The S/390 G5/G6 binodal cache

by P. R. Turgeon
P. Mak
M. A. Blake
M. F. Fee
C. B. Ford III
P. J. Meaney
R. Seigler
W. W. Shen

The IBM S/390® fifth-generation CMOS-based server (more commonly known as the G5) produced a dramatic improvement in system-level performance in comparison with its predecessor, the G4. Much of this improvement can be attributed to an innovative approach to the cache and memory hierarchy: the binodal cache architecture. This design features shared caching and very high sustainable bandwidths at all points in the system. It contains several innovations in managing shared data, in maintaining high bandwidths at critical points in the system, and in sustaining high performance with unparalleled fault tolerance and recovery capabilities. This paper addresses several of these key features as they are implemented in the S/390 G5 server and its successor, the S/390 G6 server.

Introduction

The IBM S/390* G5 server represents a significant breakthrough in terms of high-end mainframe performance and throughput. The single most important contributor to this dramatic performance improvement is the architecture and design of the binodal cache. Although it builds upon key internal cache management concepts first developed for the G4 [1], the G5 binodal cache represents a major step forward in system organization and symmetrical multiprocessor (SMP) design. The cache and memory hierarchy presented here features large, high-frequency, 256KB unified private L1 caches, an 8MB binodal, shared L2 cache, a nonblocking crosspoint data switch, and a novel system controller managing all coherency and data transfer operations. These features, when combined with the significantly more powerful G5 microprocessor [2–4], have resulted in the unprecedented system-level performance improvements demonstrated by the G5 server.

The primary design objectives behind the creation of the binodal cache architecture were to minimize access latencies at all stages of the cache hierarchy and to control critical system resource utilization. Experience gained with the S/390 G3 and G4 servers has taught that S/390 SMP performance is particularly sensitive to system bus utilization and the queuing which can ensue if this utilization reaches unsustainable levels. An optimal design approach, as demonstrated in the bipolar ES/9021, involves the use of a large, fully shared L2 cache structure. Such a design offers the benefit of optimized cache latencies and efficiencies, but is not easily implementable in today's CMOS technologies.

The binodal cache architecture is an alternative approach, offering more modularity and compatibility with CMOS design. This architecture features a split-transaction bus design in which address and command information is managed on separate, dedicated buses. This approach eliminates multiplexing operations on the performance-critical data bus. Further, the crosspoint switch elements are now embedded into the system controller function. The number of cache/switch nodes in the system has been optimized at two, as compared to 12 in the S/390 G3 and four in the S/390 G4. The total number of chips comprising the system/bus controls and L2 cache functions has been reduced as well—from 20

0018-8646/99/$5.00 © 1999 IBM

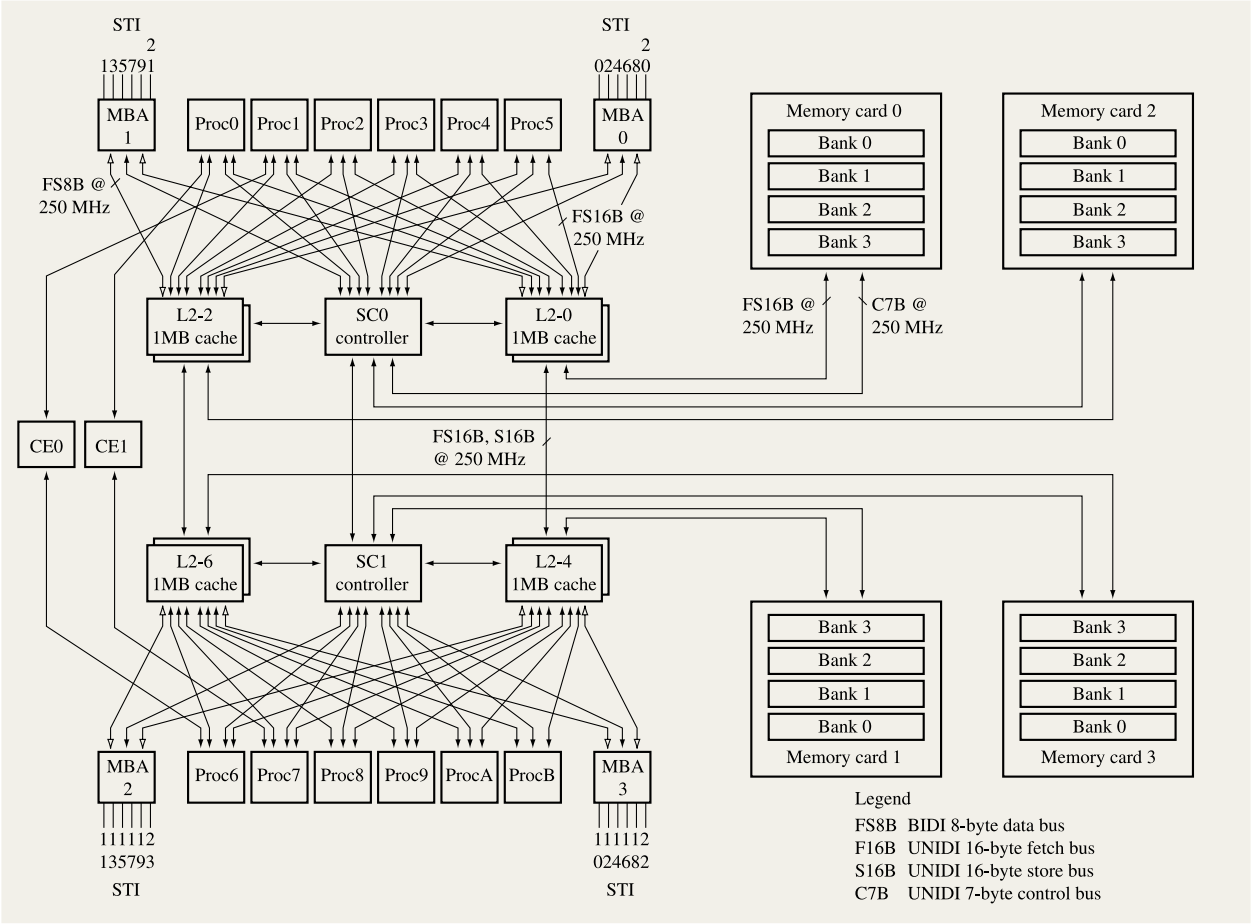

Figure 1

G5 system structure.

(G3) to 16 (G4) to 10 (G5)—a 50% reduction in CMOS components in only two years. Measurements on memory-intensive S/390 workloads have indicated that the design changes introduced in the G5 binodal cache architecture have reduced the cache and memory hierarchy component of the CPI (cycles per instruction, a processor speed metric) from two thirds in G4 to less than half of the total in G5. System bus utilization has also improved significantly. Where average utilizations in the G4 approach the 60% range, average utilizations on similar workloads are less than 30% in the G5.

This paper describes a number of the unique aspects of the binodal cache architecture, with an emphasis on those functions which contributed significantly to either system performance or system functionality. We begin with an overview of the binodal cache itself, then describe the internal controls used in shared cache and shared resource management. The next section discusses bus management and control, and is followed by brief sections describing hardware assist capabilities and main memory considerations. The unique RAS characteristics of the G5 are described next. The paper concludes with a brief description of the binodal extensions implemented in the S/390 G6 as well as possible future enhancements.

Binodal cache structure

The G5 binodal cache architecture is depicted in **Figure 1**. The key components of this SMP architecture are the L1 cache (embedded in the G5 microprocessor chip), the L2 cache, the system controller (SC), the memory, and the processor, memory, and internode buses. The L1 cache is described in [2]. The L2 cache chip, of which there are eight in a fully configured system, is a 17.36-mm × 17.43-mm CMOS 6S2 chip containing the peripherally laid out 1MB cache array (**Figure 2**). This chip contains slightly more than 59 million transistors, making it one of the densest

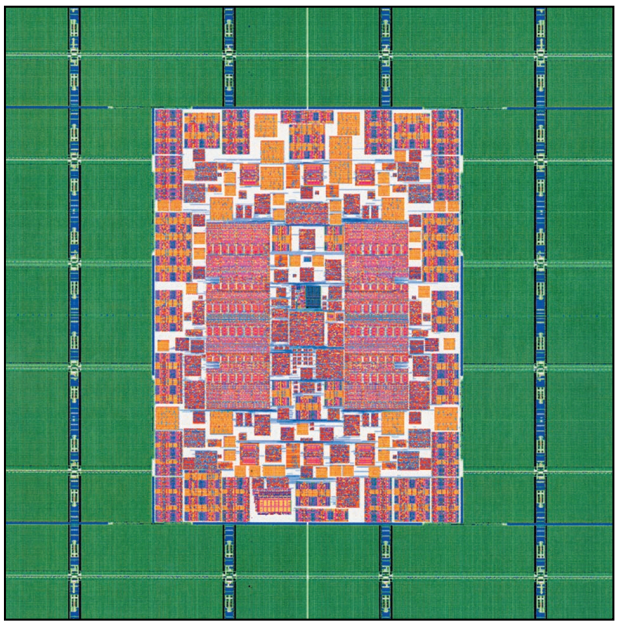

Figure 2

L2 cache chip.

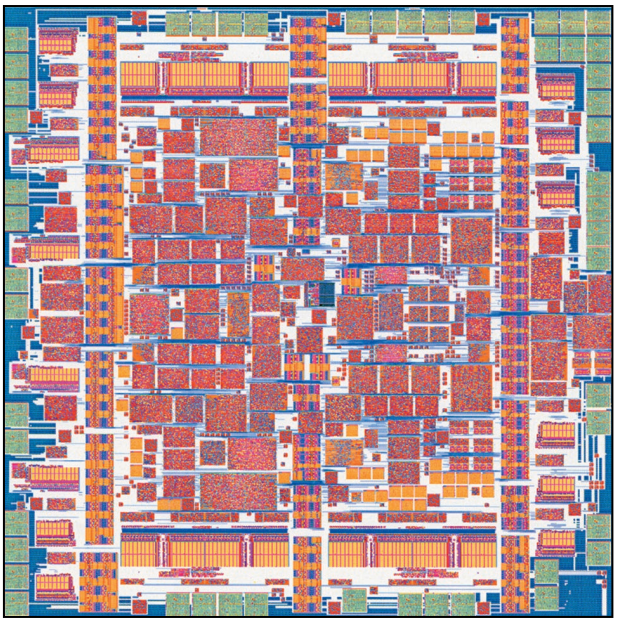

Figure 3

System controller chip.

custom CMOS chips manufactured by IBM at the time of its introduction. The 4 MB of L2 cache within each node is eight-way set-associative, with a nonsectored 256-byte cache line size. The SC chip (**Figure 3**) is a 16.45-mm × 16.52-mm CMOS 6S2 chip. It contains the cache directories, cache coherency management, common resource management, and binodal architecture controls, comprising 8.6 million transistors of complex control functions. A two-chip design was chosen to maximize L2 cache capacity while also improving system bandwidth by providing more chip pins at the node [5].

Data buses throughout the binodal cache are 16 bytes wide. The SC and L2 chips, along with all buses including the memory bus, are clocked at half the microprocessor frequency. In the high-end G5 models, all interfaces are operating at 250 MHz. Prototype versions of the G5 have been built and operated in the 300-MHz range (600-MHz microprocessor clock rate).

The cache management in the G5 binodal cache architecture uses and significantly extends a number of cache coherency concepts introduced in prior S/390 servers. As in the G4, storage consistency is managed at the L2 cache level. The private L1 caches, integrated into the microprocessor, employ a store-through approach. The L2 cache, physically spread across four L2 cache chips in each node, employs a store-in architecture. The L1 is

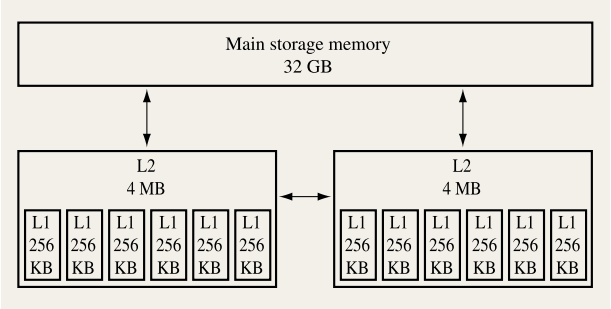

Figure 4

G5 storage hierarchy.

managed with a full subset protocol, which means that all valid lines contained in the L1 are also contained in the L2 cache. This arrangement results in a simpler cache coherency management algorithm. **Figure 4** depicts a logical view of G5 cache management.

The L2 cache management is implemented as a modified MESI (modified/exclusive/shared/invalid) algorithm. Two distinct shared ownership states are maintained in the L2 directory. The first, "locally shared,"

means that the referenced data is shared by multiple microprocessors within the same node, while the second, "globally shared," indicates that data may be shared by microprocessors in each node. This distinction becomes important when a microprocessor requires exclusive ownership of a line already stored in the L2 cache. If "locally shared," the exclusive state will be granted more quickly, since only local L1 copies must be invalidated. A "globally shared" exclusive request cannot be granted until both the local and remote copies have been properly invalidated.

A critical design feature of the binodal cache is the source for data requests which miss the local L2 cache node. Since the memory access latency is significantly higher than that of the remote node, each node in the G5 provides "modified," "shared," and "exclusive" data intervention. This can be contrasted with the G4, which used only a "modified" intervention scheme. Main memory accesses are reserved for those situations in which the data does not reside in either L2 cache node.

Another new feature of the G5 system controller is its dual-pipeline design. As in G4, shared resource utilization within the system controller is serialized via a pipelined architecture [1]. Each G5 cache node now actually contains two distinct pipelines, each with its own cache, directory, main memory, and internode interfaces and controls. The pipelines are interleaved across main memory addresses, with each directly controlling two L2 cache chips. Because of technology constraints, the two pipes are not completely independent. Some operations, such as the cross invalidate (XI) function, are shared by the two pipelines.

Pipeline selection is address-based, not availability-based, thereby avoiding address collisions between the pipelines during coherency checking. An address-based selection mechanism provides an additional advantage by enabling a simple dual-resource layout. While not perfectly mirrored and symmetrical owing to the shared functions, each pipeline effectively occupies half of the SC chip. This simplified chip layout localizes critical control paths to approximately half of the chip area. The chip layout would have been much more difficult to achieve with a single pipeline or with a more complex resource-based selection mechanism.

Another primary objective leading to the dual-pipeline approach was the goal of minimizing the performance impact of bursty data store rates. This is an important consideration, given store-through L1 caches spread across a 12-microprocessor SMP system design. Accordingly, each microprocessor is allowed to queue up to eight data store operations per pipeline within the node while waiting for SC or L2 cache facilities to become available. The dual-pipeline, binodal cache can also sustain a maximum of four store operations to the L2 cache each cycle—this translates to a cache store bandwidth of 16 GB/s when running in a fully configured high-end system.

Bandwidth and performance

A major design focus during the development of the binodal cache was to improve the CPI of the G5 server through attention to average system latencies and system bus utilization. The improved switch fabric in the G5 system design is critical to the attainment of this goal. Eliminating the G3/G4 discrete bus switch layer and integrating the crosspoint switching function into the system controller significantly reduces main memory and XI latencies. Introducing dedicated address/command and response buses reduces data bus overhead, significantly increasing available data bandwidth. Independent buses further improve bandwidth by enabling more concurrent data transfer operations to be active in the system. The end result of these improvements is that queuing is virtually eliminated, while very high sustainable bandwidths are enabled at all points in the interconnection network. Average L1 cache miss penalties have been improved by 33% or more. These factors—larger, more efficient caches with a high-bandwidth switch fabric—are the single greatest reason for the twofold performance improvement demonstrated by the G5 over its predecessor.

The interface between the microprocessors and the L2 caches remains largely unchanged from the prior G4 implementation. Each microprocessor has the ability to send a fetch or store request on each cycle. However, data bandwidth has improved, since each microprocessor now has a 16-byte-wide BIDI data bus to each pipeline in its local system controller and can therefore simultaneously transfer 16 bytes of fetch data and 16 bytes of store data on every bus cycle. To simplify processor design, only one of the two buses can be actively returning fetch data. In addition to the request and data buses, a dedicated set of invalidation buses from the system controller to its local processors are used to independently manage coherency between the L1 and L2 caches. Because the processor-to-L2 data buses are private, the G5 binodal cache provides a total of 192 bytes of processor fetch bandwidth per cycle along with 192 bytes of store capacity. This translates to a peak processor data bandwidth of 96 GB/s in a maximally configured system.

As with the processor/system controller interface, each system controller node [6] can issue a data fetch or store request to the other node on each bus cycle. However, the internode buses are implemented as four 16-byte-wide unidirectional buses, with each node driving one bus per pipeline. Unidirectional buses were used in this application to avoid BIDI switchover penalties, thereby making each internode bus capable of sustaining its peak bandwidth of 16 GB/s, regardless of the direction of

dataflow requests. This was a critical factor in helping to reduce the average cache hierarchy latencies in the system.

The interface between the system controller and memory, like the processor and internode interfaces, is a split-transaction, 16-byte-wide data bus clocked at half the microprocessor frequency. Physical packaging limitations required the use of a BIDI bus. However, because the time of flight of this bus is significantly longer because these nets originate on the MCM, cross the system board, and enter the memory card, a multicycle clocking scheme was adopted [7]. This approach enabled retention of the 2:1 frequency ratio, a critical design consideration. Lower gearing, such as 3:1 or slower, is often employed in these situations to relieve the memory bus. However, this was considered undesirable because of the adverse effects the less-aggressive gearing has on total bus utilization, as it takes proportionately more cycles to transfer the same amount of data. Therefore, the G5 memory interface gains the advantage of very high frequency without unwanted overhead in the form of dead or unused bus cycles. The G5 memory interface supports a peak data bandwidth of 16 GB/s, very nearly the sustainable rate.

The I/O subsystem is not addressed in this discussion of the G5 binodal cache architecture. However, the system-controller-to-MBA (I/O memory bus adapter) interface should be mentioned, since it had somewhat different design considerations. Because the ratio of commands to data is much lower than on the other buses, the SC-to-MBA bus was designed as an eight-byte-wide BIDI, with each MBA commanding two of these buses. To further save signal pins, the command, address, and response buses share the same BIDI. Despite this, and because the system-controller-to-MBA data buses are completely independent of each other and of all other system data buses, the G5 features a peak-I/O-to-memory bandwidth of 16 GB/s.

With all of this available bandwidth, the internal processing capacities of each node in the binodal architecture must be very robust. With 12 microprocessors and four MBAs capable of issuing a total of 28 possible requests to the two system controller nodes, each node must have adequate internal resources, along with a large switch to keep data transfers moving concurrently. The cache associated with each of the four pipelines is four-way interleaved, allowing a 16-byte data access in each interleave on each cycle. This adds up to 256 bytes of L2 cache capacity per cycle, more than enough to satisfy the theoretical 224 bytes per cycle peak required if all possible requesters hit in the L2.

L2 misses can be rather bursty, depending upon the types of transactions being processed. Therefore, the system controller nodes have sufficient resources to process 16 local L2 fetch miss operations and 16 memory store operations simultaneously. Because main memory is organized into 16 independent banks, a combination of 16 fetch and store operations to main memory can be serviced simultaneously. In addition, each system controller node can concurrently service four XI castouts (a total of eight per system).

Hardware-assisted move engine

Another important consideration in the design of the G5 cache and memory hierarchy was improving the performance of the rich S/390 set of data movement operations and instructions. Prior CMOS generations had implemented these functions using microcode and existing system resources. While execution of these instructions was correct and coherent, the prior implementations also commandeered a disproportionate share of critical system resources. The G5 response was to develop a series of hardware assists for these types of functions, the most notable of which is the hardware-assisted move engine.

The hardware-assisted move engine enables movement of blocks of data from one area in main memory to another, with minimal microprocessor and microcode activity, allowing the system to perform other tasks while the data movement is underway. Data blocks may range in size from 256 bytes to 4K bytes in a single operation. The engine makes use of line buffers within the L2 cache chip, using these as a staging area for retrieving data from main memory, then storing to a new location in main memory. The L2 cache itself is not disturbed by the data movement operation unless a portion of the destination address exists in the L2.

The move engine takes advantage of the memory controller function, one of the key controllers which make up the system controller. Because the memory controller initiates and monitors all memory transactions, the availability of each memory bank on each memory card is known. The move engine takes advantage of this by beginning the data movement operation in an available part of the block to be moved, avoiding queuing for memory resources. The resulting data fetches can also be overlapped, again to make optimum use of the memory bus.

The net effect of this dedicated hardware assist is a sevenfold improvement in page-moving efficiency (measured in processor cycles required to successfully complete a 4KB page move) in comparison with predecessor S/390 CMOS systems. The improvement can be attributed to the more efficient use of memory buses, avoidance of cache thrashing, and reduced microprocessor and microcode overhead for performing page move operations.

Memory card

The G5 memory card (**Figure 5**) had to meet an aggressive set of design objectives in order to match the performance of the cache hierarchy, including the following:

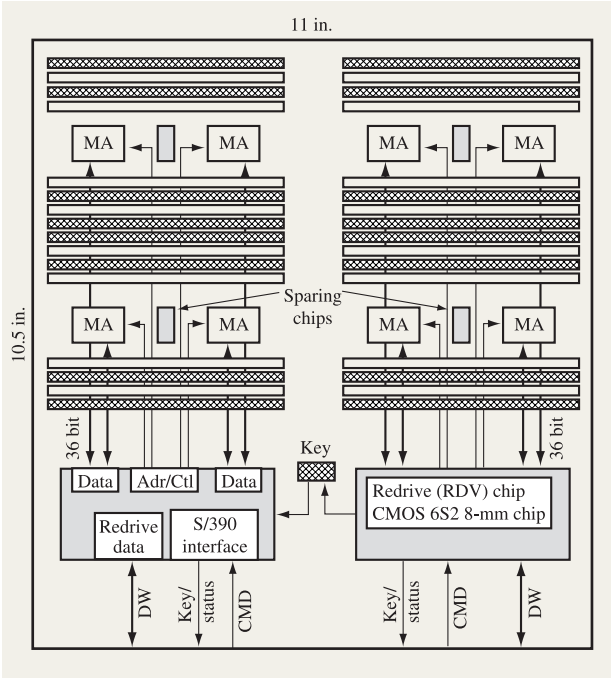

Figure 5

S/390 CMOS G5/G6 memory card.

1. Sustain four independent memory requests simultaneously, one for each address bank on a memory card. As shown in Figure 1, there are four memory cards, satisfying the need for up to 16 simultaneous memory operations in the system.
2. Achieve higher bandwidth and throughput via the split-transaction bus scheme. This approach puts pressure on technology and packaging capabilities.
3. Achieve improved bus efficiencies via tight logical coupling to the system controller.
4. Attain very high memory interface frequency. The design goal was 250 MHz or better at the card interface.
5. Design a robust memory capable of being reused in the next S/390 CMOS system.

Each memory card contains four sets of control/address registers as well as four sets of data buffers. The intent was to closely couple the memory card to the system controller, to attempt to minimize queuing at all points in the cache and memory hierarchy, and to enforce maximum bus utilization efficiencies. The system controller attempts to issue memory requests only to cards/banks which are currently available, and to chain those requests to minimize switching of the BIDI memory bus. The split-transaction buses further improve efficiency, since the data bus is not multiplexed with command or address information. Also, a status/key bus has been implemented for S/390 store protect key data transfers, further reducing data bus overhead.

The memory data bus operates at the same clock rate as the system controller. Individual requests are gapless—an entire line fetch or store will execute in 16 consecutive interface cycles. Back-to-back transfers, such as multiple fetch requests to the same card, can be chained to maximize bus efficiency. The wide potential operating range for this memory card necessitated the inclusion of programmable interface timing. Systems using this card at the slower end of its operating range can operate the interface in a standard one-cycle mode. The very-high-frequency G5 turbo (and G6) application operates the interface in multicycle mode, in which the clock used by the memory card is offset from that used by the system controller by a quarter cycle to relieve internal paths on the card. This adds one cycle to the round-trip latency but allows the interface to operate well in excess of 250 MHz, a very good tradeoff.

RAS and debug characteristics

A characteristic of high-performance cache and memory hierarchy design that is often overlooked is its reliability, availability, and serviceability (RAS). Simply stated, world-class RAS characteristics, such as dynamic error recovery and array fault tolerance, are commonly considered to be difficult, if not impossible, to obtain in a high-frequency environment. Debug aids and circumvention logic are also often considered to be mutually exclusive to a high-performance design. The S/390 G5 system controller makes those views obsolete by providing unparalleled system recovery and data fault tolerance, as well as innovative circumvention and debug aids, in addition to the previously documented system performance.

The primary objectives of the G5 system controller RAS strategy were to provide uncompromised system data integrity at all times, maximize error recoverability without adding undue design complexity, and prevent and minimize occurrences of uncorrectable errors and "hangs." In support of this strategy, the G5 SC chip contains more than 1500 fault detectors, while the L2 cache chip contains more than 250. Activation of any of these detectors results in the reporting of an error occurrence via the SC error-reporting register (ERR) structure, and prevents all other fault detectors from reporting errors until the appropriate recovery action on the currently processing error scenario has been completed.

System-controller-detected errors can result in several different types of recovery actions, largely based upon severity and potential for data corruption if not handled

properly. The recovery scenarios, in increasing order of severity and/or system disruption, are as follows:

1. *Correctable error* (*CE*) recovery. The detected error has already been, or is in the process of being, corrected. Normal system operation continues. Hardware indicators are set to reflect that a CE event has occurred. Correctable errors are most commonly associated with an ECC correction in the L2 cache or directory. In the G5, faults detected in interface commands or in the pipeline are also categorized as CE, because these faults result in immediate, transparent retry upon detection.
2. *Processor retry recovery*. The detected error can be isolated to an operation associated with a known processor. The nature of the error is such that queued store operations previously issued by this processor are not affected. These pending store operations to the L2 cache will be completed, after which the processor will be reset in order to attempt to retry the operation which failed. System operation continues unaffected for all other processors in the system.
3. *Processor instruction processing damage (IPD) recovery*. The detected error can be isolated to an operation associated with a known processor. The nature of the error indicates that previously queued store operations by this processor may be suspect. The offending processor is immediately reset in order to perform IPD recovery. This involves notifying the operating system that the task at hand must be aborted and retried. System operation continues unaffected for all other processors in the system.
4. *Processor checkstop recovery*. The detected error can be isolated to an operation associated with a known processor, but cannot be categorized as a processor retry or processor IPD type of failure. The failing processor is immediately checkstopped and removed from the active system configuration. System operation continues unaffected for all other processors in the system.
5. *System checkstop*. The detected error does not satisfy any of the above recoverable criteria. System data may be compromised if processing is allowed to continue. All system clocks are stopped immediately.

To provide for maximum design flexibility and to allow for the possibility of refining recovery actions defined for a particular type of fault, recovery actions are programmable and maskable. For example, an error associated with a particular recovery class can be escalated to a more severe recovery class via changes to initialization data. An extreme case involves escalating all detected errors to system checkstop, a feature which can prove very useful in early hardware debug.

In terms of fault tolerance, the G5 system controller and L2 cache contain several improvements over the prior CMOS systems. The L2 cache, the L2 directory, the L2 fetch and store buffers, and the remote XI buffers are all ECC-protected. In addition, all of the data buses between the processors and the L2 cache and between the L2 cache and main memory are ECC-protected. Single-bit errors in any of these areas are corrected, on the fly, with no loss in performance. In many cases, even errors which are uncorrectable via the ECC function can be recovered successfully by invoking one of the recovery scenarios described above.

The dataflow fault tolerance strategy involves localizing the source of the error as much as possible to increase the correction success, while also minimizing the chances of accumulating errors into an uncorrectable scenario. Correctable errors are corrected immediately upon detection. ECC correction stations exist at the outputs of the main memory, the L2 cache, and the store stacks, as well as at the internode interface. When a CE is detected in the L2 cache, the location is tagged for purge. This action, under hardware control, invalidates the cache location and, if the data was in "modified" state, corrects and writes the line back into main memory. This asynchronous activity prevents CEs from accumulating in the L2, thereby minimizing the probability for future uncorrectable errors (UE). On the basis of projected soft-fail rates of the CMOS technology used, the performance degradation associated with this premature invalidation of cache lines is negligible.

When an ECC station detects an uncorrectable error, propagation of that data is blocked [8]. For example, a UE detected leaving main memory is not installed in the L2, but rather is returned to the processor and tagged to initiate a memory recovery action. A UE detected in the L2 cache will not be cast out to main memory unless the line is in "modified" state. This avoids overlaying potentially good data in memory. However, if the line is "modified," it is stored to memory as a "special UE," a unique syndrome that identifies the cache, and not main memory, as the original source of the error. This will prevent an unneeded memory error recovery scenario from occurring the next time this location is accessed.

The G5 system controller also contains logic used to proactively prevent certain types of errors and lockout conditions from occurring. Three important examples of these functions are the purge function, the fast hang quiesce controller, and the priority dither function.

The purge function is invoked each time a CE or UE is detected in the L2 cache or its directory. Upon detection of an error, the address, directory compartment, and syndrome are trapped. This information is compared to the information trapped during the previous CE or UE. A perfect match indicates that the array cell in question has

failed before and is likely a hard fail. The entry is purged as described earlier, and the line deleted from active system use. When the current failure is correctable but not identical to the previous failure, the line is simply purged, but the compartment remains available for future use.

The fast hang quiesce controller [9] identifies and resolves common resource lockouts before more drastic recovery scenarios are needed. The system controller uses a global hang-detect signal to determine whether a lockout situation exists. A shorter hang-detect signal is also used to monitor processing progress through the system controller. If a requester for SC resources (a processor, for example) has been ready and queued for two entire short hang periods, but has not yet been processed, all subsequent requesters will be temporarily forced inactive to allow all currently active operations to complete. This action increases the probability that all requesters will complete by reducing contention for shared resources while ensuring that the requester which detected the lockout will itself process through the pipeline unimpeded.

Priority dither is a means by which requesters for the system controller pipeline can be prevented from gaining access to the pipe for some number of cycles. Priority dither reorders the priority arbitration logic to force operations facing lockout to gain access to the pipeline. This minimizes the opportunity for resource contention, breaking the lockout scenarios. The number of cycles and the triggering mechanism of the priority dither function are programmable for maximum flexibility.

The G5 binodal cache also features a logic trace function which is actually a "logic analyzer on the chip." The logic trace is a series of nonintrusive buffers which capture the states of critical signals on each SC cycle. As implemented in the G5, approximately 700 critical signals are trapped in buffers which are 256 entries deep. The trace buffers can be stopped and extracted at any time, providing the design engineers with the most recent 256-cycle snapshot of SC activity. The logic trace function may also be programmed to operate in "compression mode." In this mode, buffer entries are updated only if the signals being traced have changed, providing an opportunity for logic tracing across a wider window of operation. The logic trace function has proven invaluable in analyzing the complex system interactions of the binodal cache.

G6 binodal cache and future extensions

The relatively simple modularity offered by the binodal architecture enabled a significant extension to its functionality by applying a more advanced CMOS technology. Specifically, the G6 server introduces the binodal cache architecture in an IBM "copper technology," CMOS 7S, implementation. The new technology offering provides twice the density of that used in G5, doubling the L2 cache size to 2 MB per L2 chip, for a total of 16 MB per system, while maintaining the same associativity and line size as the G5. The additional level of metallization available in CMOS 7S enables changing the cache organization, doubling the number of interleaves to eight. This doubles the available L2 cache bandwidth to 512 bytes per cycle.

The other significant change introduced by the G6 server is an increase in the SMP size, a trend expected to continue. The internal resources and controls of the system controller were extended to support an additional microprocessor per node, for a total of 14 microprocessors in a fully configured system. The SMP size was more limited by technology constraints than by design—there is evidence that the basic structure of the binodal cache architecture, with appropriate modifications, can support an even larger tightly coupled SMP. The above logical changes, coupled with operating frequency improvements (maximum microprocessor frequency is 637 MHz, maximum cache hierarchy frequency is 318.5 MHz), have led to the 50% system performance improvements measured in the G6, a mere nine months following the introduction of the G5!

The performance and operation of the G5/G6 binodal cache have been extensively measured and analyzed. Despite the overwhelming success of the architecture thus far, significant improvements are still possible without necessitating complete redesign. In addition to the aforementioned potential for SMP size and cache size growth, concepts such as additional hardware assist engines, completely independent pipelines, and multiple outstanding fetch requests per requester are currently being evaluated and designed for subsequent S/390 CMOS servers.

Acknowledgments

The successful design and implementation of the G5/G6 binodal cache architecture was in every sense a team effort. The authors wish to thank the other members of the S/390 Custom SC/L2 design team: Bob Adkins, Dean Bair, Chris Berry, Yuen Chan, Tim Charest, Bill Dachtera, Joe Eckelman, Mark Fischer, Tom Foote, Cara Hanson, Glenn Holmes, Dave Hough, Christine Jones, Kevin Kark, Ed Kaminski, Anuj Kohli, Aaron Ma, Frank Malgioglio, Pradip Patel, Ed Pell, Don Plass, Tom Ruane, Bill Scarpero, Jim Schafer, Wojciech Taraszkiewicz, Gary Van Huben, George Wellwood, Bill Wollyung, Eric Young, and Adrian Zuckerman.

References

1. P. Mak, M. A. Blake, C. C. Jones, G. E. Strait, and P. R. Turgeon, "Shared-Cache Clusters in a System with a Fully

Shared Memory," *IBM J. Res. Develop.* **41,** No. 4/5, 429–448 (1997).
2. M. A. Check and T. J. Slegel, "Custom S/390 G5 and G6 Microprocessors," *IBM J. Res. Develop.* **43,** No. 5/6, 671–680 (1999, this issue).
3. G. Northrop, R. Averill, K. Barkley, S. Carey, Y. Chan, Y. H. Chan, M. Check, D. Hoffman, W. Huott, B. Krumm, C. Krygowski, J. Liptay, M. Mayo, T. McNamara, T. McPherson, E. Schwarz, L. Sigal, T. Slegel, C. Webb, D. Webber, and P. Williams, "600MHz G5 S/390 Microprocessor," *International Solid-State Circuits Conference Digest of Technical Papers*, February 1999, pp. 88–89.
4. T. J. Slegel, R. M. Averill III, M. A. Check, B. C. Giamei, B. W. Krumm, C. A. Krygowski, W. H. Li, J. S. Liptay, J. D. MacDougall, T. J. MacPherson, J. A. Navarro, E. M. Schwarz, K. Shum, and C. F. Webb, "IBM's S/390 G5 Microprocessor Design," *IEEE Micro* **19,** No. 2, 12–23 (March/April 1999).
5. P. Turgeon, P. Mak, D. Plass, M. Blake, M. Fee, M. Fischer, C. Ford, G. Holmes, K. Jackson, C. Jones, K. Kark, F. Malgioglio, P. Meaney, E. Pell, W. Scarpero, A. E. Seigler, W. Shen, G. Strait, G. Van Huben, G. Wellwood, and A. Zuckerman, "A Storage Hierarchy to Support a 600 MHz S/390 G5 Microprocessor," *International Solid-State Circuits Conference Digest of Technical Papers*, February 1999, pp. 90–91.
6. P. Mak, M. Blake, and G. Van Huben, "High Speed Remote Storage Cluster Interface Controller," filed 9/23/97, U.S. patent pending.
7. P. Meaney, G. Ingenio, T. McNamara, and P. Muench, "Method for Supporting 1½ Cycle Data Paths via PLL Based Clock System," filed 2/13/97, U.S. patent pending.
8. P. Mak, P. Meaney, W. Shen, and G. Strait, "Computer System UE Recovery Logic," filed 11/6/97, U.S. patent pending.
9. P. Mak, M. Blake, M. Fee, C. Jones, and G. Strait, "Computer System Deadlock Request Resolution," filed 2/2/98, U.S. patent pending.

Received January 7, 1999; accepted for publication May 24, 1999

Paul R. Turgeon *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (turgeon@us.ibm.com).* Mr. Turgeon is currently the manager of S/390 I/O and Connectivity Development. During the development of the S/390 G5 and G6 servers, he had design and project management responsibility for the high-performance cache, memory hierarchy, and processor subsystem designs described here. He has held various design and design management positions on the IBM 8100 Information System, ES/3090, ES/9121, and the S/390 G4, G5, and G6 systems, and has received several IBM formal awards including Outstanding Technical Achievement Awards for both the G4 and G5 designs. He holds a B.S. in electrical engineering from Rensselaer Polytechnic Institute (1979) and a Master's Certificate in project management from George Washington University (1996).

Pak-kin Mak *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (pmak@us.ibm.com).* Mr. Mak is a Senior Technical Staff Member in S/390 Custom Hardware Design. He received his B.S.E.E. degree from Polytechnic Institute of New York and his M.B.A. degree from Union College. Mr. Mak joined IBM Poughkeepsie in 1981, working on the ES/3090 BCE cache design. He has designed high-end system controllers and L2 caches for ES/9021 bipolar-based systems and was the technical team leader for the S/390 G4, G5, and G6 shared L2 cache designs. Mr. Mak currently holds three patents and has received three IBM Invention Achievement Awards, two IBM Outstanding Innovation Awards, two IBM Outstanding Technical Achievement Awards, and two Division Awards.

Michael A. Blake *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (mablake@us.ibm.com).* Mr. Blake is an Advisory Engineer in S/390 Custom Hardware Design. He was the SC Chip Logic Team Leader for the G5 and G6 systems. In addition, he designed the internode controllers used in these systems. He previously held design positions on the IBM ES/3090, ES/9021, and S/390 G4 systems. He has received several IBM formal awards, including an Outstanding Technical Achievement Award for the G4 design and an Outstanding Innovation Award for the G5 design. Mr. Blake holds one U.S. patent and has three patents pending. He holds a B.S. in electrical engineering from Rensselaer Polytechnic Institute (1981).

Michael F. Fee *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (fee@us.ibm.com).* Mr. Fee is an Advisory Engineer in S/390 Custom Hardware Design. He was the primary designer of the G5 system controller's microprocessor controller, cross invalidate, and store buffering functions. He developed and implemented logic synthesis and timing methodologies for the SC and L2 cache chips, and was responsible for achieving required cycle time targets. He has received IBM Outstanding Technical Achievement Awards for his contributions to the G4 and G5 products, and has three patents pending. Mr. Fee has worked exclusively on shared-cache system designs for IBM since receiving his B.S. in electrical engineering from Manhattan College (1989).

Carl B. Ford III *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (carl_ford@vnet.ibm.com).* Mr. Ford is an Advisory Engineer in the S/390 development

organization. He received his B.S. in electrical engineering from Rutgers University in 1983. Since joining IBM in Poughkeepsie, New York, in 1983, he has worked on processor controller code and logic design for the ES/3090 and ES/9021 bipolar mainframes and logic design for the S/390 G4, G5, and G6 CMOS mainframes. Mr. Ford was responsible for the hardware-assisted move engine and numerous other controller functions. He holds one patent and has received an IBM Invention Achievement Award and three Outstanding Technical Achievement Awards.

Patrick J. Meaney *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (meaney@us.ibm.com).* Mr. Meaney is an Advisory Engineer in S/390 Custom Hardware Design. He is the SC/L2 Timing Leader responsible for design and simulation to meet frequency goals. Mr. Meaney is also responsible for definition of SC RAS and recovery features for future S/390 CMOS systems. He was the L2 Cache Chip Design Leader for the G5/G6. Mr. Meaney holds ten U.S. patents and has six patents pending. He has held design positions on the ES/9021 and the S/390 G4, G5, and G6 systems, and has received several IBM formal awards, including an Outstanding Technical Achievement Award for the G4 and an Outstanding Innovation Award for the G5 design. He holds a B.S. in electrical and computer engineering from Clarkson University (1986) and an M.S. in computer engineering from Syracuse University (1991).

Rick Seigler *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (seigler@us.ibm.com).* Mr. Seigler joined IBM Poughkeepsie in 1980 after receiving B.S.E.E. and M.S.E.E. degrees from Rensselaer Polytechnic Institute. He worked as a Logic Designer and Systems Test Engineer on ES/3090 systems, and as a Recovery/Serviceability Systems Test Manager for ES/3090 and ES/9021 systems. In 1992 he joined S/390 Custom Hardware Design, where he now works as an Advisory Engineer; prior to that, he served one year on an IBM Faculty Loan assignment at the Georgia Institute of Technology in Atlanta. Mr. Seigler holds five patents and has received IBM Outstanding Technical Achievement Awards for his work on the G4 and G5 systems.

William Wu Shen *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (shen@us.ibm.com).* Dr. Shen is a Senior Engineer in S/390 Custom Hardware Design, currently responsible for memory subsystem architecture and memory controller interface design implementation. He holds eight U.S. patents and has ten patents pending. He joined IBM in 1978 and has held design engineering positions working on the memory subsystems of the IBM 8100 Information System, ES/3090, ES/9021, and S/390 CMOS G4, G5, and G6 systems. Dr. Shen has received several IBM formal awards, including an Outstanding Technical Achievement Award for G4 and an Outstanding Innovation Award for G5. He holds a Master of Science degree (1974) and a Doctor of Science degree (1981), both from Clarkson University. In addition to memory subsystem design, Dr. Shen is also interested in fault-tolerant design, system testing, and performance modeling.