High-speed source-synchronous interface for the IBM System z9 processor

D. M. Berger
J. Y. Chen
F. D. Ferraiolo
J. A. Magee
G. A. Van Huben

As mainframes evolve and deliver higher performance, technologists are focusing less on processor speed and more on overall system performance to create optimized systems. One important area of focus for performance improvement involves chip-to-chip interconnects, with their associated bandwidths and latencies. IBM and related computer manufacturers are optimizing the characteristics of interconnects between processors as well as between processors and their supporting chip sets (local cache, memory, I/O bridge). This paper describes the IBM proprietary high-speed interface known as Elastic Interface (EI), which is used for nearly all chip-to-chip communication in the IBM System z9™. In particular, EI is a generic high-speed, source-synchronous interface used to transfer addresses, controls, and data between CPUs, L2 caches, memory subsystems, switches, and I/O hubs. The EI has single-ended data lines, resulting in twice the performance (bandwidth per pin) of similar buses operating with two differential lines per signal.

Introduction

The Elastic Interface (EI) described in this paper is a high-speed interface for the IBM System z9*; the EI is used for nearly all chip-to-chip communication in this computing platform. We refer to the Elastic Interface as EI-2 to indicate that it is a second generation of such an interface. Other microprocessor companies, such as AMD (Advanced Micro Devices), also have research efforts directed toward high-bandwidth, low-latency interconnects such as that provided by the HyperTransport** bus. Readers are directed to [1] for more information on such technologies and on the HyperTransport Technology Consortium.

The prior related IBM platform, the IBM System z990, employed the first-generation EI primarily for off-MCM (multichip module) communications on approximately 2,500 nets (chip-to-chip wires) [2, 3]. The role of the EI-2 design for the System z9 was expanded to include both on-module and off-module communications to improve performance and bandwidth, which increased the system usage to approximately 8,500 nets. For the IBM System z9, we added unique functions and features to EI-2 in order to improve the performance, power saving, diagnostics, robustness, and MCM manufacturing yield. These functions and features include tuning circuits that provide latency reduction, new firmware, and a new serial interface control for EI-2 calibrations and diagnostics.

The Elastic Interface is a modular design that may have different bus widths, speeds, internal chip clocking, and other features tailored specifically for an application. Bus widths from one to eight bytes are supported, depending on the speed of the interface (for example, high-frequency buses limit the bus width). The data signals are single-ended and optionally source-terminated to use less power, or far-end-terminated to achieve higher data rates. The double-data-rate (DDR) bus clock is differentially driven in order to maintain the clock duty cycle and signal fidelity. The EI is scalable in speed from below 250 MB per second to ~2.0 GB per second per signal pin to facilitate bring-up (i.e., the evaluation of early hardware that operates below product speeds) and a variety of applications. A power-on or initialization sequence is sent across the interface as part of the initialization alignment procedure (IAP). The IAP individually de-skews (phase-aligns) all of the bits on the bus to the latest arriving bit and optimally places the sampling clock in the center of

0018-8646/07/$5.00 © 2007 IBM

the combined data-valid region. The EI also has a FIFO (first-in, first-out) buffer on the receive side that aids in system timing and synchronization of the two ends of the interface. Most chip-to-chip transfers within the System z9 are synchronous in order to minimize latency and improve overall system performance. The latency, or time to transfer information between chips, is measured in bit times. This latency is a programmable value in order to provide a generic solution that can be tailored for multiple system configurations and provide design flexibility. The cycle or clock edge in which data is captured on the receive chip in the local clock domain is denoted as the "target time."

The EI-2 provides unidirectional data transfer that replaces the System z990 on-module bidirectional nets [3]. In order to maintain the per-net bandwidth, the speed of the EI-2 on-module interfaces had to at least double. In fact, the System z9 on-module interfaces are three times faster than those in the System z990 design and eliminated the turnaround time associated with bidirectional nets [4]. Improved speed and removal of the bidirectional bus turnaround times for these interfaces were prerequisites for achieving the overall System z9 performance goals. In addition to de-skew and other characteristics of the Elastic Interface that help address packaging concerns, the built-in diagnostics played an important role in bring-up and design verification.

EI diagnostics were integrated in the System z9 serial interface facility (SIF). The SIF is an interactive debug and diagnostic interface that allows the real-time evaluation of bus performance while the machine runs functional workloads. Real-time evaluation of the quality of the signaling (with respect to timing margins and noise margins) and evaluation of built-in random data test (RDT) facilitated an early bring-up and assessment of the entire chip-to-chip interconnect, which involves silicon, modules, cards, board, and connectors, at or above operational speed. This is significant because full-speed testing of interconnects at the system level does not traditionally take place until late in the development cycle, a time at which late-breaking design issues can be catastrophic.

The EI-2 design facilitates electronic repair (e-repair), and this is a new feature for the Elastic Interface. On-chip switches and control bits were built into the EI-2 electronics to allow the signal wires to be statically rerouted using a spare bit on the chip and module. These spare bits were easily routed along with the original EI-2 signal wires to ensure high signal fidelity. When a manufacturing defect, such as a short or open circuit, was found in the module, the electronics could be used to circumvent the defect. The e-repair improved the MCM manufacturing yield from approximately 55% (corresponding to zero defects) to 70% (using spare nets).

The major improvements of the EI-2 over the previous-generation EI-1 in the System z990 are the following:

1. Data drivers capable of pre-distortion and data receivers with selectable modes for post-hardware power and performance tuning [4].
2. Improved data retiming capabilities, including *in situ* monitoring of the incoming signal for optimal setup and hold margins.
3. Dynamic repair capability with post-power-on data rerouting for yield recovery.
4. More robust diagnostic capabilities for hardware verification.
5. Reference-voltage forwarding on communication paths between multichip modules (MCMs) to overcome process variations between driving and receiving chips [4].
6. On-MCM nets, which run two times faster than off-MCM nets that leverage packaging differences.

As additional background, the EI is a generic, high-performance bus between CPU, memory, and I/O devices. The EI is a source-synchronous design which uses a differential DDR bus clock and single-ended signaling that employs one wire per signal or data bit, as indicated by "data<1>" to "data<xx>" in **Figure 1(a)**. The delay (dly) blocks in the datapath depict the per-signal data de-skew function. Each bit on the interface is phase-aligned, during power-on, to the latest arriving bit. The latest arriving bit or bits have zero added delay in order to minimize the bus latency. The dly block in the clock path is used to center the sampling clock in the data-valid window across all bits. The bs block in Figure 1(a) is the boundary-scan multiplexer (MUX), used for testing. The four-wide combination of MUX and latch structures depicts the receive-side FIFO before the final destination latch in the local-clock domain. The FIFO is used to facilitate the synchronization (i.e., transfer of data) between the incoming off-chip signals and the latches operating on the local chip clock domain.

A detail of the receive side FIFO is depicted in **Figure 1(b)**. The latches in the FIFO are data-gated to simplify clocking. Each latch is sequentially written and holds its current value as the other latches are written. The approach expands the valid time so that the latch may be more easily read by the local clock domain. A 4:1 MUX is used to sequentially read the outputs of the four FIFO latches, one at a time, into the destination latch that operates in the local clock domain. The FIFO is initialized during the IAP process so that the system writes the first bit in the sequence into the first latch in the FIFO. The cycle in which the first latch is read into the destination latch is set by the programmed target time,

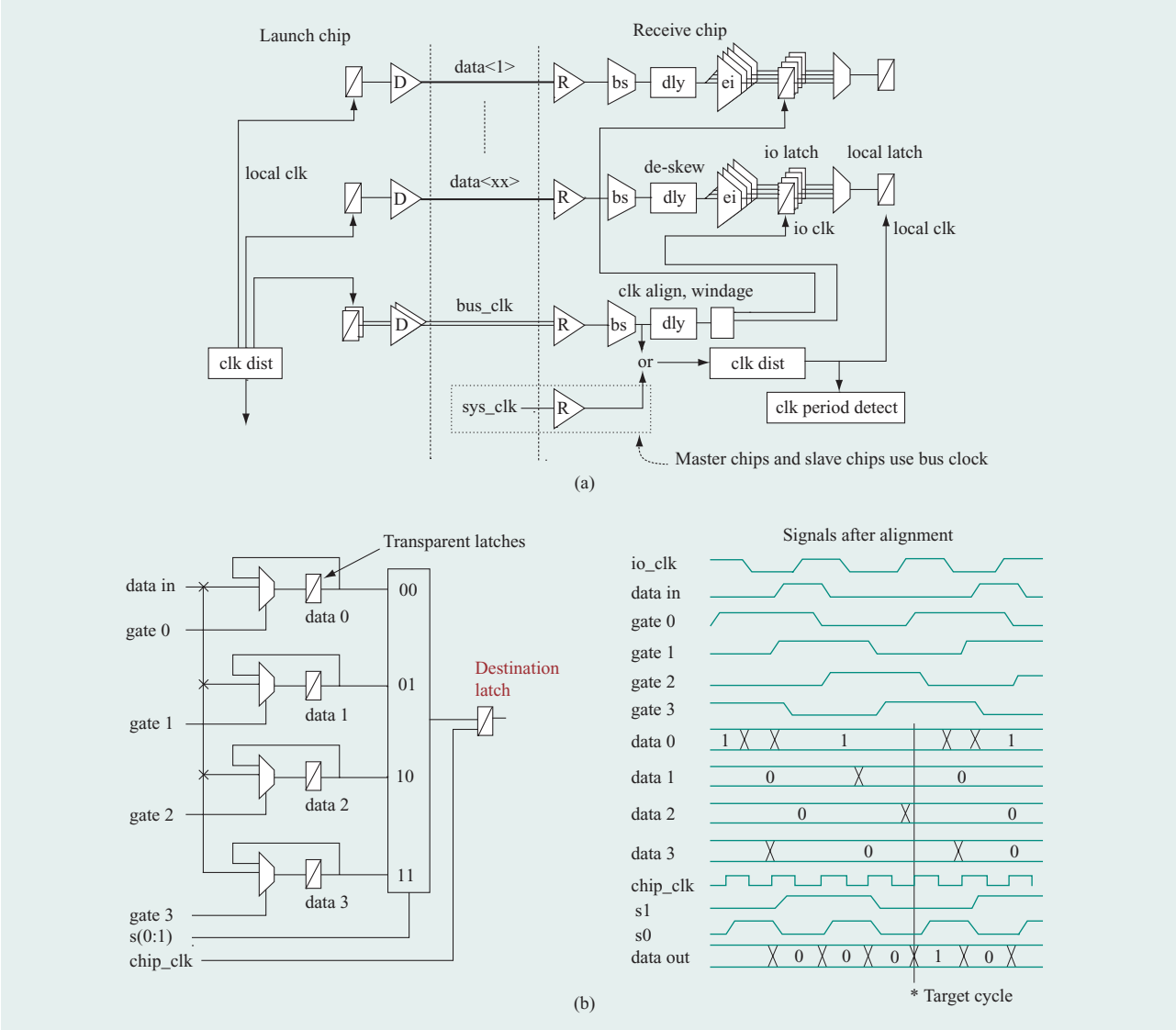

Figure 1

(a) High-level, conceptual block diagram of the Elastic Interface and its associated components. (dly: delay; bs: boundary-scan multiplexer; clk: clock; dist: distribution; ei: Elastic Interface; D: off-chip driver.) (b) Detail of EI FIFO. Note that data gates are shifted one bit at a time until a 1 is detected in data 0, and a 0 is detected in data 3, during the initialization alignment procedure (IAP). Thus, after alignment, the "1" in the IAP pattern is captured by latch 0.

which is a modulo-4 count. The target time is an analyzed value, expressed in bit times, and represents the latency in bit times between the transmit chip and the receive chip.

For the System z9, many unique features were invented and implemented to meet the System z* high-performance and reliability requirements. The following sections describe the System z9 EI-2 design and implementation, the parallelism of the z9* EI-2 bring-up and diagnostic controls provided by the serial interface facility (SIF), the advantages of various EI-2 diagnostic and self-test techniques, and the e-repair technique that enables the use of defective MCMs.

System z9 EI-2 designs and unique features

Figure 2 shows the System z9 central electronic complex (CEC), which uses EI-2 on most of its chip interface. Five types of Elastic Interface designs exist in the System z9 CEC, as listed in **Table 1** according to whether they are on-module or off-module, their elasticity (measured by the number of bit times or latch elements in the EI

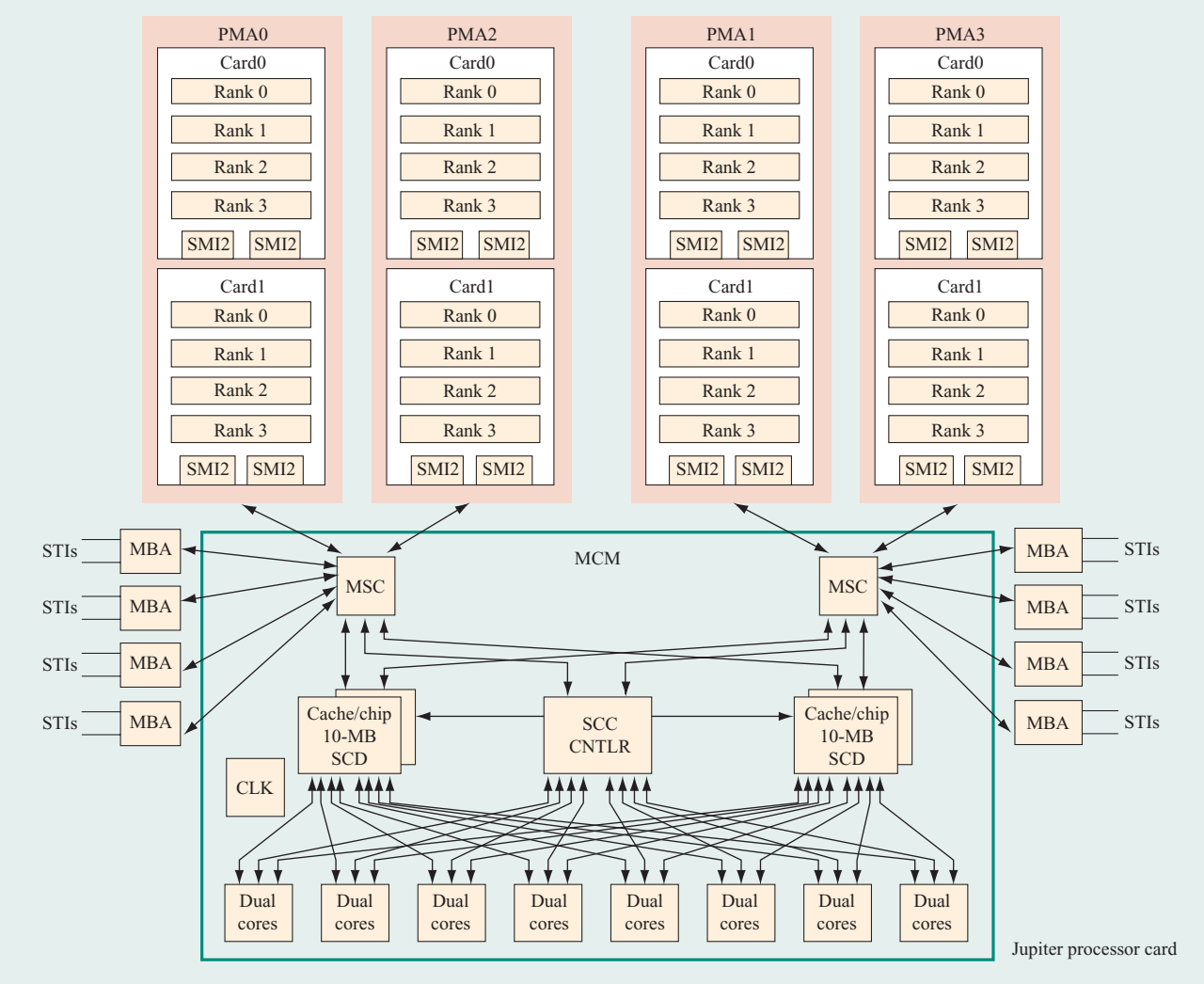

Figure 2

System z9 central electronic complex (CEC). (PMA: primary memory arrays; SCD: system control data; STI: self-timed interface; SMI2: synchronous memory interface 2; Rank: DRAM DIMM group ranking; SCC: system control chip; MBA: memory bus adapter; MSC: main storage controller; MCM: multichip module.)

receiver FIFO), and their bus clock frequency expressed as half the speed (HS) of the receive chip clock or as the full speed (FS) of the receive chip clock frequency. The EI designs are also categorized as EI master or EI slaves. The EI masters have their chip clocks synchronized by the on-chip phase-locked-loop circuits (PLLs) to a single system reference clock. EI slaves are chips that use the received bus clock as the reference source of their local clock. For example, the processor chips (also known as CP chips) employ an on-MCM half-speed EI-2 design with an elasticity of 4 for their interface to the system controller (SCC) chip and the L2 cache data (SCD) chips. In other words, the interface is on-MCM, the bus clock is half the frequency of the internal processor clock, and there is a four-bit-deep FIFO on the CP receive side.

The skew among the bits grouped within each bus clock group at the EI-2 receivers is realigned (de-skewed) by the programmable per-bit-de-skew (PBD) delay chains in the EI-2 receiver data bits, which correspond to the "dly" blocks in Figure 1(a). These PBD delay chains improve the EI-2 receiver sampling window because one common clock is used to sample all bits on the bus simultaneously. The delay differences among the bus-clock groups are compensated for by the built-in EI FIFOs. The maximum data skew among the bits, grouped within each bus-clock group at the EI-2

Table 1 Elastic Interface design types for the System z9. (CP: central processor chip; SCC: system control chip; SCD: L2 system cache data; MSC: main storage controller; SMI-2: synchronous memory interface-2; MBA: memory bus adapter; HS: half speed; FS: full speed; ps: picosecond.)

Chip	*On/off MCM*	*EI type*	*Elasticity*	*Bit rate* (ps)	*Per-bit de-skew* (ps)
CP	On	HS master	4.0	585	300
SCC/SCD/MSC	On	FS master	4.0	585	300
SCD daisy chain	On	HS master	1.5	1,170	300
SCC/SCD/MSC	Off	HS master	4.0	1,170	1,200
MBA/SMI-2	Off	FS slave	4.0	1,170	1,200

receivers, is limited by the range of the PBD delay chain. For the on-MCM buses, the maximum data skew must be less than 300 picoseconds. For the off-MCM EI-2 buses, the maximum data skew must be less than 1,200 ps. The receive bus clock is also delayed by a programmable delay chain. Limited by the length of the clock-delay element, the bus clock cannot arrive at the EI-2 receiver sampler latch more than 1.0 nanosecond earlier or more than 0.5 bit-time later than the latest arriving data bit. These EI-2 timing constraints were maintained across both the on-chip silicon paths and the wire trace length on the module, card, and board.

CEC chips and associated interfaces

The CEC shown in Figure 2 comprises the eight dual-core symmetric multiprocessor (SMP) CP chips, along with their supporting storage hierarchy and shared cache chips. The system controller (SCC) and four cache and dataflow (SCD) chips contribute the 40-MB second-level (L2) cache and manage all system coherency operations. The storage hierarchy extends beyond the shared cache to include two main storage controller (MSC) chips that provide all storage accesses to a maximum of four primary memory arrays (PMAs), which serve as the main storage of the system. Additionally, each MSC chip provides communication for up to four memory bus adapter (MBA) chips that act as a bridge between the CEC and the I/O subsystem. The CP, SCC, SCD, and MSC chips are all contained on an MCM mounted on a printed circuit board (PCB), which is called the *Jupiter processor card*. Also included on the MCM is a dedicated clock chip that provides a master oscillator to every chip on the module. Each PMA comprises two memory cards that are plugged directly into sockets on the Jupiter card. Finally, each MBA is mounted on a small daughter card that is also plugged into the Jupiter card and can be dynamically installed or removed while the system is operational.

The CEC components are combined with the necessary power and thermal entities and packaged together as a book. The book constitutes a fully functional SMP and serves as a logical node in the overall system structure. A System z9 can contain from one to four nodes. These nodes can be dynamically replaced or added while the system is operational. Because of the hierarchical nature of the packaging, the most performance-critical chip-to-chip communication occurs within the CEC MCM. The interfaces between CP, SCC, SCD, and MSC chips must maintain the required bandwidth, and many of these interfaces include logically wide (16-byte) datapaths that use a large number of pins. One of the key compromises in the System z9 design was to replace each System z990 bidirectional interface with two unidirectional interfaces. The immediate effect of this replacement might have been to double the number of signal I/O pins between chips, which would not have been practical. To overcome this challenge, we converted these unidirectional interfaces to EI-2 double-data-rate (DDR) buses, cycling at the CP frequency. Because the system controller chips run at half the frequency of the CP chips, this permitted us to multiplex the same amount of information on the DDR buses by transmitting half of the bits in the first half of the system controller (SC) cycle and the remaining bits in the second half. (Each half of an SC cycle includes one CP cycle.) In some cases, the two halves of the SC cycle transmit two logically associated pieces of information, such as eight bytes of data on the first half and the remaining eight bytes on the second half. In other cases, we chose to multiplex disparate information on the same physical bus—for example, cache access controls on the first half and diagnostic monitoring controls on the second half.

Because the CP chips and SC (SCC, SCD, and MSC) chips both operate from a shared system reference clock, their EI-2 interface is a master-to-master type. The master-to-master EI-2 design requires that the chip clocks on both ends of the interface must start on a known clock-cycle boundary. For the CP-SC EI-2 buses, the clocks on all of the SC chips start first in the same cycle, and the clocks on the CP chips start thousands (or even millions) of cycles later. The CP clocks are required to start at the same known modulo-4 cycles later in order

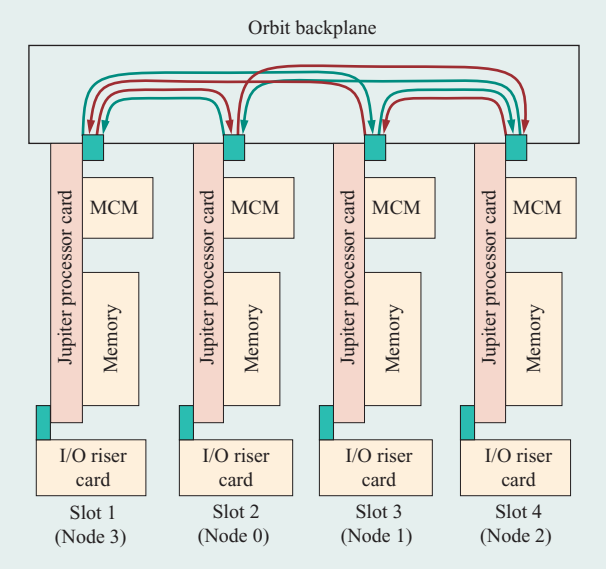

Figure 3

Physical representation of a four-node System z9.

to maintain a deterministic and fixed timing relationship. The EI-2 clocking constraints are achieved by starting the clocks on all chips on the edges of the same oscillator chip reference clock, which operates at one-eighth of the frequency of the processors.

Another CP-SC bus requirement was to achieve four CP cycles of latency because the L2 cache access latency has a direct impact on the system performance. To achieve this four-CP-cycle latency, a programmable delay circuit was used at the 2-to-1 MUX select line of the EI-2 full-speed drivers to reduce data launch delay. The clock control block, which launches data to the full-speed driver MUX, was also modified to make the first-half-latch (L1) clock rise early. Hence, this early-rising clock edge launches data early to one of the input lines of the EI-2 driver 2-to-1 MUX. CP-SC EI-2 bus latency for each chip is also adjustable by changing EI-2 receiver settings. The adjustable CP-SC bus EI-2 receiver setting permits each CP chip to have a different latency on its interface with the SC chips if some of the CP chips cannot achieve four-CP-cycle delay because of clock skew.

The SC-MSC buses, on the other hand, are full-speed EI-2 buses that have four- or six-CP-cycle latency depending on the system frequency. The full-speed EI-2 driver circuit techniques and receivers used for the SCC and SCD chips were also used for MSC chip physical designs. Within the memory cards exist four SMI-2 (second-generation synchronous memory interface) controller chips per memory card for interfacing with the DRAM chips. The MSC interfaces with the main memory on the DRAM chips via the SMI-2 chips. The MSC and SMI-2 interface is a master–slave design. The MSC EI-2 design is a half-speed EI-2 master, while the SMI-2 chips use the full-speed EI-2 slave design.

The System z9 I/O interface on the MSC chip is the first zSeries* system that implements the IBM common I/O protocols used by IBM processors to communicate with I/O bus bridges or hubs. This gives the System z9 a compatible I/O interface with the pSeries* and iSeries* systems. The MBA chips are EI-2 slaves of the MSC chips. The maximum transfer rate for the MSC–MBA interface is at or below the SC data rate. A synchronous-frequency "gear-ratio" circuit in the MSC internal logic circuits allows the MSC–MBA EI-2 buses to operate at a slower cycle time than the MSC chip internal logic in order to accommodate the slower clock frequencies of the MBA chips.

Local and remote L2 controller and data chip interface

The full System z9 configuration comprises four nodes. Each node is an independent SMP system with its memory cards and I/O cards. The four-node system contains 32 dual-core CP chips (a total of 64 processors with 54 processors for applications), and 160-MB shared Level-2 cache memory (40 MB L2 cache per node). **Figure 3** depicts a full four-node system, illustrating how each book (logical node) is physically plugged into a printed circuit board known as the *orbit backplane*. The interconnections between the nodes on the backplane form the dual concentric ring structure shown in **Figure 4**.

When the processors store or fetch data in the L2 cache on the SCD chips, data is transferred via the CP-SCD buses under the control of the CP and SCC chips. SCC-SCD buses exist for the L2 address and datapath control signals. The SCC-SCD buses are full-speed EI-2 buses which have four-CP-cycle latency. Because of an SC architecture constraint, this latency cannot be changed. The same programmable MUX select delay feature of the EI-2 full-speed driver was implemented in SCC-SCD buses in order to minimize the SCC-SCD bus delays.

In most of the EI-2 full-speed receivers, the DDR data received from the two consecutive data streams is assembled after the second data bit is received, and both bits are simultaneously delivered to the chip internal logic. To improve coherency performance for accesses of L2 data, the latency-critical address and control information is transmitted between the SCC and SCD chips on the first half of the DDR bus cycle. The EI-2 full-speed receivers on the SCD chips use the master latches' (L1) output of split-latches to deliver the first data sample from the SCC chip to the SCD chip internal logic, instead of waiting for the second data sample to arrive. The standard EI-2 full-speed receiver design was modified

especially for this purpose. The EI-2 receiver master latches' (L1) output is transmitted directly to the slave latches, which are placed at a significant distance from the EI-2 receiver logic to meet the chip timing requirement.

Another performance-critical factor in the System z9 structure involves the cross-interrogate (XI) buses that are used to broadcast coherency interrogation information among the processors. The XI address buses are from the SCC chip, redistributed through the SCD chips and forwarded to all of the processors on the MCM. To reduce the cumulative latency of the XI address buses from the SCC chip to the various CP chips, the output of EI-2 receivers on the SCD chips was routed directly to the SCD drivers of the CP interfaces. This direct connection between EI-2 receiver and driver employed the previously mentioned receiver-latch output implementation in order to bypass the internal chip delay.

Other new circuit and latch placement techniques were used to meet the chip-timing and physical-design requirements. One of the new features implemented in the full-speed EI-2 driver circuits was a configuration pin used to bypass one of the EI-2 driver master–slave latches. If this pin is asserted (i.e., configured or set to logic high), the EI-2 driver data latch for the second data sample is bypassed. The data is transmitted directly to the data launch latch. This technique allows us to place the EI-2 driver latches outside the EI-2 driver circuits to meet the driver-side internal cycle time. As mentioned previously, in the SCD chips, the bypassed full-speed EI-2 driver master–slave latches for the second data sample launch were placed farther away from the full-speed EI-2 driver circuits in order to reduce on-chip data delay. This technique was instrumental in meeting timing targets for the performance-critical cache access paths.

To share the L2 cache memory on other nodes, the two sets of two SCD chips drive EI-2 buses from the MCM to form two unidirectional fabric rings, as shown in Figure 4. The EI-2 design for the ring interface is the half-speed design, which operates at ~1.2-nanosecond bit times. Note that the SCD and SCC operate at half of the CP chip frequency. Because of the long ring-interface wire lengths and different conducting materials, these off-MCM EI-2 circuits use the pre-distortion voltage-mode drivers and reference-voltage forwarding techniques in their receivers (see [4] for further details on driver pre-distortion and voltage forwarding). When a System z9 has two or three nodes populated with Jupiter processor cards, the ring interface is completed with one or two passive jumper (passthrough) cards, which further increases the ring-interface wire lengths and delays. The ring latency is six to eight SC cycles without a jumper card, and seven to nine SC cycles with a jumper card because of the extra delay through the jumper card.

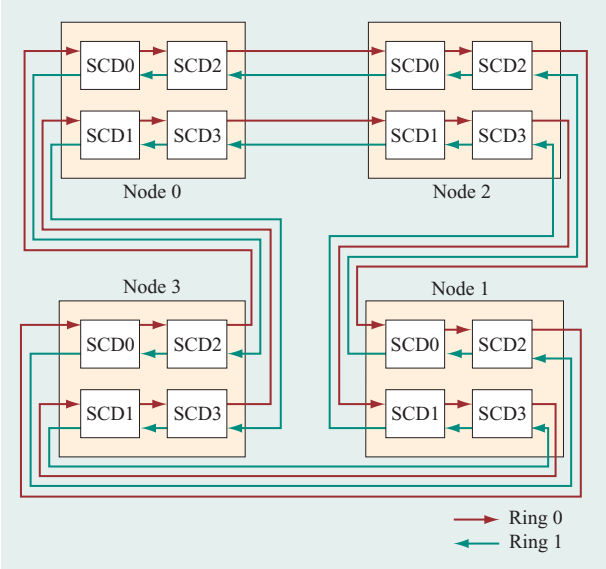

Figure 4

Two unidirectional fabric rings of a four-node System z9. One ring is depicted in red, the other in green. Each ring (interface) forms a unidirectional closed loop, using the EI-2 interface. The two rings serve to increase bandwidth and performance because any one of the four nodes communicates with its two adjacent nodes without going through a third node.

The ring-interface EI-2 receiver target cycle, which resynchronizes data with respect to the receiver chip, becomes increasingly difficult to predict because of the node-to-node clock phase skew. The half-speed EI-2 receivers have an Elasticity-4 design, which has four bit-time FIFOs. Because the ring data at the receiving end of the SCC chip and two SCD chips must receive signals in the same logical cycle, a risk existed that they would be unable to resynchronize data with the internal logic of the receiver chips. Constrained by product schedule, chip power, and various other technical risks, we found that it was infeasible to expand the Elasticity-4 design to a more complicated Elasticity-8 design. To reduce the risk associated with the elasticity design, we implemented programmable cycle-delay staging latches in the EI-2 driver side. By using these staging latches to delay the launch of the lowest-latency data, we could effectively compensate for static skew and delay fast bus paths. We were able to meet the timing requirement with the staging latch and Elasticity-4 receiver FIFOs, ensuring synchronization with the internal logic cycle of the receive chips.

In addition to the two SCD chips connecting each node on the ring, an on-MCM interface exists between the two pairs of SCD chips to complete the ring on the MCM (shown by the SCD chip connections within each node in

Figure 4). To ensure peak system performance, the on-MCM SCD–SCD interface must have one cycle of total delay. However, the system-level timing analysis had shown that a potential early-mode hold-time exposure existed at the required cycle time owing to clock uncertainty. To eliminate this hold-time risk, we designed a new 1.5-Elastic interface ("EI-Lite"). The EI-Lite design has less circuitry and less delay in the datapaths, allowing us to eliminate the early-mode risk while maintaining the goal of one cycle of latency. The design uses the same source-synchronous technique as EI-2 to achieve tolerance to clock skew, jitter, and CMOS process variations. In the EI-Lite receivers, a 1.5-depth FIFO was implemented to increase the data-valid time by half a cycle for clock uncertainty.

System z9 EI-2 bring-up and diagnostic control scheme

Background on existing hardware and requirements

Because of the large number of EI-2 interfaces in a single System z9, the design and implementation of the hardware validation and bring-up process were quite challenging. The previous generation of System z platforms, which employed the Elastic Interface-I, used hardware counters or a small group of registers to set calibration controls for each interface, but these prior systems had only small numbers of elastic interfaces.

The complexity of the System z9 EI-2 design (which controls more than 200 interfaces) and our desire to reuse as much as possible of the existing system hardware led to a design that allowed greater flexibility and speed than previous systems. New hardware and processes were designed that allowed control information to be transmitted over interfaces, either individually or as part of a broadcast that permitted a parallel bring-up of multiple interfaces. By leveraging the existing hardware, we could use a common set of controls, dataflow, and code for all interfaces.

Existing hardware

The System z9 SCC, SCD, and MSC chips configure system settings and read status through the use of a proprietary register structure which is read- and write-addressable through CP code. The serial interface facility (SIF) is the hardware used to access these registers. The SIF receives and sends data from and to the clock chip and ultimately communicates with the system support element (SE) via a 1-bit-wide bidirectional bus. Using this hardware, the typical path from SE to control register may be outlined as SE to clock chip; clock chip to CP, SCC, SCD, or MSC via SIF; and finally from SIF to the chip UBus register.

New hardware

The new hardware implemented on the System z9 was designed to take advantage of the SE-to-SIF paths and to follow the same format as used for the UBus registers. Each chip (CP, SCD, SCD, and MSC) has logic to act as a hub that receives signals from the SIF, including a 7-bit address, a 48-bit data field, a broadcast mode bit, and a 4-bit op-code (read, write, set, reset). Five bits of the address are used to specify a particular interface, and the remaining two bits are used to address one of four 48-bit registers.

Each set of four 48-bit registers is allocated for each driver–receiver pair on each chip. However, these driver–receiver pairs are not used for the same bus, but are paired on the basis of their physical location on the chip, such that the corresponding drivers and receivers are on a different chip that shares a similar set of registers. On the SCD chip, for example, an SCD-to-SCC driver may be paired with an SCC-to-SCD receiver. The set of four registers is defined as follows:

- Register 00: EI-2 control (e.g., self-test mode, IAP, and target cycle).
- Register 01: Self-healing and EI-2 status reporting. (Self-healing is described in the section on e-repair below.)
- Register 10: Driver self-healing controls.
- Register 11: Receiver self-healing controls, I/O controls, fencing, EI-calibration, and related controls. (The term *fencing* refers to a gating off from other chip logic.)

In addition to the register set, each chip has logic to act as an EI-2 calibration and diagnostic control hub, distributing data from the SIF to the interface controls and collecting status and other information to send back to the SE. Because multiple driver and receiver pairs are grouped together in sets of one, two, or four, a level of hierarchy exists between the hub and the register sets. The primary function of the hub is to direct data to the appropriate group on the basis of the three bits of the address. Another control block directs the data to up to four register sets (**Figure 5**). Four such control register sets are shown in the left group in Figure 5.

Broadcast mode

The register sets depicted in Figure 5 can be accessed in an individual mode, for calibration and diagnostic steps that need to be done on only one interface at a time, or accessed in a broadcast mode in which a common set of controls can be used by multiple interfaces, allowing calibration steps to be performed in parallel. (Examples of these diagnostic steps include wire tests, which

are described in subsequent sections of this paper.) Register-set access is an important part of the design, because a typical SE-SIF command can correspond to approximately 3,000 cycles. With more than 200 interfaces, the capability of using only one SIF command to broadcast the same controls to all interfaces drastically reduces the time needed to initiate calibration and read results.

During the initial system microcode load (IML), scan-only latches in each hub are initialized in order to specify which groups of interfaces will be part of up to five broadcast groups, one broadcast group for each bit of the address (seven bits total are used, with two bits used to specify the target register). In the individual mode, the hub typically controls a group of interfaces based on three bits of the address. Because up to eight groups can be controlled individually, 40 scan-only latches are used to specify the broadcast groups. This provides the flexibility of allowing each interface to be included in any or all of the five broadcast groups, of which one can be a subset or superset of another (see Figure 5). For example, Group 00100 is a subset of Group 00010. During a broadcast operation, the hub compares the five bits of the address against the five sets of eight groups and routes the data to the appropriate interface.

Also during IML, each driver–receiver pair sets a scan-only latch that determines whether or not a particular register set will be included in a broadcast operation. If the latch is set to ignore broadcast operation, it can be accessed only during an individul mode command.

Global status collection

Each EI-2 receiver accumulates detailed status information relating to calibration results, which can be accessed by reading the status register. For the majority of the cases, it is not necessary to collect detailed information, and only a summary of the results is needed to determine whether calibration was successful. The detailed status register for the receivers is summarized using two status bits that are sent directly to the chip EI-2 control hub. During a "broadcast read" operation, the 2-bit status from each receiver is consolidated to form a 48-bit data field to be returned to the SE. In this manner, a read operation can be issued and data can be returned in a normal fashion, but by issuing the read in a broadcast mode, the data returned can be interpreted to represent each interface. If all interfaces are calibrated successfully, no further SIF commands are required.

EI-2 diagnosis and self-test features

All interfaces on the System z have some form of error checking, such as parity checking, protocol handshaking, or error-correction coding (ECC), in order to improve reliability, availability, and serviceability (RAS). While

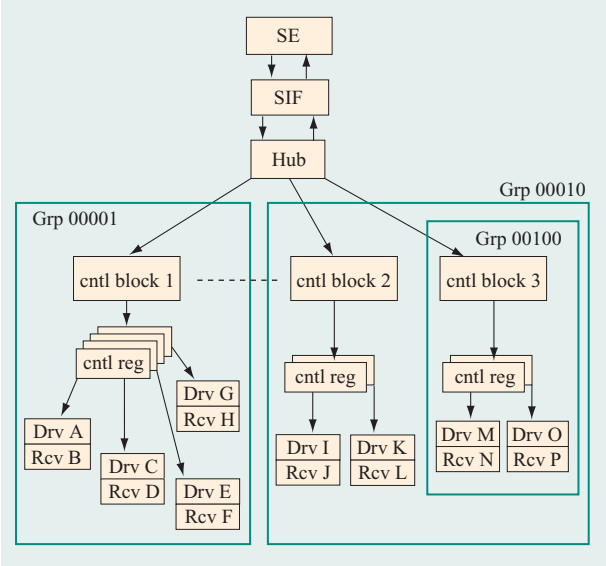

Figure 5

Overview of chip-level control structure and broadcast groups. Note that group 00100 is a subset of 00010. (SE: support element; SIF: serial interface facility; Grp: group; Cntl: control; Drv: driver; Rcv: receiver; Reg: register.)

error checking can determine that a data transaction has some form of error, isolation of the fault has traditionally been a challenge. A parity test or an ECC station can check the condition of the transfer just before the data is transmitted to the I/O logic, and such checks are typically conducted immediately after the I/O on the receiving end. When an error is detected, the failure could occur at any number of points between these error checkings. The built-in diagnostic functions of the EI allow the transmission of deterministic or pseudorandom patterns that help isolate the I/O hardware and package section during debug and fault isolation operations.

Three general categories of diagnostic function exist:

- Continuity/shorts testing (a dc test).
- Pseudorandom data testing (an ac test).
- Guard-band testing (a qualitative ac test).

The continuity/shorts test (CST) consists of two parts. One driver–receiver pair performs a test in which either a one or a zero is "walked" across all of the data signal paths between the driver and receiver. All wires on the interface are set to the same value (i.e., a one or a zero). Then the opposite value is sent for a time and then reset. The test proceeds in series from first to last wire on the interface. The frequency of the transitions is sufficiently slow that it is effectively dc relative to the length of wire on the interface. Since the clock is still used to sample

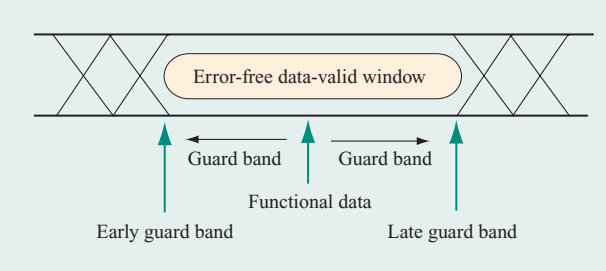

Figure 6

Elastic Interface guard bands on receive chip. The XX symbols denote invalid regions.

data at the receiver, it cannot be tested independently; however, we can infer from a case in which no data signals pass the test that the clock signal lacks continuity. Every data signal has a corresponding pass or fail register. The pass/fail register state is considered during calibration at runtime so that an interface can optimize calibration of the remaining valid signals. While the one receiver/driver pair is doing a walking one or zero test, all of the other receivers in the system are in a state that watches for transitions. These other receivers' corresponding drivers are quiescent. A transition on these quiescent interfaces is interpreted as a short arising from a signal on another interface.

Pseudorandom data testing (RDT) may be used after an interface is successfully calibrated. Synchronized pseudorandom pattern generators on the driver and receiver are used to generate and then check the data transferred from transmit chip to receive chip across the interface. Each bit is checked on a per-bit basis, using the functional datapath that included the local latch on the receive chip. The data is transferred at or above full-product speeds to ensure the proper end-to-end functioning of the interface operation. The pattern generation comes from a 12-bit counter, and thus the pattern repeats every 4,096 cycles. RDT tests can be accomplished in parallel on all interfaces in a system, which maximizes the potential for observing crosstalk among buses. As with the CST test, a pass or fail is recorded for individual signals on the interface.

The EI-2 receiver guard-band feature is an improvement over the previous first-generation Elastic Interface design, which did not have such a feature. The interface receiver has built-in guard-band logic (**Figure 6**), which continually compares samples of the data at different points in time. Ideally, the data window is as wide in time as half the period of the bus clock. However, because of noise, clock jitter, intersymbol interference, and other factors, the data-valid region (i.e., the period in time in which the data can be reliably sampled, also referred to as the *signal eye opening*), is less than one bit time, which is half the period of the bus clock. The EI calibration algorithm forces the guard bands to automatically sense the outer region of the data-valid window and then center the functional sampling point within the signal eye. Thus, the functional sampling point has the optimal set-up and hold times. The guard bands can also be used to determine the amount of timing margin by directly reading the guard-band values. These guard-band values are a much better indication, relative to measurements external to the chip, of the signal quality because they are measured by sampling latches inside the chip. Note that this is exactly the same sampling latch and margin as the functional path. There are no offsets, reflections, or loading effects that generally affect a measurement at a module pin or C4 (controlled collapse chip connection) outside the chip. (A C4 is also often referred to as a "C4 solder ball.")

A few mechanisms are built into the System z9 implementations that permit testing and monitoring of the guard-band condition. When the IAP process completes, one of the final steps is to perform a simple measure of the signal guard band. Additionally, in a system in which interfaces can be fenced off periodically, the EI-2 allows for a recalibration that takes into account the status of every guard-band compare register on a bus. When the interface is fenced, the driver sends a sequence of transitions or patterns to the driver's corresponding receiver. These patterns compensate for cases in which a bus may not have had any activity since the last recalibration. Depending on the guard-band compare register conditions, the following events may result:

1. Both guard-band registers indicate a miscompare, and the guard band is decreased one step.
2. Both guard-band registers indicate post-functional sample miscompares. De-skew of the bit is increased one step.
3. Both guard-band registers indicate pre-functional sample miscompares. De-skew of the bit is decreased one step.
4. Both guard-band registers indicate no miscompares. Guard band is increased one step.

Once recalibration is completed, all of the guard-band miscompare registers are reset.

The values of EI-2 guard bands are preserved during system operation unless reset occurs. Registers also exist to detect when a guard band reaches a maximum value (determined by the amount of silicon delay in the guard-band delay chain) or zero value, as well as detecting that de-skew registers are reaching the extremes of their ranges. One additional compare has been included as a

System z9 design point in order to allow a preset for a minimum threshold of the guard band. This condition sets a flag that serves as a warning to the firmware that the guard band has become so degraded that a signal is in danger of being incorrectly sampled.

E-repair for on-MCM EI-2 bus wires

In order to improve the manufacturing yield of the ceramic substrate used as the base of the multichip module (MCM), a redundant (spare) wire was included during manufacturing on every EI-2 clock group for all of the EI-2 buses within the MCM. For the high-speed (less than 600 ps) bit–time interface, a clock group was limited to at most 16 data lanes. If one of the data lanes is determined to lack continuity at final substrate testing (e.g., as caused by an open electrical circuit), it is recovered via the e-repair technique shown in **Figure 7**. If lane n is found to be open, the MUX selects all lanes, in the range from $n+1$ to the end of the repairable group, so that they are switched to receive the data from the previous lane. Because this repair feature is accomplished outside the EI-2 logic, it does not affect the performance of the interface.

To be certain that defective lanes are not improperly handled, the control logic must be informed of the defect to allow interface calibration to be properly accomplished. Two principal mechanisms exist to inform the control logic of the defect. Elastic Interface built-in diagnostics can self-diagnose and flag the existence of bad lanes. However, because the connectivity of the substrate and its attached chip is known and considered to be a constant, control logic can be preloaded with this repair information at power-on, saving the time required to run diagnostics during bring-up.

The technique of adding redundancy for manufacturing yield improvement is similarly employed in other parts of the System z platform, particularly in large memory arrays. Manufacturing defects detected during chip tests can be circumvented. As in the case for substrate connectivity, a defective memory subarray is in a constant state, and the configuration is loaded at power-on. In the case of EI-2 repair, this state information is retained on the MCM in electronic fuses. After all chips are attached to the MCM, and testing confirms that no additional defects have been introduced during this step, an array of electronic fuses is written with the wire-repair state information. During system power-on, firmware reads the state of the electronic fuses and transfers the information into the e-repair control latches.

Concluding remarks

In this paper, we have described the proprietary high-speed source-synchronous interface technique known as the Elastic Interface, which is used for the chip-to-chip communication within the System z9 platforms. With the second-generation Elastic Interface (EI-2) design implemented throughout almost all of the chip-to-chip interfaces, the System z9 achieves nearly twice the bus interface bandwidth with one half of the chip pins and off-chip module (MCM and card) wires compared with the previous System z990. Innovative logic, circuit, and physical design techniques were used to develop the System z9 EI-2, and these techniques were also instrumental in making the System z9 chip-to-chip communications extremely robust and reliable. The flexible parallel EI-2 scheme that utilized the existing System z SIF and UBus facilities greatly enhanced the productivity of EI-2 validation in the engineering test laboratories. The scheme permits easy EI-2 bus diagnosis via firmware during system power-on or when the system is already on and running. The EI-2 built-in self-test features were used effectively during the System z9 development, testing, and characterization stages in order to isolate defects and abnormalities. Programmable target-cycle settings allow us to use firmware to overcome any problems arising from process variations during manufacturing. We also used the EI-2 built-in self-test features in order to select and adjust numerous EI-2 settings to achieve optimal performance and to make the interface more reliable. The new e-repair feature allows

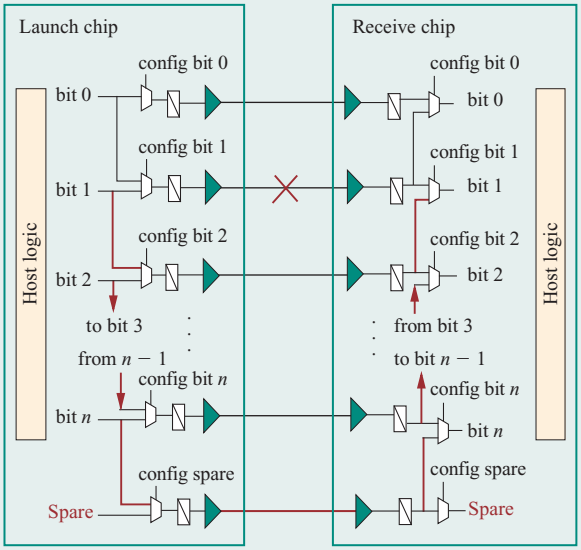

Figure 7

E-repair technique rerouting the defective wire. The red X represents an open wire in the MCM. The red lines indicate how the signals are rerouted to gain use of the spare wire and circumvent the defect. Note that bit 0 to bit n maintain their respective relationships within the chips.

the System z9 to use partially good MCMs, which otherwise would have had to be discarded.

Acknowledgments

We acknowledge the many contributions of a variety of individuals who have contributed their time and skills to achieve the success of the System z9 EI-2. The authors are especially indebted to David Webber for his innovations and extra efforts on EI-2 circuit designs; Alan Wagstaff and Jayanth Jayaram for their EI-2 design process automation and integration work; Bao Truong for his EI-2 I/O circuit-design and signal-analysis work; Ken Christian for his system-level timing analysis work; and Rob Reese and John Gullickson for their logic design and technical support.

References

1. HyperTransport Consortium, "Welcome to the HyperTransport Consortium"; see *http://www.hypertransport.org*.
2. E. Cordero, F. Ferriaolo, M. Floyd, K. Grower, and B. McCredie, "A Synchronous Wave-Pipeline Interface for POWER4," presented at the *IEEE Computer Society HOT CHIPS* Workshop, Stanford University, California, 1999.
3. T.-M. Winkel, W. D. Becker, H. Harrer, H. Pross, D. Kaller, B. Garben, B. J. Chamberlin, and S. A. Kuppinger, "First- and Second-Level Packaging of the z990 Processor Cage," *IBM J. Res. & Dev.* **48**, No. 3/4, 379–394 (2004).
4. H. Harrer, D. M. Dreps, T.-M. Winkel, W. Scholz, B. G. Truong, A. Huber, T. Zhou, K. L. Christian, and G. F. Goth, "High-Speed Interconnect and Packaging Design of the IBM System z9 Processor Cage," *IBM J. Res. & Dev.* **51**, No. 1/2, 37–52 (2007, this issue).

Received March 22, 2006; accepted for publication May 10, 2006; Internet publication December 5, 2006

Derrin M. Berger *IBM Systems and Technology Group, 2455 South Road, Poughkeepsie, New York 12601 (derrin@us.ibm.com)*. Mr. Berger received his B.S. degree in electrical engineering and his M.Eng. degree in electrical and computer engineering from Cornell University in 2001 and 2002, respectively. He worked on Elastic Interface design and simulation for the System z9 and is currently working on the System z L2 cache logic design.

Jonathan Y. Chen *IBM Systems and Technology Group, 2455 South Road, Poughkeepsie, New York 12601 (jychen@us.ibm.com)*. Mr. Chen received his B.S. and M.S. degrees in electrical engineering from Rutgers University. Prior to joining IBM, he was a research staff member at Philips Laboratories of North American Philips Corporation. Since joining IBM in 1996, he has been working on cryptographic hardware design and Elastic Interface designs for two previous System z platforms as well as for the System z9. He is currently a circuit-design team leader for the next-generation System z platform. Mr. Chen holds three U.S. patents related to the Elastic Interface.

Frank D. Ferraiolo *IBM Systems and Technology Group, 2455 South Road, Poughkeepsie, New York 12601 (ffrank@us.ibm.com)*. Mr. Ferraiolo graduated from The Pennsylvania State University, joining IBM in 1982. He is currently a Distinguished Engineer in the IBM Systems and Technology Group. He has worked with multiple IBM teams to design and develop the IBM ESCON* fiber optic interface, the self-timed interface, and, most recently, the Elastic Interface. Mr. Ferraiolo holds approximately 60 U.S. patents in data communications and has received several IBM Outstanding Innovation and Technical Achievement Awards. In 2002, he also received an IBM Corporate Award for the design of the Elastic Interface.

Jeffrey A. Magee *IBM Systems and Technology Group, 2455 South Road, Poughkeepsie, New York 12601 (jm5@us.ibm.com)*. Mr. Magee received his B.S. degree in electrical, computer, and systems engineering from Rensselaer Polytechnic Institute in 1997; he is currently an Advisory Engineer at IBM. His principal areas of focus for the System z9 were I/O development, system bring-up, and hardware characterization. For the previous System z990, he was the lead engineer responsible for implementation of cryptographic hardware features on the microprocessor. Mr. Magee continues to lead the development of high-performance interfaces for future IBM systems.

Gary A. Van Huben *IBM Systems and Technology Group, 2455 South Road, Poughkeepsie, New York 12601 (vanhuben@us.ibm.com)*. Mr. Van Huben joined IBM in 1986; he is currently a Senior Engineer in the System z Hardware Development Laboratory. He has worked on several logic design and verification areas within the processor subsystem, and is currently the System z9 Team Leader for the Storage Hierarchy Design Team. Mr. Van Huben graduated from Clarkson University with a B.S. degree in electrical and computer engineering. He holds more than two dozen U.S. patents in the areas of system coherency, protocols, and data management processes, and he has received several IBM Outstanding Innovation and Technical Achievement Awards.