

RM7965A-900UI

900 MHz 64-bit

Microprocessor

Data Sheet

Released

Issue No. 2: March 2010

Legal Information

Copyright

© 2010 PMC-Sierra, Inc. All rights reserved.

The information in this document is proprietary and confidential to PMC-Sierra, Inc., and for its customers' internal use. In any event, no part of this document may be reproduced or redistributed in any form without the express written consent of PMC-Sierra, Inc.

PMC-2100294 (R2)

Disclaimer

None of the information contained in this document constitutes an express or implied warranty by PMC-Sierra, Inc. as to the sufficiency, fitness or suitability for a particular purpose of any such information or the fitness, or suitability for a particular purpose, merchantability, performance, compatibility with other parts or systems, of any of the products of PMC-Sierra, Inc., or any portion thereof, referred to in this document. PMC-Sierra, Inc. expressly disclaims all representations and warranties of any kind regarding the contents or use of the information, including, but not limited to, express and implied warranties of accuracy, completeness, merchantability, fitness for a particular use, or non-infringement.

In no event will PMC-Sierra, Inc. be liable for any direct, indirect, special, incidental or consequential damages, including, but not limited to, lost profits, lost business or lost data resulting from any use of or reliance upon the information, whether or not PMC-Sierra, Inc. has been advised of the possibility of such damage.

Trademarks

For a complete list of PMC-Sierra's trademarks, see our web site at <http://www.pmc-sierra.com/legal/>. Other product and company names mentioned herein may be the trademarks of their respective owners.

Patents

The technology discussed is protected by one or more of the following patent grants.

U.S. Patent Numbers 5,953,748; 5,606,683; 5,760,620; 6,703,950.

Contacting PMC-Sierra

PMC-Sierra
8555 Baxter Place
Burnaby, BC
Canada V5A 4V7

Tel: +1 (604) 415-6000
Fax: +1 (604) 415-6200

Document Information: document@pmc-sierra.com
Corporate Information: info@pmc-sierra.com
Technical Support: apps@pmc-sierra.com
Web Site: <http://www.pmc-sierra.com>

Downloaded [controlled] by Venkatesh Betageri of IHS on Wednesday, 12 January, 2011 02:34:04 AM

Revision History

Issue No.	Issue Date	Details of Change
1	February 2010	Data sheet created.
2	March 2010	Modified Table 28 (Power) and Notes. Change VccInt and VccP to 1.32 V for > 835 MHz operation.

Downloaded [controlled] by Venkatesh Betageri of IHS on Wednesday, 12 January, 2011 2:34:04 AM

Table of Contents

Legal Information.....	2
Copyright.....	2
Disclaimer	2
Trademarks	2
Patents.....	2
Revision History.....	4
List of Figures	8
List of Tables.....	9
1 Definitions	10
2 Introduction	11
2.1 Features	11
3 Block Diagram.....	13
4 E9000 CPU Core	14
4.1 CPU Registers.....	14
4.2 Superscalar Dispatch	15
4.3 Seven-stage Pipeline	16
4.3.1 RM7000 Pipeline Stages.....	16
4.3.2 E9000 Pipeline Stages	17
4.4 Delay slots.....	18
4.4.1 Branch Delay.....	18
4.4.2 Load Delay	18
4.5 Branch Prediction.....	18
4.6 Integer Unit.....	18
4.6.1 Register File	19
4.7 Integer ALU	19
4.8 Integer Multiply/Divide.....	19
4.9 Floating-Point Coprocessor.....	20
4.10 Floating-Point Unit.....	20
4.11 Floating-Point General Register File.....	22
4.12 System Control Coprocessor (CP0).....	22
4.13 System Control Coprocessor Registers.....	22
4.14 Memory Management Unit (MMU).....	23
4.15 Virtual to Physical Address Mapping.....	24
4.16 Joint TLB	25

4.17	Instruction TLB	26
4.18	Data TLB	26
4.19	Interrupt Handling.....	26
4.20	Standby Mode	28
4.21	JTAG Interface	29
4.22	Reset Sequence.....	29
5	Cache Architecture.....	30
5.1	Instruction Cache	30
5.2	Data Cache	31
5.3	Secondary Cache.....	32
5.3.1	Secondary Caching Protocols.....	33
5.3.2	Fast Packet Cache Mode.....	33
5.4	Cache Modes	34
5.5	Cache Attributes.....	35
5.6	Cache Locking	36
5.7	Primary Write Buffer	36
5.8	Data Prefetch	36
5.9	Memory Latencies.....	37
6	System Interface	38
6.1	System Address/Data Bus	38
6.2	System Command Bus	39
6.3	Handshake Signals	40
6.4	System Interface Operation	41
6.5	Write Modes	43
7	Integrated Debug	44
7.1	EJTAG Debugging.....	44
7.2	Trace Buffer.....	44
7.3	Test/Breakpoint Registers	45
7.4	Performance Counters	46
8	Boot-Mode Settings	48
9	RM7000 and RM7965A Differences	53
10	Pin Descriptions	55
11	Absolute Maximum Ratings	59
12	DC Electrical Characteristics	60
13	Power.....	61
13.1	Normal Operating Conditions.....	61

13.2	Power Requirements.....	62
13.3	Typical Power Consumption.....	62
14	AC Electrical Characteristics.....	64
14.1	Capacitive Load Deration.....	64
14.2	Clock Parameters.....	64
14.3	System Interface Parameters.....	64
14.4	Boot-Time Interface Parameters.....	65
15	Timing Diagrams.....	66
15.1	Clock Timing.....	66
15.2	System Interface Timing.....	66
16	Thermal Information.....	67
17	Packaging and Pinout Information.....	68
17.1	256-pin CSBGA Package Diagram.....	68
17.2	256-pin CSBGA Alphanumerical Pinout.....	69
17.3	256-pin CSBGA Alphabetical Pinout.....	71
18	Ordering Information.....	73

List of Figures

Figure 1	Block Diagram.....	13
Figure 2	General Purpose Registers.....	14
Figure 3	Instruction Issue Paradigm.....	15
Figure 4	Pipeline Execution Diagram.....	17
Figure 5	CP0 Registers.....	23
Figure 6	Fast Packet Cache Mode.....	34
Figure 7	Typical Embedded System Block Diagram with 64-bit SysAD Bus.....	38
Figure 8	Processor Block Read.....	41
Figure 9	Processor Block Write.....	42
Figure 10	Multiple Outstanding Reads.....	43
Figure 11	Clock Timing.....	66
Figure 12	Input Timing.....	66
Figure 13	Output Timing.....	66

List of Tables

Table 1	Acronyms and Abbreviations.....	10
Table 2	Instruction Issue Rules	15
Table 3	Dual Issue Instruction Classes.....	16
Table 4	Integer ALU Operations.....	19
Table 5	Integer Multiply/Divide Operations	19
Table 6	Floating Point Latencies and Repeat Rates.....	21
Table 7	Kernel Mode Virtual Addressing (32-bit)	24
Table 8	Cause Register.....	26
Table 9	Interrupt Control Register.....	27
Table 10	IPLLO Register.....	27
Table 11	IPLHI Register	27
Table 12	Interrupt Vector Spacing.....	28
Table 13	E9000 Cache Operating Modes.....	34
Table 14	RM7965A Cache Attributes.....	35
Table 15	Cache Locking Control.....	36
Table 16	On-Chip Memory Latencies.....	37
Table 17	Watch Registers	45
Table 18	Performance Counter Control	46
Table 19	System Interface	55
Table 20	Clock/Control Interface.....	56
Table 21	Power Supply	56
Table 22	Interrupt Interface	57
Table 23	JTAG Interface	57
Table 24	Initialization Interface.....	57
Table 25	(VccIO = 3.15 V – 3.45 V)	60
Table 26	(VccIO = 2.3 V – 2.7 V)	60
Table 27	(VccIO = 1.4 V – 1.6 V) HSTL.....	60
Table 28	Normal Operating Voltages for 0.13 μ m CMOS.....	61
Table 29	VccINT Power Requirements.....	62
Table 30	Conditions for Power Requirements	62
Table 31	Device Compact Model ²	67
Table 32	Heat Sink Requirements	67

1 Definitions

Table 1 defines the abbreviations used in this data sheet.

Table 1 Acronyms and Abbreviations

Acronym or Abbreviation	Description
CPU	Central Processing Unit
CPLD	Complex Programmable Logic Device
DDR	Double Data Rate
DMA	Direct Memory Access
ECC	Error Correction Code
EJTAG	Enhanced Joint Test Action Group
FCRAM	Fast Cycle RAM
FPGA	Field-programmable Gate Array
I/O	Input/Output
LVTTTL	Low-voltage Transistor-Transistor Logic
MIPS	Millions of Instructions Per Second Microprocessor without Interlocked Pipeline Stages
MMU	Memory Management Unit
MOESI	5-State Algorithm for Cache Coherency: Modified/Owned (Modified-Shared)/Exclusive/ Shared/Invalid
NMI	Non-maskable Interrupt
PAL	Programmed Array Logic
PLL	Phase Lock Loop
RAM	Random Access Memory
ROM	Read-only Memory
SDRAM	Synchronous Dynamic RAM
SMP	Symmetric Multi-processing
SSTL	Stub Series Terminated Logic
SysAD	Multiplexed Address/Data System Bus
TAP	Test Access Port
TLB	Translation Lookaside Buffer

2 Introduction

The RM7965A is a high-performance 64-bit microprocessor with features including a seven-stage dual-issue pipeline, tightly coupled L1 and L2 caches, and sophisticated branch prediction for maintaining pipeline efficiency.

A 200 MHz 64-bit multiplexed system address and data bus (SysAD) enables a high-bandwidth I/O interface to a variety of system controllers providing connectivity to a wide range of networking peripherals. The RM7965A also contains a vectored and prioritized interrupt controller for versatile interrupt configurations.

On-chip EJTAG debug modules ensure smooth and easy debugging hardware and software by allowing both single-step and state examination. The inclusion of a pipeline-rate branch instruction trace buffer facilitates debugging under operating conditions.

2.1 Features

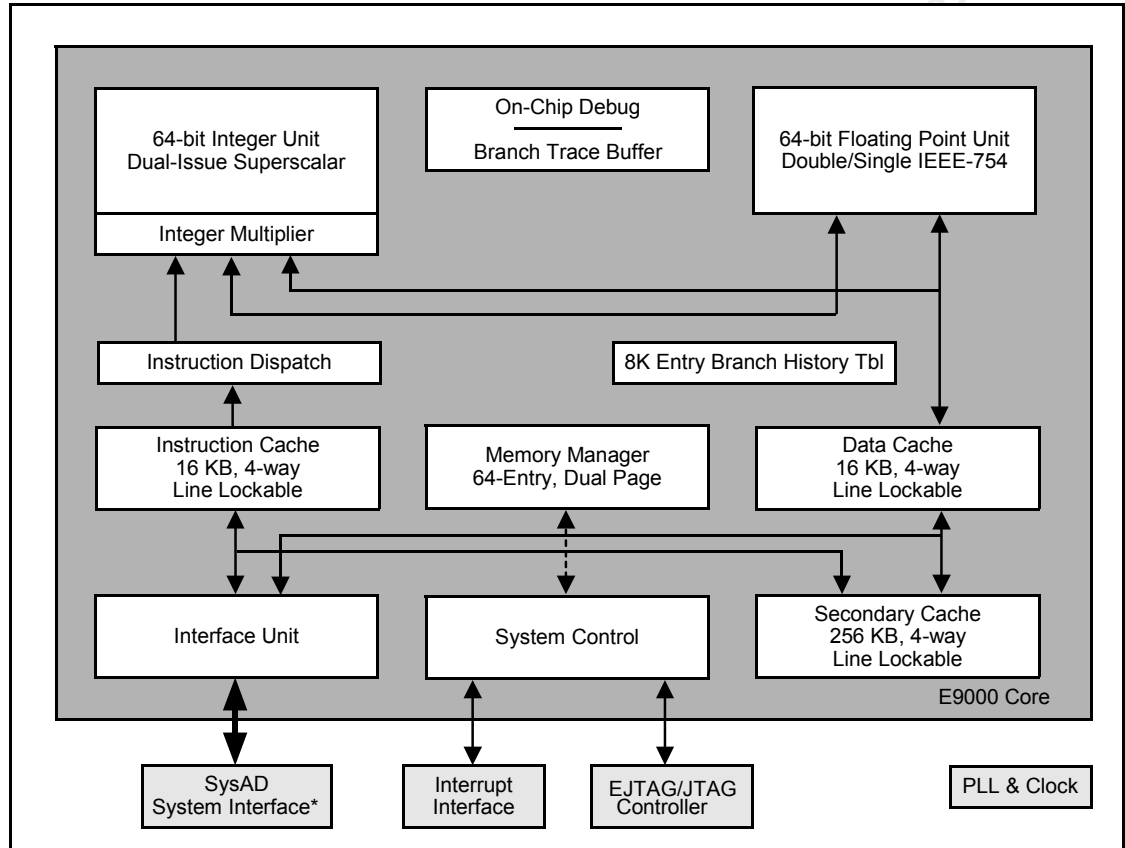
- CPU core with MIPS64-compatible Instruction Set Architecture that features:
 - 900 MHz operation.
 - Dual-issue superscalar 7-stage pipeline.
 - 16-KB, 4-way set associative L1 Instruction cache.
 - 16-KB, 4-way set associative L1 Data cache.
 - 256-KB, 4-way set associative L2 cache with industry best 5-cycle access latency.
 - Error Checking and Correcting (ECC) on L2 cache.
 - Fast Packet Cache to assist processing of packet data.
 - 8K-entry branch prediction table.
 - Fully associative 64-entry TLB with dual pages.
 - High performance Floating Point unit (IEEE 754).
 - Fixed-point DSP instructions such as Multiply/Add, Multiply/Subtract, and 3 Operand Multiply.
- High-performance system interface:
 - Multiple outstanding reads with out of order return.
 - 1600 MB/s peak throughput.
 - 200 MHz maximum frequency using HSTL signaling on the SysAD bus.
 - Multiplexed address/data bus (SysAD) supports 1.5 V, 2.5 V, and 3.3 V I/O logic.
 - Processor clock multipliers 2, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9, 10, 11, 12, 13, 14, 15, 16, 17.
- Integrated on-chip EJTAG controller.
- 64-entry dynamic Trace Buffer for use in real-time trace and debug.
- Two 32-bit virtually addressed Watch registers.
- Integrated performance counters:

- Contains 2 independent 32-bit counters.
- Counts over 30 processor events including mispredicted branches.
- Enables full characterization and analysis of application software.
- 256-pin CSBGA package (27x27 mm)

Downloaded [controlled] by Venkatesh Betageri of IHS on Wednesday, 12 January, 2011 02:34:04 AM

3 Block Diagram

Figure 1 Block Diagram



4 E9000 CPU Core

The RM7965A product consists of the E9000 core plus system interface logic. The E9000 is compatible with the MIPS64 instruction set architecture (ISA), which is a superset of the MIPS IV ISA and is fully backwards compatible with the RM7000 CPU core utilized in all RM70xx products. Also included in the E9000 core is a high performance, IEEE 754 compliant floating-point unit.

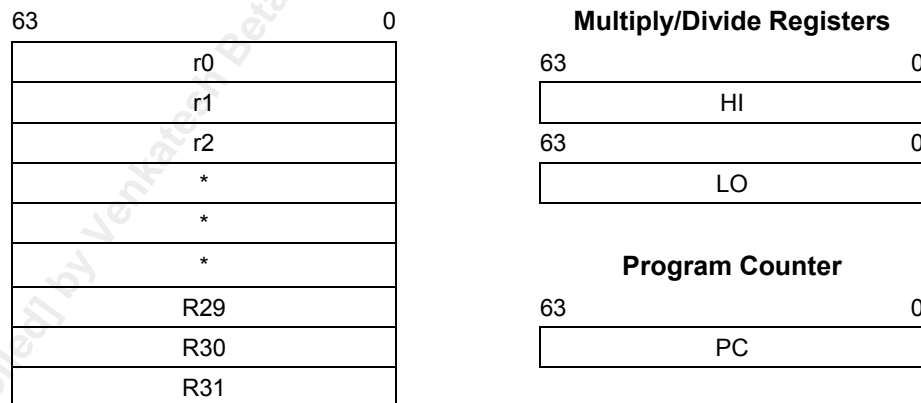
The E9000 core includes a dual-integer superscalar processor with a two level cache hierarchy, an MMU, and a sophisticated branch predictor. Support is provided for two outstanding reads with out-of-order return. The interrupt controller works in conjunction with the system interrupt controller to provide a robust interrupt architecture.

The E9000 core also contains an integrated EJTAG debug module and an integrated Test Access Port (TAP) controller, both of which allow easy debug from the JTAG interface. A 64-entry pipeline-rate trace buffer is included for real-time program flow analysis.

4.1 CPU Registers

The E9000 contains 32 general purpose registers (GPR), two special purpose registers for integer multiplication and division, and a program counter; there are no condition code bits. Figure 2 shows these processor registers. The E9000 also includes two sets of CP0 registers. The CP0 register sets contain both 32 and 64-bit registers. Only 29 of the 32 registers specified in CP0 Set 0 are implemented, and only 5 of the 32 registers in CP0 Set 1 are implemented.

Figure 2 General Purpose Registers



4.2 Superscalar Dispatch

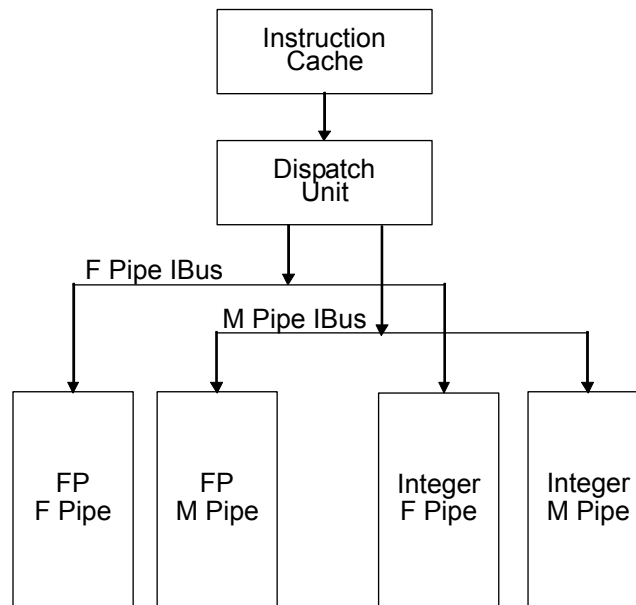
The E9000 incorporates a superscalar dispatch unit that allows it to issue up to two instructions per cycle. For purposes of instruction issue, the E9000 defines four classes of instructions: integer, load/store, branches, and floating-point. There are two logical pipelines, the *function*, or F, pipeline and the *memory*, or M, pipeline. Note that the M pipe can execute integer as well as memory type instructions.

Table 2 Instruction Issue Rules

F Pipe	M Pipe
one of: Integer ALU, branch, floating-point, integer mul, div	one of: Integer ALU, load/store

Figure 3 is a simplification of the execution unit, and illustrates the basics of the instruction issue mechanism.

Figure 3 Instruction Issue Paradigm



The figure illustrates that one F pipe instruction and one M pipe instruction can be issued concurrently but that two M pipe or two F pipe instructions cannot be issued. Table 3 specifies more completely the instructions within each class.

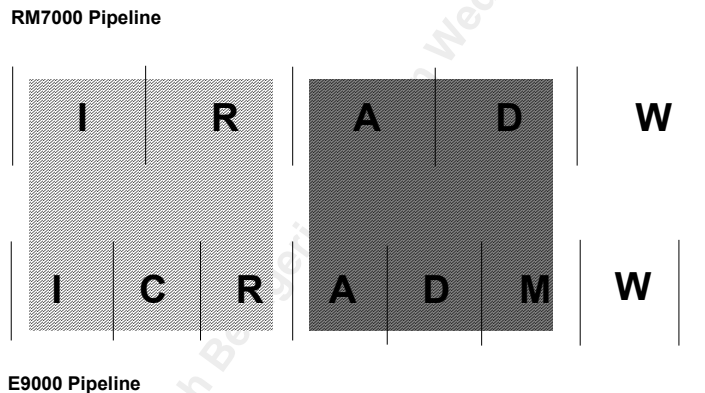
Table 3 Dual Issue Instruction Classes

Integer ALU	Load/Store	Floating- Point	Branch	Integer Mul/Div
add, sub, or, xor, shift, etc.	lw, sw, ld, sd, ldc1, sdc1, mov, movc, fmov, etc.	fadd, fsub, fmult, fmadd, fdiv, fcmp, fsqrt, etc.	beq, bne, bCzT, bCzF, j, etc.	mult, multu, mad, madu, mul, dmult, dmultu, div, divd, ddiv, ddivd

4.3 Seven-stage Pipeline

The E9000 pipeline has been increased to 7 stages versus the 5-stage RM7000 pipeline. Increasing the pipeline to 7 stages and including branch prediction allows the frequency to be increased beyond 800 MHz while maintaining high pipeline efficiency. Figure 3 illustrates the 7-stage pipeline in comparison to the 5-stage pipeline of the RM7000.

Figure 3 Pipeline Comparison



4.3.1 RM7000 Pipeline Stages

The RM7000 pipeline stages are summarized as follows:

- I: Instruction Fetch from instruction cache
- R: Register File Access
- A: Instruction Execution
- D: Data Fetch from data cache
- W: Write Back to register file

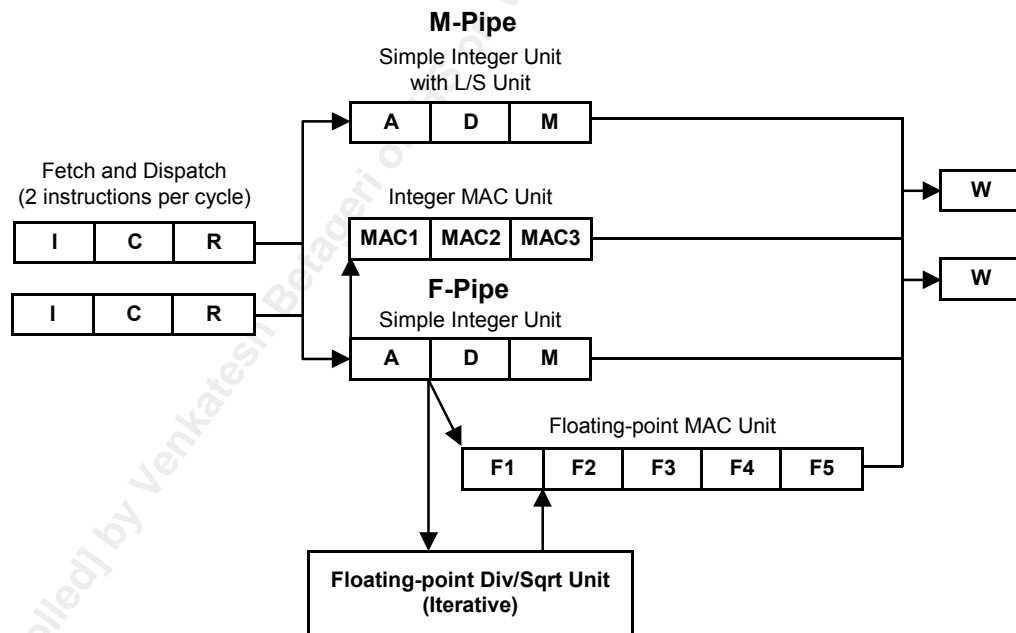
4.3.2 E9000 Pipeline Stages

In contrast to the RM7000 pipeline, the E9000 pipeline has two additional stages to allow an extra clock cycle of for both the instruction and the data pipeline regimes. The E9000 pipeline stages can be summarized as follows:

- I: Instruction Addressing
- C: Instruction Cache Access
- R: Register File Access, Instruction Decode
- A: Instruction Execution, Data Address Calculation
- D: Data Cache Access
- M: Data Bus, Data Alignment
- W: Write Back to register file

The pipeline execution diagram for the E9000 is shown below:

Figure 4 Pipeline Execution Diagram



4.4 Delay slots

The intrinsic branch and load delays are each increased by 1 in the E9000 due to the increase in pipeline length.

4.4.1 Branch Delay

The branch delay slot increases from one to two, but with branch prediction, which has been simulated to predict accurately ~95% of the time, the effective branch delay stays about one. The second, or additional, branch delay slot is hidden to the code and is taken as a one-cycle stall in the case where the branch prediction misses. When the branch prediction hits, this second slot is taken with the first instruction of the branch target code.

4.4.2 Load Delay

In the E9000, the load delay slot is increased from one to two. Compilers optimized for the E9000 are able to fill the extra delay slot with non-data dependent instructions. Even code that has not been recompiled, however, will perform nearly optimally on the E9000 core.

4.5 Branch Prediction

The E9000 has an 8K entry branch prediction table, utilizing a correlative branch prediction algorithm which increases the accuracy of prediction to greater than 95%. The correlative algorithm hashes the lower address bits with bits of dynamic prediction from all branches to derive the index for the branch entry. Using this approach a given branch instruction can have a predictor for its “inner” loop and a separate predictor for its “outer” loop.

4.6 Integer Unit

The E9000 implements the MIPS64 Instruction Set Architecture including five implementation specific instructions not found in the baseline MIPS IV ISA, but which are useful for embedded applications. These instructions are integer multiply-add (MAD), multiply-add unsigned (MADU), multiply-subtract (MSUB), multiply-subtract unsigned (MSUBU), and three-operand integer multiply (MUL).

Another instruction new to the E9000 is the Superscalar No-Operation (SSNOP) instruction. This instruction issues a NOP instruction to each integer unit.

The E9000 integer unit includes 32 general-purpose 64-bit registers, the *HI/LO* result registers for two-operand integer multiply/divide operations, and the program counter (PC). There are two separate execution units: one that can execute function (F) pipe instructions and one that can execute memory (M) pipe instructions. Refer to Table 4 for the instruction issue rules.

Note that integer multiply/divide instructions, as well as their corresponding MFHi and MFLo instructions, can only be executed in the F pipe execution unit. Within each execution unit, the operational characteristics are the same as on previous MIPS designs with single cycle ALU operations (add, sub, logical, shift), one cycle load delay, and an autonomous multiply/divide unit.

4.6.1 Register File

The E9000 has 32 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. In order to service the two integer execution units, the register file has four read ports and two write ports and is fully bypassed both within and between the two execution units to minimize operation latency in the pipeline.

4.7 Integer ALU

The E9000 has two complete integer ALUs each consisting of an integer adder/subtractor, a logic unit, and a shifter. Table 4 shows the functions performed by the ALUs for each execution unit. Each of these units is optimized to perform all operations in a single processor cycle.

Table 4 Integer ALU Operations

Unit	F Pipe	M Pipe
Adder	add, sub	add, sub, data address add
Logic	logic, moves, zero shifts (nop)	logic, moves, zero shifts (nop)
Shifter	non-zero shift	non-zero shift, store align

4.8 Integer Multiply/Divide

The E9000 has a single dedicated integer multiply/divide unit optimized for high-speed multiply and multiply-accumulate operations. The multiply/divide unit resides in the F pipe execution unit. Table 5 shows the performance of the multiply/divide unit on each operation.

Table 5 Integer Multiply/Divide Operations

Opcode	Operand Size	Latency	Repeat Rate	Stall Cycles
MULT/U, MAD/U	16 bit	4	3	0
	32 bit	5	4	0
MUL	16 bit	4	3	2
	32 bit	5	4	3
DMULT, DMULTU	any	9	8	0
DIV, DIVD	any	36	36	0
DDIV, DDIVU	any	68	68	0
MSUB, MSUBU				

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 E9000 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.

The multiply-add instructions, MAD and MADU, multiply two operands and add the resulting product to the current contents of the *Hi* and *Lo* registers. The multiply-accumulate operation is the core primitive of almost all digital signal processing algorithms. Therefore, using the E9000 eliminates the need for a separate DSP in many embedded applications.

The multiply-sub instructions, MSUB and MSUBU, multiply two operands and subtract the resulting product from the current contents of the *Hi* and *Lo* registers. The multiply-subtract operation is a core primitive of digital signal processing algorithms.

4.9 Floating-Point Coprocessor

The E9000 incorporates a high-performance fully pipelined floating-point coprocessor that 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 M pipe of the integer unit. The superscalar capabilities of the E9000 allow floating-point computation instructions to issue concurrently with integer instructions.

4.10 Floating-Point Unit

The E9000 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 E9000 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 6 gives the latencies of the floating-point instructions in internal processor cycles.

Table 6 Floating Point Latencies and Repeat Rates

Operation	Latency single/double	Repeat Rate single/double
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, fmovc	1	1
fabs, fneg	1	1

4.11 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 coprocessor load or store doubleword instruction in every cycle.

The floating-point control register file contains two registers; one for determining configuration and revision information for the coprocessor, and one for control and status information. These registers are primarily used for diagnostic software, exception handling, state saving and restoring, and control of rounding modes.

To support superscalar operations 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 for concurrent floating-point load or store and conditional move operations.

4.12 System Control Coprocessor (CP0)

The system control coprocessor (CP0) is responsible for the virtual memory sub-system, the exception control system, and the diagnostics capability of the processor.

For memory management support, the E9000 CP0 is logically identical to the CPU cores used in the RM5200 Family and the RM7000 Family. For interrupt exceptions and diagnostics, the E9000 is a superset of the RM5200 Family and the RM7000 Family, implementing additional features described in the following sections on Interrupts, Test/ Breakpoint registers, and Performance Counters.

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 TLB (JTLB), and coprocessor registers used by the virtual memory mapping sub-system.

4.13 System Control Coprocessor Registers

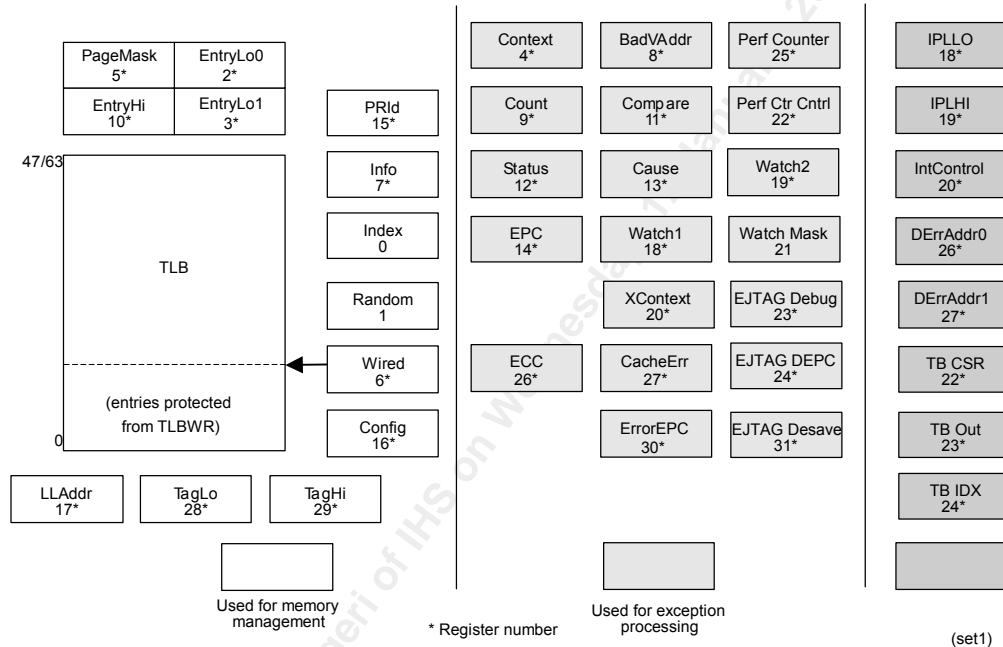
The E9000 incorporates all CP0 registers internally. These registers provide the path through which the virtual memory system's page mapping is examined and modified, exceptions are handled, and operating modes are controlled (kernel vs. user mode, interrupts enabled or disabled, cache features). In addition, the E9000 includes registers to implement a real-time cycle counting facility, to aid in cache and system diagnostics, and to assist in data error detection.

To support the non-blocking caches and enhanced interrupt handling capabilities of the E9000, both the data and control register spaces of CP0 are supported. In the data register space, which is accessed using the MFC0 and MTC0 instructions, the E9000 supports the same registers as found in previous RM7000 processors plus three new registers to support EJTAG Debugging. The three new registers are called: *EJTAG Debug*, *EJTAG DEPC*, and *EJTAG DESave*.

In the control space, the E9000 supports three new registers to support the 64-entry branch Trace Buffer: *Trace Buffer Control and Status (TB CSR)*, *Trace Buffer Out (TB Out)*, and *Trace Buffer Index (TB IDX)*. See Section 7.1

Figure 5 shows the CP0 registers.

Figure 5 CP0 Registers



4.14 Memory Management Unit (MMU)

The E9000 has an MMU with a 64 entry TLB, with each entry having dual pages for a total of 128 pages. The page size is programmable to be 4 KB, 16 KB, 64 KB, 256 KB, 1 MB, 16 MB, 64 MB, or 256 MB. Pages can be programmed to be write-protected. The TLB can operate statically or in a demand-paged environment, with TLB misses generating exceptions to load the appropriate page. The TLB replacement algorithm is random, and there is a TLB fence that can be used to lock a subset of the TLB entries, and allow the remainder to be dynamically refilled. The MMU architecture on the E9000 supports both 32 and 64-bit virtual addressing.

4.15 Virtual to Physical Address Mapping

The E9000 provides three modes of virtual addressing:

- user mode
- kernel mode
- supervisor mode

These modes allow system software to provide a secure environment for user processes. Bits in the CP0 *Status* register determine which virtual addressing mode is used. In user mode, the E9000 provides a single, uniform virtual address space of 256 GB (2 GB in 32-bit mode).

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

The E9000 core also supports a supervisor mode in which the virtual address space is 256.5 GB (2.5 GB in 32-bit mode), divided into three regions based on the high-order bits of the virtual address. Figure shows the address space layout for 32-bit operations.

Table 7 Kernel Mode Virtual Addressing (32-bit)

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

When the E9000 is configured for 64-bit addressing, the virtual address space layout is an upward compatible extension of the 32-bit virtual address space layout.

4.16 Joint TLB

For fast virtual-to-physical address translation, the E9000 uses a large, fully associative TLB that maps virtual pages to their corresponding physical addresses. As indicated by its name, the JTLB is used for both instruction and data translations. The JTLB is organized as pairs of even/odd entries, and maps a virtual address and address space identifier (ASID) into the large, 64 GB physical address space. By default, the JTLB is configured as 48 pairs of even/odd entries. The optional 64-even/odd-entry configuration is set at boot time.

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 4 KB to 16 MB (in 4x multiples). The CP0 *PageMask* 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 E9000 provides a random replacement algorithm to select a TLB entry to be written with a new mapping. The core also provides a mechanism whereby a system specific number of mappings can be locked into the TLB, thereby avoiding random replacement. This mechanism uses the CP0 *Wired* register and allows the operating system to guarantee that certain pages are always mapped for performance reasons and to avoid a deadlock condition. It 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
- write-back
- write-through with write-allocate
- write-through without write-allocate
- write-back with secondary and tertiary bypass

Note that both of the write-through protocols bypass both the secondary and the tertiary caches since neither of these caches support writes of less than a complete cache line.

These protocols are used for both code and data in the E9000, 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 RM7000 and RM5200 Families.

4.17 Instruction TLB

The E9000 uses a 4-entry instruction TLB (ITLB). The ITLB offers the following advantages:

- Minimizes contention for the JTLB
- Eliminates the critical path of translating through a large associative array
- Allows instruction address and data address translations to occur in parallel
- Saves power

Each ITLB entry maps a 4 KB 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.

4.18 Data TLB

The E9000 uses a 4-entry data TLB (DTLB) for the same reasons cited above for the ITLB. Each DTLB entry maps a 4 KB 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, 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.

4.19 Interrupt Handling

In order to provide better real time interrupt handling, the RM7965A provides 10 external hardware interrupts, each of which can be separately prioritized and separately vectored.

The performance counter is also a hardware interrupt source using **INT13**. Historically in the MIPS architecture, interrupt 7 (**INT7**) was used as the Timer Interrupt. The RM7965A provides a separate interrupt, **INT12**, for this purpose, thereby releasing **INT7** for use as a pure external interrupt.

All interrupts (**INT[13:0]**), the Performance Counter, and the Timer, have corresponding interrupt mask bits, **IM[13:0]**, and interrupt pending bits, **IP[13:0]**, in the *Status*, *Interrupt Control*, and *Cause* registers. The bit assignments for the *Interrupt Control* and *Cause* registers are shown in Table 8 and Table 9. (Note the *Status* register has not changed from the RM5200 product family and is not shown.)

Table 8 Cause Register

31	30	29:28	27	26	25	24	23:8	7	6:2	0:1
BD	0	CE	0	W2	W1	IV	IP[15:0]	0	EXC	0

Table 9 Interrupt Control Register

31:16	15:8	7	6:5	4:0
0	IM[15:8]	TE	0	Spacing

The **IV** bit in the *Cause* register is the global enable bit for the enhanced interrupt features. If this bit is clear then interrupt operation is compatible with RM5200 and RM7000 products.

In the *Interrupt Control* register, the interrupt vector spacing is controlled by the **Spacing** field as described below. The **Interrupt Mask** field (**IM[13:8]**) contains the interrupt mask for interrupts 8 through 13. **IM[15:14]** are reserved for future use.

The **Timer Enable (TE)** bit is used to gate the Timer Interrupt to the *Cause* Register. If **TE** is set to 0, the Timer Interrupt is not gated to **IP12**. If **TE** is set to 1, the Timer Interrupt is gated to **IP12**.

The setting for Mode Bit 11 is used to determine if the Timer Interrupt replaces the external interrupt (**INT5***) as an input to **IP7** in the *Cause* Register. If Mode Bit 11 is set to 0, the Timer Interrupt is gated to **IP7**. If Mode Bit 11 is set to 1, the external **INT5*** is gated to **IP7**.

In order to utilize both the external Interrupt (**INT5***) and the internal Timer Interrupt, Mode Bit 11 must be set to 1, and **TE** must be set to 1. In this case, the Timer Interrupt will utilize **IP12**, and **INT5*** will utilize **IP7**. Please also reference the logic diagram for interrupt signals in the RM7965A User Manual.

The *Interrupt Control* register uses **IM13** to enable the Performance Counter interrupt and to enable the Trace Buffer interrupt.

Priority of the interrupts is set via two new coprocessor 0 registers called *Interrupt Priority Level Lo (IPLLO)* and *Interrupt Priority Level Hi (IPLHI)*.

In the *IPLLO* and *IPLHI* registers, each interrupt is represented by a four-bit field, thereby allowing each interrupt to be programmed with a priority level from 0 to 15 inclusive. The priorities can be set in any manner, including having all the priorities set exactly the same. Priority 0 is the highest level and priority 15 the lowest. The format of the priority level registers is shown in Table 10 and Table 11. The priority level registers are located in the coprocessor 0 control register space.

Table 10 IPLLO Register

31:28	27:24	23:20	19:16	15:12	11:8	7:4	3:0
IPL7	IPL6	IPL5	IPL4	IPL3	IPL2	IPL1	IPL0

Table 11 IPLHI Register

31:28	27:24	23:20	19:16	15:12	11:8	7:4	3:0
0	0	IPL13	IPL12	IPL11	IPL10	IPL9	IPL8

In addition to programmable priority levels, the RM7965A also permits the spacing between interrupt vectors to be programmed. For example, the minimum spacing between two adjacent vectors is 0x20 while the maximum is 0x200. This programmability allows the user to either set up the vectors as jumps to the actual interrupt service routines or, if interrupt latency is not paramount, to include the entire interrupt service routine at one vector. Table 12 illustrates the complete set of vector spacing selections along with the coding as required in the *Interrupt Control* register bits [4:0], *ICR*.

In general, the active interrupt priority, combined with the spacing setting, generates a vector offset, which is then added to the interrupt base address of 0x200 to generate the interrupt exception offset. This offset is then added to the exception base to produce the final interrupt vector address.

Table 12 Interrupt Vector Spacing

ICR[4:0]	Spacing
0x0	0x000
0x1	0x020
0x2	0x040
0x4	0x080
0x8	0x100
0x10	0x200
others	reserved

4.20 Standby Mode

The RM7965A provides a means to reduce the amount of power consumed by the internal core when the CPU is not 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. If the **SysAD** bus is currently idle when the WAIT instruction completes the W pipe stage, 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[9:0]***, **NMI***, **ExtReq***, **Reset***, and **ColdReset*** 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, causes 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.

4.21 JTAG Interface

The RM7965A interface supports JTAG boundary scan in conformance with IEEE 1149.1. The JTAG interface is useful for checking the integrity of the processor's pin connections.

4.22 Reset Sequence

The RM7965A uses the same reset interface that is used on the RM7000 and RM5200 product families. This single reset interface is used to reset the entire system.

Both power on reset and cold reset completely initialize the RM7965A. The configuration mode bit stream is read into the device to configure the E9000 core and the external bus interface (see Section 8). The configuration stream is read in when the **VccOK** input signal has been asserted and **ColdReset*** and **Reset*** remain asserted. During this time, the PLL is achieving lock.

5 Cache Architecture

The E9000 cache architecture is similar to that of the RM7000. Each core contains 16-KBytes of instruction cache, 16 KB of data cache, and 256 KB of unified secondary cache. The instruction cache, data cache and secondary cache are all four-way set associative. Cache locking is supported for all of the caches, and the caches can be locked with line granularity. This is very useful for keeping frequently called routines in the cache, along with frequently accessed data structures such as look-up tables for routing and other data communications applications. The E9000 data cache is non-blocking, and the pipeline will not stall until a third cache-miss or a data dependency is encountered.

Each primary cache has a 64-bit read path and a 128-bit write path. Both caches can be accessed simultaneously. The primary caches provide the integer and floating-point units with an aggregate bandwidth of 14.4 GB/s at an internal clock frequency exceeding 800 MHz. During an instruction or data primary cache refill, the secondary cache can provide a 64-bit datum every cycle following an initial **five-cycle** latency, for a peak bandwidth of 7.2 GB/s.

5.1 Instruction Cache

The integrated 16 KB, four-way set associative instruction cache in the E9000 is virtually indexed and physically tagged. The effective physical index eliminates the potential for virtual aliases in the cache.

The data array portion of the instruction cache is 64 bits wide and protected by word parity while the tag array holds a 24-bit physical address, 14 control bits, a valid bit, and a single parity bit.

By accessing 64 bits per cycle, the instruction cache is able to supply two instructions per cycle to the superscalar dispatch unit. For signal processing, graphics, and other numerical 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 is consumed by instruction issue. For typical integer code mixes, where instruction dependencies and other resource constraints restrict the level of parallelism that can be achieved, the extra instruction cache bandwidth is used to fetch both the taken and non-taken branch paths to minimize the overall penalty for branches.

A 32-byte (8 instruction) line size is used to maximize the communication efficiency between the instruction cache and the secondary cache, tertiary cache, or memory system.

The E9000 supports cache locking on a per line basis. The contents of each line of the cache can be locked by setting a bit in the Tag RAM. Locking the line prevents its contents from being overwritten by a subsequent cache miss. Refills occur only into unlocked cache lines. This mechanism allows the programmer to lock critical code into the cache, thereby guaranteeing deterministic behavior for the locked code sequence.

5.2 Data Cache

The E9000 has an integrated 16 KB, four-way set associative data cache that is virtually indexed and physically tagged. Line size is 32-bytes (8 words). The effective physical index eliminates the potential for virtual aliases in the cache.

The data cache is non-blocking; that is, a miss in the data cache does not necessarily stall the processor pipeline. As long as no instruction is encountered that is dependent on the data reference that caused the miss, the pipeline continues to advance. Once there are two cache misses outstanding, the processor stalls if it encounters another load or store instruction.

The data array portion of the data cache is 64 bits wide and protected by byte parity while the tag array holds a 24-bit physical address, 3 control bits, a 2-bit cache state field, and 2 parity bits.

The most commonly used 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 as follows:

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 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 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, after which the write is performed as above.

3. Write-through with write allocate

Loads and instruction fetches 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 or tertiary caches. On data store operations, the cache is first searched to determine if the target address is cache resident. If it is resident, the primary cache contents are updated and main memory is written, leaving the write-back bit of the cache line unchanged; no writes occur to the secondary or tertiary caches. If the cache lookup misses, the target line is first brought into the cache, after which the write is performed as above.

4. Write-through without write allocate

Loads and instruction fetches 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 or tertiary caches. 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; no writes occur to the secondary or tertiary caches. If the cache lookup misses, only main memory is written.

5. Fast Packet Cache™ (Write-back with secondary and tertiary bypass)

Loads and instruction fetches first search the primary cache, reading from memory only if the desired data is not resident; the secondary and tertiary caches are not searched. On data store operations, the primary cache is first searched to determine if the target address is resident. If it is resident, the cache contents are updated, and the cache line marked for later write-back. If the cache lookup misses, the target line is first brought into the cache, after which the write is performed as above.

Associated with the data cache is the *store queue*. When the E9000 executes a store instruction, this multi-entry queue is 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 queue allows the E9000 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.

5.3 Secondary Cache

The E9000 has an integrated 256 KB, four-way set associative, and block write-back secondary cache. The secondary cache has a 32-byte line size, a 64-bit bus width to match the system interface and primary cache bus widths, and is protected with the same Error Checking and Correcting (ECC) mechanism used in the R4000 processor. The secondary cache tag array holds a 20-bit physical address, two control bits, a 3-bit cache state field, and two parity bits.

By integrating a secondary cache, the E9000 is able to decrease the latency of a primary cache miss without significantly increasing the number of pins and the amount of power required by the processor. From a technology point of view, integrating a secondary cache leverages CMOS technology by using silicon to build the structures that are most amenable to silicon technology; building very dense, low power memory arrays rather than large power hungry I/O buffers.

Further benefits of an integrated secondary cache are flexibility in the cache organization and management policies that are not practical with an external cache. Two previously mentioned examples are the 4-way associativity and write-back cache protocol.

A third management policy for which integration affords flexibility is cache hierarchy management. With multiple levels of cache, it is necessary to specify a policy for dealing with cases where two cache lines at level n of the hierarchy could possibly be sharing an entry in level $n+1$ of the hierarchy.

The E9000 allows entries to be stored in the primary caches that do not necessarily have a corresponding entry in the secondary. The E9000 does not force the primaries to be a subset of the secondary. For example, if primary cache line A is being filled and a cache line already exists in the secondary for primary cache line B at the location where primary A's line would reside, then that secondary entry is replaced by an entry corresponding to primary cache line A and no action occurs in the primary for cache line B. This operation creates the aforementioned scenario where the primary cache line, which initially had a corresponding secondary entry, no longer has such an entry. Such a primary line is called an *orphan*. In general, cache lines at level $n+1$ of the hierarchy are called *parents* of level n 's *children*.

Another E9000 cache management optimization occurs for the case of a secondary cache line replacement where the secondary line is dirty and has a corresponding dirty line in the primary. In this case, since it is permissible to leave the dirty line in the primary, it is not necessary to write the secondary line back to main memory. Taking this scenario one step further, a final optimization occurs when the aforementioned dirty primary line is replaced by another line and must be written back. In this case, it is written directly to memory, bypassing the secondary cache.

5.3.1 Secondary Caching Protocols

Unlike the primary data cache, the secondary cache supports only block write-back. As noted earlier, cache lines managed with either of the write-through protocols are not placed in the secondary cache. A new caching attribute, *write-back with secondary and tertiary bypass*, allows the secondary, and tertiary caches to be bypassed entirely. When this attribute is selected, the secondary and tertiary caches are not filled on load misses and are not written on dirty write-backs from the primary cache.

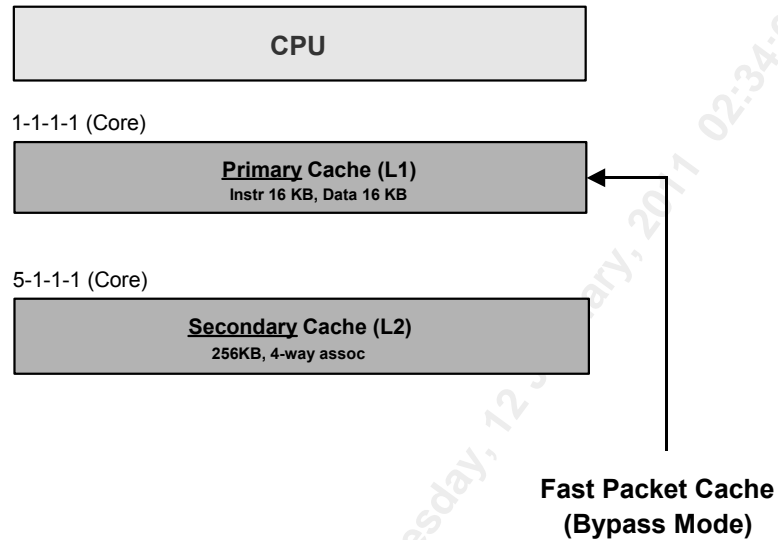
5.3.2 Fast Packet Cache Mode

It is possible to bypass the secondary cache using the Fast Packet Cache feature. Fast Packet Cache can be activated on a per page basis, and allows all accesses into cache, and all write-backs to use only the primary data cache. This is useful for manipulating transient packet data and headers without evicting other less transient data from the L2 cache.

Figure 6 illustrates the two level cache hierarchy and shows the tight coupling of the primary and secondary caches. The primary cache accesses occur at the core frequency.

If there is a primary miss that hits in secondary, then a 5-cycle miss penalty occurs. This latency is best in class for a processor in this performance range, and helps optimize the E9000 core for the highest possible performance.

Figure 6 Fast Packet Cache Mode



5.4 Cache Modes

Table 13 summarizes the E9000 cache operating modes. The coherency attributes referred to in Table 13 are written into the TLB entry to program the coherency attribute for that page.

Table 13 E9000 Cache Operating Modes

Cache Coherency Attribute	Read miss to MM	Store Hit	Store Miss	L2
000 : Write-through No Allocate	Fill L1	Store L1 and MM	Store to MM	Receives L1 displacements
001 : Write-through with Allocate	Fill L1	Store L1 and MM	Store to L1 and MM; Fill L1	Receives L1 displacements
010 : Uncached blocking; Uncached. Reads stall pipeline. Strong ordering enforced. Loads and stores complete in program order	-	-	-	-
011 : Writeback	Fill L1 and L2	Store L1	Store Miss, Hit L2: Read L2->L1 Store L1 Store Miss L1, L2: Read MM->L1, L2, Store L1	Receives L1 Displacements
100 : Reserve	-	-	-	-
101 : Reserve	-	-	-	-

Cache Coherency Attribute	Read miss to MM	Store Hit	Store Miss	L2
110: Uncached Non-Blocking; Uncached. Reads do not stall pipeline unless a data dependency exists. Strong ordering not enforced, therefore loads can be completed out of program order	-	-	-	-
111: Bypass (Fast Packet Cache); Bypass L2	Fill L1	Store L1	Fill L1, Store L1	Bypassed

5.5 Cache Attributes

The RM7965A cache attributes for the instruction, data and internal secondary caches are summarized in Table 14.

Table 14 RM7965A Cache Attributes

Attribute	Primary Instruction	Primary Data	On-chip Secondary
Size	16 KB	16 KB	256 KB
Associativity	4-way	4-way	4-way
Replacement Algorithm	cyclic	cyclic	cyclic
Line size	32 byte	32 byte	32 byte
Index	vAddr _{11..0}	vAddr _{11..0}	pAddr _{15..0}
Tag	pAddr _{35..12}	pAddr _{35..12}	pAddr _{35..16}
Write policy	N/A	write-back, write-through	block write-back, bypass
Read policy	N/A	non-blocking (2 outstanding)	non-blocking (data only, 2 outstanding)
Read order	critical word first	critical word first	critical word first
Write order	N/A	sequential	sequential
Miss restart following	complete line	first double (if waiting for data)	N/A
Protection	per word parity	per byte parity	8-bit ECC per DW

5.6 Cache Locking

The E9000 core in the RM7965A product allows critical code or data fragments to be locked into the primary and secondary caches. The user has complete control over the locking function. For instruction and data fragments in the primary caches, locking is accomplished by setting either or both of the cache lock enable bits and specifying the set in the CP0 ECC register, then executing either a load instruction for data, or a Fill_I cache operation for instructions.

Only cache lines within sets A and B of each cache can be locked. Locking within the secondary works identically to the primaries using a separate secondary lock enable bit and the same set selection field. As with the primaries, only sets A and B can be locked. Table 15 summarizes the cache locking capabilities.

Table 15 Cache Locking Control

Cache	Lock Enable	Set Select	Activate
Primary I	ECC[27]	ECC[28]=0A ECC[28]=1B	Fill_I
Primary D	ECC[26]	ECC[28]=0A ECC[28]=1B	Load/Store
Secondary	ECC[25]	ECC[28]=0A ECC[28]=1B	Fill_I or Load/Store

5.7 Primary Write Buffer

Writes to secondary cache or external memory, whether cache miss write-backs or stores to uncached or write-through addresses, use the integrated primary 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.

5.8 Data Prefetch

The E9000 supports the MIPS IV integer data prefetch (PREF) and floating-point data prefetch (PREFX) instructions. These instructions are used by the compiler or by an assembly language programmer when it is known or suspected that an upcoming data reference is going to miss in the cache. By appropriately placing a prefetch instruction, the memory latency can be hidden under the execution of other instructions. In cases where the execution of a prefetch instruction would cause a memory management or address error exception the prefetch is treated as a NOP.

The “Hint” field of the data prefetch instruction is used to specify the action taken by the instruction. The instruction can operate normally (that is, fetching data as if for a load operation) or it can allocate and fill a cache line with zeroes on a primary data cache miss.

5.9 Memory Latencies

Table 16 is a compilation of latencies for the different types of on-chip memory accesses for the E9000. Local cache accesses to the L1 occur at the CPU core frequency, and local L1 misses access L2 with a 5-cycle miss penalty.

Table 16 On-Chip Memory Latencies

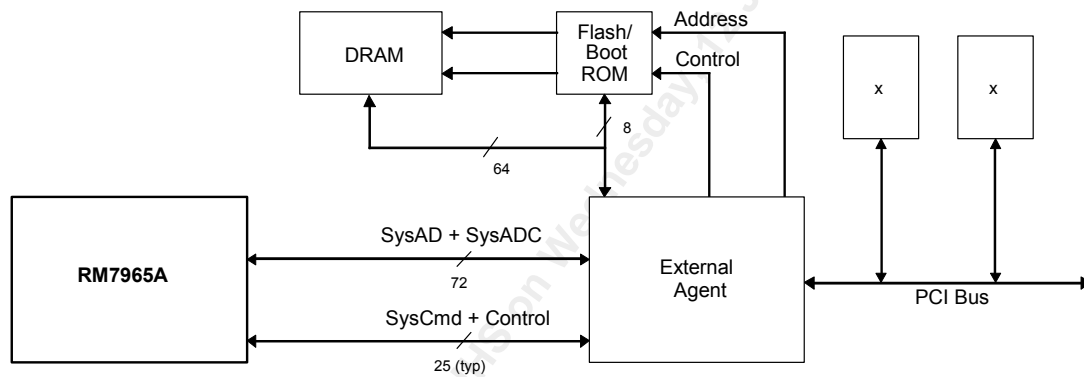
Type of Burst Memory Access	Number of Processor Clocks per Double Word
Local L1 Hit	1-1-1-1
Local L2 Hit	5-1-1-1

6 System Interface

The RM7965A product provides a high performance system interface comprised of a multiplexed address/data bus (**SysAD**), a parity check bus (**SysADC**) and a system command bus (**SysCmd**). The **SysAD** is 64-bits, **SysADC** is 8-bits, and **SysCmd** is 9-bits.

Figure 7 shows a typical embedded system using the RM7965A. The diagram shows a system with a bank of DRAMs, and an external agent or ASIC which provides DRAM control and I/O functionality.

Figure 7 Typical Embedded System Block Diagram with 64-bit SysAD Bus



There are many companion chips or system controllers that interface to the **SysAD** bus that provide connectivity to a variety of interfaces including PCI, PCI-X, Fast Ethernet, Gigabit Ethernet, and T1/T3. They typically include a boot bus that connects to flash or ROM memory, which can be utilized to boot any RM7965A CPU across the SysAD bus.

6.1 System Address/Data Bus

RM7965A product features an enhanced version of the multiplexed Address/Data bus (**SysAD**), first introduced with the debut of the RM70xxC products. The function of the SysAD bus is to transfer addresses and data between the CPU and the rest of the system. The enhanced version can run up to 200 MHz, providing up to 12.8 Gbit/s or 1.6 GB/s of bandwidth. It supports legacy designs with a seamless upgrade path for all RM70xx and RM52xx processors, and maintains compatibility with all existing and future companion chips that utilize **SysAD** functionality.

The 64-bit SysAD bus present on the RM7965A processor enables 36 bits of physical addressing and 64 bits of data, and is supported by an 8-bit parity check bus (**SysADC[7:0]**) and a 9-bit command bus (**SysCmd[8:0]**). In addition, there are ten handshake signals and ten interrupt inputs. It can run up to 133 MHz in standard LVTTL mode, or up to 200 MHz in the enhanced HSTL mode. The **SysAD** bus runs at the same frequency as the RM7965A master clock. The SysAD interface for the RM7965A also supports up to two outstanding reads, and it can return the reads out of order.

The SysAD bus is also configurable to allow easy interfacing to memory and I/O systems of varying frequencies. The data rate and the bus frequency at which RM7965A product transmits data to the system interface is programmable at boot time via mode control bits. Additionally, 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 RM7965A processor.

6.2 System Command Bus

All RM7965A processors feature a 9-bit System Command bus, **SysCmd[8:0]**. The command bus indicates whether the **SysAD** bus carries address or data information on a per-clock basis. If the **SysAD** bus carries an address, the **SysCmd** bus indicates the transaction type (for example, a read or write). If the **SysAD** bus carries data, then the **SysCmd** bus contains 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 RM7965A. Processor requests are initiated by the RM7965A and responded to by an external device. External requests are issued by an external agent and require the RM7965A to respond.

The RM7965A support 1 to 8-byte transfers as well as 32-byte block transfers on the **SysAD** bus. In the case of a sub-doubleword transfer, the 3 low-order address bits give the byte address of the transfer, and the **SysCmd** bus indicates the number of bytes being transferred.

6.3 Handshake Signals

There are 10 handshake signals on the system interface of the RM7965A. Two of these, **RdRdy*** and **WrRdy***, are common to all RM7965A CPUs. They are driven by an external agent to indicate to the RM7965A whether it can accept a new read or write transaction. The RM7965A samples these signals before deasserting the address on read and write requests.

ExtRqst* and **Release*** are also common to all RM7965A CPUs. They are used to transfer control of the **SysAD** and **SysCmd** buses from the processor to an external agent. When an external agent requires control of the bus, it asserts **ExtRqst***. The RM7965A responds by asserting **Release*** to release the system interface to slave state.

PRqst* and **PAck*** are supported by the RM7965A. These signals are used to transfer control of the **SysAD** and **SysCmd** buses from the external agent to the processor. These two pins have been added to the system interface to support multiple outstanding reads and facilitate non-blocking cache operations. When the processor needs to reacquire control of the interface, it asserts **PRqst***. The external agent responds by asserting **PAck*** to return control of the interface to the processor.

RspSwap* is used by the external agent to indicate to the processor when it is returning multiple data requests out of order. For example, when there are two outstanding reads, the external agent asserts **RspSwap*** when it is going to return the data for the second read before it returns the data for the first read.

RdType is a pin on the interface that indicates whether a read is an instruction read or a data read. When asserted, the reference is an instruction read. When deasserted it is a data read. **RdType** is only valid during valid address cycles.

ValidOut* and **ValidIn*** are used by the RM7965A and its external agents to indicate that there is a valid command and data on the **SysAD** and **SysCmd** buses. The RM7965A asserts **ValidOut*** when it is driving these buses with a valid command and data, and the external agent drives **ValidIn*** when it has control of the system interface and is driving a valid command and data.

6.4 System Interface Operation

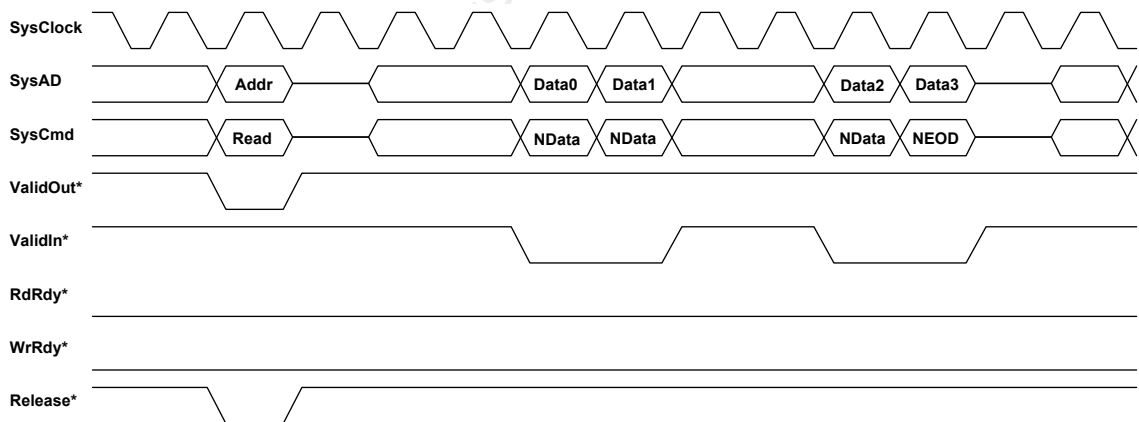
To support non-blocking caches and data prefetch instructions, the RM7965A allow two outstanding reads. An external agent may respond to read requests in whatever order it chooses by using the response order indicator pin **RspSwap***. No more than two read requests are outstanding to the external agent. Support for multiple outstanding reads can be enabled or disabled via a boot-time mode bit. Refer to Section 8 for a complete list of mode bits.

The RM7965A can issue read and write requests to an external agent, while an external agent can issue null and read responses to the RM7965A.

For processor reads, the RM7965A 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 signals the release of the system interface to slave state by asserting **Release***. The external agent can then begin sending data to the RM7965A.

Figure 8 shows a processor block read request and the external agent read response for a system.

Figure 8 Processor Block Read



In Figure 8 the read latency is 4 cycles (**ValidOut*** to **ValidIn***), and the response data pattern is DDxxDD. Figure 9 shows a processor block write where the processor was programmed with write-back data rate boot code 2, or DDxxDDxx.

Figure 10 shows a typical RM7965A sequence resulting in two outstanding reads as explained in the following sequence:

1. The processor issues a read.
2. The external agent takes control of the bus in preparation for returning data to the processor.
3. The processor encounters another internal cache miss and therefore asserts PRqst* in order to regain control of the bus.
4. The external agent pulses PAck*, returning control of the bus to the processor.
5. The processor issues a read for the second miss.
6. The RspSwap* pin is asserted to denote the out of order response. Not shown in the figure is the completion of the data transfer for the second miss, or any of the data transfer for the first miss.
7. The external agent retakes control of the bus and begins returning data (out of order) for the second miss to the processor

Figure 9 Processor Block Write

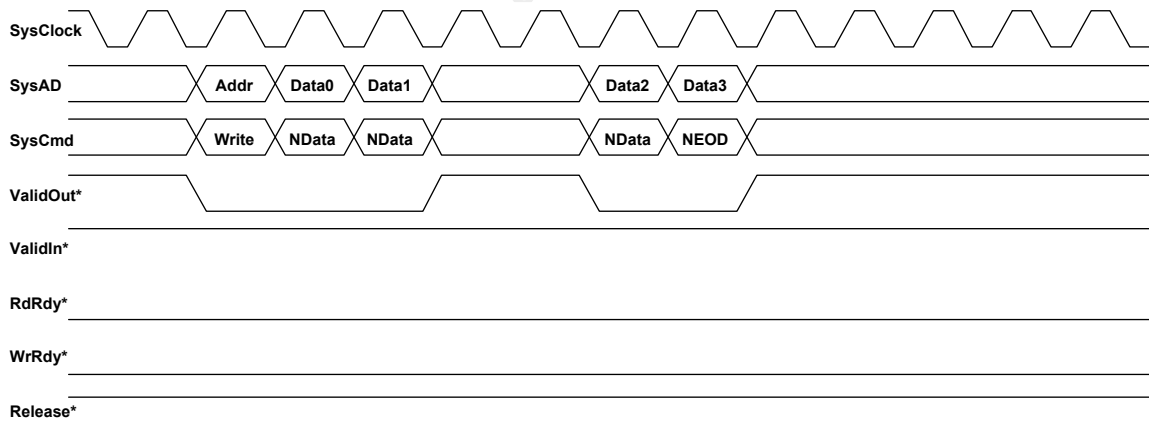
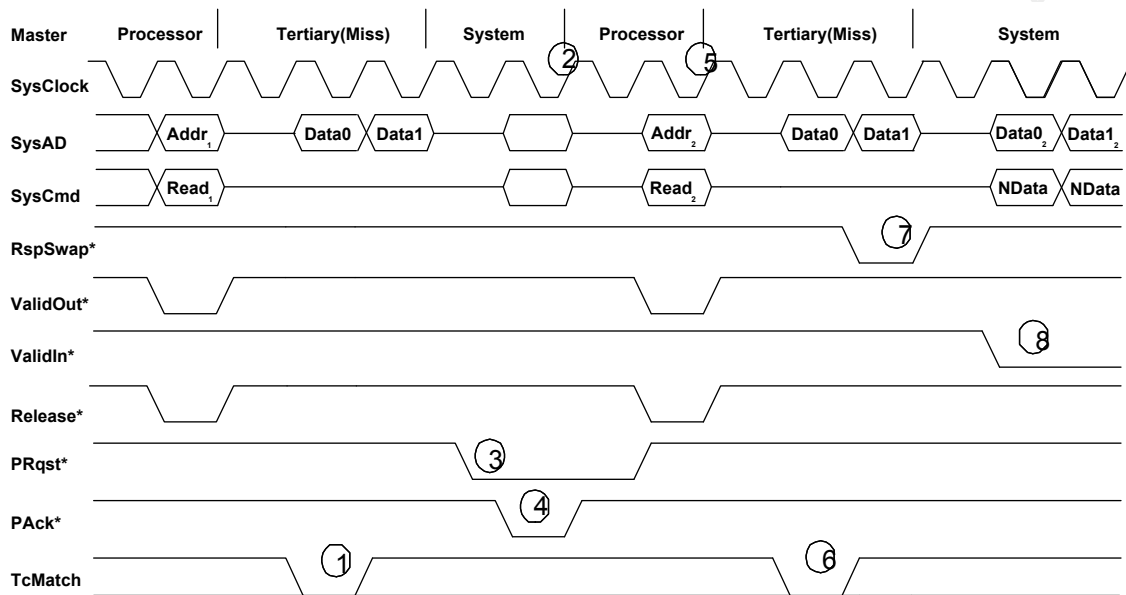


Figure 10 Multiple Outstanding Reads



6.5 Write Modes

The RM7965A implements two write modes: Pipeline Writes and Write Reissue. 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. 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 missed.

Write reissue mode is an enhancement to pipelined write mode and allows the processor to reissue missed 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 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***.

7 Integrated Debug

The E9000 has extended the debugging features found on the RM7000 and has added EJTAG Debugging and a 64-entry Branch Inst. Trace Buffer.

7.1 EJTAG Debugging

The EJTAG 2.5 standard is implemented to allow access to the processor subsystem through the EJTAG port. This allows an emulator to be plugged into the EJTAG port to single-step, modify memory and registers, and to provide hardware breakpoints. EJTAG mode on the RM7965A is selected by using the **JTAGSEL** pin. When **JTAGSEL** is set to “1”, JTAG is selected. When **JTAGSEL** is set to “0”, EJTAG is selected.

A new exception vector at 0xBFC0_0240 is allocated for EJTAG Debugging. In addition, a Debug Register Section at 0xff20_0000 and a Debug Memory Section at 0xff30_0000 to 0xff3f_fff is implemented.

Two new instructions have been added to support on-chip debugging. A Software Debug Break-Point (SDBBP) allows breakpoints to be taken by the code. Once in the debug exception handler, the Debug Return (DERET) instruction is used to exit the debug exception handler.

Three new CP0 registers have been added in the CP0 system address space to support EJTAG functionality. The *EJTAG_Debug* register at CP0_23 serves as the control and status register. The *EJTAG_DEPC* register at CP0_24 serves as the same purpose for the debug exception as the EPC register does for general exceptions. The *EJTAG_DESave* register at CP0_31 is used as a general purpose “save area” for EJTAG debug support. See Figure 5.

7.2 Trace Buffer

A Trace buffer is implemented on the processor core to allow tracing of instruction flow. The trace buffers are 64-entries deep and capture branch addresses and branch target addresses so that the precise flow of instruction execution can be reconstructed. Using this sophisticated compression technique, the reconstructed instruction length can be many times larger than the trace buffer length. The trace buffer can trigger an interrupt when it is $\frac{1}{4}$, $\frac{1}{2}$, $\frac{3}{4}$ or completely full. If no interrupt is set, the buffer will wrap around. The trace buffer shares the *IPI3* interrupt with the Performance Counters.

To support the Trace Buffer, 3 new CP0 register are implemented in the CP0 control address space. The Trace Buffer Control and Status (*TB CSR*) register is at CP0_22 and performs the function its name suggests. The Trace Buffer Index (*TB IDX*) register is at CP0_24 and is the address into the trace buffer. The Trace Buffer Out (*TB Out*) register is at CP0_23 and contains data from the read at the index given in the *TB IDX* register.

7.3 Test/Breakpoint Registers

To facilitate hardware and software debugging, the RM7965A incorporates a pair of Test/Breakpoint, or Watch registers, called *Watch1* and *Watch2*. Each Watch register can be separately enabled to watch for a load address, a store address, or an instruction address. All address comparisons are done on virtual addresses. An associated register, *WatchMask*, allows either or both of the Watch registers to compare against an address range rather than a specific address. The range granularity is limited to a power of two.

When enabled, a match of either Watch register results in an exception. If the Watch is enabled for a load or store address then the exception is the Watch exception as defined for the R4000 by Cause exception code 23. If the Watch is enabled for instruction addresses then a *Instruction Watch* exception is taken and the Cause exception code is 16. The Watch register that caused the exception is indicated by *Cause* bits 25:24. Table 17 summarizes a Watch operation.

If the *DBEN* bit is set, an address comparison will cause a Debug Exception, which vectors to 0xbfc00240.

Table 17 Watch Registers

Register	Bit Field/Function										
	63:59	58	57	56	55	54	53:40	39:2	1	0	
Watch 1	Caddr [63:59]	Store	Load	Inst	DBEN	DBOut	Rsvd	Caddr [39:2]	0	0	
Watch 2	Caddr [63:59]	Store	Load	Inst	DBEN	DBOut	Rsvd	Caddr [39:2]	0	0	
Watch Mask	Mask [63:59]	Reserved					Mask [39:2]	Mask Watch2	Mask Watch 1		

Note:

1. The *W1* and *W2* bits of the *Cause* register indicate which Watch register caused a particular Watch exception.

7.4 Performance Counters

The RM7965A supports two CP0 performance-counter registers with the *PerfCount* and *PerfControl* registers. The *PerfCount* register is a 64-bit register divided into two independent 32-bit counters, *PerfCounter0*, *PerfCounter1*. The counters can be written to by software to initialize event monitoring, and they generate a performance-counter interrupt when the most significant bit in either counter (bit 63 in Counter 2, and bit 31 in Counter 1) is set.

The *PerfControl* register is a 32-bit register containing two 5-bit fields used to select one of twenty-four event types counted by each counter, as well as a handful of bits which control the overall counting function. Note that only one event type can be counted per counter at a time, and that counting can occur for user code, kernel code or both. The event types and control bits are listed in Table 18.

Table 18 Performance Counter Control

PerfControl Field	Description
4:0	Event Type 00: Clock cycles 01: Total instructions issued (Integer and Floating Point) 02: Floating-point instructions issued (any COP1 or COP3). 03: Integer instructions issued (no COP1 or COP3). 04: Load instructions issued 05: Store instructions issued 06: Dual issued instruction pairs 07: Branch mispredictions 08: External Cache Misses 09: Stall cycles 0A: Secondary cache misses 0B: Instruction cache misses 0C: Data cache misses 0D: Data TLB misses 0E: Instruction TLB misses 0F: Joint TLB instruction misses 10: Joint TLB data misses 11: Branches taken 12: Branches issued 13: Secondary cache writebacks 14: Data cache writebacks 15: Data cache miss stall cycles (A stall occurs when the data cache is processing two misses and a third miss occurs). 16: Cache misses (all caches). 17: FP possible exception cycles 18: Slip Cycles due to multiplier busy 19: Coprocessor 0 slip cycles 1A: Slip cycles due to pending non-blocking loads 1B: Stall cycles due to full Write buffer 1C: Stall cycles due to Cache instruction 1D: Unused 1E: Stall cycles due to pending non-blocking loads - stall start of exception

PerfControl Field	Description
7:5	Reserved (must be zero)
8	Count in Kernel Mode 0: Disable 1: Enable
9	Count in User Mode 0: Disable 1: Enable
10	Count Enable 0: Disable 1: Enable
31:11	Reserved (must be zero)

The performance counter interrupt only occurs when interrupts are enabled in the *Status* register, **IE**=1, and the Interrupt Mask bit 13 (**IMI3**) of the coprocessor 0 interrupt control register is set. The performance counter shares this interrupt with the 64-entry branch Trace Buffer.

Since a performance counter can be set up to count clock cycles, it can be used as either a second timer, or a watchdog interrupt. A watchdog interrupt can be used as an aid in debugging system or software “hangs.” Typically the software is set up to periodically update the count so that no interrupt occurs. When a hang occurs the interrupt ultimately triggers, thereby breaking free from the hang-up.

8 Boot-Mode Settings

The RM7965A operating modes are initialized at power-up by the boot-time mode control interface. The serial boot-time mode control interface operates at a very low frequency (**SysClock** divided by 256), allowing the initialization information to be kept in a low cost EPROM or system interface ASIC.

The boot-time serial mode stream is defined below. Bit 0 is presented to the processor as the first bit in the stream following **VccOK** being asserted. Bit 255 is the last bit transferred. An automated mode bit generation tool (program that runs on a PC) is available on the PMC-Sierra website.

Name	Size	Field	Description
Reserved	1	0	Must be set to 0.
preBigEndian	1	1	Places the processor in big endian mode. 0: Little Endian (Little) 1: Big Endian (Big)
SI32Wide	1	2	Sets the SysAD interface width to 32 bits. 0: 64-bit SysAD (SADSz64) 1: 32-bit SysAD (SADSz32)
SADRdOverlap	1	3	Enables overlapping reads on the SysAD interface. 0: Overlap disabled (OvlpDisabled) 1: Overlap enabled (OvlpEnabled)
SADWrProt[1:0]	2	5:4	SysAD interface write protocol. 00: R4000 compatible (R4000) 01: Reserved 10: Pipelined writes (Pipelined) 11: Write re-issue (Relssue)
SADDatRate[3:0]	4	9:6	SysAD interface write transmit data rate. 0000: Dd 0001: Ddx 0010: Ddxx 0011: Ddx 0100: Ddxxx 0101: Ddxxxx 0110: Dxxdxx 0111: Ddxxxxx 1000: Dxxdxxx 1001-1111: Reserved
ECacheEn	1	10	Enables the external cache. 0: ECache Disabled 1: ECache Enabled
ECBurstMd	1	11	Sets ECache protocol for burst mode RAMs. 0: Dual Cycle Deselect, (DCD), 1: Single Cycle Deselect, (SCD).
Reserved	1	12	Must be set to 0.
Reserved	1	13	Must be set to 0.

Name	Size	Field	Description																																																																																																												
DrvStren[1:0]	2	15:14	Sets the drive strength of the pad output drivers. 00: Drive at 67%, 01: Drive at 50%, 10: Drive at 100%, 11: Drive at 83%.																																																																																																												
SyncSysAD[4:0]	5	20:16	Sample and drive sync generation for the SysAD interface. SyncSysAD[4] = 0, reserved.																																																																																																												
SyncHalfSysAD	1	21	Sample and drive sync generation is half integers for the SysAD interface. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="border-bottom: 1px solid black;">SyncHalfSysAD</th> <th style="border-bottom: 1px solid black;">SyncSysAD</th> <th style="border-bottom: 1px solid black;">Ratio</th> </tr> </thead> <tbody> <tr><td>0</td><td>0000</td><td>2:1</td></tr> <tr><td>0</td><td>0001</td><td>3:1</td></tr> <tr><td>0</td><td>0010</td><td>4:1</td></tr> <tr><td>0</td><td>0011</td><td>5:1</td></tr> <tr><td>0</td><td>0100</td><td>6:1</td></tr> <tr><td>0</td><td>0101</td><td>7:1</td></tr> <tr><td>0</td><td>0110</td><td>8:1</td></tr> <tr><td>0</td><td>0111</td><td>9:1</td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td>0</td><td>0100</td><td>10:1</td></tr> <tr><td>0</td><td>01001</td><td>11:1</td></tr> <tr><td>0</td><td>01010</td><td>12:1</td></tr> <tr><td>0</td><td>01011</td><td>13:1</td></tr> <tr><td>0</td><td>01100</td><td>14:1</td></tr> <tr><td>0</td><td>01101</td><td>15:1</td></tr> <tr><td>0</td><td>01110</td><td>16:1</td></tr> <tr><td>0</td><td>01111</td><td>17:1</td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td>1</td><td>0000</td><td>Rsvd</td></tr> <tr><td>1</td><td>00001</td><td>Rsvd</td></tr> <tr><td>1</td><td>00010</td><td>Rsvd</td></tr> <tr><td>1</td><td>00011</td><td>Rsvd</td></tr> <tr><td>1</td><td>00100</td><td>Rsvd</td></tr> <tr><td>1</td><td>00101</td><td>3.5:1</td></tr> <tr><td>1</td><td>00110</td><td>Rsvd</td></tr> <tr><td>1</td><td>00111</td><td>4.5:1</td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td>1</td><td>01000</td><td>Rsvd</td></tr> <tr><td>1</td><td>01001</td><td>5.5:1</td></tr> <tr><td>1</td><td>01010</td><td>Rsvd</td></tr> <tr><td>1</td><td>01011</td><td>6.5:1</td></tr> <tr><td>1</td><td>01100</td><td>Rsvd</td></tr> <tr><td>1</td><td>01101</td><td>7.5:1</td></tr> <tr><td>1</td><td>01110</td><td>Rsvd</td></tr> <tr><td>1</td><td>01111</td><td>8.5:1</td></tr> </tbody> </table>	SyncHalfSysAD	SyncSysAD	Ratio	0	0000	2:1	0	0001	3:1	0	0010	4:1	0	0011	5:1	0	0100	6:1	0	0101	7:1	0	0110	8:1	0	0111	9:1				0	0100	10:1	0	01001	11:1	0	01010	12:1	0	01011	13:1	0	01100	14:1	0	01101	15:1	0	01110	16:1	0	01111	17:1				1	0000	Rsvd	1	00001	Rsvd	1	00010	Rsvd	1	00011	Rsvd	1	00100	Rsvd	1	00101	3.5:1	1	00110	Rsvd	1	00111	4.5:1				1	01000	Rsvd	1	01001	5.5:1	1	01010	Rsvd	1	01011	6.5:1	1	01100	Rsvd	1	01101	7.5:1	1	01110	Rsvd	1	01111	8.5:1
SyncHalfSysAD	SyncSysAD	Ratio																																																																																																													
0	0000	2:1																																																																																																													
0	0001	3:1																																																																																																													
0	0010	4:1																																																																																																													
0	0011	5:1																																																																																																													
0	0100	6:1																																																																																																													
0	0101	7:1																																																																																																													
0	0110	8:1																																																																																																													
0	0111	9:1																																																																																																													
0	0100	10:1																																																																																																													
0	01001	11:1																																																																																																													
0	01010	12:1																																																																																																													
0	01011	13:1																																																																																																													
0	01100	14:1																																																																																																													
0	01101	15:1																																																																																																													
0	01110	16:1																																																																																																													
0	01111	17:1																																																																																																													
1	0000	Rsvd																																																																																																													
1	00001	Rsvd																																																																																																													
1	00010	Rsvd																																																																																																													
1	00011	Rsvd																																																																																																													
1	00100	Rsvd																																																																																																													
1	00101	3.5:1																																																																																																													
1	00110	Rsvd																																																																																																													
1	00111	4.5:1																																																																																																													
1	01000	Rsvd																																																																																																													
1	01001	5.5:1																																																																																																													
1	01010	Rsvd																																																																																																													
1	01011	6.5:1																																																																																																													
1	01100	Rsvd																																																																																																													
1	01101	7.5:1																																																																																																													
1	01110	Rsvd																																																																																																													
1	01111	8.5:1																																																																																																													
BIUPbRsvd[3:0]	4	25:22	Reserved, Must be set to 0.																																																																																																												
TimIntDis	1	26	Disables the timer interrupt to interrupt bit 5. 0: Timer enabled (TimerEnabled) 1: Timer disabled (TimerDisabled)																																																																																																												
Timer1X	1	27	Sets counter/timer to run at 1X processor clock frequency. 0: Normal frequency (TimerNormal) 1: 1X frequency (Timer1X)																																																																																																												

Name	Size	Field	Description
SysConfig[1:0]	2	29:28	System configuration mode bits, to <i>Config</i> register. Value software visible in Config[21:20] .
OCacheEn	1	30	Enables the core on-chip secondary caches. 0: OCache Disabled (OCacheDisabled) 1: OCache Enabled (OCacheEnabled)
OTClrEn	1	31	Enables the Ocache tag clear machine on cold reset. When OTClrEn = 1, the following will be cleared: L2 Tag, L1 DTag, L1 Itag, L1 Dcache, L1 Icache and BranchPredict RAM. 0: OTag clear machine disabled (OTClrDisabled) 1: OTag clear machine enabled (OTClrEnabled)
ParChkDis	1	32	Disables all parity checking processor-wide. 0: Par check enabled (ParChkEnabled) 1: Par check disabled (ParChkDisabled)
TLB64Ent	1	33	Enables a larger JTLB size on the core. 0: 48 entry JTLB (TLB48Entry) 1: 64 entry JTLB (TLB64Entry)
HitShrFtch	1	34	Reserved, must be set to 0.
MIPS64Compat	1	35	MIPS 64 compatibility mode. Reorganizes CP0 to be MIPS 64 compatible. 0: MIPS IV compatibility (PMCCompat) 1: MIPS 64 compatibility (MIPS64Compat)
PowerSave	1	36	Reserved, must be set to 0.
CorePbRsvd[3:0]	4	40:37	Reserved, must be set to 0.
CkPdAlgn[1:0]	2	42:41	Adjusts the MasterClock pad delay matching network. Reduces the swing on the internal MasterClock equivalent signal fed back to PLLs for matching the external MasterClock swing. These mode bits are for HSTL only. They should be left at 00 in the LVTTTL mode. 00: No swing control - Full swing (0–1.2V) - default setting for cy2210 clock driver (common mode voltage - 0.6V) 01: MasterClock (internal f/b signal) swing matched for external MasterClock swing of < 0.5V 10: External MasterClock swing is 0.5–0.4V 11: External MasterClock swing < 0.4V Default setting 00.
PLLDis	1	43	Enables or disables the PLL. 0: Enabled (PLLEnabled) 1: Disabled (PLLDisabled)
DivMa2Core	1	44	MasterClock divide by two for PLL. 0: Divide by one (DivBy1) 1: Divide by two (DivBy2)

Name	Size	Field	Description
MulFundCore[4:0]	5	49:45	<p>Fundamental clock multiplier for PLL.</p> <p>0: Multiply by 2 (MultiplyBy2) 1: Multiply by 3 (MultiplyBy3) 2: Multiply by 4 (MultiplyBy4) 3: Multiply by 5 (MultiplyBy5) 4: Multiply by 6 (MultiplyBy6) 5: Multiply by 7 (MultiplyBy7) 6: Multiply by 8 (MultiplyBy8) 7: Multiply by 9 (MultiplyBy9)</p> <p>8: Multiply by 10 (MultiplyBy10) 9: Multiply by 11 (MultiplyBy11) a: Multiply by 12 (MultiplyBy12) b: Multiply by 13 (MultiplyBy13) c: Multiply by 14 (MultiplyBy14) d: Multiply by 15 (MultiplyBy15) e: Multiply by 16 (MultiplyBy16) f: Multiply by 17 (MultiplyBy17)</p> <p>10: Reserved (MultiplyBy18) 11: Reserved (MultiplyBy19) 12: Reserved (MultiplyBy20) 13: Reserved (MultiplyBy21) 14: Reserved (MultiplyBy22) 15: Reserved (MultiplyBy23) 16: Reserved (MultiplyBy24) 17: Reserved (MultiplyBy25)</p> <p>18: Reserved (MultiplyBy26) 19: Reserved (MultiplyBy27) 1a: Reserved (MultiplyBy28) 1b: Reserved (MultiplyBy29) 1c: Reserved (MultiplyBy30) 1d: Reserved (MultiplyBy31) 1e: Reserved (MultiplyBy32) 1f: Reserved</p>

Name	Size	Field	Description
DivXCore[4:0]	5	54:50	<p>Processor core logic clock divisor from processor core fundamental clock.</p> <p>0: Reserved (DivideBy2) 1: Reserved (DivideBy3) 2: Reserved (DivideBy4) 3: Reserved (DivideBy5) 4: Reserved (DivideBy6) 5: Reserved (DivideBy7) 6: Reserved (DivideBy8) 7: Reserved (DivideBy9)</p> <p>8: Reserved (DivideBy10) 9: Reserved (DivideBy11) a: Reserved (DivideBy12) b: Reserved (DivideBy13) c: Reserved (DivideBy14) d: Reserved (DivideBy15) e: Reserved (DivideBy16) f: Reserved (DivideBy17)</p> <p>10: Reserved (DivideBy18) 11: Reserved (DivideBy19) 12: Reserved (DivideBy20) 13: Reserved (DivideBy21) 14: Reserved (DivideBy22) 15: Reserved (DivideBy23) 16: Reserved (DivideBy24) 17: Reserved (DivideBy25)</p> <p>18: Reserved (DivideBy26) 19: Reserved (DivideBy27) 1a: Reserved (DivideBy28) 1b: Reserved (DivideBy29) 1c: Reserved (DivideBy30) 1d: Reserved (DivideBy31) 1e: Reserved (DivideBy32) 1f: Divide by 1 (DivideBy1)</p>
ClockPbRsvd[3:0]	4	58:55	Reserved, must be set to 0.
MBRsvd[2:0]	3	61:59	Reserved, must be set to 0.
HSTLCntI[1:0]	2	63:62	<p>HSTL output delay control.</p> <p>Must be set to 01 in HSTL mode. Must be set to 00 in LVTTL mode.</p>
MBRsvd[45:3]	43	106:64	Reserved, must be set to 0.
HSTLCntI[3:2]	2	108:107	<p>HSTL output delay control.</p> <p>Must be set to 11 in HSTL mode. Must be set to 00 in LVTTL mode.</p>
MBRsvd[192:46]	147	255:109	Reserved, must be set to 0.

9 RM7000 and RM7965A Differences

Feature	RM7000	RM7965A
Number of CPU Cores	1	1
Pipeline Stages	5	7
Load Delay, Branch Delay	1	2
Branch Prediction	No	8K BHT/Core
Hardware Cache Coherency Support	No	No
Secondary Cache Protection	Parity	Error Checking and Correcting (ECC)
Page Size	4 KB – 16 MB	4 KB – 256 MB
Number of ASID Bits	8	12
New Instructions	—	MSUB, MSUBU, SSNOP, SDBBP, DERET
Integer Multiplier	Iterative	Pipelined
Integrated Buses	SysAD	SysAD
SysAD Bus Width	64-bit (RM7000x), RM7065x) or 32-bit (RM7035C)	64-bit
SysAD Maximum Bus Frequency	125 MHz (RM70xxA) 133 MHz (LVTTL) or 200 MHz (HSTL) for all RM70xxC CPUs	133 MHz (LVTTL) or 200 MHz (HSTL)
L3 Cache Interface	Yes (RM7000x only)	No
L3 Page Invalidate Cache Op	Stores TagLo register	Stores constant zero
On-Chip Debugging	No	Yes
EJTAG Emulator Support	No	Yes
Integrated Instruction Trace Buffer	No	Yes
Watch Register Addressing	Physical	Virtual
Number of Performance Counters	1	2

The following lists the significant additions to the RM7965A product:

- The **SysAD** bus supports both 133 MHz LVTTL and 200 MHz HSTL **SysAD** bus frequencies.
- Integrated debug support includes EJTAG TAP support and on-chip trace buffers. The added debug mechanism includes two new instructions: Software Debug Break Point (SDBBP), and Debug Exception Return (DERET). The added debug mechanism has its own exception vector located at 0xBFC00480 and its own 2 MB memory space at 0xFF200000.
- New instructions: Multiply-Subtract, both signed and unsigned (MSUB/MSUBU), and superscalar NOP (SSNOP) which issues a NOP to each pipeline.
- Branch prediction that provides the CPU core with up to 8K entries of branch history.
- Virtual Watch register addressing. This is a change from the physical Watch register addressing on the RM7000. The two Watch registers and the Watch Mask have been enlarged to reflect this change.

- 2 performance counters so 2 simultaneous events can now be counted. In addition, branch miss-predicts have been added as a performance counter event, and the multiplication stalls event has been removed.
- Increased page size range. The page size on the RM7000 can range from 4 KB to 16 MB, and on the RM7965A the range is from 4 KB to 256 MB. The ASID has been extended from 8-bits to 12-bits.
- Load delay and branch delay increases from 1 to 2.

Downloaded [controlled] by Venkatesh Betageri of IHS on Wednesday, 12 January, 2017 02:53:04

10 Pin Descriptions

The following is a list of control, data, clock, tertiary cache, interrupt, and miscellaneous pins of the RM7965A.

Table 19 System Interface

Pin Name	Type	Description
ExtRqst*	Input	External request Signals that the external agent 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 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.
PRqst*	Output	Processor Request When asserted this signal requests that control of the system interface be returned to the processor.
PAck*	Input	Processor Acknowledge When asserted, in response to PRqst* , this signal indicates to the processor that it has been granted control of the system interface.
RspSwap*	Input	Response Swap RspSwap* is used by the external agent to signal the processor when it is about to return a memory reference out of order; i.e., of two outstanding memory references, the data for the second reference is being returned ahead of the data for the first reference. In order that the processor will have time to switch the address to the tertiary cache, this signal must be asserted a minimum of two cycles prior to the data itself being presented. Note that this signal works as a toggle; i.e., for each cycle that it is held asserted the order of return is reversed. By default, anytime the processor issues a second read it is assumed that the reads will be returned in order; i.e., no action is required if the reads are indeed returned in order.
RdType	Output	Read Type During the address cycle of a read request, RdType indicates whether the read request is an instruction read or a data read.

Pin Name	Type	Description
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	System Command/Data Identifier Bus Parity For the RM7965A, unused on input and zero on output.

Table 20 Clock/Control Interface

Pin Name	Type	Description
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.
SysClock*	Input	System clock Differential clock input used only in HSTL I/O mode. Set SysClock* to VccIO or Do Not Connect for non-HSTL operation.

Table 21 Power Supply

Pin Name	Type	Description
VccInt	Input	Power supply for core.
VccIO	Input	Power supply for I/O.
VccP	Input	Vcc for PLL Quiet VccInt for the internal phase locked loop. Must be connected to VccInt through a filter circuit.
VccJ	Input	Power supply used for JTAG.
Vref_In	Input	Reference voltage for HSTL I/O. Do Not Connect for non-HSTL.
Vss	Input	Ground Return.
VssP	Input	Vss for PLL Quiet Vss for the internal phase locked loop. Must be connected to Vss through a filter circuit.

Table 22 Interrupt Interface

Pin Name	Type	Description
INT[9:0]*	Input	Interrupt Ten general processor interrupts, bit-wise ORed with bits 9:0 of the interrupt register.
NMI*	Input	Non-maskable interrupt Non-maskable interrupt, ORed with bit 15 of the interrupt register..

Table 23 JTAG Interface

Pin Name	Type	Description
JTDI/DBDI	Input	JTAG/EJTAG data in JTAG/EJTAG serial data in.
JTCK/DBCK	Input	JTAG/EJTAG clock input JTAG/EJTAG serial clock input.
JTDO/DBDO	Output	JTAG/EJTAG data out JTAG/EJTAG serial data out.
JTMS/DBMS	Input	JTAG/EJTAG command JTAG/EJTAG command signal, signals that the incoming serial data is command data.
JTRST*/DBRST*	Input	JTAG/EJTAG reset.
JTAGSEL	Input	JTAG/EJTAG select Selects JTAG when JTAGSEL=1 ; selects EJTAG when JTAGSEL=0

Notes:

1. The **JTRST*** input was added to the RM70xxC and RM7965A CPUs to directly control the reset to the JTAG state machine. JTAG boundary scan test equipment must be able to drive **JTRST*** high to allow JTAG boundary scan operation.
2. The **JTRST*** input must be connected to GND (**Vss**) through a 220 Ω to 1 K Ω pull-down resistor to force the JTAG state machine into the reset state to allow normal operation (JTAG boundary scan mode disabled).
3. The JTAG interface electrical characteristics are dependent on the **VccJ** level chosen (2.5 V or 3.3 V).

Table 24 Initialization Interface

Pin Name	Type	Description
BigEndian	Input	Big Endian / Little Endian Control 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 RM7965A that the Vcclnt power supply has been above the recommended value 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 .

Pin Name	Type	Description
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 ¹ 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.
HSTL_Sel*	Input	HSTL/LVTTL Control Asserting this signal low places the system I/O pins in HSTL mode. Pulling this signal high or allowing to float places all system I/O pins in LVTTL mode.

Note

1. In HSTL mode, maximum voltage level of the **ModeClock** is determined by VccJ.

11 Absolute Maximum Ratings

Symbol	Rating	Limits	Unit
V_{TERM}	Terminal Voltage with respect to V_{SS}	0.5 to +3.9	V
T_{CASE}	Operating Temperature		
	Commercial	0 to +85	°C
	Industrial	-40 to +85	°C
T_{STG}	Storage Temperature	-55 to +125	°C
I_{IN}	DC Input Current	±20	mA
I_{OUT}	DC Output Current ⁴	±20	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 = -2.0 V for pulse width less than 15 ns. V_{IN} should not exceed 3.9 V.
3. When $V_{\text{IN}} < 0\text{V}$ or $V_{\text{IN}} > V_{\text{CCIO}}$
4. Not more than one output should be shorted at a time. Duration of the short should not exceed 30 seconds.

12 DC Electrical Characteristics

Table 25 (VccIO = 3.15 V – 3.45 V)

Parameter	Minimum	Maximum	Conditions
V _{OL}		0.2 V	I _{OUT} = 100 μA
V _{OH}	V _{ccIO} - 0.2 V		
V _{OL}		0.4 V	I _{OUT} = 2 mA
V _{OH}	2.4 V		
V _{IL}	-0.3 V	0.8 V	
V _{IH}	2.0 V	V _{ccIO} + 0.3 V	
I _{IN}		±15 μA ±15 μA	V _{IN} = 0 V _{IN} = V _{ccIO}

Table 26 (VccIO = 2.3 V – 2.7 V)

Parameter	Minimum	Maximum	Conditions
V _{OL}		0.2 V	I _{OUT} = 100 μA
V _{OH}	2.1 V		
V _{OL}		0.4 V	I _{OUT} = 1 mA
V _{OH}	2.0 V		
V _{OL}		0.7 V	I _{OUT} = 2 mA
V _{OH}	1.7 V		
V _{IL}	-0.3 V	0.7 V	
V _{IH}	1.7 V	V _{ccIO} + 0.3 V	
I _{IN}		±15 μA ±15 μA	V _{IN} = 0 V _{IN} = V _{ccIO}

Note for Table 25 and Table 26:

- For V_{ccIO} levels in Table 25 and Table 26, set HSTL_Sel* to V_{ccIO} or Do Not Connect.

Table 27 (VccIO = 1.4 V – 1.6 V) HSTL

Parameter	Minimum	Maximum	Conditions
V _{OL}	V _{SS}	0.4 V	I _{OUT} = 16 mA
V _{OH}	V _{ccIO} -0.4 V	V _{ccIO}	
V _{IL}	-0.3 V	V _{ref} -0.2 V	
V _{IH}	V _{ref} +0.2 V	V _{ccIO} +0.3 V	
V _{REF}	0.6 V	0.9 V	
V _{IN_CLK}	-0.3 V	V _{ccIO} +0.3 V	
V _{DIF_CLK}	0.1 V	V _{ccIO} +0.6 V	
V _{CM_CLK}	0.6 V	0.9 V	

Note

- Set HSTL_Sel* to V_{SS} for HSTL operation.

13 Power

13.1 Normal Operating Conditions

Table 28 Normal Operating Voltages for 0.13 μ m CMOS

Grade	CPU Speed	Case Temp	Vss	Vcclnt	VccIO	VccP	VccJ
Commercial	900 MHz (part labeled as -900)	0°C to +85°C	0 V	1.32 V \pm 50 mV [1.30 V \pm 50 mV if operated at 835 MHz or less]	3.3 V \pm 150 mV or 2.5 V \pm 200 mV or 1.5 V \pm 100 mV	1.32 V \pm 50 mV [1.30 V \pm 50 mV if operated at 835 MHz or less]	3.3 V \pm 150 mV or 2.5 V \pm 200 mV

Notes:

1. **VccIO** should not exceed **Vcclnt** by greater than 2.5 V during the power-up sequence.
2. Applying a logic high state to any I/O pin before **Vcclnt** becomes stable is not recommended.
3. For normal operation (non-boundary-scan), **JTRST*** must be pulled down to Vss (0 V) to avoid entering JTAG test mode.
4. **VccP** must be connected to **Vcclnt** through a passive filter circuit. See RM79xx User Manual for recommended circuit.
5. Power supply, D.C. characteristics, and A.C. timing are characterized across these operating ranges, unless otherwise stated.
6. The Vcclnt and VccP voltages can be reduced (by 20 mV) to 1.30 V \pm 50 mV if the RM7965A is operated at 835 MHz or less.

13.2 Power Requirements

Table 29 VccINT Power Requirements

Conditions	Parameter	Typ	Thermal	Max	Units
900 MHz (Tcase = 85°C)	Icc Max	2.6	—	3.65	A
900 MHz (Tcase = 50°C)	Icc Wait	1.0	—	1.46	A
900 MHz (Tcase = 85°C)	Total Power (Max)	3.5	3.7	—	W
900 MHz (Tcase = 50°C)	Total Power (Wait)	1.37	—	—	W

Notes:

- Outputs loaded with 30 pF (if not otherwise specified), and a normal amount of traffic or signal activity.
- Power values are calculated using the formula:

$$\text{Power} = \sum i(\text{VDD} \times \text{IDD})$$

Where i denotes all the various power supplies on the device, VDD is the voltage for supply i in accordance with the condition, and IDD is the current for supply i.

- I/O supply power is application-dependant, but typically <20% of **VccInt**. During WAIT mode, I/O power supply should draw negligible current unless resistively loaded.

Table 30 Conditions for Power Requirements

	Typical	Power For Thermal Calculations	Maximum Current
Process	Nominal	Nominal +2 sigmas of process variation*	Nominal +6 sigmas of process variation
Voltage	Nominal Vdd	Maximum Operating Vdd	Maximum Vdd

Note

- * The power number for nominal process +2 sigma of process variation is recommended for thermal calculations as it will be the highest power dissipation of almost all parts in almost all applications. The current number for nominal +6 sigma of process variation is recommended for power supply design and is a true worst case.

13.3 Typical Power Consumption

Power consumption in an end application depends on many application-specific factors, such as the characteristics of the code being executed, operating temperature of the CPU, and loads being driven. The power consumption in an actual application can be substantially lower than the maximum guaranteed specification shown.

Downloaded [controlled] by Venkatesh Betageri of IHS on Wednesday, 12 January, 2011 02:34:04 AM

14 AC Electrical Characteristics

14.1 Capacitive Load Deration

Parameter	Symbol	Min	Max	Units	Mode
Load Derate	CLD		2	ns/25pF	LVTTTL
					HSTL

14.2 Clock Parameters

Parameter	Symbol	Test Conditions	Bus Speed				Units
			LVTTTL		HSTL		
			Min	Max	Min	Max	
SysClock High	t _{SCHigh}	Transition ≤ 2ns	3				ns
SysClock Low	t _{SCLow}	Transition ≤ 2ns	3				ns
SysClock Frequency			33.3	133	33.3	200	MHz
SysClock Period	t _{SCP}		7.5	30	5	30	ns
Clock Jitter for SysClock	t _{JitterIn}			±150		±150	ps
SysClock Rise Time	t _{SCRise}			2		1.3	ns
SysClock Fall Time	t _{SCFall}			2		1.3	ns
ModeClock Period	t _{ModeCKP}			256		256	t _{SCP}
JTAG Clock Period	t _{JTAGCKP}		4		4		t _{SCP}

14.3 System Interface Parameters

Parameter ¹	Symbol	Test Conditions	I/O Type				Units
			LVTTTL I/O		HSTL I/O		
			Min	Max	Min	Max	
Data Output ^{2,3,7}	t _{DO}	LVTTTL (V _{CCIO} =3.3V): mode[15:14]=10 (fastest) ^{5,6,7}	0.75	4.5	0.75	2.5	ns
		HSTL (V _{CCIO} =1.5V): mode[108:107:62:15:14]= 11110 (fastest) ^{5,6}					
		LVTTTL (V _{CCIO} =3.3V): mode[15:14]=01 (slowest) ^{5,6,7}	0.75	5.5	0.75	2.75	
		HSTL (V _{CCIO} =1.5V): mode[108:107:62:15:14]= 11101 (slowest) ^{5,6,7}					

Parameter ¹	Symbol	Test Conditions	I/O Type				Units
			LVTTL I/O		HSTL I/O		
			Min	Max	Min	Max	
Data Setup ⁴	t_{DS}^6	t_{rise} = see above table	2.5		1.15		ns
Data Hold ⁴	t_{DH}	t_{fall} = see above table	1.0		0.75		ns

Notes:

1. In LVTTL mode, timings are measured from $0.425 \times V_{CCIO}$ of clock to $0.425 \times V_{CCIO}$ of signal for 3.3V I/O, and from $0.48 \times V_{CCIO}$ of clock to $0.48 \times V_{CCIO}$ of signal for 2.5V I/O. In HSTL mode, timings are measured from the crossing point of **SysClock** and **SysClock*** to 0.75V of the crossing point of the signal.
2. Capacitive load for all LVTTL maximum output timings is 50 pF. Minimum output timings are for capacitive load of 20 pF.
3. Capacitive load for all HSTL minimum and maximum output timings is 20 pF.
4. Data Output timing applies to all signal pins whether tristate I/O or output only.
5. Setup and Hold parameters apply to all signal pins whether tristate I/O or input only.
6. Only mode[108:107:62:15:14]=111110 is tested in HSTL Class I mode during production test.
7. Data shown is for 3.3 V I/O. For 2.5 V I/O derate t_{DO} Max by 0.5 nS, and t_{DO} Min by 0.25 ns.

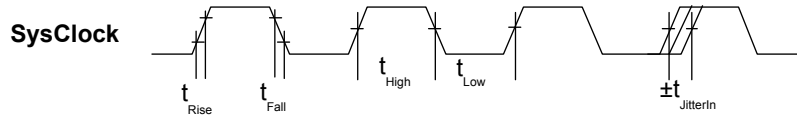
14.4 Boot-Time Interface Parameters

Parameter	Symbol	Min	Max	Units
Mode Data Setup	t_{DS}	4		SysClock cycles
Mode Data Hold	t_{DH}	0		SysClock cycles

15 Timing Diagrams

15.1 Clock Timing

Figure 11 Clock Timing



15.2 System Interface Timing

(SysAD, SysCmd, ValidIn*, ValidOut*, etc.)

Figure 12 Input Timing

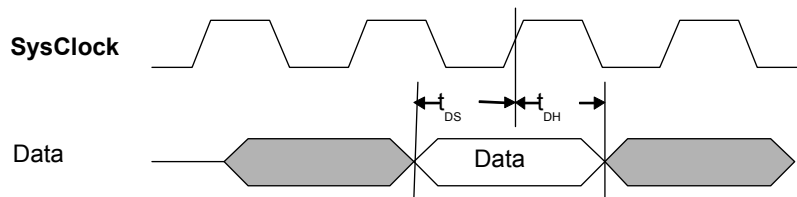
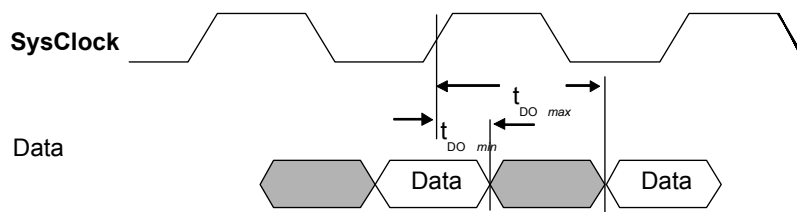


Figure 13 Output Timing



16 Thermal Information

This product is designed to operate over a wide temperature range when used with a heat sink and is suited for commercial applications such as central office equipment.

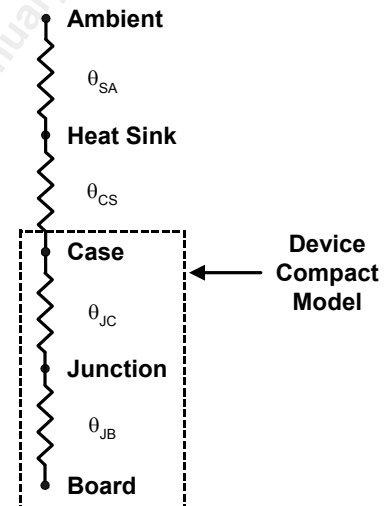
Maximum long-term operating junction temperature (T_J) to ensure adequate long-term life.	88°C
Minimum ambient temperature (T_A)	0°C

Table 31 Device Compact Model²

900 MHz	
Junction-to-Case Thermal Resistance, θ_{JC}	0.27
Junction-to-Board Thermal Resistance, θ_{JB}	6.52
Θ_{JA} (°C/W) (without heat sink)	13.23

Table 32 Heat Sink Requirements

$\theta_{SA} + \theta_{CS}$ ⁴	The sum of $\theta_{SA} + \theta_{CS}$ must be less than or equal to: $[(105 - T_A) / P_D] - \theta_{JC}$ °C/W where: T_A is the ambient temperature at the heat sink location P_D is the operating power dissipated in the package ⁵
	θ_{SA} and θ_{CS} are required for long-term operation

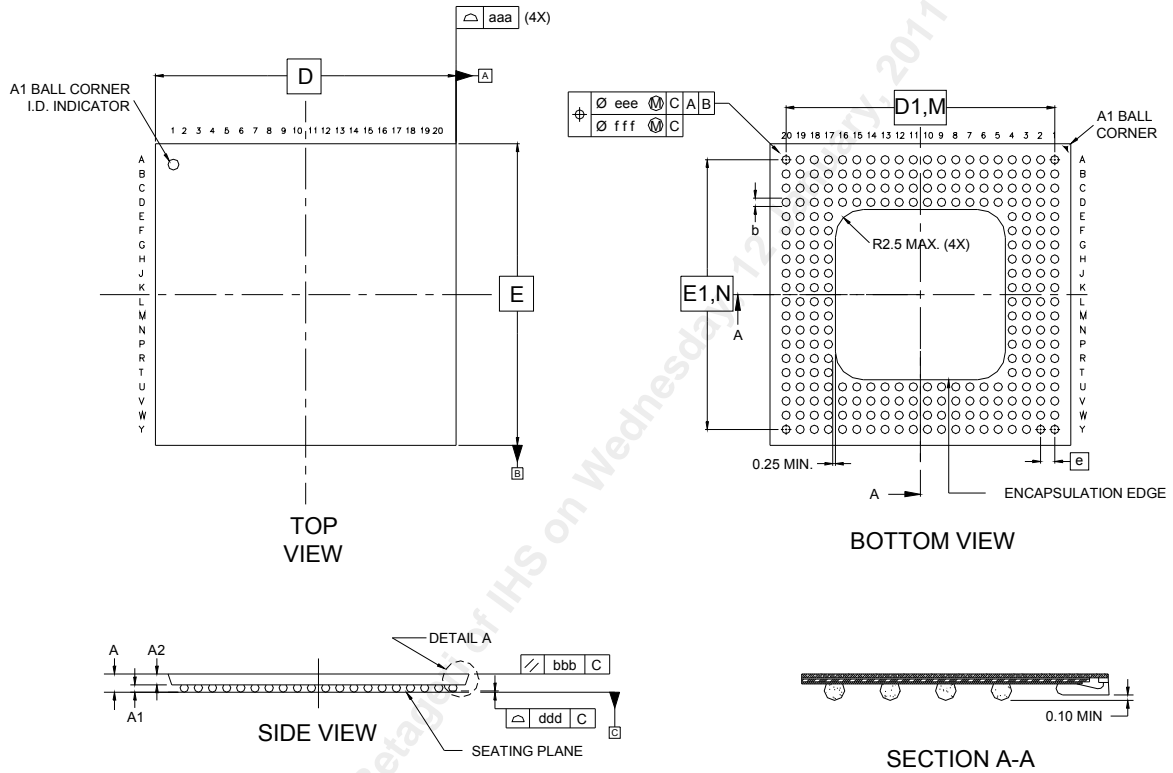


Notes:

1. The minimum ambient temperature requirement for Central Office Equipment approximates the minimum ambient temperature requirement for Commercial Equipment.
2. Short-term is used as defined in Telcordia Technologies Generic Requirements GR-63-Core; for more information about the GR-63-CORE standard, see Telcordia Technologies. *Network Equipment-Building System (NEBS) Requirements: Physical Protection: Telcordia Technologies Generic Requirements GR-63-CORE*. Issue 1. October 1995.
3. θ_{JC} , the junction-to-case thermal resistance, is a measured nominal value plus two sigma. θ_{JB} , the junction-to-board thermal resistance, is obtained by simulating conditions described in JEDEC Standard JESD 51-8; for more information about the JESD51-8 standard, see Electronic Industries Alliance 1999. *Integrated Circuit Thermal Test Method Environmental Conditions -Junction-to-Board: JESD51-8*. October 1999.
4. θ_{SA} is the thermal resistance of the heat sink to ambient. θ_{CS} is the thermal resistance of the heat sink attached material. The maximum θ_{SA} required for the airspeed at the location of the device in the system with all components in place.
5. Power depends upon the operating mode. To obtain power information, refer to the column under thermal in Table 29.

17 Packaging and Pinout Information

17.1 256-pin CSBGA Package Diagram



- NOTES: 1) ALL DIMENSIONS IN MILLIMETERS
 2) DIMENSION aaa DENOTES PACKAGE PROFILE
 3) DIMENSION bbb DENOTES PARALLELISM
 4) DIMENSION ddd DENOTES COPLANARITY
 5) DIAMETER OF SOLDER MASK OPENING IS 0.58 MM (SMD)
 6) PACKAGE COMPLIANT TO JEDEC REGISTERED OUTLINE MO-192
 VARIATION BAL-2 WITH EXCEPTION OF PROFILE TOLERANCE,
 COPLANARITY, AND MAXIMUM OVERALL THICKNESS

PACKAGE TYPE: 256 THERMALLY ENHANCED BALL GRID ARRAY - CSBGA+															
BODY SIZE: 27 x 27 x 1.62mm															
Dim.	A	A1	A2	D	D1	E	E1	M, N	e	b	aaa	bbb	ddd	eee	fff
Min.	1.47	0.55	0.92	-	-	-	-	-	-	-	-	-	-	-	-
Nom.	1.62	0.65	0.97	27.00 BSC	24.13 BSC	27.00 BSC	24.13 BSC	20x20	1.27 BSC	0.75	-	-	-	-	-
Max.	1.77	0.75	1.02	-	-	-	-	-	-	-	0.10	0.25	0.15	0.30	0.15

17.2 256-pin CSBGA Alphanumerical Pinout

Pin	Function	Pin	Function	Pin	Function	Pin	Function
A1	VccIO	B19	VccIO	D17	VccIO	J3	VccInt
A2	Vss	B20	Vss	D18	Do Not Connect	J4	VccIO
A3	Vss	C1	Vss	D19	Vss	J17	VccIO
A4	Do Not Connect	C2	Vss	D20	Do Not Connect	J18	SysAD54
A5	SysAD35	C3	VccIO	E1	SysAD5	J19	SysAD22
A6	Vss	C4	Do Not Connect	E2	Do Not Connect	J20	Vss
A7	SysAD33	C5	Do Not Connect	E3	VccInt	K1	SysAD41
A8	SysAD32	C6	Do Not Connect	E4	VccIO	K2	SysAD10
A9	Vss	C7	SysAD34	E17	VccIO	K3	SysAD42
A10	SysADC1	C8	VccInt	E18	Do Not Connect	K4	SysAD11
A11	HSTL_Sel*	C9	SysAD0	E19	Do Not Connect	K17	SysAD53
A12	Vss	C10	SysADC4	E20	SysAD59	K18	SysAD21
A13	SysADC2	C11	SysADC7	F1	Vss	K19	SysAD52
A14	SysAD62	C12	VccInt	F2	SysAD36	K20	SysAD20
A15	Vss	C13	SysAD31	F3	SysAD4	L1	SysAD43
A16	SysAD60	C14	SysAD61	F4	VccInt	L2	SysAD44
A17	Do Not Connect	C15	VccInt	F17	VccInt	L3	SysAD12
A18	Vss	C16	Do Not Connect	F18	SysAD27	L4	VccInt
A19	Vss	C17	Do Not Connect	F19	SysAD58	L17	VccInt
A20	VccIO	C18	VccIO	F20	Vss	L18	SysAD51
B1	Vss	C19	Vss	G1	SysAD38	L19	SysAD19
B2	VccIO	C20	Vss	G2	SysAD6	L20	SysAD50
B3	Vss	D1	Do Not Connect	G3	SysAD37	M1	Vss
B4	Vss	D2	Vss	G4	VccInt	M2	SysAD13
B5	Do Not Connect	D3	Do Not Connect	G17	VccInt	M3	SysAD45
B6	SysAD3	D4	VccIO	G18	SysAD26	M4	VccIO
B7	SysAD2	D5	VccIO	G19	SysAD57	M17	VccIO
B8	SysAD1	D6	Do Not Connect	G20	SysAD25	M18	SysAD18
B9	SysADC5	D7	VccInt	H1	SysAD7	M19	SysAD49
B10	SysADC0	D8	VccInt	H2	SysAD39	M20	Vss
B11	SysADC3	D9	VccIO	H3	SysAD40	N1	SysAD14
B12	SysADC6	D10	VccInt	H4	SysAD8	N2	SysAD46
B13	VREF_In	D11	VccInt	H17	SysAD24	N3	VccInt
B14	SysAD30	D12	VccIO	H18	SysAD56	N4	SysAD47
B15	SysAD29	D13	SysAD63	H19	SysAD55	N17	VccInt
B16	Do Not Connect	D14	VccInt	H20	SysAD23	N18	SysAD48
B17	Vss	D15	SysAD28	J1	Vss	N19	SysAD16
B18	Vss	D16	VccIO	J2	SysAD9	N20	SysAD17

256-pin CSBGA Alphanumerical Pinout cont'd.

Pin	Function	Pin	Function	Pin	Function
P1	SysAD15	U15	INT3*	W13	SysCmd5
P2	RspSwap*	U16	VccIO	W14	SysCmdP
P3	PAck*	U17	VccIO	W15	VccInt
P4	VccInt	U18	INT6*	W16	INT1*
P17	ColdReset*	U19	Vss	W17	Vss
P18	VccOK	U20	INT7*	W18	Vss
P19	BigEndian	V1	Vss	W19	VccIO
P20	Reset*	V2	Vss	W20	Vss
R1	Vss	V3	VccIO	Y1	VccIO
R2	Do Not Connect	V4	RDType	Y2	Vss
R3	JTDI	V5	RdRdy*	Y3	Vss
R4	JTCK	V6	VccP	Y4	ModelIn
R17	VccInt	V7	SysClock*	Y5	ValidOut*
R18	ExtRqst*	V8	VccInt	Y6	Vss
R19	NMI*	V9	Do Not Connect	Y7	VccP
R20	Vss	V10	VREF_In	Y8	Do Not Connect
T1	PRqst*	V11	VccInt	Y9	Vss
T2	JTDO	V12	SysCmd3	Y10	Do Not Connect
T3	VccIO	V13	SysCmd6	Y11	SysCmd0
T4	JTRST*	V14	VccInt	Y12	Vss
T17	VccIO	V15	INT2*	Y13	SysCmd4
T18	VccInt	V16	INT5*	Y14	SysCmd8
T19	INT9*	V17	INT4*	Y15	Vss
T20	INT8*	V18	VccIO	Y16	VccJ
U1	ModeClock	V19	Vss	Y17	INT0*
U2	Vss	V20	Vss	Y18	Vss
U3	JTMS	W1	Vss	Y19	Vss
U4	VccIO	W2	VccIO	Y20	VccIO
U5	JTAGSEL	W3	Vss		
U6	ValidIn*	W4	Vss		
U7	VssP	W5	WrRdy*		
U8	VccInt	W6	Release*		
U9	VccIO	W7	SysClock		
U10	VccInt	W8	VccInt		
U11	VccInt	W9	Do Not Connect		
U12	VccIO	W10	Do Not Connect		
U13	SysCmd7	W11	SysCmd1		
U14	VccInt	W12	SysCmd2		

17.3 256-pin CSBGA Alphabetical Pinout

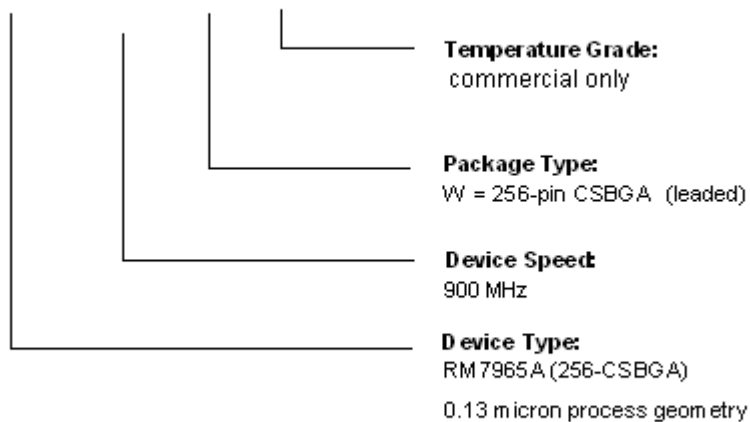
Pin	Function	Pin	Function	Pin	Function	Pin	Function
P19	BigEndian	R4	JTCK	H20	SysAD23	C14	SysAD61
P17	ColdReset*	R3	JTDI	H17	SysAD24	A14	SysAD62
A17	Do Not Connect	T2	JTDO	G20	SysAD25	D13	SysAD63
B5	Do Not Connect	U3	JTMS	G18	SysAD26	B10	SysADC0
C5	Do Not Connect	T4	JTRST*	F18	SysAD27	A10	SysADC1
C17	Do Not Connect	U1	ModeClock	D15	SysAD28	A13	SysADC2
D1	Do Not Connect	Y4	Modeln	B15	SysAD29	B11	SysADC3
C6	Do Not Connect	R19	NMI*	B14	SysAD30	C10	SysADC4
D6	Do Not Connect	P3	PAck*	C13	SysAD31	B9	SysADC5
D18	Do Not Connect	T1	PRqst*	A8	SysAD32	B12	SysADC6
E2	Do Not Connect	V5	RdRdy*	A7	SysAD33	C11	SysADC7
E18	Do Not Connect	V4	RDType	C7	SysAD34	W7	SysClock
D3	Do Not Connect	W6	Release*	A5	SysAD35	V7	SysClock*
E19	Do Not Connect	P20	Reset*	F2	SysAD36	Y11	SysCmd0
A4	Do Not Connect	P2	RspSwap*	G3	SysAD37	W11	SysCmd1
B16	Do Not Connect	C9	SysAD0	G1	SysAD38	W12	SysCmd2
C4	Do Not Connect	B8	SysAD1	H2	SysAD39	V12	SysCmd3
C16	Do Not Connect	B7	SysAD2	H3	SysAD40	Y13	SysCmd4
D20	Do Not Connect	B6	SysAD3	K1	SysAD41	W13	SysCmd5
V9	Do Not Connect	F3	SysAD4	K3	SysAD42	V13	SysCmd6
W9	Do Not Connect	E1	SysAD5	L1	SysAD43	U13	SysCmd7
R2	Do Not Connect	G2	SysAD6	L2	SysAD44	Y14	SysCmd8
W10	Do Not Connect	H1	SysAD7	M3	SysAD45	W14	SysCmdP
Y10	Do Not Connect	H4	SysAD8	N2	SysAD46	U6	ValidIn*
Y8	Do Not Connect	J2	SysAD9	N4	SysAD47	Y5	ValidOut*
R18	ExtRqst*	K2	SysAD10	N18	SysAD48	F17	VccInt
A11	HSTL_Sel*	K4	SysAD11	M19	SysAD49	G17	VccInt
Y17	INT0*	L3	SysAD12	L20	SysAD50	L17	VccInt
W16	INT1*	M2	SysAD13	L18	SysAD51	N17	VccInt
V15	INT2*	N1	SysAD14	K19	SysAD52	D10	VccInt
U15	INT3*	P1	SysAD15	K17	SysAD53	D14	VccInt
V17	INT4*	N19	SysAD16	J18	SysAD54	C15	VccInt
V16	INT5*	N20	SysAD17	H19	SysAD55	D7	VccInt
U18	INT6*	M18	SysAD18	H18	SysAD56	D11	VccInt
U20	INT7*	L19	SysAD19	G19	SysAD57	E3	VccInt
T20	INT8*	K20	SysAD20	F19	SysAD58	J3	VccInt
T19	INT9*	K18	SysAD21	E20	SysAD59	N3	VccInt
U5	JTAGSEL	J19	SysAD22	A16	SysAD60	C8	VccInt

256-pin CSBGA Alphabetical Pinout cont'd.

Pin	Function	Pin	Function	Pin	Function
C12	VccInt	Y1	VccIO	C20	Vss
D8	VccInt	V18	VccIO	F20	Vss
F4	VccInt	W2	VccIO	J20	Vss
G4	VccInt	T3	VccIO	M20	Vss
L4	VccInt	V3	VccIO	R1	Vss
R17	VccInt	W19	VccIO	V1	Vss
T18	VccInt	U4	VccIO	W1	Vss
U10	VccInt	U12	VccIO	W17	Vss
U14	VccInt	U16	VccIO	Y9	Vss
V14	VccInt	Y20	VccIO	U2	Vss
U11	VccInt	Y16	VccJ	V2	Vss
V11	VccInt	P18	VccOK	W18	Vss
W15	VccInt	V6	VccP	Y2	Vss
P4	VccInt	Y7	VccP	Y6	Vss
U8	VccInt	B13	VREF_In	Y18	Vss
V8	VccInt	V10	VREF_In	U19	Vss
W8	VccInt	A9	Vss	V19	Vss
A1	VccIO	B1	Vss	W3	Vss
D5	VccIO	B17	Vss	Y3	Vss
D9	VccIO	C1	Vss	Y15	Vss
D17	VccIO	F1	Vss	Y19	Vss
E17	VccIO	J1	Vss	R20	Vss
J17	VccIO	M1	Vss	V20	Vss
M17	VccIO	A2	Vss	W4	Vss
B2	VccIO	A6	Vss	W20	Vss
C18	VccIO	A18	Vss	Y12	Vss
B19	VccIO	B18	Vss	U7	VssP
C3	VccIO	C2	Vss	W5	WrRdy*
A20	VccIO	D2	Vss		
D4	VccIO	A3	Vss		
D12	VccIO	A15	Vss		
D16	VccIO	A19	Vss		
E4	VccIO	B3	Vss		
J4	VccIO	C19	Vss		
M4	VccIO	D19	Vss		
T17	VccIO	A12	Vss		
U9	VccIO	B4	Vss		
U17	VccIO	B20	Vss		

18 Ordering Information

RM7965A - 900 UI



Valid Combinations

RM7965A-900UI (leadless)

End of Document

Downloaded [controlled] by Venkatesh Betageri of IHS on Wednesday, 12 January, 2011 02:34:04 AM