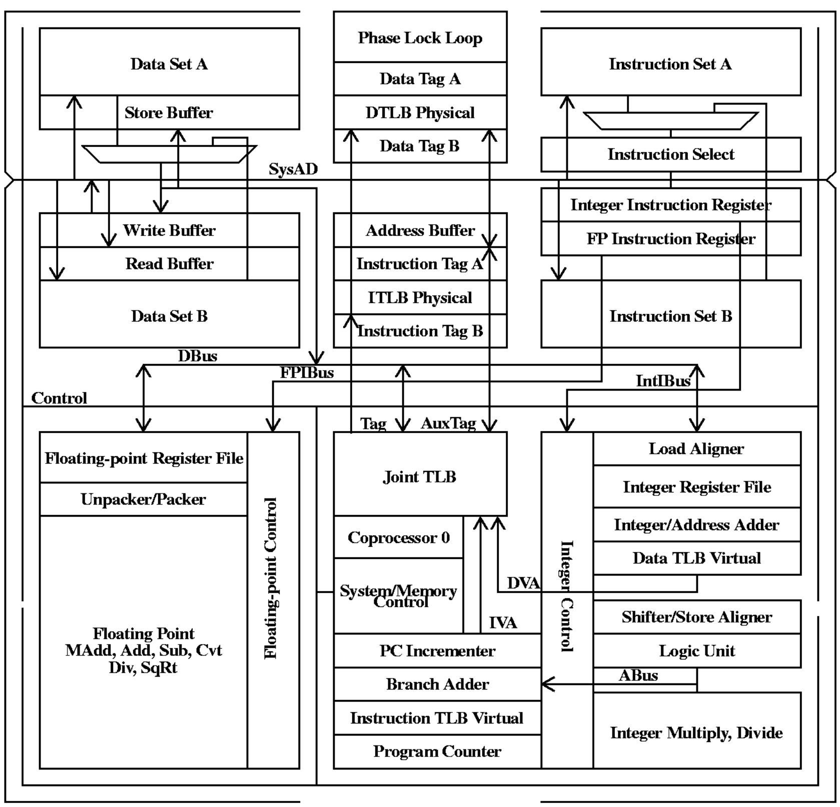
QED RISCMark™ RM5270™

64-Bit Superscalar Microprocessor

FEATURES:

- Dual Issue superscalar microprocessor - can issue one integer and one floating-point instruction per cycle
  - — 150, 200 MHz operating frequencies
  - — 260 Dhrystone2.1 MIPS
  - — SPECInt95 5.0, SPECfp95 5.3
- High performance system interface compatible with RM5260, R4600, R4700 and R5000
  - — 64-bit multiplexed system address/data bus for optimum price/performance with up to 100 MHz operating frequency
  - — High performance write protocols maximize uncached write bandwidth
  - — Supports clock multipliers (2, 3, 4, 5, 6, 7, 8)
  - — 5V compatible I/O's
  - — IEEE 1149.1 JTAG boundary scan
- Integrated on-chip caches
  - — 16KB instruction - 2 way set associative
  - — 16KB data - 2 way set associative
  - — Virtually indexed, physically tagged
  - — Write-back and write-through on per page basis
  - — Pipeline restart on first double 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)
- Integrated secondary cache controller (R5000 compatible)
  - — Supports 512K or 2MByte block write-through secondary
- High-performance floating point unit
  - — 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 instruction and 3 operand multiply instruction
  - — I and D cache locking by set
  - — Optional dedicated exception vector for interrupts
- Fully static CMOS design with power down logic
  - — Standby reduced power mode with WAIT instruction
  - — 6 Watts typical at 3.3V 200 MHz
- 304 SBGA package (31x31mm)

BLOCK DIAGRAM:



DESCRIPTION:

The QED RM5270 is a highly integrated superscalar microprocessor that implements a superset of the MIPS IV Instruction Set Architecture (ISA). It has a high performance 64-bit integer unit, a high throughput, fully pipelined 64-bit floating point unit, an operating system friendly memory management unit with a 48-entry fully associative TLB, a 16 KByte 2-way set associative instruction cache, a 16 KByte 2-way set associative data cache, and a high-performance 64-bit system interface with support for an optional external secondary cache. The RM5270 can issue both an integer and a floating point instruction in the same cycle.

The RM5270 is ideally suited for high-end embedded control applications such as internetworking, high performance image manipulation, high speed printing, and 3-D visualization.The RM5270 is also applicable to the low end workstation market where its balanced integer and floating-point performance and direct support for a large secondary cache (up to 2MB) provide outstanding price/performance

HARDWARE OVERVIEW:

The RM5270 offers a high-level of integration targeted at high-performance embedded applications. The key elements of the RM5270 are briefly described below.

Superscalar Dispatch

The RM5270 has an efficient asymmetric superscalar dispatch unit which allows it to issue an integer instruction and a floating-point computation instruction simultaneously. With respect to superscalar issue, 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 RM5270 provides unparalleled price/performance in computationally intensive embedded applications.

CPU Registers

Like all MIPS ISA processors, the RM5270 CPU has a simple, clean 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.

Pipeline

For integer operations, loads, stores, and other non-floating-point operations, the RM5270 uses the simple 5-stage pipeline also found in the R4600, R4700, and R5000 devices. In addition to this standard pipeline, the RM5270 uses an extended 7-stage pipeline for floating-point operations. Like the R5000, the RM5270 does virtual to physical translation in parallel with cache access.

Figure 2 on page 3 shows the RM5270 integer pipeline. As illustrated in the figure, up to five integer instructions can be executing simultaneously.

Integer Unit

As part of the RM52xx family, the RM5270 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 RM5270 includes two 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.

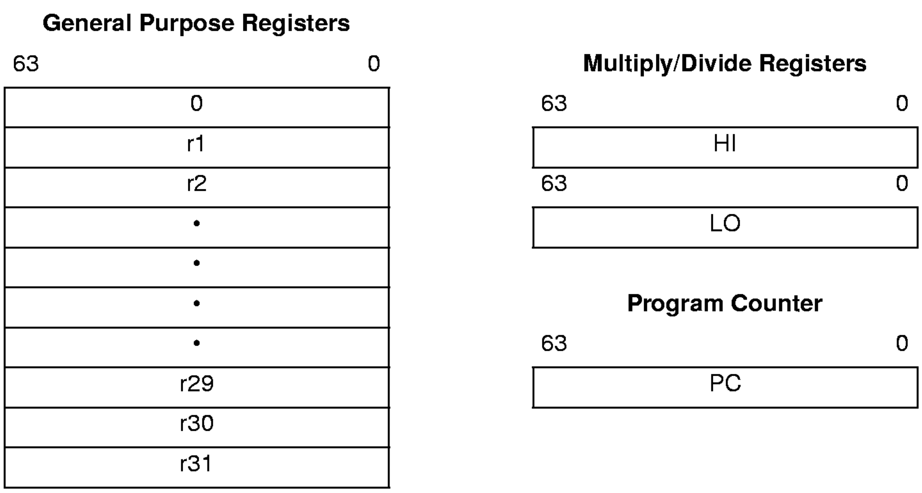
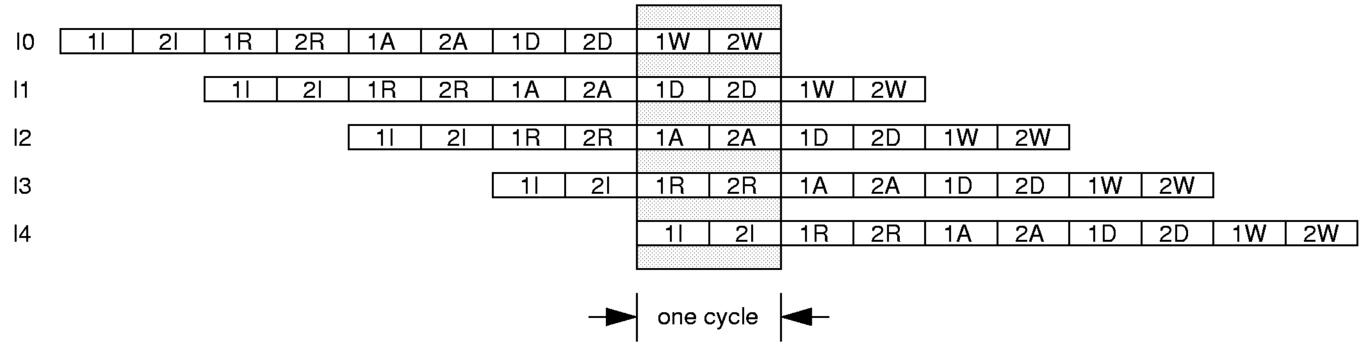


Figure 1 CPU Registers



1I-1R: Instruction cache access
2I: Instruction virtual to physical address translation
2R: Register file read, Bypass calculation, Instruction decode, Branch address calculation
1A: Issue or slip decision, Branch decision
1A: Data virtual address calculation
1A-2A: Integer add, logical, shift
2A: Store Align
2A-2D: Data cache access and load align
1D: Data virtual to physical address translation
2W: Register file write

Figure 2 Pipeline

The RM5270 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*).

Register File

The RM5270 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 RM5270 ALU consists of the integer adder/subtractor, the logic unit, and the shifter. The adder performs address calculations in addition to arithmetic operations, the logic unit performs all logical and zero shift data moves, and 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 RM5270 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.

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 RM5270 also implements the multiply instruction, **MUL**, first introduced in the R4650. 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 RM5270 is the multiply-add instruction, **MAD**, likewise introduced in the R4650. 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 RM5270 to eliminate the need for a separate DSP engine in many embedded applications.

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

By pipelining the multiply-accumulate function and dynamically determining the size of the input operands, the RM5270 is able to maximize throughput while still using an area efficient implementation.

Floating-Point Co-Processor

The RM5270 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 co-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. As described earlier, the superscalar capabilities of the RM5270 allow floating-point computation instructions to issue concurrently with integer instructions.

Floating-Point Unit

The RM5270 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 divide/square root and multiply/add is supported.

The RM5270 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.

The floating-point unit's operation set includes floating-point 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, and 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

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 co-processor, co-processor 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 RM5270 CP0 is logically identical to that of the R5000.

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 RM5270 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

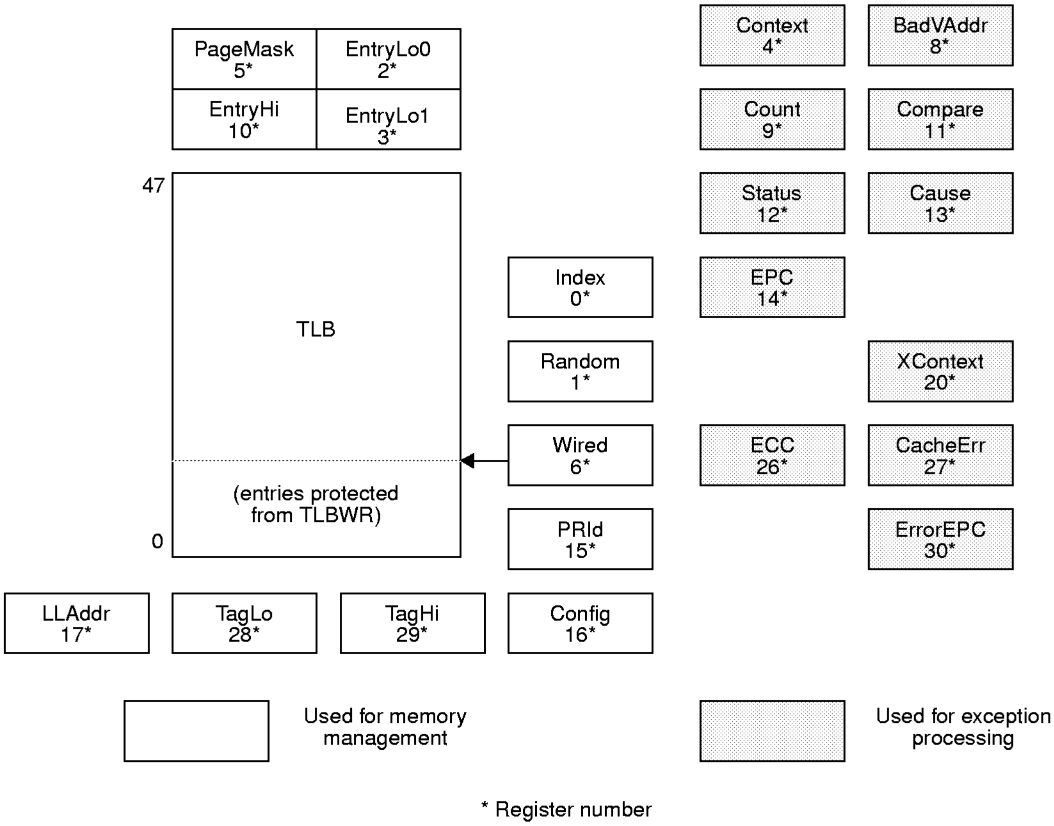


Figure 3 CP0 Registers

operating modes are controlled (kernel vs. user mode, interrupts enabled or disabled, cache features). In addition, the RM5270 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 RM5270 provides three modes of virtual addressing:

- user mode
- supervisor mode
- kernel mode

This mechanism is available to system software to provide a secure environment for user processes. Bits in the CP0 *Status* register determine which virtual addressing mode is used. In the user mode, the RM5270 provides a single, uniform virtual address space of 256GB (2GB in 32-bit mode).

When operating in the kernel mode, four distinct virtual address spaces, totalling 1024GB (4GB in 32-bit mode), are simultaneously available and are differentiated by the high-order bits of the virtual address.

The RM5270 processors also support a supervisor mode in which the virtual address space is 256.5GB (2.5GB in 32-bit mode), divided into three regions based on the high-order bits of the virtual address.

Figure 4 on page 6 shows the address space layout for 32-bit operation.

When the RM5270 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.

Joint TLB

For fast virtual-to-physical address translation, the RM5270 uses a large, fully associative TLB that maps 96 virtual pages to their corresponding physical addresses. As indicated by its name, the joint TLB, or 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). A CP0 register, *Page Mask*, 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, a

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 mode)

typical frame buffer can be memory mapped using only one TLB entry.

The second mechanism controls the replacement algorithm when a TLB miss occurs. The RM5270 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: uncached, non-coherent write-back, non-coherent write-through with write-allocate, non-coherent write-through without write-allocate, sharable, exclusive, or update. Note that both of the write-through protocols bypass the secondary cache since the secondary does not support writes of less than a complete cache line.

The non-coherent protocols are used for both code and data on the RM5270 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 RM5260, R4600 and R4700.

The coherent attributes, if used, generate coherent transaction types on the system interface. Like the R5000, however, cache coherency is not supported and, therefore, the coherent attributes should never be used.

Instruction TLB

The RM5270 uses 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 RM5270 uses 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

In order to keep the RM5270's high-performance pipeline full and operating efficiently, the RM5270 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 200MHz. For applications requiring even higher performance, the RM5270 also has a direct interface to a large external secondary cache.

Instruction Cache

The RM5270 incorporates a two-way set associative on-chip instruction cache. This virtually indexed, physically tagged cache is 16KB in size and is protected with word parity.

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

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 RM5270 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 RM5270 includes a 16KB 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

The normal write policy is write-back, which means that 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. Cache protocols supported for the data cache are:

1. **Uncached.** Reads to addresses in a memory area identified as uncached will not access the cache. Writes to such addresses will be written directly to main memory without updating the cache.
2. **Write-back.** Loads and instruction fetches will first search the cache, reading the next memory hierarchy level 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 will be updated, and the cache line 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 will first search the cache, reading from memory only if the desired data is not cache resident; write-through data is never cached in the secondary cache. 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 will be updated and main memory will also be written leaving the *write-back* bit of the cache line unchanged; no writes will occur into the secondary. 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 will first search the cache, reading from memory only if the desired data is not cache resident; write-through data is never cached in the secondary cache. 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 will be updated and main memory will also be written leaving the *write-back* bit of the cache line unchanged; no writes will occur into the secondary. If the cache lookup misses, then only main memory is written.

Associated with the Data Cache is the store buffer. When the RM5270 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 RM5270 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 will occur such that no penalty is incurred.

Secondary Cache

Like the R5000, the RM5270 has direct support for an external secondary cache. The secondary cache is direct mapped and block write-through with byte parity protection for data. The RM5270 secondary operates identically to that of the R5000 and supports the same 512K and 2 MByte cache sizes supported by the R5000 (assuming nKx18 RAM organization due to capacitive loading constraints).

The secondary interface uses the SysAD bus for data and tags while providing a separate bus, ScLine, for addresses, and a handful of secondary specific control signals (for the complete set, see Pin Listing).

A secondary read looks nearly identical to a standard processor read except that the tag chip enable signal, **ScTCE***, is asserted concurrently with **ValidOut*** and **Release***, initiating a tag probe and indicating to the external controller that a secondary cache access is being performed. As a result, the external controller monitors the secondary hit signal, **ScMatch**, and if a hit is indicated the controller will abort the memory read and will refrain from acquiring control of the system interface. Along with **ScTCE***, the processor also asserts the tag data enable signal, **ScTDE***, which causes the tag RAM's to latch the SysAD address internally for use as the replacement tag if a cache miss occurs.

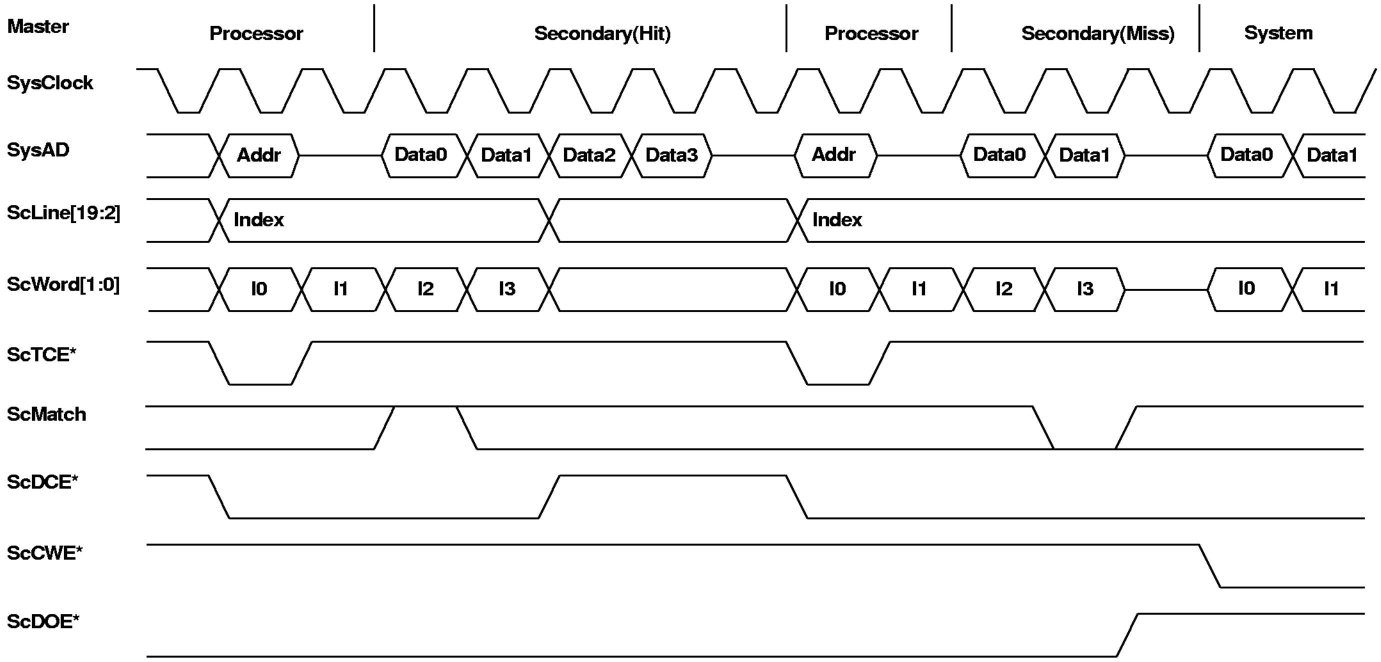


Figure 5 Secondary Cache Hit and Miss

On a secondary miss, a refill is accomplished with a two signal handshake between the data output enable signal, **ScDOE***, which is deasserted by the controller and the tag and data write enable signal, **ScCWE***, which is asserted by the processor. Figure 5 on page 8 illustrates a hit followed by a miss in the secondary cache.

Other capabilities of the secondary interface include block write, tag invalidate, and tag probe. For details of these transactions as well as detailed timing waveforms for all the secondary transactions, see the RM52XX Family Users Manual. The secondary cache can be implemented with the Motorola MCM69T618 or its equivalent.

The RM5270 cache attributes for the instruction, data, and optional external secondary caches are summarized in Table 3.

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 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 RM5270 provides a high performance 64-bit multiplexed address/data system interface for optimum price/performance. This interface is backward compatible with the R5000. Unlike the R4000 and R5000 family processors which provide only an integral multiplication factor between **SysClock** and the pipeline clocks.

The system interface consists of a 64-bit Address/Data bus with 8 check bits and a 9-bit command bus. In addition, there are 6 handshake signals and 6 interrupt inputs. The

Table 3: Cache Attributes

Characteristics	Instruction	Data	Secondary
Size	16KB	16KB	512K or 2M
Organization	2-way set associative	2-way set associative	direct mapped
Replacement Algorithm	FIFO	FIFO	direct replacement
Line size	32B	32B	32B
Index	$vAddr_{11..0}$	$vAddr_{11..0}$	pAddr20..0
Tag	$pAddr_{31..12}$	$pAddr_{31..12}$	pAddr53..19
Write policy	n.a.	write-back/ write-through	block write-through
line transfer order	read: sub-block order	read: sub-block order	read: sub-block order
	write: sequential	write: sequential	write: sequential
miss restart after transfer of	entire line	first double	NA
Parity	per-word	per-byte	per-byte
Cache locking	set A	set A	none

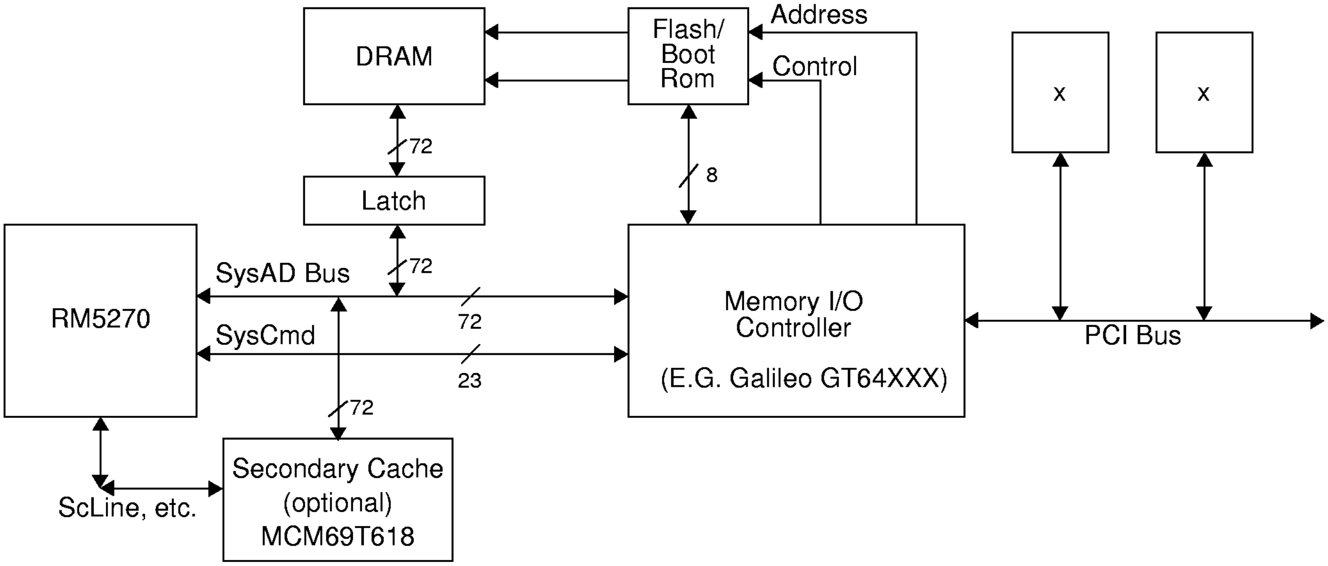


Figure 6 Typical Embedded System Block Diagram

interface has a simple timing specification and is capable of transferring data between the processor and memory at a peak rate of 800MB/sec with a 100MHz SysClock.

Figure 6 shows a typical embedded system using the RM5270. In this example, a bank of DRAMs, an optional secondary cache, 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 RM5270 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 RM5270 transmits data to the system interface are programmable via boot time mode control bits. Also, the rate at which the processor receives data is fully controlled by the external device. Therefore, either a low cost interface requiring no read or write buffering or a faster, high performance interface can be designed to communicate with the RM5270. Again, the system designer has the flexibility to make these price/performance trade-offs.

System Command Bus

The RM5270 interface has a 9-bit System Command (SysCmd) bus. The command bus indicates whether the SysAD bus carries an address or data. If the SysAD carries an address, then the SysCmd bus also indicates what type of transaction is to take place (for example, a read or write). If the SysAD carries data, then the SysCmd bus also gives information about the data (for example, this is the last data word transmitted, or the data contains an error). The SysCmd bus is bidirectional to support both processor requests and external requests to the RM5270. Processor requests are initiated by the RM5270 and responded to by an external device. External requests are issued by an external device and require the RM5270 to respond.

The RM5270 supports one to eight byte and block transfers on the SysAD bus. In the case of a sub-doubleword 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 RM5270 whether it can accept a new read or write transaction. The RM5270 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 RM5270 responds by asserting **Release*** to release the system interface to slave state.

ValidOut* and **ValidIn*** are used by the RM5270 and the external device respectively to indicate that there is a valid command or data on the SysAD and SysCmd buses. The RM5270 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 RM5270 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 RM5270 issues another request. The RM5270 can issue read and write requests to an external device, whereas an external device can issue null and write requests to the RM5270.

For processor reads the RM5270 asserts **ValidOut*** and simultaneously drives the address and read command on the SysAD and SysCmd buses. If the system interface has **RdRdy*** asserted, then the processor tristates its drivers and releases the system interface to slave state by asserting **Release***. The external device can then begin sending data to the RM5270.

Figure 7 shows a processor block read request and the external agent read response for a system with no external secondary cache. The read latency is 4 cycles (**ValidOut*** to **ValidIn***), and the response data pattern is DDxxDD. Figure 8 on page 11 shows a processor block write using write response pattern DDxxDDxx, or code 2, of the boot time mode select options. Again, there is no secondary present.

Enhanced Write Modes

Like the RM5260, R4600, R4700, and R5000, the RM5270 implements two enhancements to the original R4000 write mechanism: Write Reissue and Pipeline Writes. In write reissue mode, a write rate of one write every two bus cycles can be achieved. A write issues if **WrRdy*** is asserted two cycles earlier and is still asserted during the issue cycle. If it is not still asserted then the last write will reissue. 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 RM5270 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; i.e., the external device wants the processor interface to go from slave state to master state. Typically, a null request will be executed after an external device, that has acquired control of the processor interface via **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

In order to provide better real time interrupt handling, the RM5270 supports the same dedicated interrupt vector introduced in the R4650. When enabled by the real time executive, by setting a bit in the Cause register, interrupts vector to a specific address which 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 RM5270 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 enters Standby Mode. When the wait instruction completes the W pipe stage, if the SysAD bus is currently idle, the

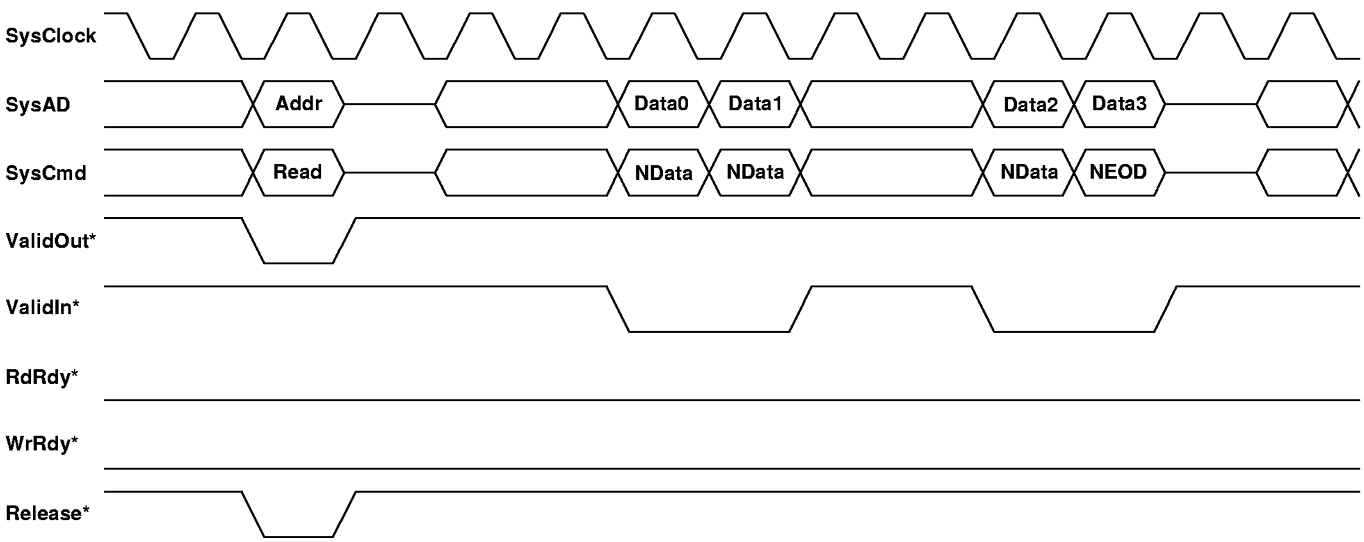


Figure 7 Processor Block Read

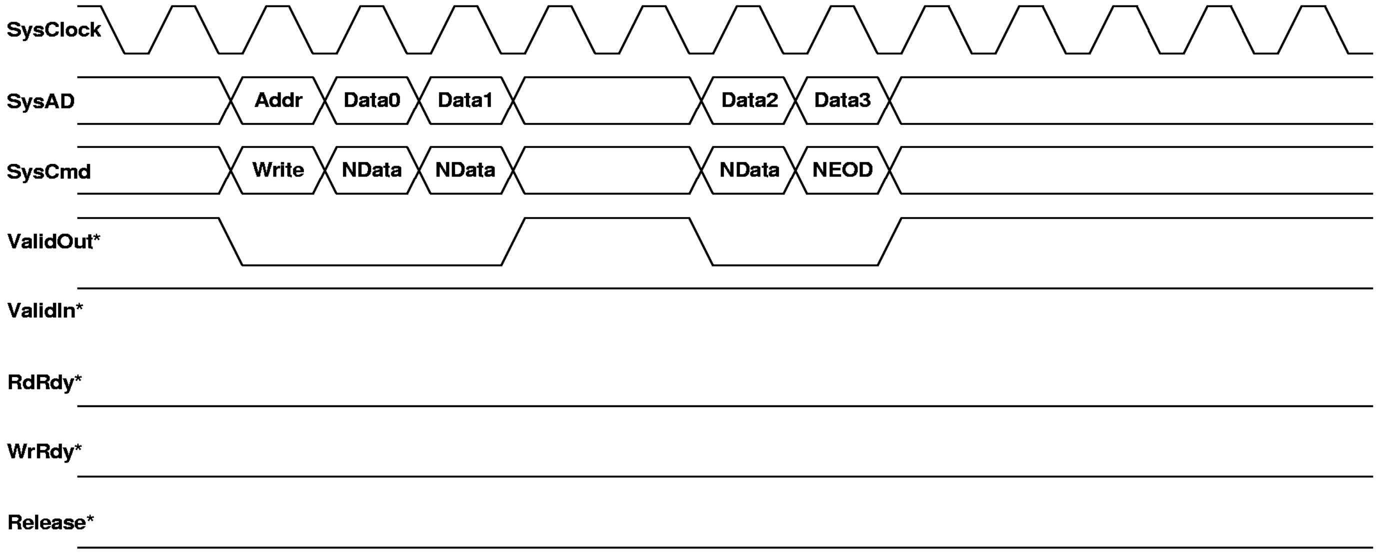


Figure 8 Processor Block Write

internal processor clocks will stop 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**. Once the processor is in Standby, any interrupt, including the internally generated timer interrupt, will cause the processor to exit Standby 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 RM5270 interface supports JTAG boundary scan in conformance with IEEE 1149.1. The JTAG interface is especially helpful for checking the integrity of the processor's pin connections.

Boot-Time Options

Fundamental operational modes for the processor are initialized by the boot-time mode control interface. The boot-time mode control interface is a serial interface operating 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; alternatively the twenty or so bits could be generated by the 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 when **VccOk** is deasserted; bit 255 is the last.

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: DxDxDxDx 4: DDxxxDDxxx 5: DDxxxxDDxxxx 6: DxxDxxDxxDxx 7: DDxxxxxxDDxxxxxx 8: DxxxDxxxDxxxDxxx 9-15: reserved
7..5	Pclock to SysClock Multiplier 0: Multiply by 2 1: Multiply by 3 2: Mutiply by 4 3: Multiply by 5 4: Multiply by 6 5: Multiply by 7 6: Multiply by 8 7: reserved
8	Specifies byte ordering. Logically ORed with BigEndian input signal. 0: Little endian 1: Big endian
10..9	00: R4000 compatible non-block writes 01: reserved 10: pipelined non-block writes 11: non-block write re-issue
11	0: Enable the timer interrupt on Int[5] 1: Disable the timer interrupt on Int[5].
12	0: Disable secondary cache 1: Enable secondary cache
14..13	Output driver strength - 100% = fastest 00: 67% strength 01: 50% strength 10: 100% strength 11: 83% strength
15	Select secondary cache RAM type 0: Dual-cycle deselect 1: Single-cycle deselect
17..16	System configuration identifiers - software visible in processor Config[21..20] register
255..18	Reserved: Must be zero

PIN DESCRIPTIONS:

The following is a list of interface, interrupt, and miscellaneous pins available on the RM5270.

Pin Name	Type	Description
System interface:		
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 RM5270, unused on input and zero on output.
Clock/control interface:		
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.
VssP	Input	Quiet VSS for PLL Quiet Vss for the internal phase locked loop.
Secondary cache interface:		
ScCLR*	Output	Secondary Cache Block Clear Requests that all valid bits be cleared in the Tag RAMs. Many RAM's may not support a block clear therefore the block clear capability is not required for the cache to operate.
ScCWE*(1:0)	Output	Secondary Cache Write Enable Asserted to cause a write to the cache. Two identical signals are provided to balance the capacitive load relative to the remaining cache interface signals.
ScDCE*(1:0)	Output	Secondary Cache Data RAM Chip Enable When asserted this signal causes the data RAM's to read out their contents. Two identical signals are provided to balance the capacitive load relative to the remaining cache interface signals
ScDOE*	Input	Secondary Cache Data RAM Output enable When asserted this signal causes the data RAM's to drive data onto their I/O pins. This signal is monitored by the processor to determine when to drive the data RAM write enable in a secondary cache miss refill sequence.
ScLine(15:0)	Output	Secondary Cache Line Index
ScMatch	Input	Secondary Cache Tag Match This signal is asserted by the cache Tag RAM's when a match occurs between the value on its data inputs and the contents of its RAM at the value on its address inputs.

Pin Name	Type	Description
ScTCE*	Output	Secondary Cache Tag RAM Chip Enable When asserted this signal will cause either a probe or a write of the Tag RAM's depending on the state of the Tag RAM's write enable signal. This signal is monitored by the external agent and indicates to it that a secondary cache access is occurring.
ScTDE*	Output	Secondary Cache Tag RAM Data Enable When asserted this signal causes the value on the data inputs of the Tag RAM to be latched into the RAM. If a refill of the RAM is necessary, this latched value will be written into the Tag RAM array. Latching the Tag allows a shared address/data bus to be used without incurring a penalty to re-present the Tag during the refill sequence.
ScTOE*	Output	When asserted this signal causes the Tag RAM's to drive data onto their I/O pins.
ScWord(1:0)	Input/Output	Secondary Cache Double Word Index Driven by the processor on cache hits and by the external agent on cache miss refills.
ScValid	Input/Output	Secondary Cache Valid This signal is driven by the processor as appropriate to make a cache line valid or invalid. On Tag read operations the Tag RAM will drive this signal to indicate the line state.
Interrupt interface:		
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.
JTAG interface:		
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.
Initialization interface:		
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 RM5270 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 two hundred and fifty six.
ModeIn	Input	Boot mode data in Serial boot-mode data input.
Reserved Pins		
Reserved	Input/Output	Reserved for RM7000 compatibility

ABSOLUTE MAXIMUM RATINGS:[1]

Symbol	Rating	Limits	Unit
V_{TERM}	Terminal Voltage with respect to GND	–0.5[2] to +5.5	V
T_{CASE}	Operating Temperature	0 to +85	°C
T_{BIAS}	Case Temperature Under Bias	–55 to +125	°C
T_{STG}	Storage Temperature	–55 to +125	°C
I_{IN}	DC Input Current	20[3]	mA
I_{OUT}	DC Output Current	50	mA

Notes: 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.

2. V_{IN} minimum = -2.0V for pulse width less than 15ns. V_{IN} should not exceed 5.5 Volts.
3. When V_{IN} < 0V or V_{IN} > VCC
4. Not more than one output should be shorted at a time. Duration of the short should not exceed 30 seconds.

RECOMMENDED OPERATING TEMPERATURE AND SUPPLY VOLTAGE:

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

DC ELECTRICAL CHARACTERISTICS:

(VccInt = VccIO = 3.3V ± 5%; T_{CASE} = 0°C to +85°C)

Parameter	133/150/175/200 MHz		Conditions
	Minimum	Maximum	
V_{OL}		0.1V	$\lvert I_{OUT}\rvert$= 20 µA
V_{OH}	VccIO - 0.1V		
V_{OL}		0.4V	$\lvert I_{OUT}\rvert$ = 4 mA
V_{OH}	2.4V		
V_{IL}	-0.5V	0.2 VccIO	
V_{IH}	0.7 * VccIO	VccIO + 0.5V	
I_{IN}		±20 µA ±20 µA ±250 µA	V_{IN} = 0 V_{IN} = VccIO V_{IN} = 5.5V
C_{IN}		10pF	
C_{OUT}		10pF	

POWER CONSUMPTION:

Parameter		Conditions Max. = 3.45 Vcc, Typ.[5] = 3.3 Vcc	150/75 MHz, 3.45V		200/100 MHz, 3.45V	
			Typ[5]	Max	Typ[5]	Max
ICC (mA)	standby	CL = 0pF	60	100	70	110
		CL = 50pF	60	100	70	110
	active	CL = 0pF, no SysAD activity	700	1450	1000	1950
		CL = 50pF, R4000 write protocol	950	1700	1250	2250
		CL = 50pF, write re-issue or pipe-lined writes	1125	1900	1475	2550

Note: 5. Typical integer instruction mix and cache miss rates.

AC ELECTRICAL CHARACTERISTICS:

(VccInt = VccIO = 3.3V ± 5%; T_{CASE} = 0°C to +85°C)

Capacitive Load Deration:

Parameter	Symbol	133/150/175/200 MHz		Units
		Min	Max	
Load Derate	C_{LD}		2	ns/25pF

Clock Parameters:

Parameter	Symbol	Test Conditions	150/75 MHz		200/100 MHz		Units
			Min	Max	Min	Max	
SysClock High	t_{SCHigh}	Transition ≤ 5ns	4		4		ns
SysClock Low	t_{SCLow}	Transition ≤ 5ns	4		4		ns
SysClock Frequency[6]			20	75	20	100	MHz
SysClock Period	t_{SCP}			50		50	ns
Clock Jitter for SysClock	$t_{JitterIn}$			±200		±200	ps
SysClock Rise Time	t_{SCRise}			4		3	ns
SysClock Fall Time	t_{SCFall}			4		3	ns
ModeClock Period	$t_{ModeCKP}$			256*t_{SCP}		256*t_{SCP}	ns
JTAG Clock Period	$t_{JTAGCKP}$			4*t_{SCP}		4*t_{SCP}	ns

Note: 6. Operation of the RM5270 is only guaranteed with the Phase Lock Loop Enabled.

System Interface Parameters:[7]

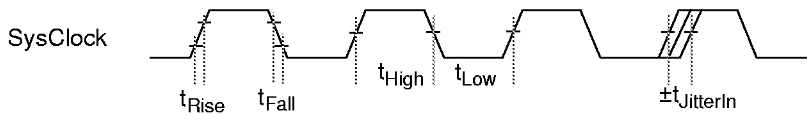
Parameter	Symbol	Test Conditions	150/75 MHz		200/100 MHz		Units
			Min	Max	Min	Max	
Data Output[8,9]	t_{DO}	mode14..13 = 10 (fastest)	1.0	4.5	1.0	4.5	ns
		mode14..13 = 11	1.0	5.0	1.0	5.0	ns
		mode14..13 = 00	1.0	5.5	1.0	5.5	ns
		mode14..13 = 01 (slowest)	1.0	6.5	1.0	6.5	ns
Data Setup[10]	t_{DS}	t_{rise} = see above table t_{fall} = see above table	3.0		2.5		ns
Data Hold[10]	t_{DH}		1.0		1.0		ns

Notes: 7. Timings are measured from 1.5V of the clock to 1.5V of the signal.
8. Capacitive load for all output timings is 50pF.
9. Data Output timing applies to all signal pins whether tristate I/O or output only.
10. Setup and Hold parameters apply to all signal pins whether tristate I/O or input only.

Boot-Time Interface Parameters:

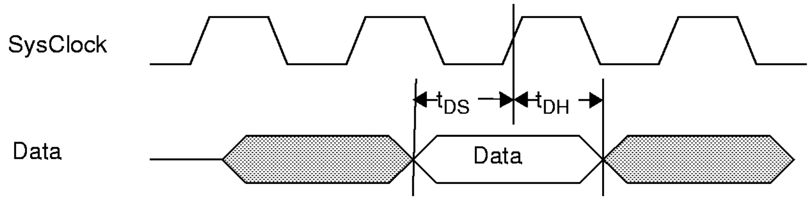
Parameter	Symbol	Test Conditions	133/150/175/200 MHz		Units
			Min	Max	
Mode Data Setup	t_{DS}(M)		4		SysClock cycles
Mode Data Hold	t_{DH}(M)		0		SysClock cycles

TIMING DIAGRAMS:

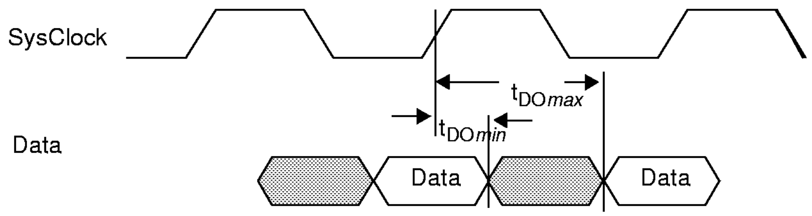


Clock Timing

System Interface Timing (SysAD, SysCMD, ValindIn*, ValidOut*, etc.)



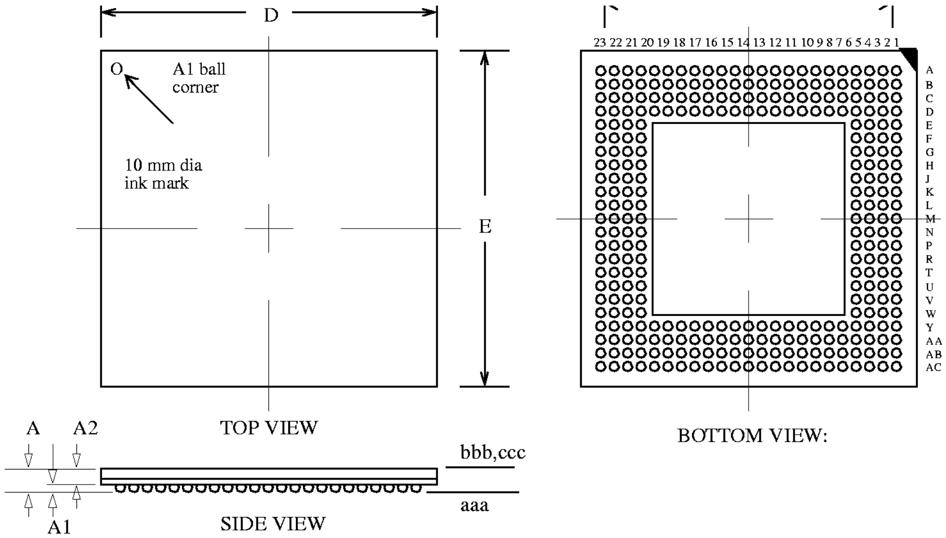
Input Timing



Output Timing

PACKAGING INFORMATION:

304 SBGA Drawing



Body Size	31.0 x 31.0 mm Package			Body Size
Symbol	**Min.**	**Nominal (mm)**	**Max.**	**NOTE**
A	1.41	1.54	1.67	Overall Thickness
A1	0.56	0.63	0.70	Ball Height
A2	0.85	0.91	0.97	Body Thickness
D, E	30.90	31.00	31.10	Body Size
D1, E1	27.84	27.94	28.04	Ball Footprint
M,N	23 x 23			Ball Matrix
M1	4			Number of Rows Deep
b	0.60	0.75	0.90	Ball Diameter
d	0.6			Min Distance Encap to Balls
e	1.27			Ball Pitch
aaa			0.15	Coplanarity
bbb			0.15	Parallel
ccc			0.20	Top Flatness
ddd	0.15	0.33	0.50	Seating Plan Clearance
P	0.20	0.30	0.35	Encapsulation Height
S			0.00	Solder Ball Placement
Deg. C/Watt	0.5			Theta JC
Deg. C/Watt	10-12			Theta JA

RM5270 304 SBGA PACKAGE PINOUT:

(Note: VccIO = VccInt)

Pin	Function	Pin	Function	Pin	Function	Pin	Function
A1	VccIO	A2	VssIO	A3	VssIO	A4	ScLine11
A5	Do Not Connect	A6	VssIO	A7	Do Not Connect	A8	VssIO
A9	SysAD32	A10	SysADC1	A11	Do Not Connect	A12	VssIO
A13	VccInt	A14	VccInt	A15	SysAD63	A16	VssIO
A17	SysAD61	A18	VssIO	A19	NC	A20	ScLine4
A21	VssIO	A22	VssIO	A23	VccIO	B1	VssInt
B2	VccIO	B3	VssInt	B4	VssIO	B5	ScLine10
B6	SysAD35	B7	SysAD34	B8	VccInt	B9	SysAD33
B10	SysADC5	B11	SysADC0	B12	Do Not Connect	B13	SysADC7
B14	SysADC6	B15	Do Not Connect	B16	SysAD30	B17	SysAD29
B18	SysAD28	B19	ScLine5	B20	VssIO	B21	VssInt
B22	VccIO	B23	VssIO	C1	VssIO	C2	VssInt
C3	VccIO	C4	VccIO	C5	Do Not Connect	C6	ScLine9
C7	SysAD3	C8	SysAD2	C9	VccInt	C10	SysAD0
C11	SysADC4	C12	VccInt	C13	SysADC3	C14	SysADC2
C15	SysAD62	C16	VccInt	C17	SysAD60	C18	ScLine6
C19	Do Not Connect	C20	VccIO	C21	VccIO	C22	VssInt
C23	VssIO	D1	ScLine13	D2	VssIO	D3	VccIO
D4	VccIO	D5	VccIO	D6	VccIO	D7	ScLine8
D8	VccInt	D9	VccIO	D10	SysAD1	D11	VccInt
D12	VccIO	D13	VccInt	D14	SysAD31	D15	VccIO
D16	VccInt	D17	ScLine7	D18	VccIO	D19	VccIO
D20	VccIO	D21	VccIO	D22	VssIO	D23	Do Not Connect
E1	VccInt	E2	ScLine14	E3	ScLine12	E4	VccIO
E20	VccIO	E21	Do Not Connect	E22	Do Not Connect	E23	ScLine1
F1	VssIO	F2	Reserved	F3	ScLine15	F4	VccIO
F20	VccIO	F21	ScLine3	F22	ScLinE0	F23	VssIO
G1	SysAD36	G2	SysAD4	G3	Reserved	G4	VccInt
G20	ScLine2	G21	VccInt	G22	SysAD59	G23	SysAD58
H1	VssIO	H2	SysAD37	H3	SysAD5	H4	Do Not Connect
H20	VccInt	H21	SysAD27	H22	SysAD26	H23	VssIO
J1	SysAD7	J2	SysAD6	J3	VccInt	J4	VccIO
J20	VccIO	J21	VccInt	J22	SysAD57	J23	SysAD56
K1	SysAD40	K2	SysAD8	K3	SysAD39	K4	SysAD38
K20	SysAD25	K21	SysAD24	K22	SysAD55	K23	SysAD23
L1	SysAD10	L2	SysAD41	L3	SysAD9	L4	VccInt
L20	VccInt	L21	SysAD54	L22	SysAD22	L23	SysAD53
M1	VssIO	M2	SysAD11	M3	SysAD42	M4	VccIO
M20	VccIO	M21	SysAD52	M22	SysAD21	M23	VssIO
N1	SysAD43	N2	VccInt	N3	SysAD12	N4	SysAD44
N20	SysAD19	N21	SysAD51	N22	VccInt	N23	SysAD20
P1	SysAD13	P2	SysAD45	P3	SysAD14	P4	VccInt
P20	VccInt	P21	SysAD49	P22	SysAD18	P23	SysAD50
R1	SysAD46	R2	SysAD15	R3	SysAD47	R4	VccIO

Pin	Function	Pin	Function	Pin	Function	Pin	Function
R20	VccIO	R21	SysAD16	R22	SysAD48	R23	SysAD17
T1	VssIO	T2	Reserved	T3	Reserved	T4	VccInt
T20	ExtRqst*	T21	VccOK	T22	BigEndian	T23	VssIO
U1	Reserved	U2	VccInt	U3	ModeClock	U4	JTCK
U20	VccInt	U21	NMI*	U22	Reset*	U23	ColdReset*
V1	VssIO	V2	JTDO	V3	JTMS	V4	VccIO
V20	VccIO	V21	Reserved	V22	VccInt	V23	VssIO
W1	JTDI	W2	VccIO	W3	Do Not Connect	W4	VccIO
W20	VccIO	W21	Reserved	W22	Reserved	W23	VccInt
Y1	Do Not Connect	Y2	VssIO	Y3	VccIO	Y4	VccIO
Y5	VccIO	Y6	VccIO	Y7	RdRdy*	Y8	Release*
Y9	VccIO	Y10	ScWord0	Y11	VccInt	Y12	VccIO
Y13	SysCmd5	Y14	VccInt	Y15	VccIO	Y16	VccInt
Y17	Int2*	Y18	VccIO	Y19	VccIO	Y20	VccIO
Y21	VccIO	Y22	VssIO	Y23	Reserved	AA1	VssIO
AA2	VssInt	AA3	VccIO	AA4	VccIO	AA5	Do Not Connect
AA6	ScMatch	AA7	ValidOut*	AA8	SysClock	AA9	VccInt
AA10	Do Not Connect	AA11	Do Not Connect	AA12	SysCmd0	AA13	SysCmd4
AA14	SysCmd8	AA15	ScTCE*	AA16	ScValid	AA17	VccInt
AA18	Int3*	AA19	Do Not Connect	AA20	VccIO	AA21	VccIO
AA22	VssInt	AA23	VssIO	AB1	VssIO	AB2	VccIO
AB3	VssInt	AB4	VssIO	AB5	ModeIn	AB6	ValidIn*
AB7	VccP	AB8	VccInt	AB9	VccInt	AB10	ScCWE0*
AB11	ScDCE0*	AB12	SysCmd1	AB13	SysCmd3	AB14	SysCmd7
AB15	ScClr*	AB16	ScTDE*	AB17	ScDOE*	AB18	Int0*
AB19	Int4*	AB20	VssIO	AB21	VssInt	AB22	VccIO
AB23	VssInt	AC1	VccIO	AC2	VssInt	AC3	VssIO
AC4	NC	AC5	WrRdy*	AC6	VssIO	AC7	VssP
AC8	VssIO	AC9	ScWord1	AC10	ScCWE1*	AC11	ScDCE1*
AC12	VssIO	AC13	SysCmd2	AC14	SysCmd6	AC15	SysCmdP
AC16	VssIO	AC17	ScTOE*	AC18	VssIO	AC19	Int1*
AC20	Int5*	AC21	VssIO	AC22	VssIO	AC23	VccIO

Note: Pins marked Reserved are for RM7000 compatibility. Do not connect to any signal or power planes.

ORDERING INFORMATION:

RM5270 -123 A

- A: Package Type — Q=Power Quad 4 (PQ-4)
- -123: Device Maximum Speed
- RM5270: Device Type

Valid Combinations:

RM5270-150Q
RM5270-200Q

Quantum Effect Design, Inc.
3255-3 Scott Blvd. Suite 200
Santa Clara, CA 95054
(408) 565-0300
(408) 565-0335 (fax)
http://www.qedinc.com

For a complete listing of QED Sales Representatives and Offices, please visit our internet website.