Introduction to the LR4000 MIPS Microprocessor

Publication ID: **J14015**
Publication Date: **October 1, 1991**

Company: **L S I LOGIC CORP**

MIPS Computer Systems, Inc.
950 DeGuigne Drive
Sunnyvale, CA 94088

Contents

Figures

CPU Overview

1

1.1 Introduction

The MIPS R4000 family consists of high performance 32-bit and 64-bit RISC microprocessors that deliver excellent processing solutions over a wide variety of applications and prices. Systems applications for this family range from inexpensive, highly-integrated desktop systems through large multiprocessor servers whose CPU performance rivals current mainframes, and whose address space requirements are not met by current generation microprocessors. Embedded processing applications for the R4000 family of microprocessors range from high-performance laser printer controllers to high-precision avionics controllers.

The R4000 provides complete application-software compatibility with the MIPS R2000, R3000, and R6000 processors. RISC/os, RISCompilers, and the thousands of application programs which run on the MIPS architecture augment this powerful family of processors and provide a complete solution to a large number of processing needs. In addition, an array of development tools offers complete support for a wide variety of R4000-based applications.

High integer performance, as well as floating-point performance, has been achieved through a number of techniques such as superpipelining, on-chip caches, a pipelined floating-point unit, two-level cache memory, and a high-performance on-chip TLB. The R4000's cache and Memory Management Unit (MMU) offer high performance in handling both large-address-space tasks and a large number of users. These features allow the design of balanced systems, suitable not only for technical and graphics applications, but also for commercial applications like transaction processing with fault

tolerant-support. Over a wide range of realistic benchmarks, the R4000 family of microprocessors delivers from 32 to 45 VAX MIPS[1] of sustained performance at 50 MHz.

The R4000 provides a compatible, timely, and necessary path from 32-bit to true 64-bit computing for users and software developers. The path to the on-chip caches is 64-bits wide. The on-chip floating-point coprocessor is a 64-bit FPU. The R4000 provides 64-bit integers, registers, and a flat 64-bit virtual address space. Compatibility with existing 32-bit application code is maintained; however, and an efficient mix of 32-bit and 64-bit programs can run on the same R4000-based machine.

The 64-bit addressing capability of the R4000 allows designers to build operating systems with extensive file mapping, allowing direct access to files without explicit I/O calls. This addressing capability paves the way for the next generation video technology and documentation with photographs and high-quality images. In addition, CAD applications with huge databases of complex structured objects, geographic information systems, and technical number crunching applications with large data sets will benefit greatly from this capability.

Figure 1-1 shows the block diagram of the R4000 CPU.

1. Geometric mean of 15 large applications compared to a VAX 11/780.

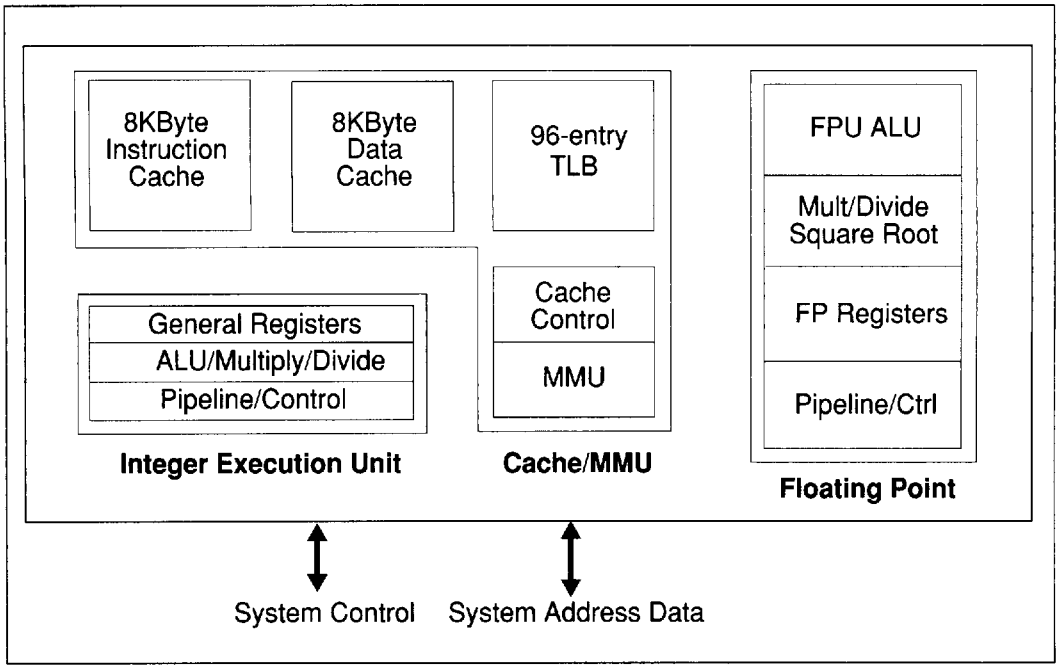

Figure 1-1 R4000 Block Diagram

The R4000 is designed for a long lifetime. Just like the original 8 MHz R2000, which has been scaled into the 40 MHz R3000A with very little redesign, the R4000 has been designed to scale easily with process technology improvements. This scalability provides R4000 customers assurance of staying on the leading edge of technology for many years to come. The R4000 executes instructions more than twice as fast as the R3000, demonstrating the scalability of the MIPS architecture. Figure 1-2 shows the performance trend of the MIPS microprocessors with respect to the computer industry.

There are three R4000 processors, packaged in three different configurations. All processors are implemented in sub-micron CMOS technology:

- The R4000SC is packaged in a 447-pin LGA and PGA and includes integrated control for large secondary caches built from standard SRAMs. It is designed for use in high-performance uniprocessor systems.

- The R4000MC is also packaged in the same 447-pin LGA and PGA includes, in addition, support for a wide variety of bus designs and cache-coherency mechanisms. The R4000MC is designed for use in large cache-coherent multiprocessor systems.
- The R4000PC is packaged in a 179-pin PGA designed for cost–sensitive systems with no secondary caches, such as inexpensive desktop systems and high-end embedded controllers.

The R4000 processors are available from MIPS' semiconductor partners: Integrated Device Technology, LSI Logic Corporation, NEC, Performance Semiconductor, and Siemens.

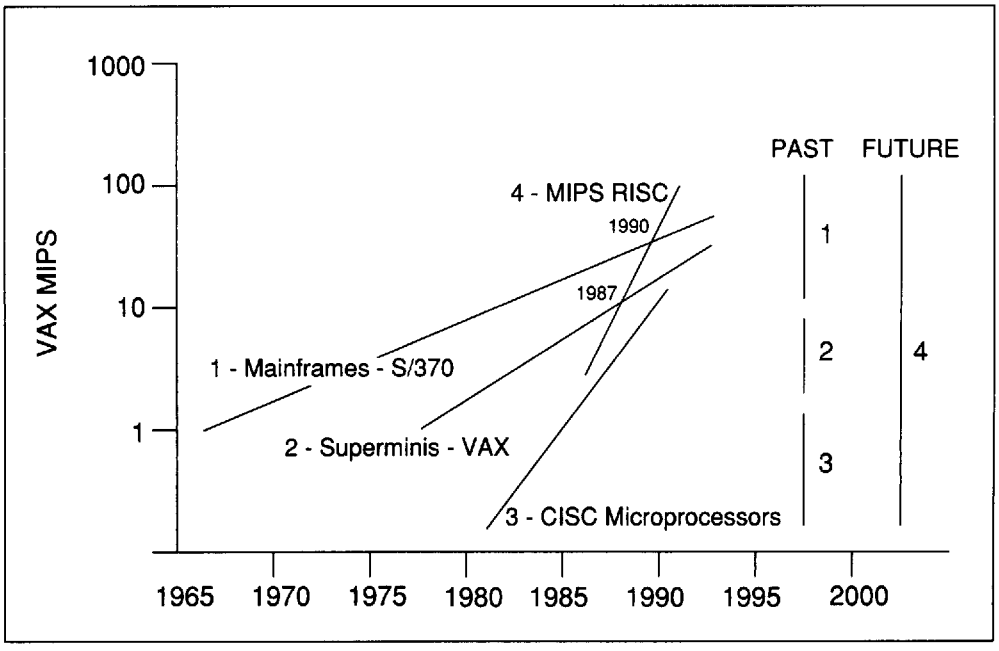

Figure 1-2 Performance Trend of MIPS RISC

1.2 Key Features

The R4000 family is designed for use in large multiprocessor systems as well as in smaller, cost sensitive uniprocessor systems. Some of the key features are:

- True 64-bit microprocessor with 64-bit integer and floating-point operations, registers, and virtual addresses.
- Fully compatible with earlier 32-bit MIPS microprocessors.
- Dual instruction issue with no restrictions on the type of instructions issued.
- 50 MHz clock frequency, 5-volt power supply.
- On-chip 8 KByte Instruction Cache, 8 KByte Data Cache, and an optional 128-bit secondary cache interface.
- On-chip Memory Management Unit (MMU) containing a fully associative TLB whose entries have a variable page size ranging from 4 KB to 16 MByte.
- On-chip ANSI/IEEE-754 standard floating-point unit with precise exceptions.
- Thirty-two doubleword (64-bit) general- purpose registers and sixteen doubleword (64-bit) floating-point registers.
- 36-bit physical address accessing 64 GBytes of physical memory.
- 6 hardware interrupts.
- 64-bit cache-coherent system interface with flexible and high-performance multiprocessing support.
- Fault tolerant support.
- Dynamically configurable big-endian or little-endian byte ordering.
- Timing flexibility for 128-bit secondary cache interface and 64-bit system interface to allow speed matching of logic and memory components.

1.3 Applications Environments

The R4000 family of microprocessors is well suited for low-cost systems, mid-range systems, and high-end systems. Low-cost systems range from high-performance embedded controllers to desktop systems. Mid-range systems include high-performance workstations and departmental servers. Multiprocessing systems and fault-tolerant systems comprise the high-end systems.

1.3.1 Low-Cost Systems

The on-chip caches, Memory Management Unit, and Floating-Point Unit on the R4000PC make it an excellent choice for an entry-level desktop system. The R4000PC offers a one-chip processor solution for low-cost desktop systems and high-performance embedded controllers. The parts count in this system configuration is minimal.

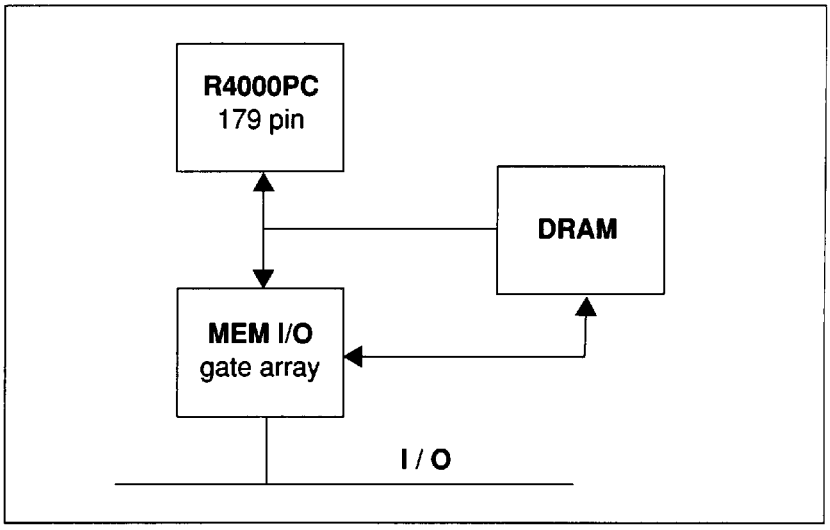

Figure 1-3 Low-Cost Configuration

1.3.2 Mid-Range Systems

The R4000SC incorporates many features that deliver high performance in a multi-user computer. The on-chip MMU, which maps over 1.5 Gbytes of virtual memory, and an interface to large secondary caches are critical to high performance and to supporting multiple users. The secondary cache interface and system interface are programmable so that the system designer can use standard SRAMs to suit the needs of individual systems.

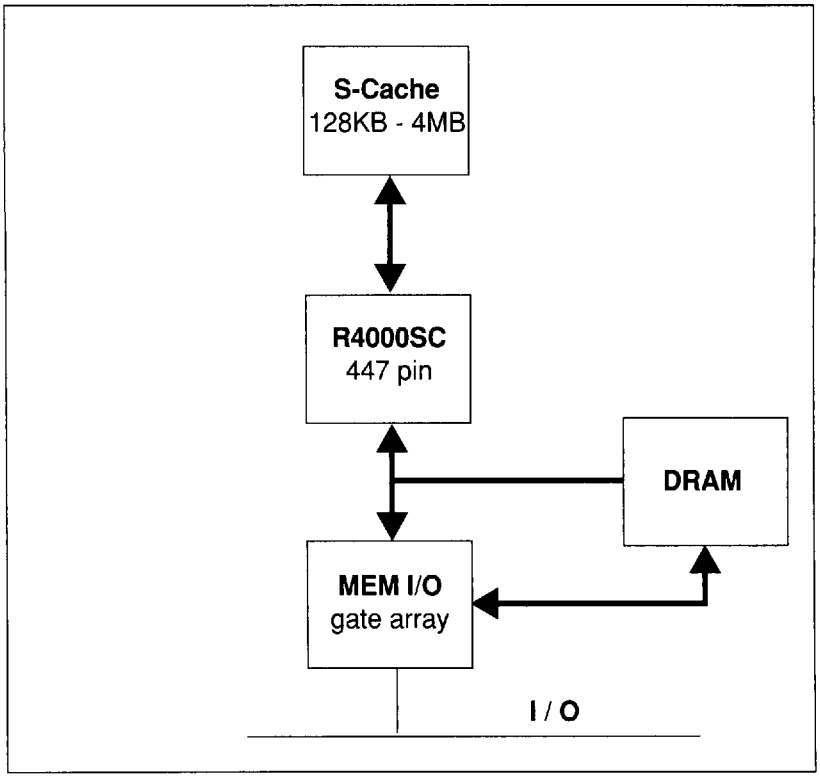

Figure 1-4 Mid-Range Configuration

1.3.3 High-End Systems

The R4000MC has, in addition to the features on the R4000SC, sophisticated support for multiprocessing systems. Using the R4000MC, a tightly-coupled multiprocessor can be built based on either snooping or directory-based cache coherence schemes. The system designer can choose the type of cache coherence scheme depending upon the application and the level of data sharing.

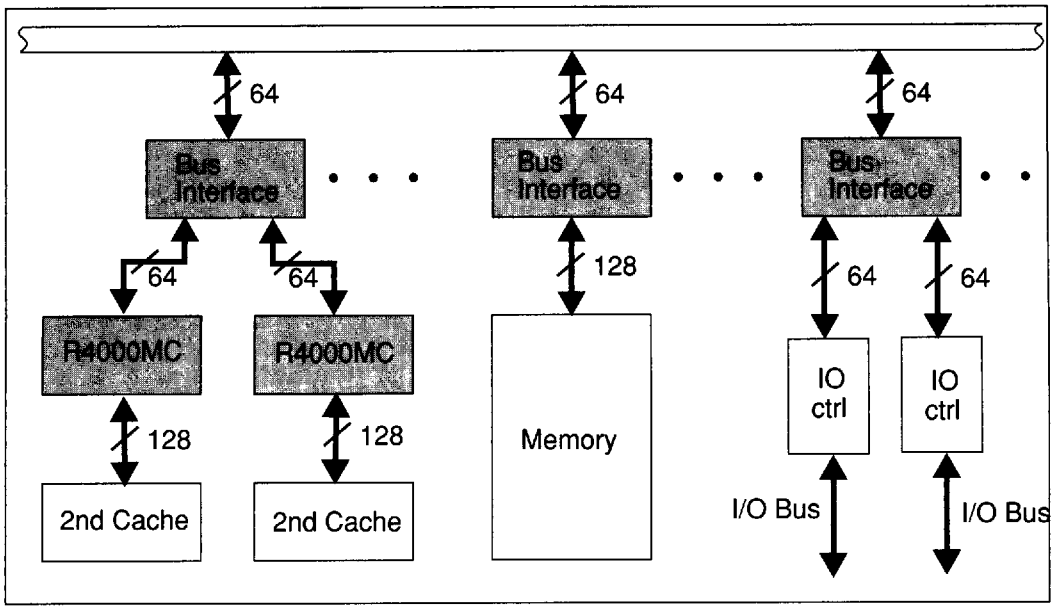

Figure 1-5 High-End Configuration

1.4 Superpipelined Implementation

In order to achieve the high performance required in a third generation RISC design, the R4000 exploits instruction-level parallelism using a superpipelined micro-architecture. The R4000 implements an 8-stage superpipeline which places no restrictions on instruction issue. Any two instructions can be issued each cycle under normal circumstances. Since the superpipeline places no restrictions on the order of instruction issue, the full benefit of the R4000 can be realized by existing application programs without any need for recompilation.

The internal pipeline of the R4000 operates at a frequency of 100 MHz, which is twice the external clock frequency. The 8-stage superpipeline of the R4000 is made possible by pipelining cache accesses, shortening register access times, implementing virtually indexed primary caches, and allowing the latency of functional units to span multiple pipeline cycles.

After extensive simulation of methods of exploiting instruction-level parallelism, superpipelining was chosen since it improves integer performance commensurate with floating-point performance. This method retains the balance of a mainframe CPU without skewing the performance heavily in favor of technical applications. Given today's

technology, when compared with the other methods of exploiting instruction-level parallelism, this approach results in less complex logic, faster cycle times, quicker design cycles, and lower cost.

1.5 Memory System

To achieve its high performance in uniprocessor and multiprocessor systems, the R4000 has support for a cache memory hierarchy that increases memory access bandwidth and reduces the latency of load and store instructions. For fast cache access, the R4000 incorporates on-chip instruction and data caches. These high-speed caches keep the pipeline full by feeding the processor at a peak rate of two instructions and two data words every pipeline cycle (10 ns).

The R4000SC and R4000MC interface to an optional secondary cache. The cache memory hierarchy is designed to minimize miss penalty and miss rates on memory accesses, thereby improving overall system performance. The size of the secondary cache can range from 128 KBytes to 4 MBytes. To minimize cache miss latency, the interface to the secondary cache is 128 bits wide.

The line size and access time of the secondary cache are configurable. This provides flexibility to design caches that meet the individual requirements of virtually any system and allows the design of a secondary cache using standard low-cost SRAMs.

1.6 Virtual Addressing

The R4000 contains a flexible, on-chip memory management unit that incorporates a fully-associative TLB, which performs the virtual to physical address translation. Each entry in the TLB can map from 4 KBytes to 16 MBytes of virtual memory for a total mapping of up to 1.5 GBytes. As a result, it is possible to map very large contiguous regions of virtual memory using a single translation. This feature is particularly useful in mapping large physical regions such as frame buffers, large databases, and in scientific and engineering number-crunching applications where large data arrays are manipulated.

In the R4000 TLB, critically needed address translations can be programmed to remain locked in the TLB. This feature is especially useful in realtime systems where predictability of system response is a critical requirement.

1.7 Instruction Set

The R4000 implements the extended MIPS Instruction Set Architecture which improves performance and adds functional capabilities while maintaining complete applications binary compatibility with earlier MIPS microprocessors. The extensions result in better code density, greater multiprocessing support, improved performance for commonly used code sequences in operating system kernels, and faster execution of floating-point intensive applications. All resource dependencies are transparent to the programmer in this instruction set. Every instruction is 32 bits wide to allow easy and fast decode.

New instructions have been defined to take advantage of the 64-bit architecture. When operating as a 32-bit microprocessor, the R4000 takes an exception on these new instructions.

1.8 Debugging Support

The R4000 provides processing resources and instructions that significantly reduce program and hardware debugging time. The coprocessor Watch Register allows the setting of data break points, which cause traps when either a load or store instruction is executed to a specified physical address. Also, instruction breakpoints can be set by using the Breakpoint instruction. Development of Cache and TLB diagnostic software is facilitated by R4000 instructions that allow the reading and writing of any location in the cache and TLB.

1.9 Software Development Tools

For software development, MIPS offers a number of programs, utilities, and tools designed for the system programmer which aid in the development of operating systems and stand-alone software for machines based on the MIPS R4000. Among these software tools are a cache simulator, which allows the user to model the cache and memory hierarchy; an instruction set simulator, which allows the user to debug stand-alone programs; and tools and utilities that allow the user to program PROMS for a test system, download stand-alone programs from a development system to the test system, and initiate program execution and debugging.

1.10 Multiple Sources for Pin-compatible Parts

Customers from around the world are assured availability of pin-compatible MIPS microprocessors from five semiconductor manufacturers. Multiple sources of interchangeable components help lower the cost of parts and assure steady supplies. This reflects a distinctly different trend in the computer industry of the 1990s. Customers have the choice of buying parts from any of the following MIPS semiconductor partners:

- Integrated Device Technology
- LSI Logic Corporation
- NEC Corporation
- Performance Semiconductor
- Siemens AG

Architecture

2

The MIPS Instruction Set Architecture is simple and based on general-purpose registers. This chapter describes the attributes of a MIPS processor as seen by the programmer, such as the data formats and addressing modes, register set organization, instruction set, and coprocessor organization.

The R4000 can be programmed to operate either as a 32-bit or a 64-bit microprocessor. When set to operate as a 32-bit microprocessor, the R4000 has thirty-two 32-bit general-purpose registers, generates 32-bit virtual addresses, and performs 32-bit operations on the contents of the general registers; the floating-point coprocessor has sixteen doubleword (64-bit) registers.

When operating as a 64-bit microprocessor, the R4000 has thirty-two 64-bit general registers, generates 64-bit virtual addresses, and performs 64-bit operations on the contents of the general-purpose registers; the floating-point coprocessor has thirty-two doubleword (64-bit) registers. New instructions have been defined to manipulate 64-bit quantities. However, it should be pointed out that all instructions in the MIPS Instruction Set Architecture, including the new instructions, are 32 bits wide.

2.1 Data Formats and Addressing

The R4000 implements integer and floating-point operations on the same die. Integer operations are performed on 32-bit or 64-bit unsigned and signed (two's complement) data. Floating-point operations are performed on data in the single- and double-precision floating-point formats. Loads and stores are performed on 8-bit, 16-bit, 32-bit, and 64-bit operands.

The MIPS Instruction Set Architecture defines an 8-bit byte, a 16-bit halfword, a 32-bit word, and a 64-bit doubleword. The byte ordering is user configurable into either big-endian or little-endian byte ordering. When configured as a big-endian system, byte 0 is always the most-significant (leftmost) byte to provide compatibility with MC 68000 and IBM 370 conventions. When configured as a little-endian system, byte 0 is always the least-significant (rightmost) byte, providing compatibility with iAPX x86, NS 32000, and DEC VAX conventions.

Within this manual, bit 0 is always the least-significant (rightmost) bit. Bit designations are always little-endian. Figures 2-1 and 2-2 show the ordering of bytes, within words, for each of the two conventions.

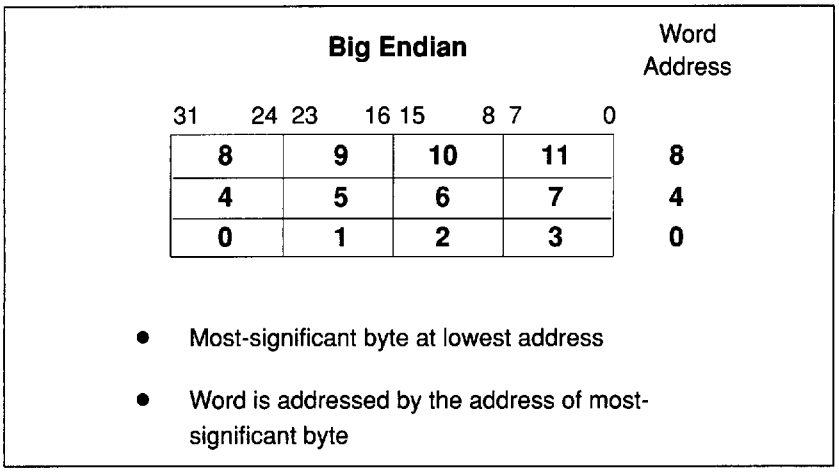

Figure 2-1 Big-endian Byte Ordering

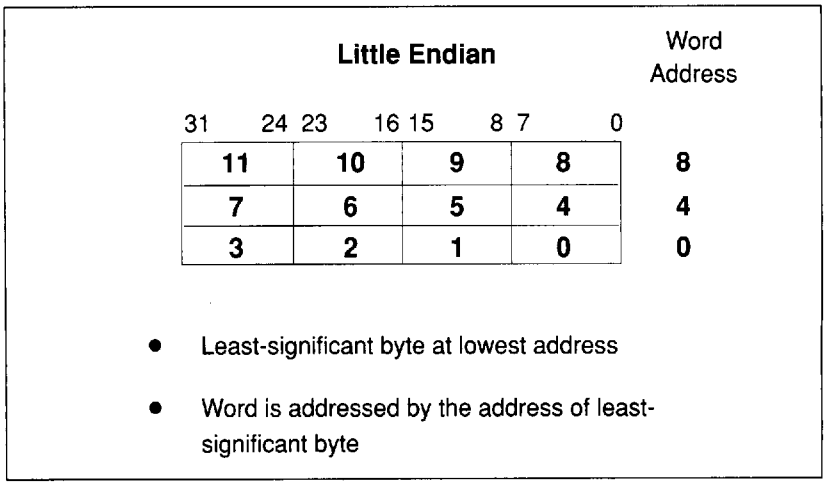

Figure 2-2 Little-endian Byte Ordering

2.2 Processing Resources

The architecture provides registers and status information that are used for computation, addressing, control, and the handling of interruptions. Some of these resources are described in this section.

2.2.1 General Registers

There are 32 general registers in the integer unit, each consisting of a single word (32 bits) or doubleword (64 bits) depending on operating mode. The 32 general registers are all equivalent with two exceptions: r0 is hardwired to a zero value, and r31 is the link register for the JUMP AND LINK instructions. r0 maintains a value of zero when used as a source register under all conditions. When used as a target, the result is discarded.

Registers are assigned uses and special names by software convention in the assembler and compiler for ease of programming and efficiency of execution. The MIPS *Assembly Language Programmer's Guide* contains detailed information on the register usage conventions.

2.2.2 Special Registers

In addition to the general registers, the integer unit has three special registers. The use of these special registers is implicit in certain instructions. The special registers are:

PC Program Counter

HI Multiply/Divide register higher word

LO Multiply/Divide register lower word

The two Multiply/Divide registers (HI, LO) store the 64-bit (32-bit mode) or 128-bit (64-bit mode) result of integer multiply operations and the quotient and remainder of integer divide operations.

2.2.3 Coprocessor Registers

Each of four coprocessor units may define up to 32 general registers and 32 control registers. Registers associated with Coprocessor 0, the system control coprocessor, are used to view and manipulate the state. associated with the exception handling, memory management, and diagnostics. Coprocessor 1 is the floating-point coprocessor, which has 32 general registers and two control registers.

NOTE: On the R4000, coprocessor number 2 is reserved for future definition; the encoding for coprocessor 3 is used to define certain 64-bit architecture extensions.

2.3 Instruction Set

The MIPS Instruction Set Architecture has a set of simple instructions, all of which are 32 bits wide. The instructions are divided into a few general classes described in the following sections.

2.3.1 Load and Store Instructions

Load and Store instructions move data between memory and general registers. Since fast and efficient load and store instructions are at the very heart of a RISC processor, the specification of these instructions is simple and regular to facilitate fast decoding. Only one addressing mode is specified: base register plus a signed 16-bit immediate offset.

Bytes, halfwords, and words may be loaded to and stored from general registers by specifying aligned addresses. Unaligned memory references are efficiently supported by two pairs of load and store instructions.

The R4000 implementation allows the instruction immediately following a load to use the contents of the register loaded. In such cases, the hardware interlocks until the load is completed. This mechanism reduces code size in the cases where the compiler is unable to schedule a useful instruction between a load and the use of the value loaded. It is still desirable to separate a load instruction from the instruction that uses the register loaded, but not required for correct functionality.

2.3.2 Computational Instructions

Computational instructions perform arithmetic, logical, and shift operations on values in registers. There are two ways in which operands are specified:

1. The two source operands and the destination operand are registers.
2. One of the source operands and the destination operand are registers. The other source operand is a 16-bit immediate value.

2.3.3 Jump and Branch Instructions

All Jump and Branch instructions have an architectural delay of exactly one instruction to reduce the pipeline branch penalty. However, the R4000 implementation of the architecture has a three-cycle branch delay. RISCompilers manage the instruction scheduling such that the delay slot is occupied by a real instruction (instead of a nop) in most instances of Jump and Branch instructions.

Jump instructions provide an unconditional transfer of flow during program execution. Branch instructions provide a mechanism to branch based on the outcome of a specified test. An extensive set of such tests allows the efficient compilation of high-level languages. In addition, there is a variant of the branch instructions in which the instruction in the delay slot is nullified if the branch is not taken. RISCompilers take advantage of this feature for compact encoding of high-level language loop constructs and to reduce code length by copying the target of a branch instruction in the delay slot when there is no inline instruction which can be moved there.

There are variants of both the jump and branch instructions which save the return pointer either implicitly in GR31 or in any general register specified by the instruction.

2.3.4 Exception Instructions

Exception instructions cause a trap to be executed. The trap causes a branch to the general exception handling vector. There are two kinds of exception instructions: those that unconditionally trap to the general exception-handling vector, and those which do so conditioned upon the result of a comparison either of the contents of two general registers or one general register and an immediate value.

2.3.5 Coprocessor Instructions

Coprocessor instructions perform operations in special-purpose hardware blocks called coprocessors. These hardware blocks execute instructions in conjunction with the CPU.

- Coprocessor 0 instructions manipulate the memory management and exception handling facilities of the processor.
- Coprocessor 1 implements floating-point operations.

Coprocessor general registers may be directly loaded from memory and stored into memory, and may be transferred between the coprocessor and processor. The coprocessor control registers may be transferred only directly between the coprocessor and processor. Coprocessor instructions may alter either the coprocessor general registers or the coprocessor control registers. There are also instructions that branch based on a condition in one of the coprocessors.

2.3.6 Synchronization Instructions

There are two instructions that support synchronization between processes executing on the same processor or on different processors. Instead of defining one synchronization primitive (such as test and set), an instruction pair is defined: Load Linked and Store Conditional, which can be used to construct any of the common synchronization primitives such as test and set, compare and swap, load and clear, and load and add.

The Load Linked instruction, in addition to doing a simple load, sets a state bit called the *link* bit. The link bit forms a breakable link between the load linked instruction and a subsequent Store Conditional instruction.

The Store Conditional instructions performs a simple store if, and only if, the link bit is set when the store is executed. If the link bit is not set, then the store will fail to execute. The success or failure of the Store Conditional instructions is indicated in the target register of the store. The link bit is reset if any of the following events take place:

1. If any other processor or device has modified the specified physical address or if an ERET instruction (Exception Return) occurs between the Load Linked and Store Conditional instructions, or
2. If a cache miss occurs between the Load Linked and Store Conditional instructions on the processor executing the synchronization instructions.

2.4 Floating-Point Coprocessor

The IEEE 754 standard compliant floating-point coprocessor implements the basic operations of single- and double-precision add, subtract, multiply, divide, and conversions as in the R3010 (floating-point coprocessor for the R3000 CPU), as well as single- and double-precision square root, explicitly rounded conversions, and load and store doubleword instructions.

Relative to the R3010, the latency (in nanoseconds) of floating-point multiply and divide has been improved by about 25 percent through the use of additional hardware, better process technology, and improved circuit design. In addition to latency improvement, the adder and multiplier have been pipelined to achieve higher repeat rates and greater throughput on vectorizable code.

Because precise exceptions are essential in mission-critical environments and highly desirable for debugging in any environment, the R4000 implements fully precise floating-point exceptions while allowing overlapped and pipelined operation.

The floating-point coprocessor, when in 32-bit mode, has sixteen 64-bit general-purpose registers which can also be accessed as thirty-two 32-bit registers. In 64-bit mode, the floating-point coprocessor has thirty two 64-bit registers. The architecture supports coprocessor load and store double instructions so that when configured as 64-bit wide registers, the floating-point unit can take advantage of the 64-bit wide path to the data cache and load or store a doubleword every cycle. Doubleword loads and stores to the floating-point coprocessor are available in both 32-bit and 64-bit modes.

Two control registers are defined. These registers, shown in Figure 2-3, are primarily involved with diagnostic software, exception handling, state saving and restoring, and control of rounding modes.

FCR Number	Usage
0	Coprocessor Implementation and revision register
1-30	Reserved
31	Rounding mode, cause, trap enables, and flags

Figure 2-3 Floating-Point Control Registers

2.5 Support for Multiprocessing

The R4000MC implements sophisticated caches capable of supporting a number of cache coherence schemes, and synchronization primitives for the efficient implementation of tightly-coupled multiprocessor systems. The R4000PC and R4000SC are designed to operate only in uniprocessor systems.

In the R4000MC, cache coherence is maintained by hardware. The system control coprocessor permits the specification of different caching protocols on a per-page basis. A page may be:

- Uncached.
- Cached but non-coherent.
- Cached and coherent exclusive (only one copy of the data in any one cache on loads and stores).
- Cached and coherent exclusive on writes (write invalidate scheme–only one copy of the data in any one cache when that datum is written to).
- Cached and coherent with updates on writes (write-update scheme).

Depending upon the amount and type of data sharing in an application, the operating system can choose the most appropriate caching strategy.

Support for processor synchronization is provided by the Load Linked and Store Conditional instructions. The Load Linked and Store Conditional instructions:

1. Provide a simple mechanism for generating all of the common synchronization primitives including test-and-set, bit-level locks, semaphores, counters, sequencers, etc. with no additional hardware overhead.
2. Operate in such a fashion that bus traffic is only generated when the state of the cache line changes.
3. Need not lock a system bus—a very important feature for larger systems.

Systems Software Model

3

The R4000 has many features that provide a powerful environment for the system programmer. These features include:

- A system control coprocessor
- Configurable memory hierarchy
- Exceptions and exception handling
- Input/output organization

This chapter briefly outlines the different modes of:

- Virtual addressing available
- TLB structure
- Virtual address translation mechanism
- Cache management
- Exceptions and exception handling
- Input/output in the R4000 family of CPUs

3.1 System Control Coprocessor

The system control coprocessor (CP0) translates virtual addresses into physical addresses, manages exceptions, and handles the transitions between kernel and user states. CP0 also controls the cache subsystem and provides diagnostic control and error recovery facilities. A generic system timer is provided for interval timing, time keeping, process accounting, and time slicing.

Registers within the system control coprocessor provide access to the virtual memory system's page map. Other registers provide mechanisms to control the operating modes (virtual addressing mode, interrupts disabled or enabled, cache enabled or isolated), and to identify and handle exceptions.

The R4000 can be programmed to operate either as a 32-bit or 64-bit microprocessor in any of the three virtual addressing modes. For example, a valid combination could be that the kernel is executing in a mode that takes advantage of the 64-bit features while an application program running under control of the kernel uses only the 32-bit features. Similarly, with slight modifications, an operating system kernel designed for the 32-bit R3000 can run applications compiled for the R4000 as a 64-bit microprocessor.

3.1.1 Modes of Virtual Addressing

The R4000 provides three modes of virtual addressing:

- User mode
- Kernel mode
- Supervisor mode

These three modes are available to system software to provide a secure environment for user processes. Bits in a status register determine which virtual addressing mode is used and whether the 32-bit or 64-bit addressing model is used. In the user mode, the R4000 provides a single, uniform virtual address space of2 GBytes in 32-bit mode and 1 Terabytes in 64-bit mode. In 64-bit mode, the R4000 translates the lower 40 bits of the virtual address although all bits are interpreted as valid virtual address bits. If bits 61 through 40 of a 64-bit virtual address are not zero or mins one, an address error exception is taken by the R4000. Since all virtual address bits are significant, future versions of MIPS processors will be able to translate more bits of the 64-bit virtual address while maintaining compatibility with current applications.

When operating in the 32-bit kernel mode, four distinct virtual address spaces, totalling 4 GBytes, are simultaneously available and are differentiated by the high-order bits of the virtual address. In 64-bit mode, the kernel address space contains five virtual address spaces totalling 3 Terabytes. The virtual address space layout is a compatible extension of the 32-bit virtual address space layout.

The R4000 processors also support a supervisor mode in which the virtual address space, in 32-bit mode, is 2.5 Gbytes, divided into two regions based on the high-order bits of the virtual address. 64-bit supervisor mode includes two virtual address spaces totalling 2 Terabytes.

The three different modes of virtual addressing are shown in Figures 3-1, 3-2, and 3-3.

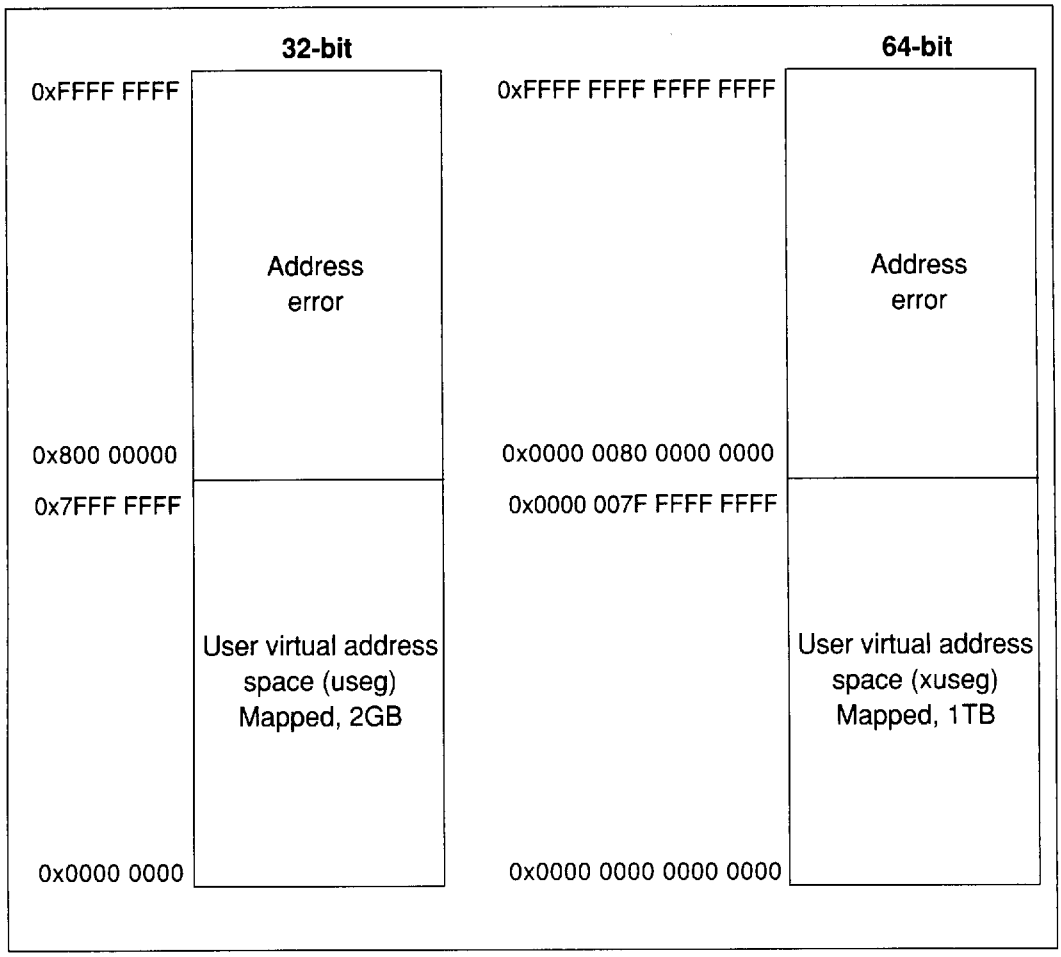

Figure 3-1 User Mode Virtual Addressing

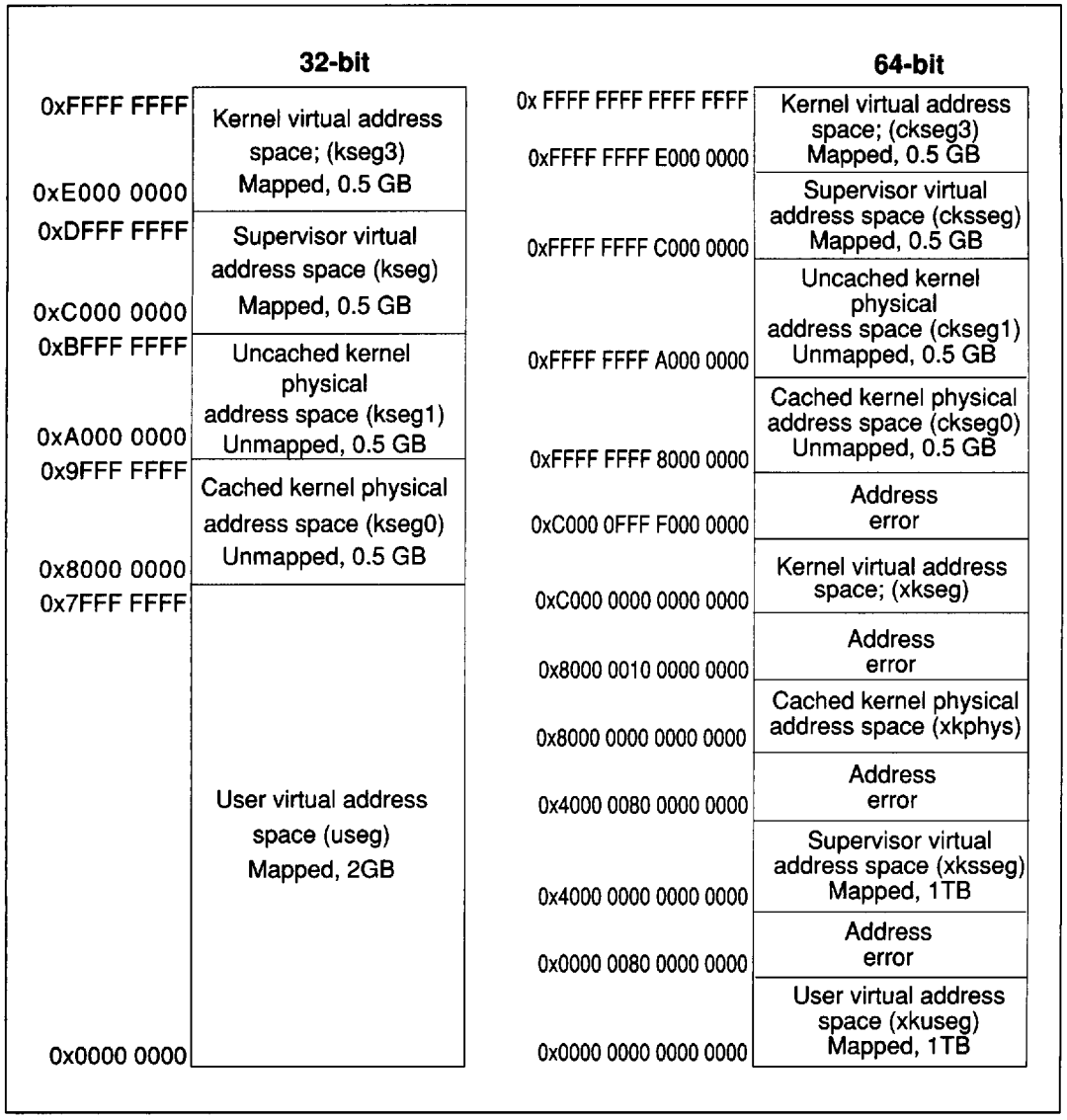

Figure 3-2 Kernel Mode Virtual Addressing

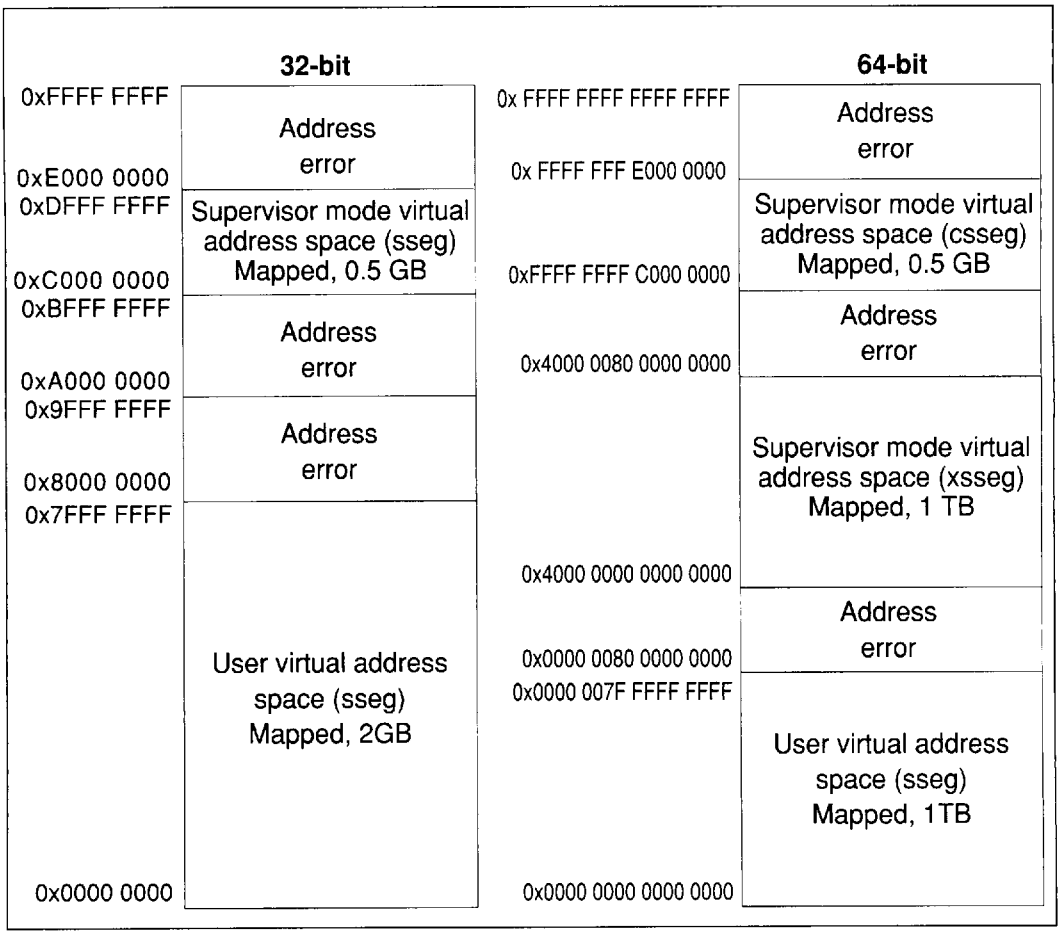

Figure 3-3 Supervisor Mode Virtual Addressing

3.1.2 TLB Structure

Mapped virtual addresses are translated into physical addresses using an on-chip, fully-associative translation lookaside buffer (TLB). The TLB on the R4000 contains translations for 48 even-odd pairs of pages, each of which map pages ranging from 4 KBytes to 16 MBytes in size. The page size is controlled on a per-pair basis by a page mask. The format of the TLB entry is shown in Figure 3-4.

If a TLB entry matches, the physical address and access control bits (C, D, and V) are retrieved; otherwise a TLB refill exception occurs. The TLB *miss handler* can, in most cases, refill the TLB in 9 instructions. If the access control bits (D and V) indicate that the access is not valid,

a TLB modification or TLB invalid exception occurs. If the C field indicates that the page is uncached, the physical address that is retrieved is used to access main memory, bypassing the cache.

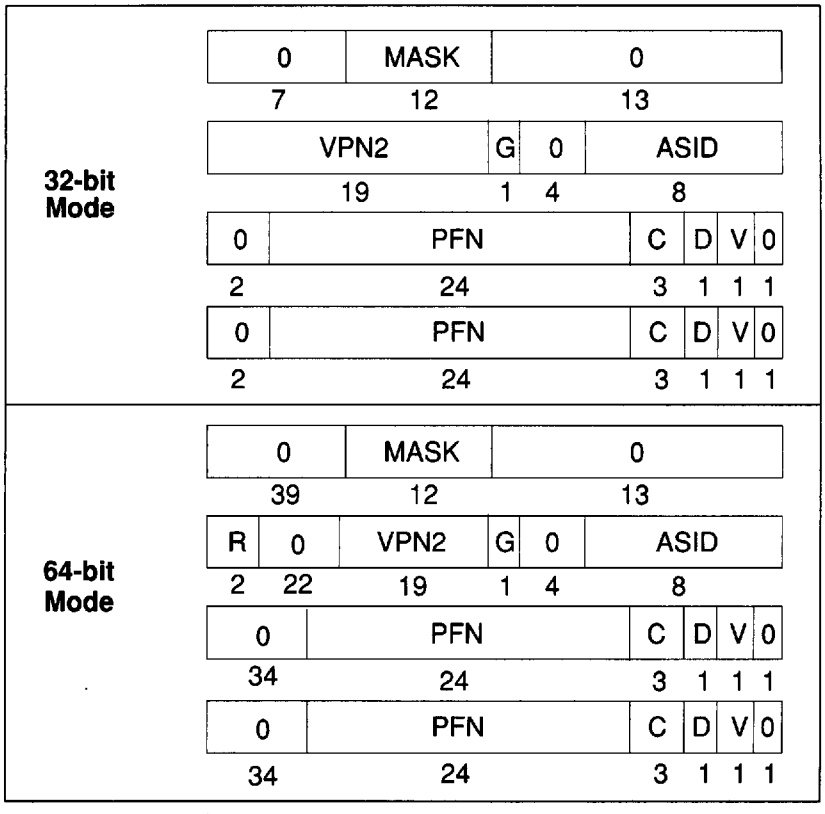

Figure 3-4 Format of a TLB Entry

where:

MASK	is the comparison mask.
VPN2	is the Virtual Page Number / 2.
ASID	is the Address Space Identifier.
PFN	is the Page Frame Number.
C	is the Cache algorithm for the page.
D	if set, page is Dirty.
V	if set, entry is Valid.
G	if set, page is Global (ignore ASID in match logic).
0	Reserved. Must be written as zeros; returns 0 on reads.
R	is the Region (00→user, 01→supervisor, 11→kernel) used to match Virtual address bits 63 and 62.

3.1.3 Virtual Address Translation

During the virtual address translation process, the R4000 compares the 8-bit Address Space Identifier (ASID) and the Virtual Page Number (VPN) of the virtual address with the contents of the TLB. Figure 3-5 illustrates the virtual address translation process.

A virtual address matches a TLB entry when the virtual page number (VPN) of the virtual address equals the VPN field of the entry, and either the Global (G) bit of the TLB entry is set, or the address space identifier (ASID) field of the virtual address matches the ASID field of the TLB entry. While the Valid (V) bit of the entry must be set for a valid translation to take place, it is not involved in the determination of a matching TLB entry.

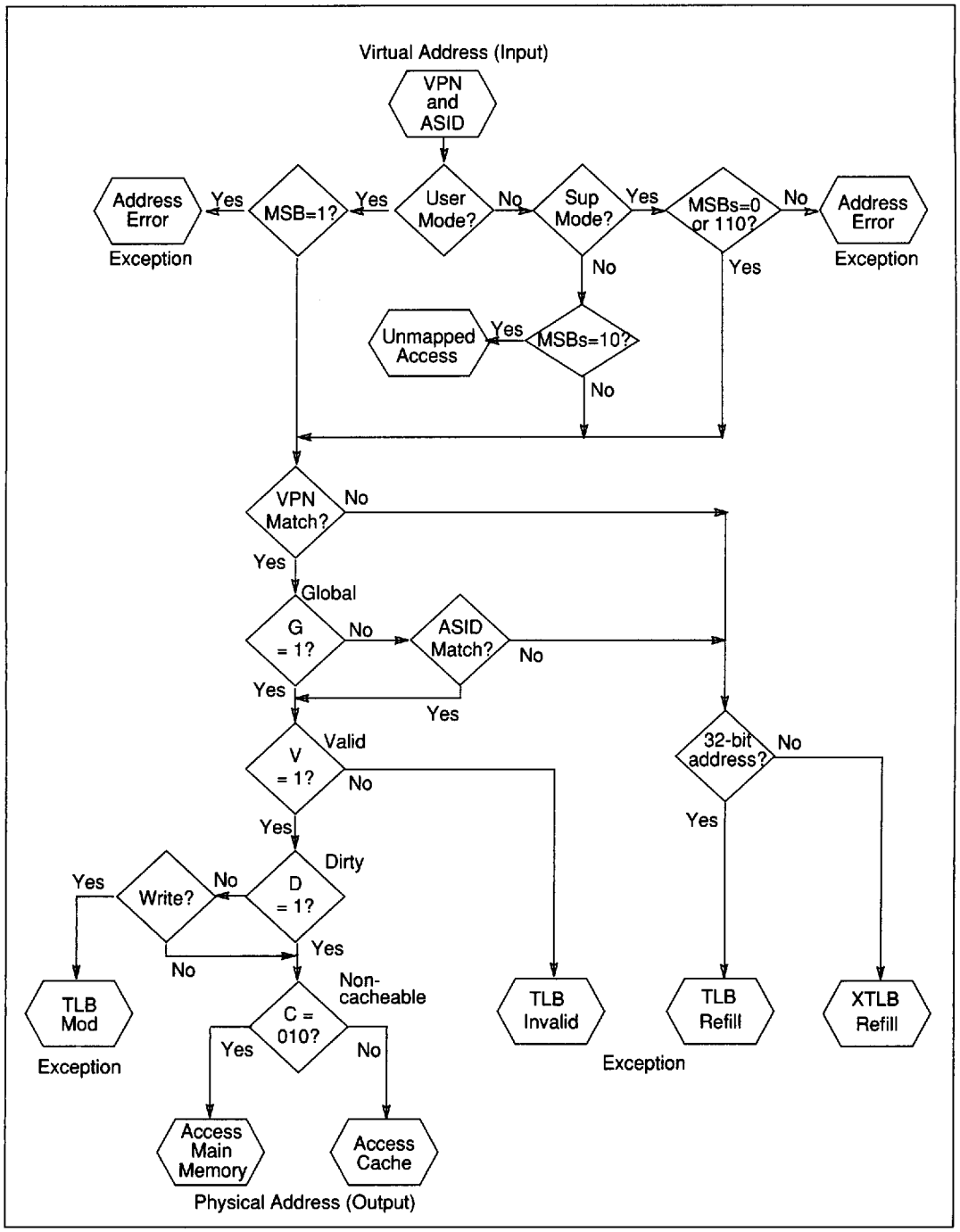

Figure 3-5 Virtual Address Translation

3.2 Cache Management

The R4000 cache management features allow the implementation of both a high-performance multiprocessing operating system and a simple uniprocessor operating system. A goal of the features is to maintain highly efficient multi-user performance and exploit the lower pipeline cycle time, which can be achieved using virtually indexed caches.

In order to be effectively matched with the R4000's superpipeline, the primary cache is virtually indexed and physically tagged. Using a part of the virtual index to address the cache enables the virtual address translation to happen in parallel with the primary cache lookup. The virtual address synonym problem that arises in the data cache is solved by flushing the shared data from the primary data cache before accessing a physical location by a different virtual address. The physically tagged cache implies that the cache need not be flushed on context switches.

The environment in which the R4000 cache management operations function has the following principal characteristics:

- The presence or absence of a secondary cache.
- A secondary cache that is either joint or split.
- Primary and secondary caches that are writeback.
- Primary caches that are virtually indexed and physically tagged.
- Secondary caches that are physically indexed and physically tagged.
- Externally initiated cache coherency operations.

Operating system environment is one where significant amounts of data copying and data initialization are performed. A number of cache management operations are provided to increase the efficiency of such operating systems activity. For details, please refer to the *R4000 User's Guide*.

3.3 Exceptions and Exception Handling

Exceptions are anomalies that occur during instruction execution that change the flow of execution and pass control to an exception handling routine. The exception handling software is responsible for efficiently handling these relatively infrequent events, such as translation misses, arithmetic overflow, I/O interrupts, and system

calls. Instructions that cause exception conditions are aborted, along with those that follow and have already begun executing, and a direct jump is executed to a designated exception handler routine.

The MIPS architecture defines a minimal amount of additional state that is saved in coprocessor registers upon an exception. This facilitates the speedy analysis of the cause of the exception, the servicing of the event that caused it, and the resumption of the original flow of execution, when applicable. To handle an exception, the processor executes, in kernel mode with interrupts disabled, an exception handler at a fixed address. To resume, the Program Counter, operating mode, and interrupt enable must be restored. This context is saved when an exception occurs.

3.3.1 Precise Exceptions

Exceptions are logically precise; the instruction that causes an exception and all those that follow it are aborted, generally before changing any state, and they are re-executed after the kernel has serviced the exception. When following instructions are killed, exceptions associated with those instructions are also killed. Therefore, exceptions are not taken in the order detected, but rather in instruction fetch order.

3.3.2 Exception Types

In descending order of priority, the exceptions recognized by the R4000 are as follows:

- Reset.
- Soft reset.
- Non-maskable interrupt.
- Address error - instruction fetch.
- TLB refill - instruction fetch.
- TLB invalid - instruction fetch.
- Cache error - instruction fetch.
- Virtual coherency - instruction fetch.
- Bus error - instruction fetch.
- Integer overflow, trap, system call, breakpoint, reserved instruction, coprocessor unusable, or floating-point exception.
- Address error - data access.

- TLB refill - data access.
- TLB invalid - data access.
- TLB modified - data write.
- Cache error - data access.
- Watch.
- Virtual coherency - data access.
- Bus error - data access.
- Interrupt.

3.4 Input/Output

In the MIPS architecture, Input/Output is memory mapped. Control over I/O modules is exercised by executing load and store instructions. In a system built using the R4000MC CPU, I/O modules may process data in memory using either cache-coherent or non-cache-coherent operations.

In an R4000PC or R4000SC system, however, I/O modules may only use non-cache-coherent operations to process data. In these systems, it is software's responsibility to make sure that the contents of the caches are updated in memory prior to an I/O output operation and are invalidated prior to an I/O input operation. The cache management primitives allow the systems programmer to efficiently carry out these operations.

Internal Block Overview

4

The R4000 RISC processor attains a high level of performance using superpipelining techniques, a high level of integration, and wide data paths. Implemented in sub-micron CMOS technology, this highly integrated chip contains instruction and data cache memory, a Floating-Point Unit, an integer unit, a Memory Management Unit, and control logic for a cache coherent system interface and an optional secondary cache interface. This chapter contains an overview of these functional unit. Figure 4-1 shows an internal block diagram of the R4000.

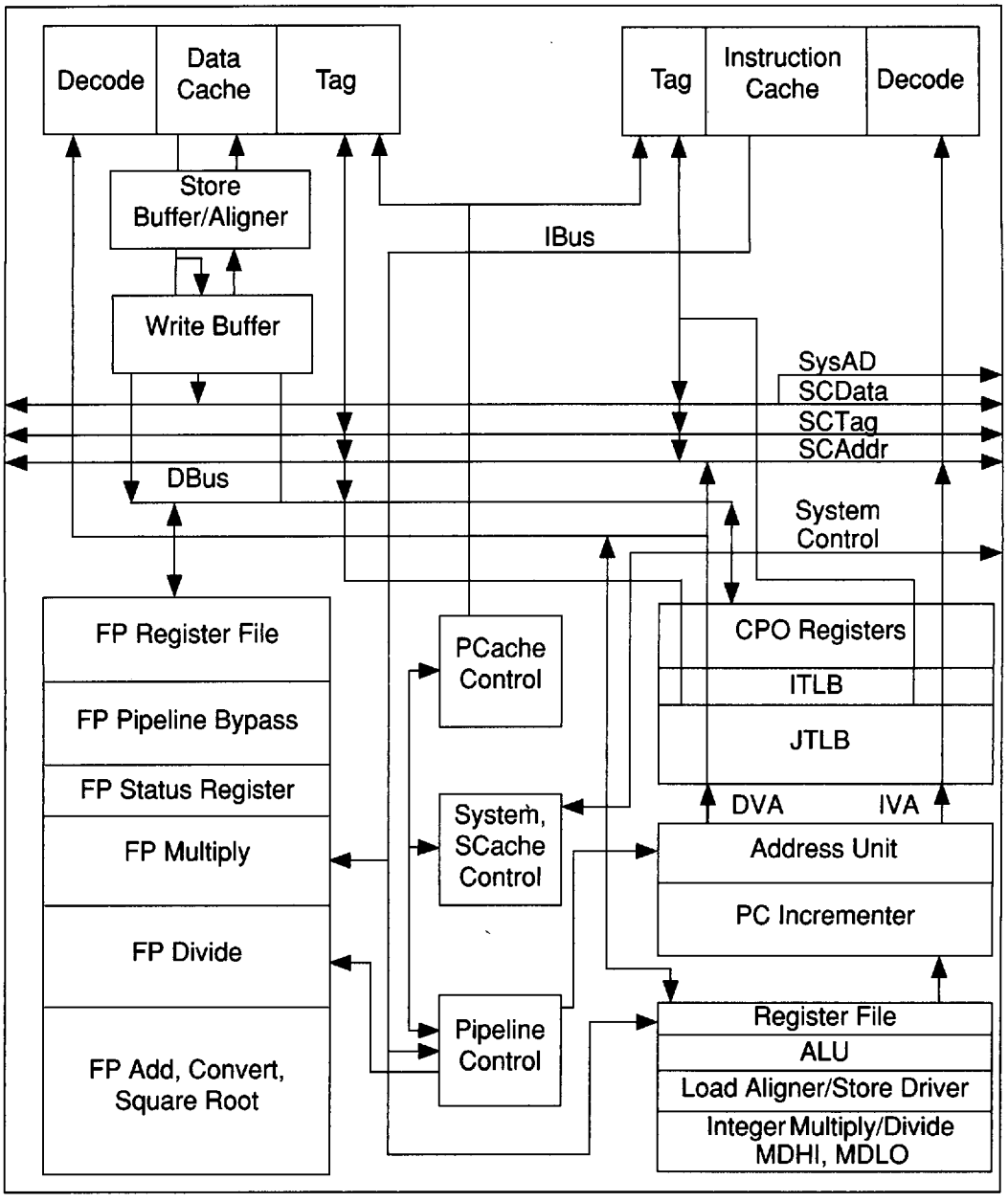

Figure 4-1 R4000 Internal Block Diagram

4.1 Cache Memory

In order to keep the R4000's superpipeline full and operating efficiently, the R4000 incorporates on-chip instruction and data caches. Each cache has its own 64-bit data path that can be accessed twice an external cycle, so the instruction and data caches can be accessed in parallel with full pipelining. Combining this feature with a pipelined access of each cache in a single external cycle, the cache subsystem provides the integer and floating-point units with an aggregate bandwidth of 1.6 GBytes per second at a master clock frequency of 50MHz.

4.1.1 Instruction Cache

The R4000 incorporates a direct-mapped on-chip instruction cache. This virtually indexed, physically tagged cache is 8 KBytes in size and is protected with byte parity. Because the cache is virtually indexed, the virtual-to-physical address translation occurs in parallel with the cache access, thus further increasing performance by allowing these two operations to occur simultaneously. The tag holds a 24-bit physical address and a valid bit, and is parity protected.

The instruction cache is 64-bits wide, and can be refilled or accessed twice per external cycle. Although the R4000 fetches one 64-bit unit per pipeline, only one 32-bit instruction is issued per pipeline cycle for a peak instruction bandwidth of 400 MBytes per second. The line size can be configured as four or eight words to allow different applications to have a line size that delivers optimum performance.

4.1.2 Data Cache

For fast, data access in a single external cycle, the R4000 includes an 8 KByte on-chip data cache. The data cache is protected with byte parity and its tag is protected with a single parity bit. It is virtually indexed and physically tagged to allow simultaneous address translation and data cache access.

The D-Cache is direct mapped, and its line size can be configured as four or eight words. The write policy is writeback, which means that a Store to a cache line does not immediately cause memory to be updated. This technique increases system performance by reducing bus traffic and eliminating the bottleneck of waiting for each Store operation to finish before issuing a subsequent memory operation.

Associated with the D-Cache is the store buffer. When the R4000 executes a Store instruction, this 2-entry buffer gets written with the store data while the tag comparison is performed. If the tag matches, then the data gets written into the D-Cache in the next pipeline cycle that the D-Cache is unaccessed. The store buffer allows the R4000 to execute two stores per master clock cycle and to perform back-to-back stores without penalty. Likewise, the R4000 can perform two loads or a load and store per master clock cycle without penalty, yielding 800 MBytes per second bandwidth without restrictions on instruction combinations.

When the D-Cache line does need to be written back to slower memory (either secondary cache or main memory), the processor writes the data to an internal write buffer which can hold a line (4 or 8 words) of data. By writing the data to this fast write buffer, the processor can continue executing instructions without having to wait until the write completes to the slower memory.

The R4000 caches are designed for easy and flexible integration in many types of multiprocessor systems. The D-Cache contains all the necessary state bits to allow the R4000 to maintain cache coherency between many R4000 processors in a system.

4.2 Memory Management Unit (Coprocessor 0)

The R4000 incorporates an on-chip memory management unit which controls the virtual memory system's page mapping, the operating modes, and exception handling. It consists of an instruction translation lookaside buffer (ITLB), a joint TLB, and a coprocessor register file.

4.2.1 Joint TLB

For fast virtual-to-physical instruction and data address translation, the R4000 uses a 96-entry translation lookaside buffer. This fully associative TLB is arranged in 48 even-odd page pairs. The TLB maps a 32-bit virtual address and an 8-bit address space identifier into a 36-bit physical address to access 64 GBytes of physical memory.

Two mechanisms are provided to assist in controlling the amount of mapped space and the replacement characteristics of the space. First, the page size can be configured, on a per-entry basis, to equal 4 KBytes to 16 MBytes in increments of powers of four. The page size is determined by the value of the CP0 *SIZE* register when the virtual-physical pair is loaded into the TLB. This feature is especially useful in display systems for mapping large physical regions such as frame

buffers, or in engineering and scientific applications where large data arrays are being manipulated. The large page size allows the TLB to map up to 1.6 GBytes of memory at any one time.

The second feature is extremely important in systems where a deterministic response time to an interrupt is necessary. Each entry in the TLB can be "locked" to guarantee that the translations remains in the buffer. The number of these entries is controlled by the CP0 *WIRED* register.

The TLB also contains information to control the cache coherency protocol for each page. Specifically, three attribute bits associated with each page determine whether the coherency protocol is uncached, sharable, update, exclusive, or non-coherent. These per-page attributes allow a more efficient multiprocessor system to be built by defining the level of data sharing for each page. The R4000PC package supports only uncached and noncoherent coherency attributes.

4.2.2 Instruction TLB

The R4000 also incorporates a 2-entry instruction TLB. Each entry maps a 4 KByte page. The instruction TLB improves performance by allowing instruction address translation to occur in parallel with data address translation. When a miss occurs on an instruction address translation, the ITLB is filled from the JTLB. The operation of the ITLB is invisible to the user.

4.2.3 System Control Coprocessor Register File

The R4000's system control coprocessor (CP0) registers provide the path through which the virtual memory system's page mapping is examined and changed, the operating modes (kernel vs. user mode, interrupts enabled or disabled, cache features) controlled, and exceptions handled. In addition, the R4000 includes registers to implement a real-time cycle counting facility, to address reference traps for debugging, to aid in cache diagnostic testing, and to assist in data error detection and correction.

4.3 Integer Unit

The R4000 integer processing unit contains a 32-entry register file, an ALU, and a dedicated multiplier/divider. The integer unit is the core processing unit in the R4000 and is responsible for instruction fetching, integer operation decoding and execution, and load and store operation execution.

4.3.1 Register File

The R4000 has thirty-two general purpose registers either 32- or 64-bits wide depending on operating mode. These registers are used for scalar integer operations and address calculation. The register file consists of two read ports and one write port, and uses bypassing to enable the reading and writing of the same register twice per cycle, which minimizes the operation latency in the pipeline.

4.3.2 ALU

The R4000 ALU consists of the integer adder and logic unit. The adder performs address calculations in addition to arithmetic operations, and the logic unit performs all shift operations. Each of these units is highly optimized and can perform an operation in a single Superpipeline cycle.

4.3.3 Integer Multiplier/Divider

The R4000 integer multiplier and divider units perform 32-bit and 64-bit signed and unsigned multiply and divide operations and execute instructions in parallel with the ALU. The results of the operation are placed in the *MDHI* and *MDLO* registers. The values can then be transferred to the general purpose register file using the Mfhi/Mflo instructions. Table 4-1 shows the number of processor internal cycles required between an integer multiply or divide and a subsequent Mfhi or Mflo operation, in order that no interlock or stall occurs.

Table 4-1: Processor Internal Cycles.

operation	32-bit signed	32-bit unsigned	64-bit signed	64-bit unsigned
MULT	12	12	20	20
DIV	78	78	133	133

4.4 Floating-Point Unit

The R4000 incorporates an entire floating-point unit on chip, including a floating-point register file and execution unit. The floating-point unit forms a "seamless" interface with the integer unit, decoding and executing instructions in parallel with the integer unit.

4.4.1 Floating-Point Execution Unit

The R4000 floating-point execution unit supports single- and double-precision arithmetic, as specified in the IEEE standard 754. The execution unit is broken into separate multiply, divide, and add/convert/square root units, which allows for overlapped operations. The adder is pipelined, allowing a new add to begin every 4 cycles.

As in the R2010 and R3010, the R4000 maintains fully precise floating-point exceptions while allowing both overlapped and pipelined operations. Precise exceptions are extremely important in mission-critical environments, such as ADA, and highly desirable for debugging in any environment.

The floating-point unit's operation set includes floating-point add, subtract, multiply, divide, square root, conversion between fixed-point and floating-point format, conversion among floating-point formats, and floating-point compare. These operations comply with the IEEE Standard 754.

Table 4-2 gives the latencies of some of the floating-point instructions in internal processor cycles.

Table 4-2: Latencies of Floating-Point Instructions

operation	single precision	double precision
ADD	4	4
SUB	4	4
MUL	7	8
DIV	23	36
SQRT	54	112
CMP	3	3
FIX	4	4
ROUND	4	4
TRUNC	4	4
FLOAT	5	5
ABS	2	2
MOV	1	1
NEG	2	2
LWC1,LDC1	3	3
SWC1,SDC1	1	1

4.4.2 Floating-Point General Register File

The floating-point register file is made up of sixteen 64-bit registers which can also be configured as thirty-two 32-bit floating-point registers. The MIPS architecture supports a coprocessor load and store double so, when configured as 64-bit wide registers, the floating-point unit can take advantage of the 64-bit wide data cache and issue a coprocessor load or store a doubleword instruction in every cycle.

4.4.3 Floating-Point Control Register File

The floating-point control registers contain a register for determining configuration and revision information for the coprocessor and control and status information. These are primarily involved with diagnostic software, exception handling, state saving and restoring, and control of rounding modes.

Interface Overview

5

This section discusses the three R4000 processors and how these processors interact with the rest of a system. Each of the external interfaces on the processors is explained briefly.

5.1 R4000 Processors

The R4000 processor is available in three different configurations: the R4000MC and R4000SC, which include a 128-bit wide secondary cache bus, and the R4000PC, with no secondary cache interface.

5.1.1 R4000SC

The R4000SC is available in a 447-pin Land Grid Array (LGA) or Pin Grid Array (PGA). This processor supports a secondary cache interface and is ideal in systems where high performance is desired. This component supports a 128 KByte to 4 MByte secondary cache made from standard SRAMs. This flexibility allows system designers to make price/performance trade-offs in cache subsystem designs.

5.1.2 R4000MC

The R4000SC is also available in a 447-pin Land Grid Array (LGA) or Pin Grid Array (PGA). This processor supports a secondary cache and configurable multiprocessor cache coherency protocols. Like the R4000SC, this processor also supports a 128 KByte to 4 MByte secondary cache made from standard SRAMs. The R4000MC is well suited for a range of designs from high-performance desktop systems to fault tolerant multiprocessor servers.

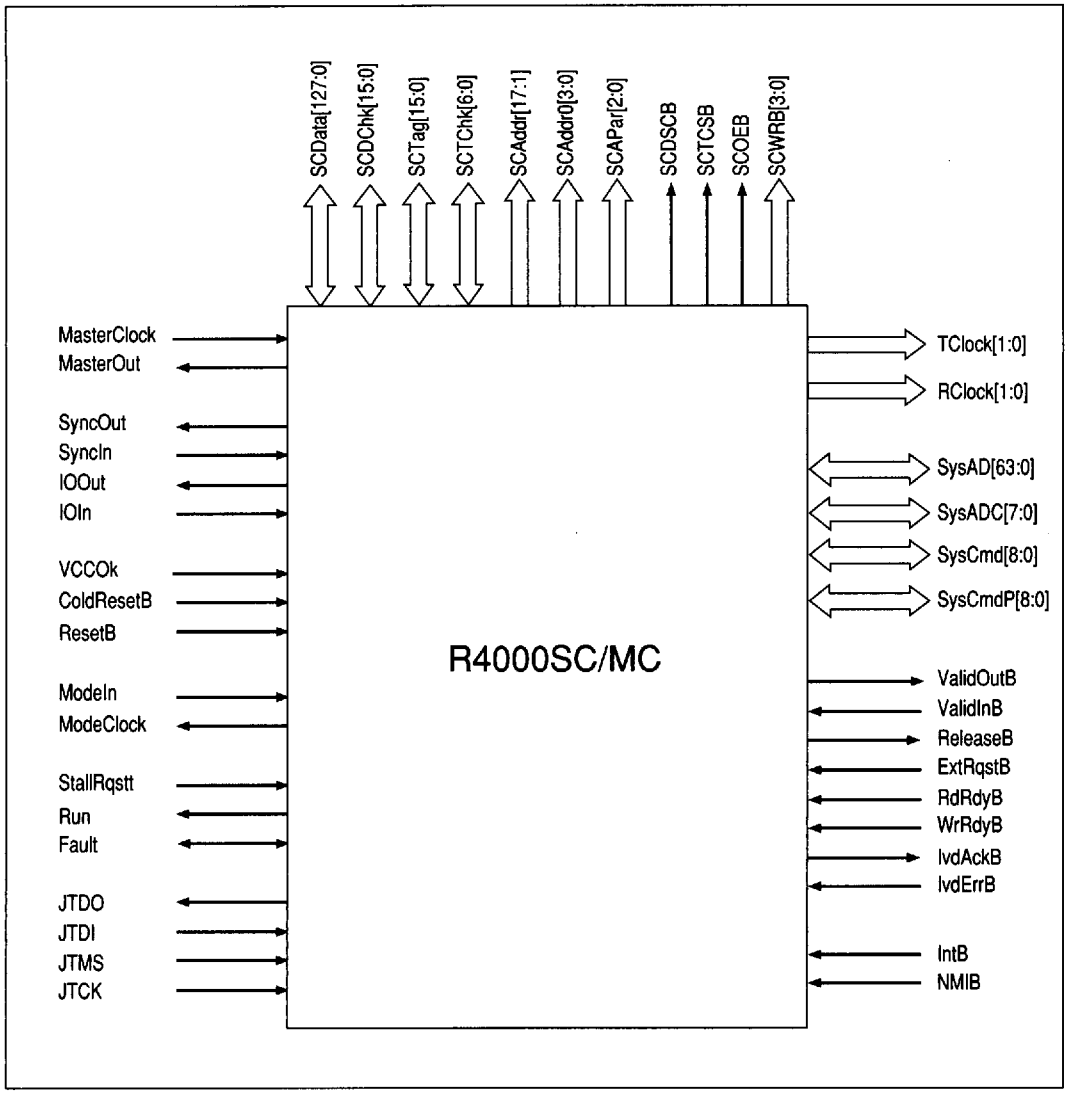

Figure 5-1 R4000SC/MC Symbolic Pinout

5.1.3 R4000PC

The R4000PC is available in a 179-pin Pin Grid Array (PGA). This configuration supports no secondary cache or cache coherency and is ideal for applications such as high-performance embedded control and low-cost desktop systems, where the on-chip caches provide

enough performance and where cost, power, and board space must be kept to a minimum. By eliminating a secondary cache, a system can be designed with fewer parts.

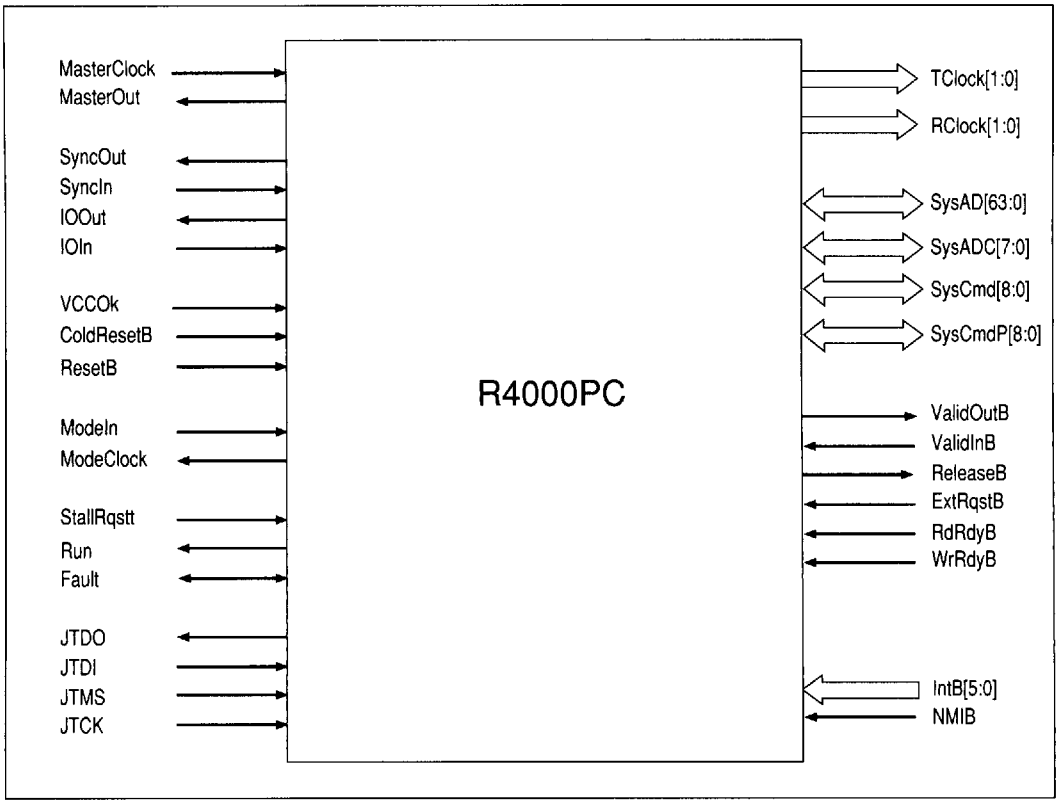

Figure 5-2 R4000SP Symbolic Pinout

5.2 Secondary Cache Interface

The R4000SC and R4000MC support a secondary cache that can range in size from 128 KBytes to 4 MBytes. The cache can be configured as a unified cache or split into an instruction cache and a data cache, and it can be designed using industry standard SRAMs. The R4000 provides all of the secondary cache control circuitry on chip, including ECC.

The secondary cache interface consists of a 128-bit data bus, a 25-bit tag bus, an 18-bit address bus, and SRAM control signals. The wide data bus improves performance by providing a high bandwidth data

path to fill the primary caches. ECC check bits are added to both the data and tag buses to improve data integrity. All double-bit errors can be detected and all single bit errors can be corrected on both buses.

The secondary cache access time is configurable, providing system designers with the flexibility to tailor the cache design to specific applications. The line size of the secondary cache is also configurable and can be 4, 8, 16, or 32 words. The line size of the primary cache must always be less than or equal to the line size of the secondary cache.

The secondary cache is physically tagged and physically indexed. The physical cache prevents problems that could arise due to virtual address aliasing. Also, a physical cache makes multiprocessing cache coherency protocols easier to implement. The R4000MC provides a set of cache states and a mechanism for manipulating the contents and state of the cache, which are sufficient to implement a variety of cache coherency protocols, using either bus snooping or directory-based schemes.

5.3 Clocking/Frequency

The R4000 processor bases all clocking methodology on a single clock input. This clock is then multiplied by two using internal phase lock loop techniques in order to get the internal pipeline operating frequency. The phase lock loops also eliminate skew in all clocks used and generated by the R4000.

This internal clock is then divided to generate two system interface clocks, RClock and TClock. The frequency of these clocks is programmable and can be set to be either half, one third, or one fourth the processor internal clock frequency. This feature makes it easier to interface to memory and I/O systems of various frequencies.

5.4 System Interface

The R4000 supports a 64-bit system interface that can be used to construct systems as simple as a uniprocessor with a direct DRAM interface and no secondary cache or as sophisticated as a fully cache coherent multiprocessor. The interface consists of a 64-bit Address/Data bus with 8 check bits and a 9-bit command bus protected with parity. In addition, there are 8 handshake signals. The interface has a simple timing specification and is capable of transferring data between the processor and memory at a peak rate of 400 MBytes/second at 50 MHz.

Figure 5-3 shows a typical desktop system using the R4000PC. A high-performance desktop workstation/server system can be built using the R4000SC with a secondary cache.

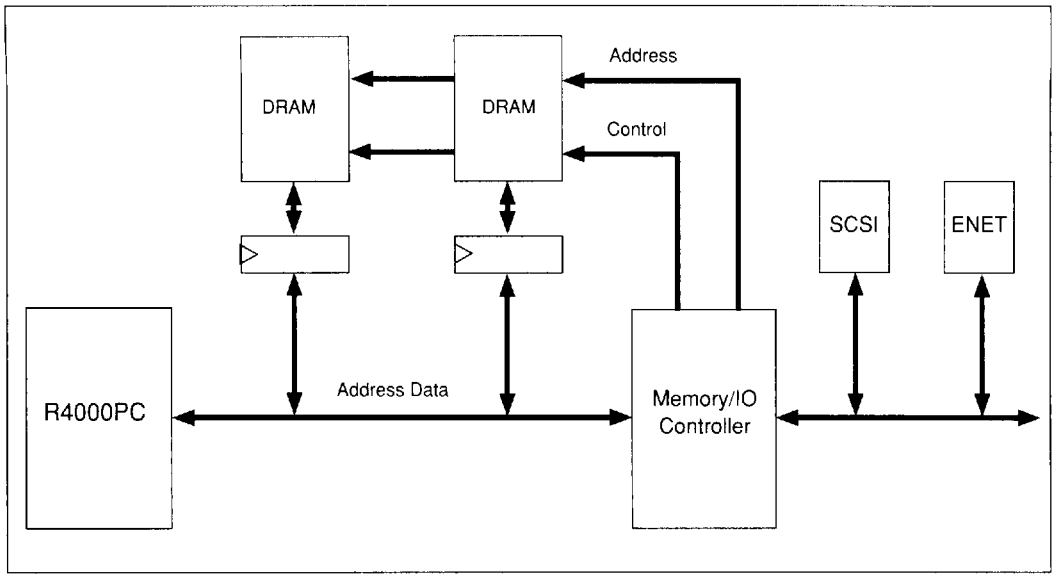

Figure 5-3 Typical Desktop System Block Diagram

The system interface allows the processor to access external resources in order to satisfy cache misses and uncached operations. The R4000MC, in addition to handling simple memory and I/O transactions, supports a number of cache coherency transactions of sufficient generality to support a variety of cache coherent multiprocessing models. In particular, the interface is designed to support both bus snooping and directory based multiprocessor models and supports both write-update and write-invalidate coherency protocols.

Figure 5-4 shows a typical multiprocessor system using the R4000MC, an interface ASIC, and a secondary cache.

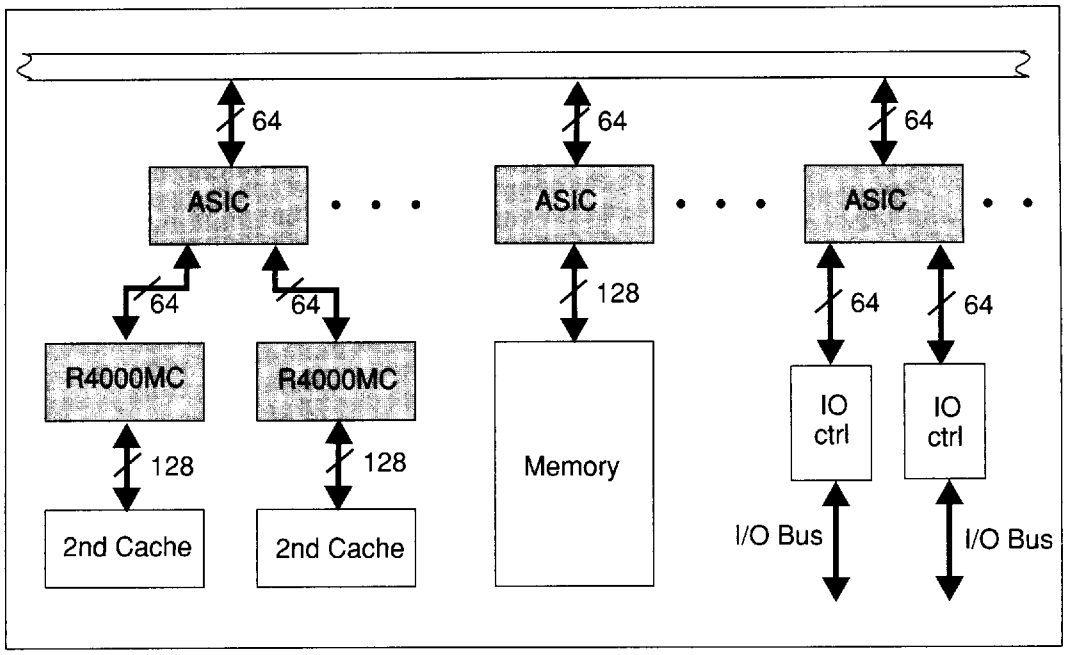

Figure 5-4 Multiprocessor System using the R4000

5.4.1 System Address/Data Bus

The 64-bit System Address Data (SysAD) bus is used to transfer addresses and data between the R4000 and the rest of the system. It is protected with an 8-bit check bus, SysADC. The check bits can be configured as either parity or ECC, for flexibility in interfacing to either parity or ECC memory systems.

The system interface is configurable to allow easier interfacing to memory and I/O systems of varying frequencies. The data rate and the bus frequency at which the R4000 transmits data to the system interface are programmable via boot time mode control bits. Also, the rate at which the processor receives data is fully controlled by the external device. Therefore, either a low-cost interface requiring no write buffering or a fast, high-performance interface can be designed to communicate with the R4000. Again, the system designer has the flexibility to make these price/performance tradeoffs.

5.4.2 System Command Bus

The R4000 interface has a 9-bit System Command (SysCmd) bus, which indicates whether the SysAD bus carries an address or data. If the SysAD carries an address, then the SysCmd bus also indicates what type of transaction is to take place (for example, a read or write). If the SysAD carries data, then the SysCmd bus also gives information about the data (for example, this is the last data word transmitted, or the cache state of this line of data is clean exclusive). The SysCmd bus is bidirectional to support both processor requests and external requests to the R4000. Processor requests are initiated by the R4000 and responded to by an external device. External requests are issued by an external device and require the R4000 to respond.

The R4000 supports byte, halfword, tribyte, word, doubleword, and block transfers on the SysAD bus. In the case of a sub-doubleword transfer, the low-order 3 address bits gives the byte address of the transfer, and the SysCmd bus indicates the number of bytes being transferred.

5.4.3 Handshake Signals

There are eight handshake signals on the system interface. Two of these, **RdRdy*** and **WrRdy***, are used by an external device to indicate to the R4000 whether it can accept a new read or write transaction. The R4000 samples these signals before deasserting the address on read and write requests.

ExtRqst* and **Release*** are used to transfer control of the SysAD and SysCmd buses between the processor and an external device. When an external device needs to control the interface, it asserts **ExtRqst***. The R4000 responds by asserting **Release*** to release the system interface to slave state.

ValidOut* and **ValidIn*** are used by the R4000 and the external device respectively to indicate that there is a valid command or data on the SysAD and SysCmd buses. The R4000 asserts **ValidOut*** when it is driving these buses with a valid command or data, and the external device drives **ValidIn*** when it has control of the buses and is driving a valid command or data.

Finally, there are two signals that are available on the large package only and are used in multiprocessing systems. They are **IvdAck*** and **IvdErr***, and they are driven by an external device to indicate the completion status of the current processor invalidate or update request.

5.4.4 R4000 Requests

The R4000 is capable of issuing requests to a memory and I/O subsystem. The system interface supports two modes of operation:

- overlap mode
- non-overlap mode

Non-Overlap Mode

The R4000PC requires a non-overlap system interface. This means that only one processor request may be outstanding at a time and that the request must be serviced by an external device before the R4000 issues another request. The R4000PC can issue read and write requests to an external device, and an external device can issue read and write requests to the R4000.

Figure 5-5 shows a processor read request. The R4000 asserts **ValidOut*** and simultaneously drives the address and read command on the SysAD and SysCmd buses. If the system interface has **RdRdy*** asserted, then the processor tristates its drivers and releases the system interface to slave state by asserting **Release***. The external device can then begin sending the data to the R4000.

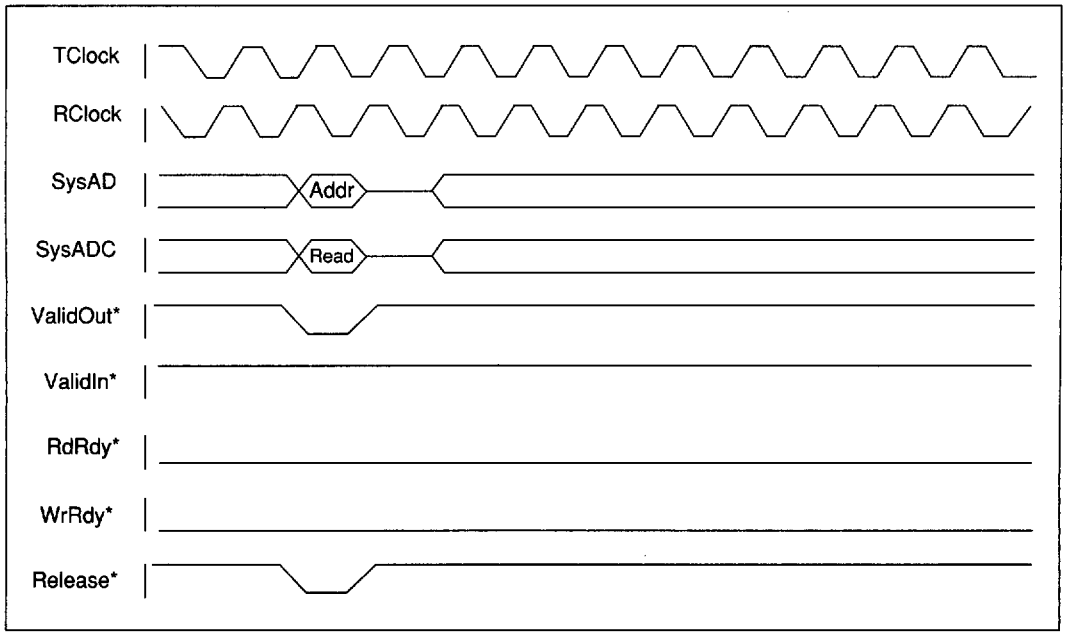

Figure 5-5 Processor Read Request

Overlap Mode

The R4000SC and R4000MC, when configured with a secondary cache, must operate in overlap mode in which they may issue multiple system interface transactions in parallel. The processor may issue a combination of a read request, an update or invalidate request, and a write request. For instance, when a dirty cache line needs to be replaced, the processor issues a read request immediately followed by a write request, without waiting for the read data to return. This has the advantage of "hiding" the write transaction between the read request and read response, thus increasing overall system performance. Overlap mode is not necessary or useful in the R4000PC since the processor contains a write buffer capable of accepting an entire primary cache line of data.

Figure 5-6 illustrates a processor request in overlap mode. This request is made up of a read, invalidate, and write request. Note that the protocol for the read, the invalidate, and the write are all similar to each other, with the exception that the processor also sends out valid data during the write request. In Figure 5-6 the processor write transaction not only occurs before the read response from the external device, but it also illustrates how an external device can hold off a write request through the deassertion of **WrRdy***.

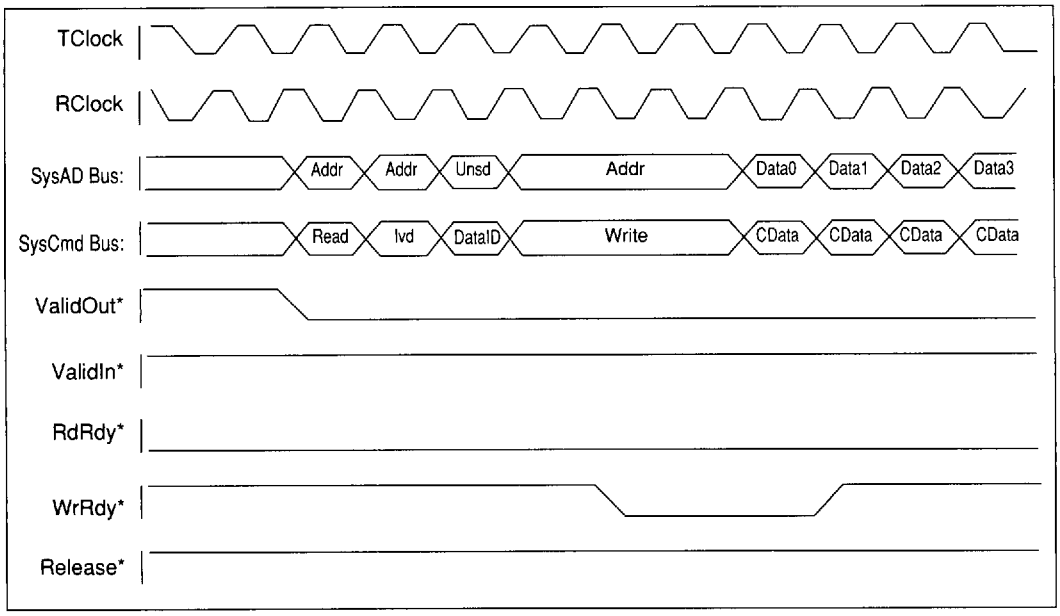

Figure 5-6 Processor Read, Invalidate, Write Request

5.4.5 External Requests

The R4000 responds to requests issued by an external device. The requests can take several forms. An external device may need to supply data in response to an R4000 read request or it may need to gain control over the system interface bus to access other resources which may be on that bus. It also may issue cache coherency requests to the processor, such as a request for the R4000 to update, invalidate, or snoop upon its caches, or to supply a cache line of data. Additionally, an external device may need to write to the R4000 interrupt register.

The following is a list of the supported external requests:

- Read
- Write
- Invalidate
- Update
- Snoop
- Intervention
- Null

Figure 5-7 shows an example of an external snoop request. The process by which the external device issues the request is very similar to the way the R4000 issues a request. The external device first gains ownership of the system interface by asserting **ExtRqst*** and waiting for the R4000 to assert **Release***. The external device then sends in a valid command by asserting **ValidIn*** and driving the SysCmd and SysAD buses with the snoop command and address. The R4000 responds to the request by asserting ValidOut* and driving the SysCmd bus with the cache state of the snooped upon line.

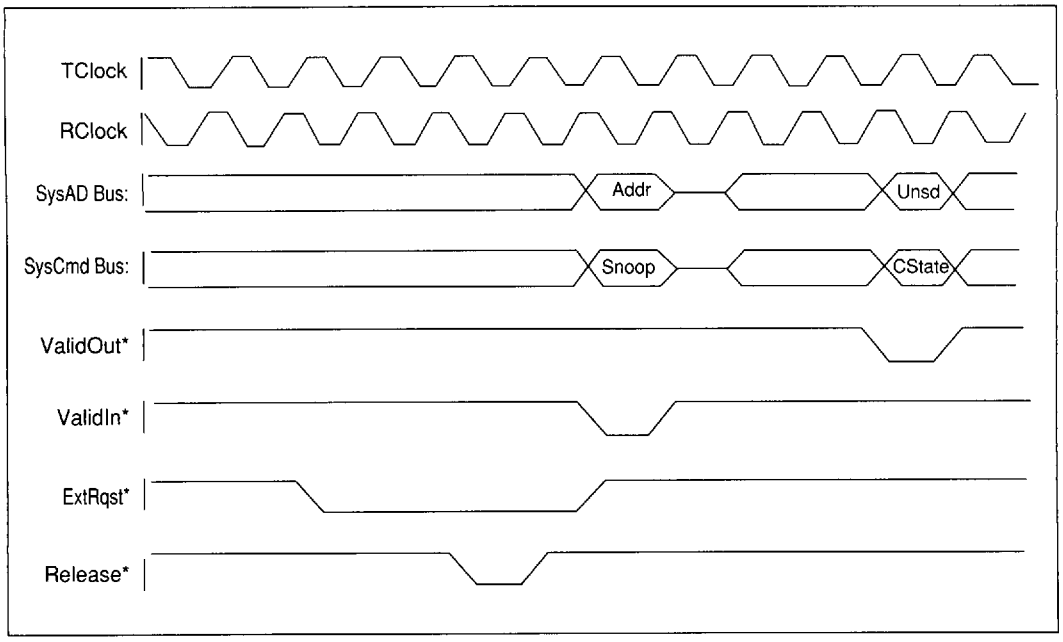

Figure 5-7 External Snoop Request

5.5 Interrupts

The R4000 processor supports six hardware interrupts and two software interrupts. The six hardware interrupts are accessible via an external write request in all three configurations, and the also through six dedicated interrupt pins on the R4000PC.

5.6 JTAG Boundary Scan

The R4000 implements JTAG boundary scan, which is intended to provide a capability for testing the interconnect between the R4000 processor, the printed circuit board, and the other components on the board. In addition, the mechanism is intended to provide a means to test the secondary cache RAM. In accordance with the JTAG specification, the R4000 contains a TAP controller, JTAG instruction registers, JTAG boundary scan register, JTAG identification register, and JTAG bypass register.

5.7 Boot Time Mode Control

Fundamental operational modes for the R4000 are initialized via the boot time mode control interface. This serial interface operates at a very low frequency (1/64th of the input clock frequency), which allows the initialization information to be kept in a low-cost EPROM. Upon reset, the R4000 reads the serial PROM to access the configuration information.

5.8 Resets

The R4000 processor uses a multi-level reset sequence that allows the implementation of a power-on reset, cold reset, and warm reset. Upon power on and cold reset, all processor internal state machines reset, and the R4000 begins fetching instructions from hex address BFC00000, which is in uncached, unmapped space. During a warm reset, however, processor internal state is preserved.

5.9 Fault Tolerant Support

Two R4000s can be connected together to form a self-checking pair. In this mode, each R4000 checks the other and a difference is indicated with the assertion of a fault signal. This provides a simple fault detection mechanism requiring minimal additional logic. The self-checking mechanism provides an essential building block for producing fault tolerant systems.