



**Document Rev. 1.3**  
**Date: 02/2000**

- Dual Issue superscalar microprocessor
  - 200, 250, 266 MHz operating frequencies
  - 319 Dhrystone 2.1 MIPS
- High-performance system interface
  - 64-bit multiplexed system address/data bus for optimum price/performance
  - High-performance write protocols maximize uncached write bandwidth
  - Processor clock multipliers 2, 2.5, 3, 3.5, 4, 4.5, 5, 6, 7, 8, 9
  - IEEE 1149.1 JTAG boundary scan
- Integrated on-chip caches
  - 32KB instruction and 32KB data - 2 way set associative
  - Virtually indexed, physically tagged
  - Write-back and write-through on a per page basis
  - Pipeline restart on first doubleword for data cache misses
- Integrated memory management unit
  - Fully associative joint TLB (shared by I and D translations)
  - 48 dual entries map 96 pages
  - Variable page size (4KB to 16MB in 4x increments)

- High-performance floating-point unit:- up to 532 MFLOPS
  - Single cycle repeat rate for common single-precision operations and some double-precision operations
  - Two cycle repeat rate for double-precision multiply and double precision combined multiply-add operations
  - Single cycle repeat rate for single-precision combined multiply-add operation
- MIPS IV instruction set
  - Floating point multiply-add instruction increases performance in signal processing and graphics applications
  - Conditional moves to reduce branch frequency
  - Index address modes (register + register)
- Embedded application enhancements
  - Specialized DSP integer Multiply-Accumulate instructions and 3-operand multiply instruction
  - I and D cache locking by set
  - Optional dedicated exception vector for interrupts
- Fully static 0.25 micron CMOS design with power down logic
  - Standby reduced power mode with WAIT instruction
  - 2.5V core with 3.3V IO's
- 208-pin PQFP package

The diagram illustrates the RISC-V architecture, showing the interconnection of various components. Key components include:

- Primary Data Cache (2-way Set Associative)**: Connected to the **DTLB** and **ITag**.
- DTLB** (Data Tag) and **ITag** (Instruction Tag) are connected to the **ITLB** (Instruction Tag) and **ITLB** (Instruction Tag).
- Primary Instruction Cache (2-way Set Associative)**: Connected to the **ITag** and **ITLB**.
- Store Buffer**, **Write Buffer**, and **Read Buffer** are connected to the **Primary Data Cache**.
- Pad Buffer** and **Address Buffer** are connected to the **Primary Instruction Cache**.
- Instruction Dispatch Unit**: Contains the **FP Instruction Register** and **Integer Instruction Register**.
- FP Bus** (Floating-Point Bus) and **Integer Bus** connect the **Instruction Dispatch Unit** to the **Floating-Point Load/Align** and **Integer Register File**.
- D Bus** (Data Bus) connects the **Primary Data Cache** to the **Floating-Point Load/Align** and **Integer Register File**.
- Floating-Point Load/Align**, **Floating-Point Register File**, and **Packer/Unpacker** are part of the **Floating-Point Control** block.
- Floating-Point MultAdd, Add, Sub, Cvt, Div, Sqrt** are part of the **Floating-Point Control** block.
- Joint TLB**, **Coprocessor 0**, **System/Memory Control**, **PC Incrementer**, **Branch PC Adder**, **ITLB Virtual**, and **Program Counter** are part of the **Integer Control** block.
- Load Aligner**, **Integer Register File**, **Integer Address/Adder**, **Shifter/Store Aligner**, and **Logic Unit** are part of the **Integer Control** block.
- DTLB Virtual**, **PLL/Clocks**, and **Int Mult, Div, Madd** are part of the **Integer Control** block.

The diagram shows the flow of data and instructions between these components, highlighting the **A/D Bus** (Address/Data Bus), **Pad Bus**, **FP Bus**, and **Integer Bus**.

# HARDWARE OVERVIEW

The RM5261 offers a high-level of integration targeted at high-performance embedded applications. The key elements of the RM5261 are briefly described below.

## Superscalar Dispatch

The RM5261 has an asymmetric superscalar dispatch unit which allows it to issue an integer instruction and a floating-point computation instruction simultaneously. Integer instructions include alu, branch, load/store, and floating-point load/store, while floating-point computation instructions include floating-point add, subtract, combined multiply-add, converts, etc. In combination with its high-throughput fully pipelined floating-point execution unit, the superscalar capability of the RM5261 provides unparalleled price/performance in computationally intensive embedded applications.

## CPU Registers

The RM5261 CPU has a simple user-visible state consisting of 32 general purpose registers, two special purpose registers for integer multiplication and division, a program counter, and no condition code bits. Figure 1 shows the user visible state.

## Integer Unit

Like the RM5260, the RM5261 implements the MIPS IV Instruction Set Architecture, and is therefore fully upward

compatible with applications that run on processors implementing the earlier generation MIPS I-III instruction sets. Additionally, the RM5261 includes three implementation specific instructions not found in the baseline MIPS IV ISA but that are useful in the embedded market place. Described in detail in a later section, these instructions are integer multiply-accumulate and 3-operand integer multiply.

The RM5261 integer unit includes thirty-two general purpose 64-bit registers, a load/store architecture with single cycle ALU operations (add, sub, logical, shift) and an autonomous multiply/divide unit. Additional register resources include: the *HI/LO* result registers for the two-operand integer multiply/divide operations, and the program counter (*PC*).

## Pipeline

For integer operations, loads, stores, and other non-floating-point operations, the RM5261 implements a 5-stage integer pipeline. In addition to the integer pipeline, the RM5261 implements an extended 7-stage pipeline for floating-point operations.

The RM5261 multiplies the input **SysClock** by 2, 2.5, 3, 3.5, 4, 4.5, 5, 6, 7, 8, or 9 to produce the pipeline clock.

Figure 2 shows the RM5261 integer pipeline. As illustrated in the figure, up to five integer instructions can be executing simultaneously.

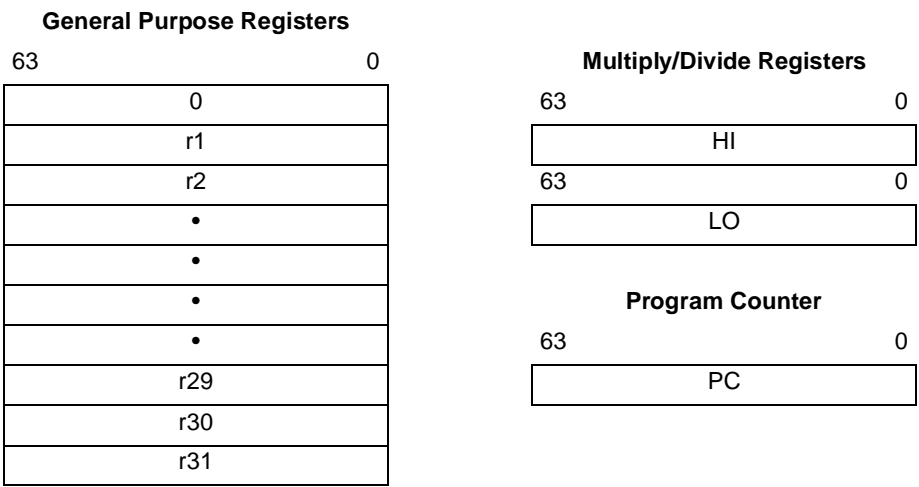
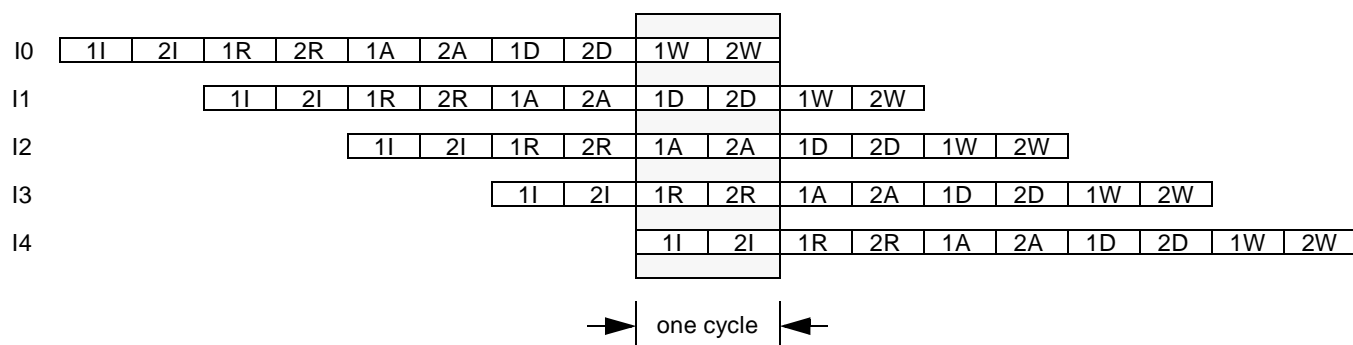


Figure 1 CPU Registers



**Figure 2 Pipeline**

## Register File

The RM5261 has thirty-two general purpose registers with register location 0 (*r0*) hard-wired to a zero value. These registers are used for scalar integer operations and address calculation. The register file has two read ports and one write port and is fully bypassed to minimize operation latency in the pipeline.

## ALU

The RM5261 ALU consists of an integer adder/subtractor, a logic unit, and a shifter. The adder performs address calculations in addition to arithmetic operations. The logic unit performs all logical and zero shift data moves. The shifter performs shifts and store alignment operations. Each of these units is optimized to perform all operations in a single processor cycle.

## Integer Multiply/Divide

The RM5261 has a dedicated integer multiply/divide unit optimized for high-speed multiply and multiply-accumulate operations. Table 1 shows the performance of the multiply/divide unit on each operation.

**Table 1: Integer Multiply/Divide Operations**

Opcode	Operand Size	Latency	Repeat Rate	Stall Cycles
MULT/U, MAD/U	16 bit	3	2	0
	32 bit	4	3	0
MUL	16 bit	3	2	1
	32 bit	4	3	2
DMULT, DMULTU	any	7	6	0
DIV, DIVD	any	36	36	0
DDIV, DDIVU	any	68	68	0

The baseline MIPS IV ISA specifies that the results of a multiply or divide operation be placed in the *Hi* and *Lo* registers. These values can then be transferred to the general purpose register file using the Move-from-*Hi* and Move-from-*Lo* (MFHI/MFLO) instructions.

In addition to the baseline MIPS IV integer multiply instructions, the RM5261 also implements the 3-operand multiply instruction, MUL. This instruction specifies that the multiply result go directly to the integer register file rather than the *Lo* register. The portion of the multiply that would have normally gone into the *Hi* register is discarded. For applications where it is known that the upper half of the multiply result is not required, using the MUL instruction eliminates the necessity of executing an explicit MFLO instruction.

Also included in the RM5261 are the multiply-add instructions, MADU/MAD. This instruction multiplies two operands and adds the resulting product to the current contents of

the *Hi* and *Lo* registers. The multiply-accumulate operation is the core primitive of almost all signal processing algorithms allowing the RM5261 to eliminate the need for a separate DSP engine in many embedded applications.

## Floating-Point Co-Processor

The RM5261 incorporates a high-performance fully pipelined floating-point co-processor which includes a floating-point register file and autonomous execution units for multiply/add/convert and divide/square root. The floating-point coprocessor is a tightly coupled execution unit, decoding and executing instructions in parallel with, and in the case of floating-point loads and stores, in cooperation with the integer unit. The superscalar capabilities of the RM5261 allow floating-point computation instructions to issue concurrently with integer instructions.

## Floating-Point Unit

The RM5261 floating-point execution unit supports single and double precision arithmetic, as specified in the IEEE Standard 754. The execution unit is broken into a separate divide/square root unit and a pipelined multiply/add unit. Overlap of the divide/square root and multiply/add operations is supported.

The RM5261 maintains fully precise floating-point exceptions while allowing both overlapped and pipelined operations. Precise exceptions are extremely important in object-oriented programming environments and highly desirable for debugging in any environment.

Floating-point operations include:

- add
- subtract
- multiply
- divide
- square root
- reciprocal
- reciprocal square root
- conditional moves
- conversion between fixed-point and floating-point format
- conversion between floating-point formats
- floating-point compare.

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

**Table 2: Floating-Point Instruction Cycles**

Operation	Latency	Repeat Rate
fadd	4	1
fsub	4	1
fmult	4/5	1/2
fmadd	4/5	1/2
fmsub	4/5	1/2
fdiv	21/36	19/34
fsqrt	21/36	19/34
frecip	21/36	19/34
frsqrt	38/68	36/66
fcvt.s.d	4	1
fcvt.s.w	6	3
fcvt.s.l	6	3
fcvt.d.s	4	1
fcvt.d.w	4	1
fcvt.d.l	4	1
fcvt.w.s	4	1
fcvt.w.d	4	1
fcvt.l.s	4	1
fcvt.l.d	4	1
fcmp	1	1
fmov	1	1
fmovc	1	1
fabs	1	1
fneg	1	1

Note: Numbers are represented as single/double precision format.

## Floating-Point General Register File

The floating-point general register file (FGR) is made up of thirty-two 64-bit registers. With the floating-point load and store double instructions (LDC1 and SDC1) the floating-point unit can take advantage of the 64-bit wide data cache and issue a floating-point co-processor load or store doubleword instruction in every cycle.

The floating-point control register space contains two registers; one for determining configuration and revision information for the coprocessor and one for control and status information. These are primarily used for diagnostic software, exception handling, state saving and restoring, and control of rounding modes. To support superscalar operation, the FGR has four read ports and two write ports, and is fully bypassed to minimize operation latency in the pipeline. Three of the read ports and one write port are used to support the combined multiply-add instruction while the fourth read and second write port allows a concurrent floating-point load or store.

# System Control Co-processor (CP0)

The system control coprocessor, also called coprocessor 0 or CP0 in the MIPS architecture, is responsible for the virtual memory sub-system, the exception control system, and the diagnostics capability of the processor. In the MIPS architecture, the system control co-processor (and thus the kernel software) is implementation dependent.

The memory management unit controls the virtual memory system page mapping. It consists of an instruction address translation buffer, ITLB, a data address translation buffer, DTLB, a Joint instruction and data address translation buffer, JTLB, and co-processor registers used by the virtual memory mapping sub-system.

## System Control Co-Processor Registers

The RM5261 incorporates all system control co-processor (CP0) registers on-chip. These registers provide the path through which the virtual memory system's page mapping is examined and modified, exceptions are handled, and operating modes are controlled (kernel vs. user mode, interrupts enabled or disabled, cache features). In addition, the RM5261 includes registers to implement a real-time cycle counting facility to aid in cache diagnostic testing and to assist in data error detection.

Figure 3 shows the CP0 registers.

# Virtual to Physical Address Mapping

The RM5261 provides three modes of virtual addressing:

- user mode
- kernel mode
- supervisor mode

This mechanism is available to system software to provide a secure environment for user processes. Bits in the CP0 register Status determine which virtual addressing mode is used. In the user mode, the RM5261 provides a single, uniform virtual address space of 1TB (2GB in 32-bit mode).

When operating in the kernel mode, four distinct virtual address spaces, totalling over 2.5TB (4GB in 32-bit mode), are simultaneously available and are differentiated by the high-order bits of the virtual address.

The RM5261 processors also support a supervisor mode in which the virtual address space over 2TB (2.5GB in 32-bit mode), divided into three regions based on the high-order bits of the virtual address.

When the RM5261 is configured as a 64-bit microprocessor, the virtual address space layout is an upward compatible extension of the 32-bit virtual address space layout.

Figure 4 shows the address space layout for 32-bit operation.

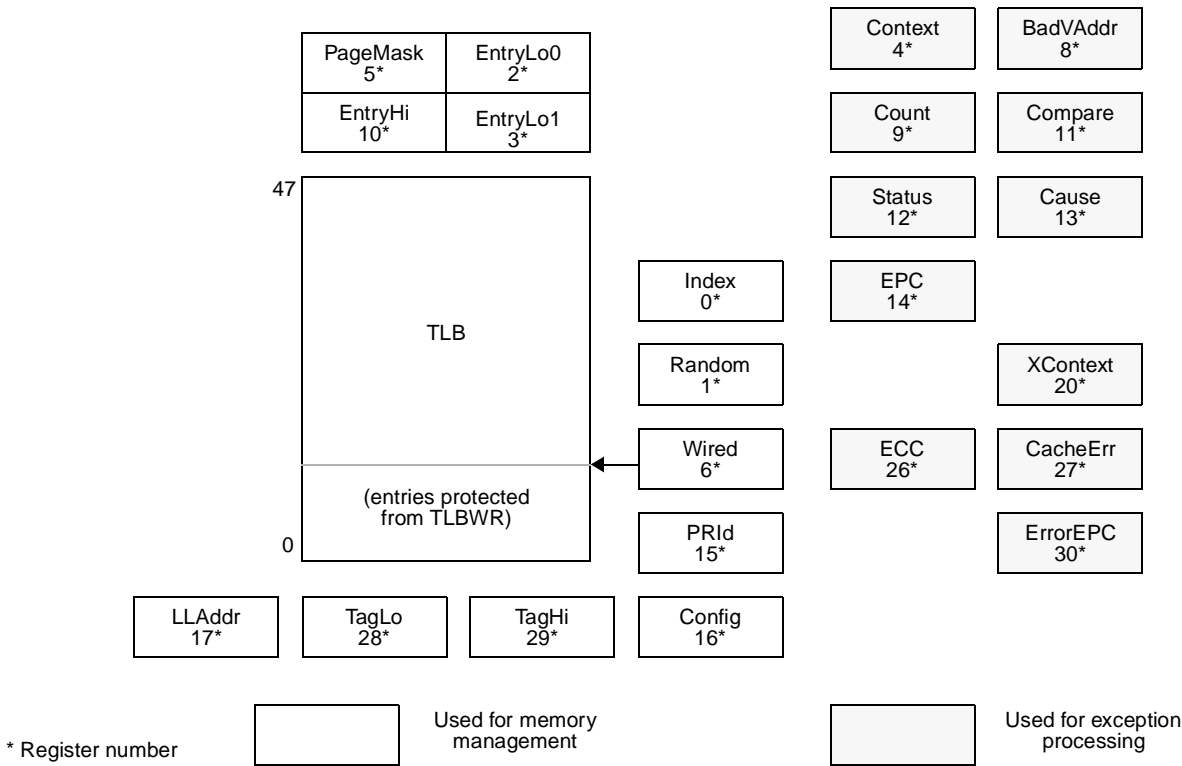


Figure 3 CP0 Registers

0xFFFFFFFF	Kernel virtual address space (kseg3) Mapped, 0.5GB
0xE0000000	
0xDFFFFFFF	Supervisor virtual address space (ksseg) Mapped, 0.5GB
0xC0000000	
0xBFFFFFFF	Uncached kernel physical address space (kseg1) Unmapped, 0.5GB
0xA0000000	
0x9FFFFFFF	Cached kernel physical address space (kseg0) Unmapped, 0.5GB
0x80000000	
0x7FFFFFFF	User virtual address space (kuseg) Mapped, 2.0GB
0x00000000	

**Figure 4 Kernel Mode Virtual Addressing (32-bit)**

## Joint TLB

For fast virtual-to-physical address translation, the RM5261 uses a large, fully associative TLB that maps 96 virtual pages to their corresponding physical addresses. As indicated by its name, the joint TLB (JTLB) is used for both instruction and data translations. The JTLB is organized as 48 pairs of even-odd entries, and maps a virtual address and address space identifier into the large, 64GB physical address space.

Two mechanisms are provided to assist in controlling the amount of mapped space and the replacement characteristics of various memory regions. First, the page size can be configured, on a per-entry basis, to use page sizes in the range of 4KB to 16MB (in multiples of 4). The CP0 *Page Mask* register is loaded with the desired page size of a mapping, and that size is stored into the TLB along with the virtual address when a new entry is written. Thus, operating systems can create special purpose maps; for example, an entire frame buffer can be memory mapped using only one TLB entry.

The second mechanism controls the replacement algorithm when a TLB miss occurs. The RM5261 provides a random replacement algorithm to select a TLB entry to be written with a new mapping; however, the processor also provides a mechanism whereby a system specific number of mappings can be locked into the TLB, thereby avoiding random replacement. This mechanism allows the operating system

to guarantee that certain pages are always mapped for performance reasons and for deadlock avoidance. This mechanism also facilitates the design of real-time systems by allowing deterministic access to critical software.

The JTLB also contains information that controls the cache coherency protocol for each page. Specifically, each page has attribute bits to determine whether the coherency algorithm is one of the following:

- uncached
- non-coherent write-back
- non-coherent write-through with write-allocate
- non-coherent write-through without write-allocate
- sharable
- exclusive
- update.

Note that both of the write-through protocols bypass the secondary cache since it does not support writes of less than a complete cache line.

The non-coherent protocols are used for both code and data on the RM5261, with data using write-back or write-through depending on the application. The write-through modes support the same efficient frame buffer handling as the R4600 and R4700.

The coherency attributes generate coherent transaction types on the system interface. However, in the RM5261 cache coherency is not supported. Hence the coherency attributes should never be used.

## Instruction TLB

The RM5261 implements a 2-entry instruction TLB (ITLB) to minimize contention for the JTLB, eliminate the timing critical path of translating through a large associative array, and save power. Each ITLB entry maps a 4KB page. The ITLB improves performance by allowing instruction address translation to occur in parallel with data address translation. When a miss occurs on an instruction address translation by the ITLB, the least-recently used ITLB entry is filled from the JTLB. The operation of the ITLB is completely transparent to the user.

## Data TLB

The RM5261 implements a 4-entry data TLB (DTLB) for the same reasons cited above for the ITLB. Each DTLB entry maps a 4KB page. The DTLB improves performance by allowing data address translation to occur in parallel with instruction address translation. When a miss occurs on a data address translation by the DTLB, the DTLB is filled from the JTLB. The DTLB refill is pseudo-LRU: the least recently used entry of the least recently used pair of entries is filled. The operation of the DTLB is completely transparent to the user.



## Cache Memory

The RM5261 incorporates on-chip instruction and data caches that can be accessed in a single processor cycle. Each cache has its own 64-bit data path and both caches can be accessed simultaneously. The cache subsystem provides the integer and floating-point units with an aggregate bandwidth of 3.2GB per second at an internal clock frequency of 200 MHz.

### Instruction Cache

The RM5261 incorporates a two-way set associative on-chip instruction cache. This virtually indexed, physically tagged cache is 32 KB in size and is protected with word parity.

Since the cache is virtually indexed, the virtual-to-physical address translation occurs in parallel with the cache access, further increasing performance by allowing these two operations to occur simultaneously. The cache tag contains a 24-bit physical address, a valid bit, and a single parity bit.

The instruction cache is 64-bits wide and can be accessed each processor cycle. Accessing 64 bits per cycle allows the instruction cache to supply two instructions per cycle to the superscalar dispatch unit. For typical code sequences where a floating-point load or store and a floating-point computation instruction are being issued together in a loop, the entire bandwidth available from the instruction cache will be consumed.

Cache miss refill writes 64 bits per cycle to minimize the cache miss penalty. The line size is eight instructions (32 bytes) to maximize the performance of communication between the processor and the memory system.

Like the R4650, the RM5261 supports cache locking. The contents of one set of the cache, set A, can be *locked* by setting a bit in the coprocessor 0 *Status* register. Locking the set prevents its contents from being overwritten by a subsequent cache miss. Refill will occur only into set B. This mechanism allows the programmer to lock critical code into the cache thereby guaranteeing deterministic behavior for the locked code sequence.

### Data Cache

For fast, single cycle data access, the RM5261 includes a 32 KB on-chip data cache that is two-way set associative with a fixed 32-byte (eight words) line size.

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.

Cache protocols supported for the data cache are:

1. **Uncached.** Reads to addresses in a memory area identified as uncached do not access the cache. Writes to such addresses are written directly to main memory without updating the cache.
2. **Write-back.** Loads and instruction fetches first search the cache, reading main memory only if the desired data is not cache resident. On data store operations, the cache is first searched to determine if the target address is cache resident. If it is resident, the cache contents are updated, and the cache line is marked for later write-back. If the cache lookup misses, the target line is first brought into the cache and then the write is performed as above.
3. **Write-through with write allocate.** Loads and instruction fetches first search the cache, reading main memory only if the desired data is not cache resident. On data store operations, the cache is first searched to determine if the target address is cache resident. If it is resident, the cache contents are updated and main memory is written, leaving the *write-back* bit of the cache line unchanged. If the cache lookup misses, the target line is first brought into the cache and then the write is performed as above.
4. **Write-through without write allocate.** Loads and instruction fetches first search the cache, reading main memory only if the desired data is not cache resident. On data store operations, the cache is first searched to determine if the target address is cache resident. If it is resident, the cache contents are updated and main memory is written, leaving the *write-back* bit of the cache line unchanged. If the cache lookup misses, then only main memory is written.

The most commonly used write policy is write-back, where a store to a cache line does not immediately cause memory to be updated. This 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. Software can, however, select write-through on a per-page basis when appropriate, such as for frame buffers.

Associated with the data cache is the store buffer. When the RM5261 executes a STORE instruction, this single-entry buffer gets written with the store data while the tag comparison is performed. If the tag matches, then the data is written into the data cache in the next cycle that the data cache is not accessed (the next non-load cycle). The store buffer allows the RM5261 to execute a store every processor cycle and to perform back-to-back stores without penalty. In the event of a store immediately followed by a load to the same address, a combined merge and cache write occurs such that no penalty is incurred. The RM5261 cache attributes for both the instruction and data caches are summarized in Table 3.

**Table 3: Cache Attributes**

Characteristics	Instruction	Data
Size	32KB	32KB
Organization	2-way set associative	2-way set associative
Line size	32B	32B
Index	vAddr <sub>11..0</sub>	vAddr <sub>11..0</sub>
Tag	pAddr <sub>31..12</sub>	pAddr <sub>31..12</sub>
Write policy	n.a.	write-back/write-through
Read order	sub-block	sub-block
Write order	sequential	sequential
miss restart after transfer of	entire line	first double
Parity	per-word	per-byte
Cache locking	set A	set A

## Write buffer

Writes to external memory, whether cache miss write-backs or stores to uncached or write-through addresses, use the on-chip write buffer. The write buffer holds up to four 64-bit address and data pairs. The entire buffer is used for a data cache write-back and allows the processor to proceed in parallel with the memory update. For uncached and write-through stores, the write buffer significantly increases performance by decoupling the **SysAD** bus transfers from the instruction execution stream.

## System Interface

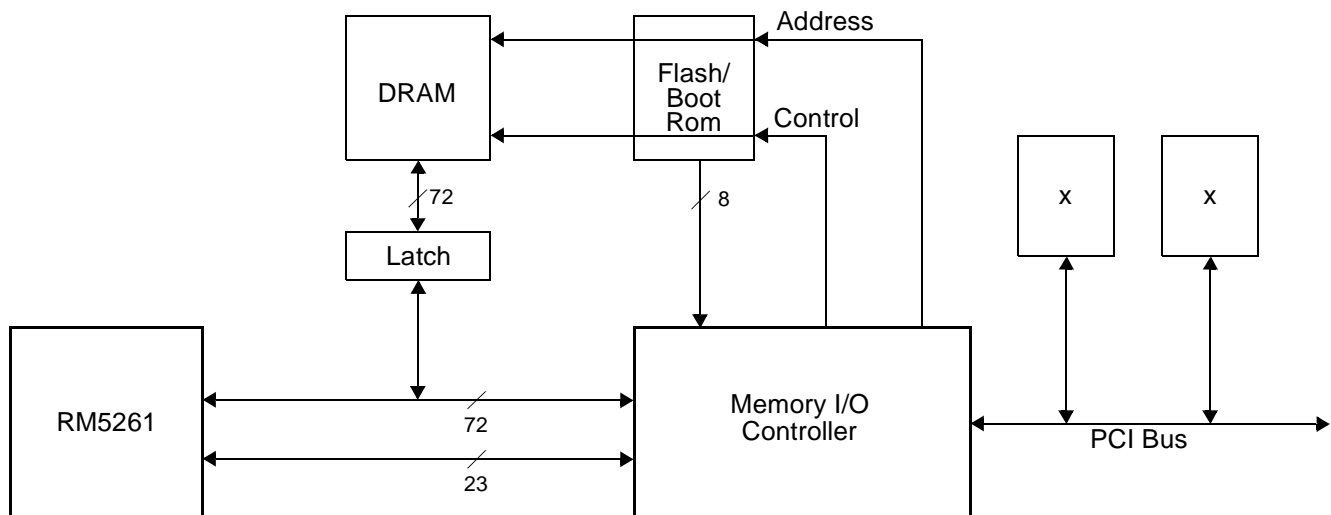
The system interface consists of a 64-bit Address/Data bus with 8 parity check bits and a 9-bit command bus. In addition, there are 6 handshake signals and 6 interrupt inputs. The interface is capable of transferring data between the processor and memory at a peak rate of 800MB/sec with a 100MHz SysClock.

Figure 5 shows a typical embedded system using the RM5261. In this example, a bank of DRAMs and a memory controller ASIC share the processor's **SysAD** bus while the memory controller provides separate ports to a boot ROM and an I/O system.

## System Address/Data Bus

The 64-bit System Address Data (**SysAD**) bus is used to transfer addresses and data between the RM5261 and the rest of the system. It is protected with an 8-bit parity check bus (**SysADC**). The system interface is configurable to allow easy interfacing to memory and I/O systems of varying frequencies.

The data rate and the bus frequency at which the RM5261 transmits data to the system interface are programmable at boot time via the mode control bits. The rate at which the processor receives data is fully controlled by the external device.



**Figure 5 Typical Embedded System Block Diagram**



# System Command Bus

The RM5261 interface has a 9-bit System Command (**SysCmd**) bus. The command bus indicates whether the **SysAD** bus carries address or data information on a per-clock basis. If the **SysAD** carries address, the **SysCmd** bus indicates what type of transaction is to take place (for example, a read or write). If the **SysAD** carries data, the **SysCmd** bus provides information about the data (for example, this is the last data word transmitted, or the data contains an error). The **SysCmd** bus is bidirectional to support both processor requests and external requests to the RM5261. Processor requests are initiated by the RM5261 and responded to by an external device. External requests are issued by an external device and require the RM5261 to respond.

The RM5261 supports one- to eight-byte transfers as well as block transfers on the **SysAD** bus. In the case of a sub-double word transfer, the three low-order address bits give the byte address of the transfer, and the **SysCmd** bus indicates the number of bytes being transferred.

## Handshake Signals

There are six handshake signals on the system interface. Two of these, **RdRdy\*** and **WrRdy\***, are used by an external device to indicate to the RM5261 whether it can accept a new read or write transaction. The RM5261 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 from the processor to an external device. When an external device needs to control the interface, it asserts **ExtRqst\***. The RM5261 responds by

asserting **Release\*** to release the system interface to slave state.

**ValidOut\*** and **ValidIn\*** are used by the RM5261 and the external device respectively to indicate that there is a valid command or data on the **SysAD** and **SysCmd** buses. The RM5261 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.

## Non-overlapping System Interface

The RM5261 requires a non-overlapping system interface, compatible with the R5000. 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 RM5261 issues another request. The RM5261 can issue read and write requests to an external device, whereas an external device can issue null and write requests to the RM5261.

For processor reads the RM5261 asserts **ValidOut\*** and simultaneously drives the address and read command on the **SysAD** and **SysCmd** buses respectively. If the system interface has **RdRdy\*** asserted, then the processor tristates its drivers and releases the system interface to the slave state by asserting **Release\***. The external device can then begin sending data to the RM5261.

Figure 6 shows a processor block read request and the external agent read response. The read latency is 4 cycles (**ValidOut\*** to **ValidIn\***), and the response data pattern is DDDD, indicating that data can be transferred on every clock with no wait states in-between. Figure 7 shows a processor block write using write response pattern DDDD, or code 0, of the boot-time mode select options.

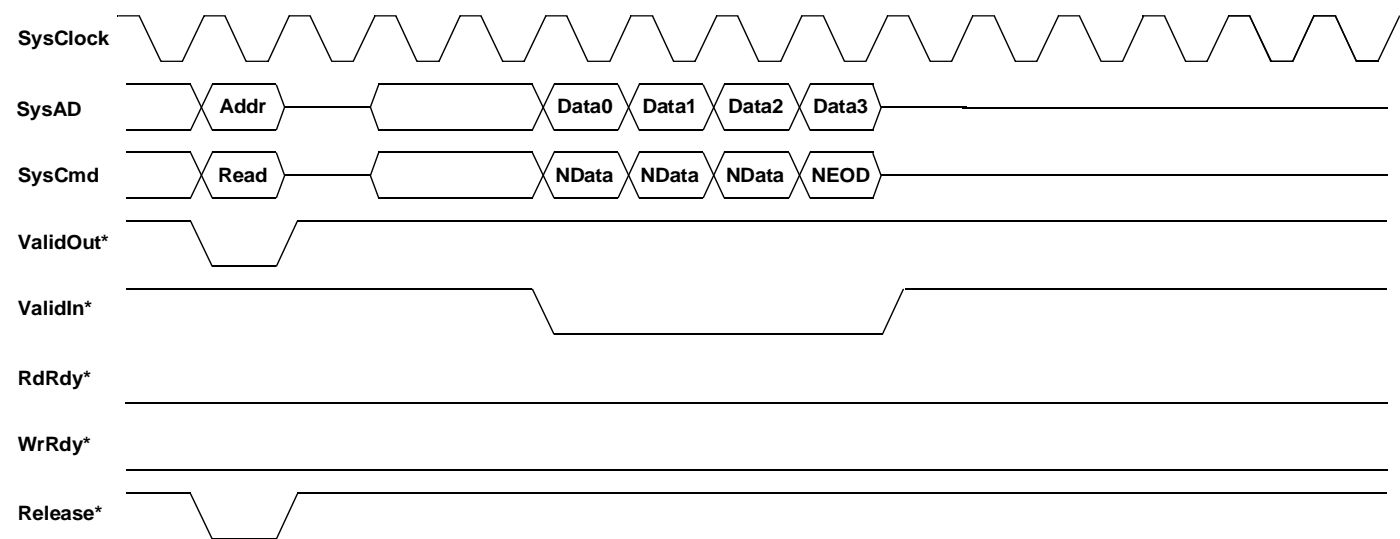


Figure 6 Processor Block Read

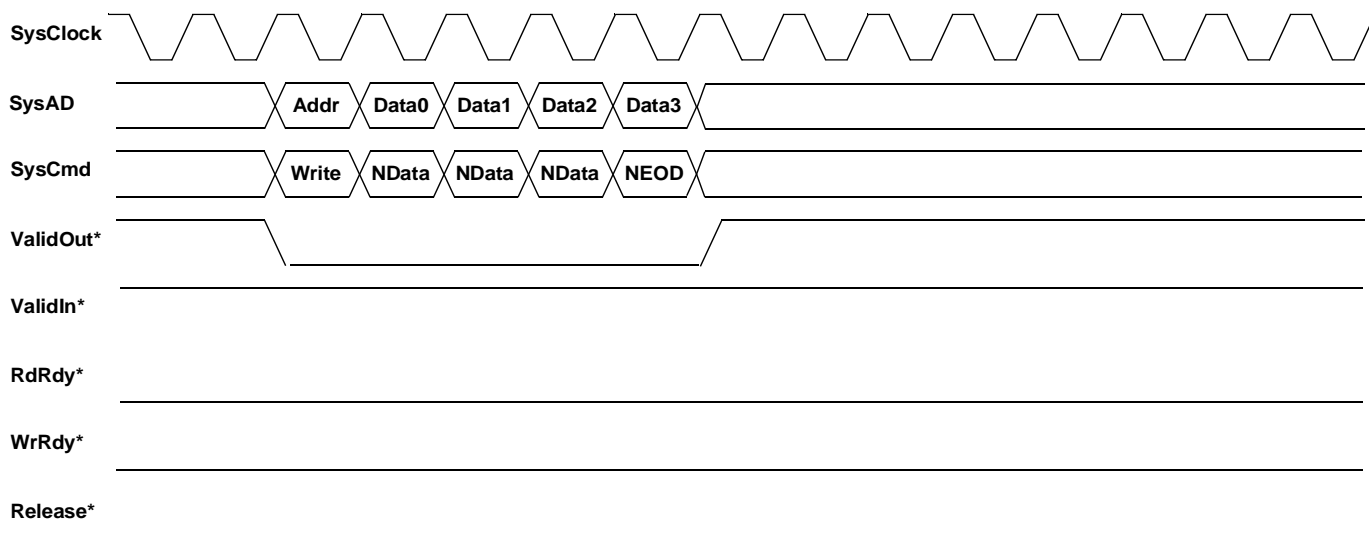


Figure 7 Processor Block Write

## Enhanced Write Modes

The RM5231 implements two enhancements to the original R4000 write mechanism: Write Reissue and Pipeline Writes. The original R4000 allowed a write on the SysAD bus every four SysClock cycles. Hence for a non-block write, this meant that two out of every four cycles were wait states.

Pipelined write mode eliminates these two wait states by allowing the processor to drive a new write address onto the bus immediately after the previous data cycle. This allows for higher SysAD bus utilization. However, at high frequencies the processor may drive a subsequent write onto the bus prior to the time the external agent deasserts **WrRdy\***, indicating that it can not accept another write cycle. This can cause the cycle to be aborted.

Write reissue mode is an enhancement to pipelined write mode and allows the processor to reissue aborted write cycles. If **WrRdy\*** is deasserted during the issue phase of a write operation, the cycle is aborted by the processor and reissued at a later time.

In write reissue mode, a write rate of one write every two bus cycles can be achieved. Pipelined writes have the same two bus cycle write repeat rate, but can issue one additional write following the deassertion of **WrRdy\***.

## External Requests

The RM5261 can respond to certain requests issued by an external device. These requests take one of two forms: *Write* requests and *Null* requests. An external device executes a write request when it wishes to update one of the processors writable resources such as the internal interrupt register. A null request is executed when the external

device wishes the processor to reassert ownership of the processor external interface (the external device wants the processor interface to go from slave state to master state). Typically, a null request is executed after an external device that has acquired control of the processor interface via the assertion of **ExtRqst\*** has completed a transaction between itself and system memory in a system where memory is connected directly to the **SysAD** bus. Normally, this transaction would be a DMA read or write from the I/O system.

## Interrupt Handling

The RM5261 supports a dedicated interrupt vector. When enabled by the real time executive (by setting a bit in the Cause register), interrupts vector to a specific address that is not shared with any of the other exception types. This capability eliminates the need to go through the normal software routine for exception decode and dispatch, thereby lowering interrupt latency.

## Standby Mode

The RM5261 provides a means to reduce the amount of power consumed by the internal core when the CPU would otherwise not be performing any useful operations. This state is known as Standby Mode.

Executing the WAIT instruction enables interrupts and causes the processor to enter Standby Mode. When the wait instruction completes the W pipe stage, and if the **SysAD** bus is currently idle, the internal processor clock stops, thereby freezing the pipeline. The phase lock loop, or PLL, internal timer/counter, and the “wake up” input pins: **Int[5:0]\***, **NMI\***, **ExtReq\***, **Reset\***, and **ColdReset\*** will continue to operate in their normal fashion. If the **SysAD** bus is not idle when the WAIT instruction completes the W pipe-

stage, then the WAIT is treated as a NOP until the bus operation is completed. Once the processor is in Standby, any interrupt, including the internally generated timer interrupt, causes the processor to exit Standby mode and resume operation where it left off. The WAIT instruction is typically inserted in the idle loop of the operating system or real time executive.

## JTAG Interface

The RM5261 interface supports JTAG boundary scan in conformance with the IEEE 1149.1 specification. The JTAG interface is especially helpful for checking the integrity of the processors pin connections.

## Boot-Time Options

Fundamental operational modes for the processor are initialized by the boot-time mode control interface. This serial interface operates at a very low frequency (**SysClock** divided by 256). The low frequency operation allows the initialization information to be kept in a low cost EPROM or system interface ASIC.

Immediately after the **VccOK** signal is asserted, the processor reads a serial bit stream of 256 bits to initialize all the fundamental operational modes. **ModeClock** run continuously from the assertion of **VccOK**.

## Boot-Time Modes

The boot-time serial mode stream is defined in Table 4. Bit 0 is the bit presented to the processor as the first bit in the stream when **VccOK** is de-asserted. Bit 255 is the last bit transferred.

**Table 4: Boot-Time Mode Bit Stream**

Mode bit	Description
0	reserved (must be zero)
4:1	Write-back data rate 0: DDDD 1: DDxDDx 2: DDxxDDxx 3: Dx Dx Dx Dx 4: DDxxxDDxxx 5: DDxxxxDDxxxx 6: Dx Dx Dx Dx Dx 7: DDxxxxxxDDxxxxxx 8: DxxxDxxxDxxxDxxx 9-15 reserved

Mode bit	Description
7:5	Pclock to SysClock Multiplier Mode Bit 20=0 / Mode Bit 20=1 0: Multiply by 2/x 1: Multiply by 3/x 2: Multiply by 4/x 3: Multiply by 5/2.5 4: Multiply by 6/x 5: Multiply by 7/3.5 6: Multiply by 8/x 7: Multiply by 9/4.5
8	Specifies byte ordering. Logically ORed with BigEndian input signal. 0: Little endian 1: Big endian
10:9	Non-Block Write Control 00: R4000 compatible non-block writes 01: reserved 10: pipelined non-block writes 11: non-block write re-issue
11	Timer Interrupt Enable/Disable 0: Enable the timer interrupt on Int*[5] 1: Disable the timer interrupt on Int*[5]
12	Reserved: Must be zero
14:13	Output driver strength - 100% = fastest 00: 67% strength 01: 50% strength 10: 100% strength 11: 83% strength
15	Reserved: Must be zero
17:16	System configuration identifiers - software visible in processor Config[21..20] register
19:18	Reserved: Must be zero
20	Select Pclock to SysClock Multiply Mode 0: Integer Multipliers 1: Half-Integer Multipliers
21	External Bus Width 0: 64-bit 1: 32-bit
255:22	Reserved: Must be zero

## PIN DESCRIPTIONS

The following is a list of interface, interrupt, and miscellaneous pins available on the RM5261.

Pin Name	Type	Description
<b>System Interface</b>		
ExtRqst*	Input	External request Signals that the system interface is submitting an external request.
Release*	Output	Release interface Signals that the processor is releasing the system interface to slave state
RdRdy*	Input	Read Ready Signals that an external agent can now accept a processor read.
WrRdy*	Input	Write Ready Signals that an external agent can now accept a processor write request.
ValidIn*	Input	Valid Input Signals that an external agent is now driving a valid address or data on the SysAD bus and a valid command or data identifier on the SysCmd bus.
ValidOut*	Output	Valid output Signals that the processor is now driving a valid address or data on the SysAD bus and a valid command or data identifier on the SysCmd bus.
SysAD[63:0]	Input/Output	System address/data bus A 64-bit address and data bus for communication between the processor and an external agent.
SysADC[7:0]	Input/Output	System address/data check bus An 8-bit bus containing parity check bits for the SysAD bus during data cycles.
SysCmd[8:0]	Input/Output	System command/data identifier bus A 9-bit bus for command and data identifier transmission between the processor and an external agent.
SysCmdP	Input/Output	Reserved for system command/data identifier bus parity For the RM5261, unused on input and zero on output.
<b>Clock/Control Interface</b>		
SysClock	Input	System clock Master clock input used as the system interface reference clock. All output timings are relative to this input clock. Pipeline operation frequency is derived by multiplying this clock up by the factor selected during boot initialization
VccP	Input	Quiet Vcc for PLL Quiet Vcc for the internal phase locked loop. Must be connected to VccInt through a filter circuit.
VssP	Input	Quiet Vss for PLL Quiet Vss for the internal phase locked loop. Must be connected to VssInt through a filter circuit.
<b>Interrupt Interface</b>		
Int*[5:0]	Input	Interrupt Six general processor interrupts, bit-wise ORed with bits 5:0 of the interrupt register.
NMI*	Input	Non-maskable interrupt Non-maskable interrupt, ORed with bit 6 of the interrupt register.
<b>JTAG Interface</b>		
JTDI	Input	JTAG data in JTAG serial data in.
JTCK	Input	JTAG clock input JTAG serial clock input.
JTDO	Output	JTAG data out JTAG serial data out.
JTMS	Input	JTAG command JTAG command signal, signals that the incoming serial data is command data.

Pin Name	Type	Description
<b>Initialization Interface</b>		
BigEndian	Input	Allows the system to change the processor addressing mode without rewriting the mode ROM.
VccOK	Input	Vcc is OK When asserted, this signal indicates to the RM5261 that the 3.3V power supply has been above 3.0V for more than 100 milliseconds and will remain stable. The assertion of VccOK initiates the reading of the boot-time mode control serial stream.
ColdReset*	Input	Cold reset This signal must be asserted for a power on reset or a cold reset. ColdReset must be de-asserted synchronously with SysClock.
Reset*	Input	Reset 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 SysClock.
ModeClock	Output	Boot mode clock Serial boot-mode data clock output at the system clock frequency divided by 256.
ModeIn	Input	Boot mode data in Serial boot-mode data input.

## ABSOLUTE MAXIMUM RATINGS<sup>1</sup>

Symbol	Rating	Limits	Unit
V <sub>TERM</sub>	Terminal Voltage with respect to Vss	-0.5 <sup>2</sup> to +3.9	V
T <sub>CASE</sub>	Operating Temperature	0 to +85	°C
T <sub>STG</sub>	Storage Temperature	-55 to +125	°C
I <sub>IN</sub>	DC Input Current	20 <sup>3</sup>	mA
I <sub>OUT</sub>	DC Output Current	50 <sup>4</sup>	mA

Note 1: Stresses greater than those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect reliability.

Note 2: V<sub>IN</sub> minimum = -2.0V for pulse width less than 15ns. V<sub>IN</sub> should not exceed 3.9 Volts.

Note 3: When V<sub>IN</sub> < 0V or V<sub>IN</sub> > VccIO

Note 4: Not more than one output should be shorted at a time. Duration of the short should not exceed 30 seconds.

## RECOMMENDED OPERATING CONDITIONS

Grade	Temperature	Vss	VccInt	VccIO	VccP
Commercial	0°C to +85°C (Case)	0V	2.5V ± 5%	3.3V ± 5%	2.5V ± 5%

Note: VCC I/O should not exceed VccInt by greater than 1.2V during the power-up sequence.

Note: Applying a logic high state to any I/O pin before VccInt becomes stable is not recommended.

Note: As specified in IEEE 1149.1 (JTAG), the JTMS pin must be held low during reset to avoid entering JTAG test mode.

## DC ELECTRICAL CHARACTERISTICS

(V<sub>ccIO</sub> = 3.3V ± 5%; T<sub>CASE</sub> = 0°C to +85°C)

Parameter	Minimum	Maximum	Conditions
V <sub>OL</sub>		0.1V	I <sub>OUT</sub>   = 20 μA
V <sub>OH</sub>	V <sub>ccIO</sub> - 0.1V		
V <sub>OL</sub>		0.4V	I <sub>OUT</sub>   = 4 mA
V <sub>OH</sub>	2.4V		
V <sub>IL</sub>	-0.5V	0.2 x V <sub>ccIO</sub>	
V <sub>IH</sub>	0.7 x V <sub>ccIO</sub>	V <sub>ccIO</sub> + 0.5V	
I <sub>IN</sub>		±20 μA ±20 μA	V <sub>IN</sub> = 0 V <sub>IN</sub> = V <sub>ccIO</sub>
C <sub>IN</sub>		10pF	
C <sub>OUT</sub>		10pF	

## POWER CONSUMPTION

Parameter		Conditions: Max: V <sub>ccInt</sub> = 2.625 Typ: V <sub>ccInt</sub> = 2.5V	CPU Clock Speed					
			200 MHz		250 MHz		266 MHz	
			Typ	Max <sup>1</sup>	Typ	Max <sup>1</sup>	Typ	Max <sup>1</sup>
V <sub>ccInt</sub> Power (mWatts)	standby	No SysAD bus activity		350		435		450
	active	R4000 write protocol with no FPU operation	1600	3200	1850	3700	1900	3800
		Write re-issue or pipelined writes with superscalar	1750	3500	2025	4050	2075	4150

Note 1: Worst case instruction mix with worst case supply voltage.

Note: I/O supply power is application dependent, but typically <10% of V<sub>ccInt</sub>.



## AC ELECTRICAL CHARACTERISTICS

(V<sub>CCIO</sub> = 3.3V ± 5%; T<sub>CASE</sub> = 0°C to +85°C)

### Capacitive Load Deration

Parameter	Symbol	Min	Max	Units
Load Derate	C <sub>LD</sub>		2	ns/25pF
IO Power Derate			21	mW/25pF/MHz
IO Power Derate @ 20pF load		4.0	5.5	mW/MHz

### Clock Parameters

Parameter	Symbol	Test Conditions	CPU Speed						Units
			200 MHz		250 MHz		266 MHz		
			Min	Max	Min	Max	Min	Max	
SysClock High	t <sub>SCH</sub>	Transition ≤ 5ns	3		3		3		ns
SysClock Low	t <sub>SCL</sub>	Transition ≤ 5ns	3		3		3		ns
SysClock Frequency <sup>1</sup>			25	100	25	125	33.3	106	MHz
SysClock Period	t <sub>SCP</sub>			40		40		30	ns
Clock Jitter for SysClock	t <sub>Jl</sub>			±200		±150		±150	ps
SysClock Rise Time	t <sub>CR</sub>			2		2		2	ns
SysClock Fall Time	t <sub>CF</sub>			2		2		2	ns
ModeClock Period	t <sub>ModeCKP</sub>			256		256		256	t <sub>SCP</sub>
JTAG Clock Period	t <sub>JTAGCKP</sub>			4		4		4	t <sub>SCP</sub>

Note 1: Operation of the RM5261 is only guaranteed with the Phase Lock Loop Enabled.

### System Interface Parameters<sup>1</sup>

Parameter	Symbol	Test Conditions	200 MHz		250–266 MHz		Units
			Min	Max	Min	Max	
Data Output <sup>2,3</sup>	t <sub>DO</sub>	mode14:13 = 10 <sup>5</sup> (fastest)	1.0	4.5	1.0	4.0	ns
		mode14:13 = 11 <sup>5</sup>	1.0	5.0	1.0	4.0	ns
		mode14:13 = 00 <sup>5</sup>	1.0	5.5	1.0	4.0	ns
		mode14:13 = 01 <sup>5</sup> (slowest)	1.0	6.0	1.0	4.5	ns
Data Setup <sup>4</sup>	t <sub>DS</sub>	t <sub>rise</sub> = see above table	2.5		2.5		ns
Data Hold <sup>4</sup>	t <sub>DH</sub>	t <sub>fall</sub> = see above table	1.0		1.0		ns

Note 1: Timings are measured from 1.5V of the clock to 1.5V of the signal.

Note 2: Capacitive load for all output timings is 50pF.

Note 3: Data Output timing applies to all signal pins whether tristate I/O or output only.

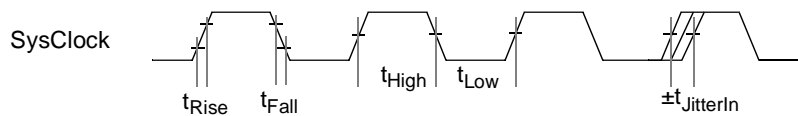
Note 4: Setup and Hold parameters apply to all signal pins whether tristate I/O or input only.

Note 5: Only mode 14:13 = 10 is tested and guaranteed.

## Boot-Time Interface Parameters

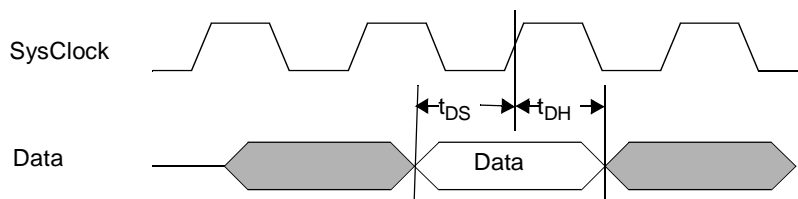
Parameter	Symbol	Test Conditions	Min	Max	Units
Mode Data Setup	$t_{DS(M)}$		4		SysClock cycles
Mode Data Hold	$t_{DH(M)}$		0		SysClock cycles

## TIMING DIAGRAMS

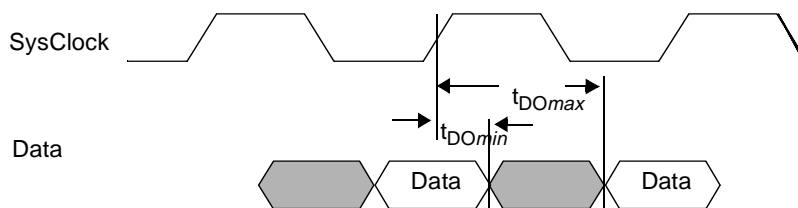


**Figure 8 Clock Timing**

### System Interface Timing (SysAD, SysCmd, ValidIn\*, ValidOut\*, etc.)

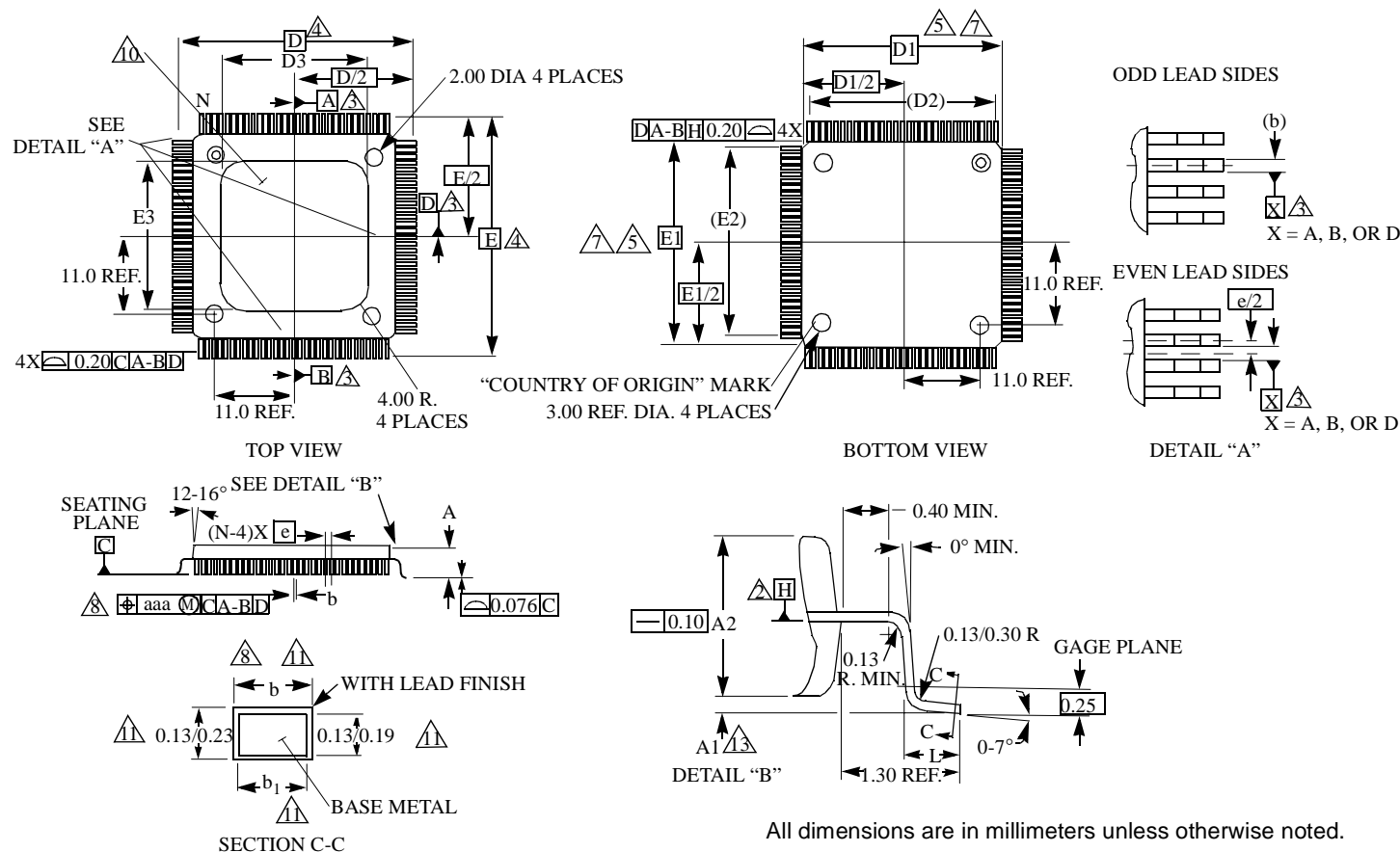


**Figure 9 Input Timing**



**Figure 10 Output Timing**

# PACKAGING INFORMATION



Symbol	208PQ4			Note
	Min	Nominal	Max	
A	--	3.70	4.07	
A1	0.25	0.33	--	
A2	3.20	3.37	3.60	
D	30.60 BSC			To be determined at seating Plane C.
D1	28.00 BSC			Dimensions D1 and E1 do not include mold protrusion. Allowable mold protrusion is 0.254 MM per side. Dimension D1 and E1 do include mold mismatch and are determined at Datum Plane H.
D2	25.00 REF.			
E	30.60 BSC			To be determined at seating Plane C.
E1	28.00 BSC			Dimensions D1 and E1 do not include mold protrusion. Allowable mold protrusion is 0.254 MM per side. Dimension D1 and E1 do include mold mismatch and are determined at Datum Plane H.
E2	25.00 REF.			
D3	21.0 REF.			
E3	21.0 REF.			
L	0.46	0.56	0.66	
N	208			
e	0.50 BSC			
b	0.17	0.22	0.27	
b1	0.17	0.20	0.23	
aaa	0.08			
ThetaJa	13.7° C/W			
ThetaJc	1.5° C/W			

Notes:

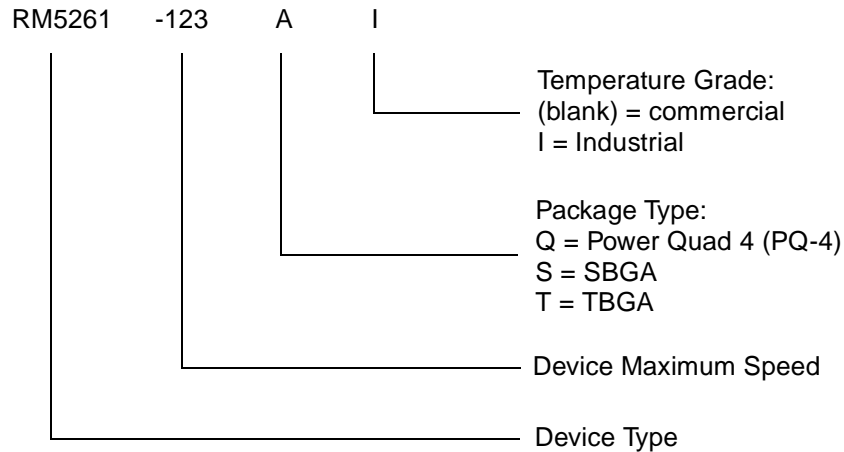
- 1 All dimensioning and tolerances confirm to ASME Y14.5–1994.
- 2 Datum Plane H located at the bottom of the mold parting line and coincident with where lead exits plastic body.
- 3 Datums A–B and D to be determined where center line between leads exits plastic body at Datum Plane H.
- 4 To be determined at seating Plane C.
- 5 Dimensions D1 and E1 do not include mold protrusion. Allowable mold protrusion is 0.254 MM per side. Dimension D1 and E1 do include mold mismatch and are determined at Datum Plane H.
- 6 “N” is number of terminals.
- 7 Package top dimensions are smaller than bottom dimensions by 0.20 millimeters and top of package will not overhang bottom of package.
- 8 Dimensions b does not include Damabr protrusion. Allowable Damabr protrusion shall be 0.08 MM. Total in excess of b dimension at maximum material condition. Damabr can not be located on the lower radius or the foot. The dimension space between protrusion and an adjacent lead shall not be less than 0.07 MM for 0.4 MM and 0.50 MM pitch package.
- 9 All dimensions are in millimeters.
- 10 The optional exposed heat shrink is coincident with the top or bottom side of the package and not allowed to protrude beyond that surface.
- 11 These dimensions apply to the flat section of the lead between 0.10 MM and 0.25 MM from the lead tip.
- 12 This drawing conforms to JEDEC registered outline MS-029. But the heat slug dimension was not specified on JEDEC.
- 13 A1 is defined as the distance from the seating plane to the lowest point of the package body.

## RM5261 208 P-QUAD PACKAGE PINOUT

Pin	Function	Pin	Function	Pin	Function	Pin	Function
1	VccIO	2	NC	3	NC	4	VccIO
5	Vss	6	SysAD4	7	SysAD36	8	SysAD5
9	SysAD37	10	VccInt	11	Vss	12	SysAD6
13	SysAD38	14	VccIO	15	Vss	16	SysAD7
17	SysAD39	18	SysAD8	19	SysAD40	20	VccInt
21	Vss	22	SysAD9	23	SysAD41	24	VccIO
25	Vss	26	SysAD10	27	SysAD42	28	SysAD11
29	SysAD43	30	VccInt	31	Vss	32	SysAD12
33	SysAD44	34	VccIO	35	Vss	36	SysAD13
37	SysAD45	38	SysAD14	39	SysAD46	40	VccInt
41	Vss	42	SysAD15	43	SysAD47	44	VccIO
45	Vss	46	ModeClock	47	JTDO	48	JTDI
49	JTCK	50	JTMS	51	VccIO	52	Vss
53	NC	54	NC	55	NC	56	VccIO
57	Vss	58	Modeln	59	RdRdy*	60	WrRdy*
61	ValidIn*	62	ValidOut*	63	Release*	64	VccP
65	VssP	66	SysClock	67	VccInt	68	Vss
69	VccIO	70	Vss	71	VccInt	72	Vss
73	SysCmd0	74	SysCmd1	75	SysCmd2	76	SysCmd3
77	VccIO	78	Vss	79	SysCmd4	80	SysCmd5
81	VccIO	82	Vss	83	SysCmd6	84	SysCmd7
85	SysCmd8	86	SysCmdP	87	VccInt	88	Vss
89	VccInt	90	Vss	91	VccIO	92	Vss
93	Int0*	94	Int1*	95	Int2*	96	Int3*
97	Int4*	98	Int5*	99	VccIO	100	Vss
101	NC	102	NC	103	NC	104	NC
105	VccIO	106	NMI*	107	ExtRqst*	108	Reset*
109	ColdReset*	110	VccOK	111	BigEndian	112	VccIO
113	Vss	114	SysAD16	115	SysAD48	116	VccInt
117	Vss	118	SysAD17	119	SysAD49	120	SysAD18
121	SysAD50	122	VccIO	123	Vss	124	SysAD19
125	SysAD51	126	VccInt	127	Vss	128	SysAD20
129	SysAD52	130	SysAD21	131	SysAD53	132	VccIO
133	Vss	134	SysAD22	135	SysAD54	136	VccInt
137	Vss	138	SysAD23	139	SysAD55	140	SysAD24
141	SysAD56	142	VccIO	143	Vss	144	SysAD25
145	SysAD57	146	VccInt	147	Vss	148	SysAD26
149	SysAD58	150	SysAD27	151	SysAD59	152	VccIO
153	Vss	154	NC	155	NC	156	Vss
157	NC	158	NC	159	NC	160	NC
161	VccIO	162	Vss	163	SysAD28	164	SysAD60
165	SysAD29	166	SysAD61	167	VccInt	168	Vss
169	SysAD30	170	SysAD62	171	VccIO	172	Vss
173	SysAD31	174	SysAD63	175	SysADC2	176	SysADC6

Pin	Function	Pin	Function	Pin	Function	Pin	Function
177	VccInt	178	Vss	179	SysADC3	180	SysADC7
181	VccIO	182	Vss	183	SysADC0	184	SysADC4
185	VccInt	186	Vss	187	SysADC1	188	SysADC5
189	SysAD0	190	SysAD32	191	VccIO	192	Vss
193	SysAD1	194	SysAD33	195	VccInt	196	Vss
197	SysAD2	198	SysAD34	199	SysAD3	200	SysAD35
201	VccIO	202	Vss	203	NC	204	NC
205	NC	206	NC	207	VccIO	208	Vss

## ORDERING INFORMATION



## Valid Combinations

RM5261-200Q  
RM5261-250Q  
RM5261-266Q

This document may, wholly or partially, be subject to change without notice. Quantum Effect Devices, Inc. reserves the right to make changes to its products or specifications at any time without notice, in order to improve design or performance and to supply the best possible product.

All rights are reserved. No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without QED's permission.

QED will not be held responsible for any damage to the user or any property that may result from accidents, misuse, or any other causes arising during operation of the user's unit.

**LIFE SUPPORT POLICY:** QED's products are not designed, intended, or authorized for use as components intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which failure of the product could create a situation where personal injury or death may occur. Should a customer purchase or use the products for any such unintended or unauthorized application, the customer shall indemnify and hold QED and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that QED was negligent regarding the design or manufacture of the part.

QED does not assume any responsibility for use of any circuitry described other than the circuitry embodied in a QED product. The company makes no representations that the circuitry described herein is free from patent infringement or other rights of third parties, which may result from its use. No license is granted by implication or otherwise under any patent, patent rights, or other rights, of QED.

The QED logo and RISCMark are trademarks of Quantum Effect Devices, Inc.

MIPS is a registered trademark of MIPS Technologies, Inc. All other trademarks are the respective property of the trademark holders.

Document Number: RM5261-DS0011300001

Quantum Effect Devices  
3255-3 Scott Blvd, Suite 200  
Santa Clara, CA. 95054  
phone (408) 565-0300 fax (408) 565-0330  
sales@qedinc.com