FAULT TOLERANT MULTIPROCESSOR SYSTEMS

Gavin Holland and Dhiraj K. Pradhan

1 Introduction

Over the past decade, the motivation for higher throughput from existing technology has forced multiprocessor and parallel computing systems into the mainstream. A recent trend is massively parallel processors (MPP), where hundreds of computing nodes are interconnected to form an integrated system. From a recent article in the IEEE, Spectrum Zorpette (1990), Table 1 portrays that more than one thousand such parallel machines sold for more than two hundred million dollars. Despite such popularity, these new systems remain largely unsuitable for applications which, because of high failure rates, demand high reliability or prolonged availability.

Company	Number sold to date	1991 Sales in $ Millions
Intel	> 325	90
Meiko Scientific	> 425	25
nCUBE	> 300	18
Parsytec Gmbh	100	8
Thinking Machines	90	85
TOTAL	> 1240	226

Table 1: Sales of Parallel Computers

Parallel systems suffer from high failure rates, despite the common perception of their inherent reliability. The following, from a recent publication by Harper and Lala (1991), points this out:

> The assertion is often made that parallel processors are intrinsically reliable, fault tolerant, and reconfigurable due to their multiplicity of processing resources. In fact, the only intrinsic attribute guaranteed by multiple processors is *a higher total failure rate*

Parallel systems, in fact, are in real need of fault-tolerant features if dependable service is to be expected akin to their uniprocessor ancestors. The MTTF of the Intel paragon system, for example, is thought to be as low as just several hours, clearly unacceptable for most applications. As parallel systems increase in scale, reliability needs increase, as well.

Because massively parallel processors currently are solely driven by performance requirements, availability and reliability have not been much investigated. The following table illustrates how availability of these systems already has become a serious problem. Scaling up these systems drastically reduces the mean time between failures (MTBF).

Fault tolerance in complex multiprocessor systems, it is generally agreed, should incorporate both hardware and software techniques. Typically designed

Manufacturer	1 Node Workstation	1000 Node MPP
DEC	35,872 *hours**	35 hours
HP	58,700 *hours**	58 hours
SUN	40,601 *hours**	40 hours
NASA/IBM AP101S	20,000 *hours*$^{\#}$	20 hours

$\star$ IEEE Spectrum, April 1992.

EDN 1992

Table 2: MTBF's for different manufacturers

in a hierarchical fashion, dependability issues need to be addressed: (1) circuit and technology level concerning the exact technology required to implement various functions (e.g., radiation-hardened memory, gallium arsenide chips for CPUs, etc.) along with circuit design techniques to improve testability and reliability of individual modules; (2) node-level architecture concerning design of the VLSI chips which implement a processing node (e.g., which functions need to be checked using which mechanism); (3) inter-node architecture relating to how these nodes are connected together and how the system will be reconfigured in the presence of faulty processors and interconnection switches and links; (4) operating system level dealing with checkpointing and rollback after certain portions of a system have been identified as faulty, the work load of a faulty processor being distributed to other processors (software errors may be present at this level); (5) applications level where the user puts in some reasonableness checks on the results of computations (programming errors may be present at this level). Fault tolerance typically is directed at each level in the design. Faults not detected (recovered from) at lower levels need to be handled at higher levels. This chapter reviews the basic issues and techniques of fault-tolerance, the next chapter providing detailed reviews of some of the existing systems.

Before the detailed discussion of various fault tolerance techniques proposed in literature, a brief review of the various classification of multiprocessors available is provided along with some examples.

2 Case Studies

This section provides an in-depth examination of three different commercial fault-tolerant multiprocessor architectures.

2.1 CASE STUDY 1: Tandem Multicomputer Systems

2.1.1 Introduction

Tandem Computers, Inc. has been developing fault-tolerant, multicomputer systems since the 1970's. Among its most recent offerings are the Cyclone and Himalaya families of reliable parallel processing systems. The following presents two of these systems: the NonStop Cyclone and the Himalaya K10000.

2.1.2 NonStop Cyclone

The NonStop Cyclone Tandem (1990), Tandem (1993), shown in Figure 1, can be classified as a MIMD, loosely coupled, distributed memory, shared bus multiprocessor, Hwang and Briggs (1984). First introduced in 1989, the Cyclone system was designed with the primary objective of preventing a single hardware or software failure from disabling the system. Its processors execute independently, running Tandem's proprietary, fault-tolerant, load-balancing, GUARDIAN90 operating system. The GUARDIAN90 operating system maintains idle backups of user processes on separate processors, transparently, in case of failure, Lee and Iyer (1993). It also provides load-balancing to minimize idle components and increase performance. A fully configured Cyclone system contains 16 processors, 2GB of main memory, and 64 I/O channels. However, further expansion is available through Tandem's FOX network and EXPAND networking software, which can connect up to 255 Cyclone systems in a single distributed, fault-tolerant network.

System Architecture The Cyclone's processors were designed around an advanced, *superscalar*, CISC architecture, Horst, Harris, and Jardine (1990). This architecture uses dual eight-stage pipelines and an instruction-pairing technique to allow simultaneous operations on multiple instructions. Additional features include separate 64KB instruction and data caches, and the use of dynamic branch prediction algorithms for the minimization of costly pipeline "bubbles." Error-correcting memory arrays, totalling up to 128MB, comprise the main memory for each Cyclone processor.

The Cyclone's architecture uses multiple, physically separate *sections*. Each section houses up to four processors interconnected by a proprietary, fault-tolerant bus called Dynabus. Dynabus is composed of redundant interprocessor busses, which have an aggregate throughput of 40MB/sec. Up to four sections can be interconnected using another proprietary, fault-tolerant interprocessor bus called Dynabus+, shown in Figure 2. Dynabus+ uses fiber-optics to achieve the same aggregate bandwidth as Dynabus over a much greater distance, up to 50 meters. This eases installation by allowing sections of the Cyclone system to be placed in different locations within a room, or in different rooms altogether. However, regardless of their location, the sections must be powered by the same

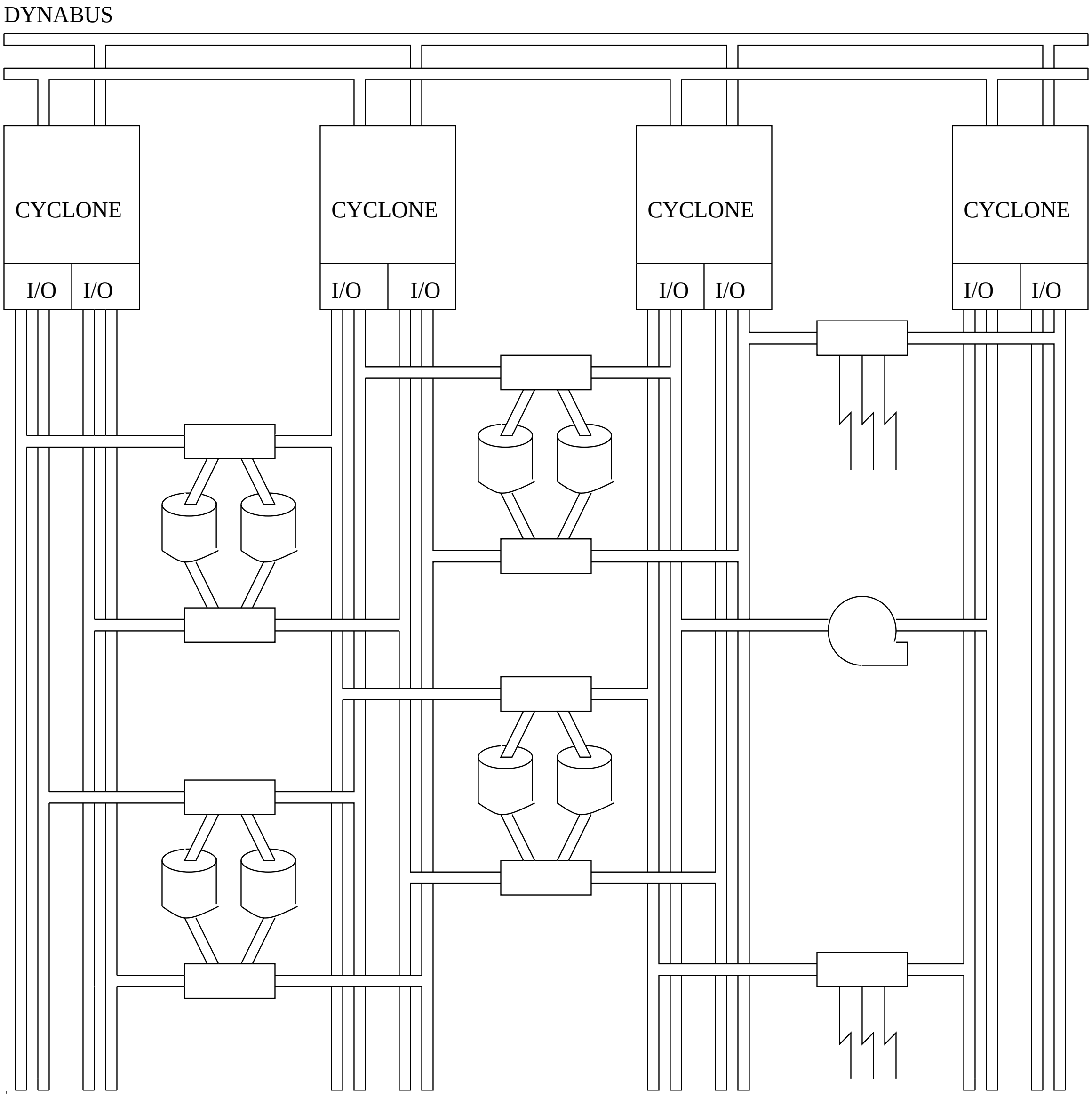

Figure 1: Tandem NonStop Cyclone System

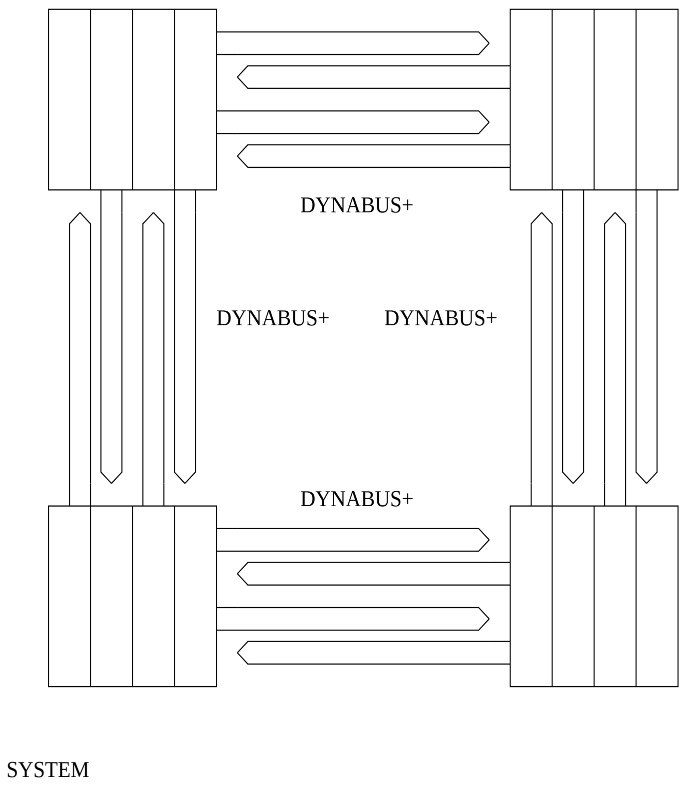

Figure 2: Four Separate Sections Connected by Dynabus+

external source. By isolating Dynabus+ from Dynabus, processors can communicate simultaneously within separate sections at the maximum bandwidth provided by Dynabus.

Each Cyclone processor can support one or two I/O subsystems consisting of two I/O channels each. Each channel supports up to 32 I/O controllers, which work with a variety of devices or communications lines. Each I/O subsystem can transmit data at a burst rate of 5MB/sec, for an aggregate bandwidth of 10MB/sec. per processor. The Cyclone also uses Direct Memory Access (DMA) I/O channels, so no processor cycles are required to move data from a channel to memory or vice versa.

Hardware Fault Tolerance The NonStop Cyclone employs redundant hardware throughout its subsystems to reduce the possibility of a single-point of failure. Using multiple processors, redundant busses and I/O controllers, mirrored disks, and multiple power supplies, Tandem has ensured that failure of a single major component will not bring the system down. Components that do fail are guaranteed to "fail-fast", terminate operation immediately after detection of an error, to reduce the possibility of error propagation. Components are also designed to be replaced on-line by on-site personnel, increasing availability while reducing the chance for a fatal second failure.

The availability of multiple processors and memories, on-line diagnostics, and various coding techniques allows continuous operation in the event of a processor or memory failure. Hardware self-checking logic is used to detect when a processor has failed. When a failed processor is detected, the GUARDIAN90 operating system distributes the processor's applications to the remaining processors, transparently, where they continue to execute unabated. Main memory, located on each processor card, is protected by coding techniques, embedded in hardware, which corrects single-bit errors and detects double-bit errors. Control store and cache memory arrays automatically recover from transient failures, but a spare RAM chip is available for use in case of a permanent failure.

Reliable interprocessor communication is handled by Tandem's *Dynabus* and *Dynabus+*. Dynabus is composed of two independent, 20MB/sec. busses, capable of simultaneous operation, which are used to connect processors within a section, as shown in Figure 1. Dynabus+ uses four unidirectional optic fibers, resulting in redundant, bi-directional links, which are used to connect sections together, as shown in Figure 2. In addition to the reliability gained by redundant links, Dynabus+ has the additional advantage of fiber-optic cabling, which reduces the risk of errors caused by electromagnetic interference.

The extra I/O subsystems and channels are to improve I/O performance and connectivity. Each Cyclone processor has two I/O subsystems, each with two I/O channels, dual-ported controllers, and mirrored disks. Each I/O subsystem has redundant channels, and each channel is capable of solely supporting an I/O controller and its attached device. The I/O controllers are dual-ported, with each port connected to two channels from separate I/O subsystems within different processors, as shown in Figure 1. This thorough level of redundancy ensures that if a processor, I/O subsystem, or channel fails, an alternate is readily available to take over access of the supported device. When a failure occurs, the operating system is designed to transfer access away from the failed component to its alternate, transparently, without losing any data that may be in transit. One of the most critical devices, the Cyclone's disks employ similar techniques for protection of their data. Each logical disk is actually a physical pair of disks, mirrored to ensure that there is a redundant copy of stored data. Each physical disk is dual-ported, with each port connected to separate I/O controllers. Disks are replaceable on-line, with new drives configured and updated automatically by the system.

The design of the Cyclone system incorporates several other techniques that improve reliability and availability, including high-speed impingement air cooling, integrated backplanes, and field-replaceable units. Excessive heat and radical temperature changes greatly reduce the life expectancy of electronic components. To reduce the chance of such failures, the Cyclone employs impingement cooling, which blows air directly down onto components to eliminate overheating and maintain a consistent temperature throughout the platform. Excessive internal cabling is also a problem, since cables restrict air flow and hamper maintenance. Thus, the Cyclone uses highly-integrated backplanes which greatly reduce the amount of internal cabling. The backplanes also enable the implementation of components onto individual, on-site replaceable, hot-pluggable cards, which Tandem calls *field-replaceable units* (FRUs). Processors, I/O controllers, fans, and power supplies are all implemented as FRUs. This is made possible through the operating system's support of transparent switching and reconfiguration of redundant resources.

Software Fault Tolerance The NonStop Cyclone system runs Tandem's proprietary GUARDIAN90 operating system, Lee and Iyer (1993). GUARDIAN90 augments the fault-tolerant hardware by providing error detection and recovery from hardware and software failures using consistency checks, special protocols, redundant data, and backup processes.

The GUARDIAN90 operating system detects failures using kernel and non-kernel consistency checks, and the "I'm alive" message protocol. Consistency checks, such as the comparison of redundant data structures and kernel-level assertions, are designed to catch local software and hardware failures, Lee and Iyer (1993). In the event that an error is detected, the operating system is designed to immediately halt the processor to ensure data integrity. Other processors can detect a halted processor using the "I'm alive" protocol. The "I'm alive" protocol keeps each processor in the system up-to-date about the status of every other processor, by exchanging messages. Every second, each processor in the system sends a special "I'm alive" message over every bus to all other processors in the system. Every two seconds, each processor checks to see if it received a message from every other processor. If a message is not received from a particular processor during that period, it is assumed to be down.

For critical applications, the GUARDIAN90 operating system creates and maintains a duplicate, backup process on a separate processor. If the primary process fails or falls victim to a hardware failure, the GUARDIAN90 operating system automatically switches to the backup. In order to reduce overhead, the backup process is implemented as a passive replica of the primary process, becoming active only when the primary process fails. Consistency between the primary and backup processes is maintained through periodic checkpointing messages. Prior to performing a critical function, like changing information in a database through an output commit, the primary process sends a checkpoint message to its backup. Contained within the message is data and status information necessary to complete the critical operation if the primary process fails. If it should happen that the backup process cannot be activated, the GUARDIAN90 operating system halts the entire processor.

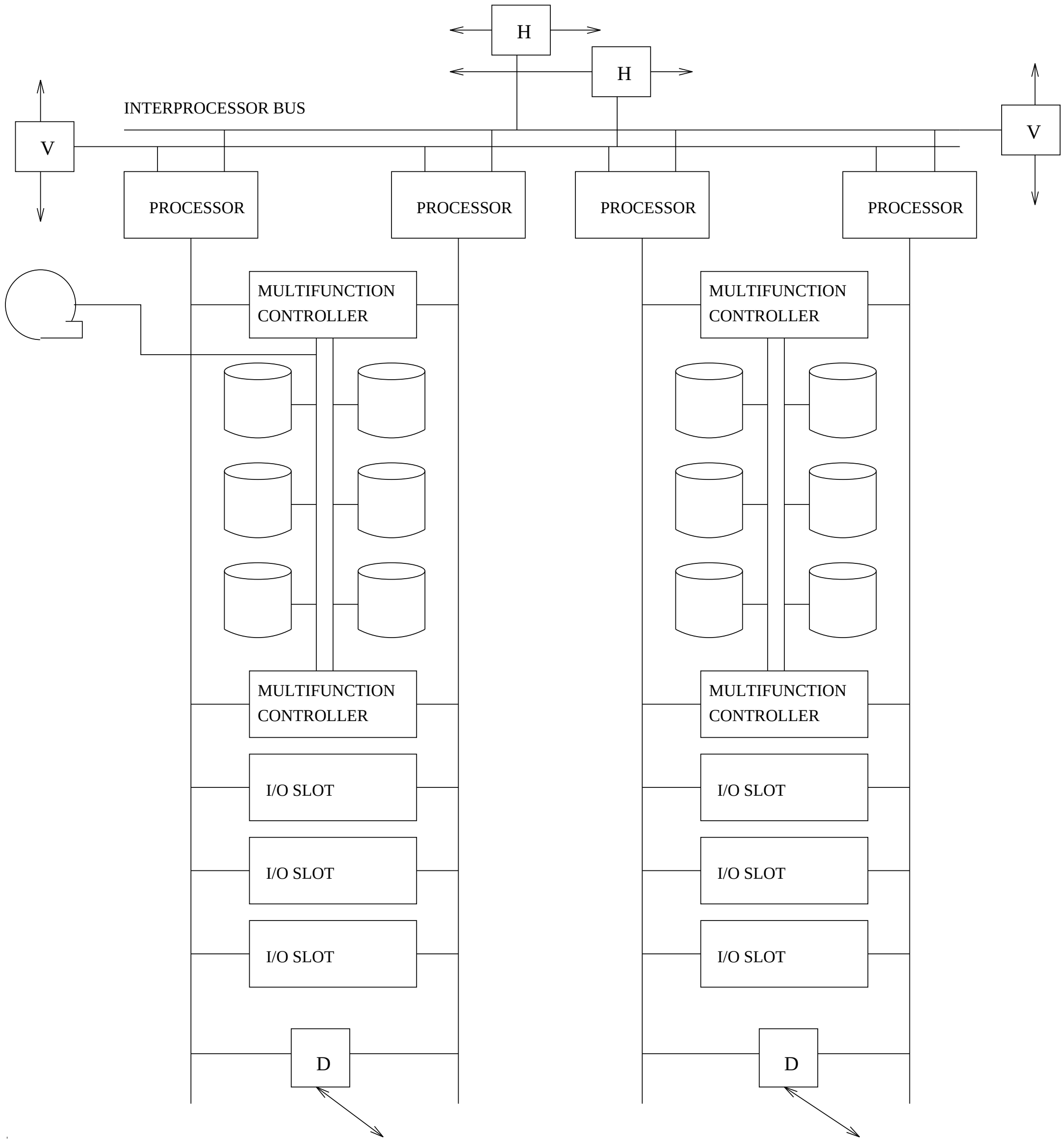

Figure 3: Tandem Himalaya K10000 System

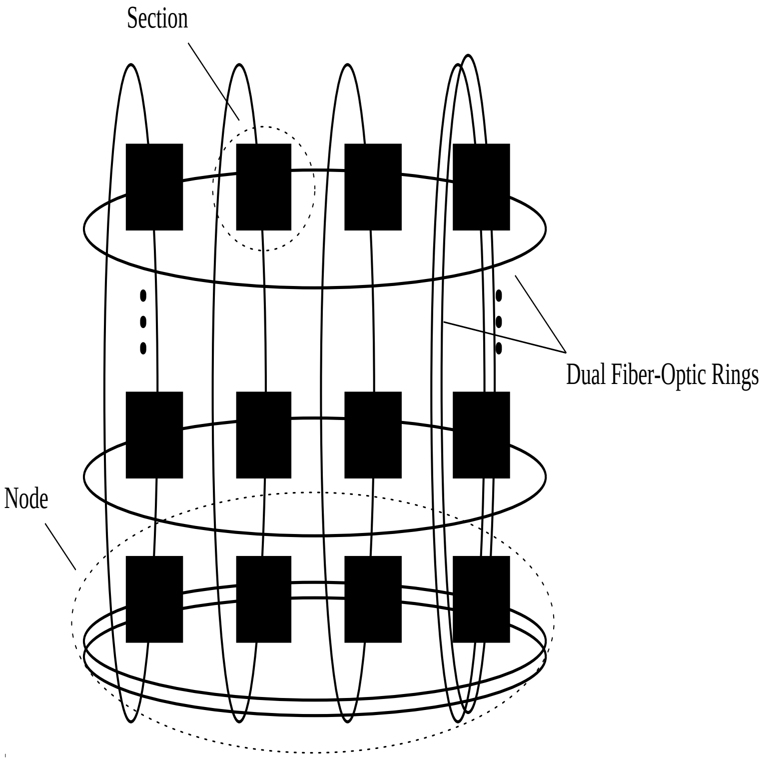

Figure 4: Himalaya K10000's Inter-Section Network

2.1.3 Himalaya K10000

The Himalaya K10000, shown in Figure 3, is a loosely-coupled, massively parallel multicomputer composed of fault-tolerant, multiprocessor sections connected in a 3-dimensional torus topology, Horst (1994). Introduced in 1993, the K10000 was designed to satisfy the ever increasing performance, reliability, and availability requirements that large commercial applications demand. The K10000's inter-section network, shown in Figure 4, gives customers the option of purchasing a system with as little as 4 processors, one full section, or as many as 4,080 processors. The loosely-coupled, sectioned architecture also allows customers to easily upgrade the system as needed. The K10000 is similar in design to the NonStop Cyclone, but their processor architectures and section interconnections differ sharply. However, like the majority of Tandem's systems, the K10000 was built with the overall objectives of eliminating single points of failure and preventing the propagation of errors; to this end, almost every component is replicated and fail-fast. Its most notable features are its CPU architecture and 3-dimensional torus network, called TorusNet.

The K10000 builds on the framework established in the Cyclone family of systems. However, unlike the Cyclone's single superscalar CISC processor, the K10000 system uses two cross-coupled, 150MHz R4400 RISC processors. The K10000's processors reside on a single *CPU* board, and are viewed as a single "logical" processor. Processor *cross-coupling* is the practice of forcing two processors into executing the same instruction and then comparing their results. In the K10000, if a fault is detected by the comparator logic, then the CPU board is immediately shut down. Thus, error latency and propagation are reduced at the cost of twice the number of processors. The K10000 CPUs are also distinguished by their large complements of cache, main memory, and I/O channels. Each CPU has 32KB of primary cache, 4MB of secondary cache, and up to 256MB of main memory. In addition, each CPU supports an I/O subsystem with four independent I/O channels.

TorusNet, Tandem's proprietary, reliable, fiber-optic network, was designed for connecting sections within the K10000. As shown in Figure 3, each section contains redundant vertical and horizontal TorusNet, fiber-optic controllers (labelled V and H respectively), connected directly to the section's interprocessor bus. The horizontal controllers are capable of connecting four sections in a ring, called a *node*. The vertical controllers are capable of connecting up to 14 nodes in a 2-dimensional torus, called a *domain*. A third pair of controllers, called the *depth* controllers (labelled D), are standard LAN or WAN controllers that are capable of connecting 16 domains together for a total of 4,080 processors. The redundant controllers and links provide protection from any single controller, link, or power supply failure from disconnecting the system and disrupting service. Furthermore, the fiber-optic cabling provides sufficient bandwidth and noise resistance for interconnecting large numbers of processors.

2.2 CASE STUDY 2: Stratus XA/R Series 300 Systems

2.2.1 Introduction

Stratus Computer, Inc. started producing fault-tolerant computing systems in 1981, and is acknowledged as being the first to provide purely hardware-based fault-tolerance, Webber and Beirne (1991). Since that time, its effective "pair-and-spare" approach has made Stratus one of the primary contenders in the fault-tolerant computing arena. In the Stratus approach, system boards are duplicated. On each board there are identical pairs of circuits which execute the same operations and continually compare results. If a comparison fails, the faulty board is immediately isolated from the system while its duplicate continues to operate unabated. The XA/R Series 300 family of systems is one of the most recent offerings from Stratus, Stratus $(1992)_a$. The Series 300 systems, designed for the lucrative on-line transaction processing market, utilize RISC technology to provide high levels of performance in a scalable MIMD, tightly-coupled, shared memory, shared bus, symmetric multiprocessing environment, Hwang and Briggs (1984). Two operating systems are offered with the Series 300 systems, the Unix-based FTX system or the proprietary VOS system, Stratus $(1992)_b$.

2.2.2 System Architecture

The XA/R Series 300 systems are composed of one or more *modules* connected via Stratus's redundant, intermodule bus called StrataLINK. Each module, shown in Figure 5, is independent and complete, containing duplexed processor, memory, and I/O boards, a battery backup system and redundant power supplies, and supported devices. System boards communicate across a proprietary, fault-tolerant bus, integrated into the backplane of the module's cabinet. On top of the hardware, Stratus provides users with their choice of operating systems: the Unix SVR4 complaint FTX or Stratus's proprietary VOS.

The Series 300 processors are based on the Intel i860 family of 64-bit RISC microprocessors. Depending on the model, a single module can have from one to six "logical" processors containing redundant sets of either 32MHz i860XR or 48MHz i860XP physical microprocessors. External cache is included on each processor board to augment the i860's own internal cache.

Main memory is shared, and can be installed in increments of 32MB or 128MB.

All I/O in the Series 300 modules is handled by redundant, dedicated I/O processing subsystems. Each I/O subsystem is composed of a duplexed pair of boards which host a group of up to 14 I/O adapters across a proprietary fault-tolerant bus. Each I/O board contains a dedicated Motorola M68030 microprocessor, 4MB of RAM, and a DMA engine. Each proprietary I/O adapter has its own microprocessor, which is responsible for diagnosis, reconfiguration, and recovery. Available adapters include those for networking (IEEE

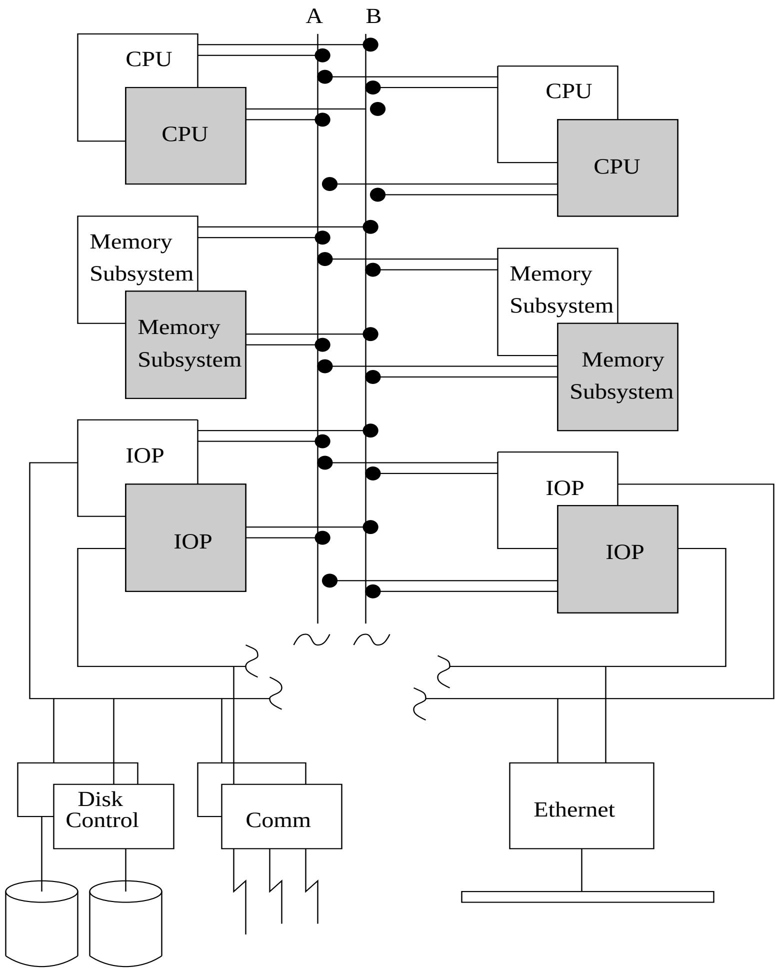

Figure 5: Stratus XA/R Series 300 Module

802.3, IEEE 802.5, and X.21), asynchronous terminals, universal serial communications (RS232, RS422, RS423, and V.35), SCSI disks, tapes, printers, StrataLINK, and Stratus's Remote Service Network.

StrataLINK is Stratus's proprietary intermodule bus. It is composed of duplexed boards driving redundant 1.4MB/sec coaxial links. The VOS operating system uses StrataLINK to present the collection of interconnected modules as a unified system. In addition, VOS supports message passing between processes across StrataLINK.

A fully configured module has six logical processors, 512MB of main memory, 105GB of disk storage, and 8 I/O subsystems capable of supporting 112 I/O adapters. Using StrataLINK, up to 32 systems may be loosely interconnected as one large multiprocessing system, for a total of up to 192 processors.

2.2.3 The Pair-and-Spare Approach

Stratus's architecture centers around the pair-and-spare approach to fault-tolerance. In the general *pair-and-spare* approach, a system is composed of a pair of identical primary components and any number of spare components. All of the components receive the same inputs and feed their outputs into a switch. The switch selects the outputs of two of the components and feeds them into a comparator. Initially, the switch chooses the outputs from the primary pair of components, while the spares are held on standby. If the comparator detects a disagreement in the components' outputs, the faulty component is identified and taken off-line. A spare is then activated to take its place.

Advantages to the pair-and-spare approach are that errors are detected immediately, with high probability, and prevented from propagating beyond the system's boundary. However, a problem with this approach is how to determine which component failed. If diagnostics are used, then there is a disruption in services until the failed component is convicted. To overcome this problem, Stratus permanently pairs components together such that if an error is detected, both components are taken off-line. The spare components are not held on standby, but operate in synchrony with the primary components, driving their outputs onto the bus. Thus, if a comparison fails, the suspect components may be quietly taken off-line while their "partner" components continue to operate unabated.

General System Board The XA/R Series 300 system boards are self-checking and fail-fast. Each primary system component is duplicated onto separate halves of a single system board, as shown in Figure 6. Both halves, labelled A and B in Figure 6, are designed to operate independently, but in synchronous lockstep. Synchronization logic ensures that the boards perform the exact same operations on the exact same input every clock cycle. During each cycle, on-board, redundant comparator logic compares the outputs from the two halves. If there is a mismatch, the comparators immediately disable output to the bus, take the

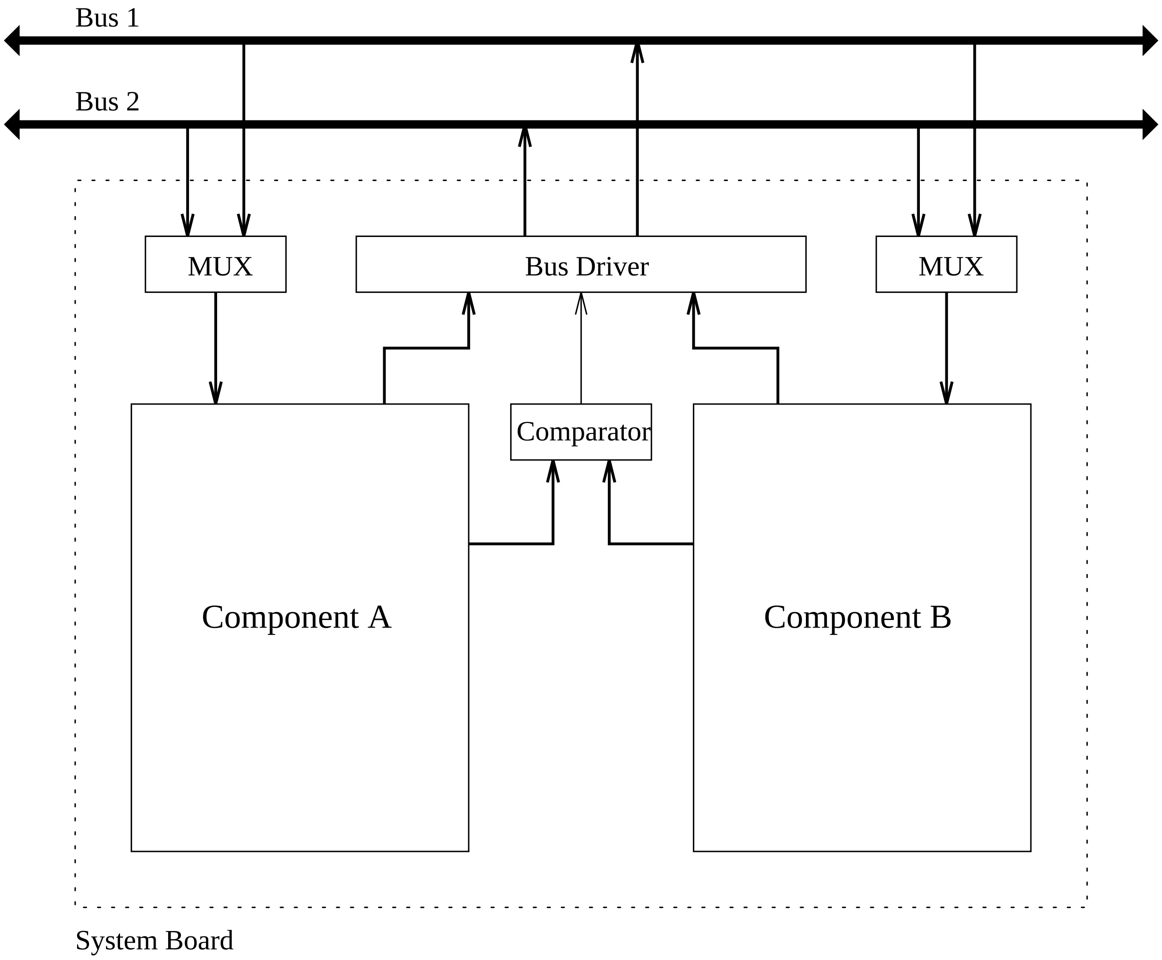

Figure 6: Stratus System Board

board off-line, raise a flag, and light a LED on the front of the board. Otherwise, the comparators enable the bus interface and allow the board to drive its output onto the bus. Since the bus interface is beyond the comparator logic, its failure cannot be detected by the board. Thus, additional logic, described later, is employed to detect a faulty bus interface. In order to reduce the possibility of a power supply failure, each system board also contains its own power regulator. Incorporating power regulation into each system board reduces the complexity of the critical power supplies, increasing overall system reliability.

Almost every system board in an XA/R Series 300 system has an identical partner board. Partner boards may either operate in synchronous lockstep or as logical pairs. Partner boards that operate in lockstep perform identical operations each clock cycle. Thus, at any one time, four circuits may place data onto the bus. To compensate, the outputs of the circuits are logically ORed by the bus. The processor and memory boards operate in synchronous lockstep. Partner boards that operate as logical pairs do not execute in lockstep, but rely on the operating system for synchronization. Unlike the boards that operate in synchronous lockstep, the failure of a logically paired board may affect its partner by increasing its workload. Disk, network, and communications boards operate as logical pairs.

Processors, Memory, and I/O The processors and memory control logic are self-checking, and operate in synchronous lockstep with their partner boards, as described previously. However, additional techniques are employed to protect the memory arrays. Each memory array is duplexed, with one array located on each partner board, and protected by error correcting codes. Additional protection is provided by "scrubber" logic, which continually and methodically checks memory locations for latent errors, correcting them as it goes.

Each I/O processing subsystem is composed of a duplexed pair of self-checking system boards. Attached to each subsystem is a proprietary fault-tolerant bus, which hosts logically paired I/O adapters. The bus is composed of redundant, independent busses, each protected by parity. The I/O adapters also contain special logic to monitor the bus for errors.

Main Bus and Power Subsystem The main bus, shown in Figure 5, is actually two independent busses, each capable of handling the full load. The busses are integrated into the backplane of the XA/R Series 300 cabinet, and are socketed for connection to each board. Both busses are protected by parity, which is monitored by detection logic located on the memory boards. The detection logic continually monitors the parity signals that cross the redundant busses. If an error is detected on one of the busses, the detection logic signals all boards to cease use of the suspect bus. However, every system board is connected to both busses, so the loss of one bus does not disable the module.

Each Series 300 module has redundant power supplies and battery backups to protect against power disturbances or supply failures. The redundant power supplies drive independent rails which supply either all the even or odd slots in the cabinet's backplane. Therefore, partner boards, which are inserted adjacent to one another in the cabinet, are powered by separate supplies. This practice ensures that the module can continue to operate in the presence of a power supply failure. To protect from brown-outs and black-outs, battery backups are available to power the module long enough for a proper shutdown. When a power failure is detected, the batteries are brought on-line and the mechanical subsystems are shutdown, but the module continues to operate. If the failure continues beyond a preset time threshold, which represents the dividing line between a brown-out and a black-out, the module goes into hibernation. The volatile state of each of the system boards is first saved into memory, and then all remaining power is dedicated to supplying the memory boards. The module can remain in this state of hibernation for a long period of time, preventing the loss of data due to extended power failure.

2.2.4 System Software

The Series 300 systems can run the VOS or FTX operating systems. VOS is Stratus's proprietary operating system, tuned for the Series 300 systems. FTX is Stratus's implementation of UNIX SVR4, with extensions for fault-tolerance. Both provide support for multiple users, symmetric multiprocessing, batch and interactive sessions, and online hardware replacement. The nature of the Stratus architecture virtually eliminates the need for software-based fault-tolerant techniques such as checkpointing, process shadowing, and interprocess "I'm alive" messaging. However, two proprietary software packages perform critical roles in maintaining continuous processing on the XA/R Series 300 systems: the Maintenance and Diagnostics software, and the Remote Service Network.

The Maintenance and Diagnostics *MD* software is responsible for tracking, diagnosing, and reporting hardware failures, system board reintegration, and configuration. If a board fails, it is up to the MD software to determine the cause of the failure and to take appropriate action. Initially, the MD software computes the board's Mean Time Between Failure *MTBF* based on its previous failures. If the calculated MTBF is lower than a preset low-water mark, then the board is assumed to be intermittently faulty and is marked for replacement. Otherwise, the MD software prompts the board to perform its self tests. If the tests are successful, the failure is classified as transient and the board is brought back into service. However, if the tests fail, the board is known to be permanently faulty and is marked for replacement.

The Remote Service Network *RSN* software is used to communicate Stratus's Customer Assistance Center *CAC*, in the event the MD software determines that a component needs to be replaced. As soon as the decision is made, the MD software notifies the RSN software, which immediately calls the CAC and reports the failure. The RSN software is then used by the CAC staff to remotely confirm the report. Once the failure has been confirmed, the CAC dispatches a replacement part via express shipping. Very often, the system administrator does not know that a part has failed until its replacement arrives. The quick response provided by the RSN software is crucial to maintaining availability, since it is possible that a second board failure could cripple the system.

2.3 CASE STUDY 3: Sequoia Series 400 System

2.3.1 Introduction

Sequoia Systems, Inc. began shipping fault-tolerant systems in 1987. Throughout its existence, Sequoia has focused its attention on businesses which are critically dependent on on-line transaction processing. The Series 400 systems, first introduced in 1991, use multiple modules of duplexed, self-checking, "off the shelf" components in a tightly-coupled multiprocessing architecture to provide flexible, scalable, and highly-available processing, Sequoia $(1992)_a$.

The Series 400 system uses multiple self-checking processor, memory, and I/O elements running in parallel to achieve high performance and high availability. *Elements* typically contain a pair of cross-coupled components which perform operations simultaneously, comparing results to detect errors. The Series 400 architecture, shown in Figure 7, can be classified as a MIMD, tightly-coupled, shared memory, shared bus, symmetric multiprocessor, Hwang and Briggs (1984). All processors run Sequoia's proprietary, UNIX-based, fault-tolerant operating system called *TOPIX*. TOPIX is capable of executing applications designed for AT&T and Berkeley versions of UNIX, providing them with automatic and transparent fault protection. A fully configured Series 400 system consists of multiple system and peripheral cabinets containing 32 processor elements, 16 memory elements of 256MB each (totalling 4GB), and 30 I/O elements, Sequoia $(1992)_b$.

2.3.2 System Architecture

Each Processor Element, shown in Figure 8, uses dual Motorola M68040 microprocessors. The M68040 is a 32-bit, 33 MHz microprocessor containing an on-chip floating point unit, separate 4KB instruction and data caches, and separate instruction and data memory management units. The M68040 can address a virtual memory address space of up to 4GB. The dual processors are augmented with; a second-level 4MB, 4-way set-associative, non-write-through cache; a PROM containing system diagnostic and fault recovery code, and a bootstrap loader; and duplexed comparator logic.

Each Memory Element, shown in Figure 9, contains up to 256MB of memory organized into one shared array. This array is divided, using independent addressing logic, into two virtual arrays which shadow each other. The addressing logic is designed such that the most significant bit of an address selects one

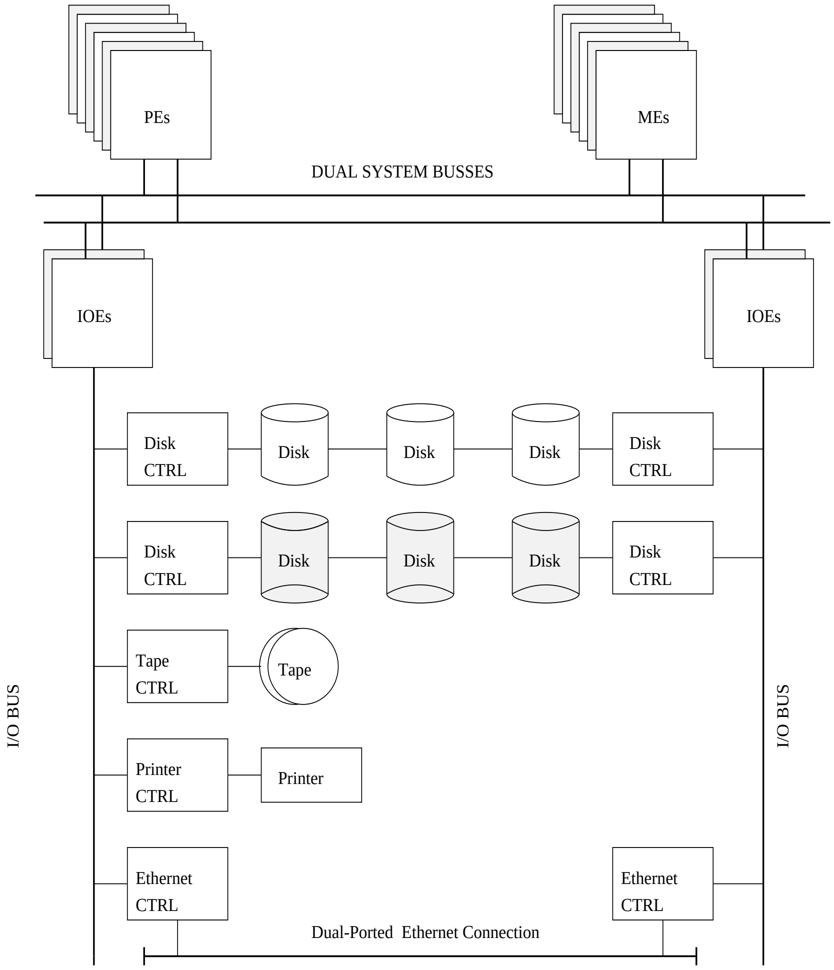

Figure 7: Components of a Sequoia System

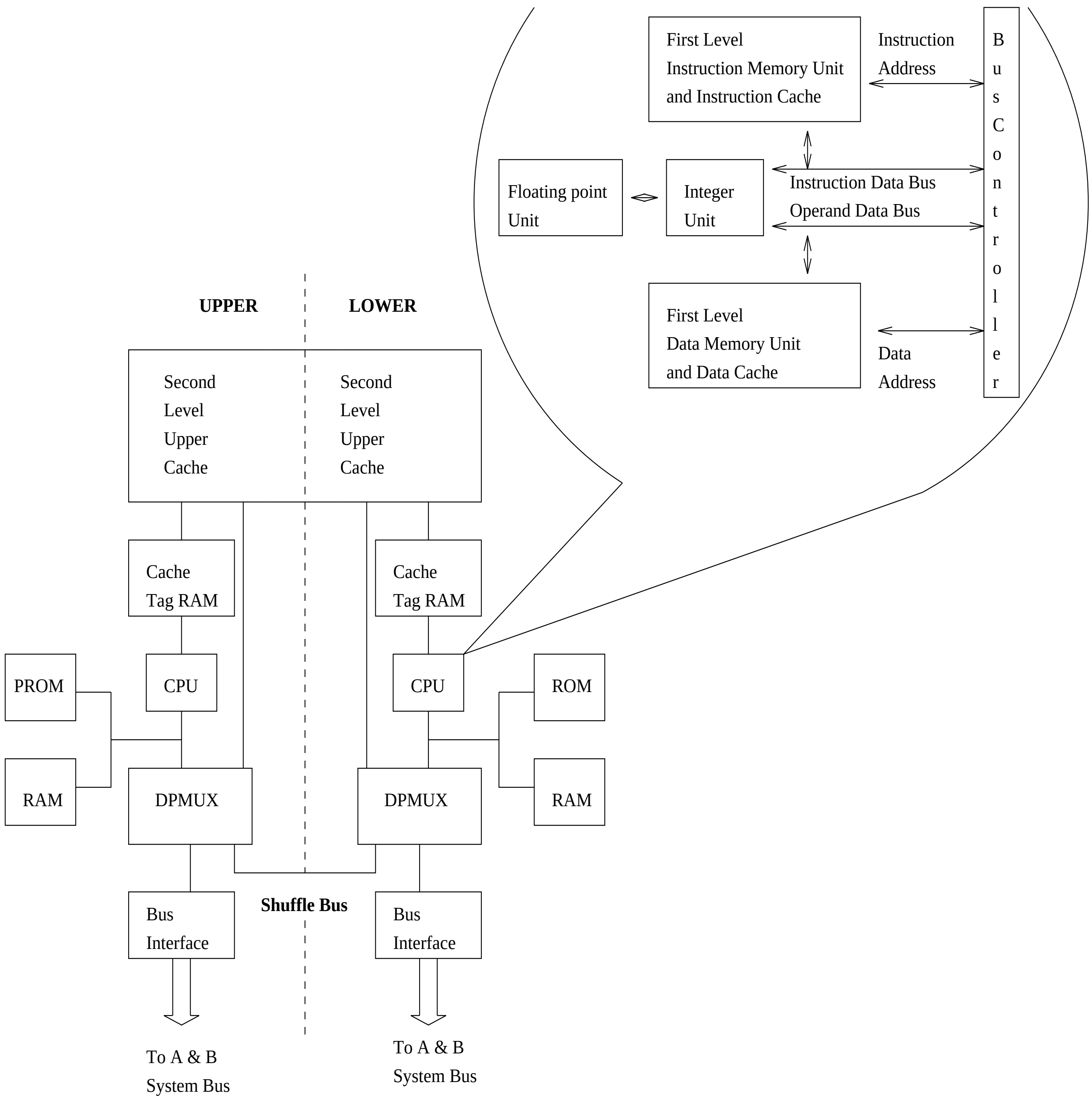

Figure 8: Functional Block Diagram of a Processing Element

of the virtual arrays, and the remaining bits form the offset address into that array. The base addresses of each of the virtual arrays is stored in physical address registers that can be dynamically reconfigured by the operating system. Each Memory Element also contains duplexed error detection and memory access logic that independently address the virtual arrays. In addition, hardware test-and-set locks are included, which are used by the operating system to control simultaneous access to shared resources. Every 16MB of memory has 1,024 test-and-set locks.

Each I/O Element consists of a duplexed Bus Adapter and a duplexed Multibus Adapter. The Bus Adapter is connected to the local memory bus and resides within the system cabinet. Its primary function is to buffer data passing between the Processor Elements in the system cabinet, and the Multibus Adapter in the peripheral cabinet. The Bus Adapter contains four channels, each having 4KB of buffer space, that are independently accessible by the Multibus Adapter and Processor Elements. These four channels form two 8KB fault-tolerant channels, which the Bus Adapter uses to double-buffer data; while a Processor Element is accessing one channel buffer, the Multibus Adapter accesses the other, and then the channels are switched. This activity results in better I/O performance. The Multibus Adapter, housed in the peripheral cabinet, contains its own duplexed Motorola M68010 microprocessors. These processors run a special operating system that is tuned for the movement of data between memory and the device controllers, located on the adapter's IEEE Multibus. The Bus Adapter and Multibus Adapter are connected by the External Bus, which has a maximum aggregate bandwidth of 5MB per second and allows 400 feet of separation between the two cabinets.

The main duplexed bus used to connect processors, memory, and I/O elements is composed of two 40-bit, 10 MHz busses. These busses are capable of an aggregate peak throughput of 80MB/sec. During normal operation, the two busses operate in parallel. However, if a fault is detected somewhere in the system, only one bus, called the "Executive Bus," transfers data. A processor chooses the "Executive Bus" in advance, using separate lines dedicated for this purpose.

2.3.3 Hardware Fault Tolerance

The Series 400 system implements various fault detection and isolation mechanisms throughout its hardware design. Such mechanisms include the use of error-control codes, the comparison of redundant operations, and protocol monitoring. The duplexed element boards usually have a combination of shared circuitry and redundant circuitry. Shared circuitry is usually protected by error-control codes, whereas redundant circuitry performs continuous comparison of redundant operations. Most elements contain both shared and redundant circuitry. All elements within the Series 400 system use one or more of these mechanisms. If a mechanism detects a failed element, the element is electrically

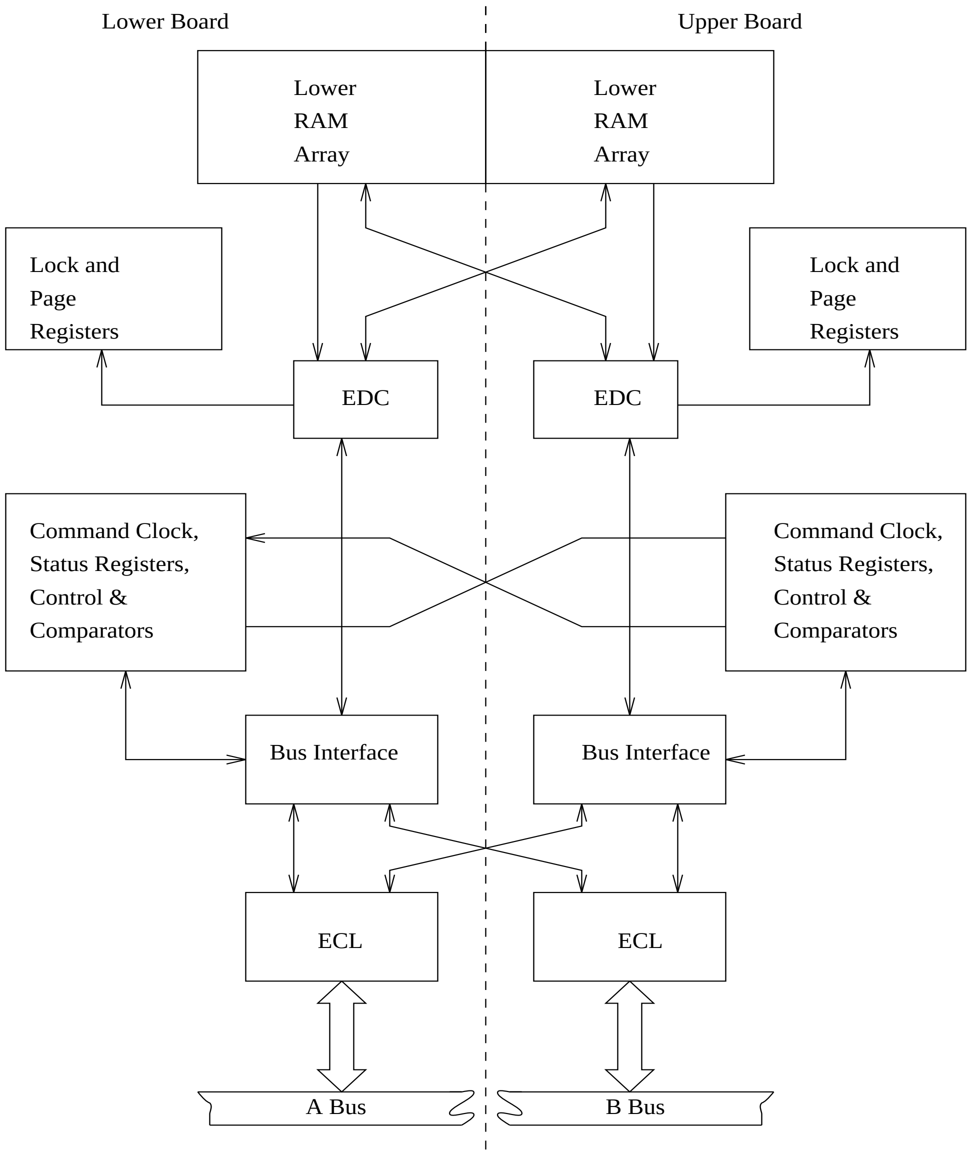

Figure 9: Functional Block Diagram of a Memory Element

isolated from the system and flagged for repair.

The redundant circuitry in the Processor Elements, such as the processors and comparators, detect faults by continuously comparing results. The shared circuitry, such as the local cache, uses parity for fault detection. Additionally, each Processor Element maintains a 128-byte status block in the main memory that describes its operational status. This block is updated every 100ms and periodically cross-checked by another Processor Element to ensure that it is operating correctly. The detection of any error results in the automatic and immediate electrical isolation and flagging of the element from the system. Thus, the elements are "fail-fast."

The memory arrays on the Memory Elements are protected with a double error detection *DED* and single error correction *SEC* code. This SEC/DED code requires an additional 8 bits for each 32-bit data word. When a read is issued to a Memory Element, the Error Detection and Correction *EDC* logic flags single-bit, double-bit, or multiple-bit errors in the element's status registers. If the error is not single-bit, the element removes itself from the physical address space. The EDC logic is protected through duplication and comparison. Finally, to ensure data integrity in the presence of Memory Element failure, modifiable data is written to two separate Memory Elements.

The components of an I/O Element, the Bus Adapter and Multibus Adapter, are protected separately. However, both contain duplexed boards with shared and redundant circuitry incorporating parity checking and comparison of redundant operations. All peripherals are dual-ported and connected to different I/O Elements in order to tolerate the failure of a single I/O Element. In addition, file mirroring, writing files to two separate disk drives, is used to enable recovery from hard disk failure.

Redundant power supplies and battery backup units reduce downtime due to power supply failures or blackouts. The main cabinet's power supply also provides filtering, conditioning, and regulation to reduce the possibility of dirty power damaging components.

2.3.4 Software Fault Tolerance

The Series 400 systems run Sequoia's TOPIX operating system. TOPIX is a high-level, UNIX-based operating system that is compatible with both the AT&T System V.3 and Berkeley, as well as X/Open XPG2 compliant. TOPIX provides support for symmetric multiprocessing, dynamic load balancing, test-and-set locking, trapping of software errors, fault detection, and fault recovery.

Each Processor Element in the Series 400 runs the TOPIX kernel from a single image in shared memory. In order to allow concurrent access to kernel data, access must be synchronized among the Processor Elements. Thus, the operating system uses the hardware test-and-set locks, located on the Memory Elements, to coordinate access to shared resources and maintain consistency of the kernel data. However, in order to recover from a failure, the operating system must maintain a consistent image of the entire system. Kernel support for fault recovery must ensure that a process can be recovered without the loss of data or program continuity, regardless of when a fault occurs. Such fault recovery is dependent on the kernel's ability to always maintain memory resident processes in an internally consistent state.

The TOPIX kernel maintains a consistently recoverable state through periodic "checkpointing." The kernel checkpoints the system by flushing the non-writethrough caches and registers of the Processor Elements to the main memory. During execution, dirty pages are stored in the local non-writethrough cache of each Processor Element. At regular intervals, or when a Processor Element is holding a lock it is no longer using, the kernel flushes the caches and registers to the main memory, effectively saving a "snapshot" of the state of the system. If a failure occurs, the kernel rolls back to the previous checkpoint and re-executes. The kernel protects against failures occurring during the flush operation by flushing data to the backup Memory Element first. Thus, if a failure occurs during flushing, the kernel can rollback to the checkpoint in the uncorrupted primary element. If the component that failed was a Processor Element, the kernel divides and relocates the failed element's workload to the remaining, operational Processor Elements. This relocation capability can also be used for dynamic load balancing, improving performance.

The TOPIX operating system incorporates built in self-checking in the form of executable assertions. When an instruction that is covered by executable assertions is performed, the assertions which cover it are compared with the results of the instruction. If the results violate the assertions, the operating system initiates fault recovery. Additional features of TOPIX include enhanced security protection of sensitive database information, support for remote diagnosis, on-line expansion, and automatic recovery and reintegration of modules.

Table 3 presents a comparison of the fault-tolerant systems discussed so far.

Table 3: Comparison of Fault Tolerant Systems

	NonStop	Stratus	Sequoia
Architecture	Loosely coupled message passing multiprocessor	Tightly coupled shared memory multiprocessor	Tightly coupled shared memory multiprocessor
Operating System	NonStop Kernel/Guardian	VOS, FTX (Unix based)	TOPIX (Unix based)
Fault tolerance scheme	Self checking modules with checkpointing among duplicate processes	Pair-and-spare hardware scheme	Self checking modules with a consistent image of process in memory
Fault tolerance in memory	1 bit error correction and 2 bit error detection of data. Address error detection.	Single bit error correction	Single error correction and double error detection, ECC logic is self checking, Duplex memory elements
Fault tolerance in bus	Checksum on bus transactions		Error detection mechanism
Application domain	OLTP	OLTP and general purpose	OLTP
Serviceability	Online service and maintenance	Online service and maintenance	Online service and maintenance
Overhead	Only one process is active which runs on duplex processors. A dormant copy of the process resides in another PE.	2 duplex PEs execute same process.	A duplex PE executes a process with a consistent image of the process in main memory.

References

[Harper91] R. E. Harper and J. H. Lala, "Fault-tolerant parallel processor," *Journal of Guidance, Control, and Dynamics*, vol. 14, no. 3, pages 554-563, 1991.

[Horst90] R. Horst, R. Harris, and R. Jardine, "Multiple Instruction Issue in the NonStop Cyclone Processor," *Proc. 17th Int. Symp. on Computer Architecture*, May 1990.

[Horst94] R. Horst, "Massively Parallel Systems You Can Trust," *COMPCON94 Digest of Papers*, San Francisco, Feb. 1994.

[Hwang84] K. Hwang and F. A. Briggs, *Computer Architecture and Parallel Processing*, McGraw-Hill, 1984.

[Lee93] I. Lee and R. K. Iyer, "Faults, Symptoms, and Software Fault Tolerance in the Tandem GUARDIAN 90 Operating System," *Proc. 23rd Int. Conf. on Fault-Tolerant Computing*, pp. 20-29, 1993.

[Sequoia92a] Sequoia Systems Incorporated, *Series 400 Technical Summary*, June, 1992.

[Sequoia92b] Sequoia Systems Incorporated, "The Series 400," Sequoia Product Report, 1992.

[Stratus92a] Stratus Computer Incorporated, "XA/R Series 300," Stratus Product Report, 1992.

[Stratus92b] Stratus Computer Incorporated, "Stratus FTX," Stratus Product Report, 1992.

[Tandem90] Tandem Computers Incorporated, "NonStop Cyclone," Tandem Product Report, 1990.

[Tandem93] Tandem Computers Incorporated, "NonStop Cyclone/R," Tandem Product Report, 1993.

[Webber91] S. Webber and J. Beirne, "The Stratus Architecture ," *Proc. 21 Intl. Symp. Fault-Tolerant Comp.*, pp. 79-85, 1991.

[Zorpette90] G. Zorpette, "The power of parallelism," *IEEE Spectrum*, pages 28-33, Sept. 1990.