RAS strategy for IBM S/390 G5 and G6

by M. Mueller
L. C. Alves
W. Fischer
M. L. Fair
I. Modi

The Reliability/Availability/Serviceability (RAS) strategy for S/390® G5 and G6 is to continue the S/390 objective of providing Continuous Reliable Operation (CRO). The RAS strategy is constructed with a set of building blocks which work closely together: error prevention, error detection, error recovery, problem determination, service structure, change management, and RAS measurement and analysis. The interdependency among the building blocks is such that removing or weakening any of them limits the ability of the design to achieve the overall CRO objective. Each building block must be fully implemented and must execute flawlessly within itself and together with the other blocks.

Introduction

Customers with mission-critical applications are increasingly relying on their systems to provide continuous reliable operation (CRO). This term is relatively new and is used primarily within RAS organizations; however, it is becoming more widely accepted by the entire engineering community. Simply stated, CRO requires a system to run without interruption while delivering error-free results.

Two basic elements must work together to achieve this objective. The first element is continuous operation, which indicates a system capable of running the customer's operation without stopping because of an error condition, for maintenance, or for system change activity. This must be achieved in conjunction with, not at the expense of, the second basic element, reliable operation. Reliable operation means that system results are error-free; that is, the integrity of all of the data is ensured. Since a running system can rarely distinguish between degrees of data criticality, all of the data must be protected. It is not sufficient for a server to run continuously by sacrificing error detection (ED) capability. For example, a server could run without stopping simply by not implementing a robust ED design. In this case, however, even though the server may continue to run, the accuracy of its results and the impact on the customer's business are unknown. On the other hand, it is equally unacceptable to achieve full data integrity while interrupting the system with each detected error. In this case, the customer could rely on the accuracy of the results, but would suffer from unpredictable availability. Neither of these conditions is tolerable.

S/390* has established a comprehensive RAS strategy which addresses all of the factors contributing to server availability and consistent reliable output. The S/390 team looks for innovative ways to extend the RAS capabilities of each product to new, superior levels.

The G5 and G6 servers deliver a complete RAS strategy, highlighted by key enhancements in fault-tolerant design. The strategy is built upon a set of fundamental RAS building blocks (**Figure 1**) that jointly provide the structure to achieve CRO.

Error prevention, the first step to high availability, starts with an understanding of the technology, the error categories, and error causes. Error prevention minimizes the errors that occur in the field. This is accomplished by ensuring a high-quality product design, using reliable components in the product, and implementing an effective manufacturing test process. Error detection is fundamental to ensuring the integrity of data. Errors must be detected

0018-8646/99/$5.00 © 1999 IBM

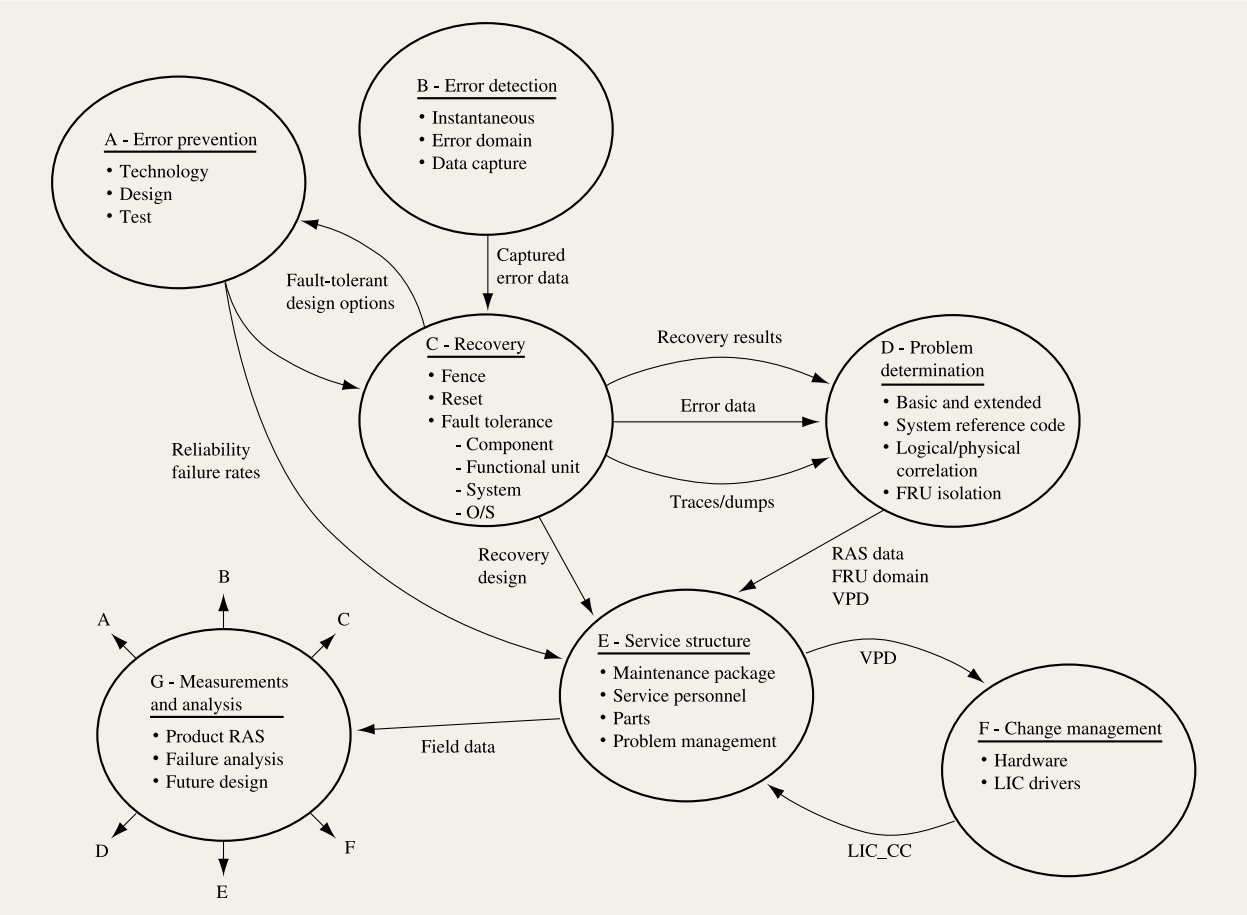

Figure 1

RAS building blocks for G5 and G6.

at the time of failure, contained, and isolated to the smallest possible entity to make recovery reasonable and to enable accurate field-replaceable unit (FRU) isolation.

The G5 and G6 servers deliver an exceptional error recovery design. Innovative as well as established fault-tolerant design methods are employed to minimize the impact of errors on the customer's application and the server's performance.

The problem determination (PD) function is responsible for rapidly analyzing server error conditions, pinpointing the cause of the error, calling for service, and automatically updating system status files.

The service structure is an around-the-clock, around-the-world process that responds quickly and effectively to address the customer's need for assistance, repair, or server growth. Change management applies to hardware and code changes required for added capacity and problem prevention.

A comprehensive RAS measurement and analysis system is in place to measure current field experience versus expected results, to develop corrective actions, and to influence future design enhancements.

Error prevention

Error prevention is the effort to reduce or eliminate completely the number of errors and defects which could occur in the field. This effort begins while the server is still in the concept stage and continues through the design, development, and manufacturing phases. This is a critical step toward CRO because it reduces the number of high-severity server-impact events, reduces the complexity of recovery design and the number of recovery events which must be handled, and reduces the need for service intervention and parts replacement.

Error prevention (**Figure 2**) is accomplished by ensuring a high-quality product design, using reliable

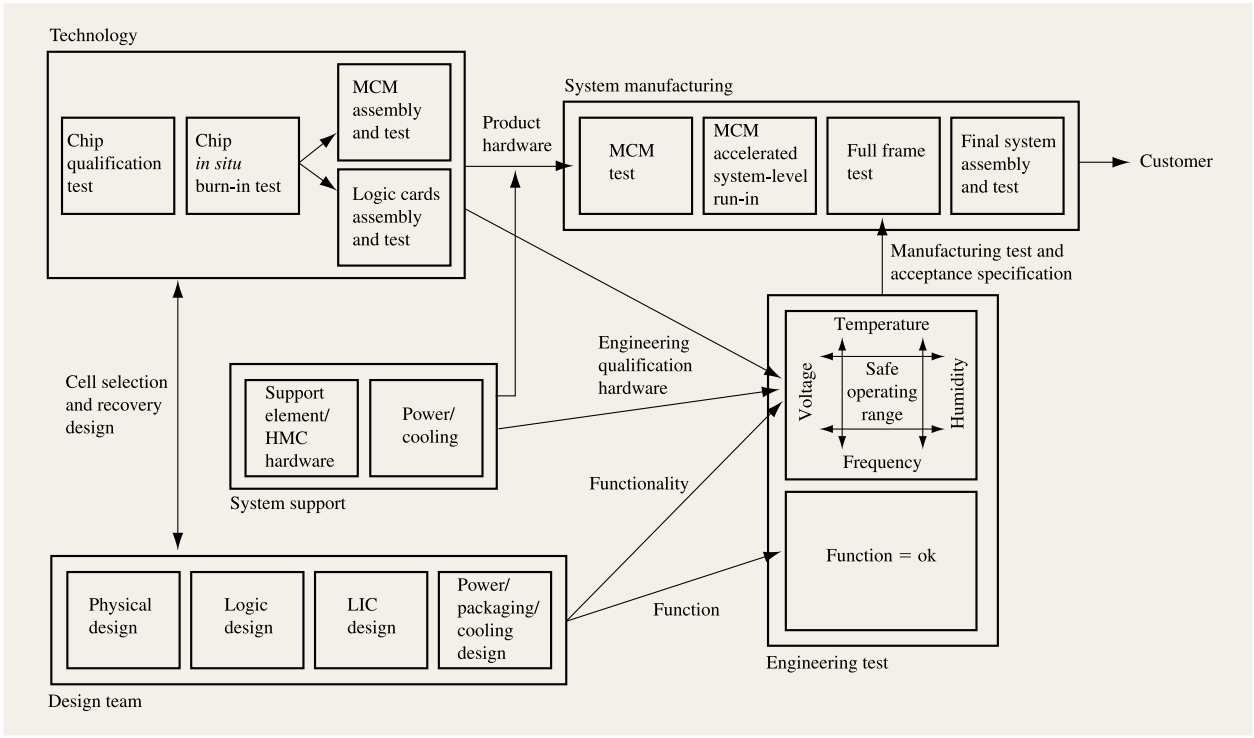

Figure 2

Prevention of errors and defects.

components, and implementing an effective manufacturing test process.

- *Product design quality*

Product design quality is a measure of the degree to which the product is delivered to the customer free of design defects. The concept of design quality applies to hardware, microcode, and technology design. The hardware and microcode design follow rigorous design rules and comprehensive design "walkthroughs." In addition, a structured and systematic simulation is performed at the unit, functional package, and server levels using a combined microcode and hardware simulator.

Many technology-related contributors to intermittent errors are avoided by imposing rules on logical and physical design. For example, technology rules limit the noise produced by simultaneous switching of off-chip drivers or by capacitive coupling. Server clock-cycle time is chosen such that the electronic circuits remain within their specified functional limits and perform flawlessly even under worst-case conditions.

Extensive testing is done by the engineering test organization to validate that the hardware and microcode functions are working according to the design specifications and that the product is operating properly under the specified operating environment. Early product usage by IBM internal locations further reduces the number of defects. In fact, very few defects actually escape to the field, and the majority of them can be corrected concurrently with the regular operation of the customer's application.

- *Reliability of technology*

Industry-standard component reliability is usually quoted in terms of hard errors and seldom includes soft errors, since the latter are still somewhat difficult to predict or measure. However, the failure-rate contribution of soft errors is becoming even more significant than that of hard errors and must be addressed.

Hard errors are persistent physical failures and appear as opens or shorts anywhere on the chips, modules, cards, or boards. Their probability of occurrence (failure rate) is a function of time. They are caused by a physical transformation at the defective location and are related to stress factors in the components.

Soft errors, on the other hand, are caused by electrical events without permanent damage. They are random, one-time events and are destructive to data, but not to the

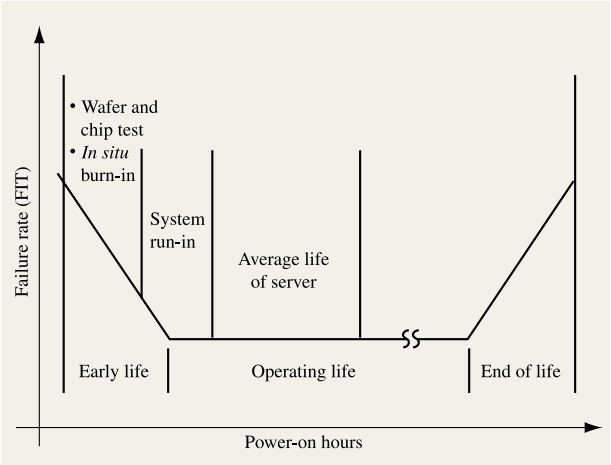

Figure 3

Typical failure-rate curve.

components. They appear as undesired state changes in latches or array-cell contents. Because of the exposure to data integrity problems, soft errors must be addressed by any server which has access to, and thus responsibility for, the customer's critical data.

The reliability of G5 and G6 server technology deals directly and thoroughly with both hard and soft errors.

Hard errors

Hard errors are physical defects that are always reproducible under a certain failure condition. The probability of error varies as a function of time in three phases (**Figure 3**) and is usually stated in terms of FIT (failure in time; 1 FIT = 1 part per million per thousand power-on hours):

- Early life: These failures decrease rapidly with time. Failures in this phase are generally attributable to randomly distributed weaknesses in materials, components, or production processes.
- Operating life: These failures have an approximately constant failure rate. Failures in this phase are Poisson-distributed.
- End of life: These failures increase with time. Failures in this phase are generally attributable to aging, wear-out, fatigue, etc., usually associated with mechanical devices.

In the G5 and G6, three major processes are implemented to achieve high reliability at the server level: reliability qualification, *in situ* burn-in, and system-level run-in.

Reliability qualification consists of component-, unit-, and system-level qualification. The component-level qualification is performed to verify the technology failure rate versus the projected value, to validate the actual failure mechanisms versus the predicted failure mechanisms, and to determine the value of the variables in the voltage and temperature acceleration equations. This qualification is performed by stressing hundreds of chips with temperature, voltage, humidity, and thermal cycling for an extended period of time. Every chip that fails during the component qualification process is analyzed in order to understand the root cause of the failure. Once the root cause is understood, corrective actions are put in place to remove the failure mechanism from the chip manufacturing process.

After qualifying the individual components, the assembled units, such as cards, multichip modules (MCMs), and power supplies, are qualified. This unit-level qualification is performed in a system environment by stressing the units with voltage, temperature, frequency, and humidity, as defined in the engineering test and acceptance (T & A) specification. As in the component-level qualification, all of the failing units are analyzed for root cause, and corrective actions are implemented.

The system-level qualification test is performed by placing complete systems under rigorous stress. This stress includes voltage, temperature, humidity, and cycle-time variations to ensure that the design provides a sufficient operational "guard band."

The G5 and G6 servers use high-reliability components in all critical functional areas. These components have approximately ten times better reliability than off-the-shelf industry-standard components. For IBM chip technology, this is achieved by the *in situ* burn-in process, which is done at high voltage and temperature to accelerate early-life failures. During this burn-in, circuits are exercised with different test patterns at the inputs. Outputs are monitored for correct results and proper voltage levels. For example, each G6 processing unit (PU) chip undergoes *in situ* burn-in on a temporary chip attachment for 48 hours at a 140°C nominal junction temperature and 1.5 times the nominal voltage while exercising millions of test patterns.

The system run-in stress is a process to accelerate early system-level failures. The run-in stress is performed to capture defects that are dependent on server cycle time and not captured during the burn-in process. All MCMs are system-stressed at a 90°C nominal junction temperature, 14% voltage bias, and at server cycle times with stressful customer-like workloads to capture most of the remaining early-life failures. Voltage- and temperature-acceleration equations [1] are used to calculate the acceleration factor, which converts run-in hours into equivalent field power-on hours (EFPOH). For

G6, the EFPOH time exceeds 9000 hours, putting the failure rate well into the flat portion of the Weibull hazard rate ("bathtub") curve (Figure 3).

Soft errors

The most likely source of soft errors is radioactive particles that hit the chip. Alpha and cosmic particles are the major contributors. Alpha hits are caused by radioactive decay of substances in the immediate vicinity of the circuits. Cosmic hits are caused by individual particles within a shower of particles initially triggered by a cosmic ray event in the high atmosphere [2]. Particle hits are external events and are random in location and time. Their error mechanism is a high number of electron–hole pairs generated in the hit area, eventually causing the electronic circuit to change its state. Critical charge (Q_{crit}) is a measure of the charge required to change the state of a cell; it is calculated from the physical layout of latches or array cells. The lower the Q_{crit}, the more susceptible the cell is to a state change. The increased density of the CMOS technology enhances server performance and provides the opportunity for higher levels of integration. However, it also reduces Q_{crit}, making newer technologies more susceptible to soft errors.

Alpha particles have a low reach and low energy, and thus have a very low probability of causing more than one latch to flip. Cosmic particles may be strong enough to flip more than one latch. However, the probability for a double-bit fault is many times less than for a single-bit fault. One identified source of alpha particles is the lead solder balls used to attach the chip to the substrate. For certain G5 and G6 chips, a low-alpha-emission lead is used to reduce the soft-error rate of the cells by a factor of 10.

Careful design consideration is given to the selection of logic and array cells. This is done to minimize exposure to potential errors, and is an iterative joint effort between the logic and physical designers and the technology group. During the G5 design phase, close inspection of latch design revealed some latch types with low Q_{crit} and thus a high soft-error rate. These were replaced by existing latch types with a higher Q_{crit}.

Redesign of array cells for a lower soft-error rate is not an option for large arrays, because it usually means using up more space per array cell. The cell size is typically chosen for highest density. The design of G5 and G6 has implemented recovery in these arrays, and it is very effective against soft errors, since these errors are temporary and usually affect only a single bit.

- *Effectiveness of the manufacturing test processes*

The manufacturing test processes, from chip to module to server, are staged with a focus on removing technology defects as early as possible in each process. At the technology level, there are chip and substrate tests, followed by the *in situ* burn-in test described earlier in this section. At the MCM assembly area, where up to 29 CMOS chips are mounted on the substrate, the tests are executed prior to and after encapsulation. The tests consist of an open/short interconnect test, the chip logic built-in self-test (LBIST), and the array built-in self-test (ABIST).

In system manufacturing, the module is initially placed in the MCM test, which is a minimum system-level test designed to remove any defects which are detectable only in a system environment. Once the module is defect-free, it is placed in the system run-in test to accelerate any early-life reliability failures, as described earlier. The next-level test is the frame test, using the maximum configuration [MCM, logic cards, power and cooling, support element (SE), and hardware management console (HMC)] and testing in accordance with the manufacturing T & A specification, which defines all of the tests and test durations required for each product. Final system assembly and test performs a set of tests, prior to shipment, on servers configured to match the customer order.

After defects have been minimized by the robustness of the design, the reliability of the technology, and an effective manufacturing test process, a certain probability of error still remains. Such errors must be detected in order to protect the integrity of the data. In addition, sufficient error data must be captured to enable effective recovery.

Error detection

Error detection is the capability of determining that a functional unit is not performing its required function. Instantaneous detection is the detection of an error prior to committing results to any other functional unit. The S/390 strategy is to protect data integrity by implementing a comprehensive instantaneous error-detection design.

Without instantaneous error detection, an error may go undetected. In this case, the result of the operation is incorrect, and data is corrupted or the instruction flow is changed. The eventual discovery of the error and any impact on the customer's operation are both unknown. Discovering the error and capturing the error information are prerequisite to determining the amount of damage to the current operation, recovering from the error, and locating the origin of the error.

All error-detection mechanisms use redundancy to make errors visible. The particular method is chosen on the basis of its capabilities and the needs of the function to be protected. Full instantaneous error detection in the dataflow and control flow protects the ongoing operation. The dataflow includes data buses, registers, buffers, storage and cache, and arithmetic logic units. The control flow includes command/status buses and finite-state

machines. While error detection is more complex within the control flow, it is absolutely required to preserve the integrity of the functional unit.

Error-correcting codes (ECCs) are used on most arrays, such as L2 cache and main memory, and on buses, such as the server memory bus. Parity is used to protect data and control paths and to protect arrays such as the L1 cache, which is a store-through cache with a copy of the data available in the L2 cache. Cyclic redundancy codes (CRCs) are used to protect Licensed Internal Code (LIC) modules. Dual execution with compare is used in the G5 and G6 PUs. Operation-graph-based event monitoring is used in the memory bus adapter (MBA) chips. The state of each individual state machine in the MBA is fed into a macro state machine which is used to check the MBA operations [3].

These methods are designed for instantaneous error detection to ensure data integrity and support error recovery. When an error is detected, erroneous results must be prevented from contaminating valuable recovery information. This is achieved by error fencing, in which the error is confined to a noncritical domain. For example, the comparator for the dual instruction/execution units (I/E units) in the PU detects any mismatch and prevents any erroneous results from spreading into the checkpoint array.

In timing-critical logic areas, performance requirements may prevent the implementation of error checking in the same cycle as that in which the error occurs. In these cases, errors may be propagated in a controlled manner, and the error is detected a few cycles later. However, the later the detection, the larger the error domain will become. Large error domains increase the difficulty of fault isolation and recovery.

Error-data capture is ensured for both "clock-running" and "clock-stopped" error scenarios. "Clock-running" means that the server or the functional element continues to function through the error. "Clock-stopped" means that the error is so severe that the server or element is no longer functioning. For the clock-running case, the error data is saved to memory before recovery is attempted. For the clock-stopped case, the error data is frozen and is scanned out by the SE.

The error data that is captured at the time the error occurs is provided to the error-recovery building block. Error-recovery and problem-determination requirements dictate the amount of error data necessary to minimize the effect of the error and ensure effective problem determination.

Error recovery

Error recovery is the capability of a functional unit to tolerate faults by minimizing the impact on the application and on server performance. Error recovery is invoked when an error is detected and associated error data is captured at the time of failure. Recovery is implemented by error correction using ECC, or by nullification of the effect of the error and restarting from a known, previously saved state. Error recovery is successful when the error is either corrected or does not recur during retry. Analog units, such as the G5 and G6 servers' power/thermal subsystem, use load balancing when one out of $N + 1$ identical units fails.

Error recovery uses the data captured by error detection to isolate the source of the error and to determine the effect of the error on the operation. Careful logic design contains the effect of an error of the smallest possible impact. Faults causing high impact require recovery as close to the error source as possible.

Recoverable events are recorded and tolerated up to a specific threshold. When the threshold is exceeded, the faulty unit is "fenced" (logically removed from the configuration) in order to preserve high error-detection capability and maintain performance.

When error recovery is not successful, the failed unit is fenced from the configuration. In certain cases in which degradation has occurred, such as an unrecoverable central processor (CP) error, the degradation is reported to the operating system as malfunction alert, together with state information, so that the task can be redispatched on another CP. Error correction is applied to arrays, such as store-in caches (L2) and S/390 customer storage or expanded storage, which contain persistent data. Error correction is also applied to data buses and command/status buses used for system-related operations. Correction of a single-line failure is required to continue operation until a deferred repair can be performed. ECC with single-bit correction and double-bit detection capability is normally used.

The major fault-tolerant design enhancements in G5 and G6 are in the area of element sparing. These enhancements include transparent sparing for PUs, dynamic random-access memory (DRAM), and L1/L2 cache lines. The G5 system structure contains PUs, cache, memory, MBAs, and cryptographic coprocessors.

- *Transparent CP/ICF sparing*

The G5 and G6 servers have implemented full transparent sparing for PUs. This enhancement over the G4 server [4] enables the hardware to activate a spare PU to replace a failed PU with no involvement from the operating system or the customer, while preserving the application that was running at the time of the error.

The G6 processor subsystem (**Figure 4**) contains up to 14 identical PU chips (up to 12 in G5), each containing a common microcode load. During the initial microcode load (IML), the system configuration assigns the function that each one will perform: CP, system-assist processor

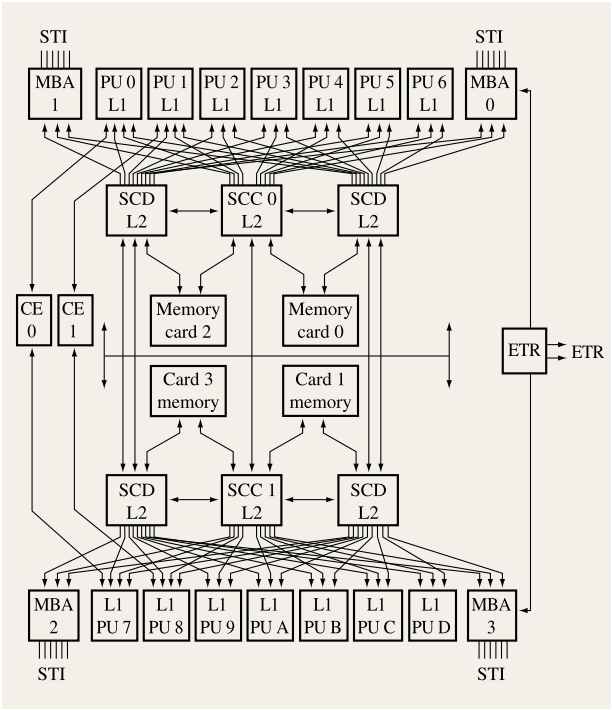

Figure 4

G6 system structure.

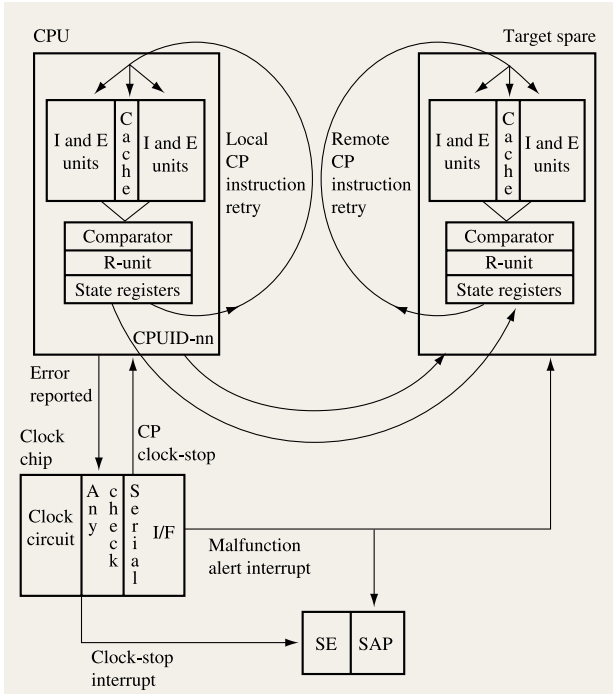

Figure 5

Transparent CP sparing.

(SAP), integrated coupling facility (ICF), or spare PU. The common hardware and microcode allow a spare PU to replace any failing PU.

The PU consists of two completely duplicated I/E units, a Level 1 cache, and a register unit (R-unit) (**Figure 5**). The R-unit contains the compare circuitry and the ECC-protected checkpoint arrays containing all of the critical architectural facilities, including register contents and instruction address. At the completion of every instruction, the results produced by the two I/E units are compared and, if equal, the results of the instruction are checkpointed for recovery in case the next instruction fails.

If the results differ, an error trigger is set and instruction-retry recovery is attempted. As part of the PU reset process, the R-unit instructs the store buffer to release any stores of committed instructions and purge any stores of the failing instruction. The R-unit then raises a fence signal to communicate to the L2 that the PU is undergoing recovery. At this point all latches and arrays can be reset, since the PU is fenced. The R-unit refresh controller reads the contents of the checkpoint array and sends that data to be loaded into the respective copies of the I/E units and the cache unit. The instruction is then retried. The results are compared again, and if the cause of the initial error was a soft or intermittent failure, the checkpoint array will be error-free.

The PU fencing can be removed and instruction processing can be restarted by retrying the original failing instruction. If the failure is permanent, the local instruction retry will fail. If the retry threshold is exceeded, the PU will remain fenced, and the next level of recovery will be attempted. An error indication is raised to the clock chip, and the PU is placed in the clock-stopped state and marked as "disabled" in the clock chip configuration registers. In addition, the clock chip sends an interrupt to all other PUs and to the SE. This interrupt causes the SE to scan out the R-unit of the clock-stopped PU and send this data to the SAP to be used later for the instruction retry recovery on a spare PU. The remaining "healthy" PUs (including the spare) also honor the interrupt and, through a selection algorithm, the "target spare" is identified.

Once the replacement for the clock-stopped PU is identified, the physical/logical ID fields in the configuration area are updated. The SAP passes the R-unit contents to the target spare, and the microcode issues a Load R-unit instruction, which begins a "self-initiated brain transplant." At the completion of the Load R-unit instruction, the

Table 1 Comparison of PU sparing between G4 and G5/G6.

PU function	*Mode*	*G4*		*G5/G6*	
		Application preserved	*Spare*	*Application preserved*	*Spare*
CP					
Uniprocessor model	Basic/native	No	Re-IML	Yes	Transparent
	LPAR	No	Reactivate	Yes	Transparent
MP model	Basic/native	Yes	Concurrent	Yes	Transparent
	LPAR-uni-dedicated	No	Reactivate	Yes	Transparent
	LPAR-uni-shared	Yes	Transparent	Yes	Transparent
	LPAR-MP-dedicated	Yes	Concurrent	Yes	Transparent
	LPAR-MP-shared	Yes	Transparent	Yes	Transparent
ICF					
	LPAR-uni-dedicated	No	Reactivate	Yes	Transparent
	LPAR-MP-dedicated	Yes	Concurrent	Yes	Transparent
SAP					
	N/A	Yes	Transparent	Yes	Transparent

spare has the identity of the clock-stopped PU and begins executing the same instruction at the point where the failed PU left off [5].

In the G4, most CP unit sparing is concurrent with the customer's operation and, with the processor availability feature (PAF), the application running at the time of the failure is recovered by another CP. However, under certain configurations and conditions, this is not possible. The implementation of transparent CP sparing in G5 and G6 addressed these limitations (**Table 1**).

- *Cryptographic coprocessor and CP sparing*

Each cryptographic coprocessor in the G5 and G6 servers features a primary path to a PU and an alternate path to a second PU (Figure 4). Only one path is active at a given time. The two PUs associated with the alternate path from the cryptographic coprocessor are the last to be configured as CPs, SAPs, or ICFs. This increases the likelihood that these PUs will be available as spares. Normally, each cryptographic coprocessor is configured to the primary CP. In case the primary CP fails, the spare PU with the alternate path replaces the primary CP transparently, maintaining the cryptographic coprocessor function.

- *Memory DRAM and cache-line sparing*

The memory cards in G5 and G6 servers are designed in such a way that each DRAM module contributes only one bit to a given checking block. This allows ECC to correct all single-bit errors, all partial module failures, and all complete module failures.

A second error in the same checking block, detected during readout, is an uncorrectable error. If such an error occurs, the current instruction is terminated and the operating system is informed by a machine-check interrupt, which reports the instruction-processing damage caused by an uncorrectable storage error, together with the failing storage address. This allows the operating system, in nearly all cases, to limit the impact to a single user or application. The storage allocated for the affected application is released. The operating system clears and tests the storage page with a special instruction which is part of the S/390 instruction set to test the usability of the page. The page may be reused when the test does not detect any failures; otherwise, it is removed from the usable page-frame pool to avoid further application impact.

The G5 and G6 servers avoid the accumulation of soft errors in seldom-accessed storage by continuously "scrubbing" the complete storage to correct single-bit errors. Scrubbing uses the error syndrome to count the errors in each DRAM module. When the count of errors exceeds a specified threshold, based on the DRAM technology, a spare DRAM module is activated. Exceeding the threshold indicates that the module may contain multiple cell failures, a bit-line failure, or a total module failure.

DRAM sparing copies the contents of the faulty module into the spare module. Any store operation stores the data bit in both DRAMs. When copying is completed successfully, the faulty module is replaced by the spare module. The replacement cannot be done when any checking block affected by the faulty module indicates an uncorrectable error, because the error syndrome cannot be used to locate the faulty bits. The error counts of all DRAM modules are accumulated and logged to be transmitted to IBM with the next service data upload. Problem determination is informed about the usage of any spare module. The self-repair using a spare DRAM avoids

downtime for memory-card replacement due to DRAM failure. Each G5/G6 memory card is shipped with four spare DRAMs. The probability that a memory card will have to be replaced because of DRAM failures during the lifetime of the server is extremely low.

The G5 and G6 servers use ECC to protect the L2 cache data and directory entries. In addition, G5 and G6 have implemented special hardware to allow failing cache lines to be logically removed from the cache while the server is running or during IML. This causes an unmeasurable performance degradation, while preventing an exposure to uncorrectable errors when an additional hard or soft error is introduced into the same checking block in which a single-cell failure already exists. Where the data in L2 has been changed, an uncorrectable error would lead to application impact as soon as this data is used by the application, in the same way as already described for uncorrectable storage errors. The ECC and line-delete features of the cache allow correction of infrequent soft errors and prevent application impact, even in the case of a permanent single-bit failure.

The G5 and G6 servers implement a cache-line relocation mechanism to self-repair the cache by using a spare cache line to replace the one containing the failure. Without this capability, the built-in logic and array self-test, which is executed when the servers are powered on, would detect the single-bit failure in the cache array and mark the whole cache data chip as faulty, causing degradation of the server to half of the cache size.

Array-word-line relocation with one-time programmable fuses is a well-known repair method used in chip manufacturing to increase the array chip yield. The address decoder of the array is personalized to replace failing word lines with spare word lines. In G5 and G6, this method is expanded to allow the support element to replace the failing cache-array word line with a spare word line during the array self-test.

The L1 cache is a parity-checked store-through cache with refresh capability (capability to obtain a valid copy of the data from the L2 cache). The L1 cache has an implementation for cache-line delete and sparing that is similar to that of the L2. In addition, the L1 cache has the capability of deleting a quarter cache while the server is running, so that the processing unit can continue to run with a minor performance degradation.

- *Support element sparing*

Every G6 server includes a standard second support element (optional on G5) which serves as a backup for the primary SE. The alternate SE is a mirrored copy of the primary SE. Its function is continuously checked. In case of a malfunction of the primary SE, a switch on the front panel of the server transfers control to the alternate. Either SE can be maintained concurrently with server operation.

The error-recovery block interacts with error prevention to achieve the optimal match between technology selection and recovery capability. Recovery results are provided to the problem-determination function in the form of threshold status, information on fenced or degraded units, and the availability of dumps and traces. The recovery design influences the service structure's parts-stocking and concurrent maintenance plans.

Problem determination

Basic problem determination (PD) is an automated analysis performed on the captured error data and recovery results to determine the root cause of the problem. The objective is to isolate the problem to a single FRU. Extended PD is a manual analysis performed by product engineering (PE) and development engineering (DE) on traces or dumps to troubleshoot very difficult problems.

Instantaneous error detection enables basic problem determination to isolate the failing unit immediately after the recovery attempt is complete. The result of the PD analysis is translated into a service call. Problem determination in the G5 and G6 servers does not rely on reproducing the fault by using diagnostic tests initiated by the operator nor on operation monitoring running in the background. Problem-determination methods requiring physical removal of cards to isolate the failing unit are used only as a last resort.

Problem determination is a distributed function. It is performed in the various subsystems such as power/thermal, central processor complex (CPC), and SE. The functions in the CPC are further split into PU subsystem, I/O subsystem, and channel subsystems for the various channel types. Problem determination is done within each individual subsystem, where the detailed knowledge of the function exists. All problem-determination functions report their individual results in an error log. Each error log consists of a unique system reference code (SRC) describing the nature of the problem, an extension code describing the logical location of the suspected components, a status byte indicating recovery results, and a detailed log containing the captured error data. The system reference code within the error log is used to search a failure information table, residing on the SE, to retrieve the repair information. The SRC description contains the list of recommended FRUs, their part numbers, resolution probabilities, and verification routines. When necessary, the SRC description also contains recommended manual isolation routines. The individual error logs are analyzed and correlated by a central function in the SE. The central function also translates the logical units into FRUs and physical

locations identifying the frame, cage, and cage slot in the server.

In G5 and G6, the majority of failures are quickly isolated to a single FRU. For example, an SRC indicating a storage controller failure will be resolved to the specific failing memory card.

Other failure scenarios are more complex and require additional analysis. For example, when the power subsystem reports an over-voltage condition, followed by a logic element in the same power domain reporting a logic error, it is very likely that the root cause is power. In this case the logic error is secondary. Another example is the generation of multiple logs as a result of a single failure, such as in a multidrop bus, configured as a data traffic concentrator with several fan-out devices. With a concentrator failure, the multiple logs contain one common element (the concentrator). The central PD function targets the FRU containing the concentrator for service.

In some cases the result of the PD analysis may implicate multiple FRUs. For example, for buses in which the receiver detects a parity error in its receive register but the sender does not detect a parity error in its corresponding transmit register, it is not possible to determine the root cause of the failure. It could be the sender's off-chip driver, the connecting cable or board wiring, or the receiver. In this case PD produces a list of potential units to be replaced, weighted by estimated failure rates and probabilities based on experience in these types of failures.

Problem determination keeps track of the status of redundant elements and determines whether service is required. Service is required when an observable degradation occurs. The degradation can be reduced performance, reduction of a safe margin for error detection, or increased likelihood of a server outage or loss of function. The PD function issues a call for service when repair or customer assistance is required; it also ensures that the vital product data (VPD) is updated to reflect current server hardware status.

Extended PD analysis is seldom necessary, since basic problem determination is extremely efficient in identifying the root cause of the problem. However, for certain types of failures, extended PD is required. For example, microcode errors normally show up when a very rare state combination occurs. Detailed analysis is required to understand the root cause. When problem determination isolates the error to a microcode defect instead of a hardware fault, it collects a variety of dumps of internal storage areas, ranging from CP, SAP, and logically partitioned mode (LPAR) microcode to the operating system. The dumps also contain the buffers of continuously running traces and state tracking information. The particular data collected is dependent on the type of code error detected and must be sufficient for the root cause to be identified and understood on the basis of this single occurrence. The microcode designers use highly automated tools to analyze the massive amount of data collected.

The logic built-in self-test (LBIST) and array built-in self-test (ABIST) are performed during power-on of the G5 and G6 servers to test the logic and array chips. Problem determination fences any failing chip before the server becomes operational, or configures alternate paths in order to avoid any foreseeable application impact. In most cases, there is little or no degradation; the customer can continue his operation and, if a repair is necessary, schedule the repair for a convenient time.

Problem determination fences the smallest possible number of logical units, allowing the rest of the server to continue operation. A catastrophic fault, causing a server outage or loss of function such as sysplex external timer reference (ETR) attachment, normally requires immediate repair. The G5 and G6 servers provide an emergency operation for most of these rare cases. Problem determination performs an automatic reconfiguration allowing the server to function in a degraded mode until the customer engineer (CE) and spare part arrive. Customers can predefine activation profiles containing their critical workloads and activate only these key partitions. This allows all remaining resources to be dedicated to these applications. Problem determination provides the service structure with information regarding the need for service and the status of the server following failure events. When repair is required, the correct FRU or FRU list is identified. If customer assistance is needed, the nature of the recovery action (and potential customer involvement) is indicated via the problem analysis panels based on the SRC and system status information. Service and manufacturing are kept current as to the system configuration by means of updates to and transmission of the VPD data file.

Service structure

The G5/G6 service structure is the collection of people, parts, programs, databases, and communications committed to respond quickly when service is required to correct a problem, and to restore the server to full capability with minimum impact on system operation and minimum customer involvement. This structure applies to the entire IBM system, including the server, the I/O, and the system software. The focus of this section is on the server, but I/O and software support are always available as required to assist in resolving any system problem. This service is available 24 hours per day year-round, beginning at installation, continuing through the one-year warranty period, which is a full warranty for all parts and labor, and

continuing after warranty, if the customer chooses, under the terms of the IBM Maintenance Agreement.

Once the problem-determination function has concluded that service is required, a call is made to request that service. This automated call function is set up by the CE, with the customer's approval, at installation time, and is generally defined to be fully automated and enabled 24 hours per day. The call is made via the modem on the HMC to the IBM Remote Technical Assistance Information Network (RETAIN) system. Data sent to RETAIN on the call includes the server identification and description, error logs, FEDC information, and whether hardware replacement is necessary. If so, the part numbers are included with the call information. Certain calls may indicate that a recovery action has occurred and that resulting conditions require customer involvement. In this case, the remote technical support center (RTSC) will call the customer to offer any assistance necessary.

Application code running in the RETAIN system packages this information into a problem-management record, forwards it to the RTSC, and logs it for problem tracking, resolution, and history.

The RTSC is the first point of contact for the call for service from the G5 and G6 servers. Their initial role is to perform call screening in order to quickly evaluate call severity, server availability status, and the need to dispatch people and parts. This recent enhancement adds a valuable focal point to the structure, speeds the response to the customer, and optimizes utilization of field service skills. The role of RTSC is to evaluate the call, contact the customer immediately, resolve the problem if possible, or initiate the dispatching of a CE and the necessary parts to the customer site.

For cases in which a failure has caused minimal or no impact to G5/G6 performance, the customer may choose to schedule the service action for a more convenient time. This is also handled by the RTSC, arranging for the CE and the part(s) to be on-site at the appointed time.

An on-line screen-driven maintenance package, called Repair and Verify, guides the CE through the service call, including locating and removing the failed part and installing the new part. In cases where multiple FRUs are implicated by PD, the repair action starts with the replacement of the first FRU in the FRU list. Automatic verification procedures or diagnostics associated with the FRU are used to verify the success of each repair step.

FRUs are designed for single-person handling and with minimum requirement for unique tools. Clear labeling of server locations and parts and positive location indicators are used to assist in inserting and removing parts.

The design of G5 and G6 includes concurrent repair of the hardware and microcode. This enables faulty hardware or microcode to be replaced while the server is up and running. There is no customer involvement, other than approval for the action, and no impact on running applications. As the S/390 servers evolved from bipolar to CMOS technology, parts integration brought with it a challenge to provide concurrent maintenance. That challenge has been met with inventive solutions such as the transparent sparing of processing units within the MCM. Field data shows that more than 80% of the repairs are performed concurrently. The concurrently maintainable hardware and microcode on the G5 and G6 are shown in **Table 2**.

Table 2 G5/G6 concurrent repair.

Hardware	*Microcode*
Processing units, via transparent sparing	CP
Channels—ESCON*, parallel, coupling links	IOP/i390
ESCON converter, FICON*	LPAR
OSAs	Channel
Power supplies	Coupling links
Cooling units	OSAs
AC inputs	Power
Battery backup	Cooling
Support element	Support element
Hardware management console	Hardware management console

A technical council, with representatives from the three major microcode development locations, product engineering, and engineering system test, selects the microcode problems with the highest impact or most pervasive nature to be corrected immediately. Special focus is placed on the implementation of the repair to be concurrently applied.

S/390 provides a comprehensive parts-stocking system to ensure fast access to all G5 and G6 server parts. The system is worldwide, and is managed on the basis of the criticality of each part to server operation, the expected replacement rate of those parts, and geography (location of server types, models, and features). The system also ensures that removed parts are returned to IBM for hardware failure analysis (see the section on measurement for a discussion) and that parts are replenished at the stocking location as required.

If the CE requires additional support during a service call, the G5/G6 service structure provides three additional levels of support. First is the RTSC, staffed by experienced field personnel and supported with access to problem-management-and-resolution databases. In particular, the RTSC has access to the G5/G6 knowledge-based system for rapid access to reference code definitions, information on previously encountered problems, and recommended action plans. The next level

of support is product engineering (PE), an expert staff specialized in specific processor subsystem design and service capabilities, and experienced in rapid problem resolution. The final level is development engineering (DE), with representatives from the team that designed and tested the server.

All of these support levels are available 24 hours per day and have worldwide access to the error data, logs, and traces that are captured and transmitted at the time of the problem, as well as to history data and problem repair or resolution data. They also have, subject to agreement by the customer, the capability to dial in to the failed server to perform remote diagnostics. This overall structure ensures that the right level of expertise can be quickly brought to bear as required on any customer problem.

The service structure's problem tracking, mentioned briefly above, is managed through the RETAIN facility and is centered on the problem management record. This function is on line and available 24 hours per day worldwide. A particular problem management hardware (PMH) record, for example, is created with the automated call for service from the G5/G6. It contains the customer identification, the server description, and the type and severity of the problem. As the problem proceeds toward resolution, the PMH is updated by the CE, the RTSC, and, when involved, PE and DE. Questions, instructions, action plans, and results concerning the service action are recorded here, creating a history log for the problem. The problem remains open until the resolution is complete, at which time it is closed.

Thus, the service for G5 and G6 is a comprehensive structure comprising server design, data capture, automatic calling, highly skilled levels of support, a responsive parts-stocking system, and thorough problem management.

The service structure is dependent on the problem-determination, error-recovery, and error-prevention building blocks for details on design, parts-replacement strategy, and error information. It provides data to change management in the form of VPD, and to RAS measurement and analysis as field performance data and error data.

Change management

Change management is the capability to introduce changes by the customer to his G5 and G6 servers. The objective is that this be nondisruptive. Both hardware and code changes must be addressed by this function. When a customer requests a hardware configuration upgrade, manufacturing must determine the current server configuration. By examining the machine's VPD, manufacturing knows exactly what is installed: for example, channel and I/O adapter cards, number of processing units on the MCM, and the amount of memory, and thus the available I/O, PU, and memory growth capacity. This status information is updated on the machine's VPD file each time a change occurs, and is returned to IBM primarily for change management. Where possible, the requested upgrade is completed without requiring hardware change.

In a significant enhancement for availability, G5 and G6 servers can be upgraded with additional CPs and ICFs concurrently. There is no longer a need for a scheduled outage to activate the new capacity. New S/390 architecture supports this capability. System configuration files and VPD are updated at the same time, providing accurate model and PU status to the customer, IBM, and other vendors. To maximize the customer's opportunity to upgrade, all G6 servers are provided with a 14-PU MCM and with two SAPs as standard.

As a further enhancement, customers with the capacity backup (CBU) feature may now activate that function concurrently. For example, CBU provides emergency additional capacity to compensate for a disaster which disrupts a portion of the customer's computing power.

The G5 and G6 servers can be maintained at the latest microcode level to provide the customer with the most current set of problem corrections and with the newest functions. As described earlier, most microcode repairs are designed for concurrent installation and activation. Major microcode releases (drivers), which contain not only the latest corrections but new function as well, require a re-IML to be activated.

RAS measurement and analysis

RAS measurement is the process by which G5 and G6 field RAS performance is assessed and the performance of the RAS building blocks is evaluated. RAS analysis is the investigation of failures to determine root cause and set action plans. The results of the work are a key input to every one of the other building blocks.

In order to achieve CRO, it is imperative to provide the status of the current product and processes, to highlight corrective actions required to meet quality objectives, and to highlight opportunities for future design enhancements.

As implemented by G5 and G6, field tracking is enabled by the extensive error detection, error data capture, and data reporting described above. In addition, data flows back to IBM by means of a scheduled operation defined by the customer called transmit system availability data (TSAD). This data describes server events and recovery actions taken since the last TSAD, whether or not there were any service actions.

Field measurement data is reviewed daily by PE, DE, and RAS to ensure that the recovery and service processes are working as designed and, if necessary, address problems requiring immediate action. The executive management team reviews these results every week,

or as the case demands. Summary results are available on an internal website for fast access by the S/390 team.

The S/390 primary RAS measurement is server average mean time between failures (MTBF) for a given product family. This is a measure of all of the critical unscheduled incidents during the accumulated lifetime of the product family. For S/390 CMOS servers, the measurement shows a binodal distribution of critical failures, in which more than 95% of the servers have no such failures, and thus no unscheduled outages. The automated problem analysis and comprehensive service structure minimize the duration of outages that do occur.

The parts-return process ensures that parts which are replaced in the field are returned to IBM for failure analysis. Each returned part contains failure information, the SRC, the SRC extension, repair information, and the problem number on a SEEPROM, a serial electrically erasable programmable memory module mounted on the part. The failure analysis activity involves manufacturing, technology, and development engineers, who analyze the failure down to its root cause and, if necessary, generate immediate corrective action.

Conclusions

The G5 and G6 servers continue the S/390 history of delivering superior RAS performance to the highly demanding traditional server marketplace, as well as to the new applications market featuring e-business, network computing, and server consolidation. To achieve this, each of the RAS building blocks—error prevention, error detection, error recovery, problem determination, service structure, change management, and RAS measurement and analysis—contains the required RAS functions and strives to execute in a flawless manner within itself and together with the other blocks. G5/G6 recovery design enhancements, such as transparent sparing for CPs, cache-line relocate, and cryptographic coprocessor alternate path, strengthen an already solid RAS implementation as G5 and G6 continue the drive to CRO. By integrating this robust server into a Parallel Sysplex*, continuous reliable operation is fully realized.

*Trademark or registered trademark of International Business Machines Corporation.

References

1. M. L. Kerbaugh, *CMOS Quality Report—Year End 1998*, IBM Microelectronics Division, Essex Junction, VT 05452, February 24, 1999.
2. T. J. O'Gorman, J. M. Ross, A. H. Taber, J. F. Ziegler, H. P. Muhlfeld, C. J. Montrose, H. W. Curtis, and J. L. Walsh, "Field Testing for Cosmic Ray Soft Errors in Semiconductor Memories," *IBM J. Res. Develop.* **40,** No. 1, 41–50 (1996).
3. T. Buechner, R. Fritz, P. Guenther, M. Helms, K. D. Lamb, M. Loew, T. Schlipf, and M. H. Walz, "Event Monitoring in Highly Complex Hardware Systems," *IBM J. Res. Develop.* **43**, No. 5/6, (1999, this issue).
4. L. F. Spainhower and T. A. Gregg, "G4: A Fault-Tolerant CMOS Mainframe," presented at the Twenty-Eighth Annual International Symposium on Fault-Tolerant Computing, Munich, Germany, June 23–25, 1998.
5. M. A. Check and T. J. Slegel, "Custom S/390 G5 and G6 Microprocessors," *IBM J. Res. Develop.* **43**, No. 5/6, (1999, this issue).

Received November 17, 1998; accepted for publication May 20, 1999

Michael Mueller *IBM System/390 Division, Schoenaicherstrasse 220, 71032 Boeblingen, Germany (mulm@de.ibm.com).* Mr. Mueller is a Senior Engineer working in the S/390 Licensed Internal Code Design group. He studied electrical engineering at the University of Stuttgart and received his Dipl. Ing. degree in 1985. He joined IBM in 1985, working in the S/370 Product Assurance Test Laboratory in Boeblingen. He has held various positions in microcode development and system design, and is currently responsible for coordinating RAS microcode and hardware interface development in Boeblingen.

Luiz C. Alves *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (alves@us.ibm.com).* Mr. Alves is a Senior Engineer working in the S/390 System Design group. He graduated from New York University in 1975 with a B.S. degree in electrical engineering and received his M.S. degree in electrical engineering in 1977 from the Polytechnic Institute of New York. He joined IBM in 1977, working in the Advanced System Manufacturing Engineering organization, where he held various technical and managerial positions. In 1985 Mr. Alves was named Manager of 3090 Field Quality Assurance; in 1987 he became the RAS manager for the 9021 processor families. He is currently responsible for defining the RAS requirements for future products.

Wolfgang Fischer *IBM System/390 Division, Schoenaicherstrasse 220, 71032 Boeblingen, Germany (fischerw@de.ibm.com).* Mr. Fischer studied electrical engineering at the University of Siegen and received his master's degree (Dipl. Ing.) in 1985. He joined IBM in January 1986 in the Hardware Development Department for S/370 in Boeblingen. Mr. Fischer has held various positions in hardware and microcode development; he is currently in microcode development and has RAS responsibilities.

Myron L. Fair *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (mlfair@us.ibm.com).* Mr. Fair is an Advisory Engineer in the S/390 Systems RAS group. He graduated from the University of Illinois in 1967 with a B.S. degree in mathematics. He joined IBM that same year in the Systems Development Division, where he held various technical positions related to field and development RAS, and in Special Contracts technical assessment. In 1980, he transferred to the Group Staff Headquarters and became Manager of Product RAS Analysis in 1984. Mr. Fair is currently the team leader for RAS requirements and objectives for S/390 servers.

Indravadan (Dan) Modi *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (modidan@us.ibm.com).* Mr. Modi is an Advisory Engineer working in the S/390 Custom Microprocessor Design group. He graduated from Gujarat University, India, in 1966 with a B.S. degree in electrical engineering and received his M.S. degree in electrical engineering in 1968 from Utah State University, Logan, Utah. He joined IBM in Pougheepsie, New York, that same year, working in the Power System Design area, where he held various technical positions. Mr. Modi moved to IBM East Fishkill in 1982 as a manager of an electrical analysis group which was responsible for substrate electrical characteristics along with delta-I and coupled-noise analysis. He returned to Poughkeepsie in 1986 to join a system assurance group, where he was responsible for system technology assurance. In 1990, he joined the RAS group, where he was responsible for technology reliability, SPQL, and development of system test and acceptance specifications. In 1998 he moved to the Custom Microprocessor Design group, where he is currently responsible for chip/MCM reliability, SPQL, and the system test and acceptance specifications for all S/390 products.