The UltraSPARC™-IIi Processor

Technology White Paper

Printed in the United States of America.
901 San Antonio Road, Palo Alto, California 94303 U.S.A

Contents

Introduction 1

The growth in global networks is transforming the nature of computing. Along with an increase in quantity, the arrival of desktop video conferencing and collaborative workgroups is changing the nature of data itself. Today, most networks are dominated by the flow of pure alphanumeric information, with only a small fraction devoted to imaging, video, geometric, pen-based, audio or new-media data. In the near future, this ratio will be reversed, and audio, video, and imaging will account for the majority of data transferred over networks and accessed by desktop computers.

Network computing has virtually replaced stand-alone applications. Until recently, word processing and spreadsheet applications drove the widespread demand for desktop systems. Today, on-site CD-ROM information and remote databases, as well as networked applications such as electronic mail and the Internet access all play an overwhelming role in the decision to buy desktops. The increasing capacity of network bandwidth and compression techniques are enabling influences on the dramatic transformation in the nature of desktop applications and data.

In the very near future, offices will be populated by a new class of machine. The *new-media* desktop will incorporate extensive network and collaborative workgroup capabilities, rapid access to pictorial databases, and support for real-time video and audio at broadcast-level quality. Users of these machines will concurrently draw upon a multitude of resources to stay productive. For example, a financial analyst may have one window providing updates from a

commodities market, a second window containing the video of a conference call with Europe, a third with a spreadsheet continuously updating financial positions, and a fourth listing incoming voice mail by origin of the caller.

When users operate in such an environment, the sources and destinations of the data will be irrelevant with no distinction between LANs and WANs. Their driving interest will be the nature of the data and not its location, and they will not perceive the network interface as a barrier to information access. The distinction between “inside” and “outside” resources will dissolve.

Integrated Media Processors

The explosion in new media processing and workplace interconnection can only be accommodated if the processor assumes additional responsibility — the conventional approach of expanding network bandwidth and increasing processor clock rates will no longer suffice. The processor must be redesigned to seamlessly interact in a complex, multifaceted networked environment that demands nothing less than peak performance. To accommodate this shift in computing needs, processors must be designed from the ground up for new media applications and network computing.

The exponential growth of new media applications is forcing processors and the systems in which they reside to shoulder the responsibility for high-speed graphics, 2-D and 3-D imaging, video processing, and image compression and decompression. Although such functions typically consist of simple operations, their sheer volume creates a flood of data.

The most economical way to implement new media support is through the creation of specialized instructions and hardware-assists on the processor. With such an approach, new media applications can effectively harness the powerful pipelined, superscalar structure of emerging processor designs. Rather than using expensive, specialized circuitry or boards, the capabilities of the processor itself can be directly tapped, with dramatic advantages in imaging and graphics applications.

Integrated processors like Sun’s UltraSPARC-IIi exemplify this technology trend. Typically, increases in functionality are accompanied by corresponding rises in cost. With high performance, sophisticated, low-cost computing in high demand, reductions in processor costs are essential, and high levels of integration are the only means to accomplish this delicate balance. Today, even faster and functional with higher levels of integration at lower cost,

UltraSPARC-IIi offers increased floating-point performance, high levels of integration, companion PCI-to-PCI bridge ASICs, and new media capabilities for today's demanding applications and environments.

The SPARC™ Architecture

SPARC™ continues to be the flagship processor technology for Sun Microsystems, and Sun's continuous refinement of SPARC's traditional strengths stands as testimony to their commitment to the architecture. Key features of SPARC include:

- *Performance and Economy*

  With its simplified instruction set, SPARC processors achieve a higher number of instructions per second with fewer transistors. Simplicity of design is the cornerstone for SPARC, enabling shorter development cycles, smaller die sizes, and ever-increasing performance.

- *Scalability*

  SPARC is scalable across a wide range of semiconductor technologies, chip implementations and system configurations. A benefit of SPARC is its flexible integration of cache, memory management, and floating-point units, enabling processors to be produced at price / performance levels suitable for systems ranging from laptops to supercomputers.

- *Open Architecture*

  Users are increasingly unwilling to pay the high costs of proprietary technology. Understanding this, Sun™ pioneered the concept of open systems in the early 1980s — unveiling one of the most significant technology marketing programs of the last several decades. Sun believes that by employing basic, standardized technologies, more parties will develop, use, and innovate around that technology, and costs will be driven down.

  A key element of SPARC is its open availability — ensuring that customers have unencumbered access to compatible technology from multiple vendors. To further encourage SPARC product development and compatibility, Sun has broadened access to complete system level and CPU component technologies. Today, Sun's Sun Microelectronics™ (SME) business unit licenses specifications, designs, masks, and chip components for SPARC.

- *Powerful Development Tools and Deployment Environment*

  The Solaris™ operating environment, Sun's implementation of the UNIX® operating system, has been available now for more than a decade, receiving hundreds of millions of hours of accumulated running time. Running on SPARC-based systems, Solaris is a time-tested, secure, highly-refined product that runs the critical applications of thousands of businesses.

  A complete set of development tools from Sun and third parties is available for developing SPARC-based hardware and software, and real-time operating systems (RTOSs). Products include optimizing and cross-compilers, debuggers, emulation systems, and source management systems.

The SPARC-V9 Architecture

SPARC Version 9 (SPARC-V9) is the most significant change to the SPARC architecture since its release in 1987. Designed to remain competitive well into the next century, SPARC-V9 incorporates several important enhancements:

- 64-bit addressing and 64-bit data
- Improved system performance through increased bandwidth, throughput, and industry-wide benchmarks
- Features to support optimizing compilers and high performance operating systems
- Superscalar implementation
- Fault tolerance
- Fast trap handling and context switching
- Big- and little-endian byte ordering

The UltraSPARC-IIi Processor

A second-generation implementation of the UltraSPARC™ pipelined processor architecture, the UltraSPARC-IIi retains complete backwards compatibility with the 32-bit SPARC-V8 specification and previous generation UltraSPARC processors, ensuring binary compatibility with existing applications. UltraSPARC-IIi not only provides 64-bit data and addressing, but has a number of other features to improve operating system and application performance:

- Nine stage pipeline; can issue up to 4 instructions per cycle
- Better cache management and greatly reduced memory latency
- On-chip 16 KB Data and Instruction caches, with up to 2 MB external cache

- On-chip graphics and imaging support
- Implemented using 0.35 micron gate, 0.25 micron poly, 5-layer metal CMOS technology operating at 2.6 volts; I/O at 3.3 volts
- Multiple SRAM modes allow flexibility and economy in designing systems
- Higher speed memory transfers (1.6 GB/sec peak)
- Implements Sun's Ultra Port Architecture (UPA) Interconnect
- Supports SPARC V9 *prefetch* instruction and outstanding memory requests
- High performance both in SPECint95 (>12) and SPECfp95 (>12) at 300 Mhz
- Ceramic Land-Grid Array packaging

UltraSPARC-IIi Performance

In addition to providing excellent SPECint 95 and SPECfp95 performance, the UltraSPARC-IIi processor is architected for high memory performance and increased bandwidth to peripherals. Table 1-1 details the representative performance characteristics of the UltraSPARC-IIi processor.

SPEC Performance[1]	SPECint 95 SPECfp 95	12.1 (peak) 12.9 (peak)
Memory Performance	Maximum E-cache read/write bandwidth Maximum DRAM random read/write bandwidth Maximum same page read bandwidth DRAM-to-DRAM memory copy DRAM-to-UPA64S memory copy	1.2 GB/sec 350 MB/sec 400 MB/sec 275 MB/sec 550 MB/sec
FP Vector	STREAM Copy (compiled) STREAM Scale (compiled) STREAM Add (compiled) STREAM Triad (compiled)	200 MB/sec 210 MB/sec 230 MB/sec 230 MB/sec
UPA64S Bandwidth	From Processor to UPA64S (PIO)	Random 64-byte writes 600 MB/sec Compressed 8-byte writes 800 MB/sec
PCI Bandwidth	To DRAM from Processor PCI bus (DMA)	Random 64-byte reads 132 MB/sec Random 64-byte writes 151 MB/sec
	To E-Cache from Processor PCI bus (DMA)	Random 64-byte reads 163 MB/sec Random 64-byte writes 186 MB/sec
	From Processor to Processor PCI bus (PIO)	64-byte writes 200 MB/sec

1. Measured on Ultra 10 systems with 512 KB second-level cache

Table 1-1 UltraSPARC-IIi provides high performance at low cost

Scope and Audience

This document is intended for the technical developers and users of uniprocessor computing platforms based on the UltraSPARC-IIi, including single board computers, portables, workstations, and servers. Additionally, the UltraSPARC-IIi processor is designed for the high-end embedded market segment, including high-end networking, telecommunications, and imaging devices. This paper explores the UltraSPARC-IIi architecture, and discusses the high-level differences between the UltraSPARC-IIi processor and its competitors.

The SPARC Microprocessor Strategy

2

The SPARC architecture is an open specification designed to encourage broad availability. Developed by Sun Microsystems in the mid 1980s, the SPARC architecture was transferred to SPARC International (an organization open to all component and systems manufacturers and software developers who want to design SPARC products) in an effort to guide SPARC evolution and promote open systems. In turning the SPARC definition over to an industry body, Sun Microsystems has ensured that SPARC is an open standard, not strictly controlled by its designer. SPARC International has numerous vendor programs promoting SPARC and related technologies to an open, competitive marketplace.

SPARC International sets both hardware and software compatibility standards, and encourages independent software vendors (ISVs) to develop for, and port to, the SPARC International “SPARC-compliant” application standard. To ensure that SPARC products are binary compatible, SPARC International established the SPARC Compliance Definition (SCD), a specification of common system software and hardware interfaces. By developing systems and software that conform to the SCD, SPARC system vendors and application developers ensure their products are binary compatible and interoperable.

Open Availability

Sun believes that open availability is critical to technological success. Key to this strategy are multiple vendors and implementations to foster competition and innovation, and broad technology licensing to ensure unencumbered access. Together, these methodologies increase the open availability of both specifications and implementations.

- **Interfaces should be openly available (published)**
- **Interface specifications should be well written**
- **Open interfaces should be available for reasonable fees**
- **Multiple implementations should be available from multiple vendors**
- **Open system interfaces should have at least one reference implementation available for a reasonable fee**
- **Open system specifications should be branded**

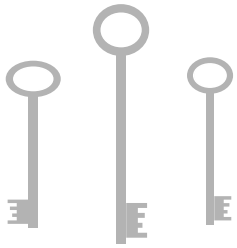

Figure 2-1 Sun believes that open availability is critical to technological success.

Multiple Vendors and Implementors

The SPARC licensing approach encourages multiple semiconductor vendors — with multiple areas of expertise — to accelerate the rate of innovation and offer different implementations of competitively priced SPARC processors. Because the SPARC architecture is open, multiple semiconductor and system vendors can simultaneously design compatible SPARC products using a variety of technologies.

System developers can take advantage of a variety of SPARC chip suppliers, software, and hardware technologies to implement a broad range of compatible systems. Today, the SPARC architecture is the underlying building block for systems spanning laptops to supercomputers. It is completely open to system developers, enabling them to bring to market competitive products that address specific customer needs without sacrificing compatibility.

Implementors can reduce development costs and shorten time-to-market by leveraging existing SPARC hardware and software technologies. This has resulted in a growing suite of innovative and competitive products including a range of binary-compatible SPARC computers from multiple vendors, and

optimized bus architectures for enhanced system performance. Today, over 12,000 development, administration, and end-user applications are available for SPARC-based systems.

Broad Technology Licensing and Component Sales

At the core of Sun Microsystems' highly successful computing model is the philosophy of open technology licensing. With the widespread industry acceptance of open systems, Sun has broadened access to complete system level and CPU component technologies. This approach, and the variety of programs it fosters, serve the entire spectrum of computing technology providers, from manufacturers of components and embedded systems, to board and platform vendors. To facilitate open licensing, Sun provides the following key elements:

- Access to all levels of technology — CPUs, support ASICs, buses, interfaces and board designs
- Early access to CPU designs and their development environment
- Access to system development tools for design, test and manufacturing
- Comprehensive granting of rights for development of standard and derivative components
- Structured, standard licensing products which provide everything vendors need to build SPARC-based and SPARC derivative components, systems and subsystems

Sun Microelectronics is chartered to actively design microprocessors and drive technology licensing and the sale of components, boards, and tools to promote this model. SME's technology licensing programs provide developers with the complete range of components, designs, interfaces and documentation — everything they need to design and build SPARC-based components, systems and subsystems. This broad licensing of SPARC technologies enables better hardware and software compatibility, promotes the architecture in the marketplace, and encourages the development and porting of applications to the SPARC environment.

SPARC Performance Scalability

The SPARC architecture enables a unique combination of semiconductor and design scalability. With its high bandwidth bus support and register window design, the SPARC design allows implementations through a range of price / performance levels.

SPARC Uniprocessor Designs

The SPARC V9 design assumes a linear, 64-bit virtual address space for user application programs, and a flexible data bus size for system architecture optimization (figure 2-2).

Figure 2-2 SPARC conceptual bl

SPARC defi ata types for the Inte ority having single-cycle execution. Integer and floating-point instructions ca ncurrently with architectural support for program concurren unit uses a windowed-register model. Although the registe defined by SPARC, component manufacturers have the f plement the optimum number of register windows to m / performance requirements for a particular market.

SPARC defines the instructions, regis re, and data types for the floating-point unit. The floating-p upports single, double, and quad precision operands and operatio thirty-two 32-bit registers, the

SPARC-V8 FPU registers can hold 32 single-, 16 double-, or 8 quad-precision values. The SPARC floating-point load and store double operations help improve the performance of double- and quad-precision programs. SPARC-V9 doubles the number of registers and adds floating-point load and store quad operations.

Cache Considerations

To keep up with processor cycle times, close coupling of the processor and memory system is required. Caches improve performance by providing faster access to data because fewer cycles are required to access cache than for a memory access. Several cache architectures commonly used, separately or in combination, include set associativity, separate data and instruction caches, and direct-mapped cache schemes.

The SPARC architecture can implement either a unified or Harvard cache organization. The Harvard cache organization utilizes independent instruction (I-cache) and data caches (D-cache), which may each supply requested data over independent buses simultaneously. At any given clock cycle, the processor can fetch one instruction from I-cache and data from the D-cache.

Memory Management Capabilities

SPARC processors access main memory through a virtually accessed cache. Virtual memory is a method by which applications are written assuming a full 64-bit address space (32-bit in SPARC-V8). Virtual address caches hide physical memory configurations and machine dependencies from software, with memory management units translating virtual addresses to physical addresses in memory. Operating systems, like Solaris, work with SPARC's virtual memory capabilities by allowing a greater total virtual address space than is physically present at any given time.

SPARC Development Tools

Sun understands that processors alone do not create systems. In order to encourage the development of sophisticated systems, they have provided a robust set of tools to aid the development of SPARC-compatible hardware and software products.

Optimizing Compilers

Optimizing compilers, like the Sun WorkShop™ compilers, provide significant improvements in program execution speed by using a wide variety of techniques to produce the most efficient sequence of machine instructions. Sun's compilers for SPARC use the latest in compiler optimization technology to produce fast, efficient, and reliable code:

- *Tail recursion elimination*, to convert self-recursive procedures into iterative procedures, saving stack manipulation time
- *Loop-invariant code motion*, to locate and remove computations that yield the same result
- *Profiling*, to allow optimizations to adapt themselves to program behavior
- *Induction-variable strength reduction*, replacing slower operations with faster ones
- *Common subexpression elimination*, to save expression values for reuse
- *Register allocation*, to hold frequently accessed data
- *Loop unrolling*, to reduce run-time by reducing loop overhead and increasing opportunities for more efficient instruction scheduling
- *Modulo scheduling*, to aggressively schedule inner loops in which programs spend most of their execution time
- *Dead code elimination*, to eliminate unreachable code
- *Loop inversion*, to convert pre-test loops into post-test loops, reducing the number of branches required per iteration
- *Global value numbering*, to improve general code quality and optimizations, enhancing integer application performance

Sun WorkShop programming tools are optimized and tuned for a variety of high-level languages, including C, C++, FORTRAN, Pascal, and Modula 2. Providing a highly-productive programming environment for the development of sophisticated software applications, these integrated tools offer a complete set of utilities for editing, compiling, linking, debugging, and tuning an application for SPARC systems.

Products for Embedded Systems

The UltraSPARC-IIi processor is designed to meet the diverse needs of embedded solutions in the networking, telecommunication, and imaging arenas. In order to ensure a full range of solutions for embedded systems

developers, Sun offers world class real-time operating systems like Sun's ChorusOS (formerly Chorus ClassiX) and Wind River's VxWorks®, as well as UltraSPARC-IIi evaluation and design kits, and a reference platform.

The UltraSPARC-IIi Architecture 3

The UltraSPARC-IIi Design Philosophy

Compute-bound execution rate is a product of three factors: the number of compiler-generated instructions in an execution trace (NI), the sustained rate of instructions completed per cycle (IPC), and the clock rate made possible by the underlying integrated circuit technology [Patterson 1989].

High performance requires advances on all three fronts. Designs benefit from better compilers and higher clock rates. However, cost-effective systems cannot rely exclusively on exotic, high-clock-rate designs. To benefit from economies of scale, mainstream semiconductor technology must be employed. UltraSPARC-IIi systems provide high performance at reasonable clock rates by optimizing the IPC. This approach presents design challenges for managing the processor pipeline and memory hierarchy, however. Average memory access time must be reduced and the number of instructions issued per cycle must rise, increasing performance without compromising the complexity or design of the processor.

Design Criteria and Choices

Known bottlenecks to issuing multiple instructions include control dependencies, data dependencies, the number of ports to the processor registers, the number of ports to the memory hierarchy, and the number of floating-point pipelines. Sun’s comprehensive research indicates that many

applications can benefit from a processor that issues multiple instructions per cycle. Table 3-1 illustrates a typical instruction mix for integer and floating-point operations.

Instruction Class	Integer Application	Floating-point Application
Integer Arithmetic	50%	25%
Floating-point Arithmetic	0%	30%
Loads	17%	25%
Stores	8%	15%
Branches	25%	5%

Table 3-1 Typical instruction mix for integer and floating-point applications

This mix represents many challenges for a superscalar design executing code at the peak rate. Considerations such as overall processor and system cost, maintaining reasonable floating-point performance, cache sizes, and time-to-market, affect the cost and performance of the chip. With UltraSPARC-IIi, Sun has attempted to improve performance in areas where economy would not be compromised. To mitigate these expected challenges, several design decisions were made that defined the characteristics of UltraSPARC-IIi:

- On-chip caches
- A wide instruction fetch (128 bits)
- Dynamic branch prediction
- Optimized memory operations in the pipeline
- Floating-point memory access which does not stall floating-point dispatch
- Write buffer to decouple the processor from store completion
- Integrated memory controller, level 2 cache, and PCI bus controller

UltraSPARC-IIi Module

The UltraSPARC-IIi microprocessor is part of a highly integrated processor module designed for high performance at low cost. The module incorporates a high performance processor, and on-chip memory and I/O control to facilitate its easy and economical integration into a system:

- *UltraSPARC-IIi microprocessor*, a high-performance, highly-integrated superscalar processor implementing the SPARC-V9 64-bit RISC architecture running at 270 MHz or 300MHz. The processor integrates a PCI bus controller to interface directly with a 32-bit PCI version 2.1 compliant bus; an I/O memory management unit to manage virtual to physical memory address mapping using translation lookaside buffers for improved performance; an external cache unit to handle instruction and data cache misses efficiently and provides high transfer bandwidth; and a memory and UPA64S control unit to manage all transactions to DRAM and the UPA64S subsystem typically used for high performance graphics.
- *System clock and clock generation*
- *Interfaces to the UPA64S subsystem, memory, and Advanced PCI Bridge (APB)*

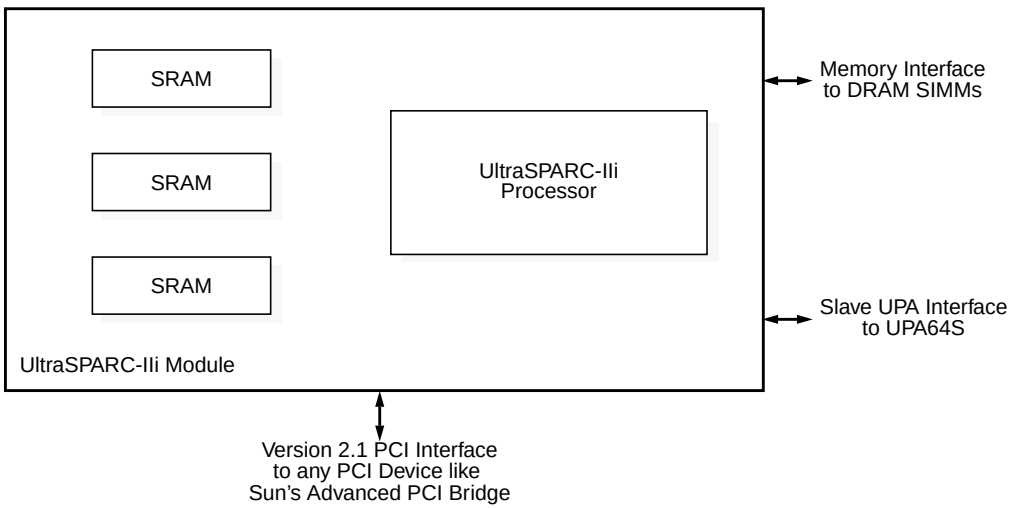

Figure 3-1 UltraSPARC-IIi module functional block diagram

UltraSPARC-IIi Processor Architecture

UltraSPARC-IIi is the newest member of Sun's family of SPARC CPUs. Designed to support low cost uniprocessor systems, UltraSPARC-IIi is ideal for networked applications needing compute power, multimedia capabilities, optimized data throughput, and low cost. The most integrated member of the SPARC family to date, UltraSPARC-IIi incorporates several key features:

- High performance, delivering more than 12 SPECint95 and 12 SPECfp95 at 300 MHz with 512 KB second-level cache
- Built using TI's state-of-the-art 0.35 micron gate, 0.25 micron poly CMOS process technology
- Low power operation for Energy Star compliance
- A fully static design, enabling significant power savings
- Nine stage pipeline, issuing up to 4 instructions per cycle
- Dynamic branch prediction
- On-chip instruction and data caches
- On-chip I/O memory management and control units (IOM and MCU)
- On-chip graphics and imaging support
- On-chip level 2 cache controller
- On-chip memory controller unit
- On-chip PCI bus controller
- Implements the UPA bus architecture

The UltraSPARC-IIi processor is a highly integrated, SPARC-V9 compliant implementation consisting of a Prefetch and Dispatch Unit, an Integer Execution Unit, a Floating-point Unit, an I/O Memory Management Unit, a Memory Control Unit, a Load and Store Unit, an External Cache Unit, a Graphics Unit, and Instruction and Data Caches (figure 3-2).

Prefetch and Dispatch Unit

The UltraSPARC-IIi Prefetch and Dispatch Unit (PDU) ensures that all execution units remain busy by fetching instructions before they are needed in the pipeline. Instructions can be prefetched from all levels of the memory hierarchy, including the instruction cache, external cache, and main memory. The PDU was designed with several features to support the high performance requirements of UltraSPARC-IIi:

- A 12-entry prefetch buffer decouples instruction prefetching from instruction dispatch and prevents pipeline stalls

- A 16 KB two-way associative I-cache that is physically indexed and tagged
- Instructions in the I-cache are pre-decoded
- A 9-stage instruction pipeline to minimize latency
- Dynamic branch prediction to allow for greater prediction accuracy

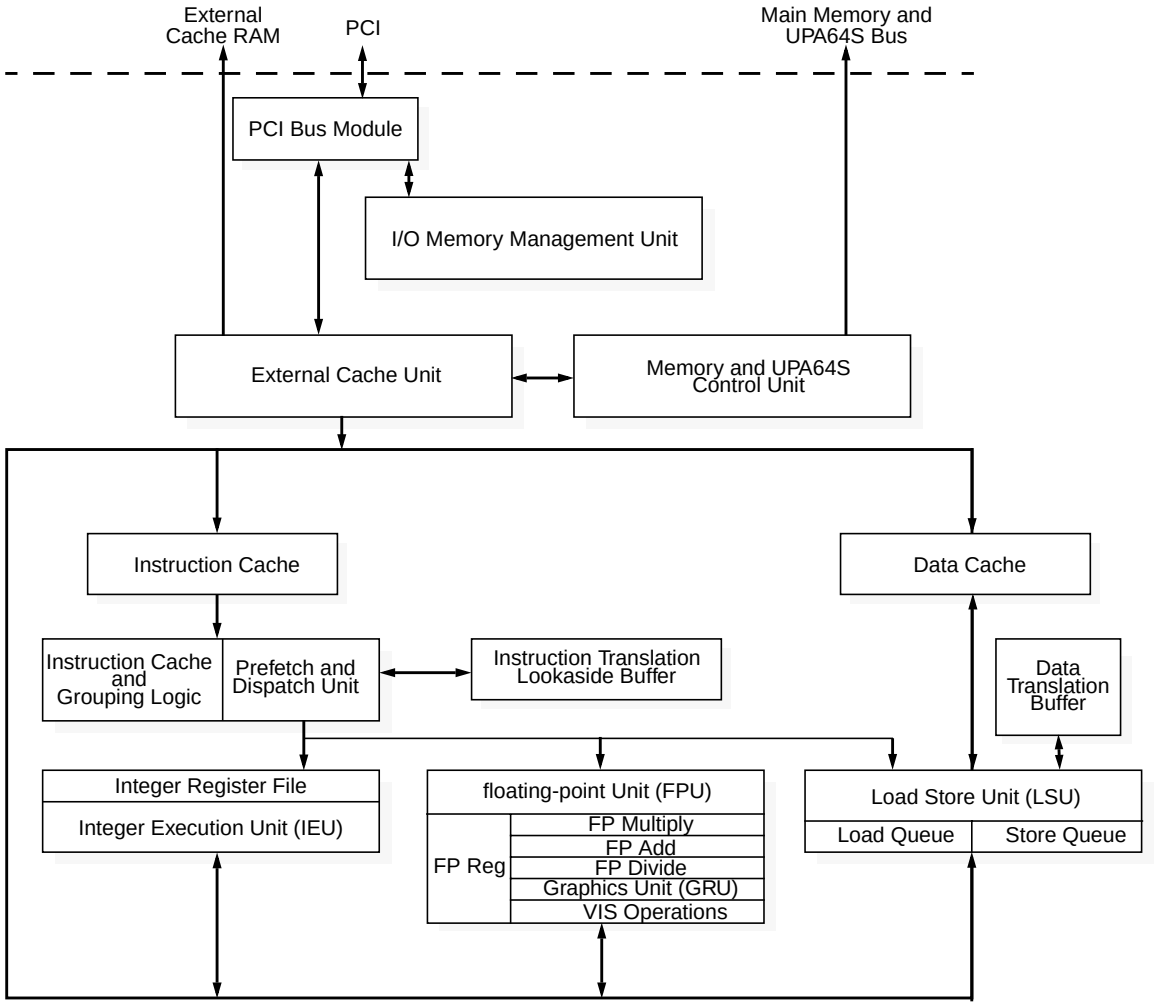

Figure 3-2 UltraSPARC-IIi functional block diagram

Pipeline Organization

UltraSPARC-IIi uses a double-instruction-issue pipeline with nine stages: *fetch*, *decode*, *grouping*, *execution*, *cache access*, *load miss*, *integer pipe wait*, *trap resolution*, and *writeback*. These stages imply that the latency (time from start to end of execution) of most instructions is nine clock cycles. However, at any given time, as many as nine instructions can execute simultaneously, producing an overall rate of execution of one clock per instruction in many cases. In reality, some instructions may require more than one cycle to execute due to the nature of the instruction, a cache miss, or other resource contentions. Three additional stages are added to the integer pipeline making it symmetrical with the floating-point pipeline, simplifying pipeline synchronization and exception handling (figure 3-3).

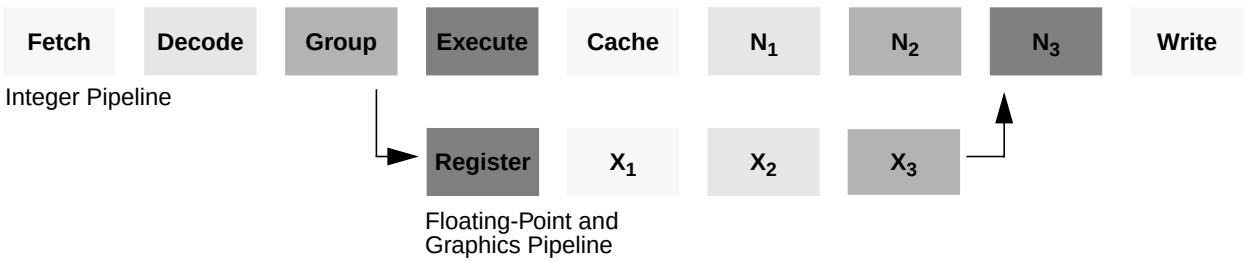

Figure 3-3 UltraSPARC-IIi processor pipeline strategy

The first stage of the pipeline is a fetch from the instruction cache. In the second stage, instructions are decoded and placed in the instruction buffer. The third stage, grouping, groups and dispatches up to four instructions. Next, integer instructions are executed and virtual addresses calculated during the execution stage. In the fifth stage the data cache is accessed. Cache hits and misses are determined, and branches are resolved. If a cache miss was detected, the loaded miss enters the load buffer. At this point, the integer pipe waits for the floating-point/graphics pipe to fill and traps are resolved. In the final stage, writeback, all results are written to the register files and instructions are committed.

Dynamic Branch Prediction

In order for reasonable computational tasks to be handled, a processor must support branching. Branches allow the execution flow of the program to change based on conditions detected by the software during execution. UltraSPARC-IIi uses dynamic branch prediction to speed the processing of branches.

Dynamic branch prediction enables the UltraSPARC-IIi processor to prefetch instructions and prepare them for use by various instruction execution units. The branch prediction mechanism is based on a two-bit state machine that predicts branches based on the specific branch's most recent history. Because the history of a branch may change every time it is encountered, the prediction of the branch must be dynamic.

- **Every branch in the instruction cache is predicted**
- **UltraSPARC-II maintains state information for 2,048 branches**
- **Dynamic branch prediction is highly efficient for looping branches**

Figure 3-4 Features of dynamic branch prediction and folding

To implement dynamic branch prediction, UltraSPARC-IIi maintains a two-bit prediction for every two instructions in the instruction cache. This unique structure ensures that every branch in the I-cache has a prediction. Unlike other processors, UltraSPARC-IIi maintains state information for up to 2,048 branches — a number that far exceeds the needs of most applications. Indeed, branch prediction schemes can have a dramatic effect on telecommunications or other applications that require a deep history for diagnostic purposes.

The two bits in the prediction field comprise a two-bit history of a branch. Initially, UltraSPARC-IIi initializes the state machine to either the *likely taken* or *not likely taken* state. After a branch is encountered, the bits are updated to reflect the most recent activity of the branch, either taken or not taken. This scheme is particularly efficient for looping branches. UltraSPARC-IIi only changes predictions for branches after two mispredictions occur.

Branch Following

Branch following is the ability to rapidly fetch predicted branch targets and helps optimize superscalar processor performance. By reducing fetch latencies, branch following results in a fuller instruction buffer and maximized instruction execution. UltraSPARC-IIi provides single-cycle branch following through a target prediction mechanism in the instruction cache. Because this branch following scheme is extremely fast, UltraSPARC-IIi can speculatively execute deep-code conditional branches.

The dynamic branch prediction and branch folding mechanisms employed by the UltraSPARC-IIi processor in conjunction with Sun WorkShop optimizing compilers provide superior performance with less complexity than other processors. With UltraSPARC-IIi, 88% of integer branches and 94% of floating-point branches are successfully predicted.

Integer Execution Unit

The Integer Execution Unit (IEU) is designed to provide maximize performance while maintaining full software compatibility, minimizing processor architectural changes to host software. The UltraSPARC-IIi Integer Execution Unit incorporates several important features:

- Two ALUs for arithmetic, logical, and shift operations
- An early-finish-detect multi-cycle integer multiplier
- A multi-cycle integer divider
- 8-window register file
- Result bypassing

Instruction Execution Strategy

Load and store operations are of interest when considering the instruction issue strategy of a microprocessor. When a load operation is issued in UltraSPARC-IIi, the IEU reads the required address operands from the register file. If, however, it is determined that any or all operands are contained in the pipeline, the register file read is bypassed, and the data is extracted from the internal pipeline. Once obtained, the address operands are added to obtain the memory address of the item needed. The calculated address is then registered in the cache, and cache access started. Once access is completed, the data is registered in the IEU and written to the register file.

When a store operation is issued, the Integer Execution Unit reads the required operands from the register file. The register read may be bypassed if needed operands are contained in the pipeline. Once the operands are obtained, the store operand register's physical address is computed. From this value, the virtual address of the store in the ALU is determined. Finally, the store data is then registered by the write-through data cache and written to the level 2 cache.

UltraSPARC-IIi attempts to execute all instructions in a single cycle with all instructions issued to the instruction pipeline at the highest possible rate. Because the processor is pipelined, latencies may occur. Once started, execution proceeds at up to four instructions per processor cycle per execution unit.

Load operations may be executed in one cycle, double word stores in two cycles, store operations in a single cycle, double word loads and stores in two cycles and all floating-point stores, both single and double precision, in one cycle.

floating-point Unit

The UltraSPARC-IIi floating-point unit (FPU) is a pipelined floating-point processor that conforms to SPARC-V9 architecture specifications. Its IEEE-compliant design consists of five separate functional units to support floating-point and graphics instructions. The floating-point instruction set includes

UltraSPARC-IIi is capable of executing two floating-point instructions per cycle.

for floating-point exceptions. The FPU performs all floating-point operations and implements a 3-entry floating-point instruction queue to reduce the impact of bottlenecks at the IU and improve overall performance.

Performance

Statistical analysis shows that, on average, 94% of FPU instructions will complete within the typical cycle count. Table 3-2 identifies expected UltraSPARC-IIi FPU performance.

Operation	Throughput (Cycles)	Latency (Cycles)
Add (Single Precision)	1	3
Add (Double Precision)	1	3
Multiply (Single Precision)	1	3
Multiply (Double Precision)	1	3
Divide (Single Precision)	12	12
Divide (Double Precision)	22	22
Square Root (Single Precision)	12	12
Square Root (Double Precision)	22	22

Table 3-2 UltraSPARC-IIi FPU execution times assuming normal floating-point operands and results

I/O Memory Management Unit

Superscalar performance can only be maintained if the IEU can be supplied with the appropriate instructions and data — a job performed by the memory hierarchy. The UltraSPARC-IIi I/O Memory Management Unit (IOM) provides the functionality of a reference MMU and an IOMMU, handling all memory operations as well as arbitration between data stores and memory.

The IOM implements virtual memory and translates virtual addresses of each running process to physical addresses in memory. Virtual memory is a method by which applications are written assuming a full 64-bit address space is

available. This abstraction requires partitioning the logical (virtual) address space into pages which are mapped into physical (real) memory. The operating system in turn translates a 64-bit address into a 44-bit address space supported by the processor. The IOM provides the translation of a 44-bit virtual address to a 41-bit physical address through the use of a Translation Lookaside Buffer (TLB).

The IOM also provides memory protection so that a process can be prohibited from reading or writing to and from the address space of another, guaranteeing memory integrity between processes. Access protection is also supported to ensure that any given process does not gain unauthorized access to memory. For example, a process will not be allowed to modify areas that are marked as read-only or reserved for supervisory software.

Finally, the IOM performs the arbitration function between I/O, D-cache, I-cache, and TLB references to memory. At any given time, a contention for memory access may arise between an I/O access involving the bus as well as internal accesses requested by I-Cache, D-Cache, and TLB references.

- **Implements virtual memory**
- **Provides virtual to physical address memory translation**
- **Provides memory protection**
- **Performs arbitration between Input/Output, Data Cache, Instruction Cache, and Translation Lookaside Buffer references to memory.**

Figure 3-5 UltraSPARC-IIi I/O memory management primary functions

Translation Lookaside Buffer

The UltraSPARC-IIi Translation Lookaside Buffer is a 64-entry, fully associative cache of page descriptors. It caches virtual to physical address translations and the associated page protection and usage information. The translation of virtual to physical addressing is handled via a table walk (cache lookup) of the TLB. A virtual address to be translated by the MMU is compared to each entry in the TLB. If the address to be translated is found in the TLB, the table walk is complete. If the address is not found, the MMU traps to software for TLB miss processing.

Memory Controller Unit

UltraSPARC-IIi employs a Memory Controller Unit (MCU). In essence, the MCU implements the function of a “traffic cop”, controlling and prioritizing access to main memory. Specifically, all transactions to the system, such as E-cache misses, interrupts, and writebacks are handled by the MCU. Additionally, the MCU manages I/O transactions through the processor to and from memory. These transactions include reads and writes from the processor to PCI devices, and DMA reads and writes from the devices to the processor.

All transactions to the DRAM and UPA64S subsystem are also handled by the MCU. The MCU communicates with the memory system at a frequency lower than UltraSPARC-IIi frequency (the ratio is typically 1/4). As factory DRAMs become available, the processor clock can be moved to one-third the frequency, depending on the memory employed. The MCU provides support for multiple outstanding load and writeback requests to the Ultra Port Architecture (UPA) bus.

Load-Store Unit

The LSU is responsible for generating the virtual address of all loads and stores, for accessing the data cache, for decoupling load misses from the pipeline through the load buffer, and for decoupling the stores through a store buffer. One load or one store can be issued per cycle. To further optimize data stores, a store compression capability allows two or more stores to be “compressed” together if they are in the same 8-byte block, so that a single data transfer occurs between UltraSPARC and the second-level cache. This frees up the data bus allowing load misses and instruction misses to be processed more rapidly.

Cache Architecture

Most high-performance microprocessors use cache to reduce bus traffic and increase system throughput. Cache stores contain copies of part of a memory image. The choice of whether to update or invalidate copies of modified blocks is called the *cache consistency protocol*. These consistency protocols ensure that copies of data remain consistent with the copies in main memory. Consistency can be supported by write-invalidate, write-through, and competitive caching algorithms. UltraSPARC-IIi employs large, separate instruction and write-through data caches.

Data Cache

The UltraSPARC-IIi Data Cache is a 16 KB direct mapped, software selectable write-through non-allocating cache that is used on Load or Store accesses from the CPU to cacheable pages of main memory. It is a virtually-indexed and virtually-tagged cache. The D-cache is organized as 512 lines with two 16-byte sub-blocks of data per line. Each line has a cache tag associated with it. On a data cache miss to a cacheable location, 16 bytes of data are written into the cache from main memory.

The D-cache is a virtually tagged cache. Virtual addresses delivered to the data cache are used when attempting to find the required word in the cache. The least significant part of the virtual address is used to access one line of the cache (direct mapped) that may contain the required word. The most significant part of the virtual address is then compared with the tag address bits for a possible match, or cache hit. This scheme ensures that cache misses are quickly detected, and that addresses are translated only on a cache miss.

For UltraSPARC-IIi, a significant advantage is gained through the use of a virtually tagged cache. Physically tagged caches require the cache to be compared with the physical address from the MMU. In such environments cache miss detection may be a bottleneck in the MMU. While physical caches are appropriate for certain environments, virtually tagged caches, such as those employed in UltraSPARC-IIi, increase performance through accelerated cycle cache miss detection.

Instruction Cache

The Instruction Cache is a 16 KB, two-way set-associative cache used on instruction fetch accesses from the CPU to cacheable pages of main memory. It is organized as 512 lines of 32 bytes of data, with each line having an associated cache tag. On an I-cache miss to a cacheable location, 32 bytes of data are written into the cache from main memory.

Like the data cache, it is physically indexed and physically tagged. The set is predicted as part of the next field so that only the 13 index bits of an address are necessary to address the cache, which matches the minimum page size. The instruction cache returns up to 4 instructions from an 8 instruction-wide line.

External Cache

It is the responsibility of the ECU to efficiently handle I-cache and D-cache misses. The ECU can handle one access every other cycle to the external cache. However, the combination of the load buffer and the ECU is fully pipelined. This low latency can effectively make the external cache a part of the pipeline. For programs with large data sets, this means data can be maintained in the external cache and instructions scheduled with load latencies based on the E-Cache latency. Floating-point applications can use this feature to effectively “hide” D-Cache misses. The size of the external cache can be 256 KB, 512 KB, 1 MB, or 2 MB, where the line size is always 64 bytes.

The ECU in UltraSPARC-IIi also provides support for the 2-2-2 and 2-2 modes of external cache operation. The interface between the processor and SRAM is run at an integer multiple to the processor.

In 2-2-2 mode, accesses to the external cache are pipelined and take six cycles (two to send the address, two to access the SRAM array, and two to return the E-cache data), returning 16 bytes of instructions or data per cycle. (In 2-2 mode, the latency is four cycles.) These low latencies can effectively make the external cache a part of the pipeline. For programs with large data sets, this means data can be maintained in the external cache and instructions scheduled with load latencies based on the E-Cache latency. Floating-point applications can use this feature to effectively “hide” D-Cache misses.

The ECU is a non-blocking cache, providing overlap processing during load and store misses. For instance, stores that hit the E-Cache can proceed while a load miss is being processed. The ECU is also capable of processing reads and writes indiscriminately without a costly turnaround penalty (only 2 cycles are needed). Snoops are also handled by the ECU.

Block loads and block stores load or store a 64-byte line of data from memory to the floating-point register file. These are processed efficiently by the ECU, providing high-transfer bandwidth without polluting the internal or external cache.

Graphics Unit

One of the most talked about trends today is the convergence of multimedia information and advanced imaging into the enterprise network. Video conferencing is another well cited application that offers clear benefit to the everyday business environment. A processor architecture can go a long way

toward promoting optimal performance for advanced multimedia applications, 2-D and 3-D imaging, image manipulation and recognition, animation, and virtual reality applications.

Sun's UltraSPARC processors are the industry's first general-purpose processors that deliver 64-bit compute power combined with the throughput needed for the advanced graphics and real-time video applications (figure 3-6). Able to decompress and manipulate video information with broadcast video resolution in real time, UltraSPARC-IIi can eliminate the need for a dedicated video processor.

- **64-bit data paths allows 8-, 16-, and 32-bit fixed point data to be operated on in parallel**
- **Enables video compression and decompression**
- **Accelerates graphics by handling computation up to 10 operations per cycle**
- **Uses block loads and stores to quickly manipulate images**

Figure 3-6 The UltraSPARC-IIi Graphics Unit provides strong support for advanced graphics and multimedia operations requiring high performance

The Graphics Unit relies on integer registers for addressing image data, and floating-point registers for manipulating that data. This division of labor enables UltraSPARC-IIi to make full use of all available internal registers, maximizing graphical throughput (figure 3-7).

The VIS™ Instruction Set

Specialized instructions that execute complex multimedia and networking operations can dramatically increase a processor's throughput. The VIS™ Instruction Set enable the execution of complex graphics operations that typically require dozens of clocks to be performed in a single clock cycle. VIS instructions can be categorized as follows:

- *Image processing*

  VIS includes image processing instructions to accelerate the scaling and rotating of images, pixel interpolation, filtering, alpha blending, and volumetric rendering. In particular, the *Partitioned Add/Subtract* instruction allows the addition or subtraction of four 16-bit or two 32-bit values to be completed in one cycle. The *Partitioned Multiply* instruction allows four 16-bit values to be multiplied by four 8-bit values to be multiplied in a single

clock cycle. *Array Addressing* converts 3-D fixed point addresses into blocked addresses. Useful for slicing through 3-D volumetric data, this feature can reduce 24 image processing instructions down to a single instruction.

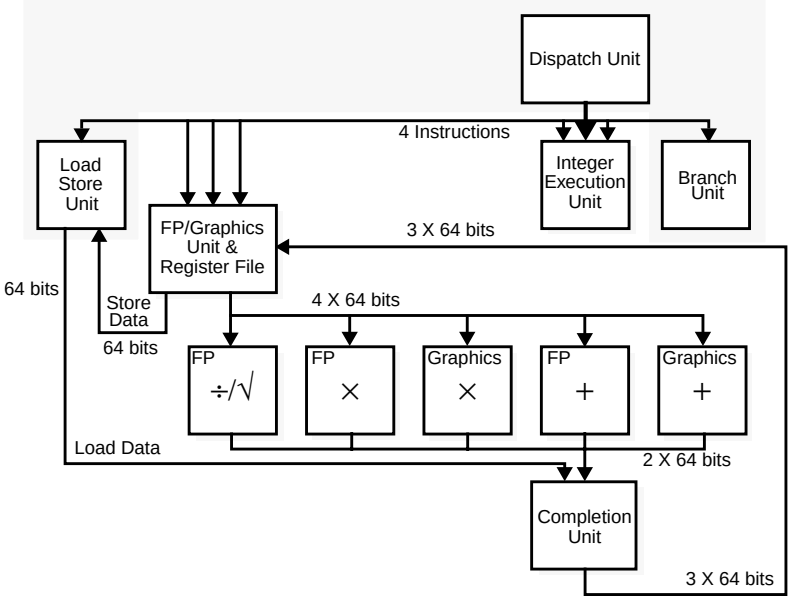

Figure 3-7 The Graphics Unit provides tremendous capabilities yet consumes less than 3% of UltraSPARC-IIi's die area

- *Video compression*

  VIS incorporates special instructions to accelerate video compression and decompression, including the H.261, JPEG, MPEG1, and MPEG2 algorithms. Other VIS instructions perform efficient color space conversion, eliminating the need for separate external hardware.

  The intrinsic value of VIS can be exposed by examining the benefit of motion estimation on MPEG compression. Motion estimation compares the value of blocks of pixels between two video frames and computes a motion vector. This process typically requires eight pixel subtracts, eight absolute values, eight pixel adds, eight loads, eight aligns, and eight additional adds. The inclusion of the motion estimation instruction reduces these 48 instructions to one.

- *Pixel format and conversion*

  Pixel format and conversion instructions allow the processor to operate directly on pixel data. The pixel data can be stored in 8-bit or 16-bit values. In particular, the *Pixel Expand* instruction allows the conversion of four 8-bit integers to four 16-bit integers yet stores the result as one 64-bit word. The *Pixel Pack* instruction converts four 16-bit or two 32-bit values to four 8-bit or two 16-bit values, and also stores the results as one 64-bit word. Use of these instructions can result in typical pixel manipulation instructions typically taking up to 34 instructions being reduced to one, greatly improving application performance.

- *Data transfer and animation speed-up*

  VIS takes advantage of the block load/store features of the UltraSPARC-IIi processor that allows 64-byte loads and stores to occur directly from the CPU to main memory, from memory to the video display (screen), or from the screen to main memory. This unique ability allows graphics workstations to sustain higher throughput, eliminating flicker and smoothing animation.

PCI Bus Controller

Unlike its predecessors, the UltraSPARC-IIi module incorporates a PCI bus controller. Compliant with version 2.1 of the PCI specification, the PCI Bus Controller enables UltraSPARC-IIi to interface directly with the PCI bus. It is optimized for 16-, 32-, and 64-byte transfers, and can support up to four PCI devices. While the PCI address space is non-cacheable for CPU references, coherent DMA is supported. All reads and writes to memory from PCI are cache coherent, and interrupt handling is synchronized to the completion of all prior DMA writes.

The PCI Bus Controller forwards transactions bidirectionally between these primary and secondary PCI buses. Key features of the PCI Bus Controller include support for PCI features:

- Single 64-byte DMA read/write buffer, single 64-byte PIO read/write buffer
- 64-bit addressing (dual address cycle) for DMA bypass
- Fast back-to-back cycles as a DMA target
- Ability to generate memory, I/O, and configuration read and write cycles
- Ability to generate special cycles
- Ability to receive memory cycles

- Peer-to-peer DMA on a single segment
- Optional external arbiter

The PCI interface of the UltraSPARC-IIi processor can be used directly or expanded using one or more PCI bridges, such as the Advanced PCI Bridge (APB™) available from Sun.

UltraSPARC-IIi Enhancements

Significant enhancements have been made to the memory system of the UltraSPARC-II and -IIi over the earlier UltraSPARC-I. One is support for the SPARC V9 *Prefetch* instruction. Prefetches primarily address floating-point vector code, in which the compiler can accurately schedule the prefetch of data sufficiently ahead of its usage, and in which execution is bounded by E-cache miss throughput.

Also key in the UltraSPARC-IIi is its ability to support multiple outstanding read-miss requests from different internal sources. At any given time, there may be any combination of one Instruction-fetch miss, one Store miss, and up to three Load / Prefetch misses outstanding to the system, so long as the total does not exceed three. To balance the support for the three outstanding reads, the UltraSPARC-II design also supports up to two outstanding dirty-victim writebacks. For backward compatibility with UltraSPARC-I, UltraSPARC-IIi can be configured to handle just one outstanding miss request and one writeback.

The performance impact of these changes can be significant to commercial applications, like on-line transaction processing and scientific applications that are floating-point intensive. Table 3-3 shows the relative performance of UltraSPARC-IIi relative to UltraSPARC-I with these new features.

		Outstanding Requests to Memory	
		Single	Multiple
Prefetch Enabled	No	1.0	1.0
	Yes	1.6	2.1

Table 3-3 Performance of UltraSPARC-IIi relative to UltraSPARC-I with and without the Prefetch and outstanding memory requests enabled.

UltraSPARC-IIi offers higher levels of integration than its predecessor, the UltraSPARC-II. With integrated PCI and memory controllers, the UltraSPARC-IIi processor is able to provide lower cost and higher performance in a scalable architecture.

Integrating UltraSPARC-IIi 4

A complete UltraSPARC-IIi subsystem includes an UltraSPARC-IIi processor, synchronous SRAM components for E-cache tags and data, and a series of transceivers. The transceivers transfer data bidirectionally between the UltraSPARC-IIi processor and the memory DIMMs (figure 4-1). Typically, the E-cache is divided into two parts: tags and data, both of which can be implemented using commodity RAMs. Separate address and data buses are provided from/to the tag and data RAMs to increase performance.

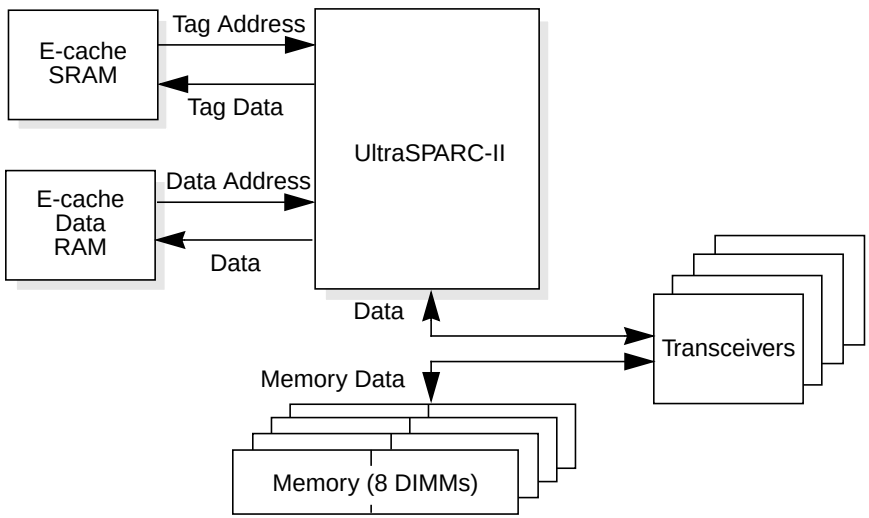

Figure 4-1 The UltraSPARC-IIi subsystem system interface

UltraSPARC-IIi Based System Integration

UltraSPARC-IIi is designed to be integrated easily into low cost PCI-based systems like Sun's Ultra™ 5 and Ultra 10 workstations. Utilizing the UltraSPARC-IIi processor module, transceiver and DRAM memory, an Advanced PCI Bridge (APB), and a UPA64S interface, UltraSPARC-IIi provides the high levels of integration necessary to produce low cost platforms (figure 4-2).

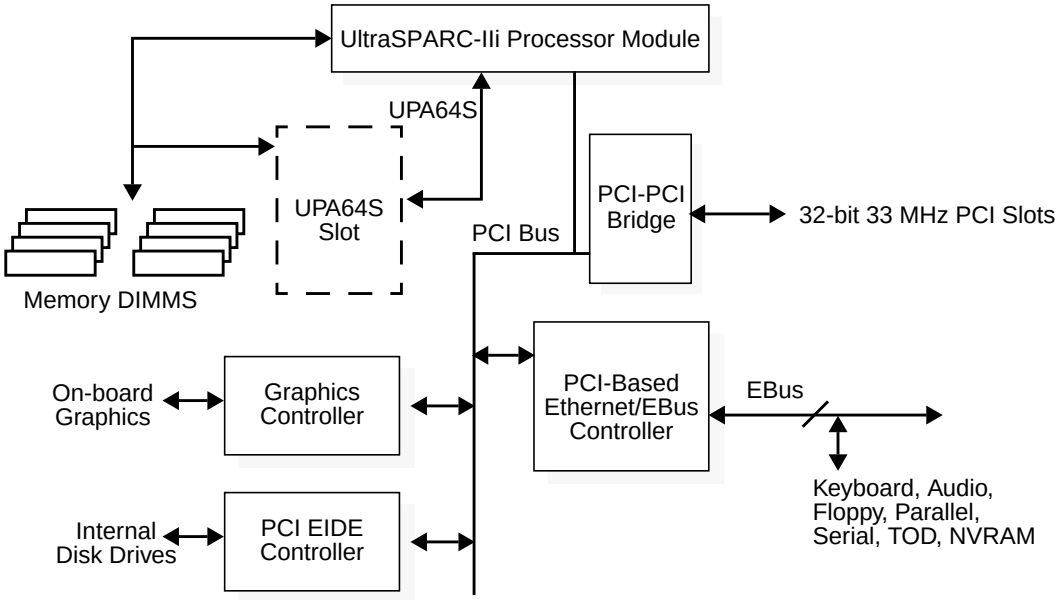

Figure 4-2 UltraSPARC-IIi reference platform

Transceivers and DRAM Memory

The UltraSPARC-IIi processor communicates with DRAM memory through a series of transceivers. The transceivers transfer data bidirectionally between the 72-bit UltraSPARC-IIi memory data bus and the 144-bit DIMM memory data bus. The DIMMs cycle data at a maximum frequency of 37.5 MHz. Four DIMM pairs can be utilized in UltraSPARC-IIi systems, for a maximum of 256 MB

using standard 168 pin JEDEC DIMMs with 16 Mbit DRAM (up to 1 GB using 64 Mbit DRAM). Designed with a 72-bit wide memory data bus, the UltraSPARC-IIi memory offers the following characteristics:

- High performance CMOS silicon gate process
- 72-bit DRAM data bus (8-bit ECC per 64 data bits)
- 3.3 V compatible
- Low power, 9 mW standby, 1,800 mW active (typical)

Advanced PCI Bridge

The UltraSPARC-IIi PCI interface can be used directly or expanded using one or more PCI bridges, including the Sun Advanced PCI Bridge (APB). The 32-bit version 2.1 compliant interface from UltraSPARC-IIi to its I/O subsystems can run at either 33 MHz or 66 MHz at 3.3 V.

The APB can be used to expand the system. Each APB enables system expansion from one 66 MHz 32-bit PCI bus to two 32-bit 33 MHz PCI buses, as well as support for 5 V PCI cards. In addition, APB supports 64-byte write posting and data prefetching, delivering higher throughput than a single 33 MHz bus can provide.

UPA64S Interface

UltraSPARC-IIi has a direct interface with the UPA64S interface, a slave-only interface protocol that can be used for any high bandwidth control or data transfers between the processor and a dedicated subsystem, such as the high performance graphics boards found in Sun workstations. The UPA64S interface runs up to 100 MHz (1/3 the processor clock rate). Transfers to and from the UPA64S interface are fully synchronous.

The Ultra Port Architecture Interconnect

Needing to ensure the correct balance between the high performance computational and graphic subsystems in its Ultra systems, Sun engineers developed UPA, a new, cache-coherent, processor-memory interconnect. As implemented in UltraSPARC-IIi based systems, the principal advantages of the UPA over existing interconnects are significant:

- Scalable bandwidth through support of multiple bused interconnects for data and addresses

- Packet switched for better bus utilization
- Higher bandwidth
- Precise interrupt processing
- Low latency memory accesses
- Buffered cross bar memory interface for increased bandwidth and greatly improved scalability
- High throughput paths to memory
- High performance graphics support with two-cycle single-word writes on the 64-bit UPA interconnect
- Better economy through centralized coherence and memory controller functions

Incorporating many features previously found only on mainframes and high performance servers, the UPA interconnect architecture incorporates several innovations designed to meet the ambitious requirements of Ultra platforms. Unlike conventional cache-coherent systems which use a globally shared snooping address bus, the UPA interconnect architecture relies on point-to-point packet switched messages from a centralized system controller to maintain cache coherence. Packet switching provides for much better bus bandwidth utilization by removing the latencies commonly associated with pure transaction-based designs.

Unlike other directory-based systems which maintain the coherence states for each data block in main memory, requiring a read-modify-write penalty for every read transaction that reaches main memory, the UPA interconnect maintains a duplicate set of all cache tags in the system and performs duplicate tag lookup and main memory initiation in parallel pipelines for each coherent transaction. This departure from conventional approaches permits minimum latency on cache misses, and effective pipelining in the interconnect allows maximum, and often “bubbleless” utilization of address, datapath, and main memory.

UPA uses a centralized system controller, which removes the need to place cache coherence logic on the processor and DMA devices, considerably simplifying the implementation. Other key features of the UPA interconnect include:

- Independent address and data
- Independent operation of I/O and processor buses
- Use of precise interrupts
- Streaming buffer
- Major buses protected by parity or ECC

The UPA clock operates at up to 100 Mhz in UltraSPARC-IIi based systems. UPA speed is coupled to the processor clock speed, with a maximum clock rate of 100 MHz for processor speeds that are exactly divisible by 100 (e.g., a processor clocked at 300 MHz will permit the maximum UPA clock rate of 100 MHz. Processor speeds not exactly divisible by 100 will result in lower UPA speeds (e.g., a 270 MHz processor yields a UPA speed of 90 MHz.)

Interaction with the External Cache

External cache access is closely coupled to a processor's pipeline. In UltraSPARC-IIi, full access to the E-cache is supported, making it appear to the system like a large data cache. The E-cache consists of *Tag RAM* and *Data RAM*. Tag RAM contains the physical tags of all cache lines, along with state information. Data RAM contains the actual data for each cache line.

UltraSPARC-IIi Support for Operating Systems

Applications performance is often heavily dependent on the operating system, making the need for processor-specific optimization critical. Today's high performance operating systems and RTOS environments require fast context switching, system calls, and efficient trap handling to mitigate wasted processing time. UltraSPARC-IIi was designed with these needs in mind:

- *Fast traps*

  UltraSPARC-IIi provides fast traps by including scratch registers and multiple levels of nested traps. These features minimize operating system overhead when resolving trap conditions, and eliminate the need for detailed checking software in trap routines.

- *Direct access to privileged registers*

  UltraSPARC-IIi permits the operating system to directly access its privileged registers, enabling more efficient operating system operation.

- *Accelerated operating system functions*

  UltraSPARC-IIi provides high bandwidth block load/stores that accelerate common operating system functions. Multiple page sizes are also supported, as is memory mapping needed for large contiguous memory regions.

UltraSPARC-IIi Support for Optimizing Compilers

UltraSPARC-IIi supports several features that permit higher application performance without sacrificing binary compatibility with older SPARC processors. UltraSPARC-IIi can issue up to four instructions per cycle, enabling scheduling techniques such as software pipelining to achieve up to two floating-point operations per cycle. In addition, all UltraSPARC-IIi load instructions are non-blocking, permitting the pipeline to continue issuing instructions even after a cache miss.

The UltraSPARC-IIi processor supports a variety of new SPARC-V9 instructions and features that compilers can exploit to generate highly optimized code:

- Non-faulting loads, permitting compilers to move load instructions ahead of conditional control structures that guard their use.
- 32 additional double-precision floating-point registers that allow the compiler to keep more intermediate operands in registers for fast access.
- Three additional floating-point condition code registers that permit compilers to more freely schedule floating-point instructions.
- Predicted branches allow the compiler to notify the hardware about the most likely direction of the branch, lowering the average penalty for incorrect branch prediction.
- Conditional move instructions permit the compiler to eliminate many branch instructions, accelerating superscalar execution.

UltraSPARC-IIi Support for Embedded Systems

The UltraSPARC-IIi processor is designed to meet the diverse needs of embedded solutions in the networking, telecommunication, and imaging arenas. In order to ensure a full range of solutions for embedded systems developers, Sun offers world class real-time operating systems like Sun's ChorusOS (formerly Chorus ClassiX) and Wind River's VxWorks, as well as UltraSPARC-IIi evaluation and design kits, and a reference platform.

UltraSPARC-IIi Versus Implementations of Other Architectures

5

The microprocessor industry is extremely competitive, and with such a large number of high performance designs to choose from, users must consider many factors when selecting a processor architecture. A series of interrelated criteria should be evaluated, including: cost, performance, scalability, open versus proprietary architectures, on-chip functions, advanced functionality, software availability, development environments, and product availability. This section discusses these and other features of the Hewlett Packard PA-8000, Cyrix MediaGX, MIPS R10000, Motorola / IBM PowerPC 603e processors, and Intel Pentium Pro and Pentium II, and compares them to UltraSPARC-IIi. These comparisons are provided for illustrative purposes, and reflect the information available at the time of writing.

Intel Pentium II

The Pentium II processor is the latest incarnation of Intel's highly popular X86 architecture. Implemented in a 0.35 micron CMOS process, models with clock speeds of up to 300 Mhz have been announced.

Optimized for 32-bit environments, the Pentium II uses a 12-stage decoupled super pipelined implementation, doing less work per stage, but through a larger number of stages than earlier Pentium processors. By reducing its pipestage time by 33% from the Pentium, Intel claims that the Pentium II can utilize a 33% faster clock speed for an equivalent manufacturing process. New Single Edge Contact (S.E.C.) packaging was introduced with the Pentium II, enabling better frequency operation and easier upgradeability.

The Pentium II employs 16 KB non-blocking, on-chip level one instruction and data caches, and a 512 KB *on-package* secondary cache. Cache access is through a dedicated cache bus, with system I/O being directed through an independent bus.

The dispatch execute unit of the Pentium II uses a five-port reservation station to prepare instructions for execution. Able to schedule up to five instructions per clock (one for each port in the reservation station), three per clock is more typical. The Pentium II has two integer ALUs, a single floating-point Unit, two address generation units, and a branch unit attached to the reservation station.

Scalable up to four processors using on-chip glue logic, the Pentium II also employs so-called dynamic execution, which is essentially the speculative execution approach employed by both HP and MIPS with their PA-8000 and R10000 processors. Dynamic execution allows the use of heuristics like branch prediction and data flow analysis to predict the most likely next instruction and to fetch and execute it out of order. Sophisticated techniques are employed to ensure that results and improperly predicted branch paths do not interfere with correct results.

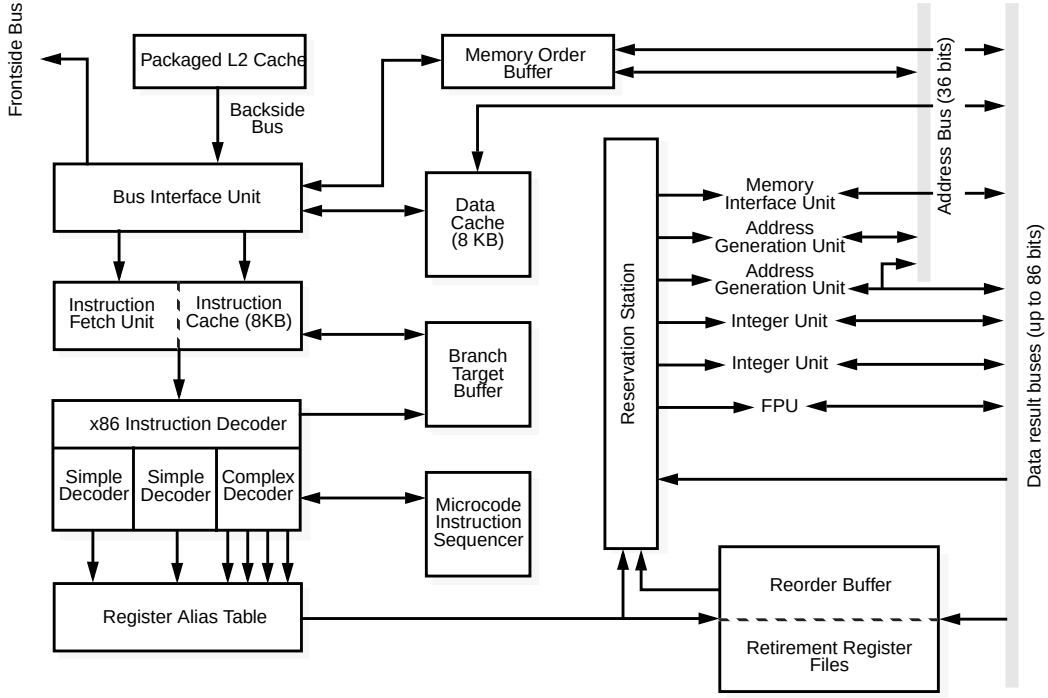

Figure 5-1 Intel II high-level architecture

Differences Between Pentium II and UltraSPARC-IIi

The bottom line in comparing the Pentium II against the UltraSPARC-IIi is performance — UltraSPARC simply outperforms the Pentium II in a variety of measures. While the two processors exhibit comparable SPECint95 performance when running at the same frequency and utilizing equivalent level 2 caches, the UltraSPARC-IIi outperforms the Pentium II in all-important floating-point performance by a significant amount. In other important areas, UltraSPARC-IIi is faster than the Pentium II in system I/O bandwidth and graphics by a factor of two or three.

Many point out the advantage of the large base of available software for the Intel x86 line, but because the Pentium II has been so carefully optimized for operation in 32-bit environments, it runs more slowly when running 16-bit applications — which are a significant percentage of the existing software base.

Intel has implemented their MMX graphics support for the Pentium Series. Even with this change, MMX functionality falls short of UltraSPARC's VIS Instruction Set, with no support for motion estimation, discrete cosine transforms, pixel masking, or 3-D rendering. In addition, UltraSPARC's set of 32 registers support the use of 4x4 matrices (commonly used with digital filtering) without extra memory accesses. Finally, the block load and store instructions of UltraSPARC allow it to move data far faster than Pentium.

- **Pentium II performance is not on par with the UltraSPARC-IIi in several measures, including floating-point, graphics, and system bandwidth.**
- **MMX graphics functionality and performance is inferior to UltraSPARC's VIS Instruction Set. VIS also includes instructions specific to network performance.**
- **Pentium II's address space is only 4 GB — inadequate for the largest commercial and technical applications. Compare with UltraSPARC's 2 TB address space.**

Figure 5-2 Advantages of UltraSPARC-IIi over the Pentium II processor.

Cyrix MediaGX

Cyrix Corporation has garnered significant interest from the personal computing industry with its highly integrated, low cost MediaGX processor. The MediaGX is a 180 MHz 64-bit data bus, X86-compatible processor that

incorporates an execution core, load/store unit, floating-point unit, branch target buffer, 16 KB unified writeback cache, on-board graphics accelerator, display controller, and a PCI I/O controller (figure 5-3).

Targeted at the PC marketplace, MediaGX uses a hardware and software combination to provide VGA quality graphics. Utilizing the proprietary Cyrix Virtual System Architecture (VSA), MediaGX off-loads some graphics support to a VSA graphics device driver that takes advantage of the processor's graphics pipeline.

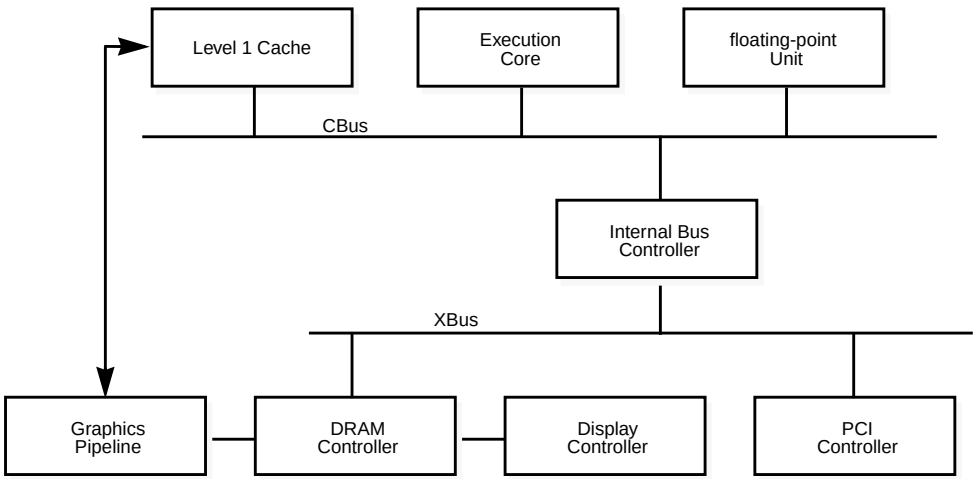

Figure 5-3 Cyrix MediaGX high-level architecture

Differences between MediaGX and UltraSPARC-IIi

There are several functional differences between the Cyrix MediaGX and UltraSPARC-IIi. Running at 180 MHz, MediaGX is significantly slower than the 300 MHz UltraSPARC-IIi. Designed as a virtual video card, MediaGX cannot compete in general purpose computing environments, providing a very low price/performance ratio.

MediaGX is an integrated processor that targets display operations. UltraSPARC-IIi is a highly integrated processor for general purpose computing environments, including high performance graphics applications. With limited

on-chip integration, MediaGX requires a more complex system integration package than the UltraSPARC-IIi module, enabling UltraSPARC-IIi to compete well in low cost computing environments.

- **UltraSPARC-IIi is a high performance 300 MHz processor — MediaGx is significantly slower at 180 Mhz**
- **SPARC is an open architecture with multiple sources — MediaGX is X86 compatible**
- **UltraSPARC-IIi is a low cost general purpose processor — MediaGX is targeted as a virtual video card on a chip**
- **UltraSPARC-IIi is network and multimedia ready — MediaGX offers limited multimedia support**
- **UltraSPARC-IIi is a highly integrated processor — MediaGX requires greater system integration efforts**

Figure 5-4 Advantages of UltraSPARC-IIi over the Cyrix MediaGX processor

Hewlett Packard PA-8000

The Hewlett Packard PA-8000 is the first product based upon the PA 2.0 architecture. A 64-bit processor with a superscalar architecture, the PA-8000 can execute four instructions per cycle with its two integer ALUs, two shift/merge units, two floating-point units, two divide/square root units, and two load/store units (figure 5-5).

Processor hardware can schedule instruction execution through the use of a Instruction Reorder Buffer, which allows execution of instructions as the data become available, regardless of the order in which they were fetched. Connections to off-chip, single-level, one megabyte instruction and data caches are provided. Cache lines can be loaded in to the buffer at the rate of four instructions per cycle. Data cache is dual ported, with each port able to process a double word per cycle.

The PA-8000 implements speculative fetching of data, with hardware support for instruction path prediction.

The PA-8000 is designed to work with a memory and I/O bus called Runway. Runway operations at 125 MHz, yielding 768 MB/sec. cache-line-sized data transfers. Up to four PA-8000 processors can share the Runway bus.

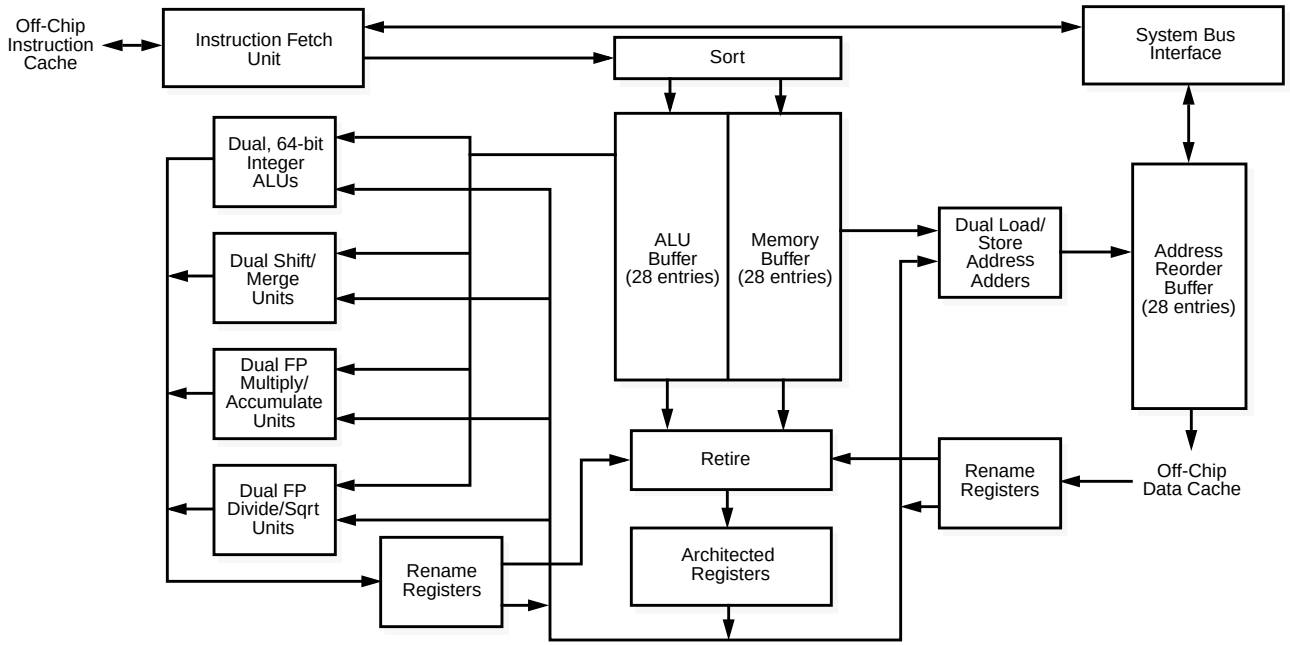

Figure 5-5 The HP PA-8000 architecture

Differences Between the PA-8000 and UltraSPARC-IIi

Because of problems with the fabrication process in the PA-8000, first level caches must be implemented off-chip. This has forced the use of expensive, 4.5 nsec. SRAMs in order to maintain promised performance. UltraSPARC-IIi implements its first level cache on-chip, ensuring very high performance and lower cost.

While the out-of-order execution capabilities can improve the performance of existing code, a simple recompilation can remove this advantage. The benefits of out-of-order execution could well be nullified by the greatly increased complexity of the chip design and the resultant impact on economics and availability. UltraSPARC-IIi uses state-of-the-art fabrication techniques, but also acknowledges the practical considerations for process yield, cost, and schedule.

Finally, the Runway bus only provides a maximum sustained throughput of 768 MB/sec., compared with the 1.6 GB/sec of UPA.

- **UltraSPARC is a high-volume processor — the HP PA-8000 is not**
- **UltraSPARC enjoys the installed base of the entire SPARC product line — HP PA-8000 has a relatively small installed base**
- **UltraSPARC is an open architecture — the PA-RISC architecture is proprietary**
- **UltraSPARC supports multimedia applications via the VIS Instruction Set — HP PA-8000 does not**
- **The technology used to produce the PA-8000 is complex, and has increased cost and reduced availability without substantial benefits to users**

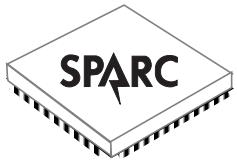

Figure 5-6 Advantages of UltraSPARC-II over the HP PA-8000

MIPS R10000

The R10000 Microprocessor from MIPS Technologies is a 4-way super-scalar architecture capable of fetching and decoding four instructions per cycle which are then appended to one of three instruction queues (integer, floating-point, or address.) Each queue can perform dynamic scheduling of instructions. While instructions are fetched and decoded in order, they can be executed and completed out-of-order. This allows to R10000 to have up to 32 instructions in the processed of being executed. The R10000 has five independently operating execution units — two integer ALUs, two floating-point units, and a load store unit for generating addresses.

The integer and floating-point throughput is achieved through the use of wide, dedicated data paths, and 32 KB on-chip instruction and data caches. Second level cache is implemented off chip, but can be accessed through a 3.2 GB/sec channel called the Avalanche bus. The R10000 Microprocessor implements the MIPS IV instruction set architecture. MIPS IV is a superset of the MIPS III instruction set architecture and is backward compatible. Clocked with an internal frequency of 195 MHz, the R10000 Microprocessor delivers peak performance of 10.7 SPECint95 and 19.0 SPECfp95.

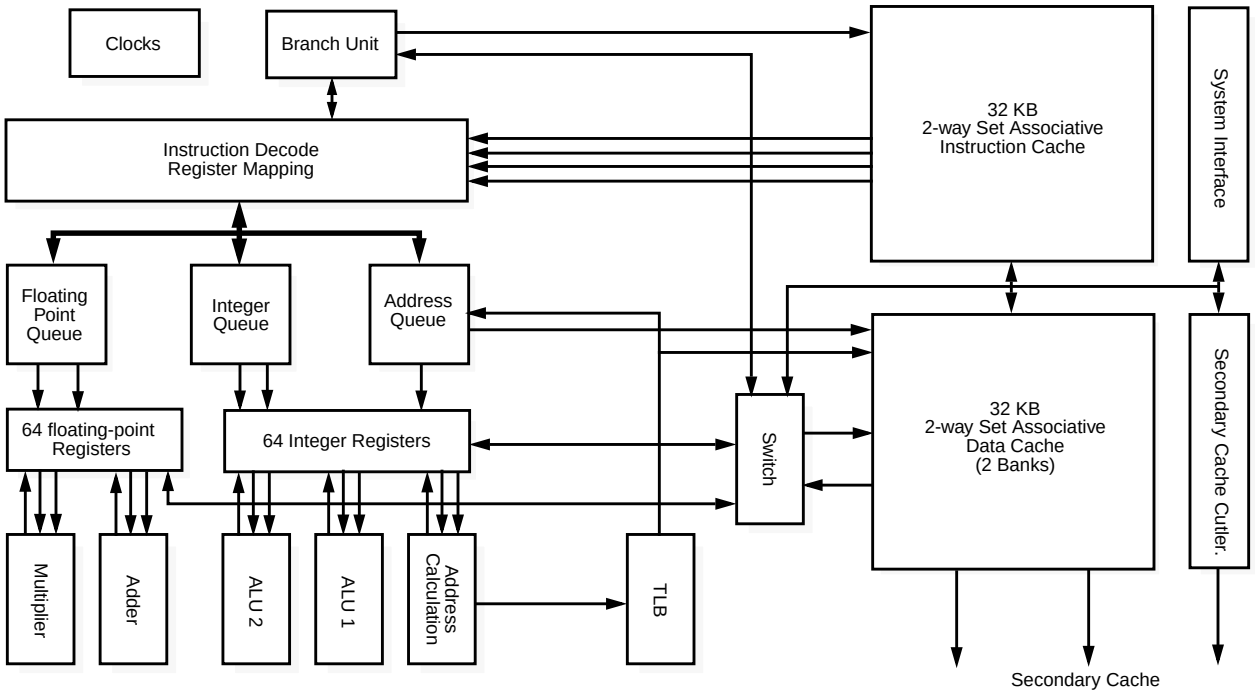

Figure 5-7 MIPS R10000 high-level architecture

Differences Between the MIPS R10000 and UltraSPARC-IIi

Like the PA-8000, while the out-of-order execution capabilities can improve the performance of existing code, a simple recompilation can remove this advantage. The benefits of out-of-order execution could well be nullified by the greatly increased complexity of the chip design and the resultant impact on economics and availability. UltraSPARC-IIi uses state-of-the-art fabrication techniques, but also acknowledges the practical considerations for process yield, cost, and schedule.

- **UltraSPARC-IIi is a high-volume processor — the R10000 is a low volume processor**
- **UltraSPARC-IIi enjoys the installed base of the SPARC product line — the R10000 has a small installed base**
- **UltraSPARC-IIi is binary compatible with other SPARC processors — the R10000 has narrow application support**
- **UltraSPARC-IIi is network and multimedia ready — the R10000 provides limited multimedia support**
- **The technology used to produce the R10000 is complex, and has increased cost and reduced availability without substantial benefits to users**
- **Sun offers the robust VIS instruction set — MIPS has not implemented any multimedia instructions to date**

Figure 5-8 Advantages of UltraSPARC-IIi over the MIPS R10000 processor

PowerPC 603e

In an effort to increase marketshare, IBM, Apple, and Motorola joined forces to produce the PowerPC processor architecture. A healthy competitor to UltraSPARC-IIi, the PowerPC 603e is a 300 MHz, superscalar, 64-bit, low power processor targeted for both consumer electronics and computers, including home entertainment and educational systems with audio, video, multimedia, and graphics support. The PowerPC 603e consists of five independent pipelined execution units (integer, floating-point, load / store, system register unit, and branch processing), a 16 KB data cache, a 16 KB instruction cache, a generous register set, and a memory management unit. A variety of chip sets are available to support EISA, ISA, PCI, and VME buses to complete motherboard designs (figure 5-9).

The PowerPC 603e offers a variety of features to improve desktop systems:

- Five independent pipelined units enable five instructions to be in process at any given time
- Separate memory management units for instructions and data
- Multiplexed 64-bit data and 32-bit address bus
- Dynamic branching predictions
- Eight deep instruction queue
- Symmetric multiprocessing (SMP) capabilities

- Software controllable power savings mode
- Intended support from a variety of operating system vendors

Differences Between PowerPC 603e and UltraSPARC-IIi

There are several functional differences between the PowerPC 603e and UltraSPARC-IIi. Running at 300 Mhz, the PowerPC 603e is less efficient than the 300 MHz UltraSPARC-IIi as testified by their respective performance numbers. The PowerPC 603e produces a SPECint95 of 7.4 and SPECfp95 of 6.1, while UltraSPARC-IIi boasts higher performance (SPECint95 >12.1 peak and SPECfp95 >12.9 peak).

Because it was designed for incorporation into systems from a disparate group of vendors, chip sets must be used to integrate the PowerPC. Depending on the target environment, different chip sets need to be used for a specific system. The highly integrated UltraSPARC-IIi by comparison, utilizes on-chip caches. Combined with its advanced processor design and UPA architecture, UltraSPARC-IIi requires little to complete its chip set, simplifying its integration into systems.

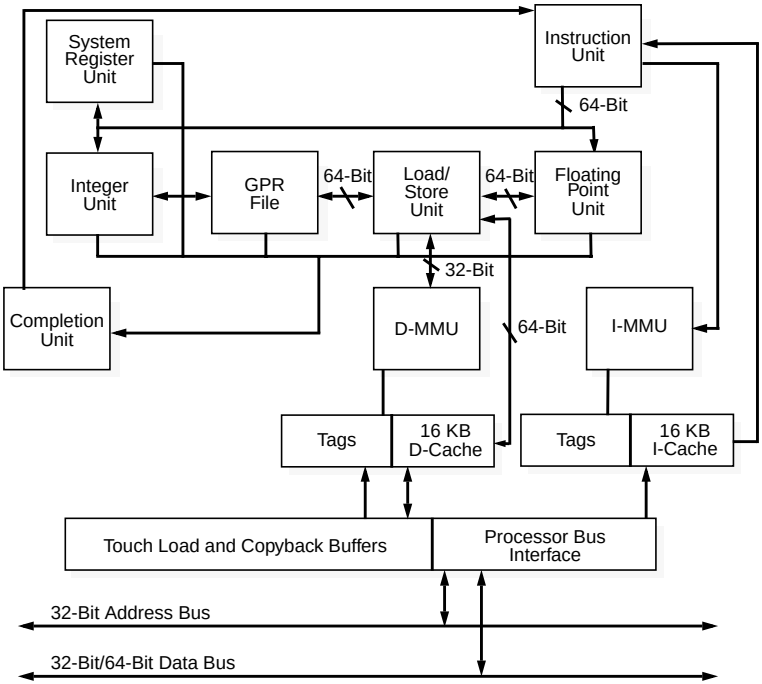

Figure 5-9 PowerPC 603e high-level architecture

With target markets of both desktop and portable computing enjoyed by IBM and Apple, the PowerPC was intended to support applications which currently run on a wide range of operating systems, including MS-DOS, OS/2, Solaris, Apple's System 7, Taligent, Microsoft Windows NT, and AIX. Recently, both IBM and Microsoft announced their intention to halt the implementation of WindowsNT on the PowerPC. Taligent has been dissolved, and the future of the Apple OS is uncertain. For those that remain, applications must be rewritten, or use costly emulation to run in a PowerPC-based system. Like all Sun processors, UltraSPARC-IIi is binary compatible with existing and emerging SPARC implementations and over 12,000 applications. Software need not be modified or recompiled to take full advantage of the features provided in Sun's UltraSPARC-IIi.

- **UltraSPARC-IIi provides the speed and multimedia capabilities required by new media systems — PowerPC supports a subset of these features**
- **UltraSPARC-IIi provides cost-effective price/performance ratios — PowerPC has a high price for comparable performance**
- **UltraSPARC-IIi supports a host of desktop and server applications — the set of applications for PowerPC remains limited**
- **UltraSPARC-IIi is binary compatible — PowerPC requires 68000-based applications to be rewritten**
- **Sun offers the robust VIS instruction set — PowerPC has not implemented any multimedia instructions to date**

Figure 5-10 Advantages of UltraSPARC-IIi over the PowerPC 603e processor

Processor Comparison Summary

	UltraSPARC-IIi	HP PA-8000	Cyrix MediaGX	MIPS R10000	Intel Pentium II	PowerPC 603e
Architecture	SPARC V-9	HP-PA	X86	MIPS III	X86	PowerPC
Open vs. Proprietary	Open	Proprietary	Proprietary	Open	Proprietary	Open
64-bit Architecture						
High Volume Processor						
High Bandwidth						
Integration Level						
On-chip MMU						
On-chip I/O Interface						
On-chip Cache						
On-chip Multimedia Support						
Features						
Clock Speed	300 MHz	180 MHz	180 Mhz	195 MHz	233-300 Mhz	300 MHz
Binary Compatibility with Existing Applications						
Performance						
SPECint95/fp95	>12/>12	11.8/18.7	N/A	10.7/19.0	11.7/8.15	7.4/6.1
Target Environment	Low cost desktops and servers High-end embedded applications: networking, telecommunications, imaging	Workstations and Servers	Desktops	Workstations	Workstations and Servers	Low-power, Low-cost, Desktops & Portables

Summary 6

With UltraSPARC-IIi, Sun provides a high bandwidth, high volume processor for 64-bit computing performance at affordable prices, addressing both the bandwidth and cost needs of new media applications. Combined with the VIS Instruction Set, UltraSPARC-IIi provides compute power, uniprocessor support, 64-bit addressing, and power management, all in a low cost, highly integrated design.

A technology leader for over a decade, Sun Microsystems has continually met the challenges of evolving user and application needs. Sun is constantly looking to, and shaping, the future of computing by investing in new technologies like the UltraSPARC-IIi. They recognize that a consistent, continuous application of resources is needed to meet the needs of a rapidly-changing computing marketplace. Investments in high performance, low-cost processor technologies, standards development, and advanced software environments and tools ensures that Sun customers will always have access to the best products available.

References

[SI 1992] SPARC International, Inc. *The SPARC Architecture Manual - Version 8*, Prentice-Hall, Englewood Cliffs, New Jersey, 1992.

[Patterson 1989] D. A. Patterson, J. L. Hennessy, *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann Publishers, Palo Alto, California, 1989.

[Smith 1982] J. E. Smith and A. R. Pleszkun, *Implementing Precise Interrupts in Pipelined Processors*, IEEE Transactions of Computers, May 1982, pages 562-573.

[Sweazey 1986] P. Sweazey, A. J. Smith, *A Class of Compatible Cache-Consistency Protocols and Their Support by the IEEE Future Bus*, Proc. 13th International Symposium on Computer Architecture, Tokyo, Japan, 1986.

Ben J. Catanzaro, *The SPARC Technical Papers*, Springer-Verlag, 1991.

SPARC Strategy and Technology, Sun Microsystems, 1991.

UltraSPARC-IIi User's Manual, Sun Microsystems, 1997.

Sun Microsystems, Incorporated
901 San Antonio Road
Palo Alto, CA 94303 USA
650 960-1300
FAX 650 969-9131
http://www.sun.com

Sales Offices

Argentina: +54-1-317-5600
Australia: +61-2-9844-5000
Austria: +43-1-60563-0
Belgium: +32-2-716-7911
Brazil: +55-11-5181-8988
Canada: +905-477-6745
Chile: +56-2-638-6364
Colombia: +571-622-1717
Commonwealth of Independent States:
+7-502-935-8411
Czech/Slovak Republics:
+42-2-205-102-33
Denmark: +45-44-89-49-89
Estonia: +372-6-308-900
Finland: +358-9-525-561
France: +33-01-30-67-50-00
Germany: +49-89-46008-0
Greece: +30-1-680-6676
Hong Kong: +852-2802-4188
Hungary: +36-1-202-4415
Iceland: +354-563-3010
India: +91-80-559-9595
Ireland: +353-1-8055-666
Israel: +972-9-956-9250
Italy: +39-39-60551
Japan: +81-3-5717-5000
Korea: +822-3469-0114
Latin America/Caribbean:
+1-650-688-9464
Latvia: +371-755-11-33
Lithuania: +370-729-8468
Luxembourg: +352-491-1331
Malaysia: +603-264-9988
Mexico: +52-5-258-6100
Netherlands: +31-33-450-1234
New Zealand: +64-4-499-2344
Norway: +47-2218-5800
People's Republic of China:
Beijing: +86-10-6849-2828
Chengdu: +86-28-678-0121
Guangzhou: +86-20-8777-9913
Shanghai: +86-21-6247-4068
Poland: +48-22-658-4535
Portugal: +351-1-412-7710
Russia: +7-502-935-8411
Singapore: +65-438-1888
South Africa: +2711-805-4305
Spain: +34-1-596-9900
Sweden: +46-8-623-90-00
Switzerland: +41-1-825-7111
Taiwan: +886-2-514-0567
Thailand: +662-636-1555
Turkey: +90-212-236 3300
United Arab Emirates:
+971-4-366-333
United Kingdom: +44-1-276-20444
United States: +1-800-821-4643
Venezuela: +58-2-286-1044
Worldwide Headquarters:
+1-650-960-1300