NEC
NEC Electronics Inc.

VR4400PC (μPD30410) 64-Bit RISC Microprocessor: On-Chip Primary Cache Memory

Preliminary Information

November 1993

Description

The VR4400PC™ is a 64-bit RISC microprocessor with the enhancement version of R4000 architecture. It delivers excellent processing solutions in a wide variety of applications.

Two other versions of the RISC microprocessor are compared with the VR4400PC below.

VR4400PC™	On-chip primary cache
VR4400SC™	On-chip primary cache and secondary cache support
VR4400MC™	Same as VR4400SC plus multiprocessing capability

Typical applications are in cost-sensitive systems without secondary cache such as inexpensive desktop systems and high-end controllers.

The VR4400PC microprocessor provides complete application software compatibility with the MIPS R2000, R3000, and R6000 microprocessors. The RISC/os and RISC compilers and thousands of application programs that run on the MIPS architecture augment this powerful family of processors and provide a complete solution to a large number of processing needs. In addition, an array of development tools supports R4000-based applications

High integer performance, as well as floating-point performance, has been achieved through techniques such as superpipelining, on-chip caches, a pipelined floating-point unit, two-level cache memory, and a high-performance on-chip translation lookaside buffer (TLB).

The cache and memory management unit (MMU) can handle large address space tasks and a large number of users. These features allow the design of balanced systems, suitable not only for technical and graphics applications but also for commercial applications.

The 64-bit wide on-chip cache path, 64-bit on-chip FPU, 64-bit integer registers, and 64-bit virtual address space provide a compatible, timely, and necessary path from 32-bit to true 64-bit computing for users and software developers. Compatibility with existing 32-bit application code is maintained; however, an efficient mix of 32-bit and 64-bit programs can run on the same VR4400PC machine.

50629-1

The 64-bit addressing capability of the VR4400PC microprocessor supports operating systems with extensive file mapping—allowing direct access to files without explicit I/O calls—and paves the way for next-generation video technology and documentation with high-quality photographs. In addition, CAD applications with huge databases of complex structures, geographic information systems, and technical number crunching applications with large data sets will benefit greatly from this addressing capability.

Another enhancement of the VR4400PC is a cache error bit (EW) added to the cache error register.

Features

- True 64-bit microprocessor with 64-bit integer and floating-point operations, registers, and virtual addresses
- Fully compatible with earlier 32-bit MIPS microprocessors
- Dual instruction issue with no restrictions on the type of instruction issued.
- 50-, 67-, or 75-MHz Master Clock (100-, 133-, or 150-MHz PClock)
- 3.3- or 5-volt power supply
- On-chip 16-Kbyte instruction cache, 16-Kbyte data cache
- On-chip memory management unit containing a fully associative TLB whose entries have a variable page size ranging from 4 Kbytes to 16 Mbytes
- On-chip ANSI/IEEE-754 standard floating-point unit with precise exceptions
- 32 doubleword (64-bit) general-purpose registers and 16 doubleword (64-bit) floating-point registers
- 36-bit physical address accessing 64 Gbytes of physical memory
- Six hardware interrupts
- 64-bit cache coherent system interface
- Dynamically configurable big-endian or little-endian byte ordering
- Timing flexibility for a 64-bit system interface to allow speed matching of logic and memory components
- System interface clock modes: divide-by-2, -3, -4, -6, and -8

VR4400PC (μPD30410)

NEC

Features (cont)

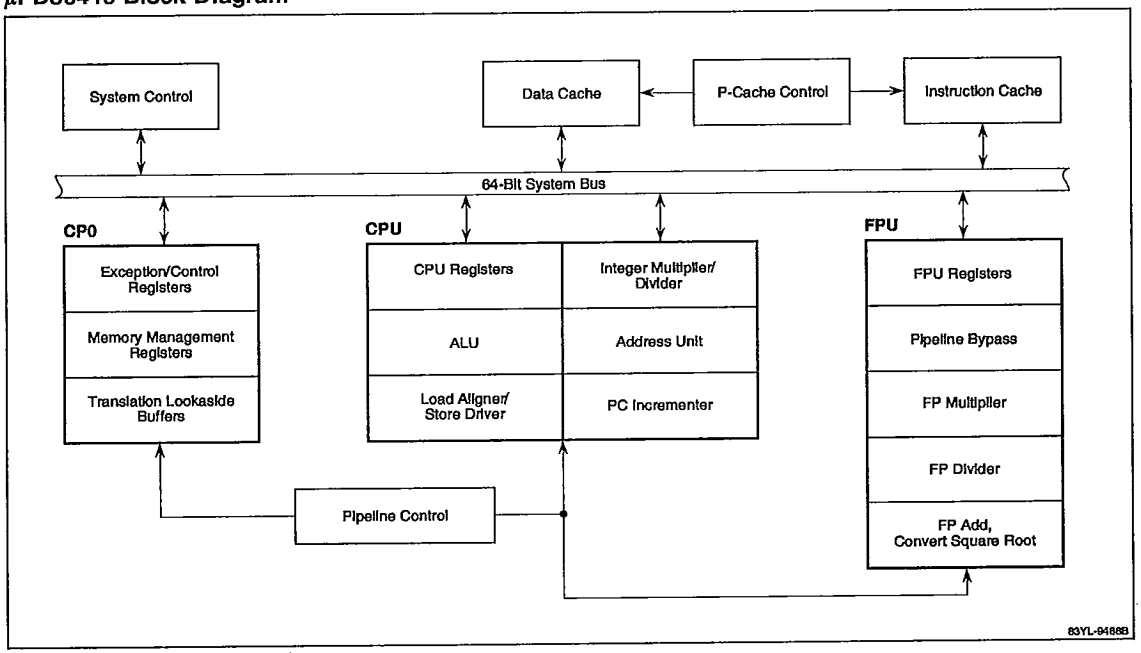
- Master/checker mode
- One-level deep uncached store buffer

Ordering Information

Part Number	PClock	Power Supply	Package
μPD30410RJ-50	100 MHz	5 V	179-pin ceramic PGA
-67	133 MHz		
-75	150 MHz		
μPD30410RP-50	100 MHz	5 V	179-pin ceramic PGA with heat-sink adapter plate
-67	133 MHz		
-75	150 MHz		

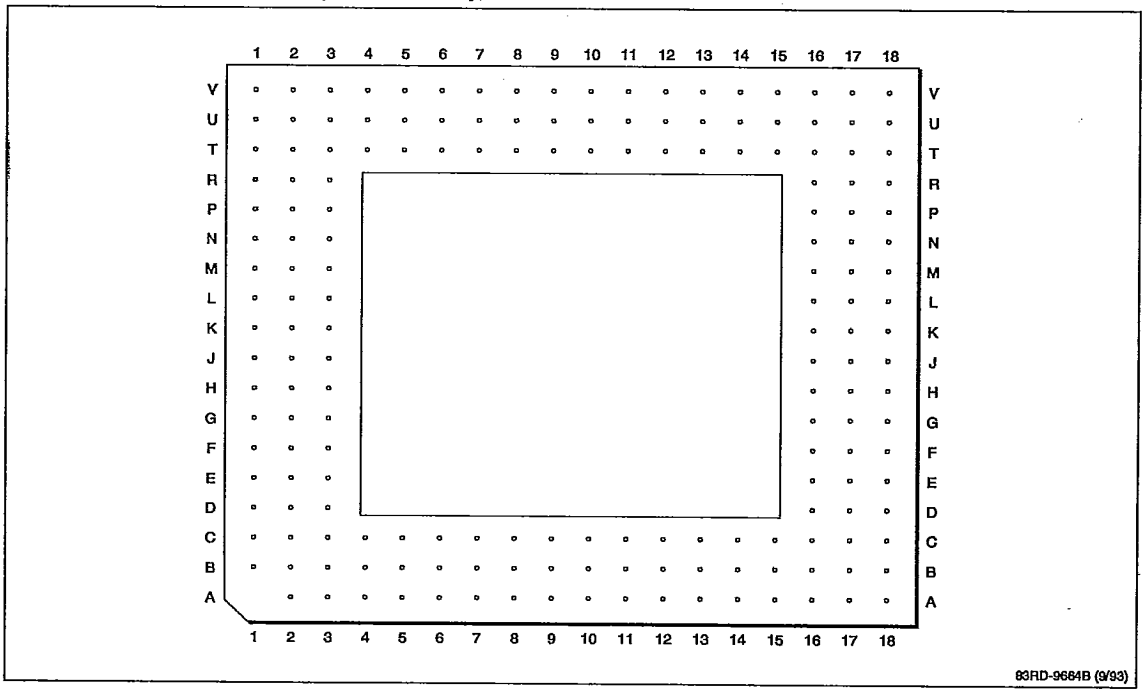
Part Number	PClock	Power Supply	Package
μPD30410LRJ-50	100 MHz	3.3 V	179-pin ceramic PGA
-67	133 MHz		
-75	150 MHz		
μPD30410LRP-50	100 MHz	3.3 V	179-pin ceramic PGA with heat-sink adapter plate
-67	133 MHz		
-75	150 MHz		

μPD30410 Block Diagram



NEC

Package Pin Configuration (Bottom View); 179-Pin PGA



Pin Assignments; System Address/Data, System Command, Clock and Control

System Address/Data	Pin No.	System Address/Data	Pin No.	System Command	Pin No.	Clock and Control	Pin No.
SysAD0	J2	SysAD40	C10	SysCmd0	E2	$\overline{\text{ColdReset}}$	R14
1	G2	41	C11	1	D3	$\overline{\text{ExtRqst}}$	S2
2	E1	42	B13	2	B2	GndP	K16
3	E3	43	A15	3	A5	$\overline{\text{Int0}}$	N2
4	C2	44	C15	4	B7	$\overline{\text{Int1}}$	L3
5	C4	45	B17	5	C9	$\overline{\text{Int2}}$	K3
6	B5	46	E17	6	B10	$\overline{\text{Int3}}$	J3
7	B6	47	F17	7	B12	$\overline{\text{Int4}}$	H3
8	B9	48	L2	8	C13	$\overline{\text{Int5}}$	F2
9	B11	49	M3			IOIn	R13
SysAD10	C12	SysAD50	N3	SysCmdP	C14	IOOut	S12
11	B14	51	Q2			MasterClock	J17
12	B15	52	R3			MasterOut	P17
13	C16	53	S3			ModeClock	B4
14	D17	54	R6			ModeIn	S4
15	E18	55	R7			$\overline{\text{NMI}}$	S7
16	K2	56	R10			RClock0	R17
17	M2	57	R11			RClock1	Q16
18	P1	58	S13			$\overline{\text{RdRdy}}$	R5
19	P3	59	T15			$\overline{\text{Release}}$	T5
						$\overline{\text{Reset}}$	AU39
SysAD20	R2	SysAD60	R15			SyncIn	J16
21	R4	61	S17			SyncOut	P16
22	S5	62	N16			TClock0	C17
23	S6	63	N17			TClock1	D16
24	S9					$\overline{\text{ValidIn}}$	P2
25	S11					$\overline{\text{ValidOut}}$	Q3
26	R12					V_{DD}Ok	M17
27	S14					V_{DD}P	K17
28	S15					$\overline{\text{WrRdy}}$	C5
29	R16						
SysAD30	Q17	SysADC0	C8			**JTAG**	
31	M16	1	G17			JTCK	H17
32	H2	2	R8			JTDI	G16
33	G3	3	L16			JTDO	F16
34	F3	4	B8			JTMS	E16
35	D2	5	H16				
36	C3	6	S8				
37	B3	7	L17				
38	C6						
39	C7						

Pin Assignments; V_{DD}, GND, and NC (No Connection)

V_{DD}	GND	NC
A2, A4, A7, A9, A11, A13, A16	A3, A6, A8, A10, A12, A14, A17, A18	B16
B18, C1	B1, C18	S10
D18, F1	D1, F18	
G18, H1	G1, H18	
J18, K1	J1, K18	
L18, M1	L1, M18	
N18, Q1	N1, P18, Q18	
R9, R18	R1	
S1	S18	
T3, T6, T8, T10, T12, T14, T17	T1, T2, T4, T7, T9, T11, T13, T16, T18	

Pin Descriptions

Interface Signals	Symbol	Input/Output	Description
System	$\overline{\text{ExtRqst}}$	Input	External Request is asserted by the external agent to request use of the system interface. The processor grants the request by asserting $\overline{\text{Release}}$.
	$\overline{\text{Release}}$	Output	Release responds to the assertion of $\overline{\text{ExtRqst}}$. The processor asserts $\overline{\text{Release}}$, signalling to the requesting device that the system interface is available.
	$\overline{\text{RdRdy}}$	Input	Read Ready is asserted by the external agent to indicate that it can accept a processor read requests or a read followed by a write request.
	SysAD(63:0)	Input/Output	System Address and Data Bus is a 64-bit bus for communication between the processor and the external agent.
	SysADC(7:0)	Input/Output	System Address and Data Check Bus is an 8-bit bus that contains check bits for the SysAD bus.
	SysCmd(8:0)	Input/Output	System Command and Data Identifier is a 9-bit bus for transmission between the processor and the external agent.
	SysCmdP	Input/Output	System Command and Data Identifier Bus Parity is a single, even-parity bit for the SysCmd bus.
	$\overline{\text{ValidIn}}$	Input	Valid Input is asserted by the external agent when it is driving a valid address or data on the SysAD bus and a valid command or data identifier on the SysCmd bus.
	$\overline{\text{ValidOut}}$	Output	Valid Out is a signal the processor asserts to indicate that it is driving a valid address or data on the SysAD bus and a valid command or data identifier on the SysCmd bus.
	$\overline{\text{WrRdy}}$	Input	Write Ready is asserted by the external agent when it can accept a processor write request.

Pin Descriptions (cont)

Interface Signals	Symbol	Input/Output	Description
Clock/Control	IOIn	Input	I/O Input is the output slew-rate control feedback loop input. (See IOOut.)
	IOOut	Output	I/O Output is the output slew-rate control feedback loop output. It must be connected to IOIn through a delay loop that models the I/O path from the processor to the external agent.
	MasterClock	Input	Master Clock is the primary clock input to establish the processor operating frequency.
	MasterOut	Output	Master Clock Output is aligned with Master Clock.
	RClock(1:0)	Output	Receive Clocks 1 and 0 are identical clocks that establish the system interface frequency.
	SyncOut	Output	Synchronization Clock Out is a signal to model the interconnect between MasterOut, TClock, RClock, and the external agent. It must be connected to SyncIn through an interconnect.
	SyncIn	Input	Synchronization Clock Input is the input of the synchronization clock.
	TClock(1:0)	Output	Transmit Clocks 1 and 0 are identical clocks that establish the system interface frequency.
	$V_{DD}P$	Input	$V_{DD}P$ is quiet V_{DD} to the internal phase-locked loop (PLL).
	GndP	Input	Quiet Ground is directed to the internal phase-locked loop.
Interrupt	$\overline{\text{Int}}(0:5)$	Input	$\overline{\text{Int}}$ is one of the six general processor interrupts; respectively, $\overline{\text{Int}}$ 0 thru 5 are ORed with bits 0 thru 5 of the interrupt register.
	$\overline{\text{NMI}}$	Input	Nonmaskable Interrupt is a hardware interrupt that can't be disabled by internal masking. It is ORed with bit 6 of the interrupt register.
Initialization	$\overline{\text{ColdReset}}$	Input	$\overline{\text{ColdReset}}$ must be asserted for a power-on reset or a cold reset. The SClock, TClock, and RClock begin to cycle and are synchronized with the deasserted edge of $\overline{\text{ColdReset}}$. It must be deasserted synchronously with MasterOut.
	ModeClock	Output	Mode Clock is a serial boot-time mode data clock output; it runs at the system clock frequency divided by 256 (MasterClock/256).
	ModeIn	Input	ModeIn is a serial boot-time mode data input.
	$\overline{\text{Reset}}$	Input	$\overline{\text{Reset}}$ must be asserted for any reset sequence. It can be asserted synchronously or asynchronously for a cold reset, or synchronously to initiate a warm reset. $\overline{\text{Reset}}$ must be deasserted synchronously with MasterOut.
	V_{DD}Ok	Input	Asserting V_{DD}Ok indicates to the processor that the power supply has been above 4.75 V (5-volt parts) or above 3 V (3.3-volt parts) for more than 100 ms and will remain stable and start the initialization sequence.
JTAG	JTDI	Input	Data is serially scanned in through the JTDI pin (JTAG Data In)
	JTCK	Output	The processor outputs a serial clock on the JTCK pin (Tag Clock Input). Both JTDI and JTMS are sampled on the rising edge of JTCK.
	JTDO	Output	Data is serially scanned out through the JTDO pin (JTAG Data Out).
	JTMS	Input	JTMS (JTAG Command) indicates the incoming signal data is command data.

ARCHITECTURE

CPU Registers

The VR4400PC microprocessor provides 32 general-purpose registers, a program counter (PC), and two registers that hold the results of integer multiply and divide operations. See figure 1. These registers are 32 or 64 bits wide depending on the mode of operation. General-purpose registers r0 and r1 have special functions.

(1) r0 is hardwired to a value of zero. It can be used as the target register for any instruction whose results can be discarded; it can also be used as a source when a zero value is needed.

(2) r31 is the link register for JumpAndLink instructions. It should not be used explicitly by other instructions.

The MIPS architecture defines three special registers whose use or modification is implicit with certain instructions. These special registers are:

PC Program Counter
HI Multiply and Divide register, higher result
LO Multiply and Divide register, lower result

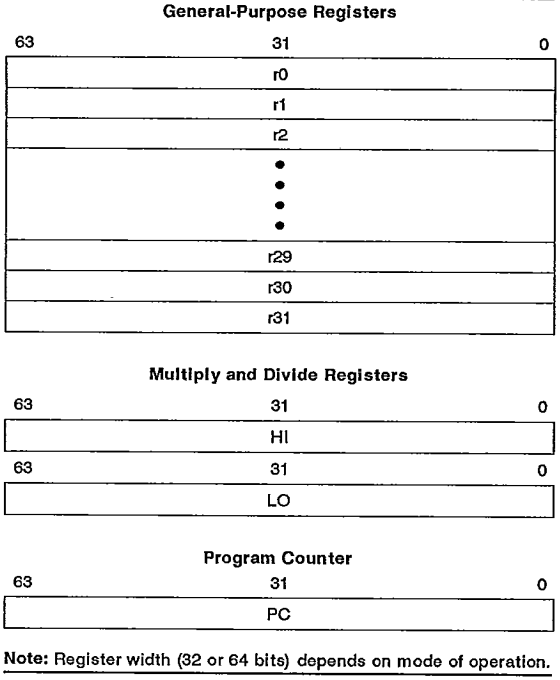
The two Multiply and Divide registers (HI, LO) store the doubleword 64-bit result or the quadword 128-bit result of integer multiply operations and the quotient (in LO) and remainder (in HI) of integer divide operations.

The VR4400PC has no Program Status Word (PSW) register; its functions are provided by the Status and Cause registers incorporated within Coprocessor 0. CP0 registers are described later.

CPU Instruction Set

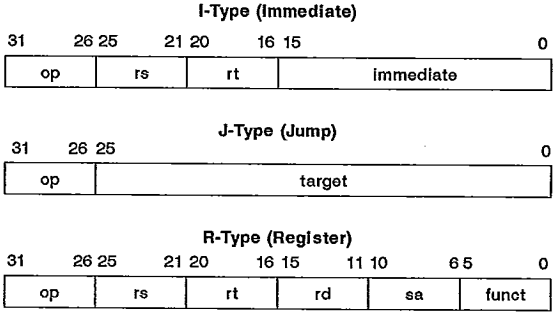
Each CPU instruction is 32 bits long. Figure 2 shows the three instruction formats: I-type (immediate), J-type (jump), and R-type (register). Using only these three instruction formats simplifies instruction decoding; more complicated (and less frequently used) operations and addressing modes can be synthesized by the compiler using sequences of these simple instructions.

Figure 1. CPU Registers



Note: Register width (32 or 64 bits) depends on mode of operation.

Figure 2. CPU Instruction Formats



The instruction set can be divided into the following groups:

- **Load and Store** instructions move data between memory and general registers. They are all I-type instructions, since the only addressing mode supported is base register plus 16-bit, signed immediate offset.
- **Computational** instructions perform arithmetic, logical, shift, multiply, and divide operations on values in registers. They occur in both R-type format (operands and result are stored in registers) and I-type format (one operand is a 16-bit immediate value).
- **Jump and Branch** instructions change the control flow of a program. Jumps are always to a paged, absolute address formed by combining a 26-bit target address with the high-order bits of the Program Counter (J-type format) or register addresses (R-type format). Branches have 16-bit offsets relative to the program counter (I-type). JumpAndLink instructions save a return address in register 31.
- **Coprocessor** instructions perform operations in the coprocessors. Coprocessor load and store instructions are I-type.
- **Coprocessor 0** instructions perform operations on CP0 registers to manipulate the memory management and exception handling facilities of the processor. Table 1 lists these instructions.
- **Special** instructions perform system calls and breakpoint operations. These instructions are always R-type.
- **Exception** instructions cause a branch to the general exception-handling vector based upon the result of a comparison. These instructions occur in both R-type format (operands and result are stored in registers) and I-type format (one operand is a 16-bit immediate value).

Table 2 is the instruction set (ISA) common to all V_R-Series processors; table 3 lists VR4400PC microprocessor instructions that are extensions to the ISA. These instructions result in code space reductions, multiprocessor support, and improved performance in operating system kernel code sequences and in situations where run-time bounds checking is frequently performed.

Table 1. CP0 Instructions

OP	Description
DMFC0	Doubleword Move from CP0
DMTC0	Doubleword Move to CP0
MTC0	Move to CP0
MFC0	Move from CP0
TBLR	Read Indexed TLB Entry
TLBWI	Write Indexed TLB Entry
TLBWR	Write Random TLB Entry
TLBP	Probe TLB for Matching Entry
ERET	Exception Return

Table 2. CPU Instruction Set (ISA)

OP	Description
Load and 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, R-type)	
ADD	Add
ADDU	Add Unsigned
SUB	Subtract
SUBU	Subtract Unsigned
SLT	Set on Less Than
SLTU	Set on Less Than Unsigned

Table 2. CPU Instruction Set (ISA) (cont)

OP	Description
Arithmetic Instructions (3-operand, R-type) (cont)	
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
Multiply and Divide Instructions	
MULT	Multiply
MULTU	Multiply 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 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
Coprocessor Instructions	
LWCz	Load Word to Coprocessor z
SWCz	Store Word from Coprocessor z
MTCz	Move to Coprocessor z
MFCz	Move from Coprocessor z
CTCz	Move Control to Coprocessor z

Table 2. CPU Instruction Set (ISA) (cont)

OP	Description
CFCz	Move Control from Coprocessor z
COPz	Coprocessor Operation z
BCzT	Branch on Coprocessor z True
BCzF	Branch on Coprocessor z False
Special Instructions	
SYSCALL	System Call
BREAK	Break

Table 3. Extensions to the ISA

OP	Description
Load and Store Instructions	
LD	Load Doubleword
LDL	Load Doubleword Left
LDR	Load Doubleword Right
LL	Load Linked
LLD	Load Linked Doubleword
LWU	Load Word Unsigned
SC	Store Conditional
SCD	Store Conditional Doubleword
SD	Store Doubleword
SDL	Store Doubleword Left
SDR	Store Doubleword Right
SYNC	Sync
Arithmetic Instructions (ALU Immediate)	
DADDI	Doubleword Add Immediate
DADDIU	Doubleword Add Immediate Unsigned
Arithmetic Instructions (3-operand, R-type)	
DADD	Doubleword Add
DADDU	Doubleword Add Unsigned
DSUB	Doubleword Subtract
DSUBU	Doubleword Subtract Unsigned
Shift Instructions	
DSLL	Doubleword Shift Left Logical
DSRL	Doubleword Shift Right Logical
DSRA	Doubleword Shift Right Arithmetic
DSLLV	Doubleword Shift Left Logical Variable
DSRLV	Doubleword Shift Right Logical Variable
DSRAV	Doubleword Shift Right Arithmetic Variable
DSLL32	Doubleword Shift Left Logical+32
DSRL32	Doubleword Shift Right Logical+32
DSRA32	Doubleword Shift Right Arithmetic+32

Table 3. Extensions to the ISA (cont)

OP	Description
Multiply and Divide Instructions	
DMULT	Doubleword Multiply
DMULTU	Doubleword Multiply Unsigned
DDIV	Doubleword Divide
DDIVU	Doubleword Divide Unsigned
Jump and Branch Instructions	
BEQL	Branch on Equal Likely
BNEL	Branch on Not Equal Likely
BLEZL	Branch on Less Than or Equal to Zero Likely
BGTZL	Branch on Greater Than Zero Likely
BLTZL	Branch on Less Than Zero Likely
BGEZL	Branch on Greater Than or Equal to Zero Likely
BLTZALL	Branch on Less Than Zero and Link Likely
BGEZALL	Branch on Greater Than or Equal to Zero and Link Likely
BCzTL	Branch on Coprocessor z True Likely
BCzFL	Branch on Coprocessor z False Likely
Exception Instructions	
TGE	Trap If Greater Than or Equal
TGEU	Trap If Greater Than or Equal Unsigned
TLT	Trap If Less Than
TLTU	Trap If Less Than Unsigned
TEQ	Trap If Equal
TNE	Trap If Not Equal
TGEI	Trap If Greater Than or Equal Immediate
TGEIU	Trap If Greater Than or Equal Immediate Unsigned
TLTI	Trap If Less Than Immediate
TLTIU	Trap If Less Than Immediate Unsigned
TEQI	Trap If Equal Immediate
TNEI	Trap If Not Equal Immediate
Coprocessor Instructions	
DMFCz	Doubleword Move from Coprocessor z
DMTCz	Doubleword Move to Coprocessor z
LDCz	Load Doubleword to Coprocessor z
SDCz	Store Doubleword from Coprocessor z

Data Formats and Addressing

The VR4400PC microprocessor uses four data formats: 64-bit doubleword, 32-bit word, 16-bit halfword, and 8-bit byte. The byte ordering is configurable as either big-endian or little-endian format.

Note: Endianness refers to the location of byte 0 within a multibyte structure.

Figures 3 and 4 show the ordering of bytes within words and the ordering of words within multiple-word structures for the big-endian and little-endian conventions

When the VR4400PC is configured as a big-endian system, byte 0 is the most-significant (leftmost) byte, thereby providing compatibility with MC68000™ and IBM 370® conventions. This configuration is shown in figure 3.

Figure 3. Addresses of Bytes Within Words; Big-Endian Byte Alignment

31 24	23 16	15 8	7 0	Word Address
8	9	10	11	8
4	5	6	7	4
0	1	2	3	0

- Most-significant byte is at lowest address.
- Word is addressed by byte address of most-significant byte.

In a little-endian system, byte 0 is always the least-significant (rightmost) byte, which is compatible with iAPX™ x86 and DEC VAX™ conventions. This configuration is shown in figure 4.

MC68000 is a trademark of Motorola, Inc; IBM 370 is a registered trademark of International Business Machines Corp; iAPX is a trademark of Intel Corp; DEC VAX is a trademark of Digital Equipment Corp.

Figure 4. Addresses of Bytes Within Words; Little-Endian Byte Alignment

31 24	23 16	15 8	7 0	Word Address
11	10	9	8	8
7	6	5	4	4
3	2	1	0	0

- Least-significant byte is at lowest address.
- Word is addressed by byte address of least-significant byte.

In this data sheet, bit 0 is always the least-significant (rightmost) bit; thus, bit designations are always little-endian (although no instructions explicitly designate bit positions within words).

Figures 5 and 6 show byte alignment in doublewords.

Figure 5. Addresses of Bytes Within Doublewords; Big-Endian Byte Alignment

64							0	Doubleword Address
16	17	18	19	20	21	22	23	16
8	9	10	11	12	13	14	15	8
0	1	2	3	4	5	6	7	0

- Most-significant byte is at lowest address.
- Word is addressed by byte address of most-significant byte.

Figure 6. Addresses of Bytes Within Doublewords; Little-Endian Byte Alignment

64							0	Doubleword Address
23	22	21	20	19	18	17	16	16
15	14	13	12	11	10	9	8	8
7	6	5	4	3	2	1	0	0

- Least-significant byte is at lowest address.
- Word is addressed by byte address of least-significant byte.

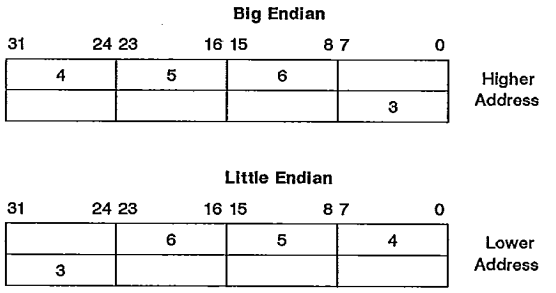
The CPU uses byte addressing for halfword, word, and doubleword accesses with the following alignment constraints.

- Halfword accesses must be aligned on an even byte boundary (0, 2, 4 . . .).
- Word accesses must be aligned on a byte boundary divisible by 4 (0, 4, 8 . . .).
- Doubleword accesses must be aligned on a byte boundary divisible by 8 (0, 8, 16 . . .).

As shown in figures 5 and 6, the address of a multiple-byte data item is the address of the most-significant byte on a big-endian configuration, or the address of the least-significant byte on a little-endian configuration.

Special instructions are provided for loading and storing words that are not aligned on 4-byte (word) or 8-byte (doubleword) boundaries: LWL, LWR, SWL, SWR, LDL, LDR, SDL, SDR. These instructions are used in pairs to provide addressing of misaligned words with one additional instruction cycle over that required for aligned words. For each of the two endianness conventions, figure 7 shows the bytes that are accessed when addressing a misaligned word with byte address 3.

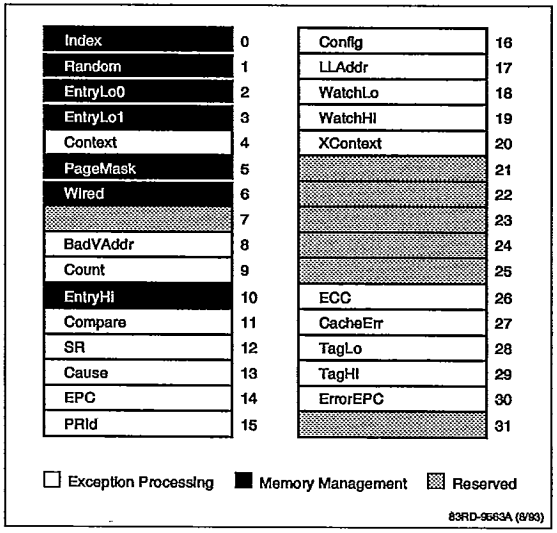
Figure 7. Example of Misaligned Words: Byte Address 3



System Control Coprocessors

The MIPS ISA allows up to four coprocessors, CP0 through CP3. Coprocessor CP1 is reserved for the on-chip, floating-point coprocessor. Coprocessor CP2 is reserved for future definition by MIPS, and the encoding for coprocessor CP3 is used to provide certain extensions to the MIPS ISA. Coprocessor CP0 is also incorporated on the CPU chip and supports the virtual memory system and exception handling. The virtual memory system is implemented with an on-chip TLB and a group of programmable registers as described in figure 8.

Figure 8. System Control Coprocessor CP0 Registers



Coprocessor CP0 translates virtual addresses into physical addresses and manages exceptions and transitions between kernel, supervisor, and user states. It also controls the cache subsystem and provides diagnostic control and error recovery facilities. The VR4400PC microprocessor also provides a generic system timer for interval timing, timekeeping, process accounting, and time-slicing.

The CP0 registers shown in figure 8 and described in table 4 manipulate the memory management and exception handling capabilities of the CPU.

Table 4. System Control Coprocessor CP0 Registers

No.	Register	Description
0	Index	Programmable pointer into TLB array
1	Random	Pseudorandom pointer into TLB array (read only)
2	EntryLo0	Low half of TLB entry for even VPN
3	EntryLo1	Low half of TLB entry for odd VPN
4	Context	Pointer to kernel virtual PTE table in 32-bit addressing mode
5	PageMask	TLB page mask
6	Wired	Number of wired TLB entries
7	—	Reserved
8	BadVAddr	Bad virtual address
9	Count	Timer count
10	EntryHi	High half of TLB entry
11	Compare	Timer compare
12	SR	Status register
13	Cause	Cause of last exception
14	EPC	Exception program counter
15	PRId	Processor revision identifier
16	Config	Configuration register
17	LLAddr	Load linked address
18	WatchLo	Memory reference trap address, low bits
19	WatchHI	Memory reference trap address, high bits
20	XContext	Pointer to kernel virtual PTE table in 64-bit addressing mode
21-25	—	Reserved
26	ECC	Secondary-cache ECC and primary parity
27	CacheErr	Cache error and status register
28	TagLo	Cache tag register
29	TagHi	Cache tag register
30	ErrorEPC	Error exception program counter
31	—	Reserved

Floating-Point Unit (FPU)

The Floating-Point Unit (FPU) operates as a coprocessor for the CPU and extends the CPU instruction set to perform arithmetic operations on values in floating-point representations. The FPU, with associated system software, fully conforms to the requirements of ANSI/IEEE Standard 754-1985, "IEEE Standard for Binary Floating-Point Arithmetic."

Full 64-Bit Operation. The FPU contains 16 64-bit registers or, optionally 32 64-bit registers that hold single-precision or double-precision values. The 16 additional floating-point registers are enabled by setting the FR bit in the Status register. The FPU also includes a 32-bit Status/Control register that provides access to all IEEE-Standard exception handling capabilities.

Load and Store Instruction Set. Like the CPU, the FPU uses a load- and store-oriented instruction set. Floating-point operations are started in a single cycle and their execution is overlapped with other fixed-point or floating-point operations.

Tightly Coupled Coprocessor Interface. The on-chip FPU appears to the programmer as an extension of the CPU (the FPU is accessed as Coprocessor CP1). This forms a tightly coupled unit with a seamless integration of floating-point and fixed-point instruction sets. Since each unit receives and executes instructions in parallel, some floating-point instructions can execute at the same rate (two instructions per cycle) as fixed-point instructions.

Cache Memory Hierarchy

To achieve its high performance in uniprocessor systems, the VR4400PC microprocessor supports a cache memory hierarchy that increases memory access bandwidth and reduces the latency of load and store instructions.

The VR4400PC incorporates on-chip 16-Kbyte instruction and 16-Kbyte data caches to keep the high-performance pipeline full. Each cache has its own 64-bit data path that can be accessed in parallel. The caches can be accessed twice in one cycle. Combining this feature with a pipeline, single-cycle access of each cache, the cache subsystem provides the integer and floating-point units with an aggregate bandwidth of 1.6 Gbytes per second at a MasterClock frequency of 50 MHz.

Memory Management System

The VR4400PC microprocessor has a physical addressing range of 64 Gbytes (36 bits). However, since most systems implement a physical memory smaller than 4 Gbytes, the CPU provides a logical expansion of memory space by translating addresses composed in a large virtual address space into available physical memory addresses. In 32-bit mode, the virtual address space is divided into 2 Gbytes per user process and 2 Gbytes for the kernel. In 64-bit mode, each 2-Gbyte space is expanded to 1 Tbyte.

Translation Lookaside Buffer (TLB). Virtual memory mapping is assisted by a TLB that caches virtual address translations. The fully-associative, on-chip TLB contains 48 entries, and each of these entries maps a pair of variable-sized pages (page size varies from 4 Kbytes to 16 Mbytes, increasing by multiples of 4).

An address translation value is tagged with the most-significant bits of its virtual address (the number of these bits depends on page size) and a per-process identifier. If there is no matching entry in the TLB, an exception is taken and software refills the on-chip TLB from a Page Table resident in memory. An entry chosen at random is replaced to make way for the new one. This TLB is referred to as the JTLB.

The VR4400PC also has a two-entry instruction TLB (ITLB) to assist in instruction address translation. The ITLB is completely invisible to software and is present for performance reasons only.

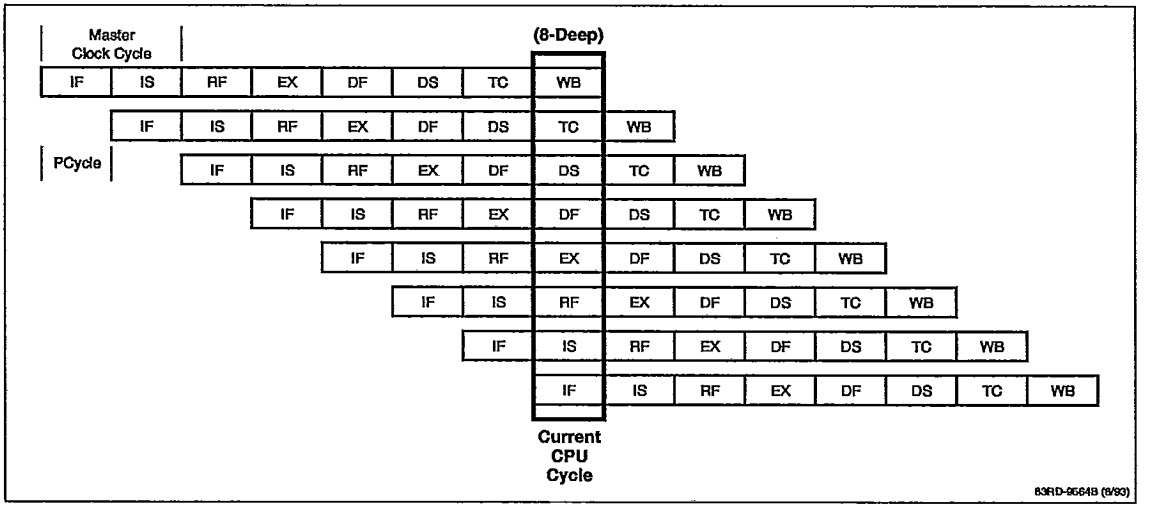
Operating Modes. The VR4400PC has three operating modes: User, Kernel, and Supervisor. The CPU normally operates in user mode until an exception is detected, forcing it into kernel mode. It remains in kernel mode until an Exception Return (ERET) instruction is executed. The supervisor mode can be used to design secure operating systems. The manner in which memory addresses are translated or mapped depends on the CPU operating mode.

Superpipeline Architecture

The VR4400PC microprocessor exploits instruction-level parallelism using a superpipelined implementation. The VR4400PC has an eight-stage superpipeline that places no restrictions on the instruction issued. Under nomal circumstances, any two instructions are issued each cycle.

The internal pipeline of the VR4400PC operates at twice the frequency of the master clock. This is shown in figure 9. The eight-stage superpipeline of the CPU achieves high throughput by pipelining cache accesses, shortening register access times, implementing virtual indexed primary caches, and allowing the latency of functional units to span multiple pipeline clock cycles (pcycles). In the rest of this document, the internal pipeline clock and clock cycles are often referred to as pclock and pcycles, respectively.

Figure 9. Eight-Stage Pipeline



Instruction Execution. The execution of a single VR4400PC instruction consists of the following eight primary steps:

IF Instruction fetch, first half. Virtual address is presented to the I-cache and TLB.

IS Instruction fetch, second half. The I-cache outputs the instruction and the TLB generates the physical address.

RF Register file. Three activities occur in parallel:
- Instruction is decoded and a check is made for interlock conditions.
- Instruction tag check is made to determine if there is a cache hit or not.
- Operands are fetched from the register file.

EX Instruction execute. One of three activities can occur:
- If the instruction is a register-to-register operation, an arithmetic, logical, shift, multiply, or divide operation is performed.
- If the instruction is a load and store, the data virtual address is calculated.
- If the instruction is a branch, the branch target virtual address is calculated and branch conditions are checked.

DF Data cache, first half. A virtual address is presented to the D-cache and TLB.

DS Data cache, second half. The D-cache outputs the instruction and the TLB generates the physical address.

TC A tag check is performed for loads and stores to determine if there is a hit or not.

WB Write back. The instruction result is written back to the register file.

The VR4400PC microprocessor uses an eight-stage pipeline; thus, execution of eight instructions at a time results in overlapping as shown in figure 9.

Uncached Store Buffer. The VR4400PC contains an uncached store buffer to improve the performance of uncached stores over that available from VR4000 processors. When an uncached store reaches the write-back (WB) stage in the CPU pipeline, the CPU must stall until the store is sent off-chip. In the VR4400PC, a single-entry buffer stores this uncached WB-stage data on the chip without stalling the pipeline.

If a second uncached store reaches the WB stage in the VR4400PC before the first one is moved off-chip, the CPU stalls until the store buffer clears the first uncached store. To avoid this stall, the compiler can insert seven instruction (NOP) cycles between the two uncached stores. If the two uncached stores execute within a loop, the two killed instructions, part of the loop branch latency, should be included in the count of seven interpolated cycles.

The timing requirements of the system interface govern the latency between uncached stores. Back-to-back stores can be sent across the interface at a maximum rate of one store for every four external cycles in SClock divide-by-two mode.

Exception Processing

The VR4400PC handles exceptions from a number of sources, including TLB missed, arithmetic overflows, I/O interrupts, and system calls. When the CPU detects an exception, the normal sequence of instruction execution is suspended; the processor exits the current mode and enters Kernel mode. The processor then disables interrupts and forces execution of a software handler located at a fixed address. The handler saves the context of the processor, including the contents of the program counter, the current operating mode, and the status of the interrupts. This context must be restored when the exception has been handled.

When an exception occurs, the CPU loads the Exception Program Counter (EPC) with a restart location where execution may resume after the exception has been serviced. The restart location in the EPC is the address of the instruction that caused the exception or, if the instruction was executing in a branch delay slot, the address of the branch instruction immediately preceding the delay slot.

A new bit, EW, is added in the VR4400PC CacheErr register to indicate if a cache error occurs while handling external requests. The error gets masked if the processor is executing in an exception handler. In such situations, cache coherency may not hold.

System Interface

The VR4400PC supports a 64-bit system interface that can be used to construct uniprocessor systems with a direct DRAM interface. The interface consists of a 64-bit multiplexed address and data bus with 8 check bits and a 9-bit parity-protected command bus. In addition, there are eight handshake signals. The interface has a simple timing specification and is capable of transferring data between the processor and memory at a peak rate of 400 Mbytes/second at 50 MHz.

Processor Interrupts

The VR4400PC supports six hardware interrupts, two software interrupts, and a nonmaskable interrupt. The processor's six hardware interrupts are accessible via external write requests. The nonmaskable interrupt is accessible via external write requests and a dedicated pin.

External writes to the processor are directed, based on a processor internal address map, to various processor internal resources. An external write to any address with SysAD(6:4) = 0 writes to an architecturally transparent register called the Interrupt Register. During the data cycle, SysAD(21:16) are the write enables for the 6 individual Interrupt Register bits and SysAD(5:0) are the values to be written into these bits. This allows any subset of the Interrupt Register to set and clear with a single write request. In the VR4400PC, bits 5:0 of the Interrupt Register are directly readable as bits 15:10 of the Cause Register. Bit 6 of the Interrupt Register is ORed with the current value of nonmaskable interrupt pin $\overline{\text{NMI}}$ to form the nonmaskable interrupt input to the processor.

Parity Error Detection

The parity method enables the processor to detect errors caused by system noise, power surges, and alpha particles during data movements. By appending a parity bit at the end of a data item, single-bit errors can be detected—but not corrected.

Odd and even are the two types of parity.

(1) **Odd Parity.** If the data is all 0s or has an even number of 1s, the parity bit set to 1 makes the total number of 1s odd.

(2) **Even Parity.** If the data has an odd number of 1s, the parity bit set to 1, makes the total number of 1s even.

The example below shows odd and even parity bits for various data values.

Data(3:0)	Odd Parity	Even Parity
0 1 1 0	1	0
0 0 0 0	1	0
1 1 1 1	1	0
1 1 0 1	0	1

Parity allows single-bit error detection, but it does not identify the bit in error. For example, suppose an odd-parity value of 00011 arrives. The last bit is the parity bit, and since odd parity demands an odd number (1, 3, 5) of 1s, this data is in error: it has an even number of 1s. However, it is impossible to tell which bit is in error.

Error Checking Operation

The processor verifies data correctness by parity code as it passes data through the system interface.

System Interface An 8-bit system address and data check bus, SysADC(7:0), contains check bits for the SysAD bus. The processor generates check bits for doubleword, word, or partial-word data transmitted to the system interface. However, the processor does not check data received from the system interface for external writes. By setting the NChck bit in the data identifier, it is possible to prevent the processor from checking read response data from the system interface.

The processor does not check addresses received from the system interface, but does generate correct check bits for addresses transmitted to the system interface.

The processor does not contain a data corrector; instead, the processor takes a cache error exception when it detects an error based on data check bits.

System Interface Command Bus. In the VR4400PC processor, the system interface command bus has a single parity bit, SysCmdP, that provides even parity over the 9 bits of this bus. The SysCmdP parity bit is generated when the system interface is in master state, but it is not checked when the system interface is in slave state.

Compatibility

Although the R4400 architecture has the new Master/Checker mode to support two processors that can be configured as a lock-stepped pair to improve the data integrity, the VR4400PC microprocessor does not provide this function because it cannot be used in the multiprocessor system. Therefore, only Complete Master mode can be configured during boot-time mode bit setting.

The VR4400PC microprocessor provides complete application software compatibility with the MIPS R2000, R3000, and R6000 processors. Although the architecture has evolved in response to a compromise between software and hardware resources in the computer system, this evolution maintains object-code compatibility for programs that execute in User mode. Like its predecessors, the VR4400PC microprocessor implements the MIPS Instruction Set Architecture (ISA) for user-mode programs, guaranteeing that the programs will execute on any MIPS hardware implementation.

BASIC FUNCTIONS

The new speed-doubler feature has been added to the VR4400PC microprocessor to increase performance. The VR4400PC is driven by the MasterClock frequency and generates the internal core clock, PClock, to drive the internal operation. The PClock frequency is twice the MasterClock frequency. Since the VR4400PC has a clock doubler driving its core, but not its system interface, it provides much higher performance than R3000 series microprocessors. The system interface clocks are generated by the CPU and are either the same as or half the MasterClock frequency. The internal PLL (phase-locked loop) logic is used to synchronize all the reproduced clocks and eliminate clock skew.

The VR4400PC has a 64-bit multiplexed address and data bus with 8 check bits, a 9-bit parity-protected command bus, and eight system interface handshake signals. The interface has a simple timing specification and is capable of transferring data between the processor and memory at a peak rate of 400 Mbytes/second at 50 MHz.

SYSTEM INTERFACE

A system event is one that occurs within the processor and requires access to external system resources. When a system event occurs, the processor issues a request or a series of requests called processor requests through the system interface to access some external resource and service the event. The processor's system interface must be connected to some external agent that understands the system interface protocol and can coordinate the access to system resources. System events include:

- A load that misses in the cache.
- A store that misses in the cache.
- An uncached load or store.

On a load or store miss, the cache line being replaced will be written back to main memory if it is in a dirty cache state. Under certain conditions, the cache operation instruction will also cause system events.

Processor Requests

A processor request is a request or a series of requests through the system interface to access some external resource. Processor requests include:

- Read Request for a block, doubleword, word, or partial word of data from main memory or another system resource.
- Write Provide a block, doubleword, word, or partial word of data to be written to main memory or another system resource.

External Requests

The external request is a request that an external agent issues to access the processor's caches or status registers through the system interface. The external requests include read, write, and null.

- Read Request for a word of data from some processor internal resource.
- Write Provide a word of data to be written to some processor internal resource.
- Null Require no action by the processor. They simply provide a mechanism for an external agent to return the system interface to the master state without affecting the processor.

Response Requests

Technically, the response request is an external request, but it has different characteristics than all other external requests. Thus, the system interface arbitration will not be performed for response requests. For this reason, the response request will be treated separately from all other external requests and called simply Response Request.

Processor Read and Write Request. When the processor or an external agent receives a read request, it must access the specified resource and return the requested data. For external read requests, the data is returned directly in response to the read request. But for the processor read requests, the requested data may not be returned directly in response to the processor read request. The response data may be returned at any time after read request, and the system interface is not used by the read between the read request and the read response. Therefore, the external agent may initiate an unrelated external request before returning the data in response to the processor read request. The return data is accomplished via a response request.

A processor read request is pending until the requested data has been returned. The processor read request is complete after the last word of response data has been received from the external agent. Note that the data identifier associated with the response data may signal that the returned data is erroneous; then the processor will take a bus error.

The processor write request is complete after the last word of data has been transmitted.

External Read and Write Requests. An external read request is complete after the processor returns the requested word of data. An external write request is complete after the word of data has been transmitted.

Flow Control Requests

The processor must manage the flow of processor requests and external requests. The processor controls the flow of external requests by the external request arbitration signals $\overline{\text{ExtRqst}}$ and $\overline{\text{Release}}$. An external agent must acquire mastership of the system interface before submitting an external request. The external agent submits a request by asserting $\overline{\text{ExtRqst}}$ and waiting for the processor to assert $\overline{\text{Release}}$ for one cycle. The processor will not assert $\overline{\text{Release}}$ until it is ready to accept an external request. Mastership of the system is always returned to the processor after an external request has been issued, and the processor will not accept a subsequent external request until it has finished the current one.

While attempting to issue a processor request, the processor will accept the external request and respond to $\overline{\text{ExtRqst}}$ by releasing the system interface to slave state. The processor can complete its entire request before an external request or release the system interface to slave state for the external request and reissue the processor request after completion of the external request. Note that the rules for governing the issue cycle are strictly applied to determine the processor action.

The processor provides signals $\overline{\text{RdRdy}}$ and $\overline{\text{WrRdy}}$ to allow an external agent to manage the flow of processor requests. $\overline{\text{RdRdy}}$ controls the flow of processor read, invalidate, and update requests; $\overline{\text{WrRdy}}$ controls the flow of processor write requests. Processor null write requests must always be accepted, since they cannot be delayed by either $\overline{\text{RdRdy}}$ or $\overline{\text{WrRdy}}$.

The processor samples $\overline{\text{RdRdy}}$ to determine the issue cycle for a processor read, invalidate, or update request that is defined to be the first address cycle for the request that asserted $\overline{\text{RdRdy}}$ two cycles previously. And the processor samples $\overline{\text{WrRdy}}$ to determine the issue cycle for a processor write request that is defined to be the first address cycle for the request that asserted $\overline{\text{WrRdy}}$ two cycles previously. If the processor issues a read or write request when neither $\overline{\text{RdRdy}}$ nor $\overline{\text{WrRdy}}$ is active, the processor will repeat the address cycle for the request until the issue cycle is accomplished.

Once the issue cycle is accomplished, data transmission will begin for a request that includes data. There will always be one and only one issue cycle for any processor request.

Processor requests are managed by the processor in two distinct modes: nonsecondary cache mode and secondary cache mode. These modes reflect the presence or absence of a secondary cache and are programmable through the boot-time mode control interface

The VR4400PC, which is in the small configuration package, must be programmed to run in the nonsecondary (primary) cache mode. In this mode, the processor will issue requests in a strict sequential fashion; that is, the processor is only allowed to have one request pending at any time. The processor will submit a read request and wait for a response request before submitting any subsequent requests. The processor write request is submitted only if there are no reads pending.

The external agent must be capable of accepting a processor read/write request at any time when no processor read request is pending and the signal $\overline{\text{RdRdy}}$/$\overline{\text{WrRdy}}$ has been asserted for at least two cycles.

HANDLING REQUESTS

The VR4400PC microprocessor generates a request or a series of requests through the system interface to satisfy system events. Processor requests are managed in primary cache mode. (Secondary cache is "not present" in the VR4400PC.)

Primary Cache Miss on a Load

When the processor misses in the primary cache on a load, it must obtain the cache line that contains the data element to be loaded from an external agent before it can proceed. If the new cache line will replace a current cache line that is in the dirty exclusive state, the current cache line must be written back before the new line can be loaded in the primary cache.

The processor examines the coherency attribute in the TLB entry for the page that contains the requsted cache line, and, if the coherency attribute is noncoherent, it issues a noncoherent read request.

The processor issues a read request for the cache line that contains the data element to be loaded. It then waits for an external agent to provide the read data in response to the read request. Then, if the current cache

line must be written back, the processor issues a write request for the current cache line.

Primary Cache Miss on a Store

When the processor misses in the primary cache on a store, it must obtain the cache line that contains the target location of the store from an external agent before it can proceed. If the new cache line will replace a current cache line that is in the dirty exclusive state, the current cache line must be written back before the new line can be loaded in the primary cache.

The processor examines the coherency attribute in the TLB entry for the page that contains the requested cache line. If the coherency attribute is noncoherent, a noncoherent read request is issued.

The processor issues a read request for the cache line that contains the data element to be loaded. It then waits for an external agent to provide the read data in response to the read request. Then, if the current cache line must be written back, the processor issues a write request for the current cache line.

Uncached Load or Store

When the processor performs an uncached load, it issues a noncoherent read request. When the processor performs an uncached store, it issues a write request.

Cache Operations

The processor provides a variety of cache operations for use in maintaining the state and contents of the primary cache. During execution of the cache operation instructions, the processor may issue write requests.

External Request Handling

An external agent must arbitrate with the processor for access to the system interface before it can submit an external request. The external agent signals that it wishes to begin an external request and waits for the processor to signal that it is ready to accept the request before issuing any new external requests. Based on its internal state and the current state of the system interface, the processor decides when to accept a new external request. The processor signals that it is ready to accept an external request based on the following criteria.

(1) If there are no processor requests pending, the processor decides, based on its internal state, whether to accept the external request or to issue a new processor request. The processor may issue a new processor request while the external agent is requesting access to the system interface to submit an external request.

(2) The processor will accept an external request after completing a processor that is in progress.

(3) While waiting for the assertion of $\overline{\text{RdRdy}}$ to issue a processor read request, the processor will accept an external request provided that the request is delivered to the processor one or more cycles before $\overline{\text{RdRdy}}$ is asserted.

(4) While waiting for the assertion of $\overline{\text{WrRdy}}$ to issue a processor write request, the processor will accept an external request provided that the request is delivered to the processor one or more cycles before $\overline{\text{WrRdy}}$ is asserted.

CLOCKING INTERFACE

The MasterClock provides the fundamental timing and the internal operating frequency for the VR4400PC microprocessor. Based on the MasterClock, a variety of clock frequencies are generated internally for internal operation and external system interaction. The PClock, twice the MasterClock frequency, supports the internal operation; the SClock is used for synchronization of external system interface signals, such as sampling the output signals and latching the input signals.

In order to work with the slow system interface, the SClock, TClock, and RClock speeds can be programmed as 1/2, 1/3, 1/4, 1/6, or 1/8 the PClock frequency through boot-time mode bit setting.

To align SyncOut, PClock, SClock, TClock, and RClock, internal phase-locked loop (PLL) circuits of the VR4400PC generate aligned clocks based on SyncOut/SyncIn. Since the PLL circuits by their nature are only capable of generating aligned clocks for MasterClock frequencies within a limited range, the minimum and maximum frequencies will be applied for MasterClock for various speed ratings of the VR4400PC.

The clocks generated using PLL circuits contain some inherent inaccuracy, or jitter, in their alignment with respect to the MasterClock. That is, a clock aligned with MasterClock by the processor's PLL circuits may lead or trail MasterClock by an amount as large as the related maximum jitter. Maximum jitter is also an important timing parameter for the clocks generated by various speed ratings of the VR4400PC.

The input signals of the VR4400PC should meet setup time t_{DS} and hold time t_{DH} requirements with respect to SClock. The setup and hold times are required for propagating data through the processor's input buffers and should satisfy the processor's input latches.

The output signals of the VR4400PC have minimum output delay t_{DM} and maximum clocking delay t_{DO} after the rising edge of the SClock. This drive-off time is the sum of the maximum delay through the processor's output drivers and the maximum clock-to-Q delay of the processor's output registers.

Certain processor inputs, such as V_{DD}Ok, $\overline{\text{ColdReset}}$, and $\overline{\text{Reset}}$, are sampled based on MasterClock, while certain processor outputs, such as Status(7:0), are driven based on MasterClock. The same setup, hold, and drive-off parameters apply to these inputs and outputs, but they are with respect to MasterClock instead of SClock.

The values of t_{DS}, t_{DH}, and t_{DO} for various speed ratings of the VR4400PC are in the AC Characteristics tables under Electrical Specifications.

Clock Interfacing in a Phase-Locked System

When the processor is in a phase-locked system, components of the external agent must phase-lock their operation to a common MasterClock. In such a system, data delivery and data sampling have common characteristics for all components, even if the components have different delay values. The transmission time (the time a signal has to propagate along the trace from one component to another) between any two components A and B of a phase-locked system can be calculated from the following equation:

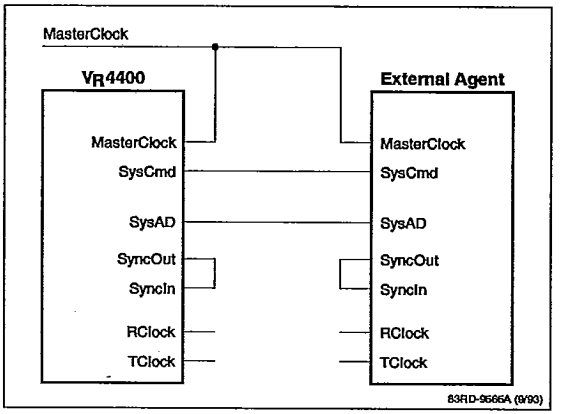
Transmission time = (SClock period)
- (t_{DO} for A)
- (t_{DS} for B)
- (Clock jitter for A max)
- (Clock jitter for B max)

Figure 10 is the block diagram of a phase-locked system employing the VR4400PC processor.

Clock Interfacing Without Phase-Lock

When the processor is used in a system in which the external agent cannot phase-lock to a common MasterClock, outputs RClock and TClock may be used to clock the remainder of the system. Two clocking methodologies are described below: one for interfacing gate-array devices and the other for interfacing discrete CMOS logic devices.

Figure 10. System With Phase-Lock



Interface to a Gate Array System. When interfacing a gate array system, both RClock and TClock are used for clocking within the gate array. The gate array buffers RClock internally and uses the buffered version to clock registers that sample processor outputs. These sample registers should be immediately followed by staging registers clocked by an internally buffered version of TClock. The buffered version of TClock should be the global system clock for the logic inside the gate array and the clock for all registers that drive processor inputs

The staging registers place a constraint on the sum of the clock-to-Q delay of the sample registers and the setup time of the synchronizing registers inside the gate array:

Clock-to-Q delay + Sync register setup time
≤ 0.25 (RClock period)
- (Maximum RClock jitter)
- (Maximum delay mismatch for internal RClock and TClock buffers)

The transmission time for a signal from the processor to an external agent composed of gate arrays in a system without phase-lock can be calculated from the following equation:

Transmission time = (75% of TClock period)
- (t_{DO} for VR4400PC)
+ (Minimum external clock buffer delay)
- (External sample register setup time)
- (Maximum VR4400PC internal clock jitter)
- (Maximum RClock jitter)

The transmission time for a signal from an external agent composed of gate arrays to the processor in a system without phase-lock can be calculated from the following equation:

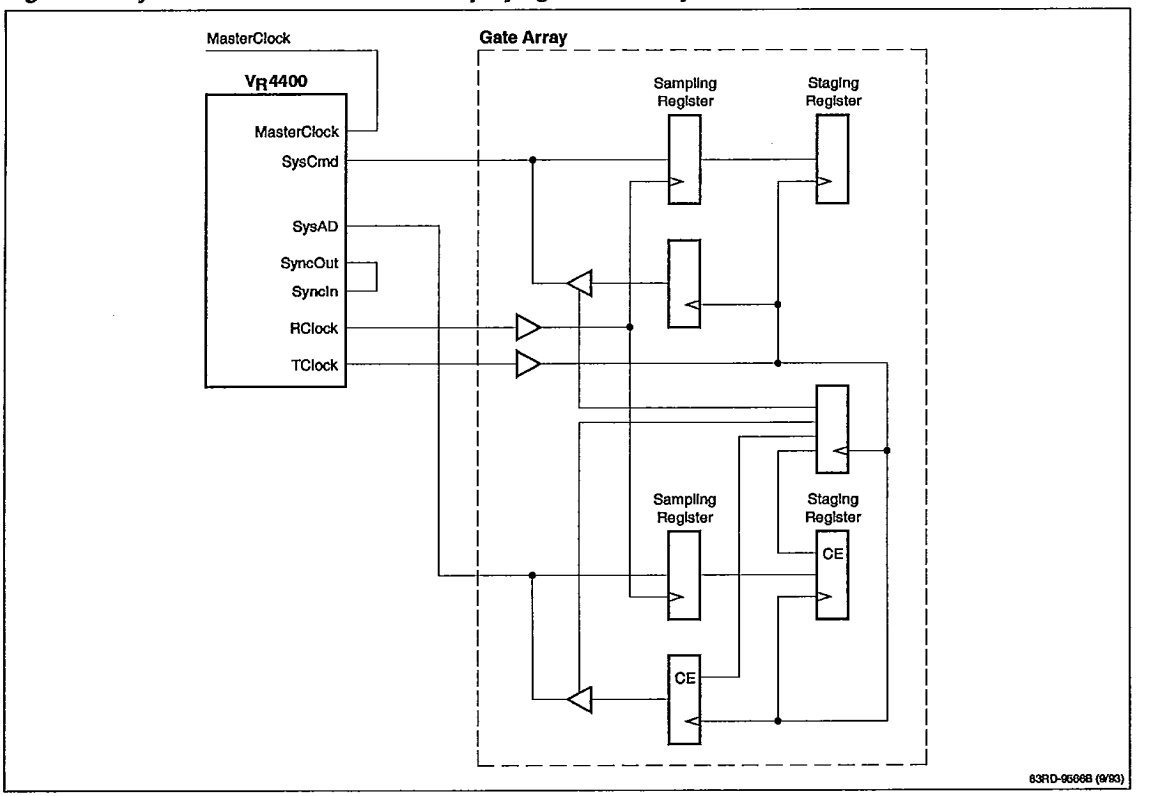
Transmission time = (TClock period)
- t_{DS} for VR4400PC)
- (Maximum external clock buffer delay)
- (Maximum external output register clock-to-Q delay)
- (Maximum TClock jitter)
- (Maximum VR4400PC internal clock jitter)

Figure 11 is the block diagram of a system without phase-lock employing the VR4400PC processor and an external agent implemented as a gate array.

Interface to a CMOS Logic System. When interfacing a CMOS logic system, matched delay clock buffers allow the processor to generate aligned clocks for the external logic. One of the matched delay clock buffers is inserted in the procesor's SyncOut-to-SyncIn clock alignment path, skewing SyncOut, MasterOut, SClock, RClock, and TClock to lead MasterClock by the delay of the matched delay clock buffer while leaving PClock aligned with MasterClock.

The remaining matched delay clock buffers can be used to generate a buffered version of TClock aligned with MasterClock. The alignment error of the buffered TClock is the sum of the maximum delay mismatch of the matched delay clock buffers and the maximum TClock jitter. The buffered TClock is used to clock registers that sample processor outputs, as the global system clock for the discrete logic that forms the

Figure 11. System Without Phase-Lock Employing a Gate Array



external agent, and to clock registers that drive processor inputs.

The transmission time for a signal from the processor to an external agent composed of discrete CMOS logic devices can be calculated from the following equation:

Transmission time = (TClock period)
- (t_{DO} for VR4400PC)
- (External sample register setup time)
- (Maximum external clock buffer delay mismatch)
- (Maximum VR4400PC internal clock jitter)
- (Maximum TClock jitter)

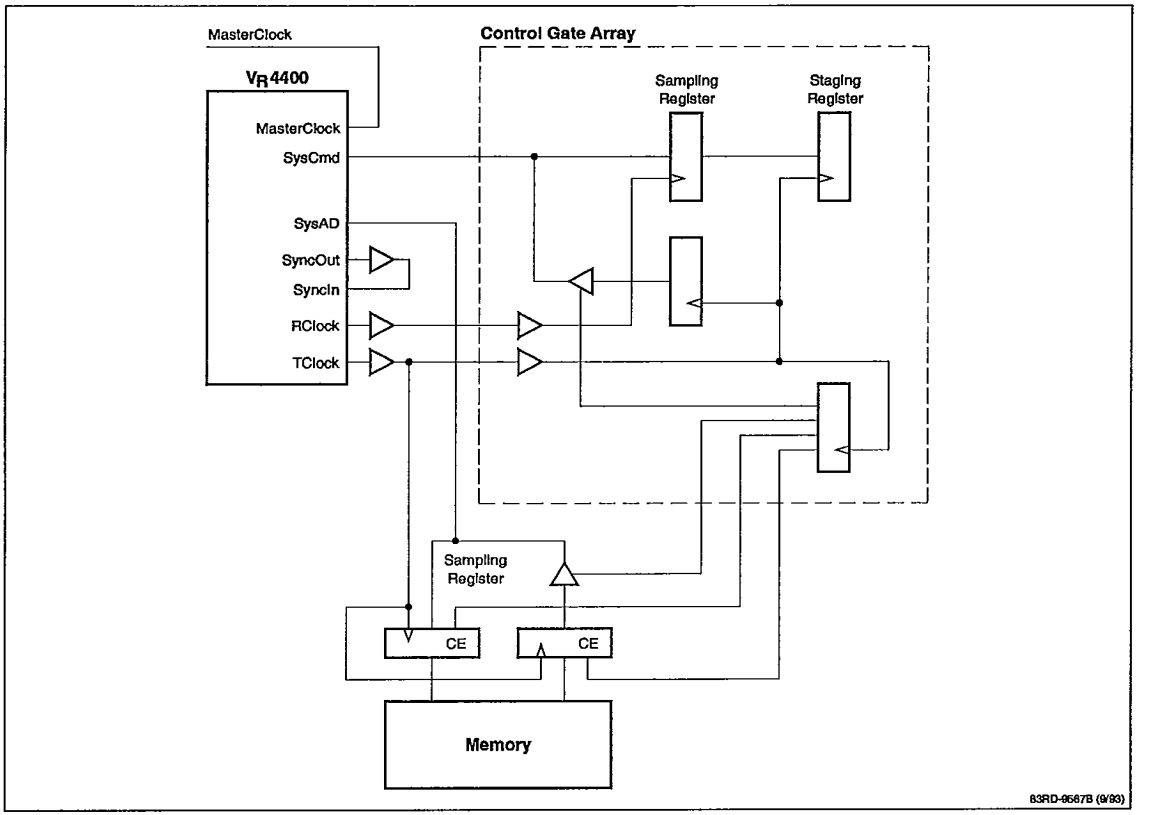
The transmission time for a signal from an external agent composed of discrete CMOS logic devices can be calculated from the following equation:

Transmission time = (TClock period)
- (t_{DS} for VR4400PC)
- (Maximum external output register clock-to-Q delay)
- (Maximum external clock buffer delay mismatch)
- (Maximum VR4400PC internal clock jitter)
- (Maximum TClock jitter)

With this clocking methodology, the hold time of data driven from the processor to an external sampling register is a critical parameter. To guarantee hold time, the minimum output delay of the processor, t_{DM}, must be greater than the sum of the minimum hold time for the external sampling register, the maximum clock jitter for VR4400PC internal clocks, the maximum TClock jitter, and the maximum delay mismatch of the external clock buffers.

Figure 12 is the block diagram of a system without phase-lock employing the VR4400PC and an external agent composed of both a gate array and discrete CMOS logic devices.

Figure 12. System Without Phase-Lock Employing a Gate Array and CMOS Logic



INITIALIZATION INTERFACE

The operation of the VR4400PC microprocessor may be reset by a multilevel reset sequence using the V_{DD}Ok, ColdReset, and Reset inputs. A power-on or cold reset accomplishes the same thing: they both completely reset the internal state machine of the VR4400PC. A warm reset also resets the internal state machine; however the processor internal state is preserved.

Fundamental operational modes for the processor are set up by the initialization interface, which is a serial interface operating at a MasterClock frequency divided by 256. The low-frequency operation allows the initialization information to be stored in a low-cost EPROM.

Immediately after the V_{DD}Ok signal is asserted, the processor reads a serial bit stream of 256 bits on ModeIn to initialize all fundamental operational modes.

After initialization is complete, the processor continues to drive the serial clock output, but no further initialization bits are read.

Initialization Interface Operation

Refer to figure 13 and the following commentary.

(1) ModeIn: Serial boot-time mode data in.

(2) ModeClock: Serial boot-time mode data clock out at the MasterClock frequency divided by 256.

(3) While V_{DD}Ok is deasserted, the ModeClock output is held asserted.

(4) When V_{DD}Ok is asserted, the first bit in the initialization bit stream must be present at the ModeIn input.

(5) The processor synchronizes the ModeClock output at the time V_{DD}Ok is asserted; the first rising edge of the ModeClock will occur 256 MasterClock cycles later.

(6) After each rising edge of the ModeClock, the next bit of the initialization bit stream must be presented at the ModeIn input. The processor will sample exactly 256 initialization bits from the ModeIn input on the rising edge of the ModeClock.

Boot-Time Modes

The correspondence between bits of the intialization bit stream and processor mode settings is illustrated in table 5. Bit 0 of the bit stream is presented to the processor when V_{DD}Ok is deasserted.

Figure 13. Timing of the Boot-Time Mode Control Interface

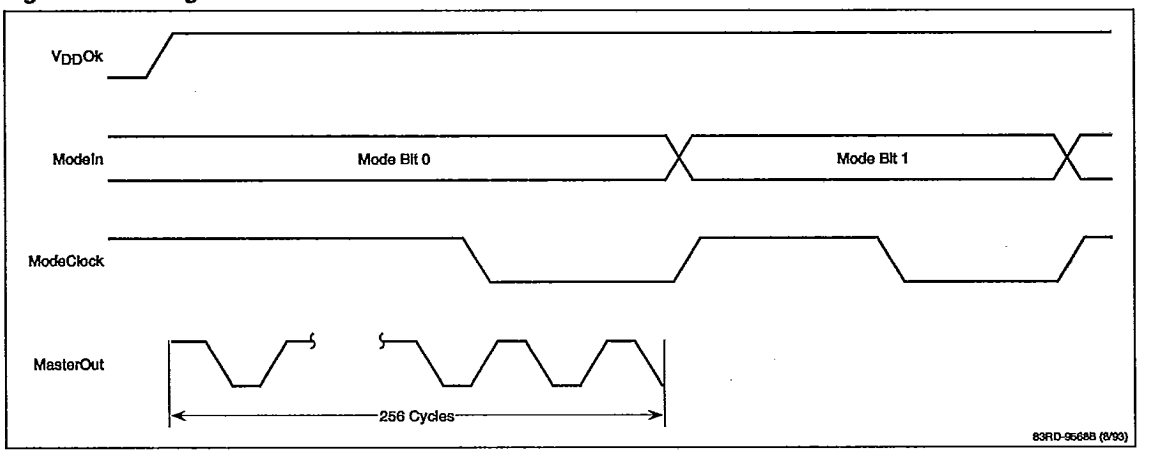


Table 5. Boot-Time Mode

Bit	Value	Processor Mode Setting
0		Block read response ordering:
	0	Sequential ordering
	1	Sub-block ordering
1		System interface check bus checking:
	0	SECDED error checking and correcting mode
	1	Byte parity
2		Byte ordering:
	0	Little endian
	1	Big endian
3		Dirty shared mode (enables transition to dirty shared state on processor update successful):
	0	Dirty enabled
	1	Dirty disabled
4		Secondary cache:
	0	Present
	1	Not present
5:6		System interface port width:
	0	64 bits
	1-3	Reserved (Note 1)
7		Secondary cache interface port width:
	0	128 bits
	1	64 bits
8		Secondary cache organization:
	0	Unified
	1	Split I/D
9:10		Secondary cache line size (MSB 10):
	0	4 words
	1	8 words
	2	16 words
	3	32 words
11:14		System interface data rate (MSB 14): D = data, x = don't care:
	0	D
	1	DDx
	2	DDxx
	3	DxDx
	4	DDxxx
	5	DDxxxx
	6	DxxDxx
	7	DDxxxxxx
	8	DxxxDxxx
	9-15	Reserved (Note 1)
15:17		PClock-to-SClock divisor (frequency relationship between SClock/RClock/TClock and PClock (MSB 17):
	0	Divide by 2
	1	Divide by 3
	2	Divide by 4
	3	Divide by 6
	4	Divide by 8
	5-7	Reserved (Note 1)
18	0	Reserved (required value)

Table 5. Boot-Time Mode (cont)

Bit	Value	Processor Mode Setting
19		Timer/Interrupt enable (allows timer, otherwise the interrupt used by the timer becomes a general-purpose interrupt):
	0	Enabled
	1	Disabled
20		Potential invalidate enable (allows potential invalidates to be issued; otherwise only normal invalidates are issued):
	0	Enabled
	1	Disabled
21:24		Secondary cache write deassertion delay; t_{WrSup} in PCycles (MSB 24)
25:26		Secondary cache write deassertion delay 2; t_{Wr2Dly} in PCycles (MSB 26)
27:28		Secondary cache write deassertion delay 1; t_{Wr1Dly} in PCycles (MSB 28)
29		Secondary cache write recovery time; t_{WrRc} in PCycles:
	0	0 cycle
	1	1 cycle
30:32		Secondary cache disable time; t_{Dis} in PCycles (MSB 32)
33:36		Secondary cache read cycle time 2; t_{RdCyc2} in PCycles (MSB 36)
37:40		Secondary cache read cycle time 1; t_{RdCyc1} in PCycles (MSB 40)
41		Secondary cache 64-bit mode uses upper/lower half of SCData (127:0):
	0	Lower half
	1	Upper half
42:45	0	Reserved (Note 2)
46		VR4400 package type.
	0	Large (447-pin): SC and MC
	1	Small (179-pin): PC
47:49		Reserved (Note 2)
50:52		Drive outputs at N x MasterClock
	001	0.5 x MasterClock
	010	0.75 x MasterClock
	100	1.0 x MasterClock
	Other	Reserved (Note 1)
53:56		Initial values for the state bits that determine the pulldown di/dt and switching speed of the output buffers (MSB 53):
	0	Fastest pulldown rate
	1-14	Intermediate pulldown rates
	15	Slowest pulldown rate
57:60		Initial values for the state bits that determine the pullup di/dt and switching speed of the output buffers (MSB 57):
	0	Slowest pullup rate
	1-14	Intermediate pullup rates
	15	Fastest pullup rate

Table 5. Boot-Time Mode (cont)

Bit	Value	Processor Mode Setting
61		Enables the negative feedback loop that determines the di/dt and switching speed of the output buffers only during Cold Reset:
	0	Disable di/dt control mechanism
	1	Enable di/dt control mechanism
62		Enables the negative feedback loop that determines the di/dt and switching speed of the output buffers only during Cold Reset and during normal operation.
	0	Disable di/dt control mechanism
	1	Enable di/dt control mechanism
63		Enable PLLs that match MasterIn and produce RClock, TClock, SClock, and internal clocks:
	0	Enable PLLs
	1	Disable PLLs
64		Controls when output only pins are tristated:.
	0	Only when $\overline{\text{ColdReset}}$ is asserted
	1	When $\overline{\text{Reset}}$ or $\overline{\text{ColdReset}}$ is asserted
65:255	0	Reserved (Note 2)

Notes:

(1) Selecting a reserved value results in undefined processor behavior.

(2) Zeros must be scanned in.

RESET

The VR4400PC microprocessor supports three types of resets:

- Power-On Reset: Starts from power supply turning on.
- Cold Reset: Restarts all clocks, but power supply remains stable. Processor operating parameters do not change.
- Warm Reset: Restarts processor, but does not affect clocks.

Power-On Reset

The sequence for a power-on reset follows:

(1) Stable V_{DD} of at least 4.75 (3.135) volts from the +5-V (+3.3-V) power supply is applied to the processor. A stable continuous system clock at the processor's desired operational frequency is also supplied.

(2) After at least 100 milliseconds of stable V_{DD} and MasterClock, the V_{DD}Ok input to the processor may be asserted. The assertion of V_{DD}Ok causes the processor to initialize the operating parameters. After the mode bits have been read in, the processor allows its internal phase-locked loops to lock, stabilizing the processor internal clock, PClock, the SyncOut-to-SyncIn clock path, and the master clock output MasterOut.

Cold Reset

A cold reset can begin when the processor has read the initialization data stream.

(1) Once the boot-time mode control serial data stream has been read by the processor, the $\overline{\text{ColdReset}}$ input may be deasserted. $\overline{\text{ColdReset}}$ must remain asserted for at least 64 MasterClock cycles after the assertion of V_{DD}Ok and deasserted synchronously with MasterClock

(2) Processor internal clock SClock and system interface clocks TClock and RClock begin to cycle with the deassertion of $\overline{\text{ColdReset}}$. The deassertion edge of $\overline{\text{ColdReset}}$ synchronizes the edges of SClock, TClock, and RClock, potentially across multiple processors in a multiprocessor system.

(3) After $\overline{\text{ColdReset}}$ is deasserted and SClock, TClock, and RClock have stabilized, $\overline{\text{Reset}}$ is deasserted to allow the processor to begin to run. $\overline{\text{Reset}}$ must be held asserted for at least 64 MasterClock cycles after deassertion of $\overline{\text{ColdReset}}$ and deasserted synchronously with MasterClock.

$\overline{\text{ColdReset}}$ must be asserted when V_{DD}Ok asserts. The behavior of the processor is undefined if V_{DD}Ok asserts while $\overline{\text{ColdReset}}$ is deasserted.

Warm Reset

To produce a warm reset, the $\overline{\text{Reset}}$ input may be asserted synchronously with MasterClock and held asserted for at least 64 MasterClock cycles before being deasserted synchronously with MasterClock. The processor internal clocks, PClock and SClock, and the system interface clocks, TClock and RClock, are not affected by a warm reset, and the boot-time mode control serial data stream is not read by the processor on a warm reset.

The master clock output, MasterClock, is provided for generating the reset related signals for the processor that must be synchronous with MasterClock.

After a power-on reset, cold reset, or warm reset, all processor internal state machines are reset, and the processor begins execution at the reset vector. All processor internal states are preserved during a warm reset, although the precise state of the caches willdepend on whether a cache miss sequence has been interrupted by resetting the processor state machines.

JTAG INTERFACE

The VR4400PC microprocessor provides a boundary scan interface using the industry standard JTAG protocol.

The JTAG boundary scan mechanism provides a capability for testing the interconnect between the VR4400PC, the printed circuit board to which it is attached, and the other components on the board. The JTAG boundary scan mechanism does not provide any capability for testing the VR4400PC itself.

In accordance with the JTAG specification, the VR4400PC contains a TAP controller, JTAG Instruction Register, JTAG Boundary Scan Register, JTAG Identification Register, and JTAG Bypass Register. However, the VR4400PC JTAG implementation provides only the external test functionality of the boundary scan register.

ELECTRICAL SPECIFICATIONS

Power Distribution

The VR4400PC microprocessor operates with high-frequency clocks. Dc power surges can result when multiple clock output buffers drive new signal levels simultaneously. For clean on-chip power, about 30 pins each are assigned to V_{DD} and GND inputs.

Liberal decoupling capacitors should be installed near the VR4400PC. Driving the 64-bit system address/data bus at high frequencies can cause transient power surges, particularly with large capacitive loads.

Low-inductance capacitors and interconnects are recommended for best high-frequency performance. Inductance can be reduced by shortening circuit board traces between the CPU and decoupling capacitors as much as possible. Capacitors specifically for PGA packages are commercially available.

Unused Inputs

For reliable operation, connect unused active-low inputs to V_{DD} through a pullup resistor, and connect active-high inputs directly to GND. Pins designated NC should always remain unconnected.

Capacitive Load

Capacitive load derating (CLD) for all versions of the VR4400PC is 2 ns/25 pF maximum.

Absolute Maximum Ratings

Parameter	Symbol	Min	Max	Unit
Supply voltage	V_{DD}			
5-volt version		−0.5	7.0	V
3.3-volt version		−0.5	3.8	V
Input voltage (Note 1)	V_{IN}			
5-volt version		−0.5	7.0	V
3.3-volt version		−0.5	V_{DD} + 0.5	V
Storage temperature	T_{ST}	−65	+150	°C
Operating case temperature	T_C	0	+85	°C

Notes:

(1) V_{IN} min = −3.0 V for pulse width < 15 ns.

(2) Not more than one output should be shorted at a time and for not more than 30 seconds.

Exposure to Absolute Maximum Ratings for extended periods may affect device reliability; exceeding the ratings could cause permanent damage.

Typical Power Dissipation in Watts

MasterClock	V_{DD} = 5 Volts	V_{DD} = 3.3 Volts
50 MHz	8.6	4.0
67 MHz	11.3	5.3
75 MHz	12.5	6.0

DC Characteristics

Functional operation range: V_{DD} = 5.0 or 3.3 volts ±5%; T_C = 0 to +80°C

Parameter	Symbol	Min	Typ	Max	Unit	Conditions
Output voltage, high	V_{OH}					V_{DD} = minimum;
5-volt version		3.5			V	I_{OH} = −4 mA or
3.3-volt version		2.4			V	I_{OL} = 4 mA
Output voltage, low	V_{OL}			0.4	V	
Clock output voltage, high (Note 2)	V_{OHC}					
5-volt version		4.0			V	
3.3-volt version		2.7			V	
Input voltage, high	V_{IH}	2.2		V_{DD} + 0.5	V	
Input voltage, low	V_{IL}	−0.5		0.8	V	(Note 1)
MasterClock input voltage, high	V_{IHC}	0.8 V_{DD}		V_{DD} + 0.5	V	
MasterClock input voltage, low	V_{ILC}	−0.5		0.2 V_{DD}	V	(Note 1)
Input leakage current	I_{Leak}			10	μA	
Input/output leakage current	IO_{Leak}			20	μA	
Input capacitance	C_{In}			10	pF	
Output capacitance	C_{Out}			10	pF	
Operating current, 5-volt versions	I_{DD}					V_{DD} = 5 V; T_C = 0°C
50-MHz			1.8	2.3	A	
67-MHz			2.3	2.9	A	
75-MHz			2.6	3.2	A	
Operating current, 3.3-volt versions	I_{DD}					V_{DD} = 3.3 V; T_C = 0°C
50-MHz			1.2	1.5	A	
67-MHz			1.5	1.9	A	
75-MHz			1.8	2.2	A	

Notes:

(1) V_{IL} min = −3.0 V for pulse width < 15 ns, except for MasterClock input.

(2) Applies to TClock, RClock, MasterOut, and ModeClock outputs.

AC Characteristics; 5-Volt, 100-MHz Version

V_{DD} = 5 V ±5%; T_C = 0 to +80°C; C_L = 50 pF

Parameter	Symbol	Min	Max	Unit	Conditions
MasterClock frequency		25	50	MHz	(Note 1)
MasterClock period	t_{MCP}	20	40	ns	
MasterClock high	t_{MCHigh}	4		ns	Transition ≤ 5 ns
MasterClock low	t_{MCLow}	4		ns	Transition ≤ 5 ns
MasterClock rise time	t_{MCRise}		5	ns	
MasterClock fall time	t_{MCFall}		5	ns	
Clock jitter	$t_{MCJitter}$		±500	ps	
ModeClock period	$t_{ModeCKP}$		256 t_{MCP}	ns	
JTAG clock period	$t_{JTAGCKP}$	4 t_{MCP}		ns	
Data output (Notes 2-5)	t_{DO}	3.5	10	ns	Max slew rate
		6	16	ns	Min slew rate
Data setup time	t_{DS}	5		ns	(Note 5)
Data hold time	t_{DH}	1.5		ns	(Note 5)
Mode data setup time	t_{MDS}	3		MClk	
Mode data hold time	t_{MDH}	0		MClk	
PClock to output (Notes 2-4)	t_{SCO}	2	10	ns	Max slew rate
		6	16	ns	Min slew rate
Four-word read cycle length	t_{Rd1Cyc}	4	15	PClk	(Note 6)
Cycles between read and write	t_{Dis}	2	7	PClk	
Eight-word read cycle length	t_{Rd2Cyc}	3	15	PClk	

Notes:

(1) Operation of the VR4400PC is guaranteed only with the phase-locked loop enabled.

(2) Maximum slew rate: Modebit (53:56) = 0 and (57:60) = F
Minimum slew rate: Modebit (53:56) = F and (57:60) = 0
MC 0.5 drive time: Modebit (50:52) = 100
MC 0.75 drive time: Modebit (50:52) = 010
MC 1.0 drive time: Modebit (50:52) = 001

(3) When the dynamic output slew rate control Modebit 61 or 62 is enabled, the initial values for the pullup and pulldown rates should be set to the slowest value: Modebit (53:56) = F and (57:60) = 0.

(4) Timing is measured from 1.5 V of SClock to 1.5 V of the signal.

(5) Data output, setup, and hold times apply to all logic signals driven out of or driven into the VR4400PC on the system interface.

(6) Number of cycles is configured through the boot-time mode control.

AC Characteristics; 5-Volt, 133-MHz Version

V_{DD} = 5 V ±5%; T_C = 0 to +80°C; C_L = 50 pF

Parameter	Symbol	Min	Max	Unit	Conditions
MasterClock frequency		25	67	MHz	(Note 1)
MasterClock period	t_{MCP}	15	40	ns	
MasterClock high	t_{MCHigh}	3		ns	Transition ≤ 5 ns
MasterClock low	t_{MCLow}	3		ns	Transition ≤ 5 ns
MasterClock rise time	t_{MCRise}		4	ns	
MasterClock fall time	t_{MCFall}		4	ns	
Clock jitter	$t_{MCJitter}$		±500	ps	
ModeClock period	$t_{ModeCKP}$		256 t_{MCP}	ns	
JTAG clock period	$t_{JTAGCKP}$	4 t_{MCP}		ns	
Data output (Notes 2-5)	t_{DO}	2	7	ns	Max slew rate
		6	12	ns	Min slew rate
Data setup time	t_{DS}	5		ns	(Note 5)
Data hold time	t_{DH}	1.5		ns	(Note 5)
Mode data setup time	t_{MDS}	3		MClk	
Mode data hold time	t_{MDH}	0		MClk	
PClock to output (Notes 2-4)	t_{SCO}	2	7	ns	Max slew rate
		6	12	ns	Min slew rate
Four-word read cycle length	t_{Rd1Cyc}	4	15	PClk	(Note 6)
Cycles between read and write	t_{Dis}	2	7	PClk	
Eight-word read cycle length	t_{Rd2Cyc}	3	15	PClk	

Notes:

(1) Operation of the VR4400PC is guaranteed only with the phase-locked loop enabled.

(2) Maximum slew rate: Modebit (53:56) = 0 and (57:60) = F
Minimum slew rate: Modebit (53:56) = F and (57:60) = 0
MC 0.5 drive time: Modebit (50:52) = 100
MC 0.75 drive time: Modebit (50:52) = 010
MC 1.0 drive time: Modebit (50:52) = 001

(3) When the dynamic output slew rate control Modebit 61 or 62 is enabled, the initial values for the pullup and pulldown rates should be set to the slowest value: Modebit (53:56) = F and (57:60) = 0.

(4) Timing is measured from 1.5 V of SClock to 1.5 V of the signal.

(5) Data output, setup, and hold times apply to all logic signals driven out of or driven into the VR4400PC on the system interface.

(6) Number of cycles is configured through the boot-time mode control.

AC Characteristics; 5-Volt, 150-MHz Version

V_{DD} = 5 V ±5%; T_C = 0 to +80°C; C_L = 50 pF

Parameter	Symbol	Min	Max	Unit	Conditions
MasterClock frequency		25	75	MHz	(Note 1)
MasterClock period	t_{MCP}	13.3	40	ns	
MasterClock high	t_{MCHigh}	3		ns	Transition ≤ 5 ns
MasterClock low	t_{MCLow}	3		ns	Transition ≤ 5 ns
MasterClock rise time	t_{MCRise}		3.5	ns	
MasterClock fall time	t_{MCFall}		3.5	ns	
Clock jitter	$t_{MCJitter}$		±500	ps	
ModeClock period	$t_{ModeCKP}$		256 t_{MCP}	ns	
JTAG clock period	$t_{JTAGCKP}$	4 t_{MCP}		ns	
Data output (Notes 2-5)	t_{DO}	2	7	ns	Max slew rate
		6	12	ns	Min slew rate
Data setup time	t_{DS}	3.5		ns	(Note 5)
Data hold time	t_{DH}	1		ns	(Note 5)
Mode data setup time	t_{MDS}	3		MClk	
Mode data hold time	t_{MDH}	0		MClk	
PClock to output (Notes 2-4)	t_{SCO}	2	7	ns	Max slew rate
		6	12	ns	Min slew rate
Four-word read cycle length	t_{Rd1Cyc}	4	15	PClk	(Note 6)
Cycles between read and write	t_{Dis}	2	7	PClk	
Eight-word read cycle length	t_{Rd2Cyc}	3	15	PClk	

Notes:

(1) Operation of the VR4400PC is guaranteed only with the phase-locked loop enabled.

(2) Maximum slew rate: Modebit (53:56) = 0 and (57:60) = F
Minimum slew rate: Modebit (53:56) = F and (57:60) = 0
MC 0.5 drive time: Modebit (50:52) = 100
MC 0.75 drive time: Modebit (50:52) = 010
MC 1.0 drive time: Modebit (50:52) = 001

(3) When the dynamic output slew rate control Modebit 61 or 62 is enabled, the initial values for the pullup and pulldown rates should be set to the slowest value: Modebit (53:56) = F and (57:60) = 0.

(4) Timing is measured from 1.5 V of SClock to 1.5 V of the signal.

(5) Data output, setup, and hold times apply to all logic signals driven out of or driven into the VR4400PC on the system interface.

(6) Number of cycles is configured through the boot-time mode control.

AC Characteristics; 3.3-Volt, 100-MHz Version

V_{DD} = 3.3 V ±5%; T_C = 0 to +80°C; C_L = 50 pF

Parameter	Symbol	Min	Max	Unit	Conditions
MasterClock frequency		25	50	MHz	(Note 1)
MasterClock period	t_{MCP}	20	40	ns	
MasterClock high	t_{MCHigh}	5		ns	Transition ≤ 5 ns
MasterClock low	t_{MCLow}	5		ns	Transition ≤ 5 ns
MasterClock rise time	t_{MCRise}		3.5	ns	
MasterClock fall time	t_{MCFall}		3.5	ns	
Clock jitter	$t_{MCJitter}$		±500	ps	
ModeClock period	$t_{ModeCKP}$		256 t_{MCP}	ns	
JTAG clock period	$t_{JTAGCKP}$	4 t_{MCP}		ns	
Data output (Notes 2-5)	t_{DO}	3.5	10	ns	Max slew rate
		6	16	ns	Min slew rate
Data setup time	t_{DS}	3.5		ns	(Note 5)
Data hold time	t_{DH}	1.5		ns	(Note 5)
Mode data setup time	t_{MDS}	3		MClk	
Mode data hold time	t_{MDH}	0		MClk	
PClock to output (Notes 2-4)	t_{SCO}	3.5	10	ns	Max slew rate
		6	16	ns	Min slew rate
Four-word read cycle length	t_{Rd1Cyc}	4	15	PClk	(Note 6)
Cycles between read and write	t_{Dis}	2	7	PClk	
Eight-word read cycle length	t_{Rd2Cyc}	3	15	PClk	

Notes:

(1) Operation of the VR4400PC is guaranteed only with the phase-locked loop enabled.

(2) Maximum slew rate: Modebit (53:56) = 0 and (57:60) = F
Minimum slew rate: Modebit (53:56) = F and (57:60) = 0
MC 0.5 drive time: Modebit (50:52) = 100
MC 0.75 drive time: Modebit (50:52) = 010
MC 1.0 drive time: Modebit (50:52) = 001

(3) When the dynamic output slew rate control Modebit 61 or 62 is enabled, the initial values for the pullup and pulldown rates should be set to the slowest value: Modebit (53:56) = F and (57:60) = 0.

(4) Timing is measured from 1.5 V of SClock to 1.5 V of the signal.

(5) Data output, setup, and hold times apply to all logic signals driven out of or driven into the VR4400PC on the system interface.

(6) Number of cycles is configured through the boot-time mode control.

AC Characteristics; 3.3-Volt, 133-MHz Version

V_{DD} = 3.3 V ±5%; T_C = 0 to +80°C; C_L = 50 pF

Parameter	Symbol	Min	Max	Unit	Conditions
MasterClock frequency		25	67	MHz	(Note 1)
MasterClock period	t_{MCP}	15	40	ns	
MasterClock high	t_{MCHigh}	3		ns	Transition ≤ 5 ns
MasterClock low	t_{MCLow}	3		ns	Transition ≤ 5 ns
MasterClock rise time	t_{MCRise}		4	ns	
MasterClock fall time	t_{MCFall}		4	ns	
Clock jitter	$t_{MCJitter}$		±500	ps	
ModeClock period	$t_{ModeCKP}$		256 t_{MCP}	ns	
JTAG clock period	$t_{JTAGCKP}$	4 t_{MCP}		ns	
Data output (Notes 2-5)	t_{DO}	2	7	ns	Max slew rate
		6	12	ns	Min slew rate
Data setup time	t_{DS}	3.5		ns	(Note 5)
Data hold time	t_{DH}	1.5		ns	(Note 5)
Mode data setup time	t_{MDS}	3		MClk	
Mode data hold time	t_{MDH}	0		MClk	
PClock to output (Notes 2-4)	t_{SCO}	2	7	ns	Max slew rate
		6	12	ns	Min slew rate
Four-word read cycle length	t_{Rd1Cyc}	4	15	PClk	(Note 6)
Cycles between read and write	t_{Dis}	2	7	PClk	
Eight-word read cycle length	t_{Rd2Cyc}	3	15	PClk	

Notes:

(1) Operation of the VR4400PC is guaranteed only with the phase-locked loop enabled.

(2) Maximum slew rate: Modebit (53:56) = 0 and (57:60) = F
Minimum slew rate: Modebit (53:56) = F and (57:60) = 0
MC 0.5 drive time: Modebit (50:52) = 100
MC 0.75 drive time: Modebit (50:52) = 010
MC 1.0 drive time: Modebit (50:52) = 001

(3) When the dynamic output slew rate control Modebit 61 or 62 is enabled, the initial values for the pullup and pulldown rates should be set to the slowest value: Modebit (53:56) = F and (57:60) = 0.

(4) Timing is measured from 1.5 V of SClock to 1.5 V of the signal.

(5) Data output, setup, and hold times apply to all logic signals driven out of or driven into the VR4400PC on the system interface.

(6) Number of cycles is configured through the boot-time mode control.

AC Characteristics; 3.3-Volt, 150-MHz Version

V_{DD} = 3.3 V ±5%; T_C = 0 to +80°C; C_L = 50 pF

Parameter	Symbol	Min	Max	Unit	Conditions
MasterClock frequency		25	75	MHz	(Note 1)
MasterClock period	t_{MCP}	13.3	40	ns	
MasterClock high	t_{MCHigh}	3		ns	Transition ≤ 5 ns
MasterClock low	t_{MCLow}	3		ns	Transition ≤ 5 ns
MasterClock rise time	t_{MCRise}		3.5	ns	
MasterClock fall time	t_{MCFall}		3.5	ns	
Clock jitter	$t_{MCJitter}$		±500	ps	
ModeClock period	$t_{ModeCKP}$		256 t_{MCP}	ns	
JTAG clock period	$t_{JTAGCKP}$	4 t_{MCP}		ns	
Data output (Notes 2-5)	t_{DO}	2	6	ns	Max slew rate
		6	10	ns	Min slew rate
Data setup time	t_{DS}	3.5		ns	(Note 5)
Data hold time	t_{DH}	1		ns	(Note 5)
Mode data setup time	t_{MDS}	3		MClk	
Mode data hold time	t_{MDH}	0		MClk	
PClock to output (Notes 2-4)	t_{SCO}	2	6	ns	Max slew rate
		6	10	ns	Min slew rate
Four-word read cycle length	t_{Rd1Cyc}	4	15	PClk	(Note 6)
Cycles between read and write	t_{Dis}	2	7	PClk	
Eight-word read cycle length	t_{Rd2Cyc}	3	15	PClk	

Notes:

(1) Operation of the VR4400PC is guaranteed only with the phase-locked loop enabled.

(2) Maximum slew rate: Modebit (53:56) = 0 and (57:60) = F
Minimum slew rate: Modebit (53:56) = F and (57:60) = 0
MC 0.5 drive time: Modebit (50:52) = 100
MC 0.75 drive time: Modebit (50:52) = 010
MC 1.0 drive time: Modebit (50:52) = 001

(3) When the dynamic output slew rate control Modebit 61 or 62 is enabled, the initial values for the pullup and pulldown rates should be set to the slowest value: Modebit (53:56) = F and (57:60) = 0.

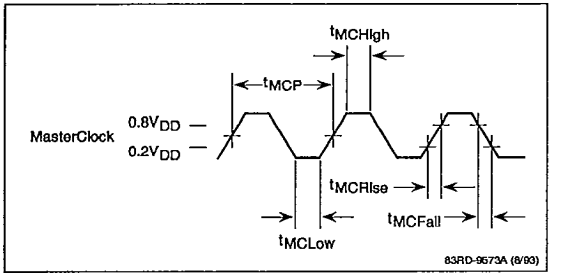
(4) Timing is measured from 1.5 V of SClock to 1.5 V of the signal.

(5) Data output, setup, and hold times apply to all logic signals driven out of or driven into the VR4400PC on the system interface.

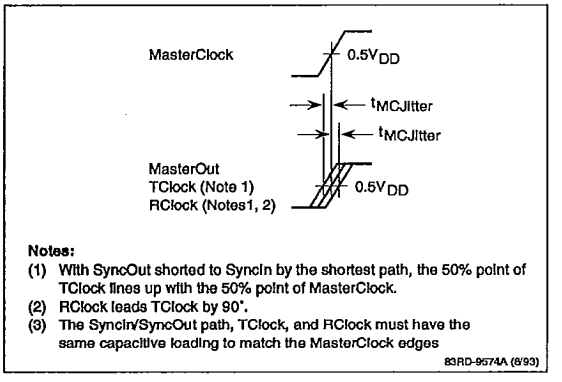
(6) Number of cycles is configured through the boot-time mode control.

Timing Diagrams

Master Clock

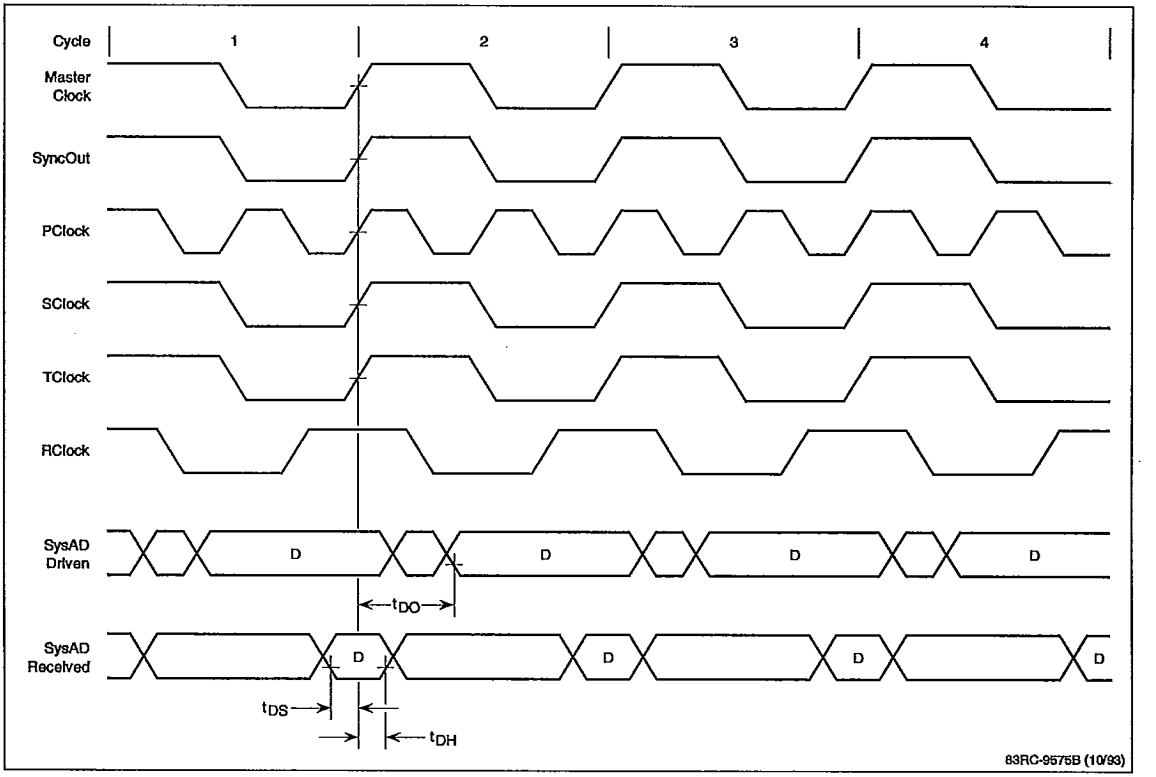


Clock Jitter



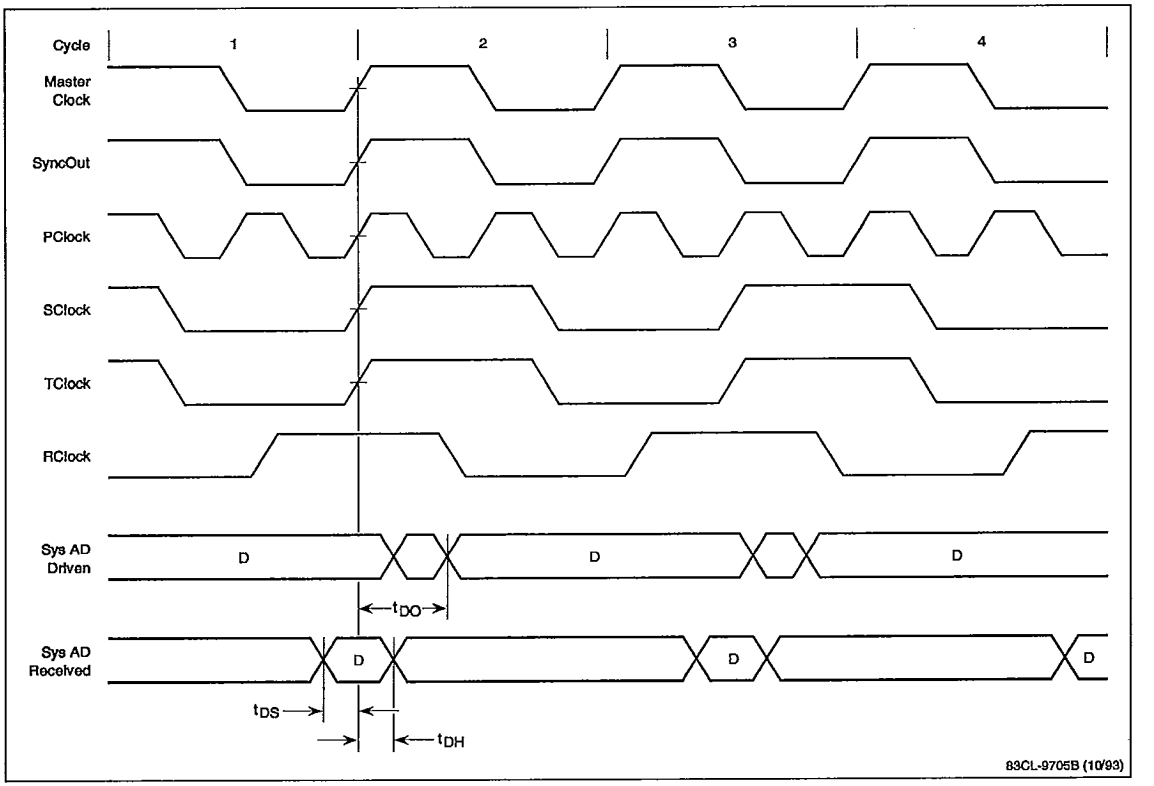
Timing Diagrams (cont)

Processor Clock; PClock to SClock, Divisor of 2



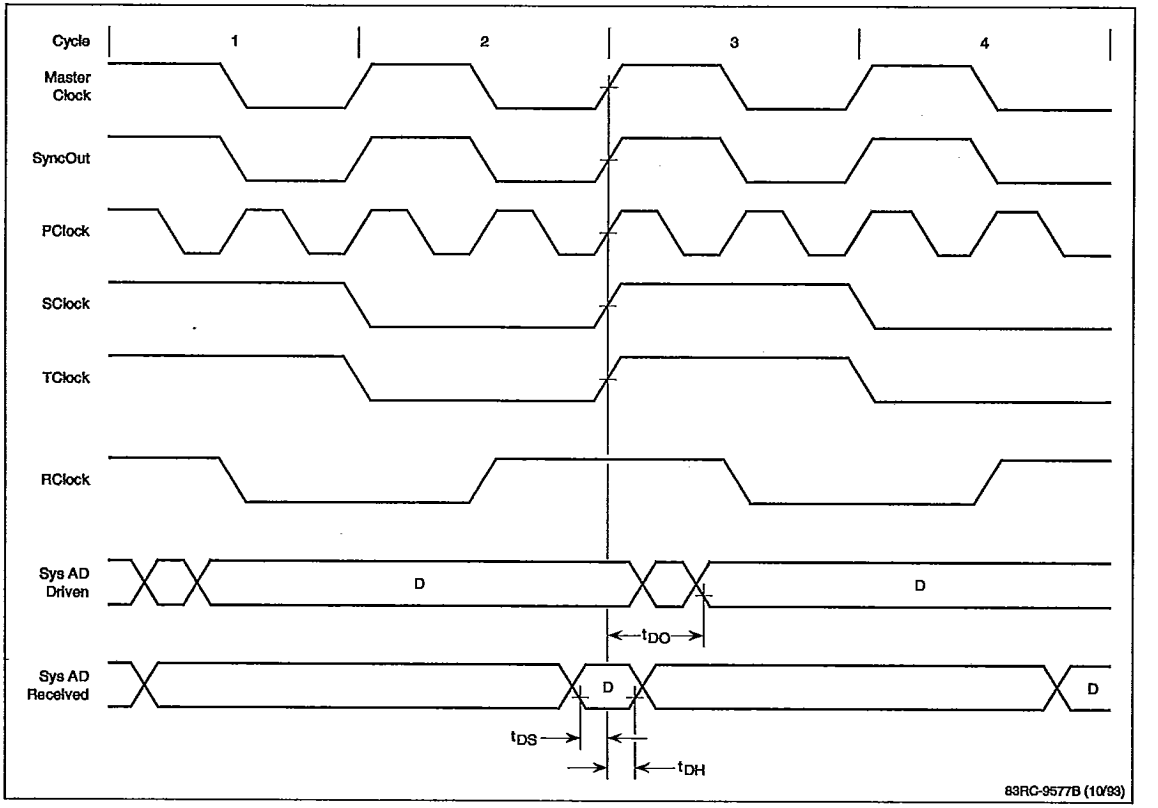
Timing Diagrams (cont)

Processor Clock; PClock to SClock, Divisor of 3



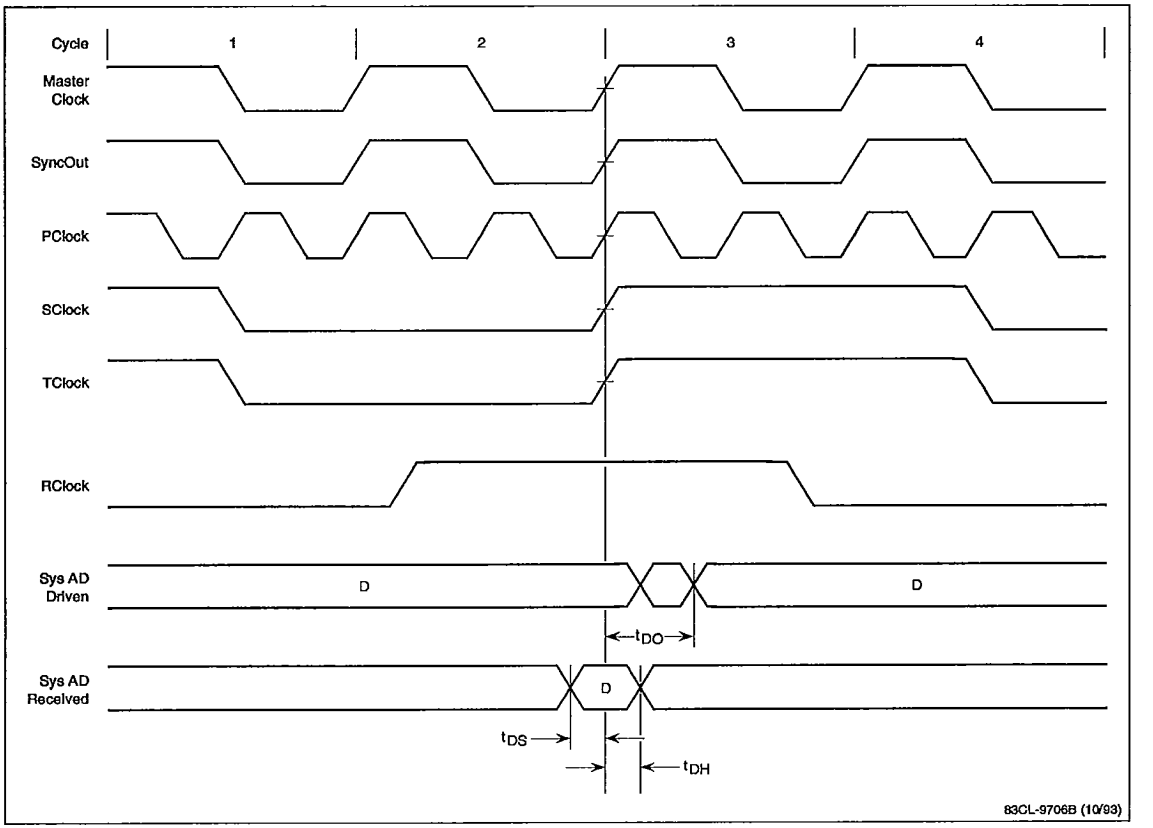
Timing Diagrams (cont)

Processor Clock; PClock to SClock, Divisor of 4



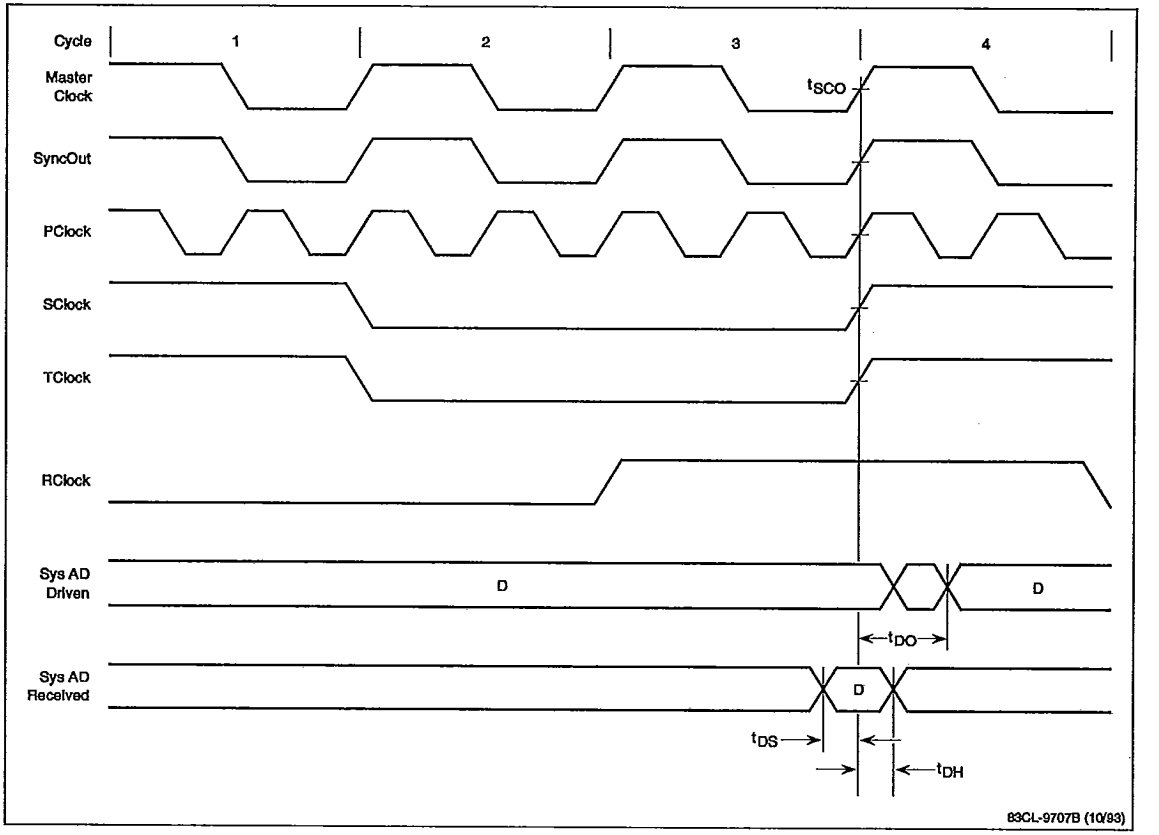
Timing Diagrams (cont)

Processor Clock; PClock to SClock, Divisor of 6



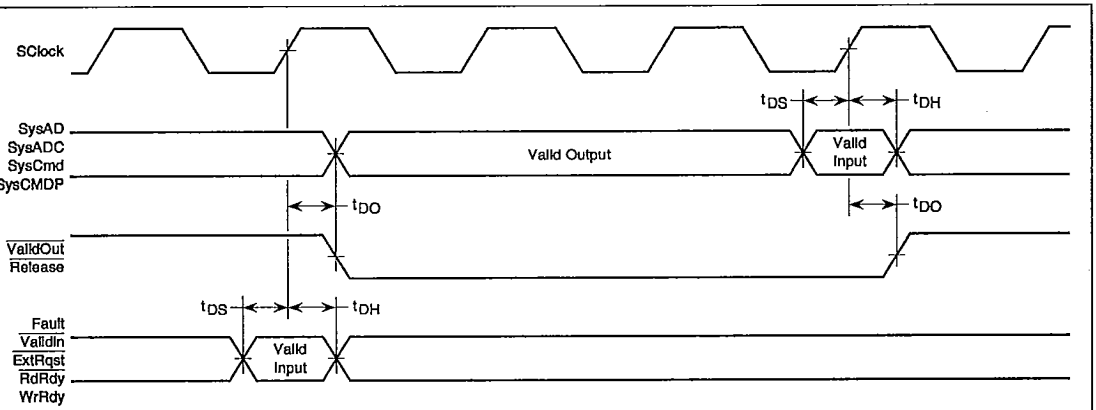
Timing Diagrams (cont)

Processor Clock; PClock to SClock, Divisor of 8



Timing Diagrams (cont)

System Interface Edge Timing Relationships

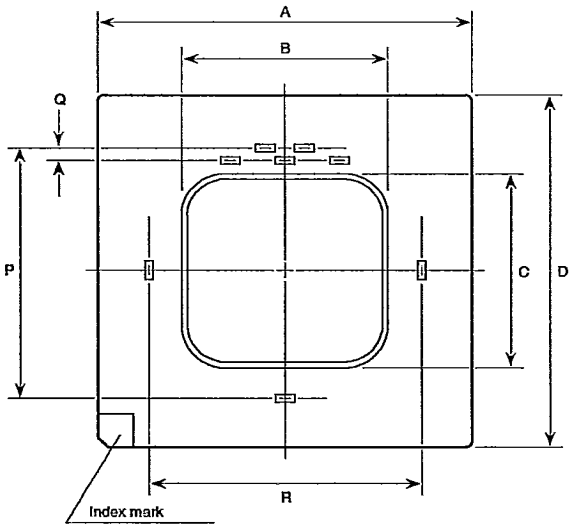


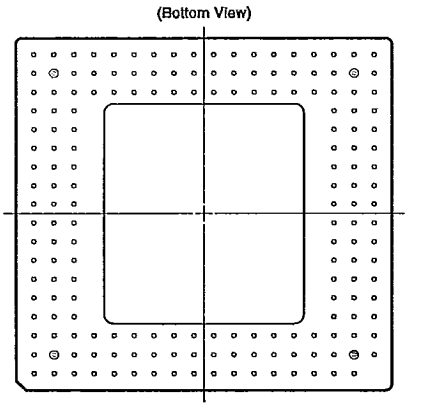
Note: These waveforms describe edge-to-edge timing relationships only.

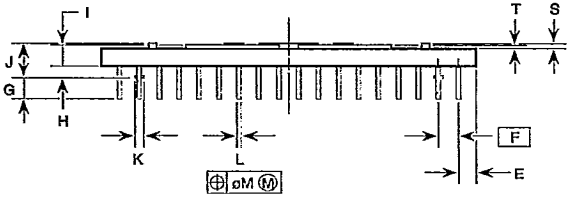
83RC-9579B (8/93)

PACKAGE DRAWINGS

179-Pin Ceramic PGA (Metal Sealed)







Item	Millimeters	Inches
A	47.24 ± 0.47	1.860 ± .019
B	26.92	1.060
C	29.46	1.160
D	47.24 ± 0.47	1.860 ± .019
E	2.03	.080
F	2.54 (T.P.)	.100 (T.P.)
G	3.3 ± 0.2	.130 ± .008
H	0.635 min	.025 min
I	3.33	.131
J	5.60 max	.221 max
K	ø1.27 ± 0.2	ø.050 ± .008
L	ø0.46 ± 0.05	ø.018 ± .002
M	0.254	.010
P	36.46	1.435
Q	1.60 min	.063 min
R	33.92	1.335
S	1.54 max	.061 max
T	1.08 max	.043 max

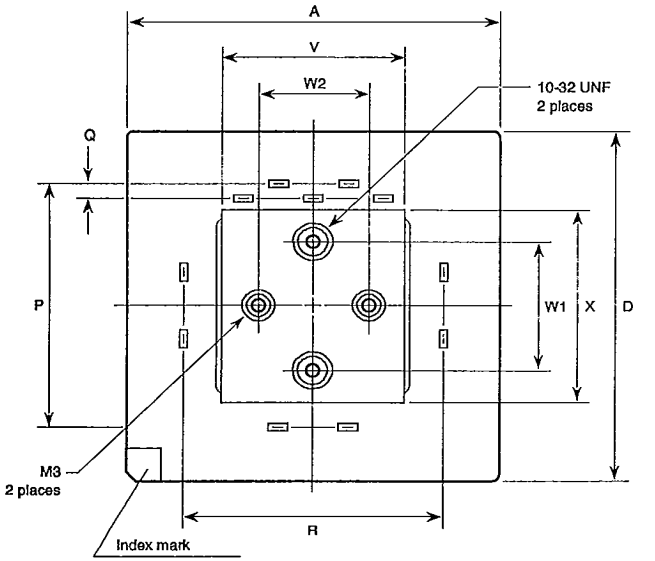
Note
Each lead centerline is located within ø0.254 mm (ø.010 inch) of its true position (T.P.) at maximum material condition.

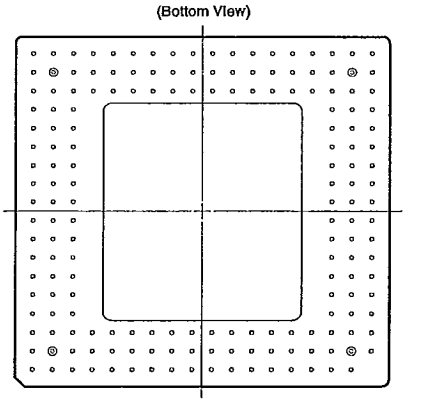
X179RJ-100A

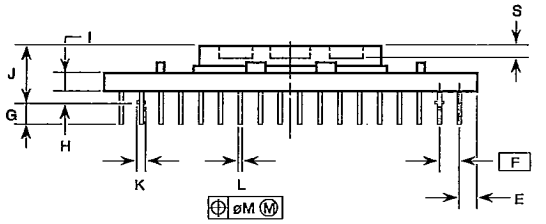
83RD-9665B (10/93)

PACKAGE DRAWINGS

179-Pin Ceramic PGA (With Heat Sink Adapter Plate)







Item	Millimeters	Inches
A	47.24 ± 0.47	1.860 ± .019
D	47.24 ± 0.47	1.860 ± .019
E	2.03	.080
F	2.54 (T.P.)	.100 (T.P.)
G	3.3 ± 0.2	.130 ± .008
H	0.635 min	.025 min
I	3.33	.131
J	11.69 max	.461 max
K	ø1.27 ± 0.2	ø.050 ± .008
L	ø0.46 ± 0.05	ø.018 ± .002
M	0.254	.010
P	36.46	1.435
Q	1.60 min	.063 min
R	33.92	1.335
S	1.54 max	.061 max
V	25.0	.984
W1	21.59	.850
W2	15.0	.591
X	29.5	1.161

Note
Each lead centerline is located within ø0.254 mm (ø.010 inch) of its true position (T.P.) at maximum material condition.

83CL9730B (11/93)