Intelligent I/O in the PC-Based Enterprise Computing Environment

Table of Contents

Introduction

Microprocessors have invaded the workplace. Office PCs have replaced typewriters and calculators, while interoffice networks are taking the place of filing cabinets and fax machines. In more technical environments, Intel Pentium® processor-based PCs and servers are quickly displacing minicomputers and high-end workstations, and massively-parallel arrays of commercial microprocessors are enabling next-generation supercomputers for the scientific and research communities. Each of these market segments has yielded readily to ongoing improvements in microprocessor price and performance.

The last holdout against this onslaught has been the enterprise-wide computer center, a market segment traditionally served by mainframes and fault-tolerant minis. To date, client/server computing has required a level of performance, reliability and connectivity that conventional PCs failed to deliver. But with ever-faster Pentium processors, the upcoming P6, and Intel's new IQ series of intelligent I/O subsystem processors, the situation is changing. Microprocessor technology may soon begin entering this last bastion of the mainframe market segment.

The Needs of Enterprise Computing

The enterprise computing market segment is a tough nut to crack. Top-of-the-line performance is critical to many applications, but MIS directors have other special needs, about which they are justifiably insistent. Before PC application servers will become ubiquitous in enterprise computing, they will need to match mainframes in the following respects:

- **Scalability**. Enterprise computing needs unsurpassed performance and scalability, with no perceptible delays due to system loading. It's easy enough to buy new PCs for employees as they're hired, but as corporate computing requirements grow, MIS capacity must expand gracefully to meet the demand.
- **Availability**. Compute power must always be on-line, ready, and raring to go. When a desktop PC dies, life goes on; a replacement can be drafted from a neighboring desk. When a mainframe dies, the whole corporation shuts down. Maintenance, diagnostics and module replacement must therefore take place with no unscheduled down-time.
- **Reliability**. If a system component *does* fail, enterprise computers cannot risk losing data. When a PC crashes, the occasional memo or spread sheet may need to be retyped. But when a mainframe dies, it may take corporate financial records, customer orders, communications systems and the employee payroll with it.
- **I/O Connectivity**. It's not hard for an isolated PC or a client node on a network to satisfy the I/O needs of a single user. If half a second is lost rewriting the screen or fetching data from a network, individual users might not notice. But mainframes and application servers face a tougher challenge. Massive amounts of data must be collected and dispersed continuously for hundreds of users, at physically remote installations, involving a large (and growing) number of network types and protocols. Bandwidth must be high to accommodate I/O demands system-wide, yet the CPU can't afford to get bogged down. More pressing tasks are always pending; there's always more work to be done and many users waiting.

Performance-wise, Pentium® processors provide unparalleled levels of throughput and the P6 will be faster still. Both product lines support symmetric multiple-processor (SMP) configurations, in which performance can scale gracefully merely by installing additional CPUs on the motherboard or backplane.

Both the Pentium and P6 processors perform extensive fault checking at the chip level, and provide parity or error-correction bits on internal and external memory arrays and buses. But statistically, CPU failures are quite rare to begin with. The vast majority of computer system failures are due to problems in the disk drives, power supplies, and fans.

Availability and reliability demands can thus be satisfied by adopting fault-tolerant system design techniques. Data mirroring or RAID (redundant arrays of inexpensive disks) storage systems can ensure that if a drive were to fail, its data could be recovered from the contents of surviving disk drives. Servers may incorporate redundant power supplies and storage modules, and circuit boards and subassemblies may be swapped out and brought on-line without disrupting system operation. Background monitoring of power supply and fan operation can reduce the likelihood of unexpected catastrophic failures.

But in order to satisfy MIS and server market segment requirements for I/O bandwidth and connectivity, computer designers will need to adopt a new architecture mindset. New forms of task partitioning, new bus protocols and new schemes for offloading system interrupts will be needed to ensure that next-generation systems are able to reach their full potential. The techniques of performance scalability, increased reliability and greater I/O bandwidth developed for mainframes in the enterprise computing market segment will be equally applicable to the high-end PC server market segment.

Balancing Performance and Bandwidth

All computers contain a CPU, memory and I/O. For efficient system operation these elements must be balanced. It does no good to crank up the speed of a CPU unless the size of the main memory also is increased; larger memories will go to waste unless there are faster channels for rolling data in and out. Amdahl's Law (formulated by Gene Amdahl, architect of the IBM 360 mainframe and founder of Amdahl Computer) states that these needs grow proportionately. Bumping CPU performance by an order of magnitude would demand ten times the memory and consume ten times the I/O bandwidth.

To date, technology advances have most benefited the CPU and memory elements. Pentium® processor-based PCs can easily deliver 100 MIPS of performance and 100 megabytes of DRAM fills an area just a few inches square.

Bandwidth needs have grown proportionately, as storage devices and network communication protocols keep improving. The SCSI-II standard raises its bandwidth to 20 Mbytes/sec; SCSI-III will double that again. Ethernet has gone from 10 Mbits/sec to 100. Fibre channel raises this nearly another order of magnitude to 100 Mbytes/sec. ATM networks communicate at from 25 to 622 Mbits/sec.

What hasn't changed much is the way PCs do I/O internally. I/O architectures are still "flat;" the host CPU must mediate the demands of all I/O subsystems and interface cards. Heavily-loaded CPUs may still spend up to 30 percent of their cycles on overhead functions such as peripheral status polling and data-block transfers. As CPU performance levels rise, conventional I/O architectures will surely choke.

The Need for a New I/O Paradigm

As processor clock rates climb and pipelines become more complex, it becomes more and more beneficial to relieve the host CPU from its I/O-related tasks. The P6 will debut this year in high-end servers. By the end of 1996, PCs with dual-P6 capability will be available for heavy-duty home-office use. These systems threaten to overwhelm earlier I/O architectures.

Moreover, as more and more network standards come into use, the same servers that in the past dealt only with local drives and a single network protocol now find themselves being pressed into service dealing with networks of multiple types. The bottom line is that system designers must rapidly increase the performance of high-end servers, desktops and consumer platforms.

But how can the I/O bottleneck be broken? As more and more data is shoveled in and out of system memory, the bandwidth of the raw I/O channel must improve. It becomes vital to perform I/O tasks intelligently, at a level closer to the I/O interface, away from any shared system resources. And subsystems must offload as many as possible of the asynchronous I/O events that now waste much of the host CPU's time.

Now, for the first time the need for a new I/O paradigm, the bus-level architectures that make such a change possible, and the products are coming together. The same technologies that boosted the rest of the system are now being applied to I/O as well.

Breaking The Bandwidth Bottleneck

PC users have seen a veritable alphabet soup of bus specifications evolve: ISA, EISA, MCA and VLB backplanes each improved raw bandwidth via wider data paths and faster bus clocks, but the improvements have failed to match the orders-of-magnitude improvements seen elsewhere with a modern PC. The disparity between processor speed and real-world I/O is growing. Bandwidth requirements have swamped existing bus architects, leaving a gaping hole in the system balance equation.

Technology advances have presented a triple whammy to backplane designers. Higher clock rates, on-chip clock multipliers, and the increased number of instructions dispatched per core CPU cycle have all increased the opportunity cost of waiting for slow transfers to complete. An Intel486™ DX microprocessor can dispatch one instruction during each bus cycle. If a peripheral device takes 250 nsec to respond, then a 33-MHz Intel486 DX microprocessor might lose up to eight instruction dispatch slots waiting for an I/O instruction to complete.

A 100-MHz Pentium® microprocessor, in contrast, can dispatch up to 50 instructions in the same 250 nsec; a 133-MHz P6, up to 100. It makes little sense in a high-end server for the CPU to freeze while awaiting data from dumb peripherals. Transfer requests must be decoupled from completion, so that other transfers may be initiated or completed while a slow peripheral responds to requests.

The PCI (Peripheral Component Interconnect) bus laid the groundwork for addressing these issues. At 33 MHz the PCI bus can sustain a 133 Mbytes/sec burst-mode transfer rate. Transfer requests are decoupled from completion; every microsecond dozens of new transfers may be initiated and/or completed. PCI peripherals also provide bus mastering capability, so that lengthy data-block transfers may proceed without the intervention of a host CPU or central DMA controller.

Repartitioning I/O Tasks

Just as memory caches provide a hierarchy of different device types and speeds, I/O architectures must be designed for increased bandwidth parallelism at different levels of the design. Mainframe and supercomputer architects have met these needs with an array of I/O or channel processors, minicomputer-class peripheral processors dedicated to performing all I/O relating to a peripheral function, such as storage. I/O processing responsibilities, intelligence, and computational power have been moved onto intelligent controllers, away from the host CPU and much closer to the actual I/O congestion.

Offloading Host Interrupts

In PCs, as long as the host CPU needs to concern itself with every I/O command or data-block transfer, performance will inherently be limited. 200+ MIPS processors will need to take a different tack. The CPU can ill afford to get bogged down by the tyranny of asynchronous events. Just keeping a P6 fed with new data and pending tasks becomes a challenging problem in system design; offloading the results it produces is just as tough.

One way an I/O processor can maximize system efficiency is by minimizing the number of asynchronous interrupts the host CPU must process. Asynchronous interrupts reduce host efficiency, since servicing an interrupt forces the CPU to stop what it's doing, save the complete machine state and begin executing from a different location. Upon completion the service routine must reload the original state, reconfirm process access rights and resume execution wherever earlier processing left off. Interrupt processing also reduces cache efficiency as service-routine instructions may flush instructions already present in the cache.

Intelligent I/O processors can ensure that an entire command sequence or network transfer protocol is completed with no host intervention. But requiring the host CPU to process even one interrupt per command sequence is still suboptimal. I/O processors can further reduce the host interrupt load by building a list of command-completion messages within shared memory. When new messages are placed into a formerly empty list the I/O processor interrupts the host CPU. As the host unloads command-completion messages from the head of the list, the I/O processor can continue appending new completion messages to the tail.

The host can thus defray the overhead of a single interrupt across multiple completion messages. More importantly, this scheme scales well: The more completion messages posted by the I/O processor to the host, the larger the number of transactions processed by the host per interrupt. A more heavily loaded system, then, actually decreases the time spent by the host on interrupt overhead.

The i960® RP Processor Intelligent I/O Subsystem

The first of Intel's IQ series of intelligent I/O processors can use each of the above methods to enhance the operation of microcomputer systems at varying levels of sophistication. In simple desktop PC configurations, the device can perform bus expandability and bridging functions, and can serve as a low-level controller in I/O adapter cards. With more advanced servers and multiprocessing systems based on 120-MHz Pentium® processors or the P6, these devices will serve as channel processors to satisfy the demands of increased communications bandwidth, hierarchical intelligence and interrupt offloading.

The first member of this family, designated the i960® RP processor, combines onto a single chip all the elements essential for high-bandwidth I/O. At the heart of the device is a complete 32-bit RISC microprocessor. The i960 architecture incorporates bit-field operations, efficient context switches, fast interrupt handling, multiple-part addressing modes, variable-length instructions and other features that make it especially well-suited for data manipulation applications. Since its 1988 introduction the i960 microprocessor family has become one of the highest volume 32-bit processor families in the industry and pioneered many of the design features (superscalar execution, functional redundancy checking and transaction-oriented bus protocols) that have since been adopted by the Pentium processor and P6 processor designs.

The i960 RP processor device contains interface logic for two complete PCI buses; an assortment of on-chip registers, memory, caches and other resources for maximum performance; an interface to off-chip local main memory for control software, algorithms and data buffers; and timers, interrupt controllers and interface logic to assure that other elements work together with peak efficiency. The processor, peripheral circuitry and other resources packed onto this single-chip device would heretofore have required an entire system of board- or box-level complexity.

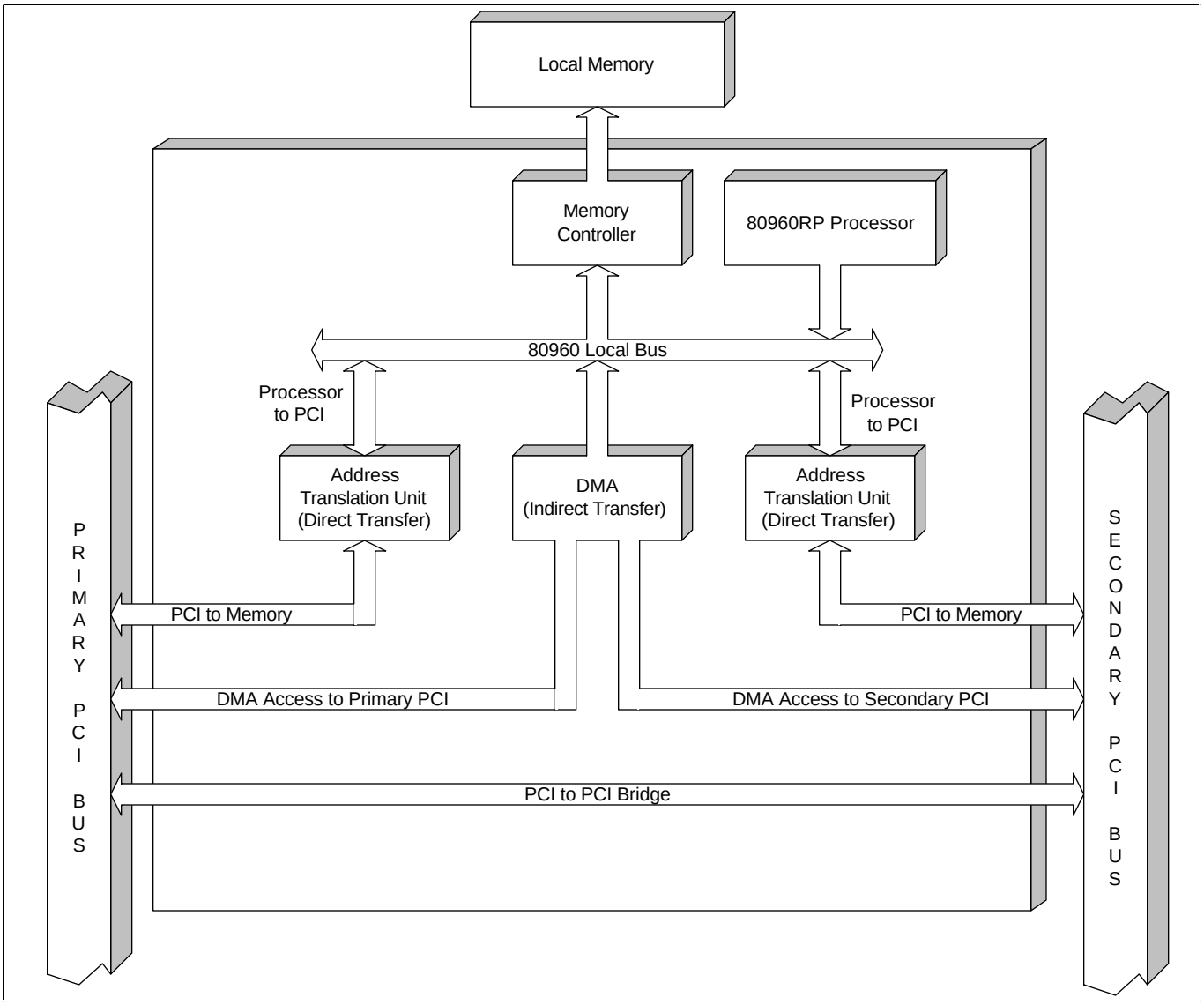

Figure 1. i960® RP Processor Data Flow Requirements

I/O Processor Overview

Figure 1 shows a block diagram of the I/O processor. It typically operates as a slave to one PCI bus, accepting firmware control algorithms and command blocks from a host CPU at the next higher level of the I/O hierarchy. The host CPU might be an Intel Pentium processor, a P6, another I/O processor, or some other device. The second PCI interface connects the component to PCI-compatible network interfaces, storage devices, or an I/O processor subsystem at the next lower level of the hierarchy.

The device can perform a plethora of functions in and around the PC. Figure 2 illustrates how an I/O processor on a motherboard can serve as an intelligent PCI backplane expander and bridge, and perform system monitoring and maintenance functions without disrupting the host CPU. Figure 3 shows how the device may be used as I/O bus expander, communications bridge, or intelligent I/O processor on an add-in adapter board, implementing data-communications network protocols or performing complex control algorithms for storage devices.

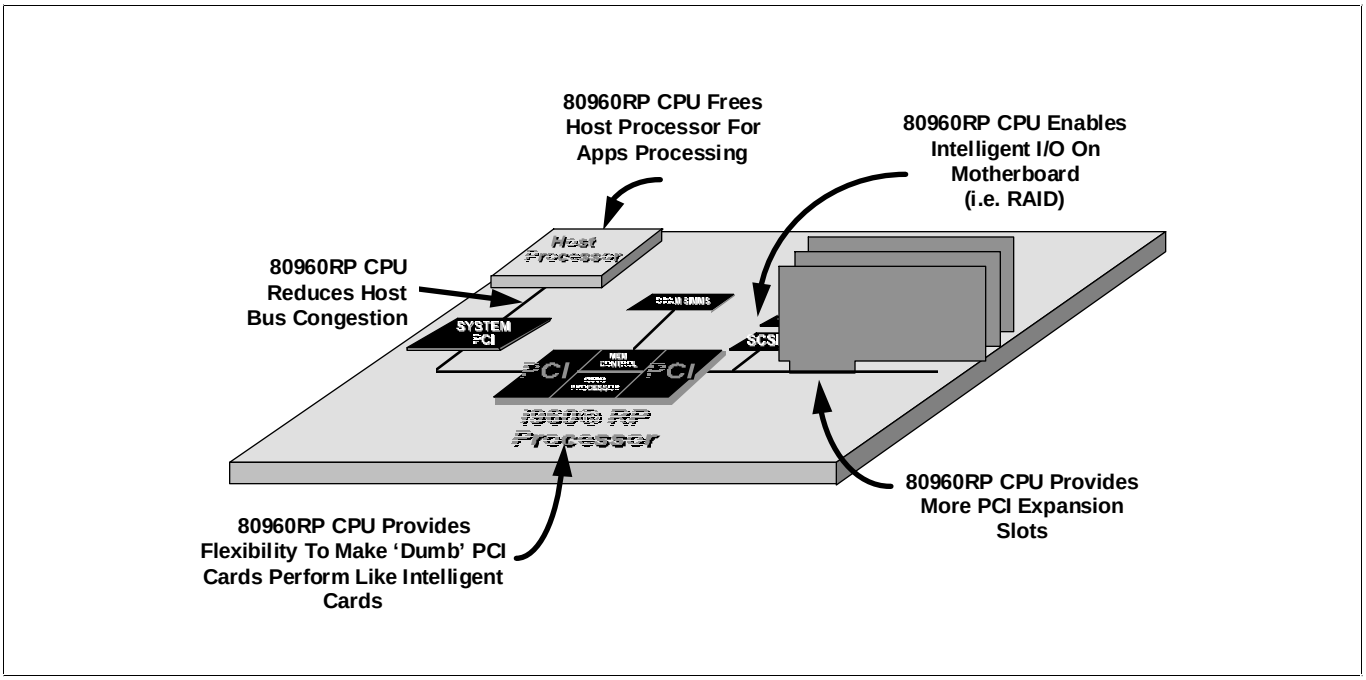

Figure 2. i960® RP Processor Applications on a PC or Server Motherboard

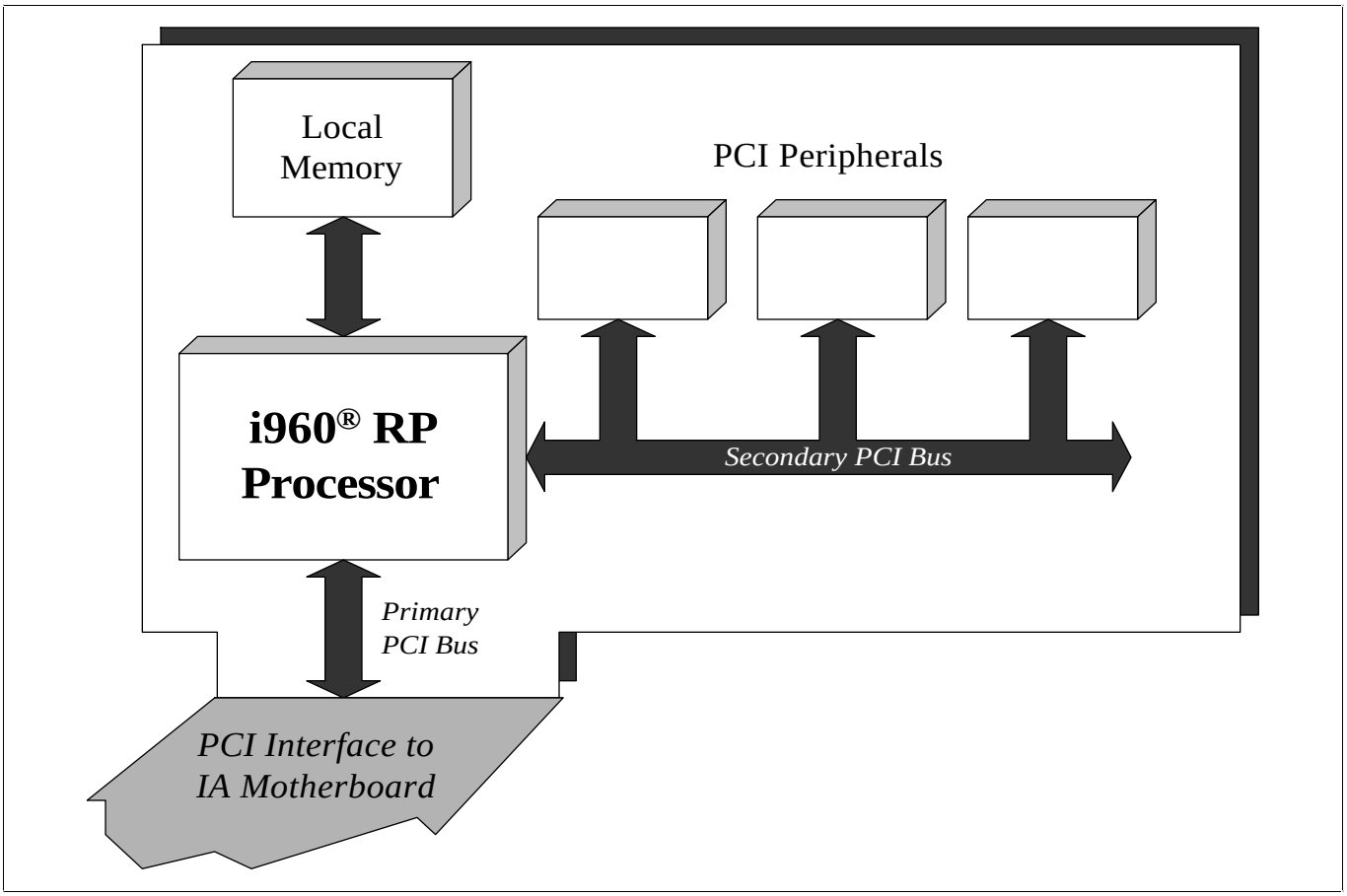

Figure 3. i960® RP Processor-Based I/O Controller on an Add-In Adapter Card

I/O Subsystem Resources

Figure 4 is a block diagram showing the on-chip elements of an i960® RP processor. These elements are described below:

- **Microprocessor Core.** At the heart of the I/O subsystem is a 32-bit CPU based on the Intel i960 architecture. Integrated within the processor are a 4KB instruction cache and a 2KB data cache, as well as 1KB of fast RAM and a dedicated cache for eight sets of local registers (128 registers, or 512 bytes total). The CPU runs at 33 MHz and can begin a new instruction every cycle, delivering more than 31 VAX MIPS.
- **Local Memory Interface.** A local memory interface provides address, data and control signals to directly support up to 256 MB of DRAM built from conventional, fast page mode, extended data out (EDO), or burst EDO devices, with or without parity. Programmable chip-select outputs enable SRAM, ROM, or Flash EPROM memory arrays of up to 16 megabytes, eight or 32 bits wide, with support for individual or burst-mode transfers.
- **Primary PCI Interface.** One of the PCI interfaces built into the I/O processor typically connects the device to the host CPU or the next higher level of the I/O hierarchy. For system architectures that place the I/O processor on the motherboard with the host CPU, this interface would connect to the main system backplane. For add-in boards, this interface would connect to the PCI-bus edge connector. This interface complies with the PCI 2.1 specification.
- **Secondary PCI Interface.** A second PCI interface connects the I/O processor to components under its control. These might include expansion connectors on a PC motherboard, PCI-compatible disk or network controller chips, or the next lower level of the I/O hierarchy. The secondary PCI interface is compatible with devices that meet the PCI 1.0 specification.

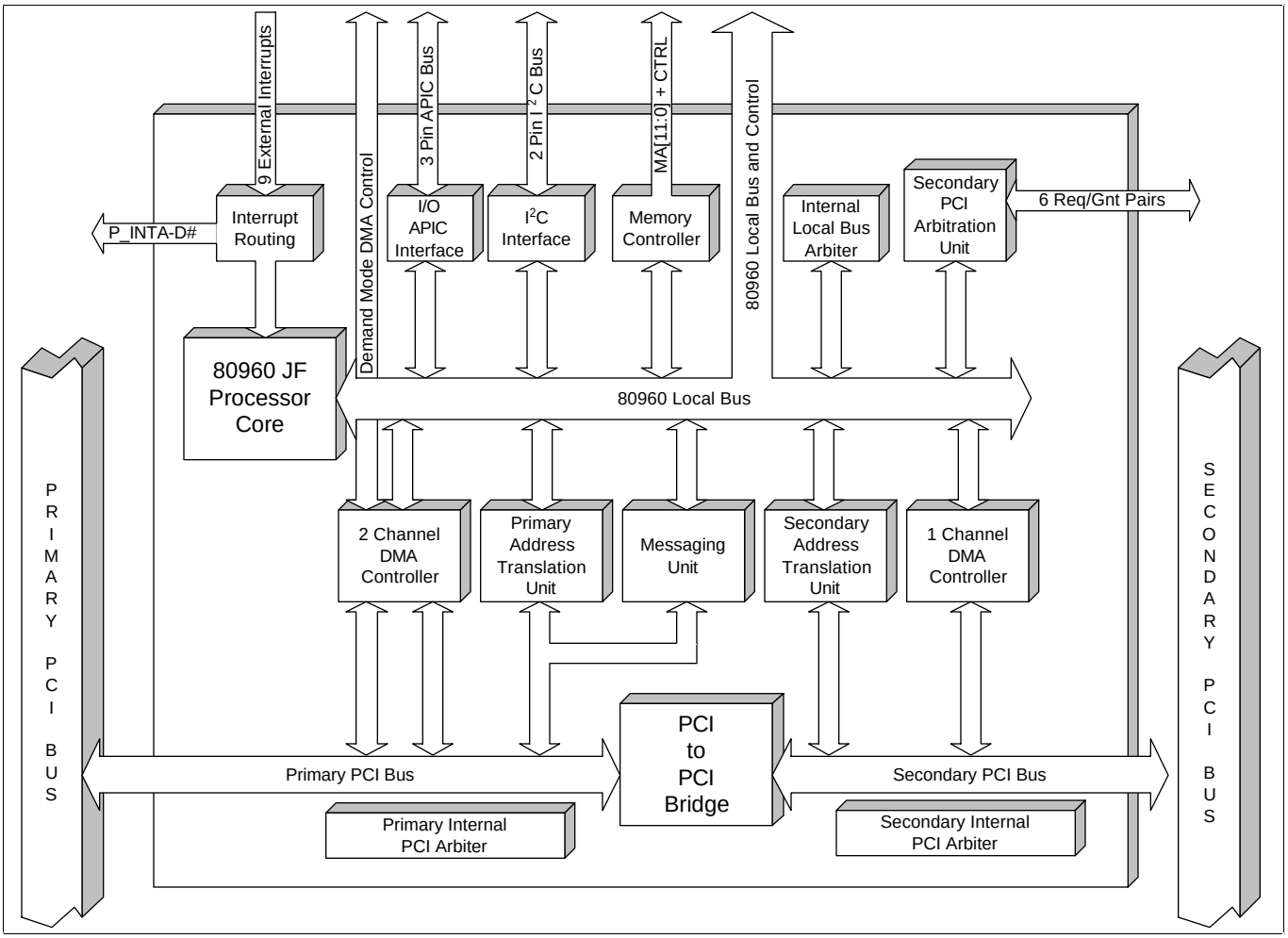

Figure 4. i960® RP Microprocessor Block Diagram

- **PCI-to-PCI Bridge.** The I/O processor can be configured to pass data between the two PCI buses with no intervention from the i960 processor core. A 64-byte FIFO lets outbound commands and data stream asynchronously from the primary bus to PCI devices on the secondary bus. A second 64-byte FIFO independently buffers in-bound transactions from the secondary PCI bus to the host.
- **Address Translation Units.** Alternatively, memory transfers on the primary PCI bus may be mapped onto the I/O processor's local memory array. An address-translation unit determines how host-system addresses map onto the memory space of the core i960 processor. A second address-translation unit maps core processor addresses onto the secondary PCI address space, so the I/O processor software can read or write data to I/O adapters as if they appeared within the i960 RP processor's own local memory space.
- **DMA Control Logic.** The I/O subsystem also contains a three-channel DMA controller. Two channels transfer data between the I/O processor's memory space and memory or peripheral devices on the primary PCI bus, and allow simultaneous in-bound and out-bound transfers at up to 132 MB/sec. The third DMA channel transfers data between the I/O processor's local memory and devices on the secondary PCI bus. Each channel includes a 64-byte data queue to assure optimum bus efficiency. Transfers to or from unaligned addresses are packed and unpacked automatically as needed to form optimally aligned 32-bit values, and up to 16 MB of data can be transferred in a single block.
- **Messaging Unit.** The messaging unit provides a mechanism by which a host processor can signal the I/O processor that new commands or data are ready to be processed. Whenever the host writes new values into the message memory space, the address and data will be recorded in dedicated registers and the I/O processor will be interrupted. Alternatively, command values can be stored in a circular queue up to 256KB long. The messaging unit maintains head- and tail-pointer comparison logic to prevent the host from writing new values into an already-full queue.
- **I^2C Interface.** The i960 RP processor contains a synchronous, multi-master serial bus interface for system-wide data acquisition and monitoring functions. When the core CPU is not occupied with more pressing tasks, this two-wire bus can be used to monitor power-supply regulation, fans, air temperature and circulation, the integrity of the system enclosure, proper seating of replaceable subsystems and processor fault status.
- **Host APIC Interface.** The i960 RP processor also provides an interface to the APIC (Advanced Programmable Interrupt Controller) bus. This interface can send interrupt messages concerning command completions, asynchronous service requests, or an exceptional system condition to a host Pentium or P6 processor. The CPU can likewise signal the I/O processor when interrupt messages have been detected and processed.
- **Core Support Functions.** The I/O processor also contains two 32-bit general-purpose timers to facilitate time-critical control functions and provide a time-base for the real-time operating system kernel. A reconfigurable interrupt control unit with programmable priorities lets interrupts be requested as a result of transactions on either PCI bus, messaging-unit flags, transactions on the I^2C bus, timer events, or external requests from up to nine inputs.

The i960 RP microprocessor will be fabricated on Intel's 0.8-micron, three-layer metal CMOS process and will contain just over a million transistors on a die 480 x 480 mils (149 mm^2) in size. It will operate with a 5 volt power supply and directly support 5-volt PCI systems, dissipating less than 3 watts at 33 MHz.

The device will be offered in a 352-lead peripheral ball-grid array (P-BGA) package that measures 1.4" square. The P-BGA package provides the high pin count needed to support three complete 32-bit bus interfaces while keeping the package small in order to reduce real-estate requirements on densely packed motherboards and add-in cards.

Target Applications

The features and capabilities of the i960® RP processor can be combined in a variety of new ways to enable new applications and implementation techniques. Some examples:

On-the-Fly Data Translation

I/O chores aren't always as simple as DMA transfers of continuous blocks of data in its natural form. RAID storage arrays typically scatter blocks of memory and apply redundancy algorithms as data is written onto multiple drives; to reread the data, interleaved blocks must be gathered and recombined. When disk errors are seen, special algorithms must be invoked to reconstruct the missing data. The incorporation of a high-performance CPU, local memory buffers and reconfigurable data-transfer channels within the i960 RP processor simplifies each of these tasks.

DMA Command Chaining

The core CPU in the i960 RP processor interacts with the on-chip DMA controller through local memory-based command blocks. The core CPU can define a chain of command blocks, and the on-chip DMA controller will automatically retrieve and interpret each command in sequence. A complex series of DMA transfers may be defined, for example, to implement the scattering and gathering of disk sectors in a RAID array. Once the DMA controller kicks off, these operations can execute to completion with no further intervention from either the host or the core i960 RP CPU.

In-the-Field Upgrades

The universality of the i960 RP processor design also offers system designers the prospect of a generic, reprogrammable, multifunction adapter card platform. A single board may connect to conventional hard-disks, RAID arrays, optical storage, network Ethernet interfaces, FDDI, or video networks in various combinations. The adapter card would provide a physical link between the host CPU and external peripherals, but the functions performed would be tailored to the system via run-time software, with different firmware and control algorithms invoked as determined by the host OS.

With different software, an interface card that initially served as a simple disk controller would be able to compress and decompress data transparently as it is written to or read from a drive, or control a fault-tolerant array of RAID devices. A network interface card might be retrofitted to implement a new network protocol. The migration of device driver software and file systems onto remote subsystems may change how people think about operating system software, opening the market segment for new vendors to leap into the OS and I/O driver fray.

Summary

Microprocessors have come a long way since the days of the first PC. The Intel Architecture has evolved into an enterprise-wide computing solution, from laptops and desktops to networks and high-end servers. Performance has grown thousand-fold, with memory systems nearly keeping pace.

About the last thing to change has been the underlying I/O architecture. As CPU core frequencies have risen while I/O standards remained fixed, as memory systems have grown but I/O remained a bottleneck, and as cache sophistication has increased while I/O functions remained uncacheable, the equilibrium in high-end servers between compute power, memory size and I/O bandwidth has fallen increasingly out of balance.

The i960® RP processor I/O subsystem has been designed to restore the desired balance. The device combines onto a single reconfigurable chip the bus interfaces, bridge-logic FIFOs, address translation units, DMA channels, message semaphores and other support circuitry needed to enable a wealth of new applications. But more importantly, by integrating a general-purpose high-performance 32-bit processor onto the same die as the I/O logic, the i960 RP processor provides the intelligence by which general-purpose circuits can adapt to new applications as quickly as they can be identified.