S/390 CMOS server I/O: The continuing evolution

by T. A. Gregg

IBM has developed a strategy to achieve the high I/O demands of large servers. In a new environment of industry-standard peripheral component interconnect (PCI) attached adapters conforming to open I/O interfaces, S/390® has developed an efficient method of quickly integrating disk storage, communications, and future adapters. Preserving the S/390 I/O programming model and the high level of data integrity expected in S/390 products and reducing development cycle time and resources have further constrained design options. At the same time, S/390 developers have redesigned the traditional I/O components into the latest chip technologies. The developers have also designed a new internal link (STI) to meet the increased I/O bandwidth and connectivity required by the high processor performance of the third and fourth generations of S/390 CMOS servers. This paper describes this strategy and how it has led to systems that retain the differentiating features of S/390 products.

Introduction

A new I/O subsystem is evolving in the CMOS-based S/390* Parallel Enterprise Servers. Breaking from the traditional S/390 I/O model in which external interfaces [Enterprise Systems Connection (ESCON*) and Parallel Channels] are used to attach external communications adapters and disk storage, internal network interface adapters (Ethernet, token ring, FDDI, and ATM) and disk storage have been added. Another significant change is the development of a new internal system link that allows the I/O infrastructure to scale with the increased processor performance of the S/390 CMOS servers. Finally, redesigns of the traditional I/O elements are improving their performance while dramatically lowering their cost and improving their reliability. All of the above changes better position S/390 as a cost-competitive server in the network computing environment. This paper describes this evolution by examining the strategy that led to today's S/390 I/O structure and that will continue to influence future development.

There are several key elements to the strategy, and one is the reuse of parts and designs. As the customer upgrades from the second generation (G2) through the fourth generation (G4) of S/390 CMOS servers, most of the I/O subsystem components can be retained, substantially reducing the cost of upgrading. Similarly, when elements are remapped into newer chip technologies, design changes are limited to those required to solve real problems. A second element of the strategy is to retain software compatibility, and the S/390 I/O programming model has been preserved. The internal communications adapters emulate the IBM 3172 interconnect controller, and the internal disk storage emulates the IBM 3990/3380/3390 disk storage subsystem. The redesigns of the traditional I/O elements naturally preserve the S/390 I/O programming model.

0018-8646/97/$5.00 © 1997 IBM

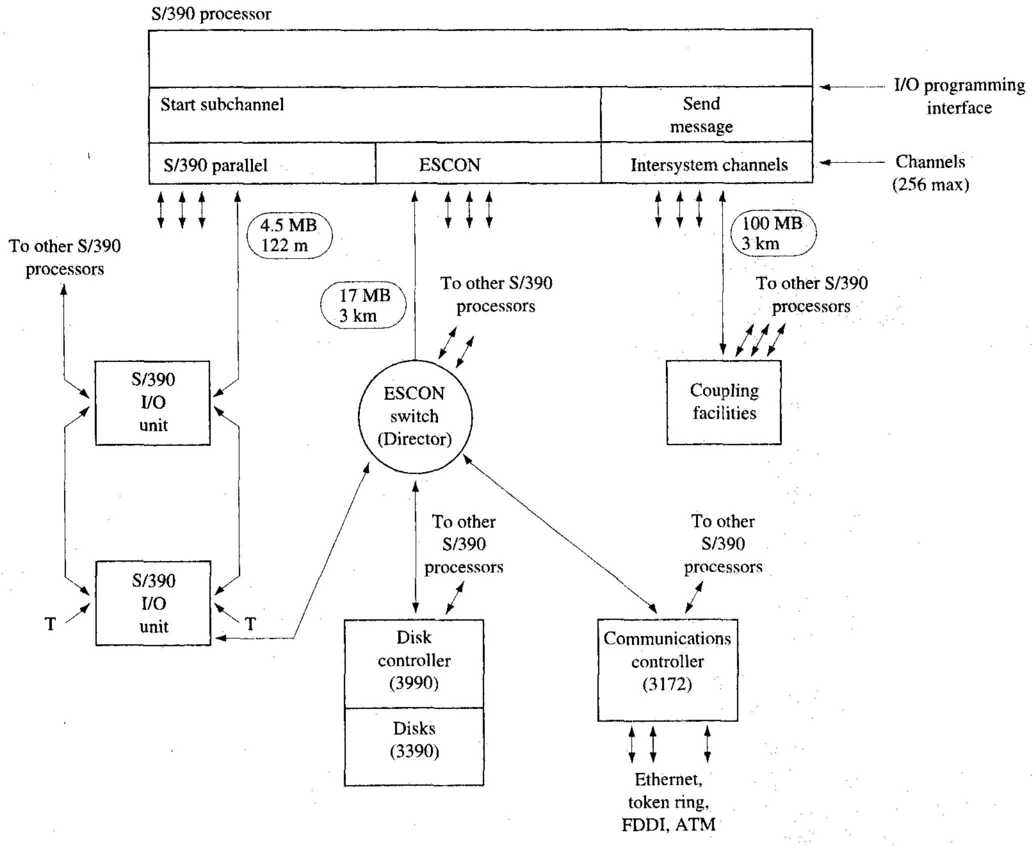

Figure 1

S/390 computer complex.

Both internal communications adapters and disk storage have been implemented with off-the-shelf components to reduce development expense while improving the time to market. Since these components are typically not designed to support the S/390 I/O programming model, adaptation layers had to be implemented. The S/390 I/O programming model is unique in the computer industry, and in the future, the S/390 I/O programming model may have to be augmented with industry-standard I/O programming models.

Early S/390 I/O evolution

In the early 1960s the S/360 computing systems had an I/O structure that connected stand-alone computers (S/390 processors) to external I/O enclosures. At that time, and for almost the next three decades, each S/360 (and S/370* and S/390) processor and associated I/O unit was enclosed in one or more cabinets approximately one meter square and one to two meters tall. The large size of the external I/O units, which consisted primarily of disk and tape storage and communications controllers, required relatively long cables for connection to the processor.

When S/360 was introduced, the I/O interface was the S/360 parallel interface [1] shown on the left side of **Figure 1**. The interface is a bus that is daisy-chained from the processor through each I/O unit to terminating resistor blocks. The physical link and protocols are very similar to those used by the Small Computer System Interface (SCSI) [2]. Over time, the parallel interface was enhanced to accommodate 122 meters of cable and peak data transfer rates of 4.5 megabytes per second.

Each I/O unit can also have more than one interface, allowing it to have both multiple connections to the host processor and connections to multiple host processors. This connectivity improves performance and availability while allowing sharing of I/O units by multiple host processors.

The parallel I/O interface became an American National Standards Institute (ANSI) standard, allowing customers to purchase processors and I/O units from a wide variety of vendors. Having a standard I/O interface also allows customers to change their computing hardware incrementally, since they can purchase processors and I/O units at different times to satisfy different requirements.

In 1990, the ESCON [3–5] I/O interface was introduced (shown in the middle of Figure 1). This interface is point-to-point optical serial, operating at 200 megabits per second and providing a peak data rate of 17 megabytes per second. Each point-to-point link segment can be up to three kilometers long. Increased connectivity is provided by a switch known as the ESCON Director, the largest having 124 ports. The I/O programming interface for ESCON remains almost identical to that used for the parallel interface. The I/O programs build channel programs in memory consisting of channel command words (CCWs), specify channel program addresses and other parameters in operation request blocks (ORBs), and issue start-subchannel instructions that specify the I/O units.

Figure 1 shows an IBM 3990 disk controller attached to an IBM 3390 disk and an IBM 3172 communications controller providing LAN and WAN connections; I/O units may also contain a mixture of parallel and ESCON interfaces, as depicted by the line from the ESCON switch to the bottom left I/O unit.

Parallel Sysplex*, introduced in 1993, requires the higher speed and lower latency provided by the intersystem channel (ISC). The ISC connects S/390 servers to coupling facilities (CFs) using point-to-point optical serial links operating at 1062 megabits per second and providing a peak data rate of 100 megabytes per second. Each point-to-point link segment can be up to three kilometers long. No switches are provided. The programming interface is different from that used by the parallel and ESCON channels and is optimized for message passing. The programs build message command blocks (MCBs), specify the location of these commands and any associated data in message operation blocks (MOBs), and issue send-message instructions that specify the coupling facility devices.

A total of 256 channel interfaces (the sum of all parallel, ESCON, and ISC) is allowed on each processor.

I/O subsystem hardware overview

Figure 2 is a block diagram of the G4 machine with particular emphasis on the I/O subsystem. Each of the boxes in the diagram represents a "book package"; in higher-performance servers, the processor book package is replaced by a multiple-chip module. A book package (or, simply, a "book") consists of a printed circuit card enclosed in a metal case. Books plug into a cage, and up to two cages fit into an enclosure or frame. The enclosures are roughly the size of a household refrigerator. Multiple enclosures can be used when more I/O is required. The processor book (or the multiple-chip module) includes up to 12 processors, with associated caches and connections to the memory. It also contains one or two hub chips called memory bus adapters (MBAs) that provide the connection to the I/O subsystem [6, 7].

Introduced with G3, a new internal machine link (or bus) called the self-timed interface (STI) connects the hub chips in the processor book to a bridge called the internal bus buffer (IBB) book. This new link is a replacement for the upper internal bus (IB), and provides higher bandwidth and longer cable lengths while using fewer chip I/O pins and a much less expensive cable than the IB. Lower IBs are then re-created on the downstream side of this bridge book. The next level of books is attached to the IB as in all previous generations. These books include the channel adapter bridges (CHAs) and the intersystem channel (ISC) books. The ISC is the connection to the coupling facilities (CFs) of the Parallel Sysplex, and each ISC book consists of a "mother" book (ISCM) that houses two "daughter" quarter-sized books (ISCD), providing two ISC interfaces. Finally, the channel adapter books provide channel bus interfaces to the parallel channel (CH3P), the ESCON channel (CH4S), the SCSI, and the Open System Adapter 2 (OSA-2) books. The parallel channel provides the S/390 parallel I/O interface, the ESCON channel provides the optical serial S/390 I/O interface, the SCSI book provides SCSI interfaces for attachment of internal disk drives, and the OSA-2 book provides FDDI, Ethernet, token ring, and ATM interfaces.

Upgrades from the G2 to the G3 machine reuse the channel adapter, parallel channel, ESCON channel, intersystem channel, and OSA-2 books. The G2 IBB book, which adapts the upper IB to two lower IBs, is replaced by the G3 IBB book, which adapts the STI link to two lower IBs. Upgrades from the G3 to the G4 machine leave practically all of the I/O subsystem intact, with the exception of the intersystem channel mother (ISCM) half book. The new ISCM book adds performance improvements to ISC. The SCSI book is offered only in the Multiprise* 2000 line of machines, and these machines have no upgrade path from previous generations.

- *Self-timed interface*

The high performance of the S/390 G3 and G4 machines requires the ability to attach more than one I/O cage; however, the cable length provided by the IB is not long enough to connect a second I/O enclosure. Further, the relatively low performance and high cable cost required the development of a new internal link.

Advances in the density and speed of CMOS chip technology have allowed the development of new system interconnects based on relatively narrow and fast point-to-

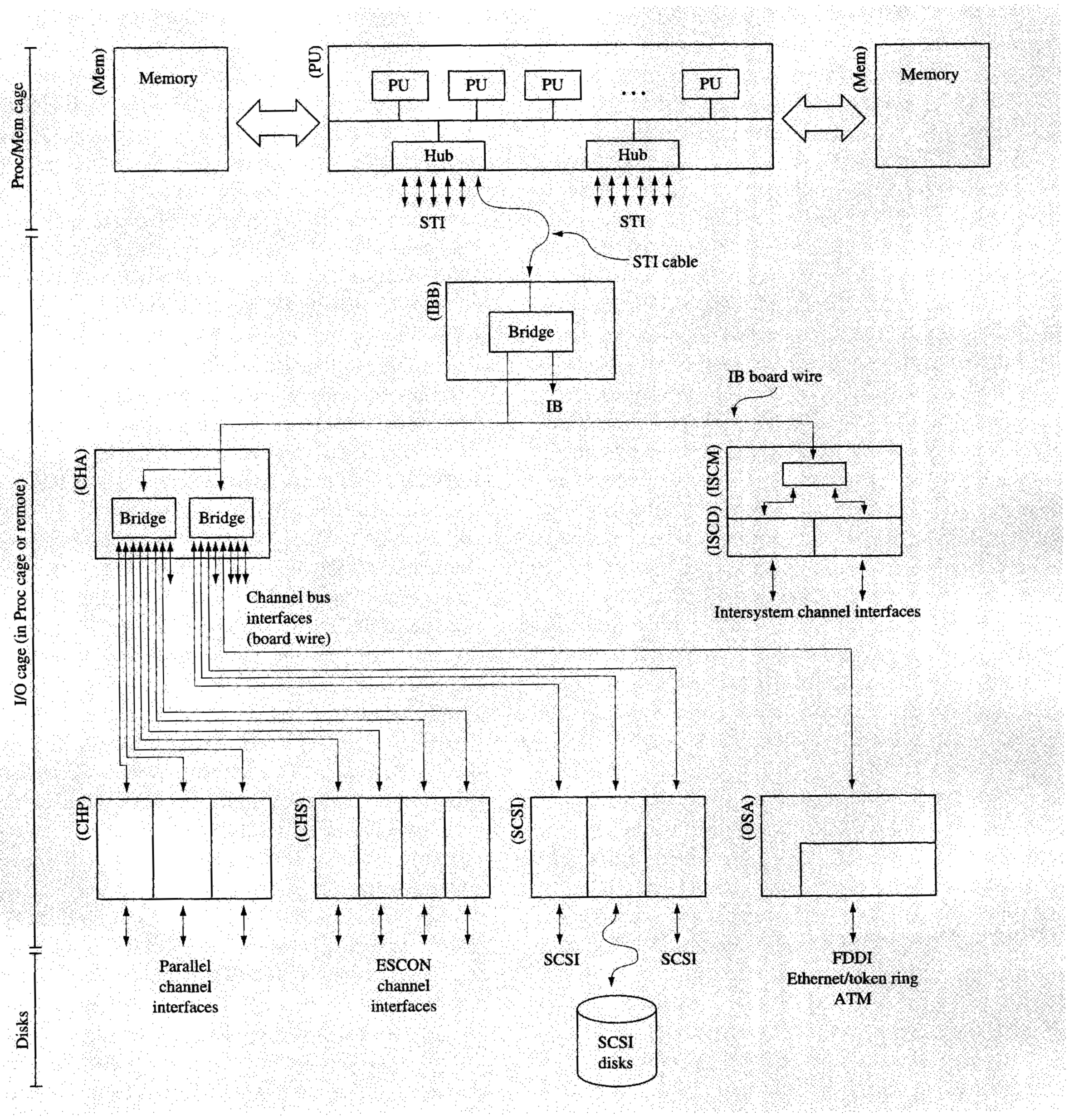

Figure 2

S/390 I/O subsystem.

point links; in the S/390 G3 and G4 machines, these links are called the self-timed interface (STI). These links are directly connected between CMOS VLSI chips and do not require any external cable driver circuits. Other examples of similar links include Tandem's System Area Network (SAN) called ServerNet† [8, 9], the IBM RS/6000* SP2* High-Performance Switch [10], and the IBM RS/6000 SP2 Switch [11]. These links depart from the traditional bit serial designs capable of relatively long-distance communications. Instead, these links use parallel data buses at medium distances, up to tens of meters.

Before STI was introduced in the G3 machines, the G1 and G2 machines used direct IB connections (the upper IB) to the MBA for I/O attachment. The number of IBs that can be driven by the MBAs, the allowed maximum cable length of the IBs, and the data rate of the IBs are all too limited to attach the I/O required by the processing power of the G3, G4, and future machines. STI was developed to overcome these limitations while at the same time significantly reducing the cost of the cable. With the introduction of STI, the CMOS machines have an I/O infrastructure that surpasses that provided in IBM's most powerful bipolar machines.

The upper IB of the G1 and G2 machines is limited to a physical cable length of about one meter. The data bus is bidirectional, with 72 bits (eight bytes plus eight parity bits), and there are several tag bits to control data transfer. The IB cable used in these generations is relatively expensive because of the high number of conductors required. In the G3 and G4 machines, the IB (the lower IB) is confined to the back planes of the cages; cables are not used. The clock rate of the IB is 27 MHz, yielding a peak transfer rate of 296 megabytes per second (in one direction at a time). The high overhead of the data transfer protocol on the IB limits the effective throughput to about 150 megabytes per second.

STI is an interface to memory, and the kinds of information flowing on this interface are almost identical to those flowing on either the IB (upper and lower) or the channel bus interfaces. The primary traffic on STI comprises fetch and store requests to and from main processor memory and is generated by the channels. Additional traffic includes interrupts (in both directions) and an eight-byte mailbox.

The STI physical interface is dual simplex, consisting of ten differential signal pairs in each direction. Eight of the signals constitute a byte of data, and the ninth is a combination tag and parity line. The tenth signal is a half-speed clock; data are sampled on both the rising and falling edges of the clock. Because differential signals are used, each STI port requires 40 chip I/O pins. On the other hand, each IB port requires over 80 chip I/O pins (all single-ended signals). The relatively low number of chip I/O pins required by the STI ports allows six ports to be packaged on each G3 and G4 hub chip, whereas only two IB ports could be packaged on each G1 and G2 hub chip.

The raw data rate of STI is 333 megabytes per second, which means that the minimum pulse width on the link is three nanoseconds. One of the problems when operating a link at these high speeds is the skew between the signals introduced by the differences between the individual cable conductors, the circuits driving and receiving the signals, and the wiring of the chip. In fact, at these speeds with the cables and chip technologies used, this skew can be larger than a clock period (three nanoseconds). With this much skew, the clock signal cannot be used to sample the data signals. To overcome this skew problem, STI is "self-timed" in that the individual data and tag/parity signals are individually time-adjusted in the receiver, with respect to the clock using dynamically adjustable electronic delay lines. The electronic delay lines are tuned in two steps. First, when the STI is initialized, a special timing pattern is transmitted by both sides of the link. This pattern consists of continuous bytes of zeros with an occasional byte of ones. When the receiver recognizes this pattern, it makes a crude adjustment of the electronic delay lines so that all bits of the byte of ones are sampled in the same byte time. The second tuning step monitors the rising and falling edges of the data and tag/parity signals. The electronic delay lines are continually adjusted to keep the sample time in the middle of the data-valid window.

Even though the raw data rate of STI is 333 megabytes per second, a proper method of controlling the flow of packets is still required to keep the effective data rate high. In the case of the IB, achieving a high utilization is difficult because of the bus structure itself. The IB has multiple nodes, and these nodes must arbitrate for the use of the bus. While these schemes allow multiple nodes, time is lost in the arbitration process. Once the bus is "granted" to a node, the request is transferred. To keep the bus free for other nodes to use between the time the command is transferred and the time the response is sent, the bus must arbitrate a second time to send the response.

The point-to-point nature of STI avoids the bus arbitration of the IB. STI uses a packet-oriented, "credit-based" flow control, which means that packets are sent only when there is sufficient buffer space at the receiver. Each request or response is assembled into a packet with header, header-checking, payload, and payload-checking fields. Multiple header and payload buffers at the sender and receiver allow the credit-based flow control to have multiple packets in flight at the same time. When there are enough send and receive buffers to accommodate the round-trip delay of the link, packets can be transmitted back to back, achieving the theoretical data transfer limit of the link. Raw link speed, logic delay, and link distance are all factors in determining the required number of buffers. In STI, there are enough buffers to tolerate distances of tens of meters without a loss in data transfer rate. The return-trip interlock is provided by credits that are passed back to the sender in the form of specialized short packets called link control words (LCWs).

Yet another problem with operating an interface at such high speeds and long distances is that the soft-error rate due to electrical noise is higher than with the IB. As a result, the error detection and recovery mechanisms are quite robust. The error checking of the packets is a powerful combination of byte-wide parity (the tag/parity signal) and a longitudinal redundancy check (LRC). The last four bytes of every header and payload constitute the LRC. If an error is detected in a packet header, sequence numbers in the header allow the hardware to automatically resend the packet. Packets in the sender buffers are not purged until the receiver acknowledges the safe arrival of the header. Errors in the payload are handled differently than errors in the header and are not automatically retried by the STI link hardware. If the header is received correctly and there is an error in the payload, the originator of the packet can be accurately identified and informed. It is then up to the originator of the damaged packet to perform the appropriate recovery.

- *Chip technology remaps*

To reduce cost and physical size and to improve reliability, portions of the ESCON, CHA, and ISC have been remapped into CMOS 5L. Where appropriate, the S/390 developers made a logic gate for gate remap without changing the function in any way. This approach reduced the redesign effort, minimized the microcode rewrite, and lessened the test time. However, in any technology remap, the developers usually find some justifiable changes to make along the way. The three remap projects are discussed below.

ESCON low cost (LC): Remap and digital serdes

The original ESCON channel CMOS hardware was remapped from three CMOS 2S logic chips and three array chips into a single CMOS 4S chip in 1992. The interfaces between the multiple chips of the earlier design were simplified in the single-chip design, but these design changes required a full simulation of the new design.

The single-chip ESCON design still requires a separate high-speed bipolar module known as the "serdes" (serializer/deserializer) for the clock recovery (analog phase-locked loop), serialization, and deserialization functions. The serdes serializes the 10-bit interface from the ESCON chip to a 200-megabit serial bit stream to drive the optical transceivers. The serdes also receives the serial bit stream from the optical transceivers, extracts the clock information, and deserializes the bit stream for the 10-bit interface to the ESCON chip. This bipolar module is relatively expensive and consumes roughly a third of the power of the entire ESCON channel.

The second ESCON remap became available on the G2 machines. It eliminates the separate bipolar serdes by incorporating a new CMOS serdes design into the ESCON chip. The CMOS design for the serdes uses a digital phase-locked loop to extract the clock information from the received serial bit stream. The phase-locked loop uses digital phase comparators to compare the received data edges to the transmitter clock (the local clock). While the remap itself from CMOS 4S to CMOS 5L reduced cost and improved reliability, the elimination of the bipolar serdes made a much larger improvement in the reliability and reduction of cost. The elimination of the separate serdes allows more ESCON channels to be packaged in each book. This new chip, known as ESCON low cost (LC), allows increasing the number of channels packaged in a book from three to four.

The logic verification of ESCON LC was performed in two independent steps. The new digital serdes design was fully simulated using test cases developed to exercise all of the serdes functions including acquisition of bit synchronism. Verification of the portion remapped from CMOS 4S avoided the complexities of a logic simulation. Rather, the remapped portion was verified through pure Boolean equivalence, a very fast process. Time-consuming test-case development and models of the operating environment are not required.

Channel adapter chips: Redesign and higher density

To complement the increased number of ESCON channels available in a single book package, the number of channel bus ports provided by channel adapter books is increased from six to eight. The earlier channel adapter design uses a pair of CMOS 2S chips. Each chip adapts the IB to three channel bus interfaces. The new chip is implemented in CMOS 5L, and is mostly a remap from the CMOS 2S version with the number of ports increased from three to eight.

- *Intersystem channel: Data movers*

The analysis of future Parallel Sysplex workloads shows increasing bandwidth demands on the intersystem channels [12, 13]. The design of the ISC used in the G1, G2, and G3 machines requires frequent intervention by the host processors during data transfer. With the anticipated new workloads, the data transfer rate is not sufficient, and the high processor utilization becomes unacceptable. By adding more function to the host adapter portion of the ISC, part of the ISCM book and most of the message setup and data transfer can be offloaded from the processors.

The G1 through G3 ISC host adapter chip is implemented in CMOS 2S, and there is one chip for each ISC. Because of the aging CMOS 2S technology, a redesign in CMOS 5L with all of its associated cost reduction and reliability improvement was due for the G4 machines, and is now available for G3 as well. With the new design, the higher density of CMOS 5L allows packaging two host adapters in the same chip.

The ISC host adapter performance improvements come from changes to the implementation of the send-message instruction. This instruction is used by S/390 servers to send messages to a coupling facility (CF). The ISC in the S/390 processor is called a sender, and the ISC in the CF is called the receiver. The send-message instruction has two variations. The synchronous send-message instruction stops execution of subsequent instructions until a response is received from the CF. This variation of the instruction is used when there is little or no data to be transferred. The asynchronous send-message instruction allows the processor to continue to the next instruction before a response is received, and the program periodically executes polling instructions to determine when the response has been received. This second variation of the instruction is used for long data transfers.

There are two types of processors within S/390 CMOS servers. Central processing units (CPUs) execute S/390 programs, and system assist processors (SAPs) execute system support code and do not execute S/390 programs. Both types are physically the same. In earlier implementations of the ISC, the synchronous send-message instruction is executed by a CPU, and the asynchronous send-message is executed by a SAP. The SAP is used in the asynchronous variation of the instruction because a processor is continually required through the long data transfers typically performed by this variation of the instruction. With the G4 machine, putting more function in the ISC host adapter not only reduces the CPU utilization, but also allows the CPU to perform the front-end processing for both synchronous and asynchronous send-message instructions. Keeping the asynchronous send-message instruction processing in the CPU avoids the inefficiency of passing work from the CPU to the SAP.

One change made to the ISC host adapter adds addressing logic that allows the CPU to immediately prepare the ISC to receive the response from the CF without waiting for the response to arrive from the CF. The ISC hardware can now store the response in memory before alerting the CPU or SAP. This function reduces the latency of the send-message instruction by freeing the processor from the task of storing the response.

The second change is used by both the sender and receiver ISCs and improves the performance of send-message instructions with data transfer. With this change, the CPU generates lists of addresses in main memory and sends the address of this list to the ISC, along with several parameters describing the data transfer. The ISC then fetches the addresses and uses them to either transmit or receive data depending on the kind of operation requested. In the receiver ISC, this function allows the CF to efficiently scatter and gather data on relatively small memory boundaries. Earlier designs require the CF to move the data multiple times. This new function creates a new I/O programming model for the CF; however, the improvement in performance is worth the development cost for the changes to the CF code. Now all of the data transfer associated with a message can be performed without intervention by either a CPU or SAP. Although the G4 machine has a new implementation for the ISC, the link architecture remains unchanged, and the G4 ISCs are compatible with all other ISC implementations.

Internal communications and disk storage

The primary reason for integrating communications adapters and disk storage into S/390 servers is to reduce cost, especially for entry systems. Without S/390 internal I/O, customers must purchase not only the S/390 server but also the stand-alone external boxes for their communications and disk storage. These boxes are attached to the S/390 server with either an S/390 parallel or ESCON channel, as shown in Figure 1. Now, storage and communications I/O can share the same physical structure with the rest of the system, reducing the number of external, stand-alone enclosures.

Integrating I/O also improves performance and reliability. Fewer parts are required when the I/O is integrated into the system's power, cooling, and service infrastructure. At the same time, particularly in the case of internal storage, the performance is improved by making design trade-offs now possible by better exploiting internal hardware interfaces.

Once the I/O is integrated, further customer value is found in the integration of the service system. With traditional S/390 external I/O, the service system of the I/O is separate from that of the S/390 server, adding more cost and operational complexity for the customer.

The implementation philosophies for internal communications and internal disk storage are quite different. The development of internal communications adapters takes the approach of keeping the functions that were performed in the external I/O units in separate hardware, and packaging this hardware in books plugged directly into S/390 cages. Internal disk storage, on the other hand, emulates many of the control unit functions that were performed in the external I/O units in one of the system assist processors (SAPs) and main memory.

• *Internal communications adapters*

Before internal communications adapters became available, S/390 customers could only use external stand-alone communications controllers such as the IBM 3172 shown in the middle of Figure 1. The first generation of S/390 internal communications adapters, called Open System Adapters (OSA-1), are designed to be integrated into both the bipolar and CMOS servers. The OSA-1 cards are physically quite large and require a special cage different from those used by the CMOS books. IBM is now delivering the second generation of internal communications adapters, known as OSA-2. The OSA-2 adapters are packaged in the standard S/390 books and do not require a special cage. These adapters allow direct connection to networks, and there are three different adapter types. The first supports Ethernet and token ring (ENTR), the second supports the fiber-distributed data interface (FDDI), and the third supports asynchronous transfer mode (ATM) [14].

The approach to OSA-2 is to design hardware and code that emulate the IBM 3172 interconnect controller attached to an ESCON channel. The S/390 I/O

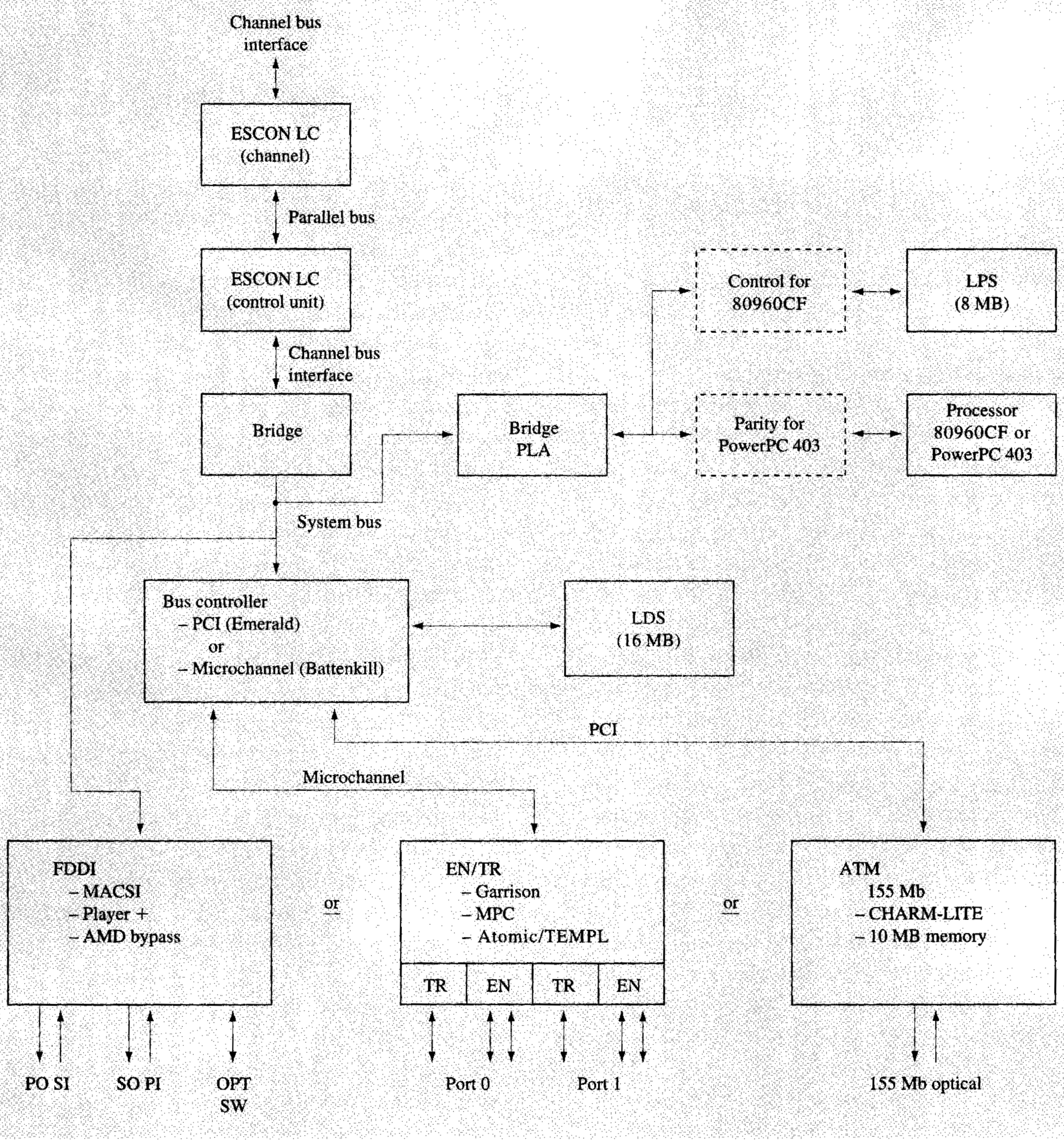

Figure 3

Internal communications adapters.

programming model remains unchanged. OSA-2 books are integrated into the S/390 server's power, package, cooling, and service infrastructure.

OSA-2 hardware

The three different OSA-2 designs are shown in a composite block diagram in **Figure 3**. All three adapters connect to the system through a pair of back-to-back ESCON LC chips providing the channel function and error isolation. The channel bus interface connects to an ESCON LC channel chip that has normal channel code with a few modifications. This chip is connected directly to a second ESCON LC chip using 10-bit parallel electrical links, bypassing the serdes functions (recall that these functions were integrated into the ESCON chip). Bypassing the serdes reduces the power and cooling requirements.

Even though the serdes functions are not used and the interface between the two ESCON LC chips is not serial, the information still follows the ESCON link architecture. To help debug the OSA-2 designs, provisions have been made to attach ESCON interface trace tools to the

Table 1 OSA-2 comparison.

	FDDI	*ENTR*	*ATM*
Adaptation processor	Intel 80960CF	Intel 80960CF	IBM PowerPC* 403GA
Bus connection for front end	System bus	Microchannel	PCI
Number of ports	2	2	1
Bit rate (Mb)	100	10/16	155
Bus driver chip set	Battenkill	Battenkill	Emerald
Front-end chip set	National Semiconductor MACSI, Player+	IBM Garrison Atomic National Semiconductor TEMPL	IBM CHARM-LITE
Protocol	TCP/IP SNA	TCP/IP SNA	LAN emulation (EN/TR, TCP/IP)

interface between the ESCON LC chips. Since the trace tools accept only serial data stream, the interface between the two ESCON LC chips may be operated in serial mode by changing a few jumpers on the OSA-2 book.

The second ESCON LC chip is loaded with control unit code and emulates the front end of a control unit. The channel bus interface of the control unit ESCON LC chip is attached to a bridge chip that connects to a relatively standard system memory bus.

From this bridge chip, the three OSA-2 designs diverge. **Table 1** shows these differences. All three designs have an adaptation processor. In the case of the ENTR and FDDI adapters, an Intel 80960CF processor is used, and the ATM adapter uses an IBM PowerPC 403GA embedded controller. Both processors have an eight-megabyte local program store (LPS) (or memory). The bridge chip also attaches to a memory controller chip that provides access to a 16-megabyte local data store (LDS) (or memory) and provides a connection to the ENTR and ATM logic. Several programmed logic devices (PLDs) are used to convert bus protocols, arbitrate bus access, and control flash memories.

All three front ends (FDDI, ENTR, and ATM) are shown at the bottom of Figure 3, though any particular adapter has only one of these front ends.

The FDDI adapter front end uses a collection of chips from National Semiconductor. The system bus connects to the medial access controller/system interface (MACSI) chip and two enhanced physical layer controller (Player+) chips. The chip set implements the physical layer defined by FDDI ANSI X3T9.5 and operates at 100 megabits per second. An AMD optical bypass switch allows the FDDI station to be offline without affecting the rest of the ring.

Each ENTR adapter contains two independent front-end chip sets that drive two independent Ethernet or token ring ports. Each port can be independently configured as either Ethernet or token ring, but not both at the same time. The front-end chip sets, with the exception of the TEMPL chip from National Semiconductor, are designed and manufactured by IBM. An IBM microchannel (MC) bus is provided by a bus controller chip called Battenkill, and this bus connects to a Garrison chip in the ENTR front-end logic. The Garrison chip converts the microchannel bus to an internal bus used by the rest of the ENTR front-end chip set. This internal bus connects to flash memory and to the MPC chip. The MPC chip contains an embedded special-purpose processor that handles the media access control (MAC) frames and moves the logical link control (LLC) frames. Attached to the MPC chip are the TEMPL and Atomic chips. The TEMPL chip supports the Ethernet physical layer defined by IEEE 802.3 and operates at 10 megabits per second. The Atomic chip supports the token ring physical layer defined by IEEE 802.5 and operates at 16 megabits per second.

The ATM OSA-2 adapter contains a "short" peripheral component interconnect (PCI) card attached to a PCI bus. The PCI bus is provided by a bus controller chip called Emerald, and the ATM PCI card is an IBM design. At the heart of the ATM PCI card is the CHARM-LITE chip. Also attached to the CHARM-LITE chip is a total of 10 megabytes of DRAM, flash memory, and the ATM transmitter and receiver. The ATM OSA-2 adapter operates at 155 megabits per second (OC-3).

All three OSA-2 adapters fit into a single book package. In the case of FDDI and ENTR, the book contains a single printed circuit board. With ATM, the front end is a short PCI card that plugs into a socket on the main circuit card containing the rest of the adapter logic.

External adapters such as the IBM 3172 usually have a control panel that can be used by the customer or service personnel to alter some of the configuration information such as the local MAC addresses, and provide hardware debugging tools such as trace information. With OSA-2, the S/390 service processor is used to provide the OSA-2 control panel. Implementation of this control panel function requires a communication path from the service processor to OSA-2. Since the only connection is over the channel bus interface and through the back-to-back ESCON LC chips, a relatively simple communication path is a service device within OSA-2 that operates using the ESCON protocols. This service device uses a dedicated unit address that is not visible to the S/390 operating system and is accessible by only the S/390 service processor. Small changes had to be made to the ESCON LC channel code so that it could transfer data to and from the hardware system area where the service system operates.

OSA-2 code

All three OSA-2 adapters support both Systems Network Architecture (SNA) and Internet Protocol (IP) network protocols. The ATM adapter supports these network protocols through LAN emulation of Ethernet and token ring.

As in the IBM 3172 communications controller, some of the protocol stack for SNA is in the OSA-2 adapter, while IP operates in "pass-through" mode, in which all of the protocol stack is in the host software. An advantage of OSA-2 over the IBM 3172 communications controller is its logical partition (LPAR) support for TCP/IP. LPAR gives S/390 servers the ability to run multiple operating systems (or images) concurrently. With the 3172 communications controller, a LAN port running TCP/IP can be attached to only a single partition. In OSA-2, a routing table is provided to examine the IP address information in the received LAN packets or frames and route them to the correct partition using the ESCON multiple-image facility (EMIF) over the ESCON LC to ESCON LC interface. Another advantage of OSA-2 is that with SNA it can support 2047 link stations rather than the 255 supported by the 3172 communications controller.

Almost all of the OSA-2 code design is based on the 3172 communications controller. Porting the design reduced the development time and improved the compatibility with the host software. The big differences are in the operating environment for the code and in the device drivers for the front-end chip sets.

• *Internal disk storage*

Traditional S/390 external disk storage

Figure 4 shows the differences between traditional S/390 external and internal disk storage. Figure 4(a) shows the traditional external disk storage, in which an S/390 server is connected to a disk control unit (CU) and disks via an external I/O interface such as parallel or ESCON. The optional "fabric" shown may consist of ESCON directors (or switches).

The external disk storage subsystem is shown below the S/390 server in Figure 4(a). The CU portion is typically an IBM 3990 storage controller, and it drives cabinets of disk drives, typically IBM 3380 or 3390 direct-access storage disks [15, 16]. The CU interprets the channel protocol from the S/390 server and translates it into the protocols understood by the disk drives. The CU also contains a large disk cache whose capacity is typically in the gigabyte range. This cache performs the usual function of storing frequently accessed read data. In some models of the CU, where the cache is properly protected against hardware and power failures, write data may also be cached, improving performance by allowing I/O write operations to complete from the point of view of the program before the data are actually written on the physical disk drive(s).

When an I/O operation to external disk storage is performed, the S/390 server sends the I/O requests through the SAP, to the channel, over the I/O interface, and to the CU. If the operation is a read and the data are in the cache, the data are quickly returned to the S/390 server without accessing any of the disk drives. If the data are not in the cache, at least one of the disk drives must be accessed before the data can be sent back to the S/390 server. After the data and ending status are received, the S/390 server informs the requesting program that the I/O operation has completed.

S/390 disk storage arranges data on the disk in variable-length fields using a device architecture called extended count, key, data (ECKD*) [17]. In this scheme, the data are written onto the disk in several fields separated by small gaps. A field is the smallest segment of data that can be written onto the disk. The count field specifies the length of the key and data fields, while the key field contains a user-definable handle and the data field contains the actual user data. The IBM 3390 disk directly implements the ECKD architecture. More modern disk drives are strictly fixed-block devices with a typical block size of 512 bytes, and there are no distinct count and key fields. To be compatible with the S/390 I/O programming, disk storage subsystems using fixed-block disk drives emulate the ECKD architecture in the CU. The IBM 9390 (RAMAC*) [18] disk storage subsystem is an example of this implementation.

S/390 internal disk storage

The design approach to internal storage, shown in Figure 4(b), is quite different from the approach used with internal communications adapters. Rather than keeping the control unit functions in the adapter, the internal disk storage adapts the internal channel bus interfaces to SCSI, uses part of main memory for the large cache and its

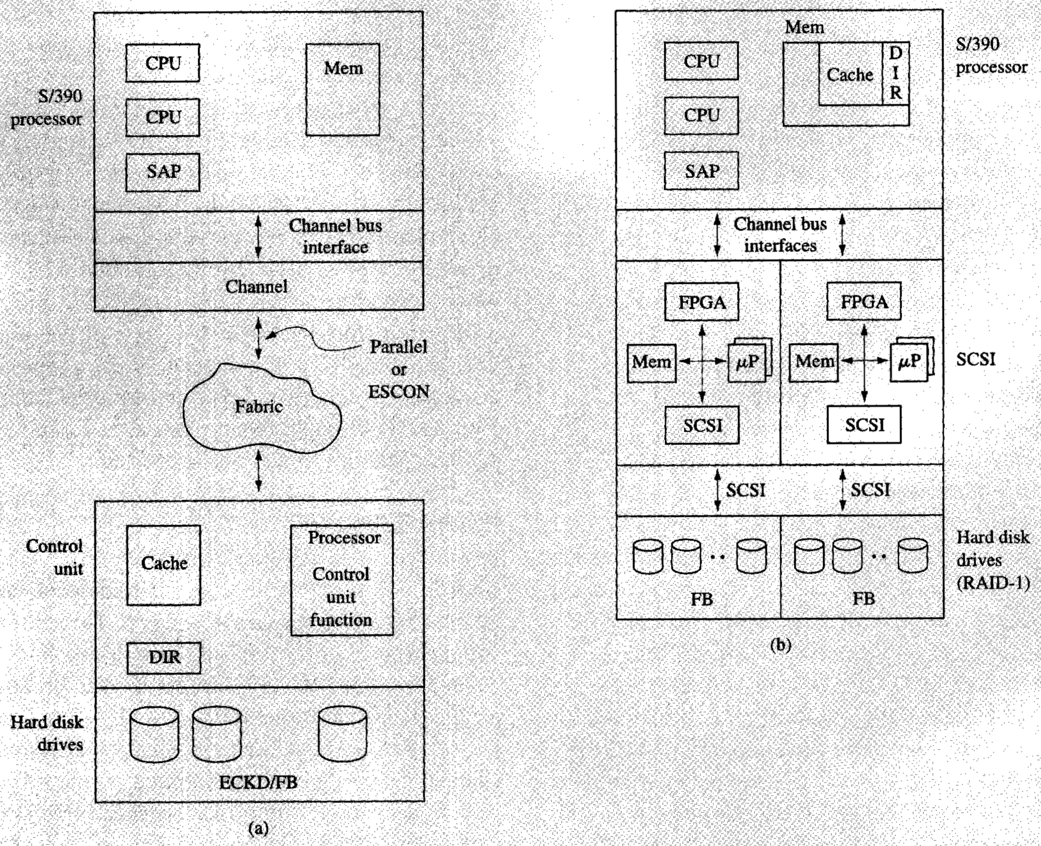

Figure 4

(a) Traditional and (b) internal S/390 disk storage.

directory, and emulates most of the control unit and channel function in one of the system assist processors (SAPs). Standard off-the-shelf SCSI disk drives are used, and since all SCSI disk drives are fixed-block, ECKD is emulated in the SAP code, and the track-mapping tables are stored on the disk drives.

The SCSI book, used to adapt the channel bus interface to SCSI, is shown in the middle of Figure 4(b). Each SCSI book has three independent channel bus interfaces to SCSI adapters (two are shown in the figure). All of the adapter components are off-the-shelf; no application-specific integrated circuits (ASICs) are used. The connection to the channel bus interface is implemented using a Xilinx 40xx series field-programmable gate array (FPGA) chip. This chip is connected to a duplexed ARM60 microprocessor, a Western Digital 33C95 SCSI controller chip, and a 128-kilobyte memory for the ARM microprocessor. A few additional components such as an EPROM, bus drivers, and passive components are also on the SCSI book. The primary function of the adapter is to receive commands from the SAP, fetch S/390 memory address lists from S/390 main memory, translate addresses from the SCSI controller chip's memory space to the S/390 main memory, and post ending status to the SAP. The adapter maintains the high level of error checking required by S/390 products through error-checking circuits in the FPGA chip and the duplexed ARM microprocessor; one ARM microprocessor checks the other. As a result, all stores and fetches to and from the S/390 main memory and all communication with the S/390 SAPs are checked for errors. Any possible malfunctions in either the SCSI adapter chip or the SCSI disk drive are isolated to the current operation and cannot disturb any of the other critical areas of the S/390 server. This level of error checking and isolation is a hallmark of S/390 servers.

Each SCSI disk drive [shown at the bottom of Figure 4(b)] is packaged in a carrier that contains a dc-to-dc power converter. The carriers are then plugged into drawers that are separate from the logic cages, and each drawer houses up to 16 disk drives. The drawer senses the replacement of a disk drive to invoke the appropriate maintenance and/or recovery actions. Up to four drawers are supported in half a frame. With disk mirroring (RAID-1) and nine-gigabyte disk drives, the maximum capacity is approximately 288 gigabytes.

Referring once again to Figure 4(b), an I/O operation starts when a CPU issues a start-subchannel instruction. The CPU microcode queues the operation to the SAP. When the SAP dequeues the work, it determines that the operation is not for an external I/O unit, such as disk storage or communications adapter, and directs it to internal disk storage. At this point, the channel and control unit emulation code take over.

The internal disk storage cache and associated directories are located in a special portion of main memory known as the hardware system area (HSA). HSA is not accessible by S/390 programs and is only accessible by internal system elements such as CPUs, SAPs, and channels. All operations to the internal disk storage go through the cache. If the operation is a read, the SAP examines the directory to determine whether the requested data are in the cache. If they are, the data are simply copied from one area of main memory (the cache in HSA) to another area of main memory (the area specified by the requesting program). This relatively fast and simple copy of data from one area of memory to another is the reason why internal disk storage has such a large performance advantage over external disk storage subsystems. If the requested data are not in the cache, the SAP initiates an operation to the disk drive(s) by putting work on the queue for the SCSI adapter. The ARM microprocessor dequeues the work, builds the appropriate SCSI commands, and sends them to the disk drive via the SCSI adapter chip. The ARM microprocessor also prepares the S/390 memory addresses in the FPGA chip for data transfer from the disk drive(s), through the SCSI chip, and up to the cache in the S/390 hardware system area of main memory. When data transfer completes, the ARM microprocessor queues status information back to the SAP. The SAP dequeues the status and moves the data from the cache to the program area. In a write operation, the SAP always stages the data in the cache before sending them to the disk drives.

To reach the high levels of data integrity found in S/390 products, internal storage implements RAID-1 (disk mirroring) [19]. All disk drives are in pairs, and write operations write to both disk drives. The performance of read operations is improved, since both disk drives of a pair can be accessed simultaneously for two different read operations. The disk drive chosen for a particular read operation is based on a utilization algorithm. If a data error is detected when a disk drive is read, the alternate disk drive is read. If this operation is successful, the data are rewritten to the disk drive in error. Errors during write operations are handled by the disk drive itself using spare area on the disk drive.

As with other parts of the system, maintenance, repair, and upgrades are handled by the S/390 server's system service element. For example, the concurrent replacement philosophy used for channels, power, etc. is applied to the SCSI adapter books and the disk drives. Functions unique to internal storage have also been added. For example, when a disk drive is replaced, the data are automatically copied to it from the other disk drive of the mirrored pair. The SCSI books can also be concurrently replaced if both disk drives of a mirrored pair are not using adapters on the same SCSI book. Finally, upgrades adding SCSI adapters and disk drives are supported.

Future S/390 I/O

In the computer industry, PCI has emerged as the standard interface for attaching I/O adapters. New PCI adapter cards are constantly being developed for new communications protocols and disk drives. For communications, 100-megabit-per-second Ethernet, one-gigabit-per-second Ethernet, ATM OC-12 (622 megabits per second), and one- and two-gigabit-per-second fibre channel (FC) PCI cards are coming on the market. New disk drives with new interfaces such as serial storage architecture (SSA) and fibre-channel-arbitrated loop (FC-AL) are driving the development of new PCI cards.

Integrating these new PCI cards into the S/390 servers will require a new hardware and code structure. While the OSA-2 ATM approach works well for ATM OC-3 and perhaps 100-megabit Ethernet PCI cards, both the data rates and physical package preclude attachment of newer PCI cards. The OSA-2 book allows only short PCI cards, and the channel bus attachment to the S/390 server limits the data rate to 18 megabytes per second in one direction at a time. The hardware structure for internal disk storage is also not a good model for PCI attachment. It is limited by the book package and the channel bus data rate, and it would be difficult to introduce a PCI interface to that design point.

From the point of view of hardware, the attachment of newer PCI cards requires the physical space for full-sized PCI cards and a data rate to keep up with these cards. For example, a one-gigabit-per-second fibre channel PCI card requires about 100 megabytes per second of bandwidth to the S/390 server. Beyond that, the two-gigabit-per-second fibre channel adapters will require double-width (64-bit) PCI interfaces.

Once the physical and bandwidth requirements for newer PCI cards are met, attachment to S/390 will require an adaptation layer to convert the protocols used by the PCI cards to the S/390 I/O programming model. Industry-standard PCI cards are designed for the UNIX** and Personal Computer (PC) I/O model. In this model, I/O requests use programs called device drivers to control the PCI cards. The PCI cards directly access all of their host's memory. Typically, request and response queues and data buffers are allocated in main memory. When the device driver is directed to perform an I/O operation, it builds the request in main memory and sends a signal to the PCI card, which fetches the request and starts the I/O operation. If any data are to be moved, the request includes the main memory addresses. When the I/O operation has completed, the PCI card stores the response in a queue in main memory. An interrupt may be sent back to the device driver to indicate that a response has been placed in main memory.

The S/390 I/O programming model [20] is much more restrictive than the UNIX and PC I/O programming models. All S/390 I/O is performed over an I/O interface to an I/O unit such as an IBM 3172 communications controller or 3990 disk control unit described earlier, and these I/O units have no direct access to main memory. The channel accesses main memory on behalf of the I/O unit. The channel sends requests to the I/O unit as single-byte commands that may be further defined by data fields, and these requests are streamed out over the I/O interface. Data transfer is also streamed over the I/O interface to or from main memory, and the responses are single bytes called ending status. No main memory addresses are present on the I/O interface. The programming model defines control blocks in main memory that contain the requests, data addresses, and ending status bytes.

When attaching a PCI card to an S/390 server, the adaptation layer must receive the requests and reformat them into a request queue accessible to the PCI card. Data transfer also requires some level of adaptation. Typically, the PCI memory address space is mapped to the S/390 server's memory address space either through address translation tables or through a store-and-forward memory in the adaptation hardware such as in the local data store in the OSA-2 adapters. A response queue in the adaptation hardware is accessible by the PCI card, and the adaptation hardware translates these responses into S/390 ending status.

When a single operating system image is controlling the PCI card, the adaptation process is relatively straightforward. However, S/390 logical partitioning (LPAR) complicates this adaptation process by allowing multiple logical partitions to share the PCI card. The nature of the PCI bus and all of today's PCI adapter cards support only one operating system image. At present, with internal disk storage, the sharing is managed by the SAP. The OSA-2 sharing is managed by either the PowerPC 403 or the Intel 80960CF adaptation microprocessors. To preserve the S/390 I/O programming model when integrating PCI cards, an adaptation layer will always be required.

In the future, faster adaptation processors will be required in order to integrate faster PCI cards into the S/390 server. Such processors add cost and complexity, and it may become desirable to eliminate the adaptation layer by adopting a more direct interface between programming and PCI cards. Such a direct interface would allow device drivers compiled into the S/390 instruction set to communicate directly with adapters through control blocks in memory, and the equivalent of a programmed I/O operation to give initiative to the adapter, similar to the way in which UNIX and PC I/O operate. This direct interface would not replace the S/390 I/O programming model, but would be added as an alternative for new programs and applications. The new I/O programming required for a direct interface may be less costly than adapting industry-standard I/O hardware to the S/390 I/O programming model. However, even with a direct interface I/O programming model, the requirement for LPAR sharing of PCI cards still requires an adaptation layer.

Another area of future S/390 server I/O development centers around the future role of the STI links. The bandwidth provided by STI makes it a natural choice for connecting chips bridging to PCI buses. Another area of interest is using STI links to connect S/390 servers directly. With the addition of data movers in the hub chips, STI links between S/390 servers can be used for interprocessor communication. Finally, STI link data rates will be improved.

Conclusions

S/390 has made some important first steps in using industry-standard components to develop its I/O structure. Development time and cost have been reduced by integrating communications adapter chip sets, a PCI card, and SCSI disk drives into the S/390 server. The development of the STI link has provided the bandwidth and physical distance required to connect the increased amount of I/O needed to support the high performance of the S/390 G3 and G4 servers. Remaps and redesigns into lower-cost, faster, and more reliable technologies have greatly improved the cost/performance of S/390 servers.

Integration of industry-standard components, redesigns of existing elements, and development of new internal links will continue in a context of maintaining the attributes of the S/390 computing environment that its customers have come to expect.

Acknowledgments

Design of the S/390 server I/O required the efforts of many people in the Poughkeepsie and Boeblingen laboratories over the past several years. Special acknowledgment goes to Dan Casper and Frank Ferriaolo for their pioneering work on STI and to Marten Halma for his dedication to internal disk storage. I would also like to thank Luke Hopkins for his help in describing OSA-2 and Lisa Spainhower and Guru Rao for their review of this manuscript.

*Trademark or registered trademark of International Business Machines Corporation.

**Trademark or registered trademark of X/Open Co., Ltd.

†ServerNet is either a registered trademark or a trademark of Tandem Computers Incorporated in the U.S. and/or other countries.

References

1. *IBM System/360 and System/370 I/O Interface Channel to Control Unit Original Equipment Manufacturers' Information*, Order No. GA22-6974; available through IBM branch offices.
2. W. D. Schwaderer and A. W. Wilson, Jr., *Understanding I/O Subsystems*, First Edition, Adaptec, Inc., Milpitas, CA, 1996, pp. 113–129.
3. J. R. Flanagan, T. A. Gregg, and D. F. Casper, "The IBM Enterprise Systems Connection (ESCON) Channel—A Versatile Building Block," *IBM J. Res. Develop.* **36,** 617–632 (1992).
4. J. C. Elliott and M. W. Sachs, "The IBM Enterprise Systems Connection (ESCON) Architecture," *IBM J. Res. Develop.* **36,** 577–591 (1992).
5. *Enterprise Systems Architecture/390 ESCON I/O Interface*, Order No. SA22-7202; available through IBM branch offices.
6. G. Doettling, K. J. Getzlaff, B. Leppla, W. Lipponer, T. Pflueger, T. Schlipf, D. Schmunkamp, and U. Wille, "S/390 Parallel Enterprise Server Generation 3: A Balanced System and Cache Structure," *IBM J. Res. Develop.* **41,** 405–428 (1997, this issue).
7. T. Schlipf, T. Buechner, R. Fritz, M. Helms, and J. Koehl, "Formal Verification Made Easy," *IBM J. Res. Develop.* **41,** 567–576 (1997, this issue).
8. R. W. Horst, "TNet: A Reliable System Area Network," *IEEE Micro*, pp. 37–45 (February 1995).
9. W. E. Baker, R. W. Horst, D. P. Sonnier, and W. J. Watson, "Flexible ServerNet-Based Fault-Tolerant Architecture," *Proceedings of the 25th International Symposium on Fault-Tolerant Computing; Digest of Papers—International Symposium on Fault-Tolerant Computing 1995*, IEEE, Piscataway, NJ, Order No. 95CH35823, pp. 2–11.
10. *SP2 High-Performance Switch—SJ34-2*, Order No. G321-5564; available through IBM branch offices.
11. *IBM POWERparallel Technology Briefing*, URL: *http://www.rs6000.ibm.com/resource/technology/sp_sw2/spswp2_1.html.*
12. J. M. Nick, J. Chung, and N. S. Bowen, "Overview of IBM System/390 Parallel Sysplex—A Commercial Parallel Processing System," *Proceedings of the IEEE Symposium on Parallel and Distributed Processing*, 1996, pp. 488–495.
13. C. L. Rao and C. Taaffe-Hedglin, "Parallel Sysplex Performance," *CMG Transactions*, No. 87, Winter 1995, pp. 3–7.
14. D. E. McDysan and D. L. Spohn, *ATM: Theory and Application*, McGraw-Hill Book Co., Inc., New York, 1994.
15. *IBM 3880 Storage Control Model 23*, Order No. GA32-0083; available through IBM branch offices.
16. *IBM 3990 Storage Control Reference*, Order No. GA32-0099; available through IBM branch offices.
17. *Introduction to Direct Access Storage*, Order No. ZR21-3208; available through IBM branch offices.
18. *IBM RAMAC Array Subsystem Introduction*, Order No. GC26-7004; available through IBM branch offices.
19. David A. Patterson, Garth Gibson, and Randy H. Katz, "A Case for Redundant Arrays of Inexpensive Disks (RAID)," *ACM SIGMOD Conference Proceedings*, Chicago, June 1–3, 1988, pp. 109–116.
20. *Enterprise Systems Architecture/390 Principles of Operation*, Order No. SA22-7201; available through IBM branch offices.

Received December 18, 1996; accepted for publication April 25, 1997

Thomas A. Gregg *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (GREGG at PKEDVM9, tomgregg@us.ibm.com).* Mr. Gregg is a Senior Technical Staff Member in the S/390 System Design group. He received an SC.B. degree in engineering from Brown University in 1972 and continued his studies under a university fellowship, receiving an SC.M. degree in electrical engineering in 1974. He joined IBM at the Poughkeepsie Laboratory in 1973. Mr. Gregg has held various technical positions in the area of I/O subsystem design. He holds numerous patents utilized in IBM ESCON and Intersystem Channel products, and has received eight IBM Invention Achievement Awards. He received an IBM Outstanding Innovation Award and an IBM Corporate Award for work on ESCON products, and an IBM Outstanding Innovation Award for work on Intersystem Channel products.