Run-control migration from single book to multibooks

T. Webel
T. E. Gilbert
D. Schmunkamp

This paper describes the migration of the hardware-implemented run-control functions from a single-book structure with one flexible service processor (FSP) and one service element (SE) per system to a multibook structure with one FSP per book and one SE per system. The new system structure required two new interfaces between the clock chips on the different books. The first interface is required for dynamic configuration data exchange between books. The alternative path via the SE would not meet the performance requirements. This interface is available in the initial millicode load flow before the L2 caches with their ring structure are operational. Another requirement is the necessity of starting and stopping all books synchronously. The second additional interface between the clock chips on different books enables this function. Nevertheless, the hardware implementation is so flexible that each book may operate independently of the other books. The clock chips are connected as a peer-to-peer network, so no special master is necessary in the system.

Introduction

A book basically consists of the chips required for the processor, cache, memory controller, memory bus adapter, clock, and some support for external connections. Previous zSeries* systems, G3 to G6, were limited to one book [1–4]. The clock chip—which provides functions to support the chips of a book during setup of the entire system and also support functions in a running environment—is the central point of the hardware-implemented run-control structure. (Run control summarizes all functions to support chips of a book during the initialization phase, start phase, in error scenarios, and in system error-recovery scenarios.) The clock chip is the gate for the service interface into the system (**Figure 1**). The service interface sends commands to the clock chip, and the clock chip provides the status of the entire system to the service element (SE). From the point of view of a clock chip, the SE is responsible for all support functions addressing several books. The service interface consists of an SE that connects to the flexible service processor (FSP) via an Ethernet connection, and the FSP connects to the clock chip via a proprietary interface. Again from a clock chip viewpoint, the FSP is responsible for all support that is handled locally within a book.

For reliability reasons, there are two FSPs connected to the clock chip via two independent interfaces, but only one FSP may be active at a time; the other is in standby mode. The new structure has a common SE, but two dedicated FSPs per book and up to four books. Between the books, there is a high-speed interface connecting the L2 chips which is used to maintain memory coherence. The L2 interface is a ring topology and requires a calibration sequence to become operational. All other information must be routed via the SE under control of firmware.

Clock-to-clock interface

In addition to the two interfaces (for reliability) between the clock chip and FSP, there is a clock-to-clock interface (**Figures 2** and **3**). It does not require any calibration, and thus has a significantly lower bandwidth than the L2 interface. On the other hand, since it is totally implemented in hardware, it is much faster than the path via the SE. Its main purpose is to exchange configuration

0018-8646/04/$5.00 © 2004 IBM

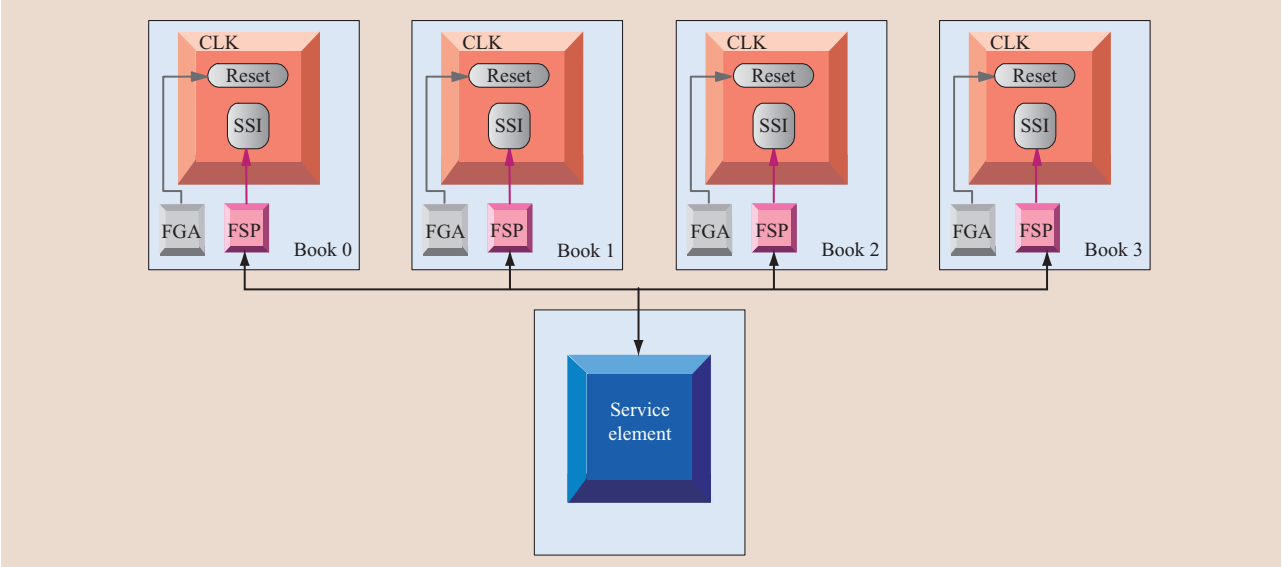

Figure 1

System structure of a four-book system. Only active FSPs are shown. (CLK: clock chip; FGA: field gate array chip, which provides reset signals to the clock chip; FSP: flexible service processor; SSI: system support interface.)

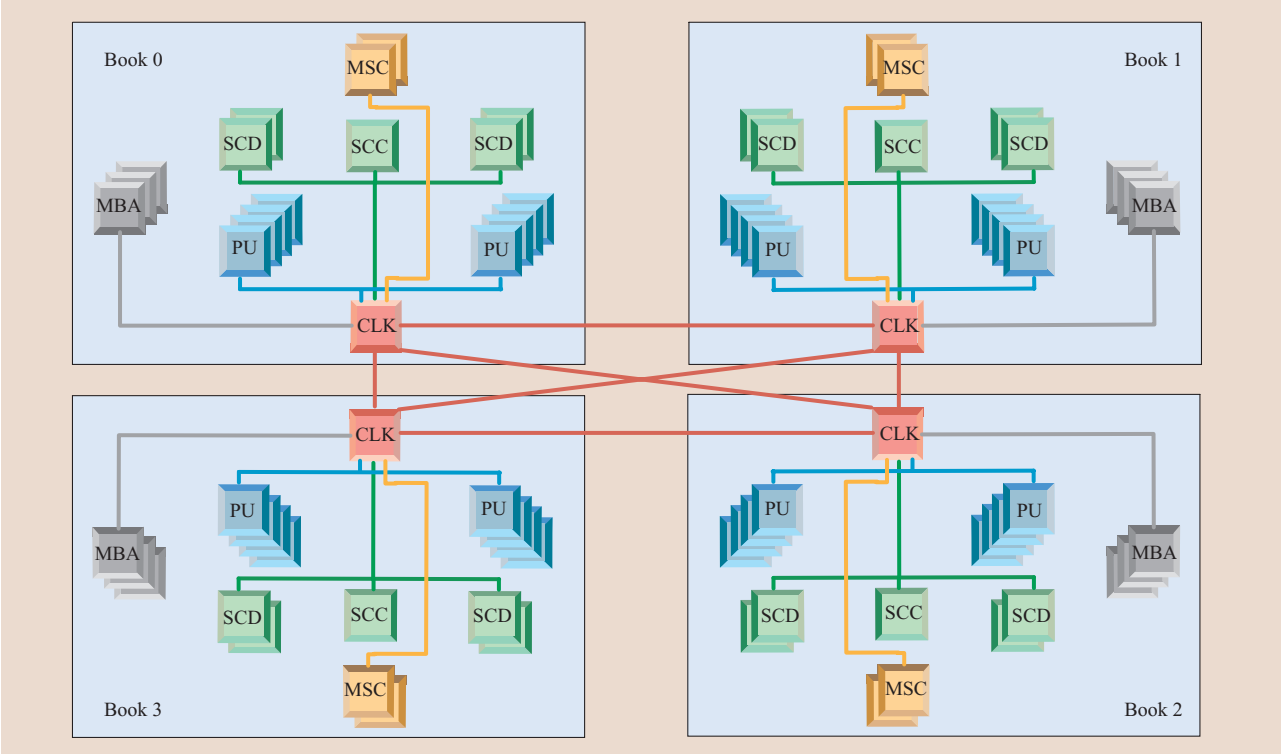

Figure 2

Wiring between clock chips. (CLK: clock chip; MBA: memory bus adapter chip; MSC: memory storage controller chip; PU: processing unit; SCC: SCE control chip; SCD: SCE data chip.)

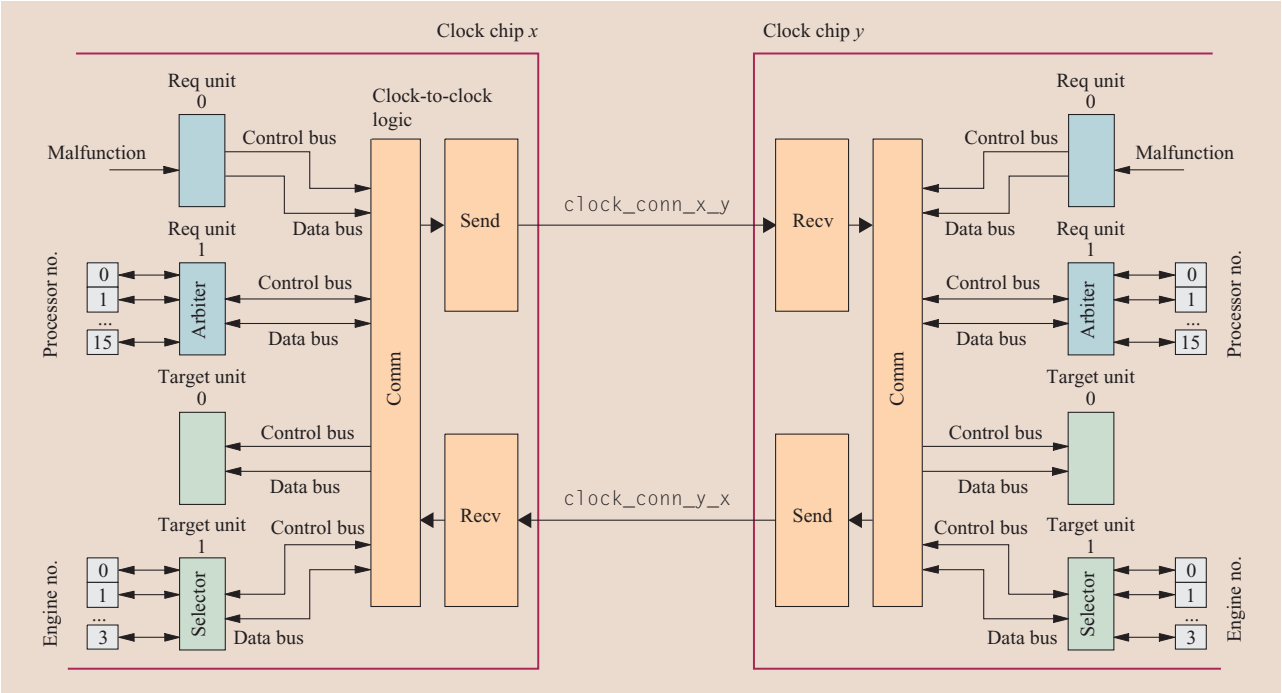

Figure 3

Clock chip internal logic.

information between the books. Configuration information consists of the availability of chips and the state of chips in a running environment of all books in the system. Thus, each clock chip is still capable of providing system-wide configuration information to the connected processing units (PUs) without having physical access to all PUs. This hardware interface is much faster than the path via the SE, is readily available after power-on, and occupies only a small number of pins. The topology of this interface is a star; each clock chip connects to each other clock chip with a 4-bit-wide bus in both directions. Though the star topology requires more signal I/Os than a ring topology, the operation of a star structure is much easier because it is not necessary to pass commands from one clock chip through to the target clock chip via an intermediate clock chip. The star structure also has advantages in systems that are not fully populated when one or more books are unplugged.

Protocol definition

The protocol definition distinguishes between two different frame types (**Figure 4**). The long frame type is used to transfer data on a unidirectional clock-to-clock bus from clock chip x to clock chip y. The short frame type is used as a delivery status identifier on the

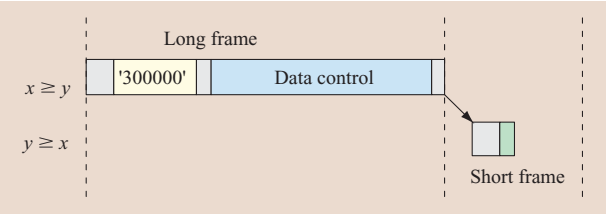

Figure 4

Inter-clock-chip frame and response.

unidirectional clock-to-clock bus from clock chip y to clock chip x. It acts as a response to the previous long frame.

The start of each frame type (Beat 1) is defined as B'1111' on the 4-bit unidirectional bus. This implies that the clock-to-clock bus always drives zeros on the bus when in idle mode. Because of the synchronous definition of the clock-to-clock bus, the following data transfer is completely cycle-dependent.

The second beat (beat 2) delivers the frame-type information. The definition of frame type is determined by the sum of 1s in the frame type; for a long frame, $sum = 1$; for a short frame, $sum = 4$. The Hamming distance (the number of digit positions in which the corresponding

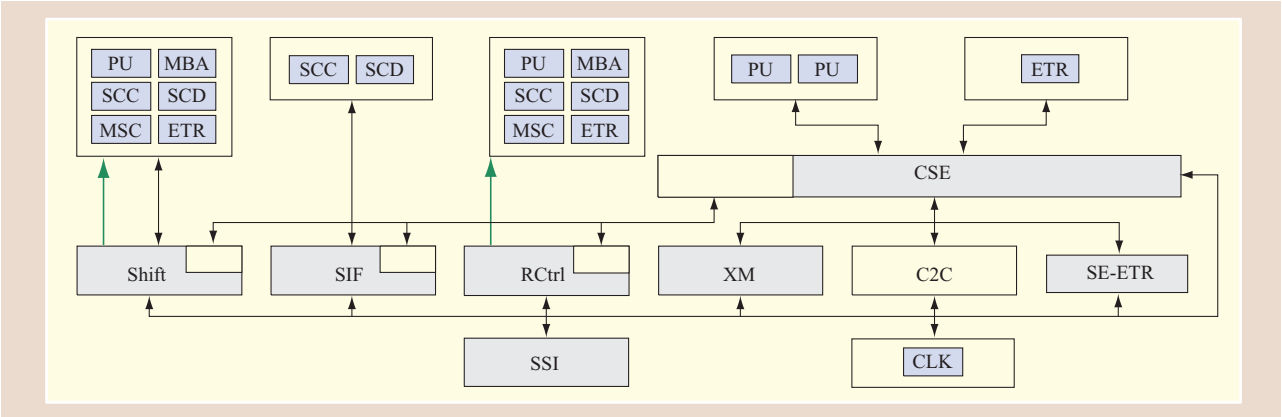

Figure 5

Access to clock-chip facilities from the processing unit (PU) and flexible service processor (FSP). (C2C: clock to clock, the interface between clock chips for data exchange; CLK: clock chip; CSE: clock-support element, the interface to PU chips for data exchange; ETR: external time reference; MBA: memory bus adapter; MSC: memory storage controller chip; RCtrl: run control, the interface to all chips of a book for run-control functions; SCC: SCE control chip; SCD: SCE data chip; SE-ETR: the internal service element (SE) interface to the external time reference (ETR) logic; Shift: interface to all chips of a book for shifting; SIF: serial interface, the interface to cache chips for data exchange; SSI: system support interface, the interface to and from the FSP; XM: X-message, the internal clock-chip logic responsible for service word communication [5].)

digits of two binary beats of the same length are different) between a short frame and long frame is 3. Therefore, single-bit errors in the frame type do not lead to wrong frame-type detection.

For a long frame, beats 3–8 contain a 32-bit command field. Beat 9 for a long frame is a modulo 16 checksum field derived from the previous 1–8 beats. Beats 10–25 are defined as a 64-bit data field. Beat 26 again contains a modulo 16 checksum field derived from the previous 10–25 beats.

The short frame is used as a response with delivery status to a previous long frame. Beat 3 has two different options. B'0000' defines *delivery status okay*, B'1111' defines *delivery status error*. Again, a Hamming distance of 4 for beat 3 protects the delivery status information against single-bit errors.

Symmetric access of clock-chip facilities

Besides the new interface described above, the new clock chip allows access to some of its internal facilities not only from the FSP, but also from the PU itself (**Figure 5**). This symmetric approach allows execution of run-control functions from a PU chip. Access from the PU is in an inband operation and is much faster than access from the FSP or even the SE (in the case of different books) and has a real-time response that is not available from the FSP. Examples for run-control functions executed by a PU are *PU sparing* and *unfencing of the L2 ring*, which are described in the following sections and shown in **Figure 6**.

New function across books

The new hardware communication layer that was introduced between the clock chips allows the following functions across books: PU sparing, system control element (SCE) ring calibration and unfenced sequence, synchronous start and stop, checking recognition and handling, and malfunction alert processing.

PU sparing

In case of a PU *checkstop* (a stoppage of the clocks of the PU chip) in a running system, the particular PU is disabled. The PU sparing operation begins when another PU then shifts the data out of the checkstopped PU and places it in main memory. A spare PU can take over the workload of the checkstopped PU. The PU that shifts the data out of the checkstopped PU may be either on the same book or on a different book. The read is nondestructive, and the state of the PU is preserved during the shift operation. The *shift engine* (the logic part of the clock chip that provides the interface to shift any chip of a book) supports a special skip command that rotates the data in the checkstopped PU without a data transfer to the spare PU. Thus, only relevant data is transferred into the spare PU, which optimizes the required time for a PU sparing action. After the spare PU has finished the sparing operation, the checkstopped PU can be analyzed by the FSP for the root cause of the checkstop.

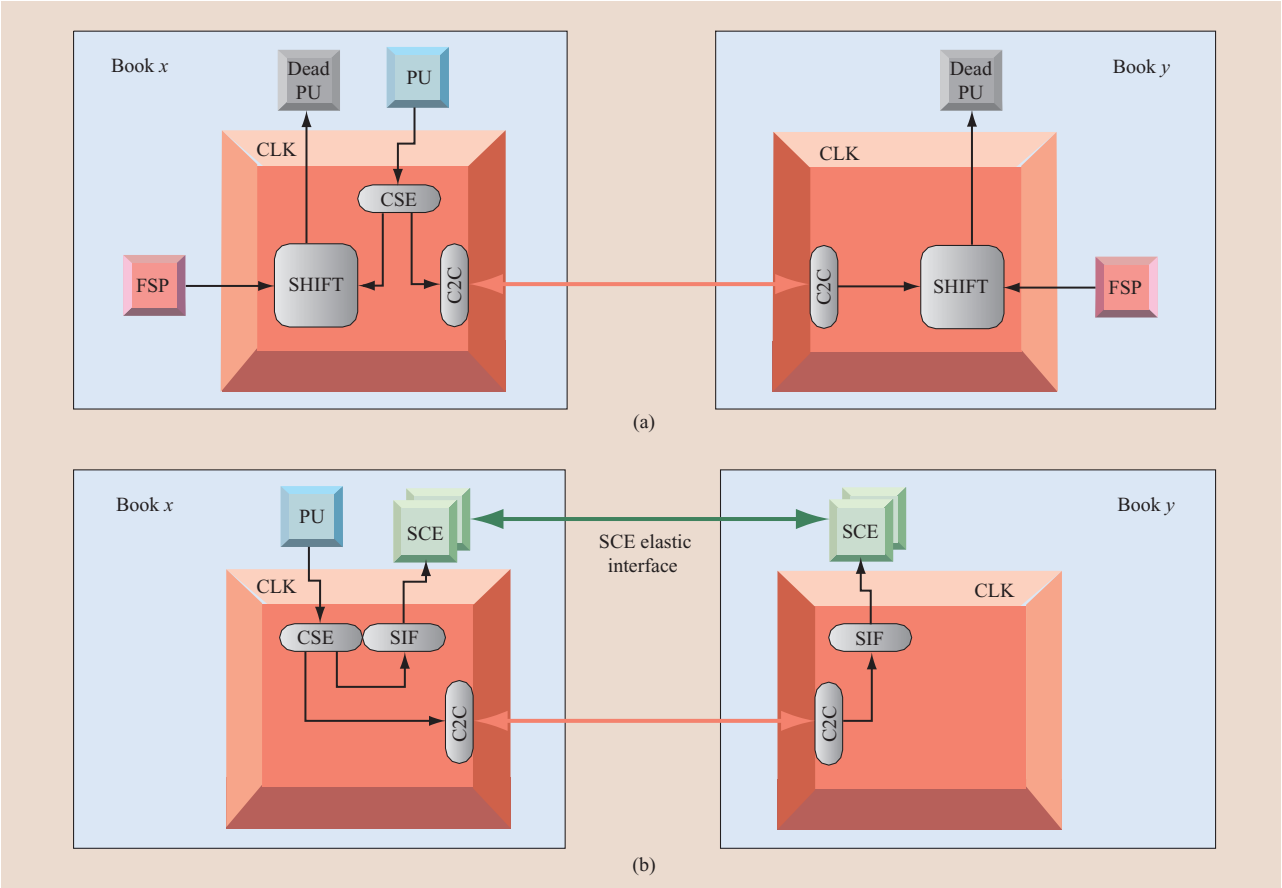

Figure 6

Run-control functions executed by a PU: (a) PU sparing operation. (b) L2 ring unfencing sequence. (C2C: clock to clock; CLK: clock chip; CSE: clock support element; FSP: flexible service processor; PU: processing unit; SCE: service control element; SIF: serial interface.)

SCE ring calibration and unfencing sequence

The SCE provides a shared L2 cache for use by all processors. It is logically split up into two chips, the SCE data chip (SCD) and the SCE control chip (SCC). One book consists of one SCC and four SCDs (see Figure 2).

The ring interface between the L2 chips of different books requires a calibration and unfencing sequence (**Figure 7**). The ring interface cannot be used prior to this calibration sequence. Thus, a PU can only communicate directly with an L2 on the same book. The clock chip provides a communication path to other books. A PU on book *x* starts the sequence by sending commands to the clock chips on books *x* and *y*. The clock chips propagate these commands to the L2 chips, which initiate the calibration procedure.

Synchronous start and stop

Each book performs the reset function and the self-test independently of the other books while under control of the local FSP. Nevertheless, there is the requirement to start the entire system synchronously (start clocks on all chips on every book at the exact same cycle). Because of the high total number of chips in the entire system, it is not possible to control all chips with a single-clock chip. In a system with only one book and one FSP controlling it, all of its chips start at the same time by default. With multibooks, each clock chip receives its start command independently of the other books. Thus, if no provision is made to synchronize the start commands, the entire system starts asynchronously. Because the software is incapable of sending the start command to all books in the same processor cycle, the synchronization must be done in hardware.

There are point-to-point nets among all clock chips (with the same star structure as above) that indicate a local start condition of the clock chip. A local start does not yet start the clocks to the system, but only sends the *armed* indication to the other clock chips. Each clock

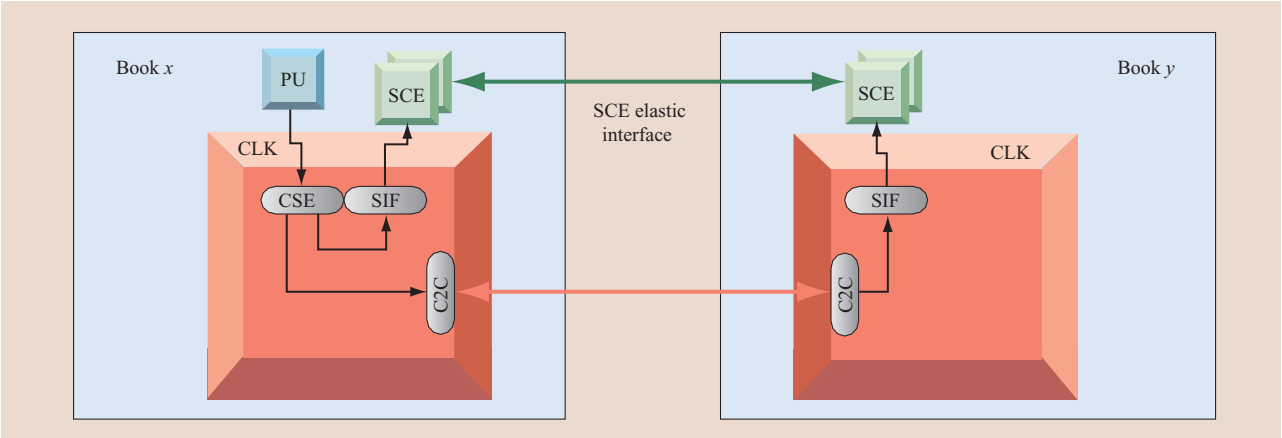

Figure 7

L2 ring unfencing sequence. (C2C: clock to clock; CLK: clock chip; CSE: clock support element; PU: processing unit; SCE: system control element; SIF: serial interface.)

chip receives the armed state indication from all other clock chips in the system and then starts the local clocks only if it is itself started and all armed signals are active. The armed signals from the other clock chips have a defined delay, so the internal start is delayed by the same number of cycles. The last clock chip that indicates a local start starts the clocks to the entire system. All chips in the entire system start in the same cycle. The connection uses the same physical structure as the clock-to-clock interface, and there is no dedicated master—it is a pure peer-to-peer connection. To support systems that do not have the maximum number of books installed or to be able to start a particular book independently of the others (for debugging purposes), there is an option bit on each clock chip to disregard a specific armed input from another clock chip.

Checking hierarchy

Each chip on a book can send a checkstop request to the clock chip on that book. Depending on the source of the checkstop request, the clock chip can take different actions:

- *System checkstop:* Stops all clocks on every chip on every book.
- *Book checkstop:* Stops all clocks on all PUs and memory bus adapter (MBA) chips on a specific book.
- *Chip checkstop:* Stops clocks on a specific PU or MBA chip on a specific book.

System checkstop

Any error condition originating from an L2 chip, a memory storage controller (MSC) chip, or the external time reference (ETR) chip causes a system checkstop. The clock chip drops the local clocks to the connected chips and resets the armed state to the other books. The remote books drop their clock lines. The action on the local book is delayed, and thus the entire system stops synchronously. The local clock chip reports an `ANY_CHIP_CHECK` condition to its FSP.

Book checkstop

A book checkstop stops all PU chips and MBA chips on that book, while the L2 chips, the MSC chips, and the ETR remain active. Thus, remote books can access the memory on the local book via the ring structure between the L2 chips.

A book checkstop request originates from the L2 chip via a special check line. The clock chip drops the clocks to all PU and MBA chips. The clock chip resets the operational bits for all PU and MBA chips. The local clock chip reports an `ANY_CHIP_CHECK` condition to its FSP.

Chip checkstop

Any check originating from a PU core or an MBA chip causes a chip checkstop. This is possible because neither the PU nor the MBA have unique data; there is always a copy in the L2. The clock chip drops the start line to the chip, while the other chips on the same book and the other books remain unchanged. The local clock chip reports an `ANY_CHIP_CHECK` condition to its FSP. A PU-sparing operation can take place at this time.

There are option bits available in the system that allow a chip checkstop to escalate to a book checkstop, and to escalate a book checkstop to a system checkstop. In

addition to that, it is possible to disregard a system checkstop request from another book.

Malfunction alert handling

Once the clock chip has detected a chip checkstop condition, it informs the other nodes in the system (which are kept running) about the event. (A book checkstop is equivalent to a multiple-chip checkstop.) The process of malfunction alert handling starts. The handling is done in two stages: local malfunction alert detection and global malfunction alert detection.

Local malfunction alert detection

The run-control logic generates a local malfunction alert if it receives a check from any PU core or MBA chip. The run-control logic sends this malfunction alert condition to all clock-to-clock interfaces and to the book operational register for the local book. There are four book operational registers on every clock chip: three for the state of the remote books and one for the state of the local book. Normally, the registers for the local book are a shadow of the operational registers (the operational register is responsible for stopping chips on a local book). If the `SUPPRESS MALFUNCTION ALERT` bit is set, the malfunction alert condition is not sent to the clock-to-clock interfaces, and the update of the book operational register for the local book is suspended. In this case, the operational register on book *x* and the book *x* operational register may contain different values.

Global malfunction alert detection

Each clock chip maintains book operational registers for the entire system. The book operational registers are updated by the clock-to-clock interfaces for remote books and are a shadow of the operational register on the local book. The update from remote books may be suspended by disabling the receive part of the clock-to-clock interface for the remote book. The update from the local book may be suspended by setting the `SUPPRESS MALFUNCTION ALERT` bit.

Concluding remarks

This paper describes the necessary hardware extensions that enabled us to migrate an existing central clock chip from the previous single-book generation of zSeries computers to the new multibook computers. Because of physical limitations, it is not possible to use one larger central clock chip per system. A new hardware communication layer has been introduced between the clock chips.

This new hardware layer allows the following functions across books:

- Processor unit (PU) sparing.
- System control element (SCE) ring calibration and unfenced sequence.
- Synchronous start and stop.
- Checking recognition and handling.
- Malfunction alert processing.

With the addition of these features, most of the service code from previous machines can be reused, and time-critical code can be run inside the book.

*Trademark or registered trademark of International Business Machines Corporation.

References

1. F. Baitinger, H. Elfering, G. Kreissig, D. Metz, J. Saalmueller, and F. Scholz, "System Control Structure of the IBM eServer z900," *IBM J. Res. & Dev.* **46,** No. 4/5, 523–535 (July/September 2002).
2. M. Mueller, L. C. Alves, W. Fischer, M. L. Fair, and I. Modi, "RAS Strategy for IBM S/390 G5 and G6," *IBM J. Res. & Dev.* **43,** No. 5/6, 875–888 (September/November 1999).
3. S. Koerner and S. M. Licker, "Run-Control and Service Element Code Simulation for the S/390 Microprocessor," *IBM J. Res. & Dev.* **41,** No. 4/5, 577–580 (July/September 1997).
4. G. Doettling, K. J. Getzlaff, B. Leppla, W. Lipponer, T. Pflueger, T. Schlipf, D. Schmunkamp, and U. Wille, "S/390 Parallel Enterprise Server Generation 3: A Balanced System and Cache Structure," *IBM J. Res. & Dev.* **41,** No. 4/5, 405–428 (July/September 1997).
5. C. Axnix, E. Engler, S. Hegewald, T. Hesmer, M. Kuenzel, and F. M. Welter, "z990 NetMessage-Protocol-Based Processor to Support Element Communication Interface," *IBM J. Res. & Dev.* **48,** No. 3/4, 435-447 (May/July 2004, this issue).

Received September 22, 2003; accepted for publication November 24, 2003; Internet publication April 27, 2004

Tobias Webel *IBM Server Group, IBM Deutschland Entwicklung GmbH, Schoenaicherstrasse 220, 71032 Boeblingen, Germany (webel@de.ibm.com).* Mr. Webel received his B.S. and M.S. degrees in electrical engineering from the University of Stuttgart in 1991 and 1995, respectively. He joined IBM in 1995 to work on logic design for advanced system initialization and system run-control structures starting with the IBM S/390* G5 system. He is one of the leading design engineers for the transition of the system run-control structure to a multibook system including reliability, availability, and serviceability (RAS) aspects, such as concurrent maintenance support. Mr. Webel is currently responsible for the clock chip design of the IBM zSeries G8 system.

Thomas E. Gilbert *IBM Systems and Technology Group, 2455 South Road, Poughkeepsie, New York 12601 (tegilber@us.ibm.com).* Mr. Gilbert joined IBM in 1974 and has held many technical and management positions. He is currently a Senior Engineer in S/390 hardware design verification working on the CMOS clock chip and system integration of the S/390 systems. His previous verification experiences include leading a team and verification of areas on various S/390 systems and CMOS channels. He has been working in design verification since 1984. Mr. Gilbert received an IBM Outstanding Innovation Award for his work on I/O subsystem drivers, an IBM Outstanding Technical Achievement Award for his work on scan ring modeling, and a Team Award for this work on the S/390 G3 common chip verification. He recently received IBM Outstanding Innovation Awards for his ABIST and LBIST verification on S/390 systems.

Dietmar Schmunkamp *IBM Server Group, IBM Deutschland Entwicklung GmbH, Schoenaicherstrasse 220, 71032 Boeblingen, Germany (schmunkamp@de.ibm.com).* Mr. Schmunkamp studied electrical engineering at the Technische Hochschule Darmstadt and received his graduate degree in 1984, joining IBM that same year. He has been working on the design of CMOS microprocessors since that time, with special interests in clocking and service interface. He is a member of the Association for Electrical, Electronic, and Information Technologies (VDE) and holds two patents in the area of clocking and RAS. Mr. Schmunkamp is currently working as a team leader on future run-control designs.