LSI LOGIC

R2000
High Performance RISC Microprocessor
Preliminary

Description

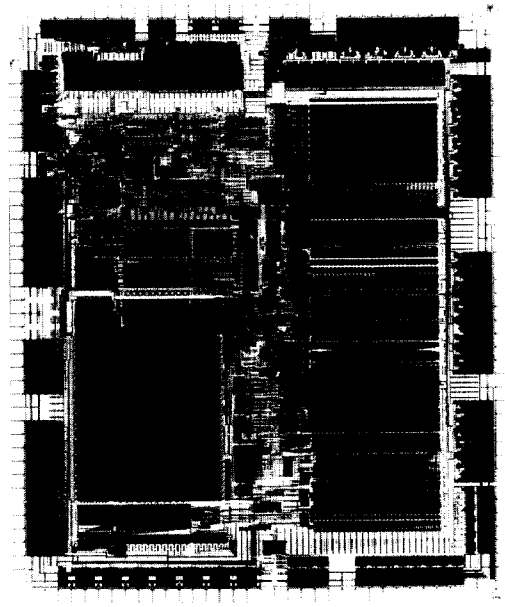
The R2000 CPU is a high-speed HCMOS implementation of the MIPS RISC (Reduced Instruction Set Computer) microprocessor architecture. The MIPS architecture was initially developed at Stanford University under the auspices of DARPA. The R2000 is an extension of the Stanford MIPS architecture developed by MIPS Computer Systems Inc. This architecture makes possible a microprocessor that can execute instructions for high-level language programs at rates approaching one instruction per processor clock. It supports up to three tightly coupled coprocessors including the single chip R2010 Floating-Point Accelerator.

Features

- Reduced Instruction Set Computer (RISC) architecture
  - MIPS instruction set
  - Simple 32-bit instructions, single addressing mode
  - Register-to-register, load-store operation
  - All instructions (except MPY and DIV) execute in a single cycle
- High performance
  - Fast instruction cycle with 5-stage pipeline
  - Efficient handling of pipeline stalls and exceptional events
- Two speed versions
  - R2000/12 12.5 MHz 8 VAX mips equivalents
  - R2000/16 16.7 MHz 10 VAX mips equivalents
- Optional devices tightly coupled for high performance
  - R2010 Floating Point Accelerator (FPA)
  - R2020 Write Buffers (WB)
- 32 general-purpose registers
- On-chip cache control
  - Separate external instruction and data cache memories
  - From 4 to 64 K bytes each
  - Both cache memories accessed during a single CPU cycle
  - Dual cache bandwidth up to 133 Mbytes/second
  - Uses standard SRAMs
    - R2000/12 35 ns access time
    - R2000/16 25 ns access time
- On-chip memory management unit (MMU)
  - Fully-associative, 64-entry translation lookaside buffer (TLB)
  - Supports 4-Gbyte virtual address space
- Multi-tasking support
  - User and kernel (supervisor) modes
- Tightly coupled coprocessor interface
  - Generates all addresses and handles memory interface control
  - Supports up to three external coprocessors
- Strong, integrated software support
  - UMIPS operating system
    - System V.3,4.3 BSD
  - Optimizing compilers
    - C, Ada (Vertix), FORTRAN, LPI-COBOL, Pascal, LPI-PL-1
- 144 ceramic pin grid array package

R2000 CPU Chip Photo

The full-custom 32-bit VLSI CMOS Reduced Instruction Set Computer includes thirty-two 32-bit registers, on-chip TLB, memory management unit, and cache control circuitry.

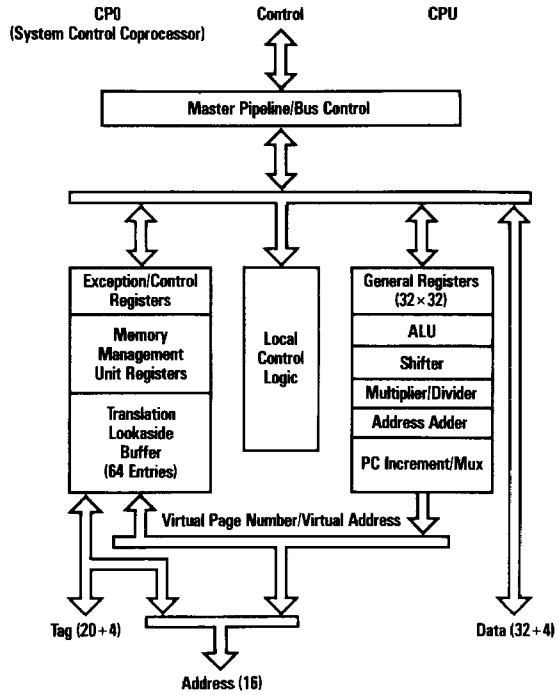


R2000
High Performance
RISC Microprocessor
Preliminary

Introduction

The R2000 processor consists of two tightly coupled processors implemented on a single chip. In addition to a RISC (Reduced Instruction Set Computer) CPU there is a system control coprocessor (CP0) which contains a TLB (translation lookaside buffer) and control registers to support a virtual memory subsystem.

Block Diagram



R2000 Instructions

All R2000 instructions are 32 bits in length. To simplify instruction decoding, only three instruction formats are supported (immediate, jump and register). The instruction set can be divided into the following groups.

- **Load/Store** instructions move data between memory and the general registers. All instructions are then executed on values stored in the general registers. There are no operations performed on operands in cache or main memory. Loads and stores are all I-type instructions since the only addressing mode supported is base register + 16-bit immediate offset.
- **Computational** instructions perform arithmetic, logical and shift operations on values in registers. These can be R-type (both operand are registers) or I-type (one operand is a 16-bit immediate) instruction formats.
- **Jump and Branch** instructions change program flow. Jumps are always to an absolute 26-bit address (J-type format for subroutine calls) or 32-bit register byte addresses (R-type for returns and dispatches). Branches have 16-bit offsets relative to the program counter (I-type). Jump and link instructions save a return address in register 31.
- **Coprocessor** instructions perform operations in the coprocessors. Coprocessor loads and stores are I-type, or have coprocessor-dependent formats. Coprocessor 0 instructions perform operations on the CP0 registers to manipulate memory management and exception handling facilities.
- **Special** instructions perform a variety of tasks including movement of data between special and general registers, trap and breakpoint. They are always R-type.

The R2000 CPU provides 32 general purpose 32-bit registers, a 32-bit program counter and two 32-bit registers which hold the results of integer multiply and divide operations. The functions traditionally provided by a program status word (PSW) register are handled by the status and cause registers in the CP0.

The R2000 supports a user and kernel (supervisor) mode. The R2000 normally operates in user mode until an exception is detected forcing it into the kernel mode. It remains in kernel mode until a restore from exception (RFE) instruction is executed. The 4-Gbyte address space is divided into 2 Gbytes for users and 2 Gbytes for the kernel.

Table 1. Instruction Summary

OP	Description
	Load/Store Instructions
LB	Load Byte
LBU	Load Byte Unsigned
LH	Load Halfword
LHU	Load Halfword Unsigned
LW	Load Word
LWL	Load Word Left
LWR	Load Word Right
SB	Store Byte
SH	Store Halfword
SW	Store Word
SWL	Store Word Left
SWR	Store Word Right
	Arithmetic Instructions (ALU Immediate)
ADDI	Add Immediate
ADDIU	Add Immediate Unsigned
SLTI	Set on Less than Immediate
SLTIU	Set on Less than Immediate Unsigned
ANDI	AND Immediate
ORI	OR Immediate
XORI	Exclusive OR Immediate
LUI	Load Upper Immediate
	Arithmetic Instructions (3-operand, register-type)
ADD	Add
ADDU	Add Unsigned
SUB	Subtract
SUBU	Subtract Unsigned
SLT	Set on Less than
SLTU	Set on Less than Unsigned
AND	AND
OR	OR
XOR	Exclusive OR
NOR	NOR
	Shift Instructions
SLL	Shift Left Logical
SRL	Shift Right Logical
SRA	Shift Right Arithmetic
SLLV	Shift Left Logical Variable
SRLV	Shift Right Logical Variable
SRAV	Shift Right Arithmetic Variable

OP	Description
	Multiply/Divide Instructions
MULT	Multiply
MULTU	Mulitply Unsigned
DIV	Divide
DIVU	Divide Unsigned
MFHI	Move From HI
MTHI	Move to HI
MFLO	Move From LO
MTLO	Move to LO
	Jump and Branch Instructions
J	Jump
JAL	Jump and Link
JR	Jump to Register
JALR	Jump and Link Register
BEQ	Branch on Equal
BNE	Branch on Not Equal
BLEZ	Branch on Less than or Equal to Zero
BGTZ	Branch on Greater than Zero
BLTZ	Branch on Less than Zero
BGEZ	Branch on Greater than or Equal to Zero
BLTZAL	Branch on Less than Zero and Link
BGEZAL	Branch on Greater than or Equal to Zero and Link
	Special Instructions
SYSCALL	System Call
BREAK	Break
	Coprocessor Instructions
LWCz	Load Word to Coprocessor
SWCz	Store Word from Coprocessor
MTCz	Move to Coprocessor
MFCz	Move from Coprocessor
CTCz	Move Control to Coprocessor
CFCz	Move Control from Coprocessor
COPz	Coprocessor Operation
BCzT	Branch on Coprocessor z True
BCzF	Branch on Coprocessor z False
	System Control Coprocessor (CP0) Instructions
MTC0	Move to CP0
MFC0	Move from CP0
TLBR	Read Indexed TLB Entry
TLBWI	Write Indexed TLB Entry
TLBWR	Write Random TLB Entry
TLBP	Probe TLB for Matching Entry
RFE	Restore from Exception

R2000 Register Organization

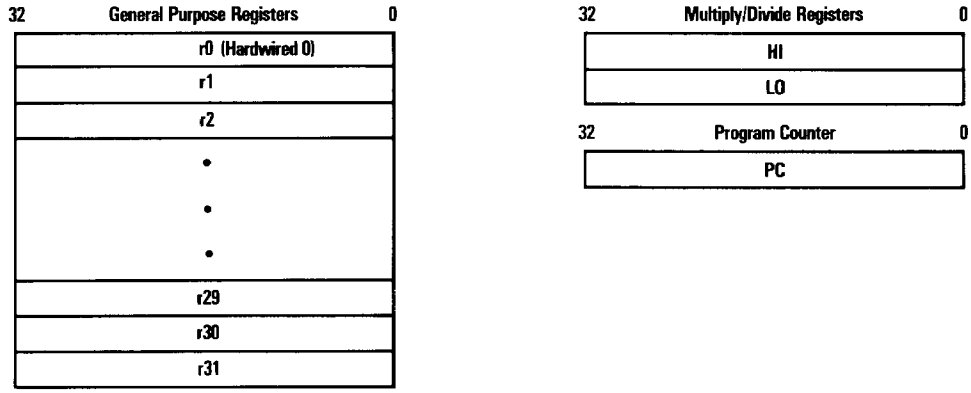


Figure 1. CPU Registers

System Control Coprocessor (CP0)

The R2000 can operate with up to four tightly coupled coprocessors (CP0 thru CP3). The system control coprocessor (CP0) is on the R2000 chip and supports the virtual memory system and exception handling functions of the R2000. The virtual memory system is implemented using a translation lookaside buffer (TLB) and a group of programmable registers.

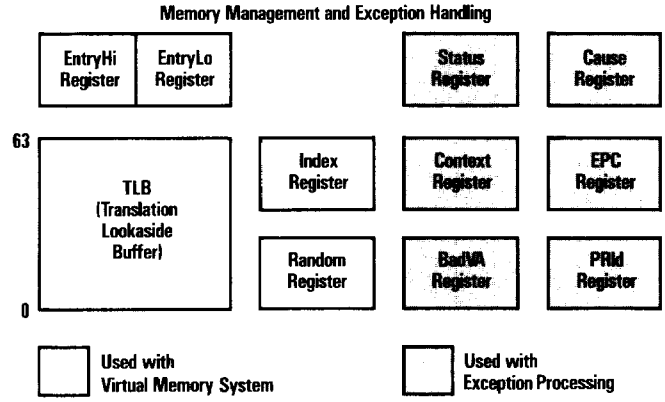


Figure 2. CP0 Registers

Coprocessors

Three types of coprocessor instructions are supported: loads and stores, internal operations, and moves between the coprocessors. The R2000 coprocessors and the main processor share the same instruction stream. Coprocessor instructions are not explicitly passed to a coprocessor by the R2000. Instead, coprocessors continuously monitor the data bus, receive instruction/data pairs, and decode valid instructions at the same rate as the main processor.

Memory Management System

The R2000 supports interfaces-to-cache memory and main memory. Often-used operands and intructions are placed into cache memory where the processor can access them quickly. Two direct-mapped caches for instructions (I-cache) and data (D-cache) can range in size from 4 Kbytes to 64 Kbytes. Cache memory access operations take a single cycle to complete. A main memory interface supports reads and writes from/to main (non-cache) memory.

The R2000 has an addressing range of 4 Gbytes (2 Gbytes for the user, 2 Gbytes for the kernel). Since most systems implement physical memory sizes under 4 Gbytes, the R2000 provides for the logical expansion of memory space by translating addresses composed in a large virtual address space into available physical memory address.

The on-chip translation lookaside buffer provides very fast virtual memory access and is well matched to the requirements of multi-tasking operating systems. The fully-associative TLB contains 64 entries, each of which maps a 4-Kbyte page, with controls for read/write access, cacheability and process identification.

The D-cache can be isolated from main memory. The processor also allows swapping of the instruction and data caches. Both operations are used to support cache flushing, diagnostics and trouble shooting.

Pipeline Architecture

The execution of a single R2000 instruction consists of five primary steps:

IF	Fetch the instruction (I-cache).
RD	Read any required operands from CPU registers while decoding the instruction.
ALU	Perform the required operation of instruction operands.
MEM	Access memory (D-cache).
WB	Write results back to the register file.

Each of these steps requires an average of one CPU cycle. The R2000 uses a five-stage pipeline to achieve an instruction execution rate approaching one instruction per CPU cycle. This pipeline operates efficiently because different CPU resources (address and data bus accesses, ALU operations, register accesses, etc.) are utilized on a non-interfering basis. Even load and store operations execute in a single cycle.

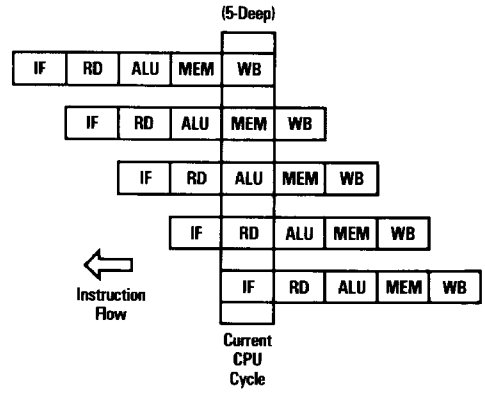


Figure 3. Instruction Pipeline

Memory System Hierarchy

The R2000 supports a high-performance memory hierarchy which centers on the use of external caches. Separate data and instruction caches allow the processor to obtain data and instructions at the CPU cycle rate. These caches are built using commercially available high-speed static RAMS. To ensure data consistency, all data written into cache should be written through into main memory. Optional R2020 write buffers are four-deep, 32-bit write buffers which capture output data from the CPU and ensure its passage on to main memory.

Software and Development Support

The UMIPS operating system is licensable in compiled form from LSI Logic and in source code form from MIPS Computer Systems, Inc. UMIPS is available in both System V.3 and 4.3BSD. UMIPS includes the full complement of UNIX software development utilities such as text editing, source code checking, source code debugging, performance analysis, document formatting, software project management and compiler generation. Compilers for C, Pascal, FORTRAN, Ada, COBOL and PL-1 are available from LSI Logic or third parties.

Board-level products are also available to use as machine code compatible execution vehicles to verify correctness and performance of machine-level instructions.

For software and applications development the M/800 and M/1000 systems are available.

Pin Descriptions

(**Note:** an asterisk* indicates an Active-LOW signal)

Data Bus (D31:00)
The 32-bit bidirectional data bus carries all data and instructions between the CPU, caches, main memory and coprocessors.

Data Parity (DataP3:0)
The 4-bit bidirectional data parity bus provides even parity for each of the four bytes of the 32-bit data bus. A data parity error is treated as a cache miss.

Address Low Bus (AdrLo15-00)
The 16-bit AdrLo output carries low-order address bits to the caches and memory subsystem. Only the 14 most significant bits are used to access cache locations. All 16 bits are used for main memory accesses along with 16 bits of the tag bus to form a 32-bit physical memory address. The AdrLo bus is set to high impedance when reset* is asserted or when the processor is brought out of reset in the test state.

Tag Bus (Tag31-12)
The 20-bit tag bus transfers cache tags into the CPU during cache reads. During cache writes, the tag bus carries tag bits into the cache. For main memory accesses, the 16 most significant bits are combined with the AdrLo bus to form a 32-bit physical address.

Tag Valid (TagV)
TagV carries the valid bit between the R2000 and the caches. During write operation, TagV is HIGH when writing a full 32-bit word to cache and LOW otherwise. During cache reads, TagV is used as one of the criteria in determining whether a cache hit has occurred.

Tag Parity Bus (TagP2-0)
This 3-bit bidirectional bus contains even parity for Tag31–12 and TagV. Tag parity is generated for cache writes and checked during caches reads. A tag parity error is treated as a cache miss.

I-Cache Read (IRd)
D-Cache Read (DRd)
These outputs are asserted during I-cache and D-cache read operations to enable the outputs of the cache RAMs.

I-Cache Write (IWr)
D-Cache Write (DWr)
These outputs are asserted during I-cache and D-cache write operations. These signals are typically used as the write-enable or write-strobe input to the cache RAMs.

I-Cache Latch Clock (IClk*)
D-Cache Latch Clock (DClk*)
These outputs are asserted during every cycle. These signals are used to latch addresses into external latches and onto the address bus for the cache RAMs.

Access Type (AccTyp2-0)
AccTyp1 and AccTyp0 indicate the data size for memory accesses and processor-coprocessor transfers as shown below.

Table 2. Access Type Bit Decoding

AccTyp 1 0	Data Size
0 0	Byte (8 bits)
0 1	Half-word (16 bits)
1 0	Three bytes (24 bits)
1 1	Word (32 bits)

AccTyp2 indicates the purpose of an access: During stall cycles, when main memory read is as a result of an I-cache miss, (AccTyp2 is HIGH) or as a result of a D-cache miss (AccTyp2 is LOW).

During run cycles, when the processor data bus will be used during the current cycle, AccTyp2 is LOW, otherwise AddTyp2 is HIGH.

Memory Write (MemWr*)
The MemWr* output is LOW when the processor is performing any write-to-memory. This signal indicates that the tag and address-low buses contain a valid byte address.

Memory Read (MemRd*)
The MemRd* input is LOW when the processor is performing any read-from-memory. This indicates that the tag and address-low buses contain a valid byte address.

Write Busy (WrBusy)
The main memory subsystem places the WrBusy* input LOW to inform the processor that it is not able to accept write data. If the processor needs to perform a write operation while WrBusy* is LOW, the processor stalls until WrBusy* becomes HIGH.

Read Busy (RdBusy)
The main memory subsystem places RdBusy input HIGH to indicate that it is not ready to supply read data requested by the processor. Whenever there is a cache miss, the processor always initiates a read stall while it performs a main memory read. When RdBusy is HIGH it causes the processor to remain in a main memory read stall until it goes LOW.

Pin Descriptions
(Continued)

Run*
The Run* input is LOW when the processor is performing a run cycle and is HIGH when the processor is performing stall cycles.

Exception*
The Exception* output is LOW when the processor is responding to an exception and its instruction pipeline has been disrupted. Coprocessors are expected to terminate any instructions in their pipelines.

Coprocessor Busy (CpBusy*)
The input is set LOW by the coprocessor if it needs more time to resolve a data dependency in the instruction stream. When this occurs, the processor initiates a stall which is terminated when CpBusy* goes HIGH.

Coprocessor Condition (CPCond3-0)
The four Coprocessor Condition inputs are generated by up to four coprocessors and used by the R2000 as condition inputs and are tested during coprocessor branch instructions. The corresponding coprocessor usable bit (Cu3..Cu0) in the status register must be set in order to test one of these condition inputs. Certain software that uses the floating-point coprocessor expects that the CpCond1 input is driven by the R2010 floating-point coprocessor.

BusError*
This input indicates that a bus error (such as a bus time-out or invalid physical address) has occurred during a RdBusy or WrBusy* stall and causes either a data or instruction bus error exception. The BusError* input is to be used only with synchronous events such as cache miss refills, uncached references and unbuffered writes. A bus error resulting from a buffered write must be signaled using one of the interrupt inputs since the processor is not in a stall and the address that caused the bus error may not still be available to the processor.

Reset*
Reset* is the synchronous initialization input. It must be LOW for a minimum of six cycles to guarantee correct processor initialization, and it must go HIGH with the R2000 clocks. When Reset* is LOW, the processor initiates a non-maskable exception and subsequently proceeds to reinitialize the system using a predefined bootstrap routine.

Interrupt 0 (Intr0*)
When Reset* is HIGH, the value of Intr0* determines byte ordering or Endianness. A HIGH results in a Little Endian ordering and a LOW results in Big Endian ordering.

Interrupt 1 (Intr1*)
When Reset* is HIGH, a LOW on Intr1* causes the processor to place all outputs into high impedance to allow external logic to drive signals for board-level testing.

Interrupt 2 (Intr2*)
When Reset* is LOW, the value of Intr2* determines whether caches are presumed present for instructions and data.

Interrupt 3 (Intr3*)
When Reset* is LOW, a LOW on Intr3* causes the processor to place its data and tag outputs into high impedance during write-busy and coprocessor-busy stalls. If Intr3* is HIGH during reset, the data and tag buses are driven during phase 2 of stall cycles. For designs that do not use buses during such stalls, enabling the bus drive prevents the buses from floating for extended periods and avoids overall system design problems.

Interrupt 4 (Intr4*)
When Reset* is LOW, a LOW value of Intr4* causes the processor to insert additional phase delay into its input clock paths. This allows coprocessors to phase lock to the processor and minimize skew.

Interrupt 5 (Intr5*)
Intr5* must be held HIGH during phase 2 while Reset* is HIGH. This will maintain compatibility with future product revisions.

SysOut*, CpSync*
Synchronizing Clock Outputs.

Clk2xSys, Clk2xSmp, Clk2xRd, Clk2xPhi
Four clock inputs. These can be adjusted to obtain optimal positioning of cache interface signals. The relative differences between the clocks are more important than the absolute clock timing. These differences are used to establish the parameters for cache timing.

Operating Parameters

Absolute Maximum Ratings[1]

Parameter	Description	Min	Max	Units
VCC	Supply Voltage	−0.5	+7.0	V
VIN	Input Voltage	−0.5[2]	+7.0	V
TST	Storage Temperature	−65	+150	C
TA	Operating Temperature	0	+70	C
CLD	Load Capacitance on Any Pin		100	pF

Notes:
1. Operation beyond the limits set forth in this table may impair the useful life of the device.
2. VIN Min. = −3.0 V for pulse width less than 15 ns.
3. Not more than one output should be shorted at a time. Duration of the short should not exceed 30 seconds.

Operating Range

Range	Ambient Temperature	VCC
Commercial	0°C to 70°C	5V±5%

DC Characteristics

Parameter	Description	Test Conditions	12.5 MHz		16.67 MHz		Units
			Min	Max	Min	Max	
VOH	Output High Voltage	VCC = Min. IOH = −4 mA	3.5		3.5		V
VOL	Output Low Voltage	VCC = Min. IOL = 4 mA		0.4		0.4	V
VIH	Input High Voltage		2.0	VCC+0.5	2.0	VCC+0.5	V
VIL	Input Low Voltage		−0.5[1]	0.8	−0.5[1]	0.8	V
VIHS	Input High Voltage		−2.5[2]	VCC+.5	3.0[2]	VCC+.5	V
VILS	Input Low Voltage		−0.5[2]	0.4	−0.5[2]	0.4	V
CIn	Input Capacitance		10		10		pF
COut	Output Capacitance		10		10		pF
IDD	Operating Current	VCC = Min.		250		300	mA

Notes:
1. VIL Min. = −3.0 V for pulse width less than 15 ns.
2. VIHS and VILS apply to Clk2xSys, Clk2xSmp, Clk2xRd, Clk2xPhi, CpBusy, and Reset*.

AC Specifications

Tables 3 through 5 list the preliminary AC electrical specifications for the R2000. All timings are referenced to 1.5 V. All output timings assume 25 pF of capacitive load. Output timings should be derated where appropriate using the values provided in Table 6.

Table 3. Clock Parameters (Refer to Figure 4)

Parameter	Symbol	Test Conditions	12.5 MHz		16.667 MHz		Units
			Min	Max	Min	Max	
Input Clock High	TCkHigh	Transition ≤ 5 ns	16		12		ns
Input Clock Low	TCkLow	Transition ≤ 5 ns	16		12		ns
Clock Period	TCkP		40	1000	30	1000	ns
Clk2xSys to Clk2xSmp	TSys-Smp		0	tCyc ÷ 4	0	tCyc ÷ 4	ns
Clk2xSmp to Clk2xRd	TSmp-Rd		0	tCyc ÷ 4	0	tCyc ÷ 4	ns
Clk2xSmp to Clk2xPhi	TSmp		11	tCyc ÷ 4	9	tCyc ÷ 4	ns
The clock parameters apply to all four 2xClocks: Clk2xSys, Clk2xSmp, Clk2xRd, and Clk2xPhi.							

Table 4. Run Operation Parameters (Refer to Figures 5–8)

Parameter	Load (pF)	Symbol	12.5 MHz		16.667 MHz		Units	Offset from SysOut*
			Min	Max	Min	Max		
Data/Tag Valid	25	tDVal	2	3.5	2	3	ns	TSys
Data/Tag Enable		tDEn	−1	−2.5	−1	−2	ns	TSys
Data/Tag Disable		tDDis	0	−1	0	−1	ns	TRd-TSys
Write Delay	25	tWrDly	0	7.5	0	5	ns	TSmp-TSys
Data Setup		tDS	11.5		9		ns	TSmp-TSys
Data Hold		tDH	−4		−4		ns	TSmp-TSys
CpBusy Setup		tCBS	15		13		ns	TSmp-TSys
CpBusy Hold		tCPH	−4		−4		ns	TSmp-TSys
Access Type [1:0]	25	tAcTy	1	10	1	7	ns	TSys
Access Type [2]	25	tAcTy2	1	20	1	17	ns	TSys
Memory Write	25	tMWr	1	10	1	7	ns	TSys
Exception	25	tExc	1	10	1	7	ns	TSys

Table 5. Stall Operation Parameters

Parameter	Load (pF)	Symbol	12.5 MHz		16.667 MHz		Units	Offset from SysOut*
			Min	Max	Min	Max		
Address Valid	25	tSAVal		38		30	ns	TSys
Access Type	25	tSAcTy		35		27	ns	TSys
Memory Read Initiate	25	tMRdI	1	35	1	27	ns	TSys
Memory Read Terminate	25	tMRdT	1	10	1	7	ns	TSys
Run Terminate	25	tStI	5	25	5	17	ns	TSys
Run Initiate	25	tRun	5	15	5	12	ns	TSys
Memory Write	25	tSMwr	5	35	5	27	ns	TSys
Exception Valid	25	tSExc	5	28	5	20	ns	TSys

Table 6. Capacitive Load Derating

Parameter	Symbol	12.5 MHz		16.667 MHz		Units
		Min	Max	Min	Max	
Load Derate	CLD	1	2.5	1	2	ns/25 pF

AC Specifications
(Continued)

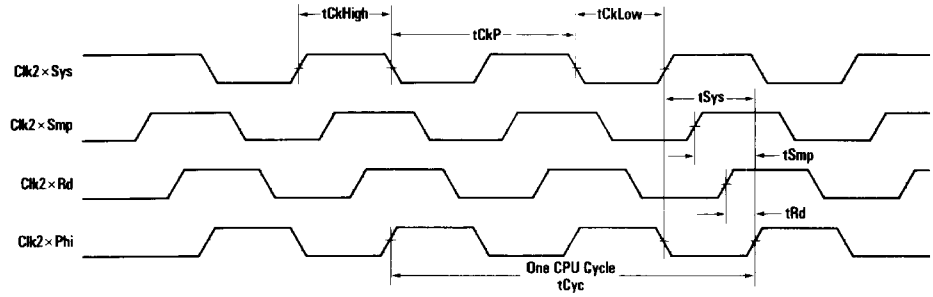


Figure 4. Clock Timing

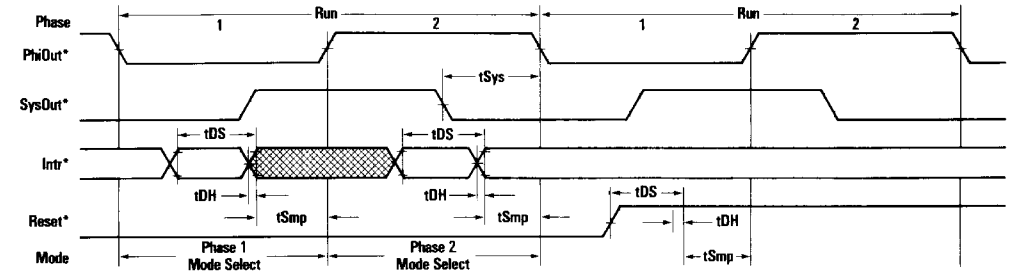


Figure 5. Reset and Mode Select Timing

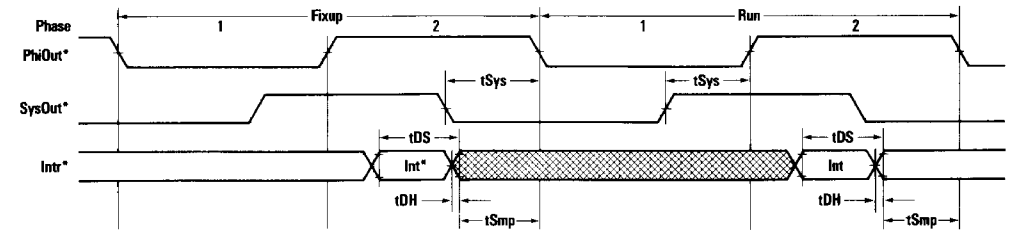


Figure 6. Interrupt Input Timing

AC Specifications
(Continued)

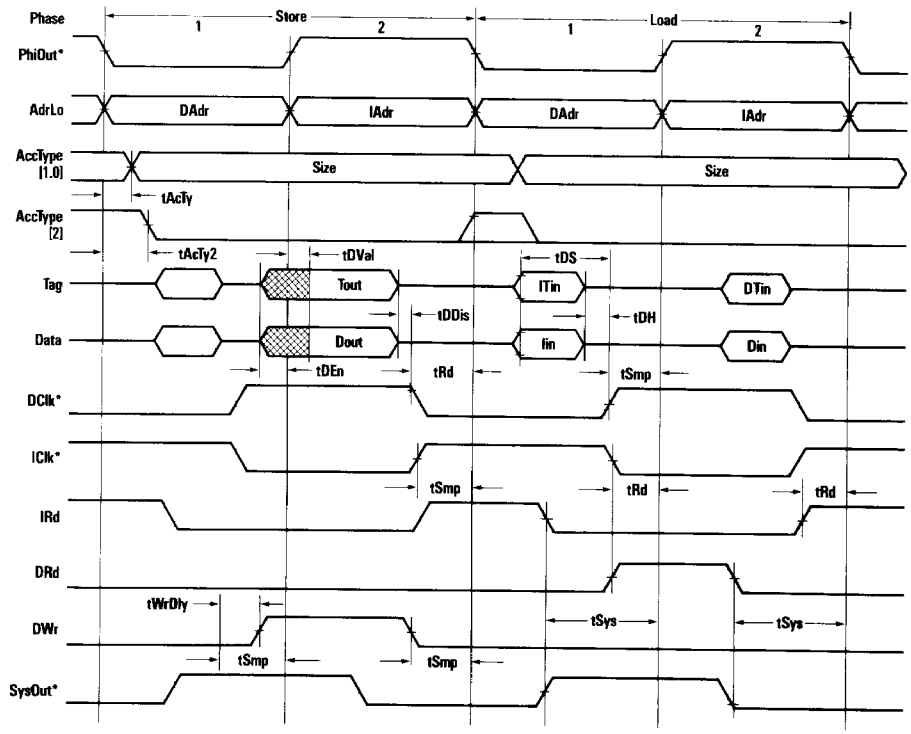


Figure 7. Detailed Cache Operation Timing

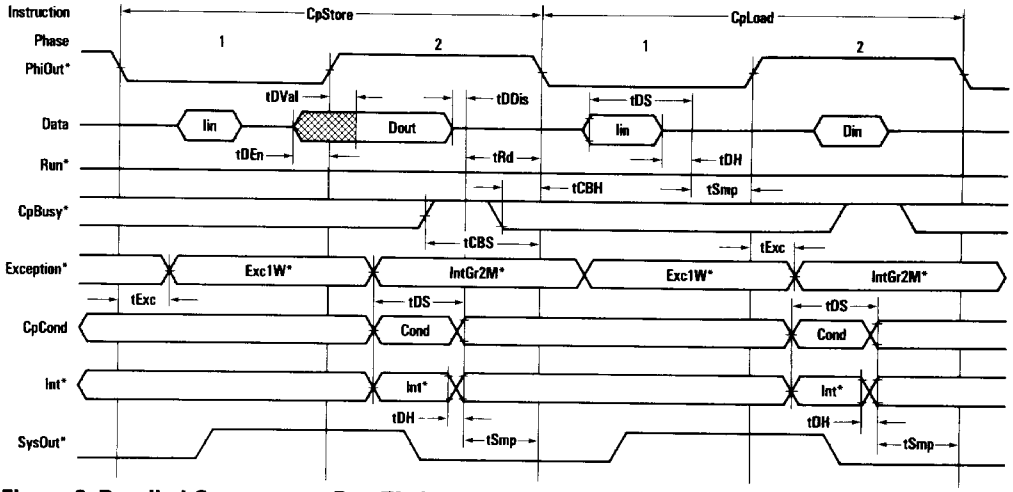


Figure 8. Detailed Coprocessor Run Timing

Package Specifications and Pin Locations

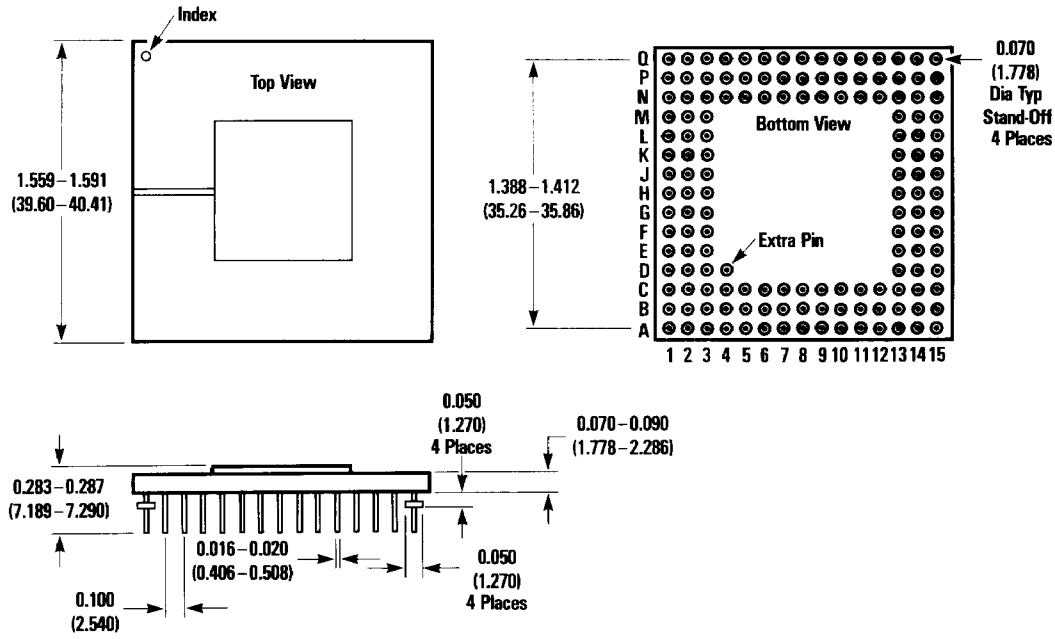


Figure 9. 144 Ceramic Pin Grid Array

Pin Assignments

Pin Name	Pin Number	Pin Name	Pin Number	Pin Name	Pin Number
Data(0)	E2	Tag(12)	B14	AdrLo(0)	C1
Data(1)	D1	Tag(13)	C13	AdrLo(1)	E3
Data(2)	F3	Tag(14)	D13	AdrLo(2)	D2
Data(3)	G2	Tag(15)	B15	AdrLo(3)	B1
Data(4)	G1	Tag(16)	E133	AdrLo(4)	C2
Data(5)	H2	Tag(17)	D14	AdrLo(5)	C4
Data(6)	H1	Tag(18)	C15	AdrLo(6)	A2
Data(7)	F2	Tag(19)	D15	AdrLo(7)	B3
Data(8)	H3	Tag(20)	E14	AdrLo(8)	C5
Data(9)	J3	Tag(21)	F14	AdrLo(9)	B4
Data(10)	J1	Tag(22)	G14	AdrLo(10)	A3
Data(11)	K2	Tag(23)	F15	AdrLo(11)	A4
Data(12)	L2	Tag(24)	H15	ArdLo(12)	B5
Data(13)	M1	Tag(25)	H14	AdrLo(13)	B7
Data(14)	N1	Tag(26)	J15	AdrLo(14)	A6
Data(15)	K1	Tag(27)	K15	AdrLo(15)	A7
Data(16)	M2	Tag(28)	J13	VCC0	F1
Data(17)	L3	Tag(29)	J14	VCC1	L1
Data(18)	N2	Tag(30)	L15	VCC2	Q1
Data(19)	N3	Tag(31)	L14	VCC3	N7
Data(20)	P2	TagP(0)	C14	VCC4	N8
Data(21)	Q2	TagP(1)	G15	VCC5	Q12
Data(22)	P4	TagP(2)	K14	VCC6	Q15
Data(23)	P1	TagV	N15	VCC7	M15
Data(24)	N5	Intr*(0)	C9	VCC8	H13
Data(25)	Q3	Intr*(1)	B9	VCC9	E15
Data(26)	P5	Intr*(2)	A11	VCC10	A15
Data(27)	P6	Intr*(3)	B10	VCC11	C8
Data(28)	Q5	Intr*(4)	C10	VCC12	A5
Data(29)	Q7	Intr*(5)	A12	VCC13	C3
Data(30)	P8	CpCond(0)	A8	VCC14	A1
Data(31)	Q4	CpCond(1)	B8	Gnd0	D3
DataP(0)	E1	CpCond(2)	A9	Gnd1	G3
DataP(1)	J2	CpCond(3)	A10	Gnd2	K3
DataP(2)	M3	AccTyp(0)	P15	Gnd3	N4
DataP(3)	N6	AccTyp(1)	M14	Gnd4	Q6
Clk2xSys	P9	AccTyp(2)	L13	Gnd5	N9
Clk2xSmp	Q10	MemWr*	N12	Gnd6	N10
Clk2xRd	P10	MemRd*	N13	Gnd7	M13
Clk2xPhi	Q9	Run*	N14	Gnd8	K13
RdBusy	C11	IRd	P12	Gnd9	G13
WrBusy*	A13	IWr	P13	Gnd10	F13
CpBusy*	B11	DRd	N11	Gnd11	C12
BusError*	B12	DWr	Q14	Gnd12	C7
Reset*	A14	IClk*	Q13	Gnd13	C6
SysOut*	Q11	DClk*	P11	Exc*	Q8
CpSync*	P14	reserved0	P3	reserved1	P7
reserved2	B2	reserved3	B6	reserved4	B13

To ensure compatibility with future versions of the R2000, make no connections to pins labeled reserved.

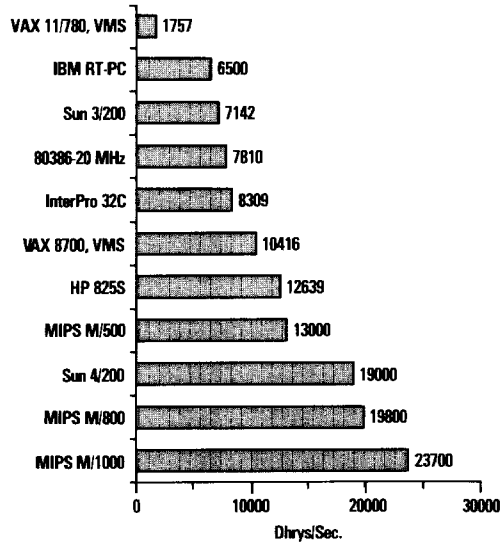
Benchmarks

The following benchmarks illustrate the performance advantage of the R2000 processor versus other CISC-and RISC-based machines in use today.

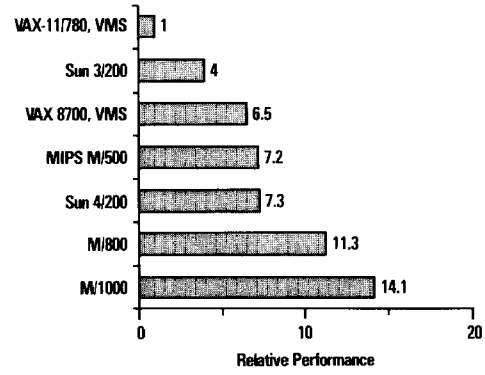
These benchmarks are based on industry standard benchmarking programs which are "compute-bound" to measure CPU performance (rather than I/O performance).

The R2000/12 is the processor used in the M/800 machine, and the R2000/16 is the processor used in the M/1000 machine. Both machines use the R2010 floating-point accelerator (FPA).

Dhrystone 1.1 Benchmark

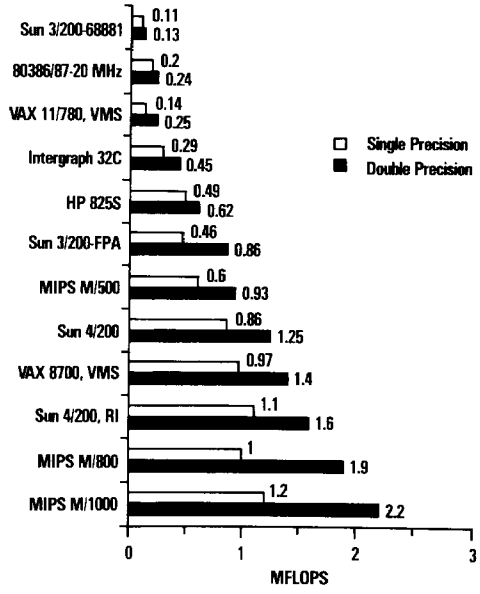


Stanford Integer Benchmark

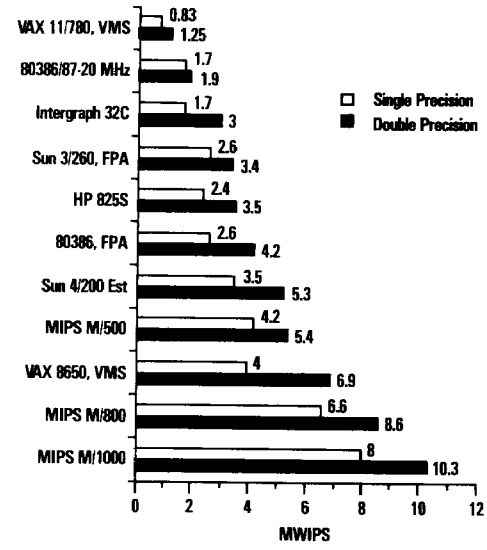


Benchmarks
(Continued)

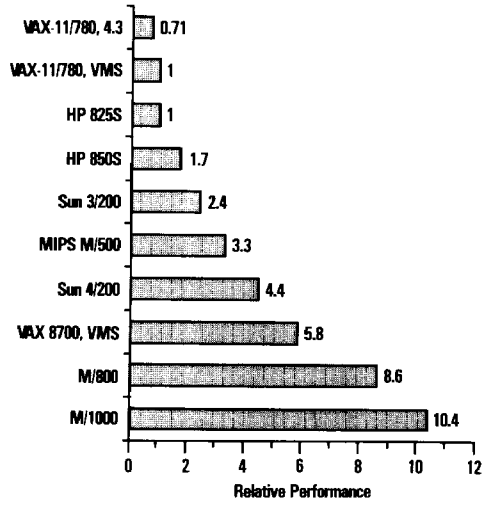
Fortran Blas Linpack Benchmark



Whetstone Benchmark



Spice MOSAMP2 Benchmark



Digital Review Benchmarks