G4: A Fault-Tolerant CMOS Mainframe

Lisa Spainhower Thomas A. Gregg

International Business Machines, Inc.
522 South Road
Poughkeepsie, NY 12601

Abstract

G4 is IBM's fourth generation CMOS microprocessor-based S/390 mainframe but the first to achieve fault tolerant equivalence - or superiority - with its predecessor ECL mainframes. CMOS technology provides much greater density and integration, assuring superior fault avoidance characteristics. The reduced power of CMOS makes bulk power redundancy and battery backup practical. However, the high density and circuit properties of CMOS pose new challenges for detection, recovery, and online repair.

G4 implements an innovative design for a high performance, fault tolerant, single-chip microprocessor. Microprocessor sparing is used as a concurrent repair mechanism. Increased memory density requires a new (76,64) S4EC/DED Error Correction Codes so that all single chip failures are correctable. As many as four I/O interfaces are packaged on an individual card, requiring both configuration management and automated maintenance procedures to assure all devices maintain connectivity during online repair.

1.0 Introduction

The integration and reliability of CMOS technology greatly improve the fault avoidance characteristics of S/390 mainframes over predecessor ECL machines. Better intrinsic failure rates coupled with huge reductions in parts count significantly increase MTTF. Reduced CMOS power requirements permit fault tolerance improvements for power and cooling, including redundant bulk power and battery backup.

Nonetheless, system fault tolerance requirements remain stringent and G4 logic designers had to develop new techniques to provide instantaneous error detection both to preserve data integrity and to permit transparent recovery. The higher density packaging resulting from increased integration also requires innovative techniques for concurrent online repair of logic components.

Section 2 will describe fault tolerance characteristics of ECL mainframes that must be met by G4 design and the technology differences between ECL and CMOS mainframes. In Section 3, design practices for S/390 ECL mainframe fault tolerance are presented. Section 4 concentrates on CMOS microprocessor design with subsections for common practices, G4 design point, and recovery and online repair. Sections 5-7 discuss the fault tolerant design characteristics of the other major system elements: memory, I/O subsystem, and power and cooling. Finally, conclusions are presented in Section 8.

2.0 Integration and concurrent repair

G4 is the fourth generation of S/390 CMOS mainframes, but the first to approximate the throughput and capacity of the most powerful ECL system, 9X2. In addition to matching performance, G4 was required to provide a level of fault tolerance equivalent to 9X2. All elements of 9X2 were designed for fault tolerance [Spa92, Spa94]. Data collected from the full-field installed base for calendar year 1995 showed that 9X2 requires repair less than once per year. 71% of the repairs are performed concurrently; there is no downtime associated with either the failure or the repair. An

additional 18% of repairs are performed on a scheduled, deferred basis. There is no downtime associated with the failure, but the repair, which is performed at a time convenient to the user during off-peak hours, does require system downtime. The remaining 11% of repairs cause unplanned system downtime. Mean time between unplanned system downtime for 9X2 systems ranges from 10-16 years. Evaluation of potential enhancements concluded that, while incremental improvements are always possible, 9X2 was nearing the limits of what was feasible for a general purpose SMP. Limited storage control logic, system clocking, and second level packaging failures could result in unplanned downtime but the product and performance costs of dynamic recovery in these areas is prohibitive.

Hardware fault tolerance, even if perfect, can't prevent other sources of downtime, particularly software. S/390 designers recognized that a new system structure was required to achieve availability demands exceeding SMP capability. S/390 offers parallel sysplex in which up to 32 mainframes can be connected into a highly available cluster. Parallel sysplex is implemented in operating system and middleware as well as hardware. It offers cluster-wide data sharing and workload management and is designed for continuous availability. A overview of parallel sysplex is described in [Nic97] and details of high availability features can be found in [Bow97].

For the first three CMOS generations, system availability was acceptable because of the intrinsic failure rate reduction of CMOS as compared to ECL and because a cluster solution was possible. The following chart highlights some of the key differences in parts utilization between 9X2 and G4:

Table 1. Parts Usage Comparison: ECL and CMOS

Component Count	ECL 10-way SMP	CMOS 10-way SMP
Multichip modules	56	1
Processor Boards	14	1
Processor Chips	5960	31
Storage Cards	48	4
Power Supplies	202	15
Cables	1388	36
Fans	87	6
Power (KW)	144	5

The challenge for G4 was to provide equivalent or superior function in fault tolerance and concurrent repair compared to 9X2. Design constraints of modern high performance microprocessors complicate the implementation of error checking. The ability to perform deferred and concurrent maintenance in 9X2 depended to a large extent on physical isolation of logical entities like CPUs and I/O channels. Denser packaging requires new techniques for graceful degradation and online repair.

3.0 S/390 fault tolerant design

Two distinct levels of fault tolerance, circuit level and system level, were developed for ECL. Design engineers follow rules and guidelines to assure consistent circuit level error checking and isolation throughout the computer regardless of the particular logic's function. The goal is to detect and contain all failures of the hardware. Every register and latch is protected by error checking. Arrays, data paths and control paths are protected either by parity or ECC. Checking mechanisms employed for state machines, ALUs, and other control logic include illegal state detection, parity predict, and pinpoint explicit redundancy. Designers conduct lengthy design reviews to evaluate and improve coverage. Especially for control logic, circuit overhead is typically high, designs are complex, and effectiveness is difficult to verify.

System level fault tolerance exploits the inherent redundancy of an S/390 system with its multiple identical functional entities, like CPUs and I/O channels. The design challenge is to permit failures, recovery, and degradation to occur without any noticeable interruption to the ongoing workload. System level fault tolerance is quite different from circuit level because the function of the logical entity is critical in determining the most effective techniques. S/390 permits, for instance, one CPU in an SMP to be varied offline and the programs it was formerly executing to be moved to any surviving CPU without needing an IPL or reboot. System level fault tolerance concepts are also applied at a lower level; large caches, for example, have more than one location into which data can be stored so each of those locations can be treated as a redundant entity. When one location is found to have a permanent fault it can be marked as no longer usable while the rest of the cache continues to function normally.

Much of the S/390 I/O subsystem has been implemented in CMOS for nearly a decade and the memory hierarchy is primarily composed of large arrays. For these two major functional segments, the established methodology of fault tolerance - circuit level and system level - is maintained for G4, although denser packaging required new implementations. The CPU however required reevaluation of existing mechanisms. G4 design requirements included not only delivery of 9X2

equivalent fault tolerance, but also equivalent performance. Traditional S/390 ECL inline error checking can adversely affect cycle in high performance CMOS design. In addition, the system had an aggressive development schedule and design, verification, and test of the checking lengthen development time.

4.0 Microprocessor design

4.1 Common practices

Existing single chip CMOS microprocessors, whether RISC or CISC, have limited hardware error detection. High coverage error checking in microprocessors is typically implemented by duplicating chips and comparing the outputs. Intel builds into its chips functional redundancy checking logic that permits master/checker Pentium microprocessor pairs [Int94] and Tandem designs similar off-chip logic to create perfectly checked microprocessor pairs for its Himalaya systems [Tan95]. Duplicate and compare is, however, adequate only for detection and G4 also required logic to permit single CPU dynamic recovery. Thus, there were no industry models for G4 to emulate.

A microprocessor has a component to fetch and decodes instructions (I-unit), one or more instruction execution elements (E-unit), and a cache. Modern microprocessors usually include parity for cache data and data paths where data is generally moved without being altered. Checking of the control, arithmetic and logical functions in the I-unit and E-unit is difficult, time-consuming, and introduces performance penalties. As a result, today's microprocessors leave these components unchecked.

Accepted wisdom is that single chip microprocessors are sufficiently reliable that failures are rare and that other mechanisms - timeouts and software detection, for instance - will discover the few problems that do occur. There is however a real exposure to malfunctions in the hardware resulting in incorrect data being written to memory. The model proposed by Horst, et al. [Hor93], projects a pessimistic bound of one in 75 microprocessors causing a data corruption error each year. The largest G4 systems have 12 microprocessors, which would result in 1 in 6 systems with a data integrity exposure each year. S/390 design guidelines have always required that data integrity exposures be considered flaws. It's impossible to predict the effectiveness of indirect detection mechanisms; in some cases timeouts and software detection will prevent data integrity problems. Even when the detection is successful, the system will usually hang and result in application downtime. It's preferred to recover transparently whenever feasible, necessitating instantaneous detection and containment of failures.

4.2 G4 design point

G4's CPU is a 17.35x17.30 mm^2 CISC microprocessor implementing the S/390 architecture. It's built with IBM Microelectronics' CMOS6S process (0.4μm lithography, 2.5V) and has approximately 7.8 million transistors. The microprocessor regularly performs at greater than 300 MHz and operates in a system at up to 400 MHz [Web97a]. The CPU is a pipelined design with two fixed-point and one floating-point execution elements. Only one unit is executing at a time and one instruction can be decoded each cycle. Additional details of the G4 microprocessor design are found in [Web97b]. Primary fault tolerance goals are to protect data integrity, recover transparently from transient failures, degrade gracefully from permanent failures, and, in most cases, repair permanent failures without system downtime.

9X2 is a superscaler ECL CPU which permits out-of-order instruction execution. It was designed to optimize cycles per instruction (CPI). Experienced S/390 CPU designers concluded that a similarly complex design in CMOS couldn't be executed on the required schedule. Instead, to achieve 9X2 equivalence, the primary performance objective was to design the microprocessor with the fastest possible cycle time. G4 has in fact demonstrated a cycle time of 2.5 ns.

ECL packaging permits extensive inline checking to be efficiently implemented [Spa92]. Each 9X2 CPU required approximately 400 chips. Often the chips were I/O pin limited and logical functions didn't always divide neatly. Developers could exploit the 'leftover' logic for decode checkers, localized functional redundancy, and other high overhead checking mechanisms without increasing the chip count. For 9X2, inline CPU error detection used about 30% circuit overhead.

Comparable inline checking presented one of the main inhibitors to achieving G4's required low processor cycle times. A common method of inline checking for detecting errors in ECL combinatorial logic uses parity prediction. Separate logic with the same inputs - both data and parity - as the actual function (ALU, shifter, state machine, etc.) calculates the parity of the function output. Parity is checked in the same or a subsequent cycle. This method detects failures in the inputs and output as well as the combinatorial logic. A simple implementation of a parity predictor is to duplicate the function and generate parity from the input parity and the output of the duplicated function. More optimized

techniques have been designed for common functions such as ALUs and shifters. The path length through the parity prediction logic for an ALU or shifter is longer than the path length through the actual ALU or shifter function. This is true for both ECL and CMOS, but the effects in CMOS can be even worse than in ECL, as described below:

1.High fan-in and fan-out capabilities in ECL result in smaller differences between the actual function and the predicted parity path lengths compared to CMOS. Higher capacitance associated with CMOS circuits as their fan-in and fan-out increases makes path lengths in this technology more susceptible to the complexity of the function.

2.The added logic for the parity prediction itself increases the chip area. In ECL, the parity prediction logic could often be packaged on the same chip as the main function, and chip to chip wiring is not affected.

3.Additional chip area in turn causes longer interconnecting wiring that also increases the path length.

4.The lower fan-out capabilities of CMOS further increases path length since the prediction circuits must be connected to critical data buses.

Of course, the primary performance objective for G4 is cycle time reduction. Another design point, for instance a superscaler design that optimizes CPI, might find the adverse effects of inline checking more tolerable.

Regardless of cycle time optimization, inline checking requires more development time. Placement of error checkers requires skill and diligence. Adequate automated tools aren't available, so coverage is determined by exhaustive design walkthroughs. Simulation of the checkers requires carefully placed error injection to assure detection, and mistakes in the design sometimes cause false detection. Extra error injection steps are also required during testing of the logic to verify the system correctly recovers from the error. Design errors in checking logic tend to be discovered very late in the development cycle, requiring unanticipated changes at critical points in the cycle.

Performance and schedule requirements dictated a new approach for the data manipulating elements of G4. The requirement for dynamic recovery, coupled with other design constraints (e.g., power, performance, second-level packaging) pointed to a highly-checked single-chip microprocessor design for G4. However, inline checking of control and arithmetic logic was prohibited by the performance and schedule penalties. The decision was made to duplicate I-unit and E-unit and compare outputs.With a carefully laid out floorplan, there are no cross-chip performance limiting paths. The cycle time is the same as if the same design were implemented on a smaller chip with only one unchecked I-unit and E-unit. The compare and detect cycle is completely overlapped in the instruction execution pipeline, so CPI is also unaffected by duplication.

Most of the microprocessor area is devoted to the cache where data is primarily moved around unchanged. Updates to memory are maintained in an ECC-protected store buffer during instruction execution and transferred to an ECC-protected L2 cache when instruction execution completes. Thus, for the on-chip cache, low-overhead parity checking is sufficient.

Total circuit overhead for cache parity, Register Unit ECC (the R-unit is described in the next section), and duplicate I-unit & E-unit is about 40%. Design, verification, and test are greatly simplified compared to inline checking. G4's larger chip area may result in lower yields and higher costs, acceptable for a relatively low volume, high performance microprocessor. This is particularly true when, as in this example, it enhances meeting all of the key requirements: fault tolerance, performance, and schedule. A high-level overview of the microprocessor design is shown in Figure 1.

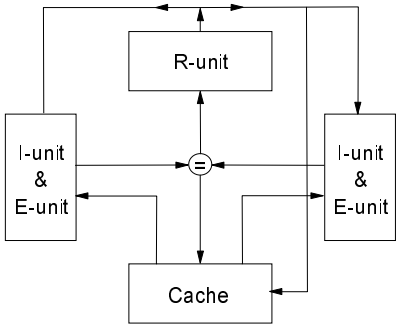

Figure 1. Microprocessor diagram

4.3 Recovery and online repair

The microprocessor also includes a new element, the R-unit. It is primarily made up of an ECC-protected register file, the checkpoint array. It can accommodate 128 32-bit and 128 64-bit registers. To permit transparent recovery, as instructions are executed, the results from the two I and E units are compared and, if equal, changes to the state of the CPU are placed in an update buffer in the R-unit. Once the execution successfully completes, the update buffer contents are placed in the R-unit's checkpoint array which keeps track of the entire state including register contents and instruction addresses.

During the n-1 stage of the instruction pipeline, results from the two I and E-units are compared, and parity is checked and ECC generated for operand stores. In the final pipeline stage (completion) the checkpoint array is updated with new state information and any pending L1 stores are sent to the store buffer. This stage will be blocked if there was an error detected in a previous stage. This assures that any completed instruction is error-free and the checkpoint array and the store buffer accurately describe the state of the machine at successful instruction execution completion.

When an error is detected, the store buffer contents are sent to L2, the CPU (except of course the R-unit) is reset, and the cache and its directory are purged and written with good parity. R-unit ECC is performed if necessary and checkpoint array contents are refreshed. The state of the CPU is returned to what it was at the last instruction completion. The checkpoint array is read again and any detected error is considered to be permanent, since it was just refreshed with good ECC. In that case, recovery will not succeed and the CPU is stopped. If the checkpoint array is error-free, then serialized instruction processing starts up, retrying the instruction that caused the error. If it is successful, the failure was transient and the CPU resumes pipelined instruction processing. If not successful, then the error is permanent and the CPU will be stopped. The recovery scenario is completely controlled by hardware.

Graceful degradation is performed by coordinated hardware and operating system action. If the error is permanent and not in the checkpoint array, the CPU will signal the operating system which will store the state information in the dispatch queue of another CPU in the system. The failed CPU will be dynamically varied out of the active configuration. The task it was executing will be restarted according to normal dispatch priorities. In essence, when retry won't work on the original CPU because of a permanent fault, the system will retry the instruction and restart the task on a different, error-free CPU.

A single G4 processor card has 12 physically identical microprocessors, up to 10 of which function as CPUs, executing application programs. One or two serve as System Assist Processors (SAP), a role including some system maintenance and I/O processing function. There is no hardware difference between a CPU and a SAP. G4 is available with all possible configurations, from two microprocessors (one CPU and one SAP) to the maximum of 12. On 9X2, each CPU was comprised of several 100+ chip modules mounted on a specialized board with its own unshared power source. So, if a CPU failed, not only could it be gracefully degraded and removed from the configuration, but its repair could also occur concurrently with continued operation of the remaining CPUs. Despite the lower failure rate of G4 microprocessors, it is still desirable to develop a scheme for concurrent repair.

For concurrent CPU or SAP repair, it was decided to always provide additional microprocessors beyond the number ordered, with the exception of those systems ordered with the physical maximum of 12. These 'spare' CPUs can assume the function of a SAP or CPU if one of the others on the card should fail, returning the system to full performance capacity. When a CPU with a permanent failure is varied out of the configuration, the system operator can assign a spare CPU to enter the configuration. Current restrictions in how CPU IDs are assigned prevent the operation from being performed automatically by the system. However, the new CPU can be added to the configuration and begin executing application programs with no interruption to the ongoing workload. Contrasted to 9X2 system online CPU repair which requires service personnel travel and parts procurement before it can be carried out, CPU sparing permits the system to be restored to full capacity in minutes as opposed to hours.

Table 2. Memory Hierarchy Characteristics

Level	Design	Data Protection	Size	Sharing
1	Store-through	Byte Parity	64 KB	no
2	Store-in	(72,64)SEC/DED ECC	768 KB	3 uPs
2.5	Store-through	(72,64)SEC/DED ECC	≤4 MB	all uPs
3	---	(76,64)S4EC/DED ECC	≤24 GB	all uPs

5.0 Memory hierarchy fault tolerance

The design objective in the memory hierarchy is to continue uninterrupted operation when data errors occur. The failure model predicts a predominance of transient failures, so all levels of the hierarchy must transparently recover from them. Additional fault tolerance provides recovery from many permanent failures. Data redundancy is provided by two primary means, store-through (write-through) cache design and Error Correction Codes (ECC). The characteristics of the four levels of the hierarchy are described in Table 2. Additional information about the design can be found in [Doe97].

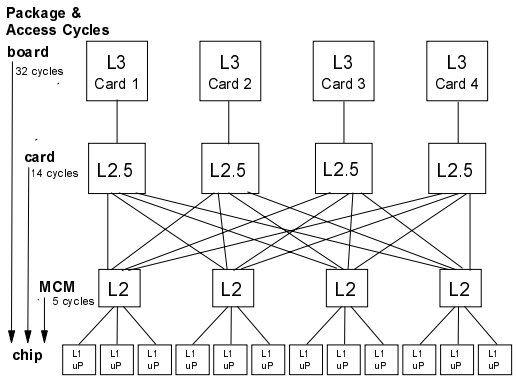

Figure 2. Memory Hierarchy Layout

L1 is the microprocessor cache. Pending instruction results are maintained in both L1 and an ECC-protected store buffer. When instruction execution completes, updated results are immediately stored into L2. Because L1 data is always replicated, byte parity is adequate for protection. Also, if a CPU experiences a permanent failure, all of the changed data is in L2 and accessible to the system.

Each of the possible maximum of four L2s is shared among three microprocessors. L2 maintains cache coherency for the system and may contain the only version of changed data. L2 enforces strict cache coherency for L1 and L2; data must be in an exclusive state in a CPU's L1 directory and no other copies of the data can exist in any L1 or L2 before that CPU can modify the data. To prevent single bit transient errors from resulting in lost data, L2 is protected by ECC. Permanent faults in L2 that might result in an uncorrectable data error can be avoided by using a cache delete capability. Faulty locations either in the data array or in the address directory can dynamically be marked as invalid and the system continues operating with a very slightly smaller L2. A complete description of how L2 manages storage can be found in [Mak97].

L2.5 is added to the system purely to enhance performance. It contains shared read-only data and permits faster access than main memory (L3). Since L2.5 never contains the only copy of modified data, if it should fail, data integrity is preserved and the system can continue operation. Nonetheless, its role in improving performance is important enough that L2.5 is protected by ECC .

L3 is G4's main memory and also its expanded storage. (Expanded storage is pageable electronic storage used primarily for database buffering. In the 9X2 systems, it was packaged as a separate physical level, L4. In G4, expanded storage is implemented as an address range of L3.) The four physical layers of the memory hierarchy, packaging level and access cycles are shown is Figure 2. As with the CPU, the greater technology integration of memory creates difficulties in providing fault tolerance equivalence to earlier S/390 systems. It is possible that certain control logic failures may result in unscheduled system downtime, but the design objective is that array failures will never result in a system failure or a customer outage. 9X2 was able to deliver this with a (72, 64) Single Error Correct/Double Error Detect Error Correction Code primarily because the memory was designed so that each chip supplied one bit of the ECC word. That meant that any failure mechanism of the chip - single cell, word line, bit line, or complete chip kill - was still correctable by the code.

Improvements in chip and memory packaging density require that today's memories use multiple bits per chip for each ECC word. In the case of G4, each chip provides four bits. In order to continue to meet the design objective it became necessary to devise a new ECC. Any single chip failure - one to four bit - must be correctable and permit the system to continue operating. When a chip is b-bit ($b \geq 2$) wide, an access to a 64-bit data word may have a b-bit block or byte error. There are codes to variously correct single b-bit and detect double b-bit errors. For G4, a code with 4 bit correction capability, or a S4EC code was needed. Because the system design includes dynamic online repair of chips with massive failures, it was not necessary to design a (78,64) D4ED code which would have required an extra chip per checking block.

Background scrubbing is performed on L3 data to reduce the frequency of transient single bit failures. Automatic online repair of faulty DRAMs is done using built-in spare chips. Counts of correctable errors are maintained on a per chip basis. When a threshold is exceeded, the data from the over-threshold chip is dynamically transferred to an error-free spare chip. A chip with systematic failures, e.g., word line or chip kill,

will rapidly exceed threshold and be removed from the system. However, it's possible that prior to the systematic failure another chip or other chips may have single bit failures that do not result in the threshold being exceeded. There is an exposure that a previous single bit error could align with a sudden systematic failure and escape detection. The (76,64) S4EC/DED ECC is therefore designed to assure that all single bit failures of one chip (and a very high probability of double and triple bit failures) occurring in the same double word as a 1-4 bit error on a second chip are detected [Fuj92].

6.0 I/O subsystem fault tolerance

The G4 I/O subsystem is system hardware that connects main memory and CPUs to devices and their controllers over various standard interfaces. Extensive inline checking is implemented both to detect errors and preserve data integrity. The design is described completely in [Gre97]. The fault tolerant design objectives of the I/O subsystem are to exploit the redundant paths between all devices and main memory and to minimize the scope of any failures.

The first objective is primarily accomplished through architecture common to the device and its controller, system hardware, and operating system. The system is configured with multiple paths between main memory and the I/O devices. The I/O subsystem controls the status of available paths to devices, the choice of paths to devices, and the routing of data transfer operations. State information about ongoing I/O operations is maintained in main memory. Permanent faults in the redundant paths can be dynamically bypassed and failing hardware can be removed from the configuration without affecting other ongoing data transfers. If a path experiences a permanent fault between active data transfers, the device can reconnect to a different hardware path than it was initially using. If the failure occurs in the midst of data transfer, the device may reconnect or it may request operating system recovery.

Limiting the scope of the recovery action is achieved with a three tier recovery scheme. The most restricted occurs when there is an error on the interface between a device and the system hardware. The operation is retried by the I/O subsystem if detected by the device and by the device if detected by the I/O subsystem. Most errors of this nature are transient and if recovery is successful, the operating system is not interrupted. The second type of recovery is performed when the error occurs within the system hardware and all state information for that path is valid. In this case, the data transfer in process may be terminated and the operating system will restart the operation. The third type of recovery takes place when state information is damaged. The operating system will terminate and subsequently restart all operations using that path.

A G4 system can have as many as 256 standard interfaces. Because of the density of CMOS technology and packaging constraints, there are up to four interfaces per card. Although the cards can be replaced online, the packaging introduces some difficulties. First, the system must be properly configured so that redundant paths don't use interfaces on the same card. Second, when the repair is performed, the other three paths must be temporarily gracefully degraded out of the active configuration. This is controlled by automated maintenance procedures. The LED which indicates the card to be replaced will not be lit until all necessary actions have been completed. In the future, as packaging densities increase, it's likely that the system will be designed with spare interfaces, much like memory has spare DRAMs.

Concurrent repair procedures for the SAP, which performs both I/O and system maintenance function, were discussed in the microprocessor section. G4 systems have either one or two SAPs. For systems with two SAPs, one is designated master and the other slave. If the slave fails, the master continues. If the master fails, the previous slave is dynamically designated as master and the system continues running without experiencing any downtime. Because many systems - depending on the throughput needs of the I/O subsystem - have only one SAP, it is most critical to invent a recovery technique for single SAP failure. Otherwise, the entire system would be unavailable until the processor card was replaced. The only difference between a CPU and a SAP is the code that it is running. Therefore in any system with at least two CPUs and one SAP, a CPU can be dynamically and automatically reconfigured as a SAP.

7.0 Power and cooling fault tolerance

To meet mainframe availability requirements, the power and cooling subsystem is designed with no single points of failure and concurrent repair capability of virtually all components. In G4, the lower power consumption make meeting these requirements easier than in 9X2; the maximum power for the largest G4 is 5 KW while the maximum power of the largest 9X2 is 144 KW. In ECL mainframes, the high cost of high power components made duplication of bulk power elements

prohibitively expensive, but for CMOS it becomes practical .

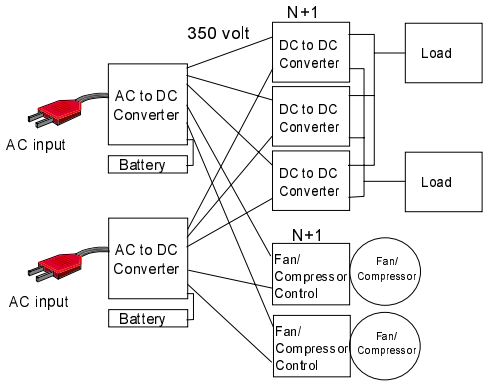

Figure 3. Power and cooling configuration

Figure 3 shows the major components of the power and cooling subsystem. Separate AC power inputs feed dual AC to DC converters, and each of these has several 350 volt DC outputs. Also, each can be connected to a 350 volt gel cell battery. The lower power requirements of the CMOS mainframes allows battery backup to be attractive, and they provide at least 10 minutes of backup power in the largest mainframes. The AC to DC converters are fully redundant, and when they are both on line, the load is shared. If one of them fails, the other takes over the 350 volt load. Also, three phase AC input current is used, and one of the phases can fail without causing an outage. If one of these converters fails, it can be concurrently replaced.

Both AC to DC converters feed all DC to DC converters and controllers for the fan and compressor motors through individual point to point 350 volt cables. The DC to DC converters provide multiple output voltages for the circuitry, the fans move air for cooling the circuitry, and the compressors are part of a refrigeration system to cool the processor module of the largest G4. The DC to DC converters, controller/fan assemblies, and controller/compressor assemblies are all in an 'N+1' configuration. For example, if two DC to DC converters are required to supply the load, a third is added for online redundancy. In the case of the fans and compressors, two are always used. All of these components can be concurrently replaced. In the case of the fan and compressor motors, some failures are detected by measuring the back EMF of the motor field coils. When a fan fails for any reason, the speed of the second fan is increased to compensate for the loss. Under normal operation, the speed of the fans is kept to a minimum to reduce acoustic noise. With the compressors, only one of the pair operates at the same time. A switchover is made every 24 hours to ensure that both assemblies are operational. When a controller/compressor fails, it can be concurrently maintained. Quick disconnects are provided at the evaporators for the refrigerant hoses. Coordinated control of all power converters and fan/compressor controllers uses duplicated communication paths. The control interface to the power and cooling subsystem is provided by a Thinkpad notebook computer.

The bulk of the I/O subsystem hardware is packaged into various circuit cards that provide many different I/O interfaces, and most of these cards can be concurrently replaced. Live insertion of circuit cards causes large noise spikes on the power supply, so a low noise mechanism of applying power is required. 9X2 has only a few different I/O card types, and during concurrent removal and insertion, the power to the individual cards is ramped up and down by a system of multiple pin lengths of the card connector. The longest pins supply bulk power while shorter pins control solid state switches (FETs) on the card. These switches supply current for ramping up and down the power. To meet modern I/O requirements of the CMOS mainframes, many more card types of widely varying power requirements are needed. For example, ESCON I/O cards require several times less power that ATM cards. These requirements drive a more flexible design called the 'soft switch.' Instead of multiple length pins, each I/O card position has a pin that controls the soft switch on the card. Before an I/O card is removed, the control pin is deactivated causing the power on the card to drop. After a card is inserted, the control pin is activated and the power to the card is turned on.

To reduce human errors during concurrent card replacement, a group of LED indicators, one for each card, is used to positively identify the card to be replaced. 9X2 uses indicators on the cards themselves. On G4, these indicators have been moved off the cards and onto the card cage. This is an improvement over the 9X2 design because the indicator can also flag the empty card slot that will receive the new card and because the indicators are provided in a uniform way for all card types. Some CMOS mainframes have internal disk drives. These are configured as RAID 1 (mirroring), and individual power control similar to the soft switch allows their concurrent replacement.

8.0 Conclusions

G4 is designed with a rich set of fault tolerant features. New denser CMOS technology offers advantages in fault avoidance and makes practical the implementation of enhanced power and cooling fault tolerance. It also poses challenges in error detection and online repair. G4 has a unique, fault tolerant single-chip microprocessor and a new scheme for online CPU repair. A S4EC Error Correcting Code in main memory coupled with DRAM sparing eliminates array chips as a cause of system downtime. Automated procedures allow three correctly functioning I/O interfaces to be gracefully degraded when the card they share with a fourth, failed interface must be concurrently repaired. Further increases in I/O interface packaging density will probably lead to a design for built-in spares, as has already been done for microprocessors and memory. At year end 1997, G4 systems were experiencing a mean time to unplanned system downtime of 22-26 years, exceeding the fault tolerant performance of 9X2.

Acknowledgments

The authors wish to thank Cyril Price and John Liptay for recounting the G4 microprocessor design process. We are especially grateful to Prof. Dave Rennels, our thought-provoking and helpful shepherd. Thanks to Guru Rao for encouraging us to write this paper.

References

[Bow97] N.S. Bowen, J. Antogini, R.D. Regan, and N.C. Matsakis, "Availability in parallel systems: Automatic process restart", *IBM Systems Journal,* **36**, No. 2, 284-300 (1997)

[Doe97] G. Doettling, K.J. Getzlaff, B. Leppla, W. Lipponer, T. Pflueger, T. Schlipf, D. Schmunkamp, and U. Wille, "S/390 Parallel Enterprise Server Generation 3: A Balanced System and Cache Structure", *IBM Journal of Research and Development,* **41**, No. 4-5, 405-428 (1997)

[Fuj92] E. Fujiwara and M. Hamada, "Single *b*-bit Byte Error Correcting and Double Bit Error Detecting Codes for High-Speed Memory Systems", *Proceedings of the 1992 IEEE International Symposium on Information Theory,* 494-501 (1992)

[Gre97] T.A. Gregg, "S/390 CMOS Server I/O: The Continuing Evolution", *IBM Journal of Research and Development,* **41**, No.4-5, 449-462 (1997)

[Hor93] Robert Horst, Doug Jewett, and Daniel Lenoski, "The Risk of Data Corruption in Microprocessor-based Systems, *Proceedings Fault-Tolerant Computing Symposium,* 576-585 (1993)

[Int94] Intel Corporation, *Pentium Family User's Manual, No. 1: Data Book,* Order No. 241428, 12-7 - 12-8 (1994)

[Mak97] P. Mak, M.A. Blake, C.C. Jones, and P.R. Turgeon, "Shared Cache Clusters in a System with a Fully Shared Memory", *IBM Journal of Research and Development,* **41**, No. 4-5, 429-448 (1997)

[Nic97] J.M. Nick, B.B. Moore, J.Y. Chung, and N.S. Bowen, "S/390 cluster technology: Parallel Sysplex", *IBM Systems Journal,* **36**, No. 2, 172-201 (1997)

[Spa92] Lisa Spainhower, Jack Isenberg, Ram Chillarege, and Joe Berding, " Design for Fault-tolerance in System ES/9000 Model 900", *Proceedings Fault-Tolerant Computing Symposium,* 38-47 (1992)

[Spa94] Lisa Spainhower, Thomas A. Gregg and Ram Chillarege, "IBM's ES/9000 Model 982's Fault-Tolerant Design for Consolidation", *IEEE Micro,* **14**, No. 1, 48-59 (1994)

[Tan95] Tandem Computers Incorporated, "NonStop Himalaya Range: K200, K200, and K20000 Servers", *NonStop Servers Product Description* (1995)

[Web97a] Charles F. Webb, et al., "A 400MHz S/390 Microprocessor", *IEEE International Solid-state Circuits Conference,* 168-169 (1997)

[Web97b] C.F. Webb, J.S. Liptay, "A high-frequency custom CMOS S/390 microprocessor", *IBM Journal of Research and Development,* **41**, No. 4-5, 463-473 (1997)