Fujitsu PRIMEQUEST: 32-way SMP Open Servers with Powerful Reliability Features

Toshiyuki Shimizu, Yasuhide Shibata, Haruhiko Ueno, Takumi Takeno, Shinya Kato,
Seishi Okada, Nobuo Uchida, Hiroyuki Adachi, Hideki Maeda, Takato Noda, Akira Kabemoto
Fujitsu Limited
{t.shimizu, yshibata, ueno.haruhiko, takumi, kato_shin,
s-okada, uchida.nobuo, adachih, maeda.hideki, t.noda, kabemoto.akira}@jp.fujitsu.com

Abstract

The PRIMEQUEST is an open architecture 32-way true-SMP server with powerful reliability features, which was designed and is manufactured by Fujitsu Limited. The PRIMEQUEST server was designed with a focus on high reliability and high performance. It incorporates processors of the Intel® Itanium® processor family (IPF).

We utilized rapidly improving silicon device technologies, specifically, high-density and high-speed technologies, in this system to obtain not only high performance but also high reliability.

High-frequency data transfer technology, 90-nm silicon technology, and packaging technology providing high densities and optimized cooling were utilized in order to create a high performance system. We developed a new cache-state snooping method. With this method, both high-frequency operation and low-latency memory access even in a large-scale SMP are achieved. And flexible I/O functions are also realized. A system with 32 CPUs and 667-MHz DDR2 memory running at a system bus frequency of 1.3 GHz provides a memory bandwidth of 170 GB/s throughout the system.

For high reliability, inter-chip and intra-chip connections are fully duplicated, and many checkers are properly integrated. These checking mechanisms ensure continuous operation by detecting faults and then disconnecting faulty parts and failed paths. We implemented this duplication extensively in the machines of the PRIMEQUEST series; the technology is called dual synchronous system architecture (DSSA). Our estimate of the resulting availability in a clustered system is 99.999%.

1. Introduction

Computer systems and network systems have been getting more and more complex, and the dependency on these systems is growing. Accordingly, computer systems in many application areas must have greater reliability. Highly reliable systems have been realized with so-called “closed systems.” A closed system is designed to have dedicated hardware, operating systems, and applications[1]. By using these specially designed components, a system can be made very reliable compared with an ordinary system. However, not all applications can be ported to a closed system or redesigned for it. Users want to build and run their applications in a more distributed manner or expansive environment, such as provided by an open system.

The reliability of the Linux and Windows operating systems has been increased, and a lot of applications have been written for use under them. Direct and indirect use of applications running on open platforms is spreading in many critical areas. Reliable open system platforms are required for running these applications.

At the same time, system cost and performance should be controlled to get the proper balance. Additional costs for increased system reliability should be minimized as much as possible while the same level of functionality and processing performance is provided. Since the selection of available open standard components has a wide range of prices, functions, and performance, the system can be expected to provide the highest performance and the best cost-performance. Users want to run their applications on a system that not only has very high reliability but also is moderately or reasonably priced. Such computer systems should have configurable levels of reliability, from fair to excellent.

In such systems, improvements in transistors have resulted in higher performance and densities. The rate of this improvement is keeping pace with Moore's Law. However, neither the cost and density of LSI packages and connector pins nor development of complete systems, including power supplies and cooling systems, has followed the pace of advances in transistors. We had to consider the balance between these parameters in designing an architecture for commercial products.

Adequate cooling is the key to high-density and reliable systems. Keeping devices cool reduces the possibility of silicon device failures and provides a sufficient margin for operation.

In addition, our system has to maintain stable performance. Users expect the same performance from any two runs, with clean memory allocation as well as with fragmented memory. Directory-based ccNUMA systems have been widely studied, and many good systems have been installed and operated[8]. However, maintaining stable performance is still not an easy task. Uniform memory access latency and uniform, wide memory bandwidth throughout the system are very important features to run existing applications without any modification and obtain the expected performance.

For field operations, manageability is another important factor. System operation should not be stopped even if a part fails or becomes defective. Users want to modify the system configuration and replace faulty parts while the system is running.

The next section of this paper discusses the concept of the PRIMEQUEST design and our design decisions. The subsequent section presents the dual synchronous system architecture (DSSA), which enables high performance while keeping an open system. It also presents flexible I/O technology for flexible combination of CPUs and I/O units and partitioning without a performance penalty. A high-frequency signal transmission technology is also presented. Finally, section 5 concludes the paper.

2. Concept of the PRIMEQUEST Design

The PRIMEQUEST lineup consists of open industry standard servers that have the same level of reliability as mainframes, make use of open and de-facto standards such as with mass-produced processors, and run Linux and Windows. These servers are optimized for data centers, which require flexible partitioning and concurrent support of multiple OSs for both scale-up and scale-out system configurations.

This section explains our design concepts for a reliable open system. Openness, SMP, and flexible I/O were the top three priorities for our system during designing of the machine.

2.1 Openness with powerful reliability features

An open system is useful because it can quickly incorporate improvements in technology. It should not require modifications to applications, middleware, and operating systems. Even at the hardware level, we worked carefully to conform to de-facto standards and follow open standards. Regarding software, any Linux and Windows application should run on the PRIMEQUEST server without any special care required, in the same manner as on other open platforms, including those from other vendors. In addition, it should meet conditions for high reliability and high usability.

By isolating specific implementations and considering general mechanisms, we designed an architecture that can produce continuous improvements in performance and reliability features. We decided to make the PRIMEQUEST server fully duplicatable to meet requirements for high performance and high reliability. This decision was a very simple but essential one, since it meant the PRIMEQUEST server would be an open server. Complex mechanisms often require intervention from the operating system and applications. For system duplication with a reasonable overhead, we utilized state-of-the-art silicon device technologies. Additional functions and mechanisms are integrated in silicon devices, and reliability features are provided with numerous transistors. For example, to decrease the number of signal pins, transistors were used to expand the bandwidth per pin. Duplication logic and compare logic are also integrated in each LSI device to increase reliability. Using numerous transistors to meet requirements, the PRIMEQUEST server is built as a high-reliability and high-performance open SMP server.

We developed and implemented above duplex architecture and called it the dual synchronous system architecture (DSSA), which is transparent to operating system and applications.

2.2 Large SMP system with a constant memory bandwidth and low latency

We chose to build a large SMP system because it is capable of running many types of applications in parallel, is easy to manage, and has good memory utilization throughout the system. ccNUMA is commonly used to build large shared-memory systems, and it provides good performance[8]. We chose snoop-

based SMP, however, for stable performance, precise fault location, and failure recovery.

Affinity between memory and CPUs should be properly controlled for good stable performance, especially in a ccNUMA system. Linux is being optimized for non-uniform memory access latency[5]. We believe that affinity between memory and CPUs is not easy to control when memory becomes fragmented after a long period of operation. Moreover, regarding fault location and failure recovery, the complexity of managing errors in a ccNUMA system is high because it has a lot of intermediate states distributed in the system over a long period. What is worse, the states are not synchronized.

To obtain good performance from snoop-based SMP, a wide memory bandwidth and low memory access latency are the keys. We developed and implemented a high-speed signal transmission technology, which we call MTL. MTL can decrease the number of signal wires used while keeping the total system bus bandwidth equal to the sum of the front side bus bandwidths of the CPU. In addition, a new snooping design is employed to minimize memory access latency.

2.3 Flexible I/O: reconfigurable I/O and computation blocks

Input and output systems are different from computation blocks (CPUs and memory). These systems write results to storage devices and carry out external communication. These tasks cause side effects, and their interfaces are difficult to replace and reconnect. In contrast, data in computation blocks have rather short lifetimes, and the blocks are easier to re-execute. If I/O interfaces can be managed separately from computation blocks, maintenance of parts can be performed separately.

Systems integrating I/O interfaces and computation blocks in the same module require additional interfaces or a SAN switching network to provide a reconfiguration feature[6, 7]. Since the demand for computational power and I/O interfaces depends on the application, separate modules are preferable.

One system provides a reconfigurable I/O interface by incorporating a switching network[1]. However, the performance bottleneck caused by transmission through the network and the additional cost overhead should be removed. Another system attaches I/O interface boards to the system snoop bus[3]. This implementation is simple and symmetric. It also offers flexibility in combining computation blocks and I/O interfaces. For example, computational power can be increased by decreasing the number I/O interfaces and adding computation blocks in their place. Since this system requires many connections to the system snoop bus, lower latency and high-frequency operation are difficult goals to reach.

PRIMEQUEST uses a new snooping design, which moves snoop functions from the I/O interface blocks to the system bus crossbar. This design enables both flexible I/O configuration and low latency memory access. The design is called the center snooping design in this paper.

3. Implementation

3.1 PRIMEQUEST architecture

The PRIMEQUEST server is an SMP server that accommodates up to 32 CPUs or sockets with 1 TB of memory and 128 PCI-X bus slots. As shown in Figure 1, the system accommodates up to eight system boards (SBs) and eight I/O boards (I/O units). One SB has four CPU sockets, 32 DIMMs, four memory controller chips (LDXs), and one system controller chip (FLN). One I/O unit has one IO controller chip (FLI) and two physical layer chips (FLPs). One I/O unit supports four PCI-X buses and an external I/O box. The I/O box supports twelve PCI-X buses. SBs and I/O units are connected by address crossbar chips (GACs) and data crossbar chips (GDXs).

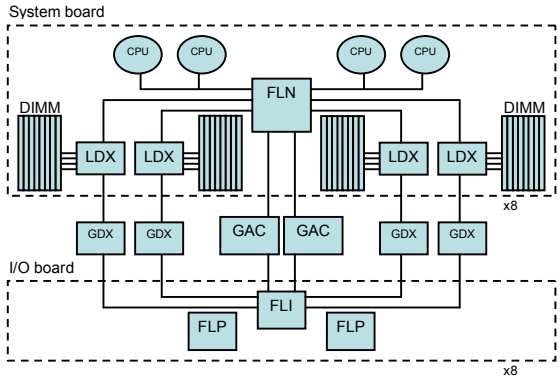

Figure 1. PRIMEQUEST system configuration.

The system can be configured as a mirror mode or non-mirror mode. Each mirror mode duplicates data and addresses for reliable operation, and non-mirror mode utilizes all resources to transfer data and addresses.

The CPUs are processors belonging to the Itanium processor family (IPF)[4], and the front side bus (FSB) operates at 667 MHz. Therefore, the bandwidth of the

FSB (10.7 GB/s) should be sufficient for service throughout the system without any bottleneck. Table 1 lists system data bandwidths. Service for all demand from the FSB is provided in non-mirror mode or a mirror mode.

Table 1. System data bandwidths (GB/s)

	Non-mirror mode	Mirror mode
Front side bus	10.7 x 16	10.7 x 8
System bus	170	85
Memory bus	170	85

Users can select from combinations of operating modes and mirror modes, based on their reliability requirements. Fujitsu's clustering software, PRIMECLUSTER, is utilized for maximum reliability. PRIMECLUSTER can detect a failure in a process quickly and hand over the process to a standby node. The down time, which is equal to the switching time, and failure rate are multiplied and totaled for all assumed failures to calculate the system down time per year. The resulting annual system down time would be five minutes. Therefore, availability should be 99.999%.

3.2 Dual synchronous system architecture (DSSA)

The DSSA essentially duplicates data paths, integrates many checkers at the appropriate locations, detects faults, disconnects any faulty part, and continues operation without any support from operating systems and applications. The DSSA also checks output of each LSI device, and this check contributes to precise fault location. Figure 2 shows the DSSA conceptual design and how the system is duplicated and checked.

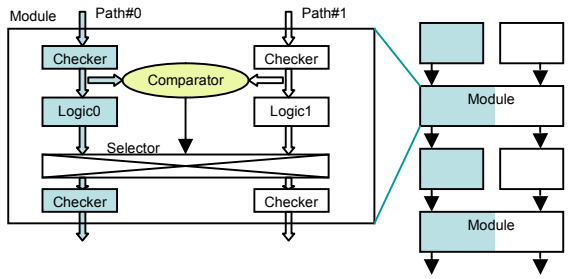

Figure 2. DSSA conceptual design.

In duplex mode or a mirror mode, the same data, address, and control signals are sent to both path#0 and path#1. All addresses and data are ECC protected, and they are checked and corrected at the input ports of LSI devices. 4-bit block correction and 4-bit double block detection (class of S4EC-S8/4ED [2]) are employed for data paths. If a correctable error is detected, the error is corrected, operation is not interrupted, and duplex operation is maintained. If an uncorrectable error is detected, the data path carrying the correct data is selected. If an uncorrectable error is detected repeatedly, the system disconnects the faulty path. However, only the faulty component path is disconnected; other parts of the system continue running in a mirror mode. This architecture minimizes the effects of errors. If no ECC error is found but a compare error is detected, the system stops in order to prevent error propagation.

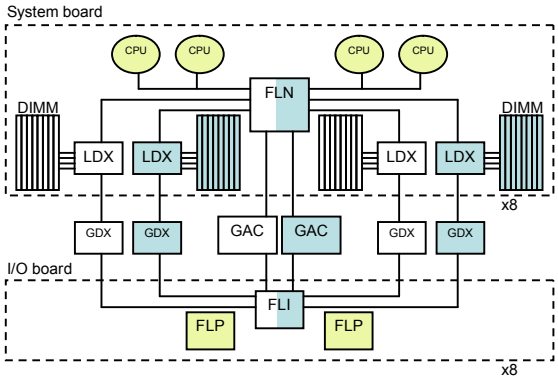

Figure 3. System mirror operation.

Note that several modules can be implemented into a single LSI device, and because any error would likely occur in a small portion of the LSI device, the effect of the error is limited.

For data path components, two simply implemented LSI devices can be used for each path. Figure 3 shows how the system mirror mode is put into operation. Each colored part (for path#0) and white part (for path#1) of a component uses the same functions synchronously, makes a comparison with the other part, and checks for errors. The FLN and FLI implement duplex paths and compare logic while each group consisting of an LDX, GDX, and GAC implements a single path without a comparator.

The DSSA realizes a highly reliable system by duplicating components and selecting error-free data. It keeps the complexity and hardware overhead low by allowing use of single data path logic LSI devices with simple designs.

The current system operating mode does not support duplication and comparison of I/O units and CPUs.

Table 2 lists the combinations of mirror modes in the system. The PRIMEQUEST server is capable of running in different mirror modes in different partitions.

Interaction between partitions is isolated by the partitions. System mirror mode duplicates both address and data buses, and memory is mirrored. The total size of available memory is reduced by half in system mirror mode, but address mirror mode duplicates only address paths so memory is fully utilized. These different reliability options are useful for server consolidation with the system, such as when a database server runs in fully duplicated mode while application servers run with a moderate reliability level. System bus resources are fully utilized for optimum performance when the system is used for research and development, which do not require duplication and compare.

Table 2. Combinations of mirror modes

Mode	Non-mirror	Address mirror	System mirror
Non-mirror	OK	-	-
Address mirror	-	OK	OK
System mirror	-	OK	OK

We implemented three error reporting levels, which enable starting of a different error recovery process based on the severity of an error. If a critical error is detected, a dedicated error reporting mechanism notifies the system management module, which is independent from the operating system and applications. This notification helps to analyze the fault and minimize the down time.

3.3 Center snooping design

The center snooping design moves snoop functions from the I/O board to the system bus chips. By moving the distributed snoop functions to system bus chips, we reduced snooping latency and enabled symmetrical configuration of I/O units and CPUs. The center snooping design can be used for highly flexible I/O grouping and a high-performance system bus.

Figure 4 shows operation with this design. In a conventional implementation, snooping latency is limited by the longest path (case A). Case A assumes the I/O interface block (I/O unit) and its LSI device (FLI) are far away from other components, specifically, the GAC and FLN, which would enable flexible I/O. Addresses are broadcast from the GAC to both the FLN and FLI. The GAC has to wait for the latest response, which is from the FLI in this case. If the distance is large, delivery of address information takes a long time because of the signal frequency limit and the propagation delay due to the wire length.

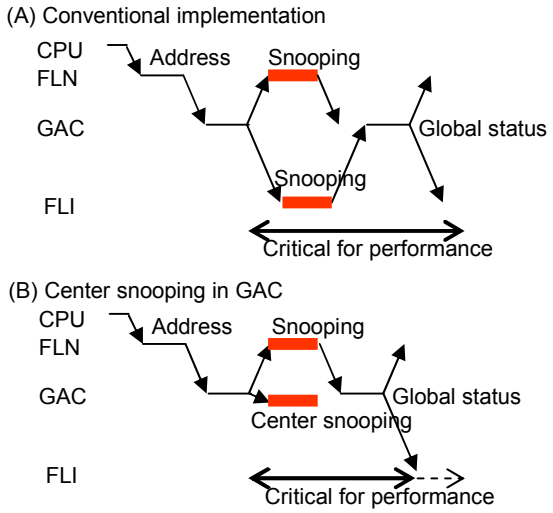

Figure 4. Difference between locations of snooping.

In case B, the center snooping design moves the functions to the system bus LSI device (GAC) at the center of the snooping. As a result, the critical section for performance has a very short span regardless of the physical distance on the I/O unit.

The conventional design requires a minimum number of transistors since each LSI device implements functions exclusively. However, the center snooping design duplicates controls and states for data access to the GAC to reduce latency.

This design also decreases the number of signal wires used. The number of GAC signal pins is decreased in this design from 1804 (estimated) to 1196 by eliminating snoop-related signal pins, and one-chip implementation at a reasonable cost is possible.

3.4 MTL

A high data bandwidth system bus should operate with a sufficient margin. The numbers of connections and wires affect the system cost and reliability. Ensuring global synchronizing operations is the key to SMP design. We developed and implemented a 1.3-GB/s single-ended transmission technology, called MTL, for system bus connections. MTL enables a system bandwidth of 170 GB/s with a few signal pins.

MTL uses source synchronous parallel transmission allowing a data skew. The receiving logic compensates for the skew, and phase-aligned data are provided to the core logic. These mechanisms guarantee that the globally synchronized timing in this large 32-way system is visible.

Although MTL requires many logic gates as compared to an ordinary I/O interface, this is negligible with 90-nm technology. We confirmed error-free MTL operation at 1.6 GHz in our laboratory. Table 3 outlines the PRIMEQUEST chipsets in which MTL is fully employed.

4. Product Design

4.1 Configuration and packaging

Figures 5 and 6 outline operation with the center snooping design. The arrows with numbers indicate how a data load is processed in the system.

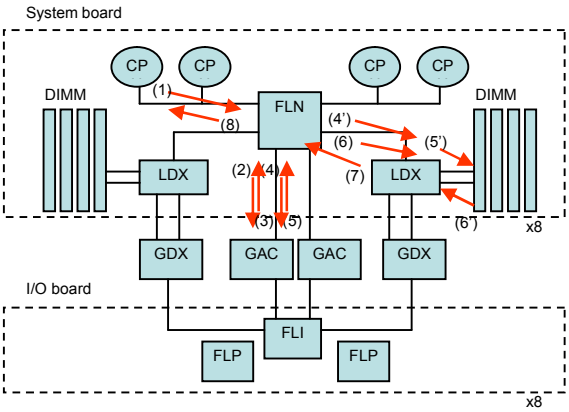

Figure 5. Data access steps.

(1) The CPU issues the load to the FLN through the FSB, (2) the FLN issues a snoop request to the GAC, (3) the GAC broadcasts the address to all FLNs to initiate snoop processing, (4) the FLN replies to the snoop request (4') while it issues a request to the LDX to start the DIMM (5'), (5) the GAC merges the snoop status and broadcasts the results to the FLNs, and (6) the FLN signals the LDX to transfer the data in order to send the data to the requesting CPU. Finally, (7) the LDX sends the data to the FLN, and (8) the FLN forwards the data to the CPU. If the owner of the data is the FLI, the GAC will detect it between steps 3 and 5. The GAC would then request the data from the FLI.

As discussed in section 3.3 and shown in Figure 4, snooping in the center snooping design is done by only the FLN and GAC. The GAC and FLI do not interact, which is required by ordinary implementations. Since snooping is done between only the FLN and GAC, we had to focus only on minimizing the distance between them. We could place the I/O unit rather far away from the core of snooping. As Figure 6 shows, the FLN on the SB and the GAC on the AXI are very near each other, jointed by the center plane. The length of each signal wire between the FLN and GAC is about 0.5 m, and that between the GAC and FLI is 1.0 m. Since I/O units and SBs, each of which have memory and CPUs, are separated, users can flexibly configure partitions based on the demand for I/O, CPUs, and memory.

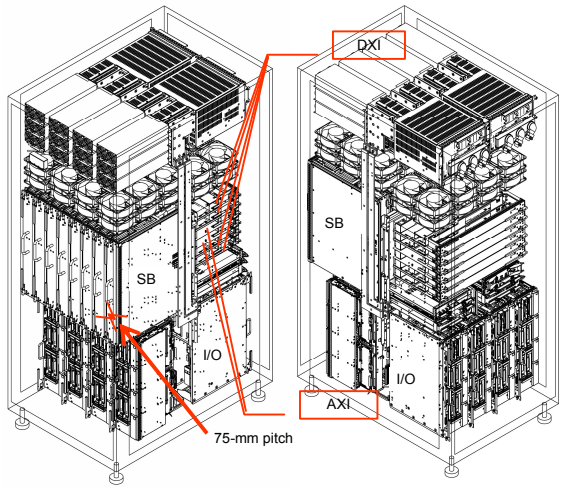

Figure 6. Packaging.

Table 3. PRIMEQUEST chipset

		FLN	LDX	GAC	GDX	FLI	FLP
Main function		CPU control	DIMM control	Address control	Data control	I/O control	Phy. control
Technology (nm)		90	90	90	90	90	130
Signals	AGTL+	516	-	-	-	-	-
	MTL	592	295	1152	912	292	-
	DDR2	-	582	-	-	-	-
	Other	84	50	44	47	650	497
	Total	1192	927	1196	959	942	497
Transistors (mega)		210	25	29	33	70	16

4.2 Cooling system

Keeping CPUs and memory cool is essential to maintaining steady system operation over a long period. It extends the life of parts, reduces the possibility of failures, and provides a margin for drifts in frequency and voltage.

Our design of the shape of cooling ducts uses guided channel cooling technology. We added a taper channel that provides fresh air directly to the processor core. This enables a board to have four 130-W CPUs with a narrow pitch, 75 mm.

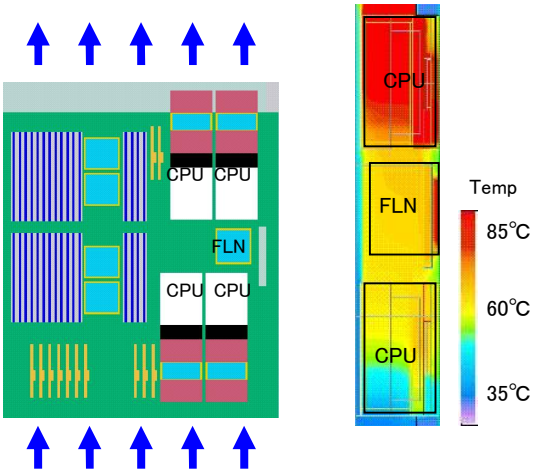

Figure 7. Simulated results (ordinary cooling).

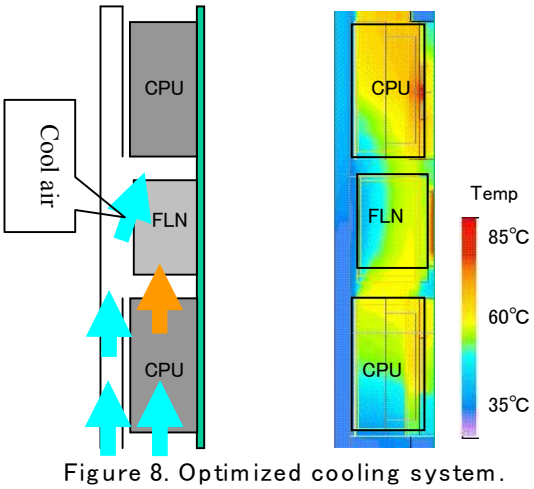

Figure 8. Optimized cooling system.

Figure 7 shows the layout of an SB. A total of four CPUs with a maximum power consumption of 130 W are mounted. Since the two pairs of CPUs are serially mounted, the upper CPUs absorb heat from hot air, and no ordinary cooling technique can cool the upper CPUs. The right side of Figure 7 shows simulated results for an ordinary cooling technique. In contrast, Figure 8 shows that both CPUs are kept cool with the added channel, even at the maximum power consumption.

Table 4 lists PRIMEQUEST design specifications. The PRIMEQUEST server provides enough system bus performance for several CPU generations and enough bandwidth for memory bandwidth intensive applications such as R&D applications.

Table 4. PRIMEQUEST design specifications

CPU frequency	1.6 to 2+ GHz
Max. number of CPUs	32 sockets
FSB frequency	667 MHz, max.
System bus frequency	1.3 GHz, max.
Snooping bandwidth	170 GB/s, max.
Data bandwidth	190 GB/s, max.*

* 8 x 667 x (8 + 8 x 4) x 8/9 = 190 GB/s

4.3 Implementation and production for reliability

The PRIMEQUEST server is built for high reliability, which was a consideration in not only the architecture but also implementation and production.

Emulation and simulation were utilized in implementation to verify the logics. All addresses and data packets are ECC protected. In addition, LSI devices are equipped with error generators to simulate errors, so designers and testers can verify that all anticipated errors are detected and corrected and that the error recovery process is successfully completed. These mechanisms are utilized in the design verification phase by LSI designers and in the production phase at the factory. The mechanisms are also utilized by middleware, or clustering software, to check its recovery function.

Key devices are tested at the operating frequency and a burn-in test is performed at the factory to check for initial faults. The system is shipped to customers only after it passes a long-duration heavy-load running test under a high temperature.

5. Conclusion

We have developed a high-end enterprise open server called PRIMEQUEST. It employs up to 32 CPUs and has many reliability features. The PRIMEQUEST server makes use of rapidly improving silicon device technologies, such as the higher speeds and densities that they provide, to satisfy cost and performance requirements. Our technologies in this system are the DSSA and the center snooping design. The DSSA duplicates almost all parts and paths, and it operates synchronously. It detects errors and performs recovery. It does not require any modification to applications, middleware, and operating systems. Consequently, many applications can run on this reliable system without special care required. The center snooping design realizes flexible I/O without a performance penalty. It also decreases the number of signal pins used, and this reduces the possibility of failures.

MTL expands the bandwidth per signal pin and enables synchronous system operation throughout the system. The logic overhead of MTL is negligible with 90-nm technology.

We will continue to develop the system to obtain better performance and more reliability. Furthermore, we plan to evaluate the performance of the system with middleware and operating systems.

6. Acknowledgements

Part of this work was funded by the New Energy and Industrial Technology Development Organization.

The authors would like to thank the many managers and engineers who were in charge of LSI, PCB, and packaging design and development, and everyone else who has supported us since the beginning of this work, from design and development to implementation and evaluation of the PRIMEQUEST server.

A lot of people, not only Fujitsu staff but also CAD service providers, CPU and parts vendors, and operating system developers, have worked hard to build the system from scratch within a very short period, specifically, twenty-four months. With their help, we were able to design and implement all mechanisms, using our collective knowledge to create a reliable and high-performance system.

We also wish to thank the reviewers of our paper for their valuable comments on improvements.

7. References

[1] Hewlett-Packard Development Company, "HP NonStop Advanced Architecture" (September 2003), http://h71028.www7.hp.com/ERC/downloads/ADVARCWP.pdf.

[2] S. Kaneda and E. Fujiwara, "Single Byte Error Correcting - Double Byte Error Detecting Codes for Memory Systems," Dig. IEEE Int. Symp. Fault-Tolerant Computing (October 1980): 45-51.

[3] Sun Microsystems, "The Sun Fire™ 15K Server for High Performance and Technical Markets" (September 25, 2001), http://www.sun.com/servers/white-papers/SunFire15K_HPCTechMkts.pdf.

[4] N. Quach, "High availability and reliability in the Itanium processor," IEEE Micro 20(5), September/October 2000: 61-69.

[5] Discontig Project, http://discontig.sourceforge.net/.

[6] Naoki Izuta et al., “Overview of PRIMEPOWER 2000/1000/800 Hardware” (February 7, 2001), http://magazine.fujitsu.com/us/vol36-2/paper03.pdf.

[7] Alan Charlesworth, “Starfire: Extending the SMP Envelope,” IEEE Micro (January/February 1998): 38-49.

[8] Zarka Cvetanovic, “Performance analysis of the Alpha 21345-based HP GS1280 Multiprocessor,” in The 30th Annual International Symposium on Computer Architecture (June 9-11, 2003), 218-228.