NEC
NEC Electronics Inc.

VR4400SC (μPD30411) 64-Bit RISC Microprocessor: Primary/Secondary Cache Memory

Preliminary Information November 1993

Description

The VR4400SC™ 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 VR4400SC below.

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

System applications range from inexpensive highly integrated desktop systems through large multiprocessor servers whose CPU performance rivals current mainframes and whose address space requirements are not by current generation microprocessors.

The VR4400SC 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 like transaction processing with fault tolerant support.

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 VR4400SC machine.

The 64-bit addressing capability of the VR4400SC 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.

The other enhancements of the VR4400SC are (1) fully functional status pins and (2) 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 internal clock)
- 3.3- or 5-volt power supply
- On-chip 16-Kbyte instruction cache, 16-Kbyte data cache, and 128-bit secondary cache interface support
- 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
- Dynamically configurable big-endian or little-endian byte ordering

50630-1

VR4400SC (μPD30411)

NEC

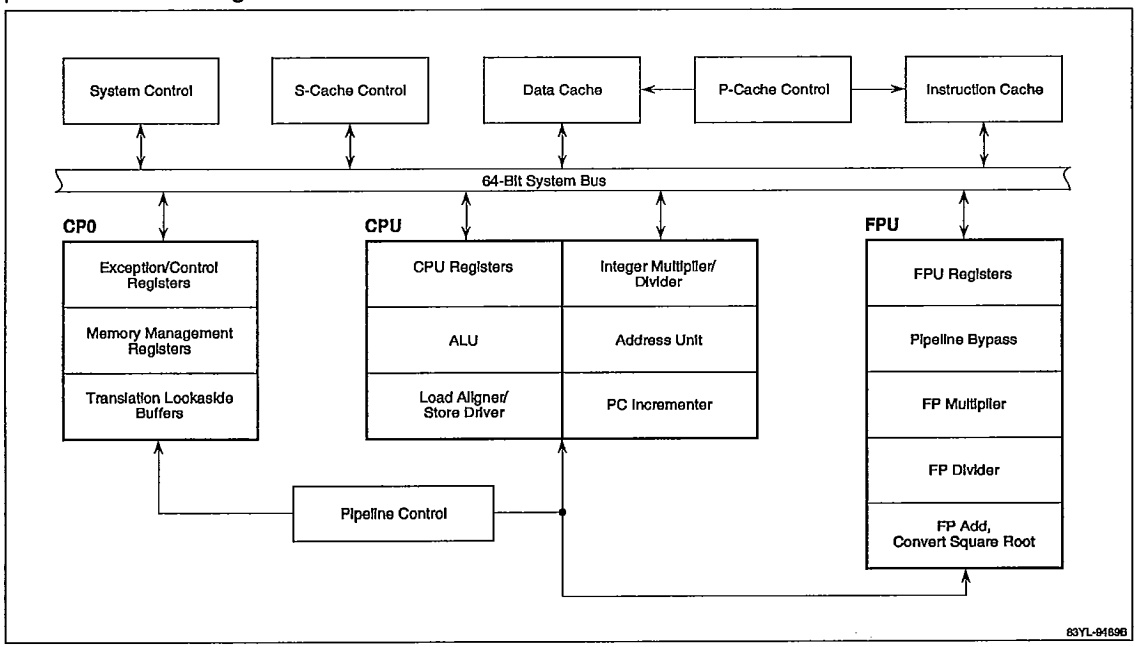
- Timing flexibility for 128-bit secondary cache interface and 64-bit system interface to allow speed matching of logic and memory components
- System interface clock modes: divide-by-2, -3, -4, -6, and -8
- Master/checker mode
- One-level deep uncached store buffers

VR4400PC, VR4400SC, and VR4400MC are trademarks of NEC Corporation.

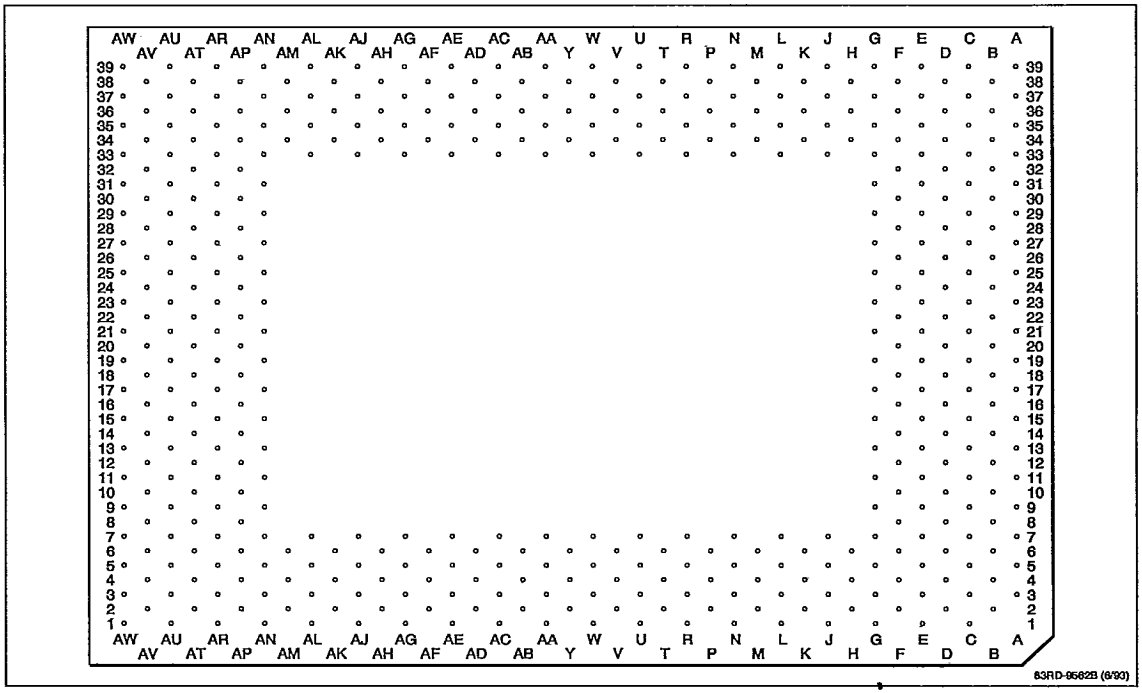
Ordering Information

Part Number	PClock	Power Supply	Package
μPD30411RJ-50	100 MHz	5 V	447-pin ceramic PGA
-67	133 MHz		
-75	150 MHz		
μPD30411RP-50	100 MHz	5 V	447-pin ceramic PGA with heat-sink adapter plate
-67	133 MHz		
-75	150 MHz		
μPD30411LRJ-50	100 MHz	3.3 V	447-pin ceramic PGA
-67	133 MHz		
-75	150 MHz		
μPD30411LRP-50	100 MHz	3.3 V	447-pin ceramic PGA with heat-sink adapter plate
-67	133 MHz		
-75	150 MHz		

μPD30411 Block Diagram



Package Pin Configuration (Bottom View); 447-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	T2	SysAD40	A23	SysCmd0	G1	$\overline{\text{ColdReset}}$	AW37
1	M2	41	A27	1	E3	$\overline{\text{ExtRqst}}$	AV2
2	J3	42	A31	2	B2	GndP	Y34
3	G3	43	A35	3	B12	GndSense	U37
4	C1	44	C37	4	B16	$\overline{\text{Int0}}$	AL1
5	A3	45	E39	5	B20	IOIn	AV32
6	A9	46	H38	6	B24	IOOut	AV28
7	A13	47	M38	7	B28	MasterClock	AA37
8	A21	48	AE1	8	B32	MasterOut	AJ39
9	A25	49	AJ1			ModeClock	B8
SysAD10	A29	SysAD50	AM2	SysCmdP	A37	ModeIn	AV8
11	A33	51	AR1			$\overline{\text{NMI}}$	AV16
12	B38	52	AU3			RClock0	AM34
13	E37	53	AW5			RClock1	AL33
14	G39	54	AW11			$\overline{\text{RdRdy}}$	AW7
15	L39	55	AW15			$\overline{\text{Release}}$	AV12
16	AD2	56	AW23			$\overline{\text{Reset}}$	AU39
17	AH2	57	AW27			SyncIn	W39
18	AL3	58	AW31			SyncOut	AN39
19	AN3	59	AW35			TClock0	H34
SysAD20	AU1	SysAD60	AU37			TClock1	J33
21	AW3	61	AR39			$\overline{\text{ValidIn}}$	AN1
22	AW9	62	AL39			$\overline{\text{ValidOut}}$	AR3
23	AW13	63	AG39			V_{DD}Ok	AE39
24	AW21					V_{DD}P	AA33
25	AW25					V_{DD}Sense	W33
26	AW29					$\overline{\text{WrRdy}}$	A7
27	AW33						
28	AV38					$\overline{\text{Fault}}$	C39
29	AR37						
SysAD30	AM38	SysADC0	A17	**JTAG**		Status0	U33
31	AH38	1	R39	JTCK	U39	Status1	U35
32	RI	2	AW17	JTDI	N39	Status2	V36
33	LI	3	AD38	JTDO	J39	Status3	W35
34	H2	4	A19	JTMS	G37	Status4	W37
35	EI	5	T38			Status5	AC37
36	C3	6	AW19			Status6	AC35
37	A5	7	AC39			Status7	AC33
38	A11						
39	A15						

Pin Assignments; Secondary Cache Data

Secondary Cache Data	Pin No.	Secondary Cache Data	Pin No.	Secondary Cache Data	Pin No.	Secondary Cache Data	Pin No.
SCData0	R3	SCData40	C23	SCData80	AC7	SCData120	AR21
1	R7	41	F24	81	AE5	121	AP24
2	L5	42	E27	82	AG7	122	AU27
3	F8	43	D30	83	AR5	123	AT30
4	C9	44	C33	84	AR9	124	AU33
5	F12	45	E35	85	AR11	125	AN33
6	G15	46	L35	86	AN15	126	AL37
7	E17	47	R33	87	AP16	127	AG33
8	G21	48	AF4	88	AU21		
9	C25	49	AJ3	89	AN23		
SCData10	G25	SCData50	AJ7	SCData90	AR25	SCDChk0	G19
11	E29	51	AP8	91	AP28	1	T34
12	G31	52	AT10	92	AU31	2	AP20
13	C35	53	AR13	93	AR33	3	AD34
14	K36	54	AR15	94	AL35	4	C19
15	N35	55	AT18	95	AH34	5	R37
16	AE3	56	AU23	96	U7	6	AU19
17	AG5	57	AT26	97	N3	7	AE37
18	AK4	58	AR27	98	N7	8	C17
19	AN9	59	AN29	99	C5	9	N37
SCData20	AU9	SCData60	AP32	SCData100	E9	SCDChk10	AU17
21	AN13	61	AN35	101	C11	11	AG37
22	AT14	62	AJ35	102	C13	12	E19
23	AR17	63	AE33	103	F16	13	R35
24	AT22	64	V4	104	E21	14	AR19
25	AU25	65	R5	105	G23	15	AE35
26	AN27	66	N5	106	C27		
27	AR29	67	E5	107	F28		
28	AN31	68	G9	108	E31		
29	AR35	69	E11	109	G33		
SCData30	AK36	SCData70	G13	SCData110	J37		
31	AG35	71	D14	111	N33		
32	T6	72	C21	112	AD6		
33	L3	73	D22	113	AG3		
34	L7	74	E25	114	AJ5		
35	E7	75	G27	115	AU5		
36	G11	76	C31	116	AN11		
37	E13	77	F32	117	AU11		
38	E15	78	J35	118	AU13		
39	G17	79	M34	119	AN17		

Pin Assignments; Secondary Cache Address, Tag, and Control

Secondary Cache Data	Pin No.	Secondary Cache Tag	Pin No.	Secondary Cache Control	Pin No.
SCAdd1	AL5	SCTag0	K4	$\overline{\text{SCDCS}}$	M6
2	AG1	1	G7	$\overline{\text{SCOE}}$	N1
3	AE7	2	C7	$\overline{\text{SCTCS}}$	J1
4	AC1	3	D10		
5	AC5	4	C15	$\overline{\text{SCWrW}}$	J5
6	AC3	5	D18	$\overline{\text{SCWrX}}$	J7
7	AA1	6	F20	$\overline{\text{SCWrY}}$	H6
8	AB4	7	E23	$\overline{\text{SCWrZ}}$	G5
9	AA5	8	D26		
10	AA7	9	C29		
SCAddr11	AA3	SCTag10	G29		
12	W3	11	E33		
13	Y6	12	G35		
14	W5	13	L33		
15	W7	14	L37		
16	W1	15	P36		
17	U3	16	AF36		
		17	AJ37		
		18	AJ33		
		19	AN37		
SCAddr0W	AN7	SCTag20	AU35		
X	AN5	21	AR31		
Y	AM6	22	AU29		
Z	AL7	23	AN25		
		24	AR23		
SCAPar0	U5	SCTChk0	AN21		
1	U1	1	AN19		
2	P4	2	AU15		
		3	AP12		
		4	AU7		
		5	AR7		
		6	AH6		

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

V_{DD}	GND	NC
A39	B4, B14, B22, B30, B36	
B6, B10, B18, B26, B34	D2, D6, D12, D20, D28, D34, D38	Y2
D4, D8, D16, D24, D32, D36	F4, F6, F10, F18, F26, F34, F36	AV24
F2, F14, F22, F30, F38	K2, K34	AA35, AA39
H4, H36	M4, M36	
K6, K38	P6, P38	
P2, P34	V2, V34	
T4, T36	Y4, Y36	
V6, V38; Y38	AB6, AB36, AB38; AF2, AF34	
AB2, AB34; AD4, AD36; AF6, AF38	AH4, AH36	
AK2, AK34; AM4, AM36	AK6, AK38	
AP2, AP10, AP18, AP26, AP38	AP4, AP6, AP14, AP22, AP30, AP34, AP36	
AT4, AT8, AT16, AT24, AT32. AT36	AT2, AT6, AT12, AT20, AT28, AT34, AT38	
AV6, AV14, AV22, AV30, AV34; AW1, AW39	AV4, AV10, AV18, AV26, AV36	

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 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 processor read, invalidate, or update requests in both secondary cache and non-secondary cache mode; or it can accept a read followed by a write request, a read followed by a potential update request, or a read followed by a potential update followed by a write request in secondary cache mode.
	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.
Clock/Control	$\overline{\text{Fault}}$	Output	Fault is asserted by the processor to indicate a mismatch output of boundary comparators and an indication of system interface input parity or ECC errors.
	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.
	Status(7:0)	Output	Status7 through Status0 indicate the current operational status of the processor.

Pin Descriptions (cont)

Interface Signals	Symbol	Input/Output	Description
Clock/Control (cont)	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).
	V_{DD}Sense	Input/Output	V_{DD} Sense is a special pin used only in component testing and characterization. It provides a separate, direct connection from the on-chip V_{DD} node to the package pin without connecting to the in-package power planes. Testing fixtures treat V_{DD}Sense as an analog output pin; the voltage at this pin directly exhibits the behavior of the on-chip V_{DD}. Thus, characterization engineers can easily observe the effects of di/dt noise, transmission line reflections, etc. V_{DD}Sense should be connected to V_{DD} in functional system designs
	GndP	Input	Quiet Ground is directed to the internal phase-locked loop.
	GndSense	Input/Output	Ground Sense provides a separate, direct connection from the on-chip ground node to a package pin without having to connect to the in-package ground planes. GndSense should be connected to Gnd in functional system designs.
Secondary Cache	SCAddr (17:1), SCAddr0 (W:Z)	Output	Secondary Cache Address Bus is an 18-bit address bus for the secondary cache. The least significant bit (bit 0) has four output lines, SCAddr0 (W, X, Y, Z), to provide additional drive current.
	SCAPar(2:0)	Output	Secondary Cache Address Parity Bus is a 3-bit bus that carries the parity of the SCAddr bus and the cache control line $\overline{SCWr}$. Below are the individual bit definitions. Even parity for SCAddr(17:12) and $\overline{SCWr}$. Even parity for SCAddr(11:6) and $\overline{SCDCS}$. Even parity for SCAddr(5:0) and $\overline{SCTCS}$.
	SCData(127:0)	Input/Output	Secondary Cache Data Bus is a 128-bit bus used to read and write cache data from and to the secondary cache data RAM.
	SCDChk(15:0)	Input/Output	Secondary Cache Data ECC Bus is a 16-bit bus carrying two 8-bit ECC fields that cover the 128 bits of SCData from/to secondary cache. SCDChk(15:8) corresponds to SCData(127:64). SCDChk(7:0) corresponds to SCData(63:0).
	$\overline{SCDCS}$	Output	Secondary Cache Data Chip Select is a signal for the secondary cache data RAM.
	$\overline{SCOE}$	Output	Secondary Cache Output Enable is a signal for the secondary cache data and tag RAM.
	SCTag(24:0)	Input/Output	Secondary Cache Tag Bus is a 25-bit bus used to read or write cache tags from and to the secondary cache.
	SCTChk(6:0)	Input/Output	Secondary Cache Tag ECC Bus is a 7-bit bus carrying an ECC field covering the SCTag from and to the secondary cache.
	$\overline{SCTCS}$	Output	Secondary Cache Tag Chip Select is a signal for the secondary cache tag RAM.
	$\overline{SCWrW}$, $\overline{SCWrX}$, $\overline{SCWrY}$, $\overline{SCWrZ}$	Output	Secondary Cache Write Enables are signals for the secondary cache RAM.

Pin Descriptions (cont)

Interface Signals	Symbol	Input/Output	Description
Interrupt	$\overline{Int0}$	Input	Interrupt is the general processor interrupt, bitwise ORed with bit 0 of the interrupt register.
	$\overline{NMI}$	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	$\overline{ColdReset}$	Input	$\overline{ColdReset}$ 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 $\overline{ColdReset}$. It must be deasserted synchronously with MasterOut.
	ModeClock	Output	Boot-Time 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	Boot-Time Mode Data In is a serial boot-time mode data input.
	$\overline{Reset}$	Input	$\overline{Reset}$ 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. $\overline{Reset}$ 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 VR4400SC 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

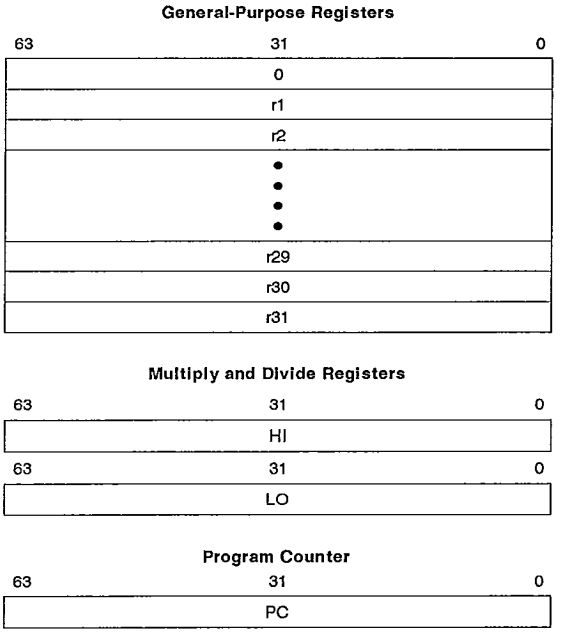
The two Multiply and Divide registers (HI, LO) store the doubleword 64-bit result or the quadword 128-bit result of integer multiply operations and the quotient (in LO) and remainder (in HI) of integer divide operations.

The VR4400SC 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

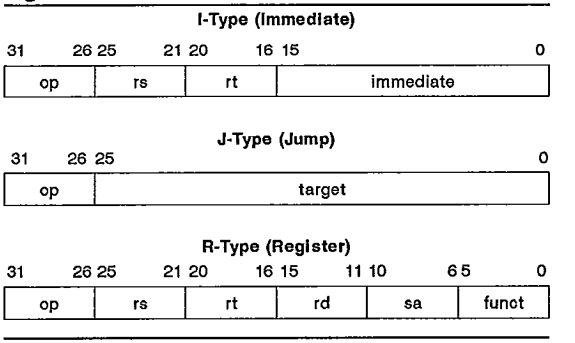
Each CPU instruction is 32 bits long. Figure 2 shows the three instruction formats: I-type (immediate), J-type (jump), and R-type (register). Using only these three instruction formats simplifies instruction decoding; more complicated (and less frequently used) operations and addressing modes can be 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 VR-Series processors. Table 3 lists VR4400SC 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 VR4400SC 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 VR4400SC 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.

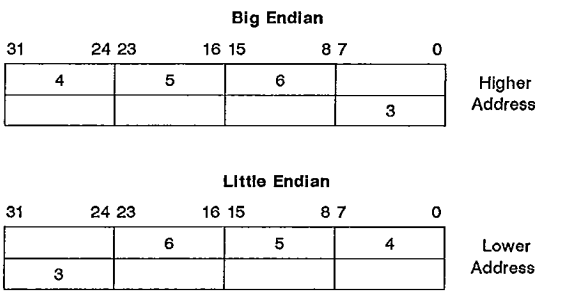
The CPU uses byte addressing for halfword, word, and doubleword accesses with the following alignment constraints.

- 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.

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 VR4400SC 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.

Figure 8. System Control Coprocessor CP0 Registers

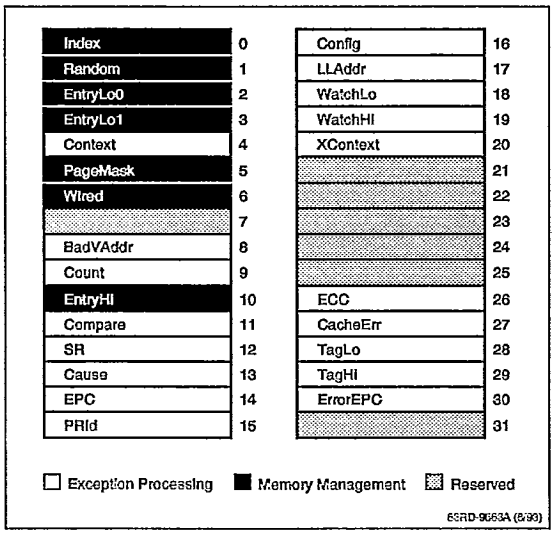


Table 4. System Control Coprocessor CP0 Registers

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

Table 4. System Control Coprocessor CP0 Registers (cont)

No.	Register	Description
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 VR4400SC microprocessor supports a cache memory hierarchy that increases memory access

bandwidth and reduces the latency of load and store instructions. The two-level cache memory hierarchy consists of on-chip instruction and data caches and an optional external secondary cache that can vary in size from 128 Kbytes to 4 Mbytes.

The secondary cache is assumed to be one bank of industry-standard static RAM (SRAM) with output enables. The secondary cache consists of a quadword (128-bit) wide data array and a 25-bit wide tag array. Check fields are added to both the data and tag arrays to improve data integrity. The secondary cache may be configured as either a joint cache or split instruction/data cache. The maximum secondary cache size is 4 Mbytes and the minimum is 128 Kbytes for a joint cache and 256 Kbytes for split instruction/data cache. The secondary cache is direct-mapped and is addressed with the lower part of the physical address.

On-Chip Caches. The VR4400SC incorporates on-chip 16-Kbyte instruction and 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.

Secondary Cache Interface. The VR4400SC provides all of the secondary cache control circuitry, including ECC protection, on chip. The secondary cache interface consists of a 128-bit data bus, a 25-bit tag bus, an 18-bit address bus, and SRAM control signals. The 128-bit wide data bus minimizes cache miss penalty, and allows the use of standard low-cost SRAMs in the secondary cache design.

Memory Management System

The VR4400SC 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 VR4400SC 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 VR4400SC 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 VR4400SC microprocesor exploits instruction-level parallelism using a superpipelined implementation. The VR4400SC 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 VR4400SC 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.

Instruction Execution. The execution of a single VR4400SC 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 addres 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 VR4400SC 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 VR4400SC 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 VR4400SC, 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 VR4400SC 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 VR4400SC 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.

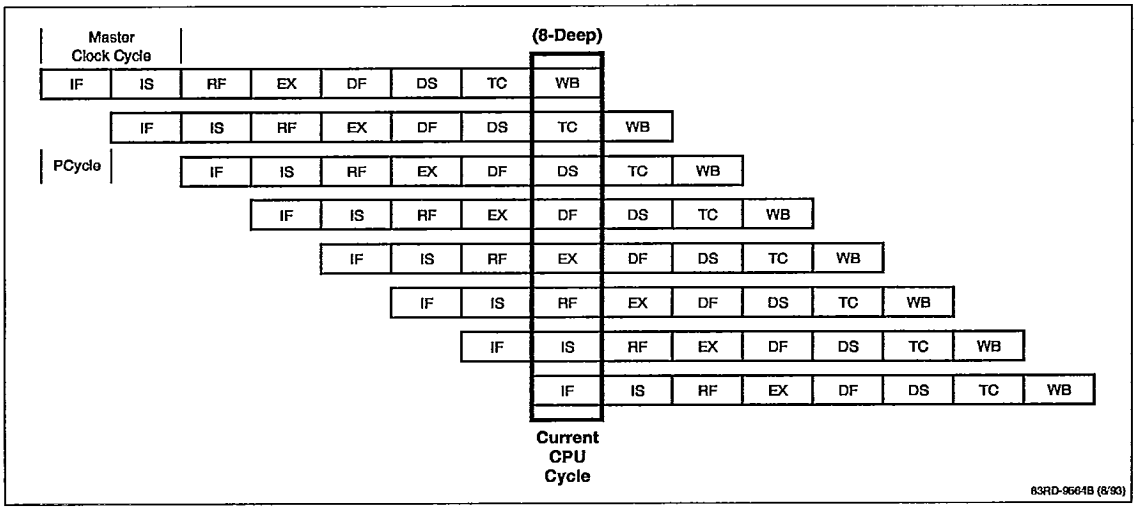
When an exception occurs, the CPU loads the Exception Program Counter (EPC) with a restart location where execution may resume after the exception has been serviced. The restart location in the EPC is the address of the instruction that caused the exception or, if the instruction was executing in a branch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

A new bit, EW, is added in the VR4400SC 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 VR4400SC supports a 64-bit system interface that can be used to construct uniprocessor systems with a direct DRAM interface and a secondary cache 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.

Figure 9. Eight-Stage Pipeline



Processor Interrupts

The VR4400SC supports one hardware interrupt, two software interrupts, and a nonmaskable interrupt. The processor's hardware interrupt is 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 VR4400SC, 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.

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 VR4400SC 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 VR4400SC 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 VR4400SC implements the MIPS Instruction Set Architecture (ISA) for user-mode programs; guaranteeing that the programs will execute on any MIPS hardware implementation.

ERROR CHECKING AND CORRECTING

The error checking and correcting (ECC) code of VR4400SC detects and sometimes corrects errors caused by system noise, power surges, and alpha particles during data movements. The processor does two types of error checking: parity error detection and single-bit error correction/double-bit error detection (SECDED).

Parity Error Detection

Parity is the simplest error detection scheme. By appending a parity bit to the end of an item of data, single-bit errors can be detected —but not corrected.

There are 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. To resolve this problem, SECDED ECC was developed.

Secondary Cache Protection

The ECC code chosen for secondary cache data bus and tag bus protection is single-bit error correction and double-bit error detection (SECDED) code. The SECDED ECC code is an improvement upon the parity scheme; not only does it detect single-bit and certain multibit errors, it corrects single-bit errors.

Secondary Cache Data Bus. The SECDED ECC code protecting the secondary cache data bus has the properties listed below.

- Corrects single-bit errors.
- Detects double-bit errors.
- Detects 3- or 4-bit errors within a nibble.
- Provides 64 data bits protected by 8 check bits, and yields 8-bit syndromes. (The syndrome is a generated value used to detect an error and locate the position of the single bit in error.)
- It is a minimal-length code; each parity tree used to generate the 8-bit syndrome has only 27 inputs, the minimum number possible.
- Provides byte exclusive-ORs (XORs) of the data bits as part of the XOR trees used to build the parity generators. This allows selection of byte parity out of the XOR trees that generate or check the code.
- Single-bit errors are indicated either by syndromes that contain exactly three 1s, or syndromes that contain exactly five 1s in which bits 0-3 or bits 4-7 of the syndrome are all 1s.
- Double-bit errors are indicated by syndromes that contain an even number of 1s.
- 3-bit errors within a nibble are indicated by syndromes that contain five 1s in which bits 0-3 and bits 4-7 of the syndrome are not all 1s.
- 4-bit errors within a nibble are indicated by syndromes that contain four 1s. Because this is an even number of 1s, 4-bit errors within a nibble look like double-bit errors.

Secondary Cache Tag Bus. The SECDED ECC code protecting the secondary cache tag bus has the following properties.

- Corrects single-bit errors.
- Detects double-bit errors.
- Detects 3- or 4-bit errors within a nibble.
- Provides 25 data bits protected by 7 check bits, yielding 7-bit syndromes.
- Provides byte XORs of the data bits as part of the XOR trees used to build the parity generators. This allows selection of byte parity out of the XOR trees that generate or check the code.
- Single-bit errors are indicated by syndromes that contain exactly three 1s. This makes it possible to decode the syndrome to find which data bit is in error with three-input NAND gates. For the check bits, a full 7-bit decode of the syndrome is required.
- Double-bit errors are indicated by syndromes that contain an even number of 1s.
- 3-bit errors within a nibble are indicated by syndromes that contain either five 1s or seven 1s.
- 4-bit errors within a nibble are indicated by syndromes that contain either four 1s or six 1s. Because these are even numbers of 1s, 4-bit errors within a nibble look like double-bit errors.

Error Checking Operation

The processor verifies data correctness by using either the parity or the SECDED code as it passes data from the system interface to the secondary cache, or as it moves data from the secondary cache to the primary caches or to 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 correct check bits for doubleword, word, or partial-word data transmitted to the system interface. As it checks for data

correctness, the processor passes data check bits from the secondary caches, directly without changing the bits, to the system interface. However, the processor does not check data received from the system interface for external updates and 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. Software, in conjunction with an off-processor data corrector, is responsible for correcting the data when SECDED code is employed.

System Interface Command Bus. In the VR4400SC 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. The input parity is reported through the Fault pin.

Secondary Cache Data Bus. The 16 check bits, SCDChk(15:0), for the 128-bit secondary cache data bus are organized as 8 check bits for the upper 64 bits of data and 8 check bits for the lower 64 bits of data.

System Interface and Secondary Cache Data Bus. The 8 check bits, SysADC(7:0), for the system address and data bus provide even-byte parity or are generated in accordance with a SECDED code that also detects any 3- or 4-bit error in a nibble. The 8 check bits for each half of the secondary cache data bus are always generated in accordance with the SECDED code.

Secondary Cache Tag Bus. The 7 check bits, SCTChk(6:0), for the secondary cache tag bus are generated in accordance with the SECDED code, which also detects any 3- or 4-bit error in a nibble.

The processor generates check bits for the tag when it is written into the secondary cache and checks the tag whenever the secondary cache is accessed.

The processor contains a corrector for the secondary cache tag; the tag corrector is not in-line for processor accesses due to primary cache misses. The processor traps when a tag error is detected on a processor access due to a primary cache miss. Software, using the processor cache management primitives, is responsible for correcting the tag. When executing the cache management primitives, the processor uses the corrected tag to generate write back addresses and cache state.

For external accesses, the tag corrector is in-line; that is, the response to external accesses is based on the corrected tag. The processor still traps on tag errors detected during external accesses to allow software to repair the contents of the cache if possible.

BASIC FUNCTIONS

The new speed-doubler feature has been added to the VR4400SC microprocessor to increase performance. The VR4400SC 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 VR4400SC 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 VR4400SC 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.

The VR4400SC supports the secondary cache interface by driving a 128-bit data bus with 16 ECC (Error Checking and Correcting) bits, 25-bit tag access bus with 7 ECC bits, 17-bit address bus with 4 duplicated address 0 bits, and 3-bit even parity bus. In addition, there are 7 secondary cache interface bits for controlling the flow of secondary cache accessing.

SYSTEM INTERFACE

A system event is an event 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 both the primary and secondary caches.
- A store that misses in both the primary and secondary caches.
- An uncached load or store.

On load or store miss in both caches, 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, write, and null write requests.

Processor Read Requests. When a processor issues a read request, the external agent must access the specified resource and return the requested data.

A processor read request can be split from the external agent's return of the requested data; in other words, the external agent can initiate an unrelated external request before it returns the response data for a processor read. A processor read request is completed after the last word of response data has been received from the external agent.

Note that the data identifier associated with the response data can signal that the returned data is erroneous, causing the processor to take a bus error.

Processor read requests that have been issued, but for which data has not yet been returned, are said to be pending. A read remains pending until the requested read data is returned.

The external agent must be capable of accepting a processor read request when these two conditions are met:

- There is no processor read request pending.
- The signal $\overline{\text{RdRdy}}$ has been asserted for two or more cycles.

Processor Write Requests. When a processor issues a write request, the specified resource is accessed and the data is written to it. (Processor write requests are described here; external write requests are described later.)

A processor write request is complete after the last word of data has been transmitted to the external agent.

The external agent must be capable of accepting a processor write request when these two conditions are met:

- There is no processor read request pending.
- The signal $\overline{\text{WrRdy}}$ has been asserted for two or more cycles.

Processor Null Write Requests. The processor null write request indicates that an expected write has been obviated as a result of some external request. Since the processor accepts external requests between the issue of a read-with-write-forthcoming request that begins a cluster and the issue of the write request that completes a cluster, it is possible for an external request to eliminate the need for the write request in the cluster.

For example, if the external agent issued an external invalidate request that targeted the cache line the processor was attempting to write back, the state of the cache line would be changed to invalid and the write back for the cache line would no longer be needed. In this event, the processor issues a processor null write request after completing the external request to complete the cluster.

Processor null write requests do not obey the $\overline{\text{WrRdy}}$ flow control rules for issuance; rather they issue with a single address cycle regardless of the state of $\overline{\text{WrRdy}}$. Any external request that changes the state of a cache line from dirty exclusive or dirty shared to clean exclusive, shared, or invalid obviates the need for a write back of that cache line.

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.

External Read Request. In contrast to a processor read request, data is returned directly in response to an external read request; no other requests can be issued until the processor returns the requested data. An external read request is complete after the processor returns the requested word of data.

The data identifier associated with the response data can signal that the returned data is erroneous, causing the processor to take a bus error.

Note: The processor does not contain any resources that are readable by an external read request; in response to an external read request, the processor returns undefined data and a data identifier with its Erroneous Data bit (SysCmd5) set.

External Write Request. When an external agent issues a write request, the specified resource is accessed and the data is written to it. An external write request is complete after the word of data has been transmitted to the processor.

The only processor resource available to an external write request is the Interrupt register.

External Null Request. An external null request requires no action by the processor. It simply provides a mechanism for an external agent to either return control of the secondary cache to the VR4400SC, or to return the system interface to the master state without affecting the processor.

Read Response Request

Technically, the read 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.

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 ExtRqst and Release. An external agent must acquire mastership of the system interface before submitting to an external request. The external agent submits a request by asserting ExtRqst and waiting for the processor to assert Release for one cycle. The processor will not assert 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 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 RdRDY and WrRdy to allow an external agent to manage the flow of processor requests. RdRdy controls the flow of processor read, invalidate, and update requests; WrRdy controls the flow of processor write requests. Processor null write requests must always be accepted, since they cannot be delayed by either RdRdy or WrRdy.

The processor samples RdRdy to determine the issue cycle for a processor read, invalidate, or update request defined to be the first address cycle for the request that asserted RdRdy two cycles previously. And the processor samples 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 WrRdy two cycles previously. If the processor issues a read or write request when neither RdRdy nor 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

Secondary Cache Mode

A processor in the large configuration package may be programmed to run either secondary cache mode or nonsecondary cache mode. In this mode, the processor issues the requests individually as in nonsecondary mode in groups. These request groups, which lead with processor read request, are called Clusters.

Cluster. A cluster consists of a processor read request followed by one or two additional processor requests issued while the read request is pending. All requests must be finished before data is returned in response to the leading read request. The cluster can be read with forthcoming write and write.

The external agent must accept all requests in the cluster before returning data in response to the leading read request. If not, the behavior of the processor is undefined.

Read With Forthcoming Write Request. The processor issues a read with forthcoming write request instead of an ordinary read request for the cluster containing the processor write request. It is identified by a bit in the command for the processor read request. The write request in the cluster must obey the WrRdy flow control rule.

Null Write Request. Since the processor accepts external requests between read with forthcoming write request and write request in the cluster, the processor might issue the null write request instead of write request to terminate the cluster. For instance, if the external agent generates an external invalidate request to invalidate a cache line that the processor attempts to write back, the data in this cache line no longer needs to be written back after being invalidated by the external agent. Consequently, any external request that changes the state of a cache line from dirty to clean or invalidate obviates the need for a write back of that cache line. The null write request does not obey the $\overline{\text{WrRdy}}$ flow control rule for issue. It will be issued regardless of the state of $\overline{\text{WrRdy}}$.

In the secondary cache mode, an external agent must be capable of accepting a cluster any time that:

- No processor request is pending.
- $\overline{\text{RdRdy}}$ has been asserted for at least two cycles

Also, it must be capable of accepting a processor write request any time that:

- Read with forthcoming write request is pending or no processor request is pending.
- $\overline{\text{WrRdy}}$ has been asserted for at least two cycles.

After issuing a processor read request, the processor does not issue a subsequent read request until it has received a response request for the read request, whether this read request began a cluster or not.

After issuing a processor invalidate or update request, or after being no longer potential for an invalidate or update request, the processor does not issue a subsequent request until the invalidate or update request has been acknowledged.

After issuing a write request, the processor does not issue a subsequent request until at least four cycles after the issue cycle of the write request.

Nonsecondary 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 VR4400SC microprocessor generates a request or a series of requests through the system interface to satisfy system events. Processor requests are managed in two distinct modes: secondary cache mode and nonsecondary cache mode. The following sections detail the sequence of requests generated by the processor for each system event in these two cache modes.

Load Miss

On the load miss in both primary and secondary cache cycles, the processor must obtain the cache line containing the data element to be loaded from the external agent before it can proceed.

If a current dirty exclusive cache line will be replaced by the new cache line, the current cache line must be written back before the new line can be loaded into the primary and secondary caches.

The processor examines the coherency attribute in the TLB entry for the page containing the requested cache line and executes one of the following.

(1) If the coherency attribute is exclusive, the processor issues a coherent read request that also requests exclusivity.

(2) If the coherency attribute is noncoherent, the processor issues a noncoherent read request.

Secondary Cache Mode. If the current cache line does not have to be written back and the coherency attribute for the page containing the requested cache line is not exclusive, the processor issues a coherent block read request for the cache line containing the data element to be loaded.

If the current cache line needs to be written back and the coherency attribute for the page containing the requested cache line is exclusive, the processor issues a cluster consisting of an exclusive read with forthcoming write request, followed by a write request for the current cache line.

Nonsecondary Cache Mode. If the cache line must be written back on a load miss, the read request is issued and completed before the write request is handled. The processor takes the following steps.

(1) The processor issues a noncoherent read request for the cache line containing the data element to be loaded. (Only noncoherent and uncached attributes are supported in the nonsecondary cache mode.)

(2) The processor then waits for an external agent to provide the read response.

If the current cache line must be written back, the processor issues a write request to save the dirty cache line in memory.

Store Miss

On the store miss in both primary and secondary cache cycles, the processor must obtain from the external agent the cache line containing the store target location. 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 block read request is issued.

Secondary Cache Mode. If the new cache line replaces a current cache line in the dirty exclusive state, the current cache line must be written back before the new line can be loaded in the primary and secondary caches. The processor requests issued are a function of the page attributes listed below.

(1) Noncoherent Page Attribute. If the current cache line must be written back and the coherency attribute for the requested cache line is noncoherent, the processor issues a cluster consisting of a noncoherent block read with forthcoming write request for the cache line containing the store target location, followed by a block write request for the current cache line.

(2) If the current cache line does not need to be written back and the coherency attribute for the page that contains the requested cache line is noncoherent, the processor issues a noncoherent block read request for the cache line containing the store target location.

(3) Exclusive Page Attribute. If the current cache line must be written back and the coherency attribute for the page that contains the requested cache line is exclusive, the processor issues a cluster consisting of a coherent block read request with exclusivity and write-forthcoming, followed by a processor block write request for the current cache line.

(4) If the current cache line does not need to be written back and the coherency attribute for the page that contains the requested cache line is sharable or exclusive, the procesors issues a coherent block read request that also requests exclusivity.

Nonsecondary Cache Mode. The processor issues a read request for the cache line that contains the data element to be loaded, and awaits the external agent to provide 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. If the new cache line replaces a current cache line whose Write Back (W) bit is set, the current cache line moves to an internal write buffer before the new cache line is loaded in the primary cache.

Store Hit

Nonsecondary Cache Mode. In nonsecondary cache mode, all lines are set to the dirty exclusive state. This means store hits cause no bus transactions.

Secondary Cache Mode. Same as the nonsecondary mode. But the secondary cache read cycle is generated to load the cache line that contains the data element to be loaded into the primary cache if miss in primary cache. If the current cache line in primary cache needs to be written back to secondary cache, the secondary cache write cycle is generated before the new cache line is read from secondary cache.

Uncached Loads or Stores

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

External requests have a higher priority than uncached stores. When using the uncached store buffer on the VR4400SC processor, it is possible for the external agent to receive cached and uncached stores out of program order.

If an external intervention or snoop is issued to the VR4400SC processor while the uncached store is still in the store buffer (the uncached store data has not yet been stored off-chip), the cached store has hit in the primary cache and is in the tag check (TC) stage of the pipeline. In this case, the external agent sees the state of the internal caches after the cached store but before the result of the uncached store is available off the chip.

The Sync instruction can force the uncached store to occur before the cached store.

Cache Operations

The processor provides a variety of cache operations to maintain the state and contents of the primary and secondary caches. During execution of the cache operation instructions, the VR4400SC processor can issue write requests.

CLOCKING INTERFACE

The MasterClock provides the fundamental timing and the internal operating frequency for the VR4400SC 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 and 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 speed 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 VR4400SC 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 at various speed ratings of the VR4400SC.

The clocks generated using PLL circuits contain some inherent inaccuracy, or jitter, in their alignment with 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 at various speed ratings of the VR4400SC.

The input signals of the VR4400SC 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 input latches. The output signals of the VR4400SC 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 output registers.

Certain processor inputs, such as V_{DD}Ok, $\overline{\text{ColdReset}}$, and $\overline{\text{Reset}}$, are sampled based on MasterClock, while certain 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 VR4400SC are in the AC Characteristics tables under Electrical Specifications.

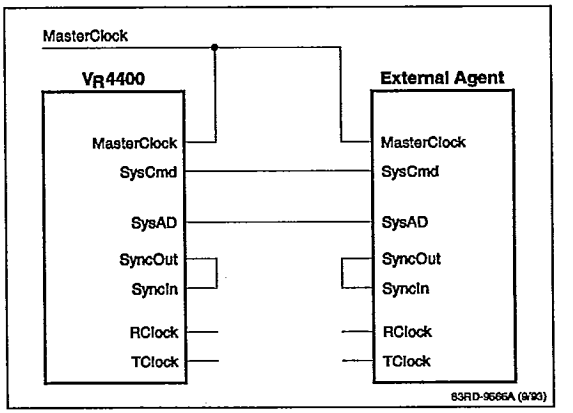
Clock Interfacing in a Phase-Locked System

When the processor is used 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 has 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 VR4400SC processor.

Figure 10. System With Phase-Lock



Clock Interfacing Without Phase-Lock

When the processor is 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.

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 VR4400SC)
+ (Minimum external clock buffer delay)
– (External sample register setup time)
– (Maximum VR4400SC 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 VR4400SC)
– (Maximum external clock buffer delay)
– (Maximum external output register clock-to-Q delay)
– (Maximum TClock jitter)
– (Maximum VR4400SC internal clock jitter)

Figure 11 is the block diagram of a system without phase-lock employing the VR4400SC 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 processor'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 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 VR4400SC)
– (External sample register setup time)
– (Maximum external clock buffer delay mismatch)
– (Maximum VR4400SC 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 VR4400SC)
– (Maximum external output register clock-to-Q delay)
– (Maximum external clock buffer delay mismatch)
– (Maximum VR4400SC 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 VR4400SC internal clocks, the maximum TClock jitter, and the maximum delay mismatch of the external clock buffers.

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

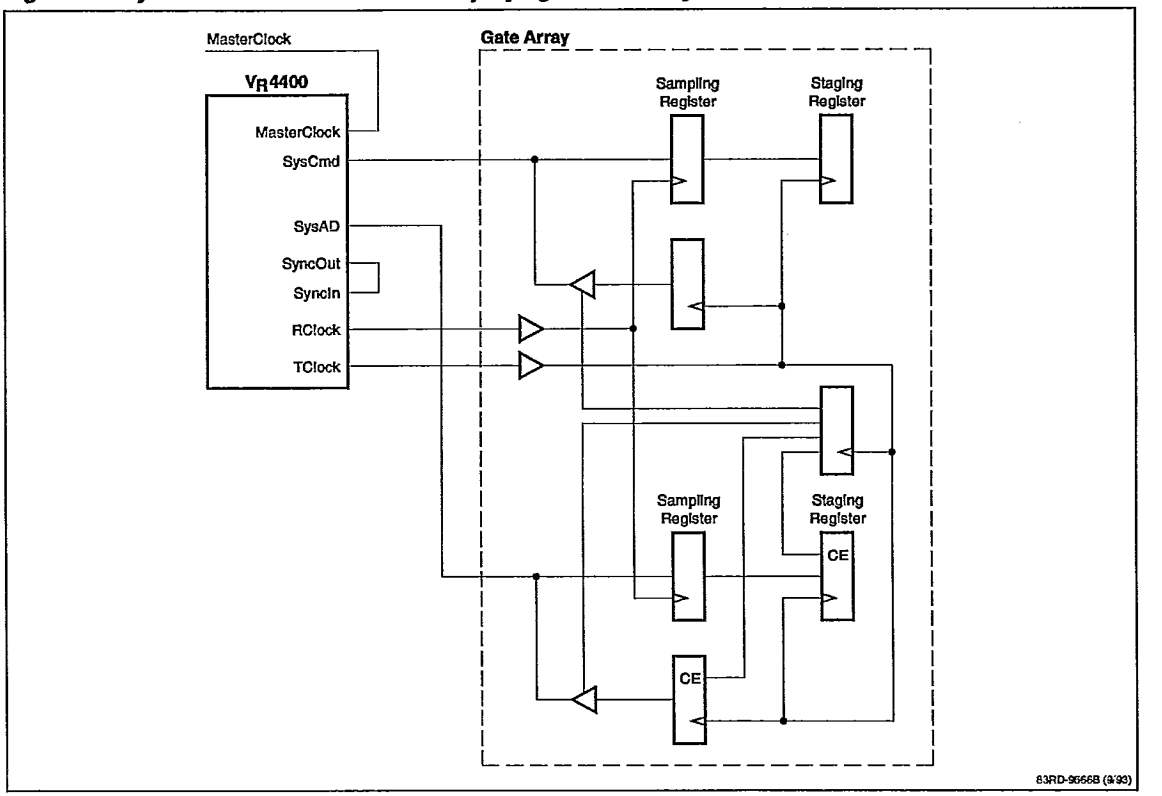


Figure 12 is the block diagram of a system without phase-lock employing the VR4400SC and an external agent composed of both a gate array and discrete CMOS logic devices.

Processor Status Outputs

The VR4400SC processor outputs 8 status bits, Status(7:0), aligned with each rising edge of MasterClock. The 8 bits are treated as two hex fields that indicate the internal processor status (codes 0 thru F in table 5) during the T-2 PCycle, Status(3:0), and T-3 PCycle, Status(7:4).

When status is examined at time T (the first PCycle of MasterClock), the status code indicates whether the machine was stalled or running during the previous T-2 and T-3 PCycles.

(1) If the machine was stalled, the status code indicates the type of stall.

(2) If the machine was running, the status code identifies the type of instruction—successfuly completed—that occupied the WriteBack pipeline stage during the T-2 or T-3 PCycle.

(3) The status code also indicates if an instruction in the T-2 or T-3 PCycle was killed and for what reason.

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

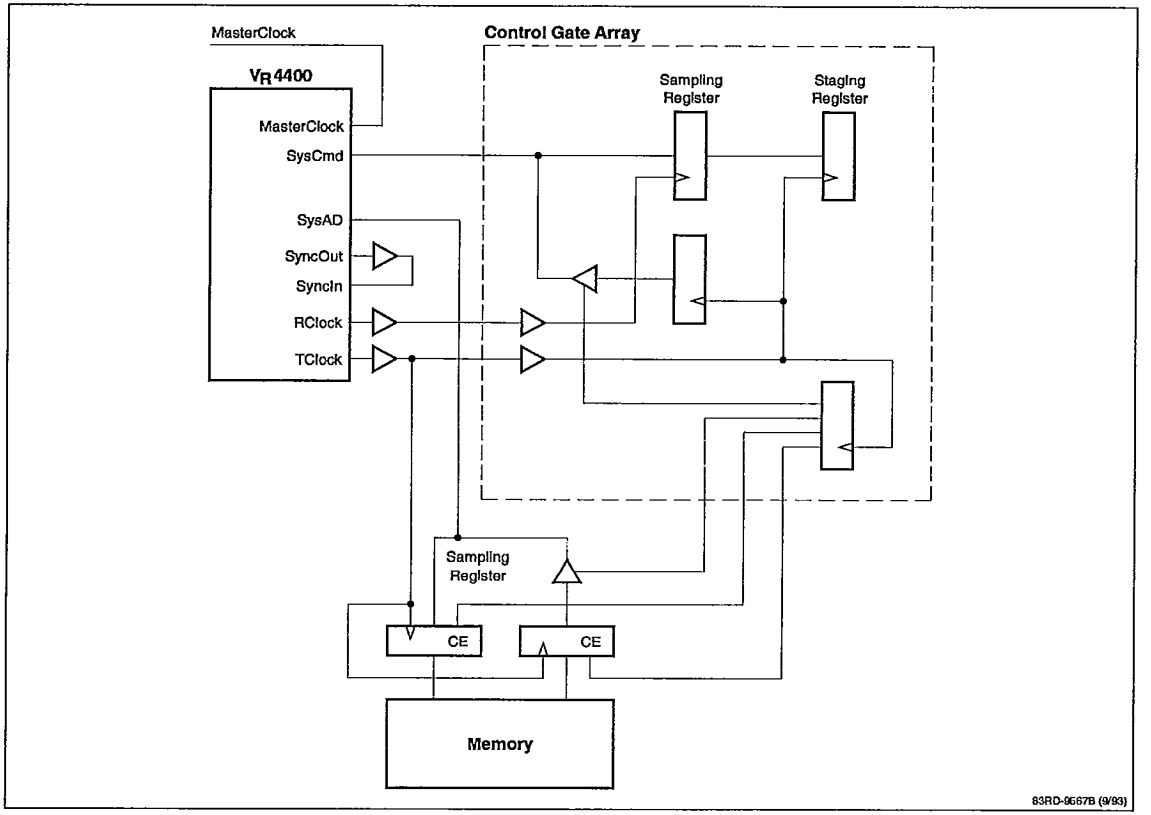


Table 5. VR4400SC Internal Status Outputs

Status Code	Cycle	Processor Internal Status
0	Run	Other integer instruction (not load, store, or conditional branch); includes ERET and Jump instructions
1	Run	Load
2	Run	Untaken conditional branch
3	Run	Taken conditional branch
4	Run	Store
5		Reserved
6	Stall	MP stall
7	Run	Integer instruction killed by slip
8	Stall	Other stall type
9	Stall	Primary instruction cache stall
A	Stall	Primary data cache stall
B	Stall	Secondary cache stall
C	Run	Other floating-point instruction (not load, store, or conditional branch)
D	Run	Instruction killed by branch, jump, or ERET
E	Run	Instruction killed by exception
F	Run	Floating-point instruction killed by slip

INITIALIZATION INTERFACE

The operation of the VR4400SC 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 VR4400SC. 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 Mode

The correspondence between bits of the intialization bit stream and processor mode settings is illustrated in table 6. 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

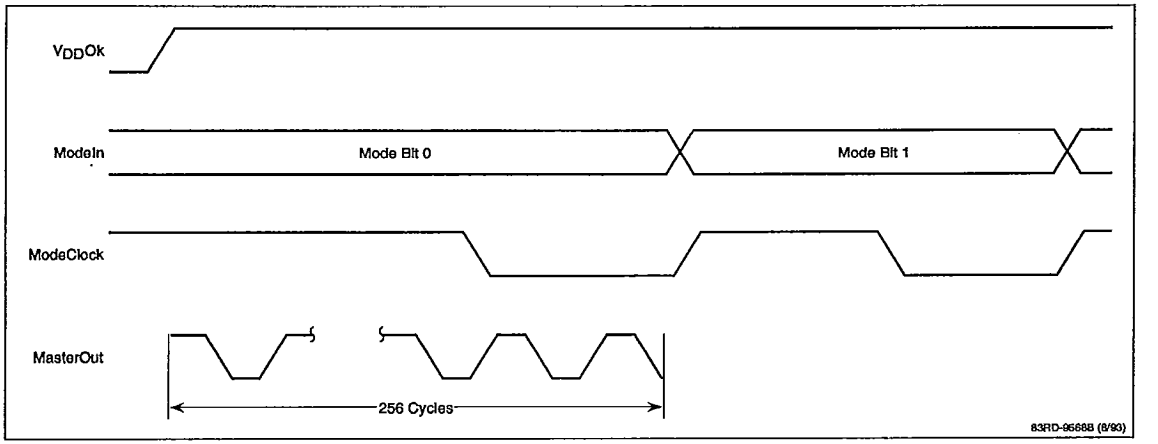


Table 6. Boot-Time Mode

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

Table 6. Boot-Time Mode (cont)

Bit	Value	Processor Mode Setting
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)
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)

Table 6. Boot-Time Mode (cont)

Bit	Value	Processor Mode Setting
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
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

Table 6. Boot-Time Mode (cont)

Bit	Value	Processor Mode Setting
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 VR4400SC 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 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 ColdReset input may be deasserted. ColdReset must remain asserted for at least 64 MasterClock cycles after the assertion of V_{DD}Ok and must be deasserted synchronously with MasterClock.

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

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

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

Warm Reset

To produce a warm reset, the 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 will depend on whether a cache miss sequence has been interrupted by resetting the processor state machines.

JTAG INTERFACE

The VR4400SC 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 VR4400SC, the printed circuit board to which it is attached, and the other components on the board. In addition, the JTAG boundary scan mechanism provides a rudimentary capability for low-speed logical testing of the secondary cache RAMs. The JTAG boundary scan mechanism does not provide any capability for testing the VR4400SC itself.

In accordance with the JTAG specification, the VR4400SC contains a TAP controller, JTAG Instruction register, JTAG Boundary Scan register, JTAG Identification register, and JTAG Bypass register. However, the VR4400SC JTAG implementation provides only the external test functionality of the boundary scan register.

SECONDARY CACHE INTERFACE

The VR4400SC microprocessor contains interface signals for an optional external secondary cache. This interface consists of:

- 128-bit data bus
- 25-bit tag bus
- 18-bit address bus
- Various static random access memory (SRAM) control signals

The 128-bit-wide data bus minimizes the primary cache miss penalty and allows standard, low-cost SRAMs in the design of the secondary cache.

Data Transfer Rates

The interface to the secondary cache maximizes service of primary cache misses. The secondary cache interface, SCData(127:0), supports a data rate that is close to the processor-to-primary-cache bandwidth during normal operation. To ensure that this bandwidth is maintained, each data, tag, and check pin must be connected to a single SRAM device.

The SCAddr bus, together with SCOE, SCDCS, and SCTCS signals, drives a large number of SRAM devices. Consequently, one level of external buffering between the processor and the cache array is used.

Duplicating Signals

Buffered signals control the speed of the secondary cache interface. Critical control signals are duplicated by design to minimize this limitation; the SCWR signal and SCAddr0 have four versions so that external buffers are not needed to drive them. When an eight-word (256-bit) primary cache line is used, these signals can be controlled quickly, reducing the time of back-to-back transfers.

Each duplicated control signal can drive up to 11 SRAMs; therefore, a total of 44 SRAM packages can be used in the cache array. This allows a cache design using 16-Kbyte by 64-bit, 64-Kbyte by 4-bit, or 256-Kbyte by 4-bit standard SRAMs.

Other cache designs within the above constraint are also acceptable. For example, a smaller cache design can use 22 SRAMS, 8-Kbyte by 8-bit; this design presents less load on the address pins and control signals and reduces the overall parts count.

The benefit of duplicating SCAddr0 is greater in systems that use fast sequential static cache RAM and an eight-word primary cache line. If SCAddr0 is attached to the SRAM address bit that affects column decode only, the read cycle time should approximate the output enable time of the RAM. For fast static RAM, this cycle time should be half the nominal read cycle time.

Accessing a Split Secondary Cache

When the secondary cache is split into separate instruction and data portions, assertion of the high-order SCAddr17 bit enables the instruction half of the cache.

It is possible to design a cache that supports both joint and split instruction/data configurations of less than the maximum cache size; in doing so, SCAddr(12:0) must address the cache in all configurations. SCAddr17 must support the split instruction/data configuration, and any of SCAddr(16:14) bits can be omitted because of the fixed width of the physical tag array.

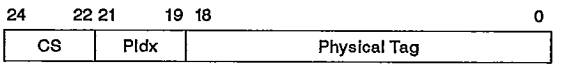
SCDChk Bus

The secondary cache data check bus, SCDChk, is divided into two fields to cover the upper and lower 64 bits of SCData. This form is required by the 64-bit width of internal data paths.

SCTag Bus

The secondary cache tag bus, SCTag, is divided into three fields as shown in figure 14. The CS field indicates the cache state: invalid, clean exclusive, dirty exclusive, shared, or dirty shared. The PIdx field is an index to the virtual address of primary cache lines that can contain data from the secondary cache. Bits 18:0 contain the upper physical address.

Figure 14. SCTag Fields



The $\overline{SCDCS}$ and $\overline{SCTC}$ signals disable reads or writes of either the data array or tag array when the opposite array is being accessed. These signal are useful for saving power on snoop and invalidate requests since access to the data array is not necessary. The signals also write data from the primary data cache to the secondary cache.

Operation of the Secondary Cache Interface

The secondary cache can be configured for various clock rates and static RAM speeds. All configurable parameters are specified in multiples of PClock, which runs at twice the frequency of the external system clock, MasterClock.

During boot time, secondary cache timing parameters are programmed through the boot-time mode bits.

Parameter	Number of PCycles
t_{Rd1Cyc}	4-15
t_{Rd2Cyc}	3-15
t_{Dis}	2-7
t_{Wr1Dly}	1-3
t_{Wr2Dly}	1-3
t_{WrRC}	0-1
t_{WrSUp}	3-15

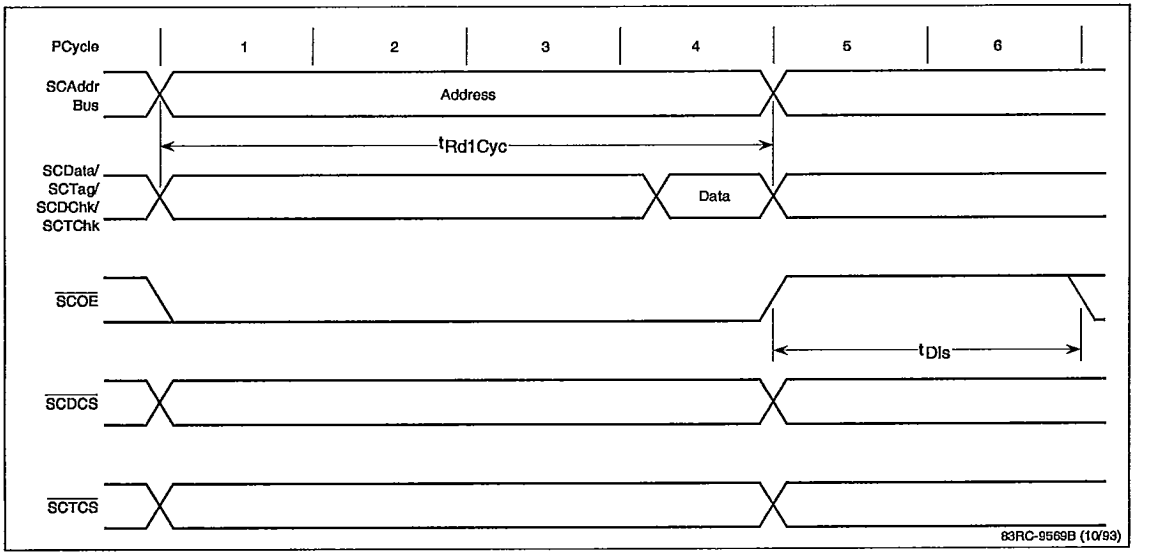
Read Cycles

There are two basic read cycles: four-word and eight-word. Each secondary cache read cycle begins by driving an address out on the address pins. The output enable signal $\overline{SCOE}$ is asserted at the same time.

Four-Word Read Cycle. The four-word read cycle (figure 15) has two user-accessible timing parameters:

- t_{Rd1Cyc} Read sequence cycle time, which specifies the time from assertion of the SCAddr bus to sampling the SCData bus.
- t_{Dis} Cache output disable time, which specifies the time from the end of a read cycle to the start of the next write cycle.

Figure 15. Timing Diagram of a Four-Word Read Cycle



Eight-Word Read Cycle. The eight-word read cycle (figure 16) has an additional user-accessible parameter beyond the four-word read cycle: t_{Rd2Cyc}, the time from the first sample point to the second sample point.

In an eight-word read cycle, the low-order address bit, SCAddr0, changes at the same time as the first read sample point.

Read Cycle Abortion. All read cycles can be aborted by changing the address; a new cycle begins with the edge on which the address is changed. Additionally, the period t_{Dis} after a read cycle can be interrupted any time by the start of a new read cycle. If a read cycle is aborted by a write cycle, $\overline{SCOE}$ must be deasserted for the t_{Dis} period before the write cycle can begin.

Read cycles can also be extended indefinitely. There is no requirement to change the address at the end of a read cycle.

Write Cycles

There are two basic write cycles: four-word and eight-word. The secondary cache write cycle begins with the assertion of an address onto the address pins.

Four-Word Write Cycle. The four-word write cycle (figure 17) has three timing parameters:

- t_{Wr1Dly} Delay from assertion of the address to assertion of $\overline{SCWR}$.
- t_{WrSUp} Delay from assertion of the second data doubleword to deassertion of $\overline{SCWR}$.
- t_{WrRc} Delay from deassertion of $\overline{SCWR}$ to the beginning of the next cycle

The timing parameter t_{WrRc} is 0 for most cache designs. The upper data doubleword and the lower data doubleword are normally driven one cycle apart to reduce peak current consumption in the output drivers. Either can be driven first.

Eight-Word Write Cycle. The eight-word write cycle (figure 18) has one additional parameter (t_{Wr2Dly}) beyond the four-word write cycle. This time period begins when the low-order address bit SCAddr0 changes and ends when $\overline{SCWR}$ is asserted for the second time. The lower half of SCData is driven on the same edge as the change in SCAddr0.

Timing. When data is received from the system interface, the first data doubleword can arrive several cycles before the second data doubleword. In this case, the cache state machine enters a wait state that extends $\overline{SCWR}$ until the t_{WrSUp} period after the second data item is transmitted.

Figure 16. Timing Diagram of an Eight-Word Read Cycle

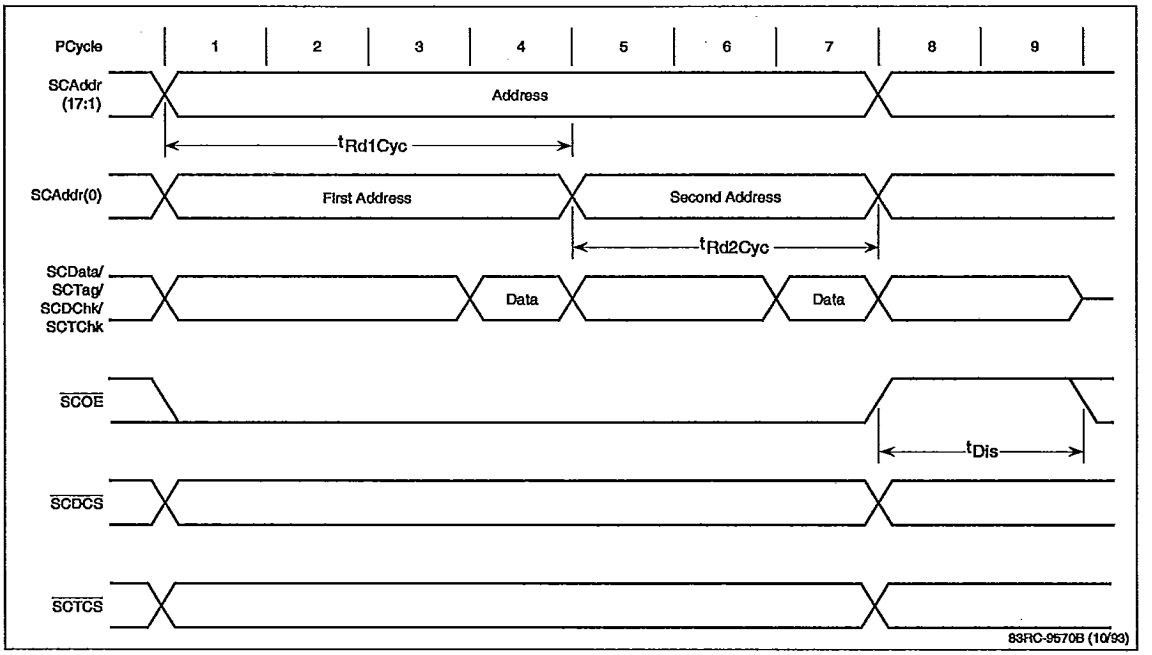


Figure 17. Timing Diagram of a Four-Word Write Cycle

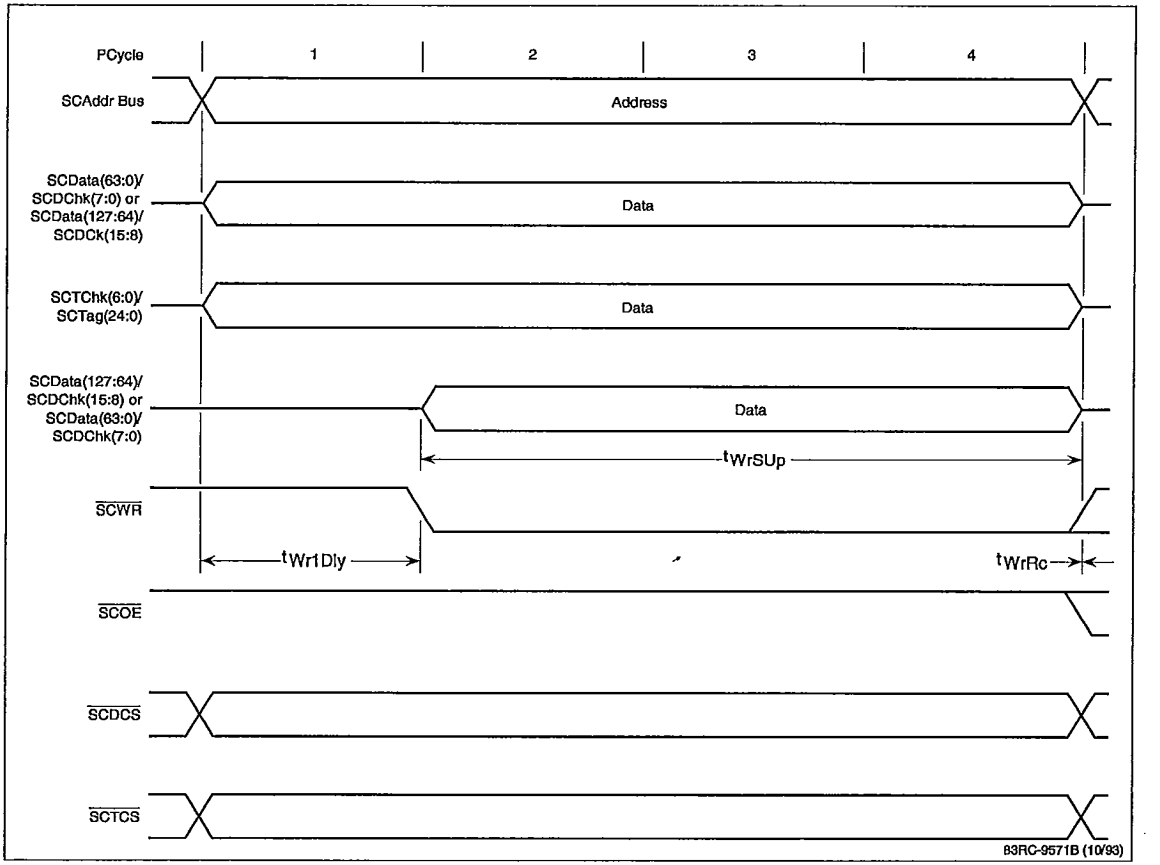
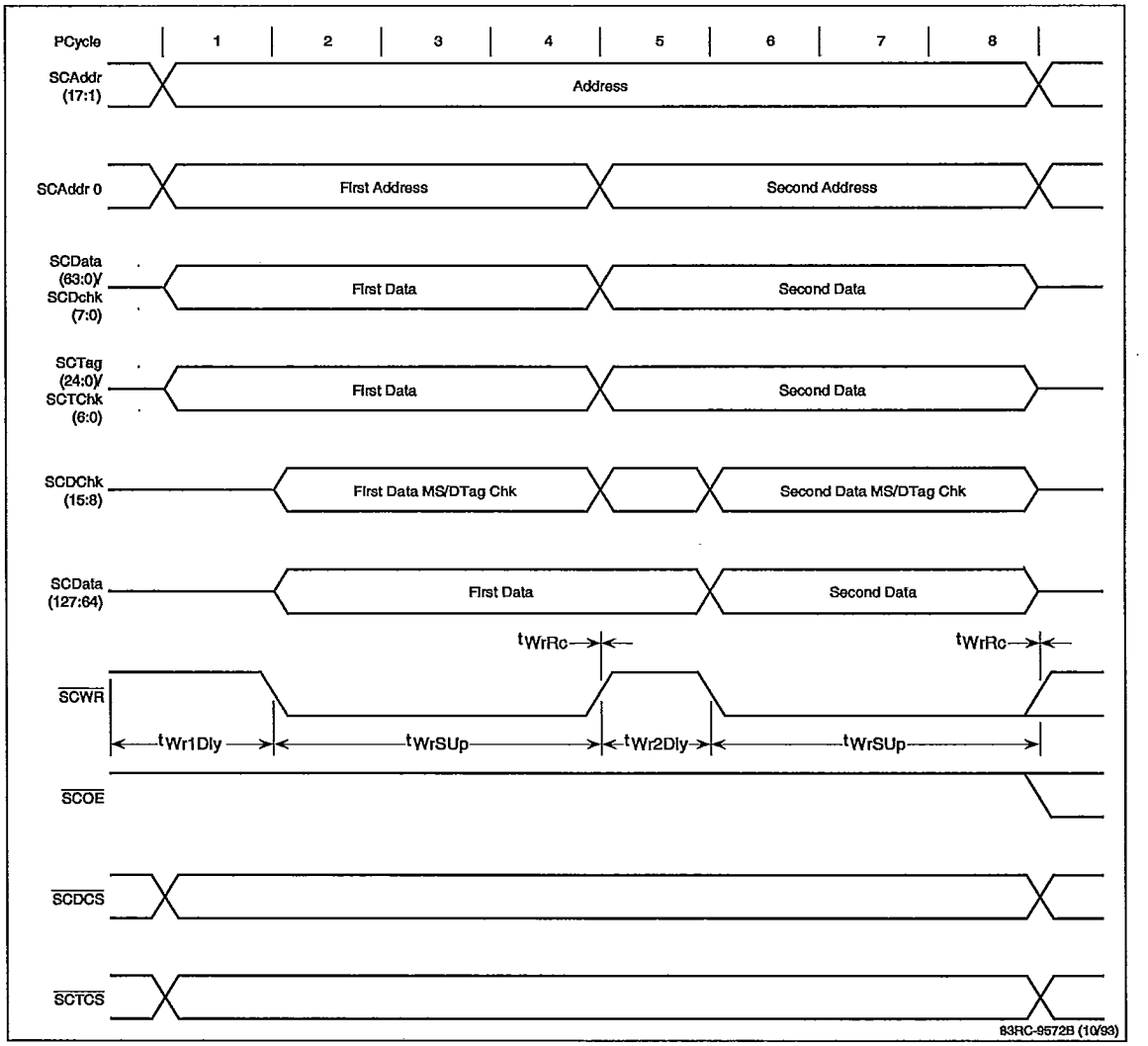


Figure 18. Timing Diagram of an Eight-Word Write Cycle



ELECTRICAL SPECIFICATIONS

Power Distribution

The VR4400SC 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, more than 50 pins each are assigned to V_{DD} and GND inputs.

Liberal decoupling capacitors should be installed near the VR4400SC. Driving the 128-bit secondary cache data bus or 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 VR4400SC 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
3.3-volt version		2.4			V	
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 version	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 version	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
Secondary cache data setup	t_{SCDS}	5		ns	
Secondary cache data hold	t_{SCDH}	2		ns	
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	
Cycles between Addr and $\overline{SCWr}$	t_{Wr1Dly}	1	3	PClk	
Cycles from deassertion of $\overline{SCWr}$ to start of next cycle	t_{WrRc}	0	1	PClk	
Cycles from 2nd doubleword to $\overline{SCWr}$	t_{WrSUp}	2	15	PClk	
Cycles between 1st and 2nd data words in eight-word write	t_{Wr2Dly}	1	3	PClk	

Notes:

(1) Operation of the VR4400SC 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 VR4400SC on the system interface. Secondary cache signals are specified separately.

(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
Secondary cache data setup	t_{SCDS}	5		ns	
Secondary cache data hold	t_{SCDH}	1.5		ns	
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	
Cycles between Addr and $\overline{SCWr}$	t_{Wr1Dly}	1	3	PClk	
Cycles from deassertion of $\overline{SCWr}$ to start of next cycle	t_{WrRc}	0	1	PClk	
Cycles from 2nd doubleword to $\overline{SCWr}$	t_{WrSUp}	2	15	PClk	
Cycles between 1st and 2nd data words in eight-word write	t_{Wr2Dly}	1	3	PClk	

Notes:

(1) Operation of the VR4400SC 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 VR4400SC on the system interface. Secondary cache signals are specified separately.

(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
Secondary cache data setup	t_{SCDS}	3.5		ns	
Secondary cache data hold	t_{SCDH}	1		ns	
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	
Cycles between Addr and $\overline{SCWr}$	t_{Wr1Dly}	1	3	PClk	
Cycles from deassertion of $\overline{SCWr}$ to start of next cycle	t_{WrRc}	0	1	PClk	
Cycles from 2nd doubleword to $\overline{SCWr}$	t_{WrSUp}	3	15	PClk	
Cycles between 1st and 2nd data words in eight-word write	t_{Wr2Dly}	1	3	PClk	

Notes:

(1) Operation of the VR4400SC 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 VR4400SC on the system interface. Secondary cache signals are specified separately.

(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
Secondary cache data setup	t_{SCDS}	3.5		ns	
Secondary cache data hold	t_{SCDH}	1.5		ns	
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	
Cycles between Addr and $\overline{SCWr}$	t_{Wr1Dly}	1	3	PClk	
Cycles from deassertion of $\overline{SCWr}$ to start of next cycle	t_{WrRc}	0	1	PClk	
Cycles from 2nd doubleword to $\overline{SCWr}$	t_{WrSUp}	2	15	PClk	
Cycles between 1st and 2nd data words in eight-word write	t_{Wr2Dly}	1	3	PClk	

Notes:

(1) Operation of the VR4400SC 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 VR4400SC on the system interface. Secondary cache signals are specified separately.

(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
Secondary cache data setup	t_{SCDS}	3.5		ns	
Secondary cache data hold	t_{SCDH}	1.5		ns	
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	
Cycles between Addr and $\overline{SCWr}$	t_{Wr1Dly}	1	3	PClk	
Cycles from deassertion of $\overline{SCWr}$ to start of next cycle	t_{WrRc}	0	1	PClk	
Cycles from 2nd doubleword to $\overline{SCWr}$	t_{WrSUp}	2	15	PClk	
Cycles between 1st and 2nd data words in eight-word write	t_{Wr2Dly}	1	3	PClk	

Notes:

(1) Operation of the VR4400SC 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 VR4400SC on the system interface. Secondary cache signals are specified separately.

(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
Secondary cache data setup	t_{SCDS}	3.5		ns	
Secondary cache data hold	t_{SCDH}	1		ns	
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	
Cycles between Addr and $\overline{SCWr}$	t_{Wr1Dly}	1	3	PClk	
Cycles from deassertion of $\overline{SCWr}$ to start of next cycle	t_{WrRc}	0	1	PClk	
Cycles from 2nd doubleword to $\overline{SCWr}$	t_{WrSUp}	2	15	PClk	
Cycles between 1st and 2nd data words in eight-word write	t_{Wr2Dly}	1	3	PClk	

Notes:

(1) Operation of the VR4400SC 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.

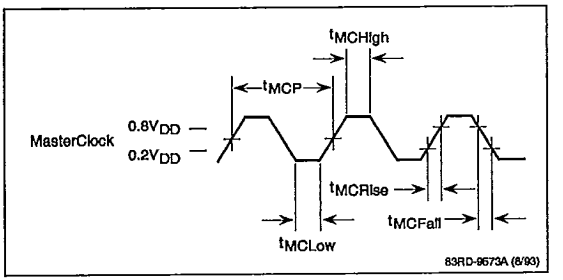
(4) Timing is measured from 1.5 V of SClock to 1.5 V of the signal.

(5) Data output, setup, and hold times apply to all logic signals driven out of or driven into the VR4400SC on the system interface. Secondary cache signals are specified separately.

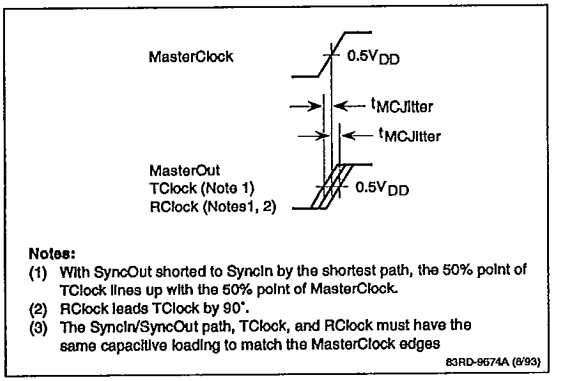
(6) Number of cycles is configured through the boot-time mode control.

Timing Diagrams

Master Clock

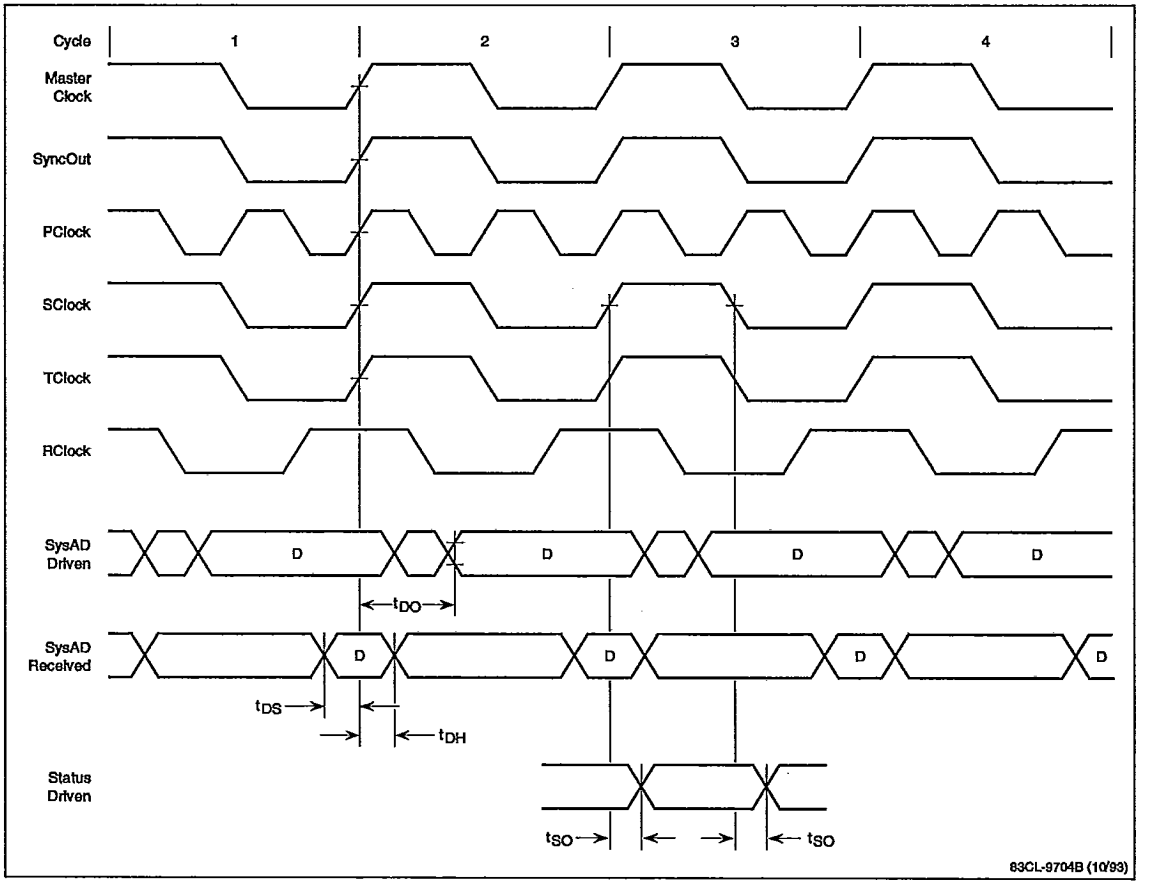


Clock Jitter



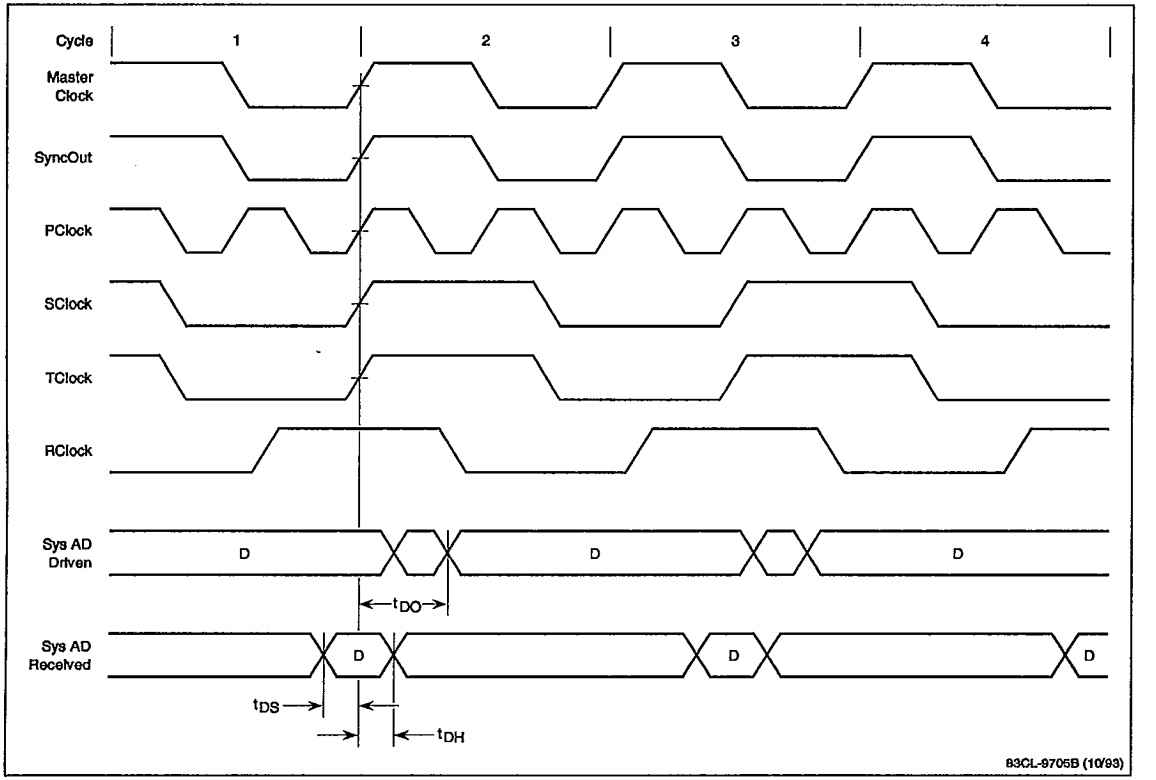
Timing Diagrams (cont)

Processor Clock; PClock to SClock, Divisor of 2



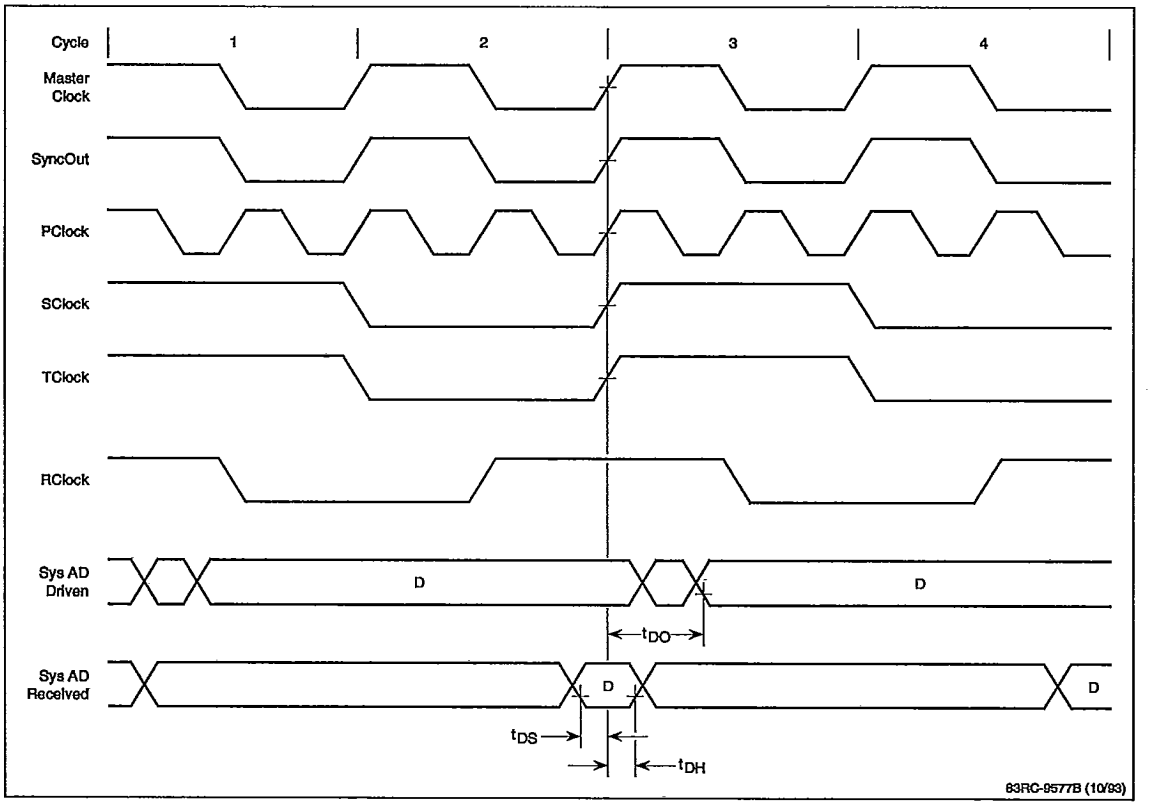
Timing Diagrams (cont)

Processor Clock; PClock to SClock, Divisor of 3



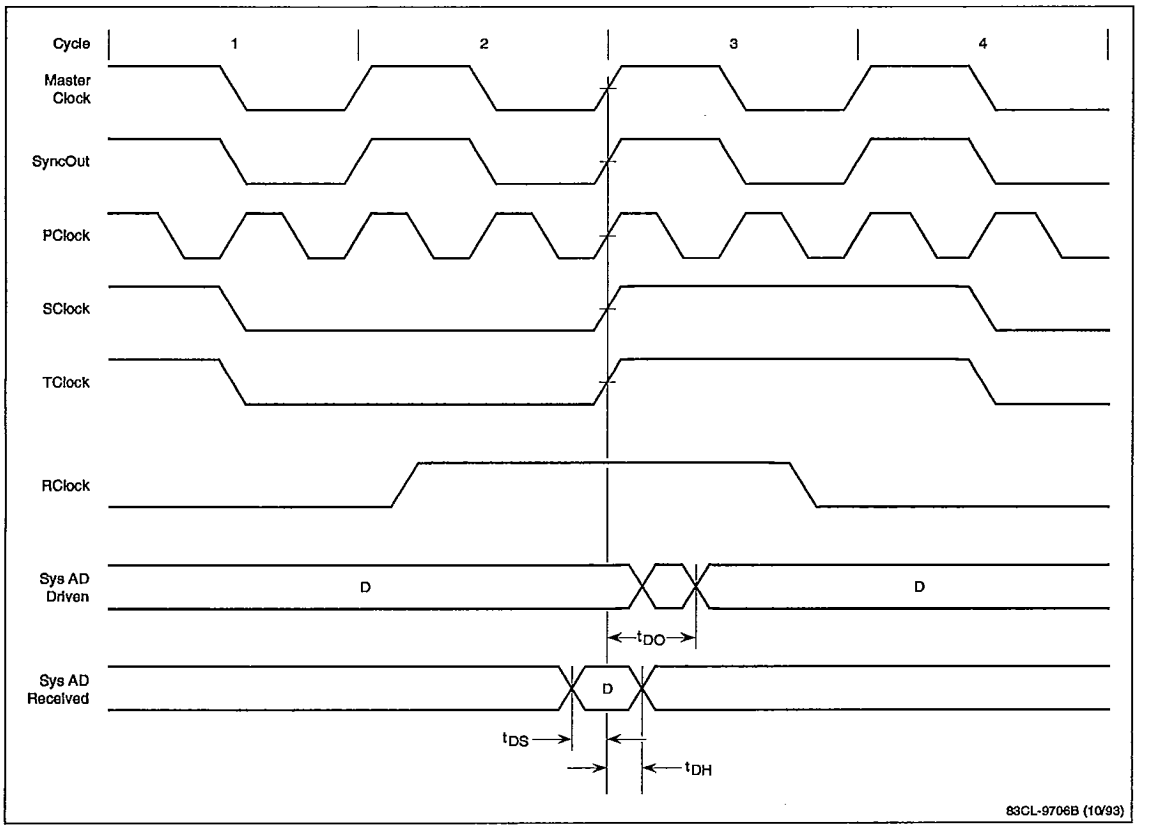
Timing Diagrams (cont)

Processor Clock; PClock to SClock, Divisor of 4



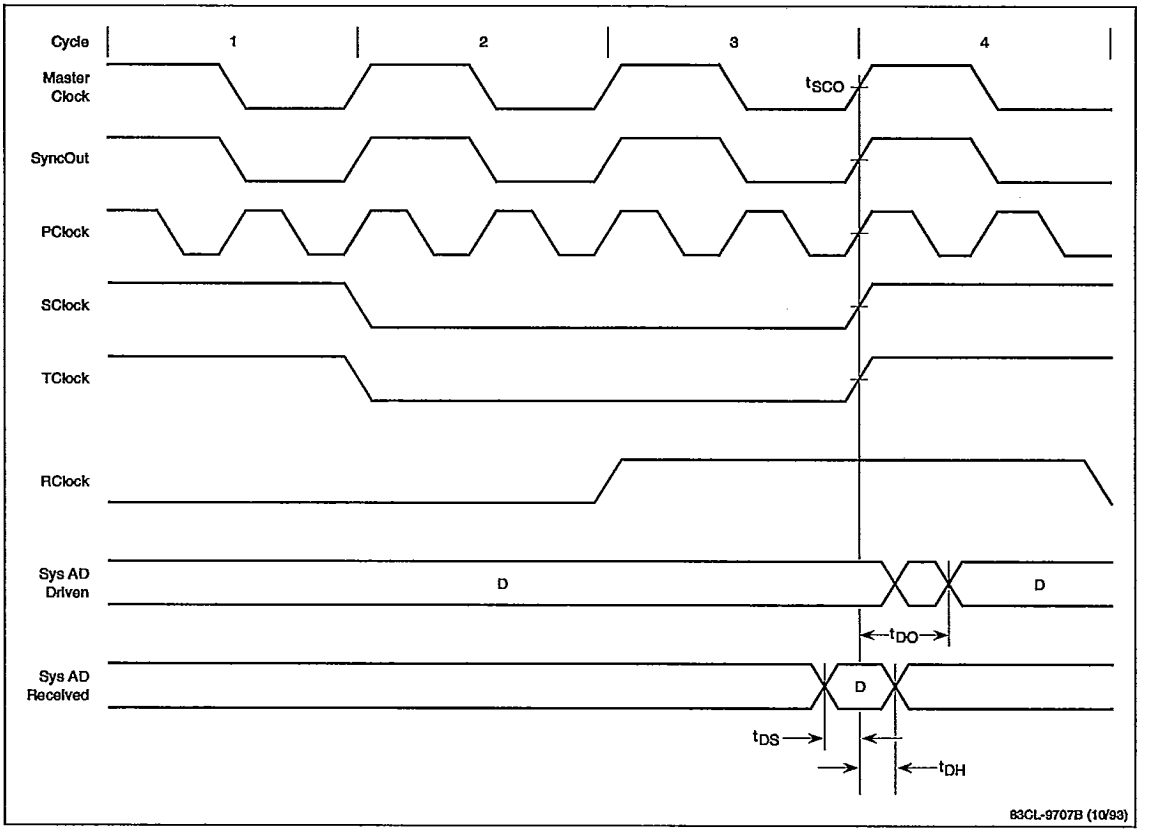
Timing Diagrams (cont)

Processor Clock; PClock to SClock, Divisor of 6



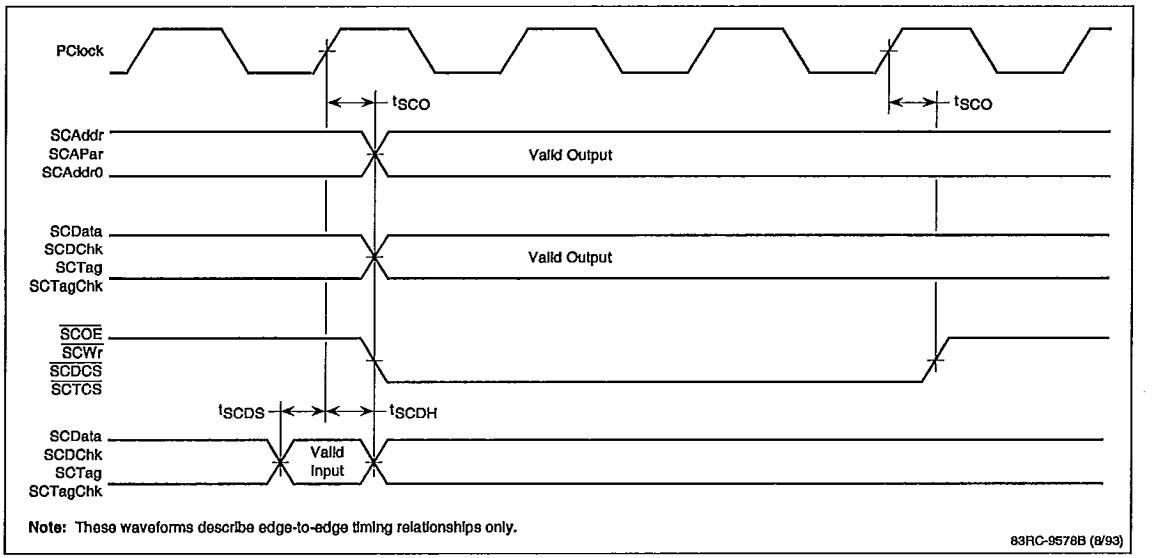
Timing Diagrams (cont)

Processor Clock; PClock to SClock, Divisor of 8



Timing Diagrams (cont)

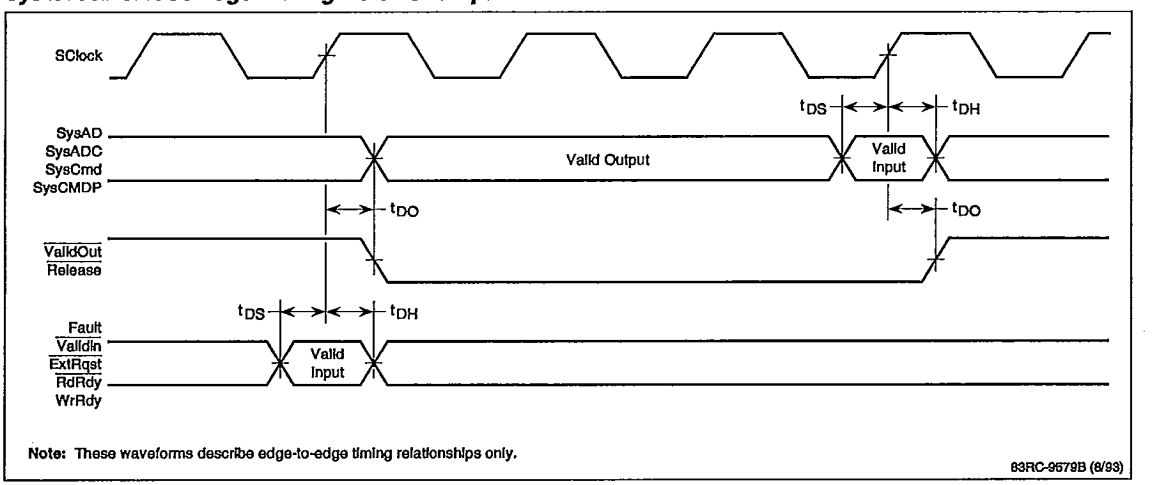
Secondary Cache Edge Timing Relationships



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

83RC-9578B (8/93)

System Interface Edge Timing Relationships

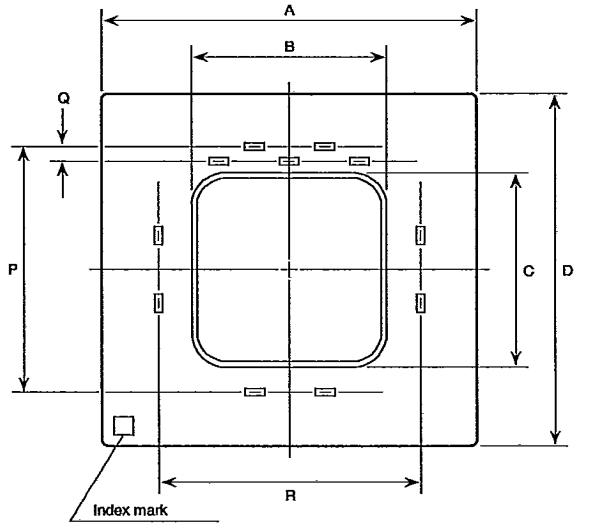


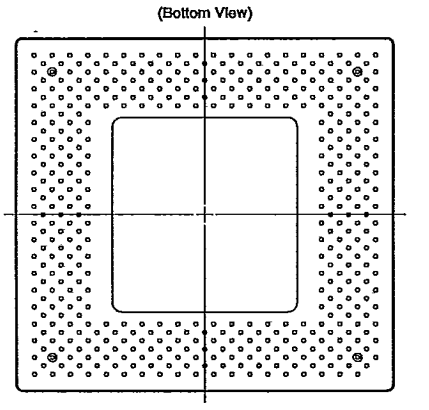
Note: These waveforms describe edge-to-edge timing relationships only.

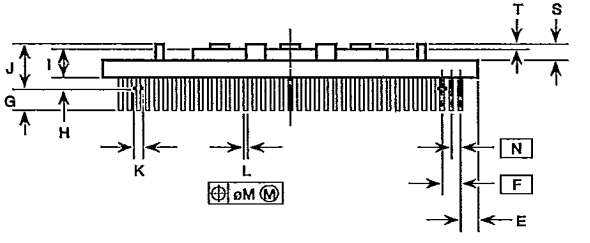
83RC-9579B (8/93)

PACKAGE DRAWINGS

447-Pin Ceramic PGA (Metal Sealed)







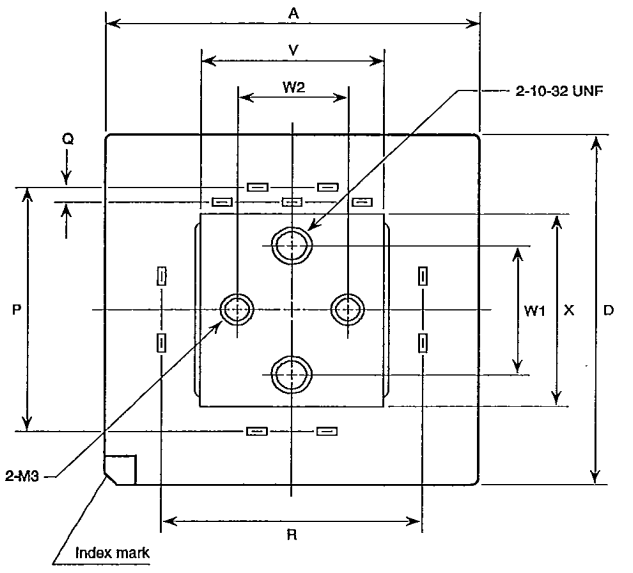
Item	Millimeters	Inches
A	52.32 ± 0.52	2.060 ± .021
B	26.92	1.060
C	29.46	1.160
D	52.32 ± 0.52	2.060 ± .021
E	2.03	.080
F	2.54 (T.P.)	.100 (T.P.)
G	3.3 ± 0.2	.130 ± .008
H	0.635 min	.025 min
I	4.34	.171
J	6.19 max	.244 max
K	ø1.27 ± 0.2	ø0.050 ± .008
L	ø0.46 ± 0.05	ø0.018 ± .002
M	0.254	.010
N	1.27 (T.P.)	.050 (T.P.)
P	36.28	1.428
Q	1.60 min	.082 min
R	33.72	1.328
S	1.54 max	.061 max
T	0.42 max	.017 max

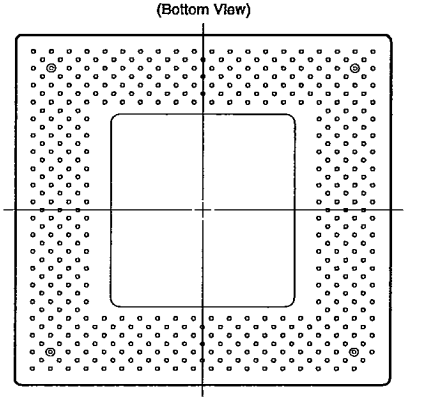
Note
Each lead centerline is located within ø0.254 mm (ø.010 inch) of its true position (T.P.) at maximum material condition.

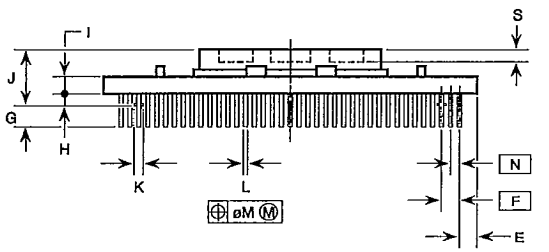
X447RG-50A-2 83RD-9580B (8/93)

PACKAGE DRAWINGS (cont)

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







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

Item	Millimeters	Inches
A	52.32 ± 0.52	2.060 ± .021
D	52.32 ± 0.52	2.060 ± .021
E	2.03	.080
F	2.54 (T.P.)	.100 (T.P.)
G	3.3 ± 0.2	.130 ± .008
H	.635 min	.025 min
I	4.34	.171
J	11.69 max	.461 max
K	ø1.27 ± 0.2	ø0.050 ± .008
L	ø0.46 ± 0.05	ø0.018 ± .002
M	0.254	.010
N	1.27 (T.P.)	.050 (T.P.)
P	36.28	1.428
Q	1.80	.082 min
R	33.72	1.328
S	1.54 max	.061 max
V	25.0	.984
W1	21.59	.850
W2	15.0	.591
X	29.5	1.161

X447RP-50A

83RD-9581B (8/93)