The structure of chips and links comprising the IBM eServer z990 I/O subsystem

E. W. Chencinski
M. J. Becht
T. E. Bubb
C. G. Burwick
J. Haess
M. M. Helms
J. M. Hoke
T. Schlipf
J. M. Turner
H. Ulland
M. H. Walz
C. H. Whitehead
G. Zilles

The performance of large servers is to a high degree determined by their I/O subsystems. In the z990 server, nearly all of the components in the I/O path have been considerably improved in performance, capability, and cost. A 2-GB/s enhanced self-timed interface (eSTI) was introduced which is capable of absorbing the ever-increasing data rates of modern high-speed adapters. The I/O bandwidth available from a single node (three memory bus adapter, or MBA, chips, each with four eSTI ports) now equals 48 GB/s. As a consequence, both the MBA chip and the STI multiplexer switch (STI switch) chip had to be completely redesigned. In addition to these two chips, this paper describes the eSTI design itself and the Sweep chip, which integrates the function of four bidirectional adapter chips, one switch chip, and a clock chip.

Introduction

For a long time computer performance and processor performance have been synonymous for many people. This is reflected in the treatment of I/O in modern textbooks about computer architecture. For example, in [1], which is a standard reference in computer architecture, about 100 of the 1100 pages are devoted to I/O. However, in recent years the data regarding the crucial importance and value of the I/O subsystem design has become evident even to small organizations with small computer installations. There are two major reasons for this realization: One is the Internet, through which access to data and services is available around the clock. The other is that organizations are now seamlessly integrating their information technology (IT) infrastructure to match their business process flow. This realization, however, has been common knowledge for decades within zSeries* server-based organizations. For example, in [2] it is stated that for large mainframe computers the total cost of the I/O infrastructure typically exceeds the total cost of the processor complex by a factor of 10. A clear indication of this general mind shift is the emergence of new standards such as Infiniband (IB), which defines a new I/O and cluster interconnect and which uses concepts that have been available in zSeries I/O for decades [3, 4]. For transactions and database applications, performance characteristics are dominated by the quality of the I/O subsystem.

The familiar industry trend of exponentially increasing processing capacity in computing systems results in an exponential increase in I/O subsystem load. This rapid increase in the amount of data passing through system I/O has made fast access to that data a crucial challenge for the zSeries I/O subsystem. To remain a leader in that area, the I/O infrastructure of the z990 had to be completely redesigned, both for performance and for cost. In this paper three chips are described. The MBA chip and the STI switch chip make up the first-level network. The sweep chip is a low-cost adapter to zSeries serial channels. The eSTI interface used by the MBA chip and the STI switch chip is described as well.

I/O topology

Figure 1 shows the structure of the z990 central electronic complex (CEC) and its interconnection to external nets

0018-8646/04/$5.00 © 2004 IBM

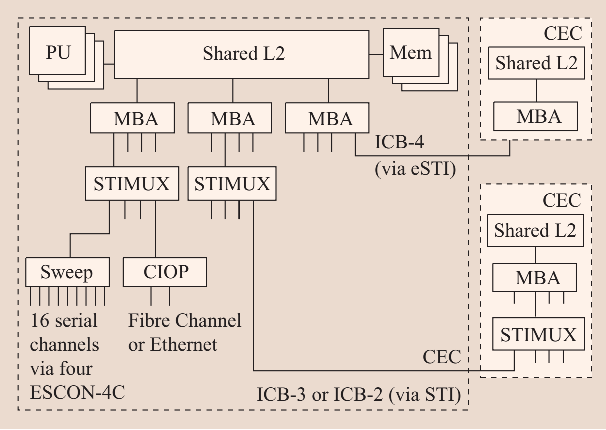

Figure 1

I/O overview.

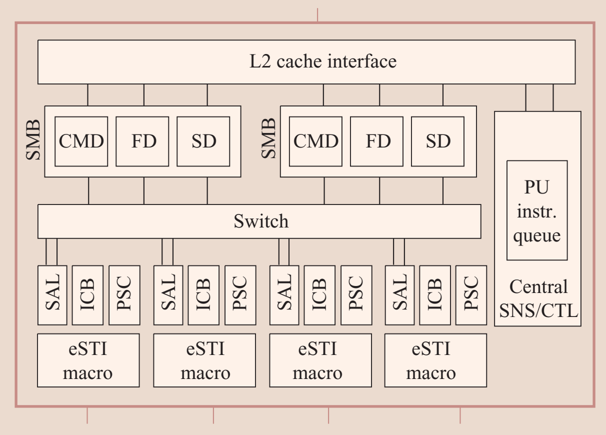

Figure 2

MBA block diagram.

such as communication networks (wide-area networks, local-area networks) and storage area networks. On a single node, a shared Level 2 (L2) cache connects up to 16 processors and three memory bus adapter (MBA) chips for the connection to I/O adapters, as well as several main memory boards. The MBA connects to the STI switch chip through four new high-speed eSTI buses, each of which is capable of 2 GB/s in each direction. The four links of the STI switch chip connect to three different types of chips: 1) The Sweep chip with additional logic on the ESCON* (Enterprise Systems Connection) card provides connections for serial channels. 2) CIOP (Common I/O Platform) books provide connection to Fibre Channel (FICON*) links or Ethernet links. The ISC (Inter-System Channel) chip provides support for coupling systems together via light fibers. One of the main objectives of zSeries systems is the clustering of CECs using the Parallel Sysplex* technology [5, 6]. For the highest bandwidth and the lowest latency, ICB-4 (Internal Channel Bus V4) links interconnect z990 systems. For comparison, the z900 is connected through a 1-GB/s multispeed self-timed interface (mSTI) coming out of a link downstream from the STI switch. Geographically dispersed sysplexes are built using the ISC optical interconnect.

MBA chip

Figure 2 shows the block diagram of the MBA chip. There are four ports. Each port consists of

1. An eSTI macro.
2. The ICB logic, which implements the short-distance low-latency sysplex clustering protocol.
3. The port sense/control (PSC) logic, which directs PU (processing unit) instructions to their target I/O registers. These PU instructions perform the reading and writing of registers existing in the ICB logic or in logic attached via the eSTI links.
4. An STI adaptation layer (SAL), which connects all port components to the switch. The switch connects the four port logic elements to two independently operating speed-matching buffers (SMBs), each consisting of an 8-deep command (CMD) queue and data buffers for fetch and store data (FD, SD). The switch in turn is connected to the L2 cache interface logic.

Finally, there is the central sense/control (SNS/CTL) logic, which contains a 64-deep instruction queue for PU sense/control instructions (instructions to read/write registers in I/O units).

The cycle time of operation of the chip has been improved in comparison with the z900 server generation [7], and all buses within the chip have been doubled in width to provide increased bandwidth and reduced latency. Up to four data transfers can be concurrently active in the switch—two fetch data transfers from the fetch buffers to the SALs and two store data transfers from the SALs to the SMB. The number of outstanding operations had to be doubled compared with the previous-generation machine. This improvement was implemented to counter the degradation that would otherwise have resulted from memory access times not scaling with processor or I/O performance improvements. Every port can have up to eight operations outstanding.

Significant enhancements have been made in the ICB logic. The number of buffer sets has been doubled. In addition to the native ICB-4 protocol (clustering between z990 machines), two variations of the sysplex protocol to previous-generation machines are supported. In the central SNS/CTL logic, the queue for SNS/CTL instructions had to be increased from support for 20 PUs to support up to 64 PUs. In previous machines there had been an individual signal for each PU which allowed the chip to signal interrupt conditions to the PU. In a node-based system with up to 64 PUs, it proved inefficient to support so many signals; therefore, a packet-switched interrupt signaling approach was introduced. Because of pin constraints at the L2 cache, the bus width has been halved; however, the frequency has been doubled. The 8-byte full-duplex bus runs at a cycle time of 1.6 ns, resulting in a bandwidth of approximately 8 GB/s. Data is transferred using a double data clocking scheme; at the beginning a calibration phase is executed which compensates for the skew between data signals caused by different wire lengths.

A huge improvement was achieved with the eSTI. Compared to an already high-speed 2-GB/s bus in the earlier system, the eSTI in z990 supports 4 GB/s of bidirectional bandwidth.

The memory bus adapter for z990 has implemented a new method for reading from data buffers for the case in which the read port is faster than the write port. The MBA performs the task of adopting data transfers with write-bandwidth A (2-GB/s eSTI logical macro port width w_1 and frequency $f_1 = A/w_1$) to a different read-bandwidth B (5-GB/s system communications chip bus with width w_2 and frequency $f_2 = B/w_2$). For this purpose, a speed-matching buffer (array) is used, as shown in **Figure 3**.

Two approaches are possible: 1) Read starts after full write is done (disadvantage: high latency), or 2) read starts while write is ongoing. The second approach is chosen. The typical problem area is that a buffer underrun can occur, since the application may start the read process too early. For example, the last read cycle could be before the last write cycle. The task is to optimize latency by choosing the minimum delay t_{wait} from first write to first read cycle, thus avoiding buffer underrun. Classical approaches use firmware-programmable registers, where the required delay is calculated by firmware based on frequencies, bus width, number of gaps during transfer (e.g., link control words) and length of the transfer unit.

The MBA is operated in various configurations in various environments: Protocol modes may have different numbers of gap cycles, different systems use different frequencies and line lengths, test procedures even do dynamic frequency changes. In the previous system, the settings had to be programmed to the worst-case scenario, which often resulted in a suboptimized latency in the functional mode of operation.

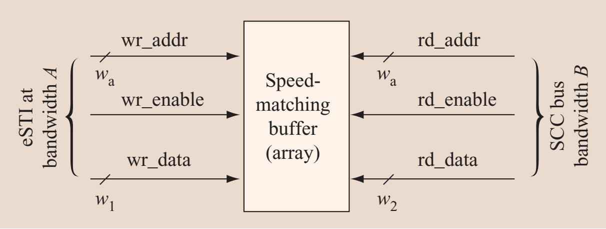

Figure 3

Speed-matching buffer. (w: width; w_a: address width; wr: write; rd: read.)

The newly introduced approach is to manipulate the delay t_{wait} via feedback control in such a way that it is decreased (to minimize latency) or increased (to prevent buffer underrun). The delay t_{wait} is implemented as a number of delay cycles n_{del}, which is decreased as long as it is not in the danger zone, where the next decrease of the delay value n_{del} would cause buffer underrun. The implementation uses a programmable underrun delay detection scheme; this automatically increases the delay when it detects that underrun has entered the programmable tolerance. With this implementation the z990 MBA is able to optimize the DMA (direct memory access) latency for a wide range of environmental conditions.

As described in the previous section, the MBA operates subunits at different frequencies. The system specification defines nominal frequencies for the units, but under some circumstances it is essential to operate the MBA in a wide, not always nominal, frequency range. Examples include a) prototype testing with high-speed links that do not allow the nominal speed in early chip deliveries; b) high-performance applications with selected fast parts; and c) new application environments (e.g., reuse in predecessor systems). The unpredictability of such conditions requires particular accuracy in the verification phase on interfaces between the multiple clock domains. Classical approaches typically focus on one clock domain crossing interfaces at three corners (max, nominal, min). To verify that the circuits allow accurate dataflow under various non-optimal conditions, a new method of verification of multiclock domains using a cycle simulator was introduced.

The assumption was made that the model could be operated for each frequency up to 30% faster than nominal or up to 2.5 times slower than nominal (which is not unrealistic for early hardware deliveries). The simulation environment was coded such that during initialization of the simulation model, all clock domains were programmed with a randomly selected frequency inside the assumed off-specification range. This approach verifies proper behavior over all clock domains.

STI switch chip

The STI switch chip is used to increase the number of available self-timed interface (STI) links in z990 systems

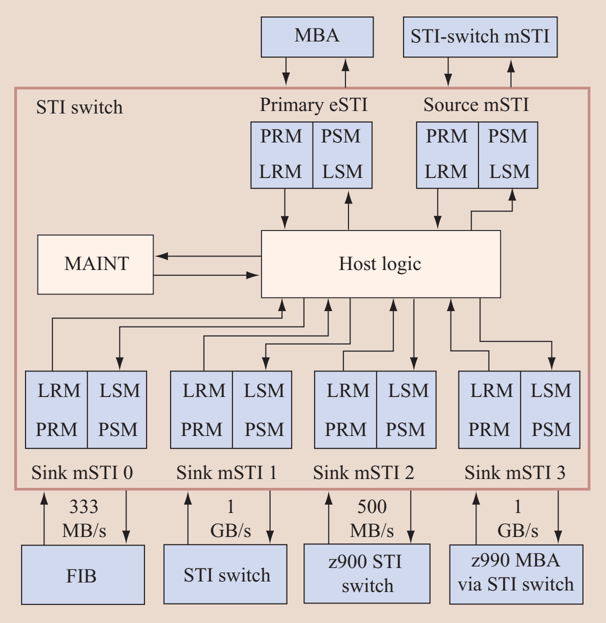

Figure 4

High-level dataflow of the STI switch chip. (LRM, LSM: logical receive and send macros; PRM, PSM: physical receive and send macros.)

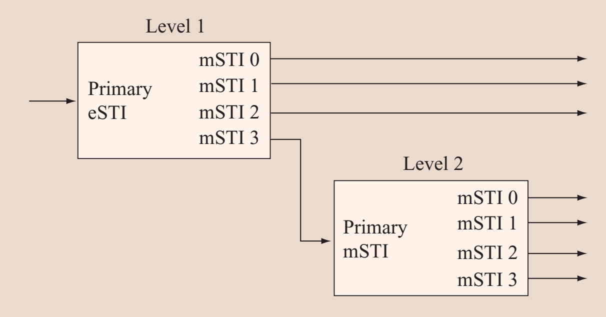

Figure 5

STI switch application.

and to connect I/O devices to the new 2-GB/s eSTI link. This is accomplished by fanning one STI link out to four STI links. **Figure 4** shows a high-level diagram of the STI switch chip.

There are two source links on the STI switch chip—an eSTI port and an mSTI (multispeed STI) port. The eSTI allows connections to 2-GB/s links. The mSTI allows connections at 333 or 500 MB/s or 1 GB/s. The FIB (fast internal bus buffer) chip in Figure 4 is further described in the Sweep section of this paper.

Figure 5 shows the applications in which the mSTI and the eSTI are used as the source port. The configuration that uses the mSTI port as the source port is known as the “cascaded configuration” and allows for a much higher fan-out of the STI links. Currently a maximum depth of two chips is supported. The source mSTI port is used only on a second-level switch chip. Use of the eSTI port is reserved for the first-level STI switch.

The host logic in the switch is connected to the source port, either eSTI or mSTI, the four sink ports, all mSTIs, and the maintenance logic. The STIs perform the link protocol elements required to de-skew, to de-serialize, and to reliably synchronize the incoming data. Data is then presented to the host logic as 128-bit (quadword) groups. The host logic is responsible for the following link and transport functions: packet and global LCW (link control word) routing, register reads and writes, and error reporting.

Figure 6 is a high-level diagram of the host portion of the STI switch chip. It shows two source ports on the left: an eSTI port and an mSTI port. Only one of these two ports can be configured for use: eSTI on a Level 1 STI switch and mSTI on a Level 2 STI switch. Dataflow from a source link is accomplished in the following manner. The incoming data is synchronized by the source port logical receive macro (LRM) and presented to its north controller with advanced control information indicating the number of words and the destination ID. The north controller trains on the incoming dataflow until it finds a valid packet, either when the header is valid or when the whole packet is valid. The routing logic can then determine the destination of the packet: either one of the mSTI LSMs (logical send macros) or the sense control engine. It then stores the LRM buffer into a packet order queue to preserve packet order on a per-destination basis and to determine whether the corresponding pipeline to the destination is full. Assuming that the pipeline is not full, the host grants one of the LRM buffers access using a round-robin scheme based on the destination of the packets, and the data is transferred in quadwords from the LRM buffer through the arbiter to the destination.

The sense control engine is the hardware facility that executes packets intended for reading or writing registers within the STI switch chip itself.

Dataflow from a sink port is accomplished in the following manner. The incoming data is presented by the mSTI LRM to its corresponding south controller with valid bits indicating a valid packet, either when the header is valid or when the whole packet is valid. In the case of a sink port, there is only one destination, the source port. The LRM buffer is then stored in a packet order queue, and the corresponding pipeline is checked as to whether

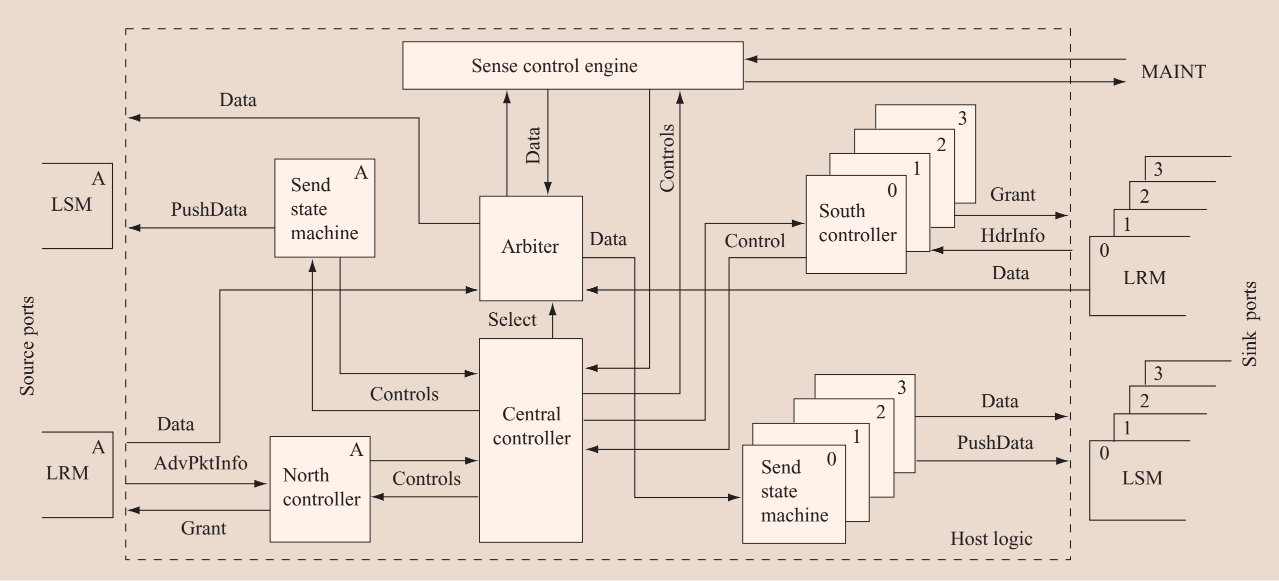

Figure 6

STI switch host dataflow. (HdrInfo: header information.)

or not it is busy. Note that the source port receives packets from the four sink ports and from the sense control engine. Assuming that the pipeline is not full, the host grants the LRM buffer access using a round-robin scheme based on the source of the packet, and the data is transferred to its destination through the arbiter cross-point switch.

There are two types of STI ports: The mSTI port is a multispeed port and is compatible with all legacy STI ports; the eSTI port is a new type of STI interface. This port runs at higher speeds; the protocol and the logical macro have been changed to accommodate this. There have also been enhancements made to the mSTI ports for error handling, packet ordering, and retrying packets when errors occur. The logical send and logical receive macros are the components that implement the dataflow and protocol of the STI ports.

Figure 7 shows the basic data path of the logical STI macro. The arrows going to and from the host represent 128-bit data buses with 16 bits of parity. The arrows to and from the link represent 96-bit data buses with 12-bit K-buses that help define the 8/10 code. The 8/10 code disclosed in [8] is an encoding scheme that converts any data stream of 8-bit words into a data stream of 10-bit words that has regular switching characteristics. These regular switching characteristics ensure that the 10-bit data stream signal maintains signal strength as it passes through the capacitors that are serial to the link path.

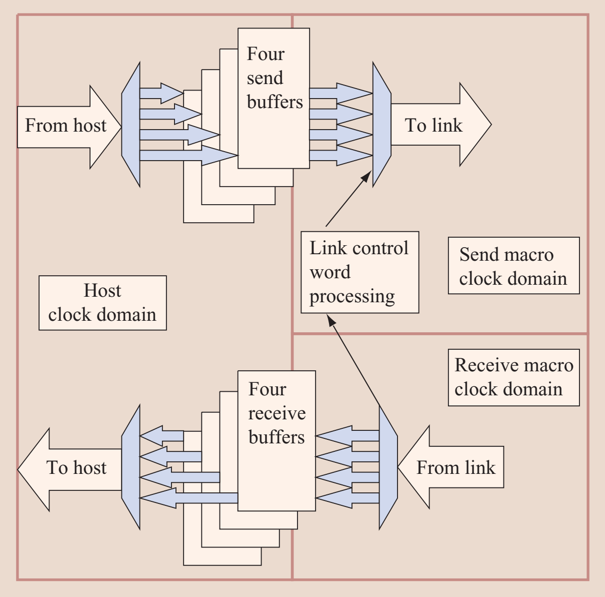

Figure 7

Logical STI macro data path.

The arrows between the send and receive macros are control words to handle handshaking occurring in the link.

There are three clock domains in each logical macro. The host clock domain runs on the host clock and is synchronous with it; the send clock domain is synchronous

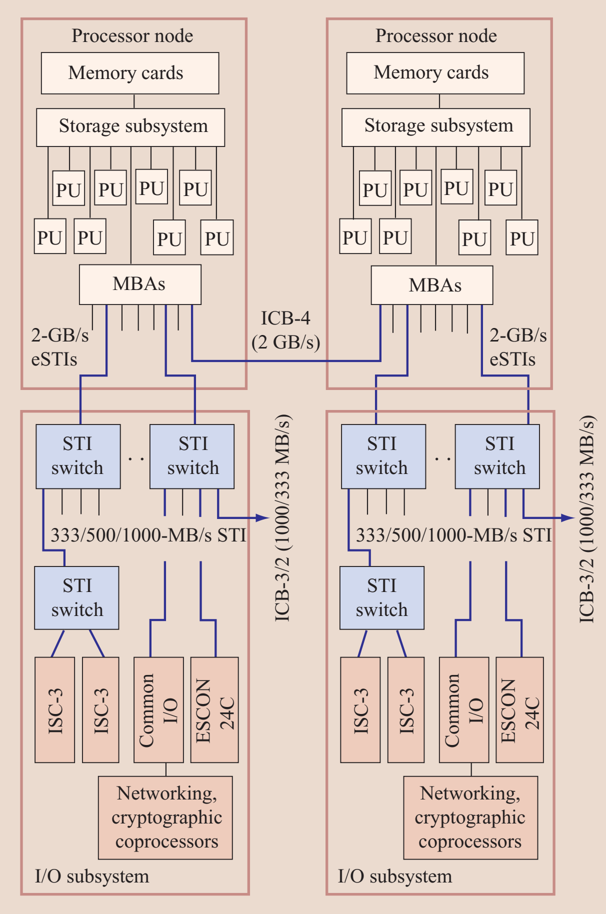

Figure 8

eServer z990 system diagram showing STI usage.

with the physical send macro; and the receive clock domain is synchronous with the physical receive macro. All signals on the chip that cross asynchronous boundaries between the domains are handled within the logical macros through a series of metastability latches and signal stretching. Data packets are stored in the buffers of the logical macro under one clock domain. Status signals then cross the asynchronous boundary to indicate to the next clock domain that the data in the buffer is valid and can be read out at any time. Link control words are received from the link and sent across the boundary to the send macro clock domain, where they are processed.

The eSTI macros shown in Figure 7 each contain four send and four receive buffers to allow for greater bandwidth. The mSTI ports contain only two send buffers. They have two data buffers on the receive side that float between four header buffers. This allows the receive macro to hold four packets at a time, but only two of them can have data payloads. This design was unchanged from previous implementations to maintain compatibility. In either case, the send macro tracks which of its own buffers are available, as well as which of the attached receive buffers are available across the link.

In normal operation, when no errors occur, the STI macros use a simple handshaking technique to ensure that packets are being transferred properly. A packet is loaded into a send buffer by the chip host. That packet is then sent across the link to a receive buffer chosen by the sender. For mSTI, the packet is sent with parity and longitudinal redundancy checking (LRC) on both the header and data. The eSTI macro sends the packet using an 8/10 code and cyclical redundancy checking (CRC) on both the header and data. When the receive macro detects a header arriving on the link, it checks that the parity and LRC word are correct for mSTI, and that the 8/10 code and CRC word are correct for eSTI. When the header is completely received and determined to be valid, an acknowledgment (ACK) is sent back across the link. Once this acknowledgment is sent, the receive macro has accepted the packet and will pass it on even if there are errors later in the transfer. If an error does occur on part of the packet following the header, the packet will be passed on with the error and with an "abort" attached to the packet to allow the requester to retry the operation. When the send macro across the link receives this ACK, it will mark its send buffer as free and be ready to accept another packet. After the host of the chip across the link takes the packet out of the receive buffer, the receive macro sends another message that tells the send buffer that the receive buffer across the link is now empty and another packet can be sent to it.

If an error occurs on the link during the transfer of a packet header, that header is assumed to be not reliable, and the entire packet is ignored by the receive macro. No ACK is sent. After a counter timeout, the send macro requests this ACK. Since the receive macro dropped that packet, it does not send a matching ACK, and the send macro recognizes that the packet was lost. It will re-send the packet up to seven times before giving up and declaring the link bad.

There is also a mode called *data packet retry mode.* In this mode, the ACK for a packet is not sent after the header is checked for validity; it is sent after the entire packet has been checked and it has been determined that there were no errors during the transfer. If an error occurs at any time during the packet transfer, the packet is dropped, and the recovery described above will happen (i.e., the send macro will request the ACK, not get a match, and resend the packet). This mode slows down the link a bit because the send buffers cannot be filled as quickly. However, if there are errors on the link, this mode allows retry of packet transmission to occur by link protocol logic instead of having the error reported back to the originator of the packet.

Enhanced self-timed interface (eSTI)

The self-timed interface (STI) continues to be the main interconnect used throughout the eServer* z990 I/O subsystem. Building on the eServer z900, in which the STI unidirectional bandwidth topped out at 1 GB/s [9], a z990 STI port now supports this speed in addition to the standard 333-MB/s and 500-MB/s speeds [10]. These STI ports have been named multispeed STIs, or mSTIs.

In the eServer z990, mSTI is used on the backplane of the I/O subsystem in addition to providing sysplex connectivity to legacy zSeries servers [11, 12]. The mSTI links are also known as the secondary STI links.

To keep pace with the increased processing power of the eServer z990, a corresponding increase in the STI bandwidth was required. To satisfy this requirement, the eSTI was developed. The unidirectional bandwidth of the eSTI is 2 GB/s, twice that of the mSTI. This is a bidirectional bandwidth of 4 GB/s.

Figure 8 shows the uses for the STI throughout the eServer z990. Each processor node can contain up to twelve primary 2-GB/s eSTI ports, with up to 48 more secondary 333/500/1000-MB/s mSTI ports. A primary eSTI link can also carry sysplex traffic between nodes within a z990 or between processor nodes on different eServer z990s.

The eSTI is built upon the same principles as the mSTI in that data is transmitted using both edges of a clock that is also sent along with the data to the receiving end of the link. To reliably capture the data at the receiving end of the link, the data is sent through a delay chain where edge-detecting circuitry locates both edges of the data, which is then sampled at the midpoint between the two data edges. Ultimately each data bit is realigned with the transmitted clock at the receive end of the interface and also captured using both edges of the clock. With this method, skew between the conductors of the interface can be compensated for and removed. References [9] and [10] provide a more detailed discussion on the principles of STI operation.

Table 1 shows the major differences between the mSTI and the eSTI. Besides the previously mentioned differences in the bandwidth, one of the most notable differences is the use by the eSTI of the well-known 8B/10B encoding of the transmitted data. Among other reasons, this was done in order to accommodate the potential use of optical fibers in place of copper cables as the interconnection medium.

Since the transmitted data is now encoded using the 8B/10B code, the detection of code violations (CVs), or invalid code points, can be exploited to determine whether an error occurred in the transmission medium. In effect, the detection of CVs now becomes the first means of detecting errors in the link data. This is in contrast to the mSTI links, which employed parity error detection for this same purpose.

Table 1 Major differences between MSTI and eSTI.

	mSTI	*eSTI*
Bandwidth	333/500/1000 MB/s	2000 MB/s
Bit time	3/2/1 ns	600 ps
Cycle time	6/4/2 ns	1200 ps
Host interface	4 bytes + 4 parity	12 bytes
Link bus width	8 data, 1 parity + clock	12 data + clock
Differential signals	10 pairs (20 wires)	13 pairs (26 wires)
Data encoding	none	8B/10B code
Data striping	1 bit per byte per wire (all bit 0 - wire 0, etc.)	1 byte per wire (byte 0 - wire 0, etc.)
Serialization	4:1	10:1
Bit de-skew	3 bit times	10 bit times
Transmission coupling	dc	ac
Usage	Backplane max. 10 m cable	max. 10 m cable

Another difference between eSTI and mSTI is the number of conductors that are used; mSTI sends eight data bits and one parity bit during each bit interval, whereas eSTI transmits 12 data bits during each bit interval. For mSTI the bit rate (in bits per second) and the data rate (in bytes per second) are the same, but this is not the case for eSTI. Table 1 shows that the eSTI bit time is 600 ps, which yields a data rate of 1.67 Gb/s. Since 12 data bits are transmitted each bit interval and an 8B/10B encoded byte is represented by 10 bits, the data rate is 1.2 × 1.67 = 2.0 GB/s.

Sweep chip

The IBM eServer z990 continues to support the ESCON channels using a serial optical interface, as in the z900 processor [7]. The operating speed is 17 MB/s over a distance of three kilometers. The I/O card providing this interface is the ESCON-16 card, which drives 16 ESCON channel ports. Fifteen channel ports can be active, while the sixteenth is reserved as a spare port. The ESCON-16 card is plugged into the IBM eServer z990 I/O cage.

The new component on the ESCON-16 card is the Sweep chip implemented in CMOS (complementary metal oxide semiconductor) technology. The Sweep chip merges the functions which in the precursor card had been implemented with three chips: the fast internal bus buffer (FIB), the BiDi (bidirectional) bus distributor (BBD), and

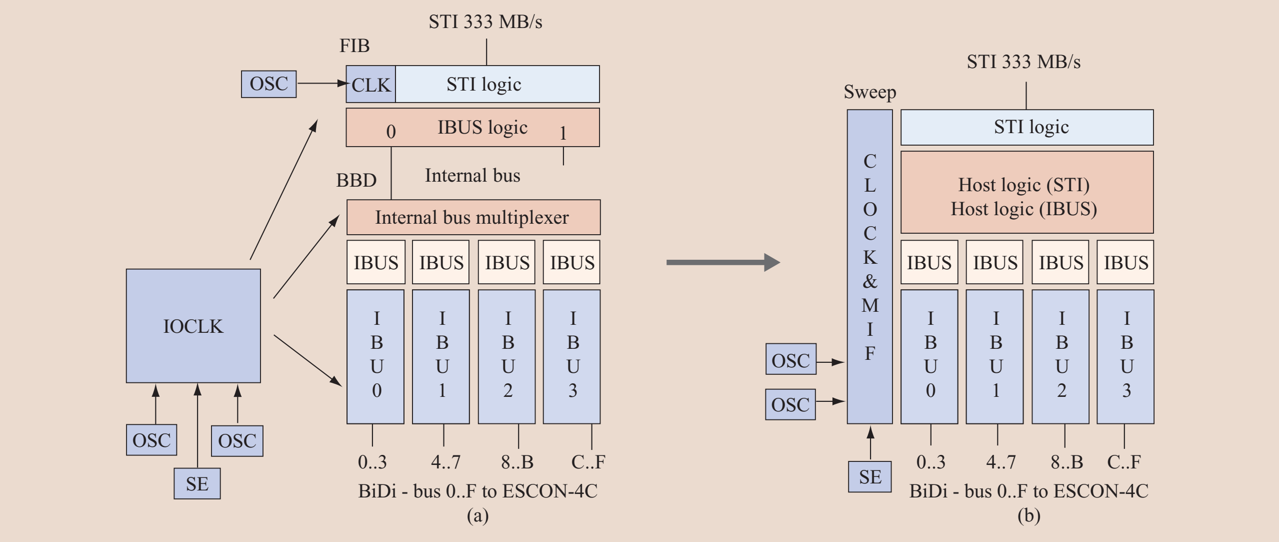

Figure 9

Sweep chip diagram: (a) pre-z990; (b) z990.

the I/O clock chip (IOCLK) [7]. The progress of CMOS technology has allowed us to combine these three chips into a single one. **Figure 9** provides an overview of the logical elements merged to create the Sweep chip. It also illustrates the connection to the service element (SE) which controls the initialization and clock state of many chips within the z990 system.

Among the other major components on the ESCON card are the four ESCON-4C chips [7], each of which controls four ESCON channel ports (via electrical-to-optical converters) and connects via four BiDi buses to the BBD (Sweep).

The remainder of this description is about Sweep. The main data path through Sweep is between the STI bus [9] and the sixteen BiDi buses connecting to four ESCON-4C chips. On the precursor card, the interface with the STI bus is controlled by FIB and the interface with the BiDi buses by the BBD. Between FIB and BBD is the "internal bus," which allows for eight participants, called internal bus units (IBUs). In BBD the logic to support four BiDi buses is defined as such an IBU. Since the logic content of the BBD chip has been reused in Sweep with minor modifications, Sweep contains four IBUs as well.

Multiple simultaneous operations

Improvements in Sweep are related to the independent operation of these four IBUs. Each IBU can issue four types of operations: direct memory access (DMA) Fetch, DMA Store, Register Read, and Register Write. In FIB-BBD, basically only one of the four IBUs at a time could have an operation in progress. Under certain restrictions, one DMA operation allowed another DMA operation from a different IBU to be handled simultaneously: This was an internal bus characteristic. In Sweep such a limitation no longer exists. Each IBU can start an operation of arbitrary type, so that four operations can be outstanding simultaneously without having interdependencies.

Another area of improvement is in the scheduling of the four IBUs. Whenever an IBU wishes to start an operation, it raises a request to the arbiter. The arbiter grants (the request from) an IBU based on whether access to the STI bus is available. Each IBU can use three levels of priority for its requests. The arbiter grants the IBUs on the basis of a round-robin pointer for each priority level to guarantee fairness. In order to achieve maximum throughput with minimum latency, careful consideration has been given to the scheduling of consecutive operations. The goal is to start a new operation before the previous one has fully completed, so that the critical interfaces can be kept busy without losing cycles. In this optimization process it has been taken into account that logic functions involved in the arbitration process are operated with different and variable clock domains.

On the precursor ESCON-16 card, the IOCLK chip provides the clock signals and maintenance interface (MIF) functions for the dataflow chips FIB and BBD. The MIF functions include generation of clocks, control of shift modes, and execution of chip self-test operations. These functions are monitored by a service element (SE). Sweep merges all of these functions together with the dataflow logic on the same chip. Sweep has six clock domains and six independent self-test areas. There is a separate self-test for each IBU. Other than in BBD, each IBU is an independent entity. In case the self-test for one IBU fails, it can be taken out of operation and the remaining ones can continue to work properly. Of course, if one of the central layers in Sweep (e.g., the host logic) fails in self-test, the total chip must be taken out of operation.

Summary

The chips, links, and design features that comprise the z990 I/O subsystem combine to provide an extremely rich fabric of I/O and coupling functionality. The improvements include cost benefits, such as the integration into the Sweep chip of elements that had been spread across numerous chips. Improvements to performance and functionality have been made in all of the chips, continuing a long-term trend of such improvements in zSeries systems. These improvements include the higher (2-GB) bandwidth of the eSTI links and additional buffering built into the STIs and the data paths. Furthermore, additional resources, such as wider buses, have been added to improve traffic flow. The implementation of newly integrated faster, denser chips has been detailed. In addition to structural and logical changes, the integration of elements that existed in previous vintage, slower, and less dense chips has been described. Finally, design improvements for error recovery have been made. The high-bandwidth, high-reliability, and low-latency capabilities of this subsystem have resulted from years of similar broad-ranging improvements built on a foundation of system architecture.

*Trademark or registered trademark of International Business Machines Corporation.

References

1. J. Hennessy, and D. Patterson, *Computer Architecture: A Quantitative Approach*, Third Edition, Morgan-Kaufmann Publishers, San Francisco, 2002.
2. G. F. Pfister, *In Search of Clusters*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1998; ISBN: 0138997098; pp. 469–470.
3. T. Shanley, *Infiniband*, Addison-Wesley Publishing Co., Reading, MA, 2002.
4. IBM Corporation, *z/Architecture Principles of Operation* (SA22-7832); see *http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi/*.
5. *IBM Syst. J.,* special Sysplex issue, Vol. 36, No. 2 (1997).
6. *IBM J. Res. & Dev.,* special Sysplex issue, Vol. 36, No. 4 (July 1992).
7. D. J. Stigliani, Jr., T. E. Bubb, D. F. Casper, J. H. Chin, S. G. Glassen, J. M. Hoke, V. A. Minassian, J. H. Quick, and C. H. Whitehead, "IBM eServer z900 I/O Subsystem," *IBM J. Res. & Dev.* **46,** No. 4/5, 421–445 (July/September 2002).
8. P. Franaszek and A. Widmer, "Byte Oriented DC Balanced (0,4) 8B/10B Partitioned Block Transmission Code," U.S. Patent 4,486,739, December 4, 1984.
9. J. M. Hoke, P. W. Bond, R. R. Livolsi, T. C. Lo, F. S. Pidala, and G. Steinbrueck, "Self-Timed Interface of the Input/Output Subsystem of the IBM eServer z900," *IBM J. Res. & Dev.* **46,** No. 4/5, 447–460 (July/September 2002).
10. J. M. Hoke, P. W. Bond, T. Lo, F. S. Pidala, and G. Steinbrueck, "Self-Timed Interface for S/390 I/O Subsystem Interconnection," *IBM J. Res. & Dev.* **43,** No. 5/6, 829–846 (September/November 1999).
11. T. A. Gregg, K. M. Pandey, and R. K. Errickson, "The Integrated Cluster Bus for the S/390 Parallel Sysplex," *IBM J. Res. & Dev.* **43,** No. 5/6, 795–806 (September/November 1999).
12. T. A. Gregg and R. K. Errickson, "Coupling I/O Channels for the IBM eServer z900: Reengineering Required," *IBM J. Res. & Dev.* **46,** No. 4/5, 461–474 (July/September 2002).

Received September 22, 2003; accepted for publication February 2, 2004; Internet publication May 25, 2004

Edward W. Chencinski *IBM Systems and Technology Group, 2455 South Road, Poughkeepsie, New York 12601 (chencins@us.ibm.com).* Mr. Chencinski received a B.S. degree in electrical engineering from Lehigh University in 1980, joining IBM that same year. He was involved in the ES/3090 SCE and expanded storage hardware design, as well as the logic support element design of the ES/9000. In the early 1990s Mr. Chencinski joined the G3/G4 CMOS cryptographic hardware processor design team, focusing on pervasive functions, simulation, timing, and modular exponentiation. He is currently a Senior Engineer leading the STI switch chip design team.

Michael J. Becht *IBM Systems and Technology Group, 2455 South Road, Poughkeepsie, New York 12601 (becht@us.ibm.com).* Mr. Becht is a Staff Engineer in the IBM eServer I/O Hardware Development group. He received his B.S. degree in electrical engineering from the University of Delaware in 1998. That same year he joined IBM in Poughkeepsie, New York, where he was involved in the development of the SP switch for pSeries supercomputers. Since then he has held various technical positions and is currently engaged in the development of next-generation I/O for zSeries supercomputers.

Tim E. Bubb *IBM Systems and Technology Group, 2455 South Road, Poughkeepsie, New York 12601 (bubb@us.ibm.com).* Mr. Bubb is an Advisory Engineer in the IBM eServer I/O Hardware Development group. He received his B.S. degree in electrical engineering from the Virginia Polytechnic Institute in 1988, and his M.S. degree from Purdue University in 1989. He joined IBM at Poughkeepsie, New York, in 1990 and has held various technical and management positions in the eServer I/O design area. Mr. Bubb has received an IBM Outstanding Innovation Award for his work on the Hydra I/O subsystem, and he has received two IBM Outstanding Technical Achievement Awards for his work on the Multiprise 3000 and IBM eServer z900 I/O subsystems.

Carolynn G. Burwick *IBM Systems and Technology Group, 2455 South Road, Poughkeepsie, New York 12601 (burwick@us.ibm.com).* Ms. Burwick is a Staff Engineer in the IBM eServer I/O Hardware Development group. She received her B.S. degree in computer engineering from the Rochester Institute of Technology in 1997. She joined IBM in Poughkeepsie, New York, the following year and has held various technical positions in the eServer I/O area. Ms. Burwick is currently engaged in the development of next-generation I/O for zSeries supercomputers.

Juergen Haess *IBM Systems and Technology Group, IBM Deutschland Entwicklung GmbH, Schoenaicherstrasse 220, 71032 Boeblingen, Germany (haess@de.ibm.com).* Mr. Haess received his M.S. degree in electrical engineering from the University of Karlsruhe, Germany, in 1980; since joining IBM, he has worked on I/O adapter development for many years and for several released products. In 1994 he moved to Poughkeepsie for half a year to transfer the design of the G3, STI-based FIB chip to Poughkeepsie and to coordinate its design and bringup with Boeblingen. During various jobs he has received an IBM division award, an IBM team award, and three IBM Invention Achievement Awards for his work. In 1997 he joined the CPU development team, where he became a member of the FPU team in 1999. To support the Sweep design, he joined the I/O team to adapt the FIB design. He currently works in floating-point design for next-generation IBM pSeries and zSeries processors. Mr. Haess is an author or coauthor of eight patents and thirteen publications, including a conference proceeding and two journal articles.

Markus M. Helms *IBM Systems and Technology Group, IBM Deutschland Entwicklung GmbH, Schoenaicherstrasse 220, 71032 Boeblingen, Germany (helms@de.ibm.com).* Mr. Helms studied electrical engineering at the Berufsakademie Stuttgart and received the Dipl.Ing. degree in 1993, joining the IBM laboratories in Boeblingen that same year as an R&D engineer. He worked in various technical positions (verification, logic design, architecture) in the zSeries I/O Adapter area. Most of his time was spent on development of the MBA. His current focus in the InfiniBand I/O Architecture and its common design implementation for the IBM eServer. Mr. Helms has received an IBM Outstanding Technical Achievement Award, an IBM Invention Achievement Award, and numerous informal awards.

Joseph M. Hoke *IBM Systems and Technology Group, 2455 South Road, Poughkeepsie, New York 12601 (jmhoke@us.ibm.com).* Mr. Hoke is an Advisory Engineer in the IBM eServer I/O Hardware Development group. He received the B.S. degree in electrical engineering from the University of Illinois at Chicago in 1987 and continued his studies under a university fellowship, receiving the M.S. degree in electrical engineering from Northwestern University in 1989. He joined IBM at Poughkeepsie, New York, in 1989 and has held various technical positions in the eServer I/O area. Mr. Hoke holds several patents used in the IBM ESCON and Sysplex products, and he has received two IBM Invention Achievement Awards. He has received an IBM Outstanding Technical Achievement Award for his work on ESCON and his work on the G5 Server, and another for his contributions to the eServer zSeries.

Thomas Schlipf *IBM Systems and Technology Group, IBM Deutschland Entwicklung GmbH, Schoenaicherstrasse 220, 71032 Boeblingen, Germany (schlipf@de.ibm.com).* Mr. Schlipf is a Senior Technical Staff Member in the IBM Systems and Technology Group. He received his M.S.E.E. degree from the University of Karlsruhe, Germany, in 1983. In 1985, after working for a time at the Robert Bosch Company, Germany, he joined the IBM Server Group development laboratories in Boeblingen. Since then he has worked on the hardware design of I/O chips. He has led three MBA projects and has received an IBM Outstanding Innovation Award, an IBM Outstanding Technical Achievement Award, and two IBM Invention Achievement Awards. Mr. Schlipf is a member of the IEEE.

Jeffrey M. Turner *IBM Systems and Technology Group, 2455 South Road, Poughkeepsie, New York 12601 (turner@us.ibm.com).* Mr. Turner received a B.Eng. degree in electrical engineering from Rensselaer Polytechnic Institute in 1978 and an M.Eng. degree in 1979. In 1979 he joined IBM in East Fishkill, New York. He has held various technical positions in embedded systems, memory controller, and I/O subsystem design, and has received IBM Outstanding Innovation and Outstanding Technical Achievement Awards for his work on the zSeries Open Systems Adapter. Mr. Turner is currently a Senior Technical Staff Member in the eServer I/O area.

Hartmut Ulland *IBM Systems and Technology Group, IBM Deutschland Entwicklung GmbH, Schoenaicherstrasse 220, 71032 Boeblingen, Germany (hulland@de.ibm.com).* Mr. Ulland received his M.S. degree (Dipl.Ing.) in electrical engineering from the Technical University (RWTH) Aachen in 1969 and joined the IBM Boeblingen laboratory, where he worked on various System/370-oriented projects in the Advanced Technology Department. From 1982 to 1985 he was on assignment to Boca Raton, Florida, where he led the design of the I/O connection for a RISC-based system. Since 1986 Mr. Ulland has served as a logic designer and/or team leader in S/390 I/O and PU-related projects. He has served as the team leader of the Sweep design since 2001, and has received two IBM Invention Achievement Awards.

Manfred H. Walz *IBM Systems and Technology Group, IBM Deutschland Entwicklung GmbH, Schoenaicherstrasse 220, 71032 Boeblingen, Germany (mhwalz@de.ibm.com).* Mr. Walz received the Dipl.Ing. degree in electrical engineering from the Berufsakademie Stuttgart in 1979. In 1979 he joined the IBM development laboratories in Boeblingen, working on memory for the 43XX systems. From 1985 to 1995 Mr. Walz led several memory development projects. In 1996 he joined the I/O subsystem development team. He has led the MBA development for two generations of the zSeries Systems. Mr. Walz has served as a lecturer at the Berufsakademie Stuttgart; he is a member of the IEEE.

Carl H. Whitehead *IBM Systems and Technology Group, 2455 South Road, Poughkeepsie, New York 12601 (whitehea@us.ibm.com).* Mr. Whitehead is a Senior Engineer in the eServer I/O Hardware Development group. He received a B.S. degree in electrical engineering from Manhattan College in 1979. He subsequently joined IBM at Poughkeepsie, New York, and has held various technical positions in the eServer processor and I/O areas. Mr. Whitehead has received three IBM Outstanding Technical Achievement Awards for his contributions to several generations of zSeries I/O subsystems.

Gerhard Zilles *IBM Systems and Technology Group, IBM Deutschland Entwicklung GmbH, Schoenaicherstrasse 220, 71032 Boeblingen, Germany (zilles@de.ibm.com).* Mr. Zilles studied information technique at the Fachhochschule Jülich. In 1978, after some practical experience at the nuclear research center in Jülich, he joined the IBM Laboratory in Boeblingen, working for ten years on storage controller and memory card hardware design. Since 1988 he has worked on various I/O chip designs as engineer/team leader, and has worked for a year on sysplex firmware development. Recently he has been involved in the EETR and Sweep projects.