mips
Open RISC Technology

R4400 MICROPROCESSOR

PRODUCT INFORMATION

Satya Simha

MIPS Technologies, Inc.
2011 N. Shoreline Blvd
P.O. Box 7311
Mountain View, CA 94039-7311

Publication date: March 22, 1996

MIPS Technologies, Inc. reserves the right to make changes to any products herein at any time without notice in order to improve function or design. MIPS does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under patent rights nor imply the rights of others.

The R4000 /R4400 is available from the following manufacturers:

Integrated Device Technology, Inc. (Attention - RISC Microprocessor Marketing)
2975 Stender Way
Santa Clara, CA   95052-8015
Tel: (408) 727 6116

LSI Logic Corporation (Attention - MIPS Division)
1551 McCarthy Blvd.
Milpitas, CA   95035
Tel: (408) 433 8000

NEC Corporation (Attention - Microcomputer Division)
401 Ellis St.
Mountain View, CA 94039
Tel: (415) 960-6000

Performance Semiconductor Corporation (Attention - Microprocessor Marketing)
610 E. Weddell Drive
Sunnyvale, CA   94089
Tel: (408) 734 9000

Siemens Components Inc. (Attention - Integrated Circuit Division)
10950 Tantau Ave.
Cupertino, CA 95014
Tel: (408) 777-4500

Toshiba Corporation
9740 Irvine Blvd.
Irvine, CA 92713
(714) 583-3000

1.0 Features

- True 64-bit microprocessor
    - G 64-bit integer operations
    - G 64-bit floating point operations
    - G 64-bit registers
    - G 64-bit virtual address space
- High performance microprocessor
    - G Superpipelined architecture
    - G No issue restrictions for dual instruction issue
    - G Over 90 SPECint and over 100 SPECfp at 75 MHz
- 50/67/75 MHz input and output clock frequency
    - G On-chip clock doubler for internal 100/134/150 MHz pipeline
- High level of integration
    - G 64-bit integer CPU
    - G 64-bit floating point unit
    - G Direct mapped, separate instruction and data caches (16 KB each)
    - G Flexible MMU with 48-entry TLB mapped into even/odd page pairs (96 pages)
- 64 GB physical address space
- Available in 179-pin PGA (R4400PC) & 447-pin PGA (R4400SC/MC)
- Write-back cache
- Burst reads and writes of 32 or 64 bytes
- Fully binary compatible with the R3000A family of microprocessors
- Standard operating system support includes:
    - G Microsoft Windows NT
    - G UNIX System V.4
- Processor family for a wide variety of applications
    - G Personal computers running NT
    - G Desktop workstations
    - G Deskside or departmental servers
    - G High-performance embedded applications

2.0 Description

The R4400 supports a wide variety of processor based applications from 32-bit desktop systems through high-performance 64-bit on line transaction processing (OLTP) systems manipulating large data bases. The R4400 provides complete upward application software compatibility with the R3000 family of microprocessors as well as with the R4200 family. Microsoft Windows NT and UNIX V.4 insure the availability of thousands of application programs geared to provide a complete solution to a large number of processing needs. The R4400 provides a good balance of integer and floating point performance using a super-pipelined architecture. It also is a true 64-bit processor with a 64-bit address space as well as 64-bit computing and at the same time allowing for 32-bit applications to run as well. This data sheet provides an overview of the R4400 microprocessor. For further detail, please refer to the R4000 User’s Manual as well as the data sheets provided by the various R4400 manufacturers.

2.1 Data Format

The R4400 defines a 64-bit doubleword, a 32-bit word, a 16-bit half word, and an 8-bit byte. The byte ordering is configurable either in big-endian or little-endian format.

3.0 R4400 Family

The R4400 processor is available in three different configurations: the R4400MC and the R4400SC, which include a 128-bit secondary cache bus; and the R4400PC, with no secondary cache interface.

3.1 R4400PC Configuration

The R4400PC is available in a 179-pin Pin Grid Array (PGA). This configuration does not support a secondary cache interface directly nor does it implement cache coherency in hardware. This is ideal for applications such as high-performance embedded applications or low-cost desktop systems.

3.2 R4400SC Configuration

The R4400SC is available in a 447-pin Pin Grid Array (PGA). This processor supports a secondary cache interface and is suited for high-performance applications. The component supports a 128 KByte to 4 MByte secondary cache which can be designed using standard static RAMs.

3.3 R4400MC Configuration

The R4400MC is also available in the 447-pin Pin Grid Array (PGA). Like the R4400SC, the R4400MC also supports a secondary cache interface which can be used to design a system with secondary cache ranging from 128 KBytes to 4 MBytes in size using standard static RAMs. In addition, it supports configurable cache coherency protocols for multiprocessing systems.

4.0 Block Diagram of the R4400

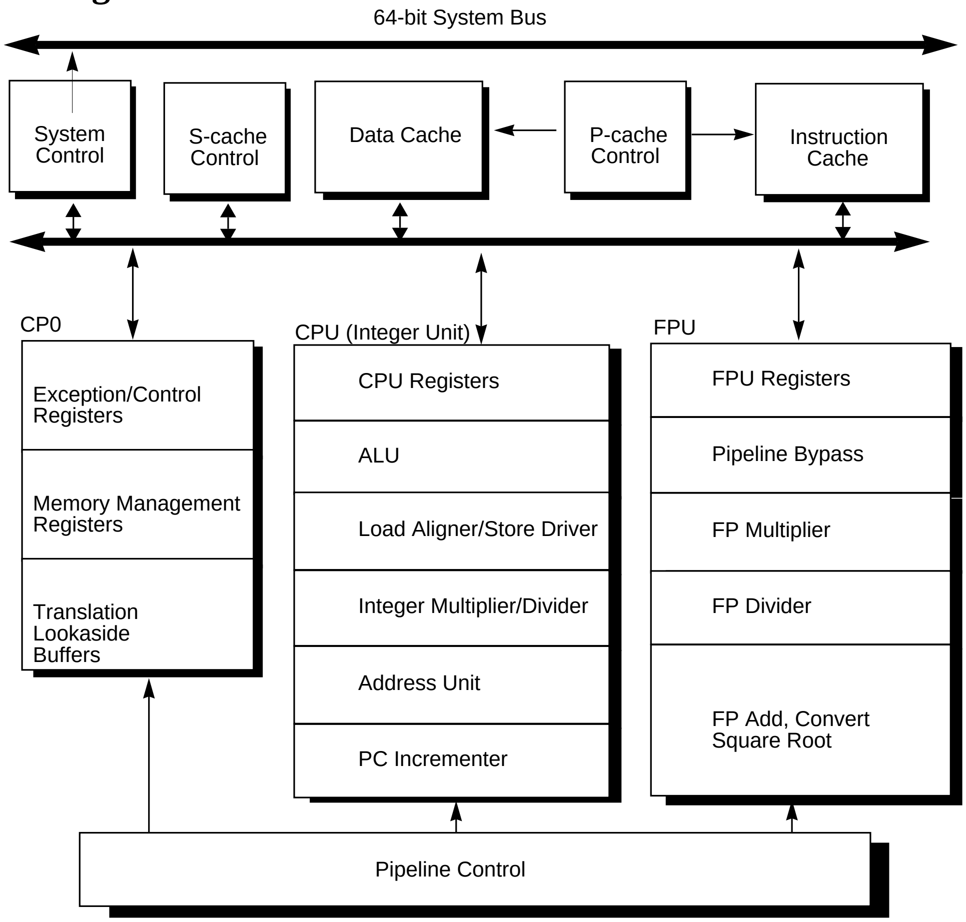


Figure 1 Block diagram of the R4400

5.0 Integer Unit

The R4400 has thirty-two 64-bit general purpose registers. The integer unit is capable of handling 64-bit operands. The integer unit also has separate multiply and divide units that allow multiply or divide operations to take place in parallel with other instructions.

5.1 CPU Register Overview

The CPU provides the following registers

- 32 general purpose registers
- a Program Counter (*PC*) register
- 2 registers that hold the results of integer multiply and divide operations (*HI* and *LO*).

CPU registers can be either 32 bits or 64 bits wide, depending on the R4400 processor mode of operation. Figure 2 shows the register set of the R4400 microprocessor.

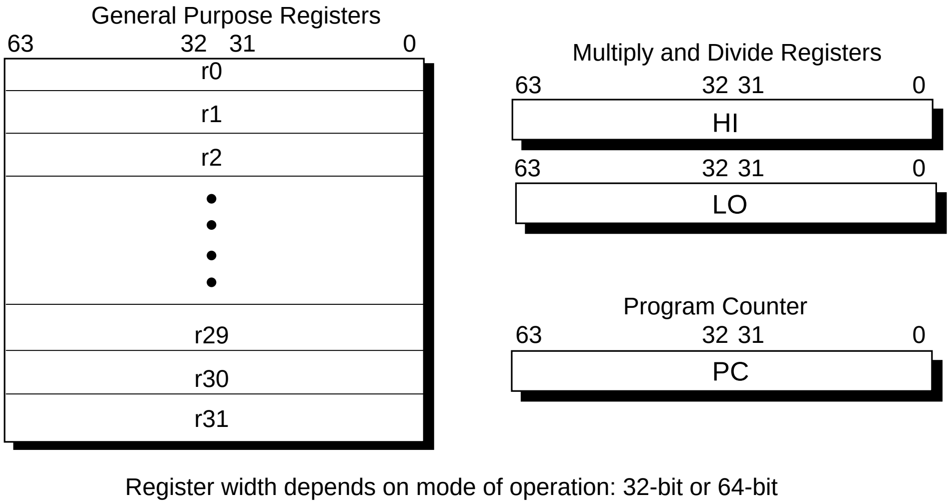

General Purpose Registers
63
32
31
0
r0
r1
r2
r29
r30
r31
Multiply and Divide Registers
63
32 31
0
HI
63
32 31
0
LO
Program Counter
63
32 31
0
PC
Register width depends on mode of operation: 32-bit or 64-bit

Figure 2 Register Set of the R4400

Two of the CPU general purpose registers have assigned functions:

- *r0* is hardwired to a value of zero, and can be used as the target register for any instruction whose result is to be discarded. *r0* can also be used as a source when a zero value is needed.
- *r31* is the link register used by Jump and Link instructions. It should not be used by other instructions.

The CPU has three special purpose registers:

- *PC* — Program Counter register
- *HI* — Multiply and Divide register higher result
- *LO* — Multiply and Divide register lower result

The two registers HI and LO hold:

- the product of integer multiply operations, or
- the quotient (in *LO*) and remainder (in *HI*) of integer divide operations

5.2 Pipeline

The integer unit (shown in Figure 1)has an 8-stage pipeline. Each stage in the pipeline takes one clock cycle that is twice the frequency of the supplied input clock. The eight stages of the pipeline are shown in Figure 3 . An instruction execution will therefore take eight clock cycles. An instruction can take longer if the required data is not in the cache. In other words, the R4400 can execute an instruction every cycle.

The eight pipeline stages are described below:

IF - Instruction Fetch, First Half

During the IF stage, the following occurs:

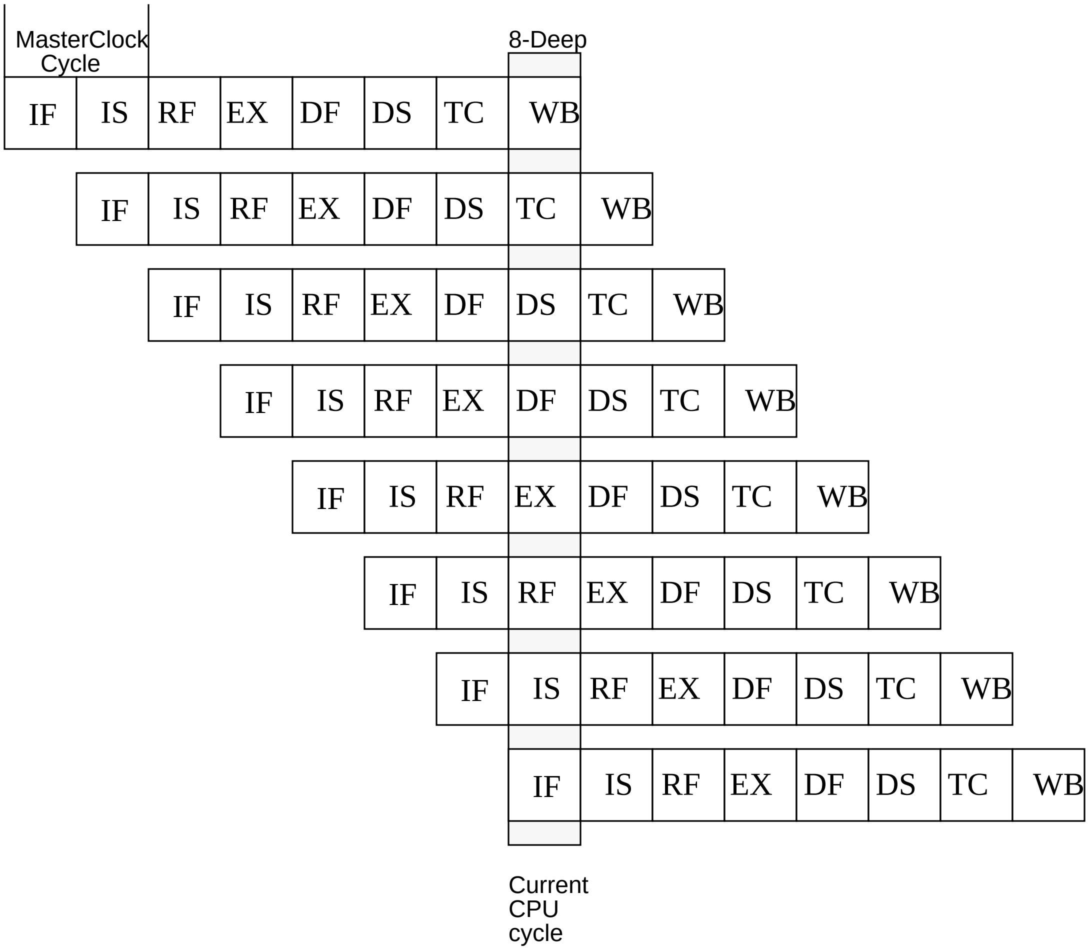


Figure 3 Instruction pipeline stages in the R4400

- Branch logic selects an instruction address and the instruction cache fetch begins.
- The instruction translation lookaside buffer (ITLB) begins the virtual-to-physical address translation.

IS - Instruction Fetch, Second Half

During the IS stage, the instruction cache fetch and the virtual-to-physical address translation are completed.

RF - Register Fetch

During the RF stage, the following occurs:

- The instruction decoder (IDEC) decodes the instruction and checks for interlock conditions.
- The instruction cache tag is checked against the page frame number obtained from the ITLB.
- Any required operands are fetched from the register file.

EX - Execution

During the EX stage, one of the following occurs:

- The arithmetic logic unit (ALU) performs the arithmetic or logical operation for register-to-register instructions.
- The ALU calculates the data virtual address for load and store instructions.
- The ALU determines whether the branch condition is true and calculates the virtual branch target address for branch instructions.

DF - Data Fetch, First Half

During the DF stage, one of the following occurs:

- The data cache fetch and the data virtual-to-physical translation begins for load and store instructions.
- The branch instruction address translation and translation lookaside buffer (TLB)[1] update begins for branch instructions.
- No operations are performed during the DF, DS, and TC stages for register-to-register instructions.

DS - Data Fetch, Second Half

During the DS stage, one of the following occurs:

- The data cache fetch and data virtual-to-physical translation are completed for load and store instructions. The Shifter aligns data to its word or doubleword boundary.
- The branch instruction address translation and TLB update are completed for branch instructions.

TC - Tag Check

For load and store instructions, the cache performs the tag check during the TC stage. The physical address from the TLB is checked against the cache tag to determine if there is a hit or a miss.

WB - Write Back

For register-to-register instructions, the instruction result is written back to the register file during the WB stage. Branch instructions perform no operation during this stage.

5.3 Integer Multiply/Divide

The R4400 has a separate unit to perform integer multiply or divide operations. Registers HI and LO in Figure 2 are used to hold the result of a multiply operation. The results can be read using the MFHI and MFLO instructions. The following Table describes the number of processor cycle

[1] The TLB is described in Chapter 4.

required to resolve an interlock or stall between various multiply or divide instructions and a subsequent Move From HI register (MFHI) or Move From LO register (MHLO) instruction.

Table 1: Multiply/Divide Instruction Cycle timing in PCycles

Instruction	Single word	Double word
MULT	10	20
DIV	69	133

6.0 Primary Caches

The R4400 has separate instruction and data caches. The caches are organized as direct mapped and are each 16 KB in size. The R4400 caches allow the processor pipeline to execute at the rate of one clock cycle per instruction and also minimizes the load latency. The caches are virtually indexed to allow data accesses to take place in parallel with the address translation, but are physically tagged to eliminate unnecessary cache flushing. The caches implement a writeback mechanism for write operations.

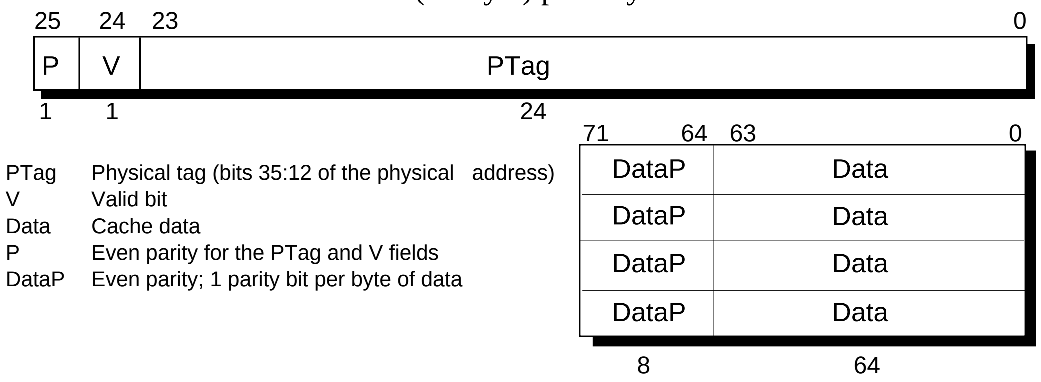
6.1 Primary Instruction Cache (I-Cache) Organization

Each line of primary I-cache data (although it is actually an instruction, it is referred to as data to distinguish it from its tag) has an associated 26-bit tag that contains a 24-bit physical address, a single valid bit, and a single parity bit. Byte parity is used on I-cache data.

The R4400 processor primary I-cache has the following characteristics:

- direct-mapped
- indexed with a virtual address
- checked with a physical tag
- organized with either a 4-word (16-byte) or 8-word (32-byte) cache line.

Figure 4 shows the format of an 8-word (32-byte) primary I-cache line.



PTag	Physical tag (bits 35:12 of the physical address)
V	Valid bit
Data	Cache data
P	Even parity for the PTag and V fields
DataP	Even parity; 1 parity bit per byte of data

Figure 4 Primary Instruction Cache format

6.2 Data Cache

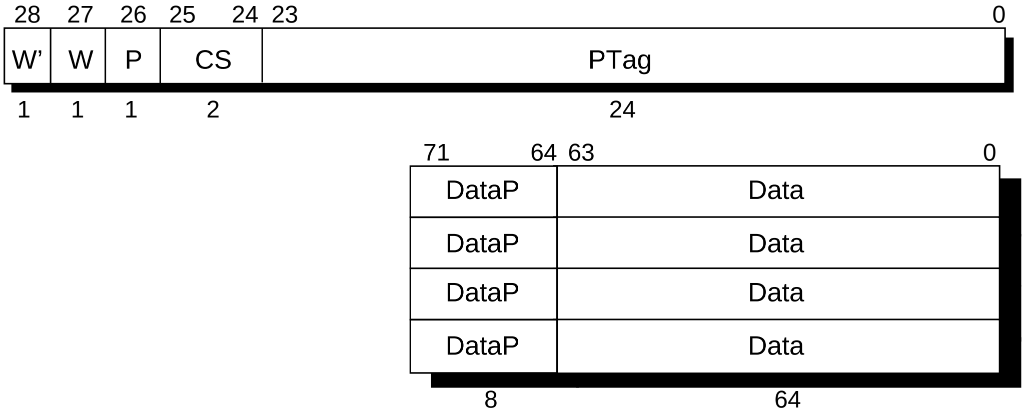
The R4400 also implements a separate data cache for fast accesses to data. The data cache is 16 KB in size and is direct mapped. The data cache has a line size of 4-words or 8-words.

The data cache is protected by byte parity like the instruction cache. The tag on the data cache is protected with a single parity bit. The data cache is virtually indexed and is physically tagged. Unlike the instruction cache, the data cache contains two cache state bits that indicate whether the cache line is valid, shared, clean exclusive, or dirty exclusive.

In addition, there is a W bit in the data cache which is protected by a parity bit called the W’ bit. It is the W bit and not the cache state bits that indicates whether or not the primary cache contains modified data that must be written back to memory. Figure 5 shows the format of an 8-word primary data cache line.

The data cache has a store buffer associated with it. When the R4400 executes a store instruction, the store data gets written into the 2-entry store buffer while the tag comparison is being performed. If the tag matches the data gets written into the data cache in the next cycle that the data cache is not accessed. The store buffer allows the R4400 to execute two stores per master cycle and to perform back to back stores without penalty, yielding a high bandwidth without restriction on instruction combinations.

When the data cache does need to be written to memory, the R4400 writes the data to an internal write buffer which can hold a line (4 or 8 words) of data. This allows the processor to continue executing instructions without having to wait for writes to be retired to the memory system.



W’ Even parity for the write-back bit
W Write-back bit (set if cache line has been written)
P Even parity for the PTag and CS fields
CS Primary cache state:

0 = Invalid in all R4400 configurations
1 = Shared (either Clean or Dirty) in R4000MC configuration only
2 = Clean Exclusive in R4000SC and MC configurations only
3 = Dirty Exclusive in all R4000 configurations

PTag Physical tag (bits 35:12 of the physical address)
DataP Even parity for the data
Data Cache data

Figure 5 R4400 8-word primary Data Cache Line Format

6.3 Write Buffer (also Store Buffer)

The R4400 contains a one-deep write buffer for uncached store operations. The write buffer helps in avoiding stalls when there are multiple write operations and this improves performance significantly.

7.0 Secondary Cache Interface

The R4400SC and R4400MC support a secondary cache that can range in size from 128 KB to 4 MB. The secondary cache can be configured as a unified cache or split into instruction and data caches. The secondary cache can be designed using industry standard SRAMs. The R4400 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 control signals for SRAMs such as output enables and write enables. The wide data bus improves performance by providing a high bandwidth data path to fill the primary caches. ECC 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 detected and corrected on both tag and data buses.

The secondary cache access time is configurable during initialization. The line size of the secondary cache is also configured during reset and can be 4, 8, 16, or 32 words. It is imperative however to make the line size of the primary cache to be always less than or equal to the secondary cache.

The secondary cache is physically tagged and physically indexed. The physical address to the cache prevents problems that could arise due to virtual aliasing. A physically addressed cache also makes cache coherency protocols easier to implement.

8.0 Cache Coherency Capability

The R4400MC 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.

Figure 6 shows the R4400MC issuing processor coherency requests and accepting external coherency requests.

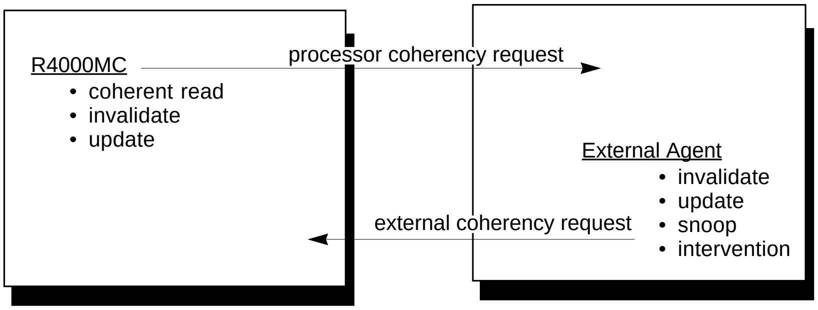


Figure 6 Coherency requests: Processor and External

The R4400MC processor issues the following processor coherency requests:

- processor coherent read requests
- processor invalidate requests
- processor update requests

The R4000MC processor accepts the following external coherency requests:

- external invalidate requests
- external update requests
- external snoop requests
- external intervention requests

8.1 Processor Synchronization

In a multiprocessor system, it is essential that two or more processors working on a common task execute without corrupting each other's subtasks. Synchronization, an operation that guarantees an orderly access to shared memory, must be implemented for a properly functioning multiprocessor system.

MIPS instructions *Load Linked* (LL) and *Store Conditional* (SC) provide support for processor synchronization. These two instructions work very much like their simpler counterparts, load and store. The LL instruction, in addition to doing a simple load, has the side effect of setting a bit called the *link bit*. This link bit forms a breakable link between the LL instruction and the subsequent SC instruction. The SC performs a simple store if the link bit is set when the store executes. If the link bit is not set, then the store fails to execute. The success or failure of the SC is indicated in the target register of the store.

The link is broken in the following circumstances:[1]

- if an external request (Invalidate, Snoop, or Intervention) changes the state of the line containing the lock variable
- an external update to the cache line containing the lock-variable
- upon completion of an ERET (return from exception) instruction.

The most important features of LL and SC are:

- They provide a mechanism for generating all of the common synchronization primitives including test-and-set, counters, sequencers, etc., with no additional overhead.
- When they operate, bus traffic is generated only if the state of the cache line changes; lock words stay in the cache until some other processor takes ownership of that cache line.

9.0 System Control Coprocessor

The R4400 incorporates a system control coprocessor unit on chip. The system control coprocessor, also known as CP0, is responsible for translating virtual addresses to physical addresses, exception handling, as well as some diagnostic capability. CP0 contains a 48-entry translation lookaside buffer (TLB) as well as several registers that are identified by unique

[1] The most obvious case where the link is broken occurs when an invalidate to the cache line is the subject of the load. In this case, some other processor has successfully completed a store to that line.

register numbers. Figure 7 shows the system control coprocessor CP0 registers and the 48 entry TLB..

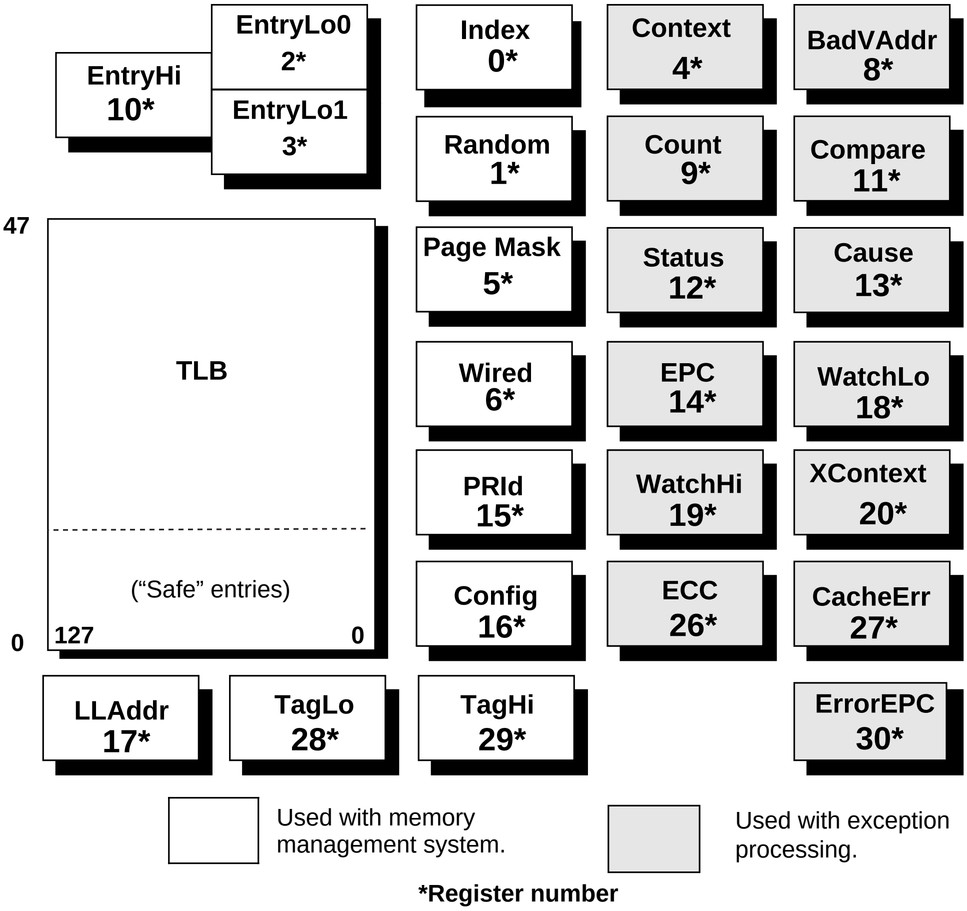


Figure 7 CP0 Registers and 48-entry TLB

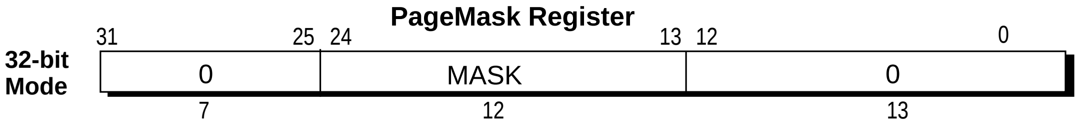
9.1 Memory Management Unit

The R4400 processor provides a full-featured memory management unit that uses an on-chip Translation Look-Aside Buffer (TLB) to translate virtual addresses into physical addresses.

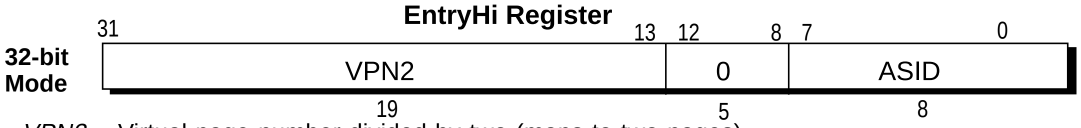
The MMU provides very fast virtual memory translation. The virtual memory extends the address space available to programs by translating programs in a large virtual space into physical memory space.

The R4400 has an extended 36-bit address bus providing it with a 64 GB physical address space. The virtual address is either 32-bits or 64 bits wide making the R4400 a true 64-bit processor. In 32-bit address mode, the user virtual address space is 2 GB while in 64-bit mode the virtual address space is 1 TeraByte.

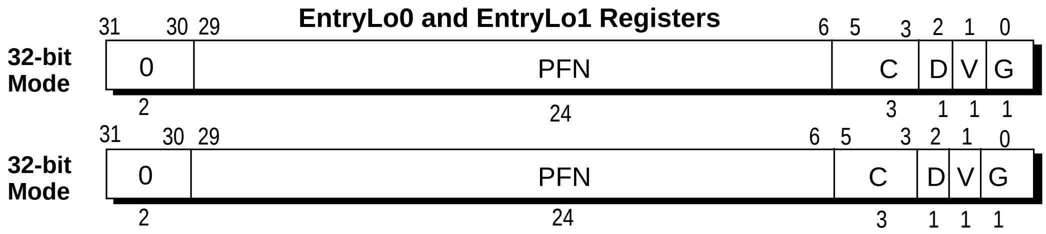
The memory management unit supports page sizes in the address space to vary from 4 KB to 16 MB. The MMU consists of 48 entries which provide mapping to 48 odd/even page pairs (96 pages). The format of the entry and the TLB (in 32-bit mode) is shown in Figure 8. There is an 8-bit address space (ASID) field that allows the R4400 to execute several processes simultaneously without the need to flush the TLB. In addition, several bits control other attributes like noncacheable pages, page sizes, dirty, and valid entries. For 64-bit mode operations please refer to the R4000 User's Manual.



Mask .... Page comparison mask.
0........... Reserved. Must be written as zeroes, and returns zeroes when read.



VPN2.... Virtual page number divided by two (maps to two pages).
ASID..... Address space ID field. An 8-bit field that lets multiple processes share the TLB; each process has a distinct mapping of otherwise identical virtual page numbers.
0........... Reserved. Must be written as zeroes, and returns zeroes when read.



PFN...... Page frame number; the upper bits of the physical address.
C .... Specifies the cache coherency attribute for the mapped page.
D .... Dirty. If this bit is set, the page is marked as dirty and, therefore, writable. This bit is actually a write-protect bit that software can use to prevent alteration of data.
V........... Valid. If this bit is set, it indicates that the TLB entry is valid; otherwise, a TLB miss occurs.
G .... Global. If this bit is set in both EntryLo0 and EntryLo1, then the processor ignores the ASID during TLB lookup.
0........... Reserved. Must be written as zeroes, and returns zeroes when read.

Figure 8 Fields of an R4400 TLB Entry

9.2 Operating Modes

The R4400 has three operating modes:

- User mode
- Supervisor mode
- Kernel mode

These modes are available to system software to provide a secure environment for user processes. The three address spaces corresponding to the operating modes are shown in Figure 9.

In the user mode, the R4400 provides a single, uniform virtual address space of 2 GB. When operating in the supervisor mode, the R4400 provides a virtual address space of 2.5 GB, divided into two regions based on the high-order bits of the virtual address. In the kernel mode, the R4400 provides a virtual address space of 4 GB. Figure 7 shows the three modes along with the accessible regions in each mode. Any attempt to access an address space that is not permitted results in an address error.

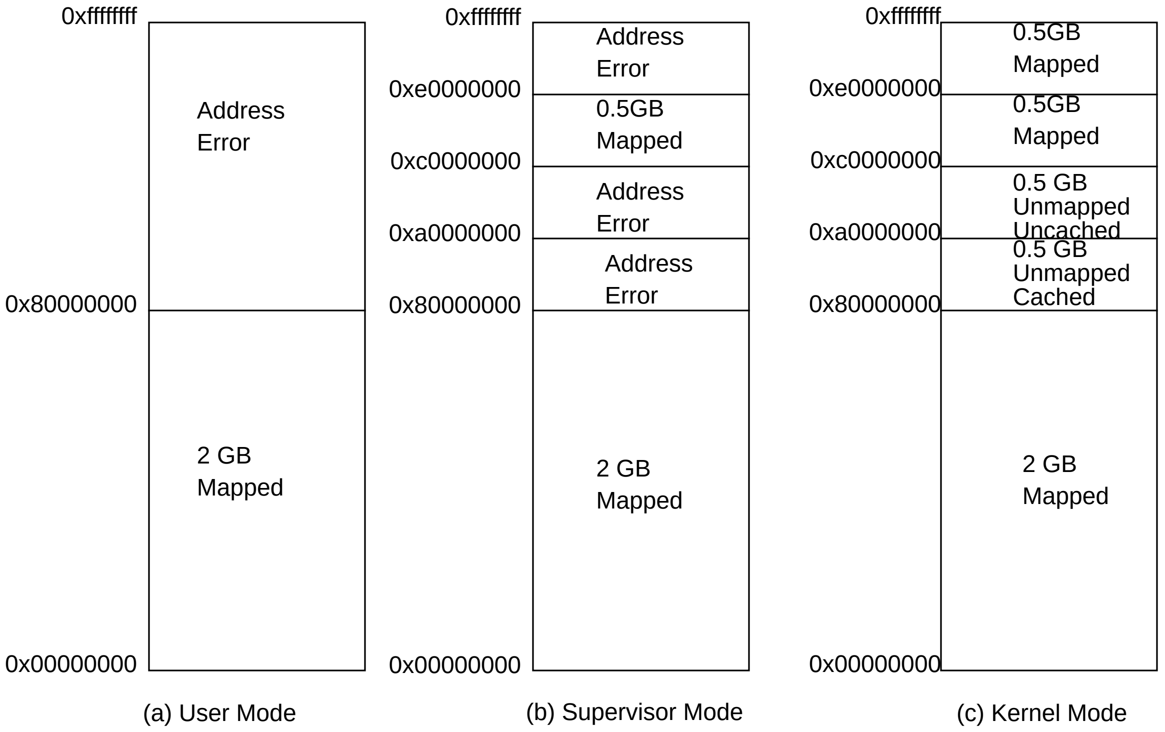


Figure 9 Address mapping (32-bit) in the R4400

10.0 Floating Point Unit

The floating point unit of the R4400 is fully compliant to the requirements of the ANSI/IEEE-Standard 754-1985, *IEEE Standard for Binary Floating point Arithmetic*.

The FPU operates as a coprocessor for the CPU (it is assigned coprocessor label *CP1*), and extends the CPU instruction set to perform arithmetic operations on floating-point values.

Figure 10 illustrates the functional organization of the FPU.

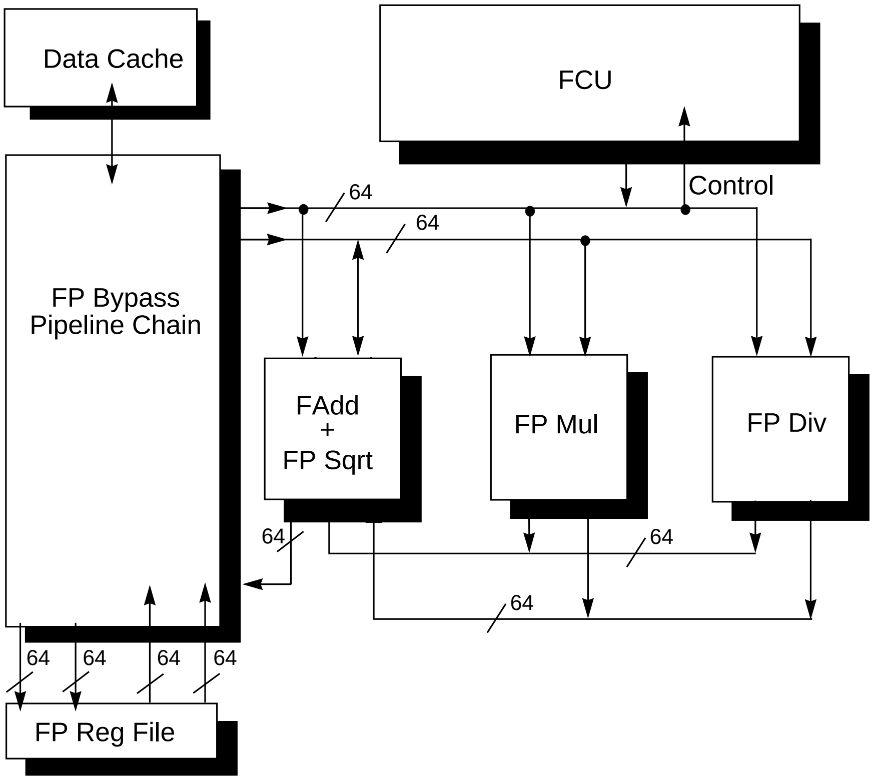


Figure 10 FPU Functional Block Diagram

The FPU has 32 registers which can be accessed as thirty-two 32-bit registers or thirty-two 64-bit registers. There are separate add, multiply, and divide units. This allows multiply, divide, and add operations to take place in parallel. However, the multiply and divide units use the adder unit to complete the end of an operation and this places restrictions on issuance of instruction in those cycles. The floating point units also permit load, store, and move operations to proceed in parallel because it has these arithmetic units separate from the integer unit.

Floating point operations can also be pipelined. The multiplier can begin a new double-precision multiplication every four cycles, and a new single-precision multiplication every three cycles. The adder generally begins a new operation one cycle before the previous cycle completes; therefore, a floating-point addition or subtraction can start every three cycles.

The FPU coprocessor pipeline is fully bypassed and interlocked.

11.0 Interrupts

The R4400PC has five maskable interrupts and one non-maskable interrupt. In addition, the R4400 supports two software interrupts. There are dedicated hardware pins in the R4400PC to signal hardware interrupts to the CPU. The R4400SC has only one dedicated hardware interrupt pin and a non-maskable interrupt pin.

In addition, the hardware interrupts are accessible by external write requests to the processor. An external write to the R4400 with SysAD[6:4] = 0 writes to an architecturally transparent register called the Interrupt Register. During the data cycle, SysAD[22:16] are the write enables for the six hardware interrupts and SysAD[6:0] are the values to be written to the interrupt bits. This allows any subset of the interrupt register to set and clear with a single write request.

12.0 Clocks

The R4400 bases all its internal and external clock on MasterClock which is an input signal to the R4400. All the R4400 clocks are described below.

MasterClock

The processor bases all internal and external clocking on the single **MasterClock** input signal. The processor generates the clock output signal, **MasterOut**, at the same frequency as **MasterClock** and aligns **MasterOut** with **MasterClock**, if **SyncIn** is connected to **SyncOut**.

MasterOut

The processor generates the clock output signal, **MasterOut**, at the same frequency as **MasterClock** and aligns **MasterOut** with **MasterClock**, if **SyncIn** is connected to **SyncOut**. **MasterOut** clocks external logic, such as the reset logic.

SyncIn/SyncOut

The processor generates **SyncOut** at the same frequency as **MasterClock** and aligns **SyncIn** with **MasterClock**.

SyncOut must be connected to **SyncIn** either directly, or through an external buffer. The processor can compensate for both output driver and input buffer delays (and, when necessary, delay caused by an external buffer or PCB traces) when aligning **SyncIn** with **MasterClock**. Figure 10-7 gives an illustration of **SyncOut** connected to **SyncIn** through an external buffer.

PClock

The processor generates an internal clock, **PClock**, at twice the frequency of **MasterClock** and precisely aligns every other rising edge of **PClock** with the rising edge of **MasterClock**.

All internal registers and latches use **PClock**. Input and output registers going to the pins use SClock.

SClock

The R4400 processor divides **PClock** by 2, 3, or 4 (as programmed at boot-mode initialization) to generate the internal clock signal, **SClock**. The R4400 processor divides **PClock** by 2, 3, 4, 6 or 8 (as programmed at boot-mode initialization) to generate **SClock**. The processor uses **SClock** to sample data at the system interface and to clock data into the processor system interface output registers.

The first rising edge of **SClock**, after **ColdReset*** is deasserted, is aligned with the first rising edge of **MasterClock**.

TClock

TClock (transmit clock) clocks the output registers of an external agent, and can be a global system clock for any other logic in the external agent.

TClock is identical to **SClock**. The edges of **TClock** align precisely with the edges of **SClock** and **TClock** can also be aligned with **MasterClock**, when **SyncIn** is connected to **SyncOut**.

RClock

The external agent uses **RClock** (receive clock) to clock its input registers. The processor generates **RClock** at the same frequency as **SClock**, although **RClock** leads **TClock** and **SClock** by 25 percent of **SClock** cycle time.

Figure 11 shows the clocks for a **PClock**-to-**SClock** division by 2.

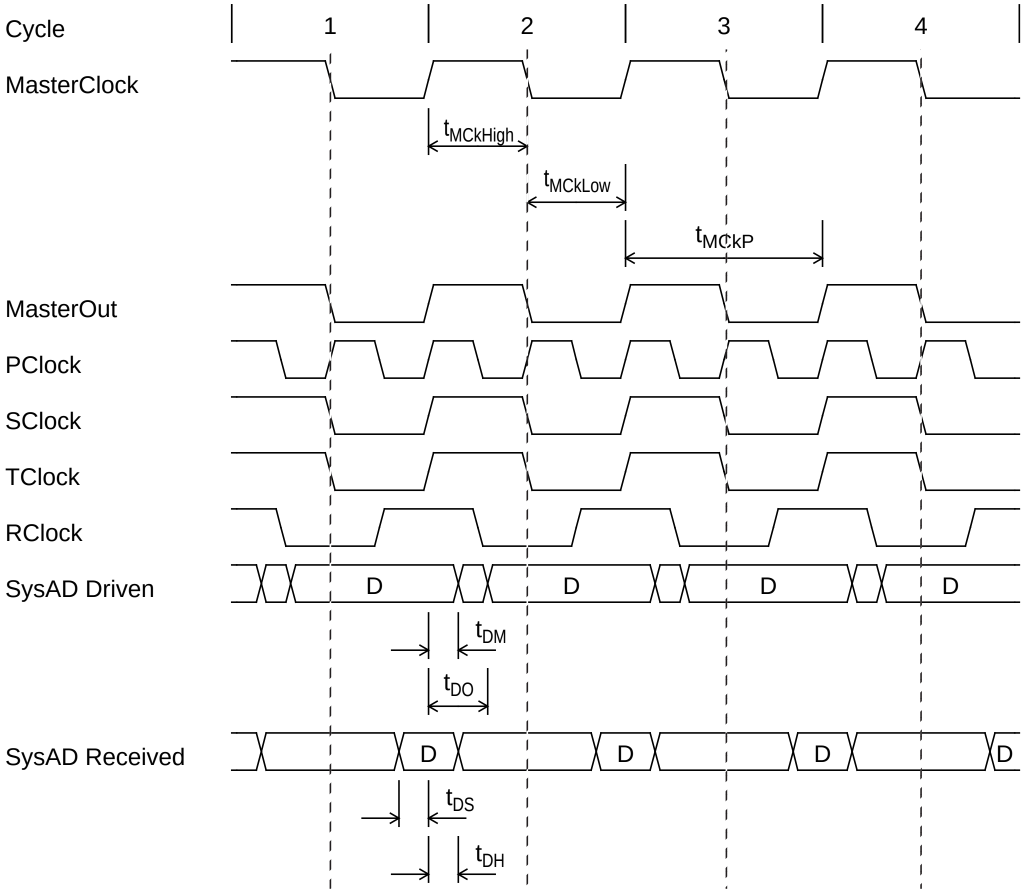


Figure 11 R4400 Clocks

13.0 System Interface

The R4400 processor supports a 64-bit address/data interface that can construct systems ranging from a simple uniprocessor with main memory to a multiprocessor system with caches and complete cache coherency. The System interface consists of:

- 64-bit address and data bus, **SysAD**
- 8-bit SysAD check bus, **SysADC**
- 9-bit command bus, **SysCmd**
- eight handshake signals:

- RdRdy *, WrRdy *
- ExtRqst *, Release *
- ValidIn *, ValidOut *
- IvdAck *, IvdErr *

The processor uses the System interface to access external resources such as memory and I/O when there are cache misses and uncached operations.

A request through the system interface consists of the following:

- an address
- a system command that identifies the precise nature of the request
- a series of data if the request os for a write, read response, or update

13.1 SysAD Bus

The SysAD bus is a a 64-bit, bi-directional, multiplexed address/data bus that is used primarily for transferring address and data between the R4400 and the rest of the system.

The system interface is configurable to allow interfacing to memory and I/O that may be of varying frequencies in order to alleviate system design concerns. The system frequency is programmable during the reset sequence via the boot mode control bits.

13.2 SysCmd Bus

The system command bus indicates whether the system interfaces has an address or data with one bit. During an address cycle, the SysCmd bus further indicates the type of transaction such as a read or a write transaction. During data cycles, the SysCmd bus indicates whether the data is the last data word of the burst transfer. The SysCmd bus is bi-directional to support both R4400 requests as well as external requests to the R4400.

13.3 Handshake Signals

Various handshaking signals are present on the R4400. RdRdy* is used to indicate whether the external system can accept a new read transaction. WrRdy* is used to indicate whether the external system is ready to accept a write transaction from the R4400.

ExtRqst* and Release* are used by the R4400 to transfer control of the processor to an external agent. When Release* is asserted by the R4400, the SysAD bus as well as the SysCmd bus may be driven by an external agent. Release* is also used to inform an external agent that the R4400 is ready to accept a read response during read request cycles.

The R4400 processor uses **ValidOut*** and the external agent uses **ValidIn*** to indicate valid command/data on the **SysCmd**/**SysAD** buses.

13.4 R4400PC Protocol

In an R4400PC, requests are always issued in a sequential fashion. Only one request can be pending at a time. For example, if the R4400 issues a read request, the processor waits for the read response before any subsequent request is issued. Write request are similarly issued only if there are no pending reads.

13.5 Read Cycle

A processor may issue a read request due to the following reasons:

- Cache miss
- Uncached read
- Write miss

When a processor issues a read request, the external agent (a memory controller, for example) must respond by providing the requested data to the R4400. The processor may need to arbitrate for the bus and assert Release* before the external agent can return the requested data. A processor read request is completed after the last word of the requested data is returned to the processor.

During the first cycle of a processor read, the R4400 issues the address on the SysAD bus and the read command on the SysCmd bus. ValidOut* is asserted at the same time signalling that the address and command are valid. In the next cycle, the processor bus tristates so that the external system can start driving the multiplexed SysAD bus with the requested data. This is signalled with the assertion of Release*. Figure 12 illustrates a processor read request.

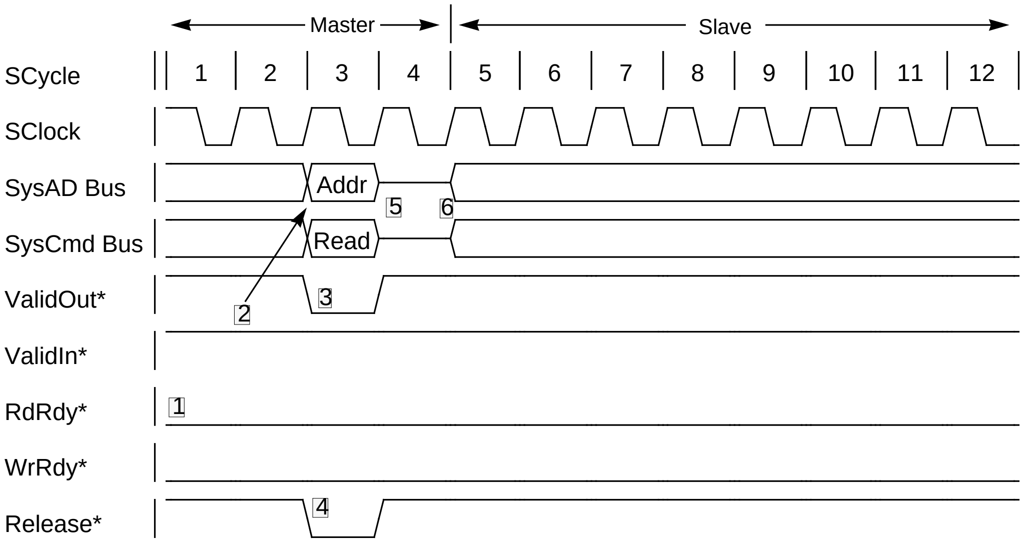


Figure 12 Processor Read Request Protocol

13.6 Write Cycle

When a processor issues a write request, it sends the address where the data needs to be written in the first cycle on the SysAD bus. WrRdy* must be asserted low at least two cycles for the processor to send the write request.

The R4400 can issue write requests based on two protocols:

- Doubleword, partial doubleword, word, or partial word writes use a word[1] write request protocol.
- Block writes use a block write request protocol.

Processor doubleword write requests are issued with the System interface in master state, as described below in the steps below; Figure 13 shows a processor noncoherent single word write request cycle.

1. A processor single word write request is issued by driving a write command on the **SysCmd** bus and a write address on the **SysAD** bus.
2. The processor asserts **ValidOut***.
3. The processor drives a data identifier on the **SysCmd** bus and data on the **SysAD** bus.

The data identifier associated with the data cycle must contain a last data cycle indication. At the end of the cycle, **ValidOut*** is deasserted.

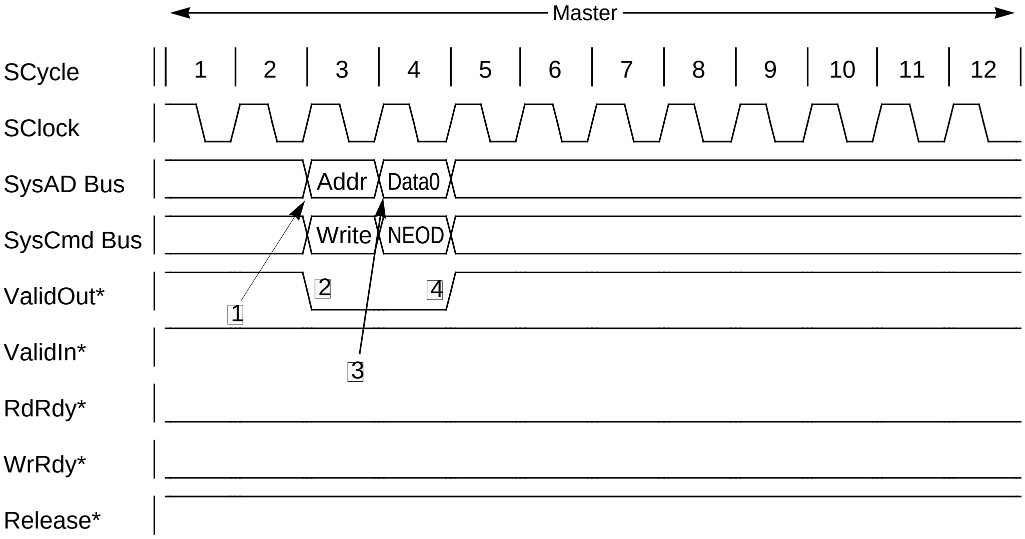


Figure 13 Processor Write Request

13.7 External Requests

External requests can only be issued with the System interface in slave state. An external agent asserts **ExtRqst*** to arbitrate (see External Arbitration Protocol, below) for the System interface, then waits for the processor to release the System interface to slave state by asserting **Release*** before the external agent issues an external request. If the System interface is already in slave state—that is, the processor has previously performed an uncompelled change to slave state—the external agent can begin an external request immediately.

After issuing an external request, the external agent must return the System interface to master state. If the external agent does not have any additional external requests to perform, **ExtRqst*** must be deasserted two cycles after the cycle in which **Release*** was asserted. For a string of external requests, the **ExtRqst*** signal is asserted until the last request cycle, whereupon it is deasserted two cycles after the cycle in which **Release*** was asserted.

The R4400PC supports the following external request protocol

[1] Called *word* to distinguish it from *block* request protocol. Data transferred can actually be doubleword, partial doubleword, word, or partial word.

- read - request asks for a word of data from the processor from the processor’s internal resource.
- write - request provides a word of data to the processor’ internal resource
- null - request requires no action by the processor. It provides a mechanism for the external agent to return control of the System Interface to the master state without affecting the processor.

14.0 JTAG

The JTAG boundary scan mechanism allows testing of the connections between the processor, the printed circuit board to which it is attached, and the other components on the circuit board. The JTAG mechanism does not provide any capability for testing the R4400 processor itself.

The R4400 implements the 16-state TAP controller as defined in the IEEE JTAG mechanism. The R4400 also contains a JTAG instruction register, a boundary-scan register, and a bypass register. The processor executes the standard JTAG EXTEST operation associated with EXternal Test functionality testing.

15.0 Initialization

Fundamental operational modes for the processor are initialized via the boot time mode control interface. The boot time mode control interface is a serial interface operating at a very low frequency (1/256 of the MasterClock) to allow the initialization information to be kept in a low cost EPROM.

Immediately after the VCCOk signal is asserted, the processor will read a serial bit stream of 256 bits to initialize all fundamental operational modes. After initialization is complete, the processor will continue to drive the serial clock output but no further initialization bits will be read.

The correspondence between bits of the initialization bit stream and processor mode settings is illustrated in Table 2: . Bit 0 of the bit stream is the bit presented to the processor when VCCOk is de-asserted.

Please note the following:

1. Selecting a reserved value results in undefined processor behavior.
2. Zero’s must be scanned in for all reserved bits.

Table 2: Boot Time Modes

Bit	Name	Mode Setting
0	**BlkOrder**	Secondary Cache Mode Block read response ordering. 0 Sequential ordering. 1 Sub-block ordering.
1	**EIBParMode**	Specifies nature of system interface check bus. 0 SECDED error checking and correcting mode. 1 Byte parity.
2	**EndBlt**	Specifies byte ordering. 0 Little Endian ordering. 1 Big Endian ordering.

Table 2: Boot Time Modes

Bit	Name	Mode Setting
3	**DShMdDis**	Dirty shared mode, enables transition to dirty shared state on processor update successful. 0 Dirty shared mode enabled. 1 Dirty shared mode disabled.
4	**NoSCMode**	Specifies presence of secondary cache. 0 R4400SC & R4400MC. 1 R4400PC.
5:6	**SysPort**	System Interface port width. 0 64 bits. 1-3 Reserved.
7	**SC64BitMd**	Secondary cache interface port width. 0 128 bits. 1 Reserved.
8	**ElSpltMd**	Specifies secondary cache organization. 0 Secondary cache unified. 1 Secondary cache split instruction vs. data.
9:10	**SCBlkSz**	Secondary cache line size, bit 10 is the most significant. 0 4 words. 1 8 words. 2 16 words. 3 32 words.
11:14	**XmitDatPat**	System interface data rate, bit 14 is the most significant. 0 D 1 DDx 2 DDxx 3 DxDx 4 DDxxx 5 DDxxxx 6 DxxDxx 7 DDxxxxxx 8 DxxxDxxx 9-15 Reserved.
15:17	**SysCkRatio**	PClock to SClock divisor, frequency relationship between SClock, RClock, and TClock and PClock, bit 17 is the most significant. 0 Divide by 2. 1 Divide by 3. 2 Divide by 4. 3 Divide by 6. 4 Divide by 8. 5-7 Reserved.
18	**Reserved**	Must be set to 0.

Table 2: Boot Time Modes

Bit	Name	Mode Setting
19	**TimIntDis**	Timer Interrupt disables timer interrupts, and the interrupt used by the timer becomes a general-purpose interrupt. 0 Timer Interrupt enabled. 1 Timer Interrupt disabled.
20	**PotUpdDis**	Potential update enable allows potential update to be issued. Otherwise only compulsory updates are issued. 0 Potential updates enabled. 1 Potential updates disabled.
21:24	**TWrSUp**	Secondary cache write deassertion delay, TWrSup in PCycles, bit 24 is the most significant. 0-2 Undefined. 3-15 Number of PClock cycles; Min 3, Max 15.
25:26	**TWr2Dly**	Secondary cache write assertion delay 2, TWr2Dly in PCycles, bit 26 is the most significant. 0 Undefined. 1-3 Number of PClock cycles; Min 1, Max 3.
27:28	**TWr1Dly**	Secondary cache write assertion delay 1, TWr1Dly in PCycles, bit 28 is the most significant. 0 Undefined. 1-3 Number of PClock cycles; Min 1, Max 3.
29	**TWrRCk**	Secondary cache write recovery time, TWrRc in PCycles, either 0 or 1 cycle. 0 0 cycle 1 1 cycle
30:32	**TDis**	Secondary cache disable time, TDis in PCycles, bit 32 is the most significant. 0-1 Undefined. 2-7 Number of PClock cycles; Min 2, Max 7.
33:36	**TRd2Cyc**	Secondary cache read cycle time 2, TRdCyc2 in PCycles, bit 36 is the most significant. 0-2 Undefined. 3-15 Number of PClock cycles; Min 3, Max 15.
37:40	**TRd1Cyc**	Secondary cache read cycle time 1, TRdCyc1 in PCycles, bit 40 is the most significant. 0-3 Undefined. 4-15 Number of PClock cycles; Min 4, Max 15.
41:45	**Reserved**	Must be set to 0.

Table 2: Boot Time Modes

Bit	Name	Mode Setting
46	**Pkg179**	R4400 Package type. 0 R4400SC & R4400MC. 1 R4400PC.
47:49	**CycDivisor**	This mode determines the clock divisor for the reduced power mode. When the RP bit in the Status Register is set to one, the pipeline clock is divided by one of the following values. Bit 49 is most significant. 0 Divide by 2. 1 Divide by 4. 2 Divide by 8. 3 Divide by 16. 4-7 Reserved.
50:52	**Drv0_50** **Drv0_75** **Drv1_00**	Drive the outputs out in (M x MasterClock) period. Drv_50 (bit 50) is the least significant bit and Drv1_00 (bit 52) is the most significant bit. 52 51 50 Drive-off time 0 0 1 0.50 * MasterClock Period 0 1 0 0.75 * MasterClock Period 1 0 0 1.00 * MasterClock Period All other combinations are reserved.
53:56	**InitP**	Initial values for the state bits that determine the pull-down di/dt and switching speed of the output buffers. Bit 53 is the most significant. 0 Fastest pull-down rate 1-14 Intermediate pull-down rates. 15 Slowest pull-down rate.
57:60	**InitN**	Initial values for the state bits that determine the pull-up di/dt and switching speed of the output buffers. Bit 57 is the most significant. 0 Slowest pull-up rate 1-14 Intermediate pull-up rates. 15 Fastest pull-up rate.
61	**EnblDPLLR**	Enables the negative feedback loop that controls switching speed of the output buffers only during ColdReset. 0 Disable di/dt mechanism. 1 Enable di/dt mechanism.
62	**EnblDPLL**	Enables the negative feedback loop that determines the di/dt and switching speed of the output buffers during ColdReset and during normal operation. 0 Disable di/dt control mechanism. 1 Enable di/dt control mechanism.
63	**DsblPLL**	Enables PLLs that match MasterIn and produce RClock, TClock SClock and the internal clocks. 0 Enable PLLs. 1 Disable PLLs.

Table 2: Boot Time Modes

Bit	Name	Mode Setting
64	**SRTristate**	Controls when output-only pins are tristated 0 Only when **ColdReset*** is asserted. 1 When **Reset*** or **ColdReset*** are asserted
65:255	**Reserved**	Must be set to 0.

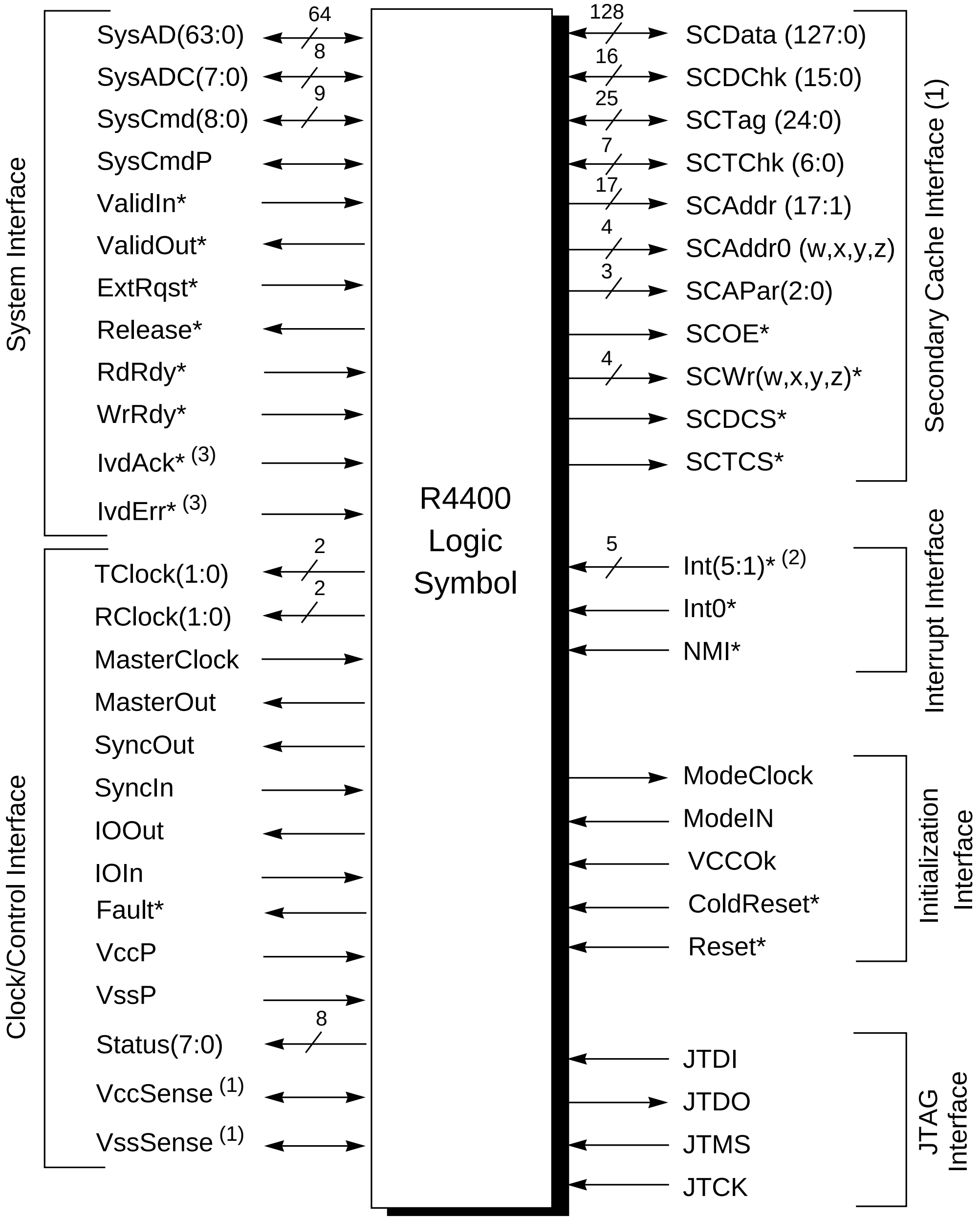
16.0 Pin Configuration

This section describes the signals used by and in conjunction with the R4400. The signals include the System interface, the Clock/Control interface, the Secondary Cache interface, the Interrupt interface, the Joint Test Action Group (JTAG) interface, and the Initialization interface.

Signals that are active low have a trailing asterisk—for instance, the low-active Read Ready signal is **RdRdy***. The signal description also tells if the signal is an input or output.

Table 3 gives a list of the signals and the packages that they are available in. All refers to R4400PC, R4400SC, and R4400MC packages.

Figure 14 illustrates the functional groupings of the processor signals.



(1) = R4400SC and R4000MC only (2) = R4400PC only
(3) = R4400MC only

Figure 14 R4000 Processor Signals

Table 3: Pin Description

Pin	Dir	Description	Pkgs
SysAD[63:0]	I/O	A 64-bit bus used for address and data transmission between the processor and an external agent.	All
SysADC[7:0]	I/O	An 8-bit bus containing check bits for the SysAD bus	All
SysCmd(8:0):	I/O	A 9-bit bus used for command and data identifier transmission between the processor and an external agent	All
SysCmdP	I/O	A single even parity bit over the SysCmd bus	All
ValidIn*	I	Signals that an external agent is driving a valid address or valid data on the SysAD bus and a valid command or data identifier on the SysCmd bus during this cycle	All
ValidOut*	O	Signals that the processor is driving a valid address or valid data on the SysAD bus and a valid command or data identifier on the SysCmd bus during this cycle.	All
ExtRqst*	O	Signals that the system interface needs to submit an external request.	All
Release*	O	Signals that the processor is releasing the system interface to slave state	All
RdRdy*	I	Signals that an external agent is capable of accepting a processor read, invalidate, or update request in both non-overlap and overlap mode or a read followed by a potential invalidate or update request in overlap mode	All
WrRdy*	I	Signals that an external agent is capable of accepting a processor write request in both non-overlap and overlap mode.	All
Int*(5:1)	I	Five of six general processor interrupts, bit-wise ORed with bits 5:1 of the interrupt register.	PC only
Int*(0)	I	One of six general processor interrupts, bit-wise ORed with bit 0 of the interrupt register	All
NMI*	I	Non-maskable interrupt, ORed with bit 6 of the interrupt register	All
ModeIn	I	Serial boot mode data in	All

Table 3: Pin Description

Pin	Dir	Description	Pkgs
ModeClock	O	Serial boot mode data clock out at the system clock frequency divided by 256	All
JTDI	I	JTAG Serial Data In	All
JTDO	O	JTAG Serial Data Out	All
JTMS	I	JTAG command signal, signals that the serial data in is command data	All
JTCK	O	JTAG serial clock input	All
TClock(1:0)	O	Two identical transmit clocks at the operation frequency of the system interface	All
RClock(1:0)	O	Two identical receive clocks at the operation frequency of the system interface	All
MasterClock	I	Master clock input at the operation frequency of the processor.	All
MasterOut	O	Master clock output aligned with MasterClock	All
SyncOut	O	Synchronization clock output	All
SyncIn	I	Synchronization clock input	All
IOOut	O	Output slew rate control feedback loop output. Must be connected to IOIn through a delay loop that models the IO path from the R4400 to an external agent.	All
IOIn	I	Output slew rate control feedback loop input	All
VCCOk	I	When asserted, this signal indicates to the R4400 that the +5 volt power supply has been above 4.75 volts for more than 100 milliseconds and will remain stable. The assertion of VCCOk will initiate the reading of the boot time mode control serial stream	All
ColdReset*	I	This signal must be asserted for a power on reset or a cold reset. The clocks SClock, TClock, and RClock begin to cycle and are synchronized with the de-assertion edge of ColdReset*. ColdReset* must be de-asserted synchronously with MasterClock	All

Table 3: Pin Description

Pin	Dir	Description	Pkgs
Reset*	I	This signal must be asserted for any reset sequence. It may be asserted synchronously or asynchronously for a cold reset, or synchronously to initiate a warm reset. Reset* must be de-asserted synchronously with MasterClock	All
GrpRun*	O	Run pulse generated after a group of instructions completes	All
GrpStall*	I	Stall signal which will stall the processor after the current group of instructions completes	All
Fault*	O	Mismatch output of boundary comparators	All
VccP	I	Quiet Vcc for the internal phase locked loop	All
VssP	I	Quiet Vss for the internal phase locked loop	All
SCData(127:0)	I/O	A 128-bit bus used to read or write cache data from/to the secondary cache	SC.MC
SCDChk(15:0)	I/O	A 16-bit bus which conveys two ECC fields that cover the upper or lower 64 bits of the SCData from/to the secondary cache	SC.MC
SCTag(24:0)	I/O	A 25-bit bus used to read or write cache tags from/to the secondary cache	SC.MC
SCTChk(6:0)	I/O	A 7-bit bus which conveys an ECC field that covers the **SCTag** from/to the secondary cache	SC.MC
SCAddr(17:1)	O	A 17-bit bus which addresses the secondary cache	SC.MC
SCAddr0Z	O	Bit 0 of the secondary cache address	SC.MC
SCAddr0Y	O	Bit 0 of the secondary cache address	SC.MC
SCAddr0X	O	Bit 0 of the secondary cache address	SC.MC
SCAddr0W	O	Bit 0 of the secondary cache address	SC.MC
SCAPar[2:0]	O	The secondary cache address even parity bus cover the following bits: SCAPar[2]7 bits:SCWrB, SCAddr[17:12] SCAPar[1]7 bits:SCDCSB, SCAddr[11:6] SCAPar[0]7 bits:SCTCSB, SCAddr[5:0]	SC.MC

Table 3: Pin Description

Pin	Dir	Description	Pkgs
SCOE*	O	A signal which enables the outputs of the secondary cache RAMs	SC.MC
SCWrZ*	O	Secondary cache write enable	SC.MC
SCWrY*	O	Secondary cache write enable	SC.MC
SCWrX*	O	Secondary cache write enable	SC.MC
SCWrW*	O	Secondary cache write enable	SC.MC
SCDCS*	O	A signal which enables the chip select pins of the secondary cache RAMs associated with SCData and SCDChk	SC.MC
SCTCS*	O	A signal which enables the chip select pins of the secondary cache RAMs associated with SCTag and SCTChk	SC.MC
IvdAck*	I	Signals that a processor invalidate or update request has completed successfully	SC.MC
IvdErr*	I	Signals that a processor invalidate or update request has completed unsuccessfully	SC.MC
Status(7:0)	O	An 8-bit bus that indicates the current operation status of the processor	SC.MC
VccSense	I/O	This is a special pin used only in component testing and characterization. It provides a separate, direct connection from the on-chip VCC node to a package pin without attaching to the in-package power planes. Test fixtures treat VccSense as an analog output pin; the voltage at this pin directly shows the behavior of the on-chip VCC. Thus, characterization engineers can easily observe the effects of di/dt noise, transmission line reflections, etc. VCCSense should be connected to VCC in functional system designs	SC.MC
VssSense	I/O	VssSense provides a separate, direct connection from the on-chip VSS node to a package pin without attaching to the in-package ground planes. VSSSense should be connected to VSS in functional system designs	SC.MC

17.0 Differences from the R4000

The R4400 implements the same MIPS III ISA as the R4000. Essentially, the R4400 is an upgrade to the R4000 and is designed to allow operating frequencies of up to 75 MHz. In addition, several additional features have been implemented in the R4400 to offer superior performance. The R4400 has these following enhancements to the R4000:

- fully functional Status pins
- Master/Checker mode
- larger primary caches
- uncached store buffer
- divide-by-6 and divide-by-8 modes
- cache error bit, *EW*, added to the *CacheErr* register.

18.0 References

Heinrich, Joe., “*MIPS R4000 User’s Manual*”, PTR Prentice Hall, 1993