM-controller IC. Today, high-end Pentium- and Pentium II-based PCs still rely on the host CPU to handle many I/O tasks, including control of LAN and data-storage interfaces. A typical ATA disk interface, for instance, has little intelligence and regularly interrupts the host CPU to transfer individual blocks of data.

A largely unintelligent I/O subsystem suffices in most PCs and workstations because one user typically has few active tasks. In some cases, the host

@ a glance

- The I₂O initiative seeks to improve system performance and to eliminate the need for custom device drivers that match every OS/ device combination.
- Manufacturers are now shipping I₂O building-block products to build a complete system.
- To succeed, I₂O must match the performance of other systems with intelligent I/O rather than simply improving traditional systems that lack intelligent-I/O subsystems.
- The I₂O spec affords flexibility for varying configurations, but that flexibility could undermine the initiative's success.

INTELLIGENT I/O

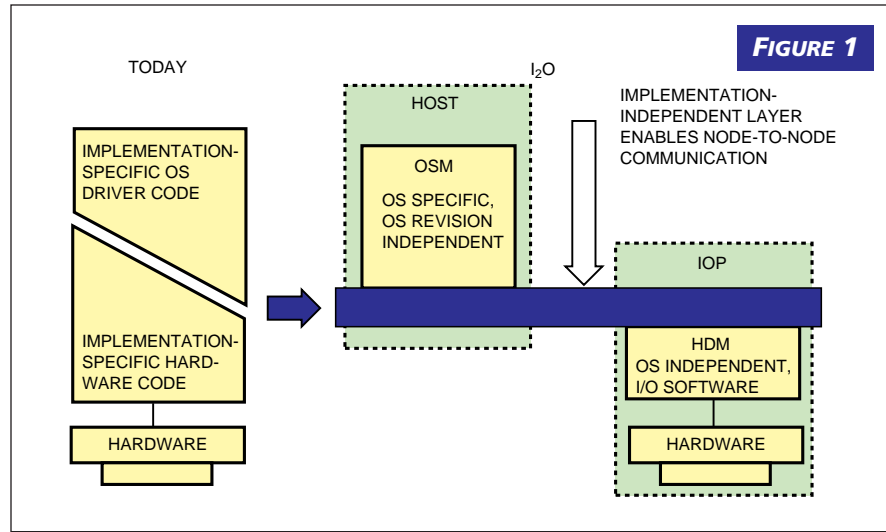
CPU has no other work during an I/O operation, so it may as well handle as much of the operation as possible. Even when the user runs multiple tasks, the percentage of cycles the CPU spends on I/O operations pales in comparison with the percentage spent on the application code.

In general-purpose computing, intelligent I/O proves to be far more attractive in servers that fulfill the needs of many users. Such systems include file and print servers but can also include application servers. Electronic designers, for example, regularly rely on high-end application servers to handle tasks such as large simulations that exceed desktop-system capabilities. Note that intelligent I/O can be equally applicable in some embedded systems. Flight and other types of simulators rely on a great deal of I/O. Real-time data-acquisition systems may do little more than capture data and store it in the data-storage subsystem (see box I₂O proves a match for some embedded systems.)

Intel clearly targeted the I₂O initiative at the server market. In reality, high-end servers from companies such as Sun (Mountain View, CA), Silicon Graphics (Mountain View, CA), and even Compaq (Houston) have long deployed μ P-based I/O subsystems. I₂O, however, could make intelligent I/O a reality even in the commodity PC-based servers that virtually every PC vendor sells, making the servers suitable for deployment in enterprisewide networks. Meanwhile, I₂O could deliver significant benefits, including lower price, shorter development time, and lower support cost even into the high-end server market.

I₂O mission statement

The I₂O initiative has two goals. First, it seeks to eliminate the drain on overall system performance that occurs when the main CPU handles low-level I/O interrupts. Second, it attempts to eliminate the necessity for system, motherboard, and add-in-card vendors to test and support unique drivers for every combination of I/O device and OS on the market. Revision 1.5 of the I₂O specification, published in March 1997, defines support for block-oriented devices, such as disk drives; streaming devices, such as tape drives; and network and SCSI devices.



I/O relies on a split-driver model to abstract the host OS from the device-specific details of the I/O subsystem.

Note that SCSI disk drives generally fall into the block-oriented device class, but the spec authors include a separate SCSI class because of the plethora of SCSI devices available.

To reduce I/O load on the CPU, I₂O offloads the responsibility for I/O processing to I/O processors (IOPs). The I₂O SIG documentation describes these IOPs as targeting the I/O tasks, but any μ P is eligible for the IOP designation; the documentation specifies no particular processor. In the I₂O environment, the host CPU can hand off an I/O request to the IOP, and the IOP doesn't interrupt the host CPU until it has completely fulfilled the request handling all low-level I/O-device interrupts in the process.

By necessity, any system with intelligent I/O must split the I/O driver into a code segment that runs on the host CPU under control of the host OS and a segment that runs on the IOP. The I₂O spec uses such a split-driver model (Figure 1) and defines formalized boundaries and communication interfaces that can yield interoperable OS- and hardware-independent I/O subsystems. Although the concept Figure 1 depicts appears straightforward, the I₂O spec is complex because the authors want to allow designers flexibility in how to implement intelligent I/O. The I₂O spec seeks to be useful in environments ranging from single-processor servers to multiprocessor servers to server clusters. Moreover, it seeks to support both data storage and

networking subsystems.

Message passing

In an I₂O environment, all host and I/O nodes must include transport-layer software with a messenger service and resource manager. The scheme uses a message-passing interface for communication between nodes, similar to the message-passing protocols Intel developed for Multibus II. On the host side, an OS services module (OSM) lies atop the messaging layer. OS vendors will develop the OSM and host messaging layers to translate host-OS-specific I/O calls to the I₂O message-based format. Microsoft (Redmond, WA), Novell (Provo, UT), and SCO (Santa Cruz, CA) will have I₂O service patches available for their Windows NT, Netware, and Unix products this quarter. In the future, these vendors will include I₂O support as a standard feature in new OS versions, and the I₂O SIG claims that Sun, Silicon Graphics, Digital (Maynard, MA), and Hewlett-Packard (Santa Clara, CA) may add support in their Unix flavors as well.

Although implementing an OSM is far from trivial, it is a well-defined and bounded effort that is largely being handled by big companies with big R&D budgets. The I/O nodes, however, necessarily have more implementation options and may require subsystem designers to handle some or all of the software-development tasks. The 1.5 revision of the I₂O spec defines two types

of I/O nodes. The first type comprises shell-only or private-platform implementations. In a shell-only implementation, the IOP hosts message- and transport-layer services as well as a hardware-device module (HDM), which implements device-specific operations.

To understand relative complexity, think of an HDM as the device-specific portion of a traditional monolithic SCSI or LAN driver. Shell-only I₂O implementations use the IOP to handle one type of function for example, a shell-only SCSI or RAID adapter. With shell-only devices, you get a new IOP with each I/O channel you add, and the HDM is implemented as firmware.

The second type of I₂O node, called a core-compliant node, includes an IOP that can host multiple HDMs for different I/O functions. I₂O core nodes rely on a specialized real-time OS, called an I₂O real-time OS (IRTOS) that runs on the IOP. The IRTOS hosts the transport and message layers as well as the HDMs (Figure 2). The core-type node also introduces more I₂O terminology. An HDM is actually a type of a general class of I₂O software modules called device-driver modules (DDMs). Core nodes also host a second type of DDM, called an intermediate-service module (ISM), which you can use to implement custom services for one or more HDMs. For



The MegaRUM motherboard from American Megatrends accepts an optional i960 I/O processor that can control dual SCSI channels in an I₂O implementation.

example, you can implement RAID software as an ISM that in turn controls SCSI disk drives under an HDM. Conceptually, you might think of ISMs as occupying a layer between the host OSM and the device HDM, although the ISM and HDM execute as peer tasks on the IRTOS. In a core node, the IRTOS downloads the appropriate DDMs from the host.

Although the ISM concept allows you to layer functions, such as RAID control, the concept will become more important in the future. The I₂O SIG is working on Revision 2 of the spec, which will include peer-to-peer services. Some vendors, such as Xpoint Technologies, are offer-

ing samples of early implementations. To understand the value of peer services, consider a core-compliant IOP that controls both disk and LAN interfaces. A peer-to-peer ISM function could directly transfer a file from the LAN to the disk without intervention from the host CPU (Figure 3). Likewise, an ISM could handle disk-to-tape backup or disk-mirroring functions. Other possibilities for implementation in an ISM include LAN protocols, as well as data-compression and error-correction functions. Some I₂O proponents believe that you can use ISMs even to offload application-specific functions. For example, you could write a database application in such a way that an ISM executing on an IOP handles filtering or index functions.

Hardware and PCI

Note that these I₂O concepts don't imply or rely on one bus or other hardware scheme to connect host and I/O nodes. Although the PCI bus clearly will be the most popular bus for implementation, the spec allows designers the freedom to layer the message-passing scheme on top of any connection. The spec also includes a section that defines how you can map I₂O onto the PCI bus, because it has become a de facto standard in servers.

Shell-only I₂O cards will include an

I₂O PROVES A MATCH FOR SOME EMBEDDED SYSTEMS

At first glance, you might decide that Intelligent I/O (I₂O) targets only servers, thereby dismissing the technology for embedded applications particularly those with real-time requirements. However, if the technology fulfills its lofty goals in servers, you may find it a good match for any embedded system that performs a lot of I/O operations. Moreover, the core implementation of the I₂O model with the inherent I₂O real-time OS (IRTOS) provides a convenient way to wed real-time tasks with high-level application code running under Windows NT or Unix.

I₂O could fit into your plans in several ways. For starters, there is no technical obstacle that could prevent an RTOS vendor from developing an OS services module (OSM) driver, thus providing access to any networking or storage device

developed for servers and other applications. RTOS vendors might even find supporting I₂O to be easier than developing a variety of standalone drivers for specific peripherals. Understand, however, that no RTOS vendors have announced plans to deploy an OSM.

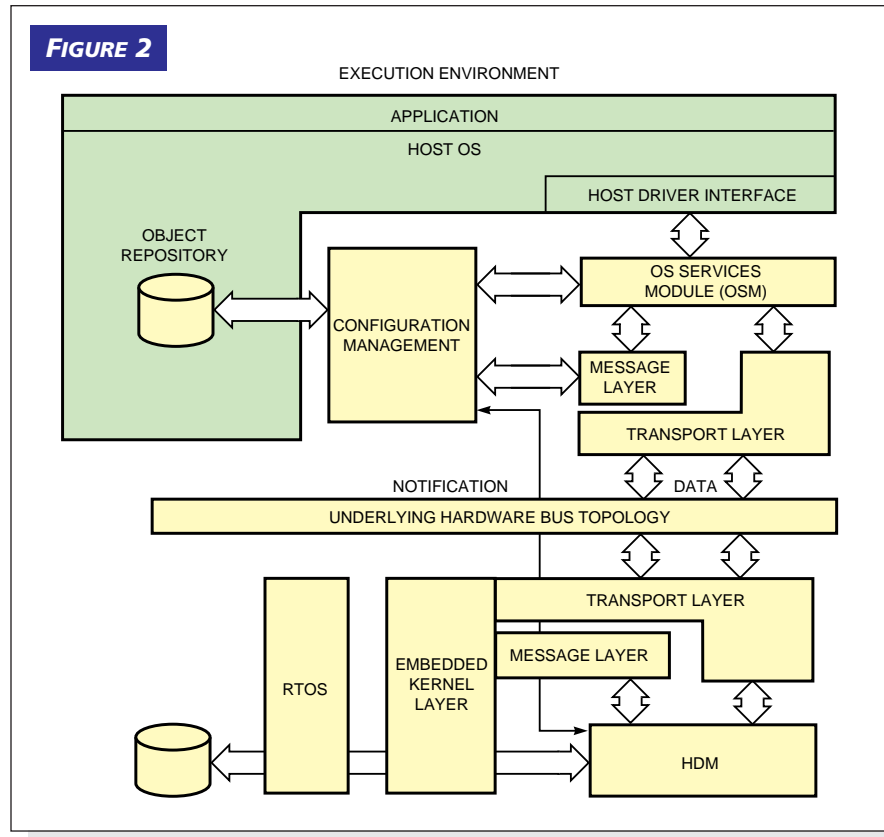
Still, you can immediately take advantage of the I₂O split-driver model to combine a real-time subsystem with an OS such as Windows NT. Increasingly, embedded products, such as telephony systems, use a mixture of Windows NT for a user interface and some other software for real-time tasks. Moreover, such systems regularly use CompactPCI as the bus interface of choice, and I₂O maps to CompactPCI just as readily as to standard PCI.

INTELLIGENT I/O

IOP on a PCI card along with the data-storage or LAN interface. Core-compliant implementations will come in several flavors. For example, a PCI add-in card could include a core-compliant IOP with several functions connected via a secondary PCI bus. In fact, most core-compliant PCI implementations will include a bridge to a secondary PCI bus. The secondary bus provides additional benefits when implementing peer-to-peer transfers because some transfers are completely absent from the primary PCI bus leaving that bus free for other activity. You will also find motherboards that include a core-complaint IOP mounted either directly on the motherboard or on a daughtercard.

A look at the hierarchy of I₂O building blocks and available products further illustrates the concepts of the standard and can help you understand what you can accomplish today. For an overview of key vendors and products, see Table 1. Starting at the lowest level, systems will need new BIOS software to realize many I₂O concepts especially for core implementations. American Megatrends, Award, and Phoenix all have I₂O-compatible BIOS offerings. A BIOS change is necessary for several reasons. For example, consider the American Megatrends MegaRUM motherboard that includes dual Pentium II μ Ps, a daughtercard socket for a i960 IOP, and dual SCSI channels. The company offers the board with software RAID support, or you can add the optional i960 for hardware RAID support. In an I₂O configuration with the optional IOP, the BIOS must first initialize the i960. It then must configure routing logic to give the IOP control of the two SCSI channels.

After the BIOS, you need to consider the IOP and, perhaps, a PCI-bridge IC, whether you plan to pursue a core or a shell I₂O implementation. In theory, you should be able to choose almost any processor, but, in reality, your choices are few. To meet the I₂O spec, you need to augment a μ P with only a message queue and some control logic. ICs from Digital, Galileo, and PLX all include the necessary queue and logic along with the primary PCI interface and a bridge to a secondary PCI bus. These ICs sell for \$40 to \$100, depending almost entirely on volume. Digital supports the Stron-



The core-compliant implementation of I₂O relies on a specialized real-time OS executing on an I/O processor to host device-specific driver modules and even I/O functions, such as RAID controllers.

gARM μ P, Galileo supports MIPS μ Ps, and PLX supports PowerPC, i960, 68000, and NEC V830 family μ Ps. Meanwhile, Intel offers the only single-chip option; its i960RP IC includes the bridge, message queue, and a 960 core.

You can compare the available offerings based on traditional characteristics, such as price, performance, integration level, and power consumption. You can even choose a different processor and develop your own support circuitry using programmable-logic ICs. However, you'll likely find that the key differentiating factor is software support. Available software support allows you to choose relatively freely among the aforementioned processors for shell-only designs. PLX, for example, offers a \$495 software-development kit with its bridge ICs, which you can use to develop your own HDM. Should you decide to build a core implementation, however, you will find your choices much more software limited.

Today, only Wind River Systems offers an IRTOS, a prerequisite for core implementations. The company has developed IxWorks, a specialized version of its VxWorks RTOS, for the i960 and StrongARM μ Ps. IxWorks is free. An environment for DDM development costs \$12,000 for the first seat and \$3000 for each additional seat.

It's interesting to note that Intel will own Digital StrongARM and bridge technology, assuming that the US government doesn't block Intel's proposed acquisition of Digital's semiconductor operation. If the acquisition goes through, StrongARM's fate is unclear. Currently, Intel has no license to the ARM core. And neither Advanced RISC Machines (Los Gatos, CA) nor its ARM licensee has legal rights to the Digital enhancements that turned an ARM core into StrongARM. It's difficult to tell what might be fact and what might be posturing for negotiating position, but Intel hints

INTELLIGENT I/O

that it has no interest in ARM. A number of board vendors, however, plan to use StrongARM rather than the i960, because StrongARM provides significantly better cost and power consumption. Obviously, you should closely monitor this situation if StrongARM interests you.

More IRTOSs and, therefore, μ P choices could be on the way. Integrated Systems (Sunnyvale, CA) has licensed PLX sI₂O Manager software and is rumored to be working on an IRTOS based on Integrated Systems own pSOS kernel. Wind River also supports a number of other μ Ps with VxWorks but has so far developed IxWorks versions only with Intel and Digital as partners. Sponsorship by the IC partners made it possible for Wind River to develop and distribute the IRTOS for free, although it still charges for the development system. It remains to be seen whether other μ P vendors will underwrite such developments or whether market forces and potential competition result in IRTOS offerings for other μ Ps.

You can also turn to data-storage-interface IC vendors for help with I₂O implementations. For example, Symbios Logic offers an integration kit that can help you develop a PCI-based I₂O disk subsystem. The kit includes a reference design based on the i960 and a Symbios SCSI IC. For \$695, you get the reference design and the company's Symplicity software that you can extend to customize a product. Other interface-IC companies, such as Adaptec, will provide similar assistance with reference designs.

Board-level implementations

Early on, however, expect board-level implementations to dominate the I₂O market. RAID-industry stalwarts, such as Adaptec, American Megatrends, Distributed Processing Technology, and Mylex, have I₂O RAID boards. Like many companies with early I₂O products, American Megatrends has offered intelligent I/O products for some time and supports the

MegaRUM in I₂O or non-I₂O environments. The company's strategy illustrates a way that technology and system vendors can gradually move toward the standard. The company supplies a shell-only HDM that works with the i960 IOP. The company has customized non-I₂O drivers for a number of OSs that work

tionality to an ISM. However, one startup, Bytestream Data Systems, has announced plans to implement RAID in an ISM. The potential advantages of ISM-layer RAID include possible standardization of host-based RAID configuration software and a potentially easier transition to new interfaces such as Fibre Channel.

Adaptec and Symbios have announced Fibre Channel boards that support the I₂O architecture. Some pundits have mistakenly stated that Fibre Channel products would have to wait for the next revision of the I₂O spec, because revision 1.5 includes a SCSI class and not a Fibre Channel class. In reality, Fibre Channel subsystems work well with the block-device class. And, you can buy such I₂O-ready host adapters for less than \$1000.

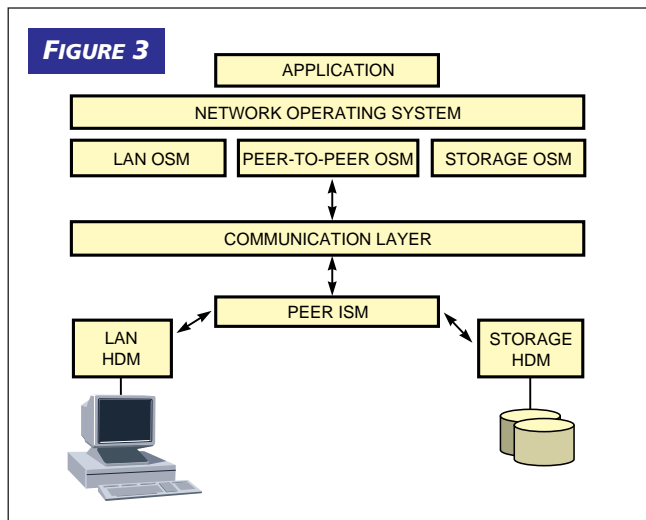
Does I₂O hold H₂O?

So, now we're back to the original question: Can I₂O fulfill its promises? At this point, no one can definitively

answer the question, but you can pinpoint the largest obstacles standing in I₂O's way. First, consider performance. I₂O depends on a significant abstraction layer, which, as history indicates, should hurt performance. For example, whenever the Windows abstraction layer grows, you need faster systems to run the OS. In the case of I₂O, OS companies are being asked to develop generic OSM drivers that, when combined with HDMs, must ultimately compete with monolithic, hand-tuned drivers that I/O experts have customized.

Most ardent I₂O supporters' simulations and comparisons show I₂O system performance compared with nonintelligent I/O subsystems. For example, Intel's I₂O Architecture Overview white paper includes graphs showing that a server CPU reaches 90% usage with two standard Fast Ethernet adapters. In comparison, the same system reaches only 59% CPU usage supporting four I₂O LAN adapters. Well, you should certainly hope to get improved performance in this case realized by support for more

FIGURE 3



I₂O offers the flexibility to establish peer-to-peer communications between devices such as disk drives, tape drives, and LANs.

with the HDM. You can also use I₂O OSM drivers as they become available. The ease of this transition path makes it likely that shell-only implementations will prove more popular than core implementations for some time.

Both Adaptec and Mylex have gone one step further, however, offering core-compliant boards that use the i960 to execute IxWorks. In reality, the companies can also offer shell-only implementations. Still, by developing full IRTOS-based products, the companies have been able to test their products more easily than have vendors of shell-only products. The IRTOS provides a robust software environment for development, and Wind River Systems and Intel have assisted the testing process at so-called plug fests.

To date, companies such as Adaptec, American Megatrends, and Mylex have kept their RAID control logic and firmware carefully locked behind the HDM interface. Such companies consider their RAID technology a key product differentiator and are unlikely to move such func-

INTELLIGENT I/O

users with intelligent adapters. I₂O supporters argue that establishing an intelligent I/O standard will ultimately increase the use of the technology and lower prices to near the level of unintelligent systems. If performance matters, however, you need to compare I₂O systems with other systems that implement intelligent I/O.

In some cases, I₂O also faces tough challenges from what might be considered nonintelligent I/O implementations. Given historical precedent, ICs integrate increasingly more portions of a given task and, in the process, improve performance. Take high-end SCSI and Fibre Channel ICs from QLogic, for example. The ICs essentially integrate an IOP and require little host intervention during I/O. QLogic's Marketing Executive Skip Jones notes that his company's ISP chip family averages fewer than one interrupt per I/O transaction, essentially already offloading CPU overhead and bus traffic. Moreover, QLogic ICs require no extra memory as do I₂O IOP or the message-passing overhead. QLogic, a member of the I₂O SIG, plans to support the standard, but Jones warns that you should deploy I₂O only when it can make effective use of the message-passing model.

Making I₂O plug-and-play

Assuming that I₂O supporters can minimize overhead because of the abstraction layer and generic drivers, can the group deliver on plug-and-play intelligent I/O? Down the road, an MIS manager with an I₂O server could ideally decide to add an I₂O storage board or LAN adapter without regard for the manufacturer of the system, motherboard, or adapter card. The OS drivers may work with any device, but mixing and matching I₂O boards could become a problem.

The sheer flexibility of the I₂O spec makes the plug-and-play task daunting. For example, consider a server motherboard with a primary PCI bus and an onboard IOP controlling a secondary PCI bus. Both of the PCI buses have accessible connectors, and the IOP includes a core-compliant IRTOS that controls LAN and storage adapters connected via the secondary bus. Now, imagine that an MIS manager wants to add another SCSI channel and orders a new PCI card labeled I₂O-compatible.

FOR MORE INFORMATION...

Manufacturer	Circle No.	I ₂ O Products
Adaptec Inc Milpitas, CA 1-408-945-8600 www.adaptec.com	301	PCI SCSI ICs and boards, RAID ICs and boards
American Megatrends Inc Norcross, GA 1-770-246-8600 www.ami.com	302	SCSI and RAID ICs and boards, motherboards and BIOS software
Award Software Mountain View, CA 1-408-237-6800 www.award.com	303	BIOS software
ByteStream Data Systems Inc Maynard, MA, 1-508-461-1041	304	ISM RAID software, PCI RAID controller
Digital Semiconductor Hudson, MA 1-510-490-4753 www.digital.com/semiconductor	305	StrongARM and Alpha Ps, PCI bridge ICs
Distributed Processing Technology Maitland, FL 1-407-830-5522 www.dpt.com	306	SCSI and RAID boards
Galileo Technology San Jose, CA, 1-408-451-1400 www.galileot.com	307	PCI bridge ICs for MIPS Ps
Intel Corp Santa Clara, CA 1-408-765-8080 www.intel.com/procs/servers/i2otech	308	960 and Pentium Ps, motherboards, PCI-bridge ICs, core logic
Matrox Networks Dorval, PQ, Canada 1-514-969-6080 www.matrox.com	309	Fast Ethernet LAN cards
Mylex Corp Fremont, CA 1-510-796-6100 www.mylex.com	310	Motherboards, PCI SCSI and RAID boards
Phoenix Technologies San Jose, CA 1-408-570-1000 www.phoenix.com	311	BIOS software
PLX Technology Sunnyvale, CA 1-408-774-9060 www.plxtech.com	312	PCI-bridge ICs, HDM software, HDM development kits, reference designs
QLogic Corp Costa Mesa, CA 1-714-438-2200 www.qlc.com	313	SCSI and Fibre Channel ICs
Symbios Logic ICs, Fort Collins, CO 1-970-223-5100 www.symbios.com	314	PCI SCSI and Fibre Channel boards and adapter card development kits, reference designs, PCI-bridge ICs
Tyan Computer Milpitas, CA, 1-408-956-8000 www.tyan.com	315	Motherboards
Wind River Systems Alameda, CA, 1-510-748-4100 www.wrs.com	316	IxWorks IRTOS and development tools
Xpoint Technologies Boca Raton, FL 1-561-241-8447 www.xpoint-tech.com	317	Software-development tools; OSM, ISM, and HDM software; peer-to-peer software; integration, engineering, and testing services

INTELLIGENT I/O

Unknown to the MIS manager, however, the new board has a shell-only I₂O implementation. Ideally, the MIS manager should plug the new board with its own IOP into the primary PCI bus to avoid the bridge latency to the secondary bus. In connecting via the primary bus, however, it is unclear that an ISM running on the motherboard IOP could communicate with the HDM on the shell-only adapter. If you plug the adapter into the secondary bus, the IOP on the motherboard can clearly control the new card, but, IOPs connected in series must twice interpret I₂O messages, and transfers to the new card incur bridge latency.

This hypothetical scenario could get worst. Assume that a server vendor sells the above-described motherboard. The vendor could offer additional LAN and storage adapters designed to work behind the core-compliant IOP. Should you plug one of these I₂O-compatible adapters lacking an onboard IOP into the primary PCI bus, there is no guarantee that the system would work. The motherboard IOP could possibly control the new board as a peer across the primary PCI bus, but that is not the intent. Moreover, what is to keep someone from buying that I₂O-compatible adapter and plugging it into an I₂O-ready motherboard with an unpopulated IOP socket?

Perhaps you can assume that buyers of I₂O technology will be competent enough to choose the correct products and to know how to connect them. Certainly, MIS managers should be able to do so, but the troublesome situation of multiple configurations will only get worse. In addition to peer-to-peer services, Revision 2 of the specification (due this year) will add support for 64-bit addressing, hot-plug PCI, and system-area-network (SAN) attached I/O. Hot-plug capability means that some system entity will need to load the correct DDMs whenever a user inserts a new card. SAN-attached I/O will address I₂O message transmission across a storage backbone connecting clusters of storage subsystems.

I₂O proponents are also looking beyond Revision 2 toward a third type of I₂O implementation, called fixed-function. Still in the early planning stage, this new implementation would allow the messaging layer to be absorbed into

hardware, resulting in lower cost and perhaps higher performance devices. It is unclear whether the fixed-function implementation would fully support the I₂O concept or just a subset. The added complexity from all of these additions will provide an even greater challenge to system designers attempting to supply OS independence and to users seeking mix-and-match interoperable intelligent I/O products from different vendors.



You can reach Technical Editor Maury Wright at (619) 748-6785, fax (619) 679-1861, ednwright@ncimail.com.

VOTE

Please use the Information Retrieval Service card to rate this article (circle one):

High
Interest
586

Medium
Interest
587

Low
Interest
588