PRELIMINARY PRODUCT INFORMATION

NEC

MOS INTEGRATED CIRCUIT

μPD30100

VR4100™
64-BIT MICROPROCESSOR

The μPD30100 (VR4100) is a high-performance, 64-bit RISC (Reduced Instruction Set Computer) type microprocessor employing the RISC architecture developed by MIPS.

The VR4100 is compact and consumes little power so that it can be used in battery-driven, high-performance portable systems. In addition, a multiplier circuit is provided to increase the speed.

FEATURES

- Employs MIPS 64-bit RISC architecture
- High-speed processing
  - 5-stage pipelining
  - Multiplier circuit
- Instruction set compatible with VR4000™ series (conforming to MIPS-I/II/III)
  - Sum-of-products operation instructions added
  - Floating-point operation instructions deleted
- Low power dissipation of 120 mW TYP. (at 40 MHz, 3.3 V)
- Supports three power control modes in addition to normal operation modes
- Virtual memory management unit (MMU)
- 32-bit address/data multiplexed bus facilitating system designing
- Supply voltage: 2.2 to 3.6 V
- Package: 100-pin plastic TQFP

APPLICATIONS

- Battery-driven portable systems
- Embedded controller, etc.

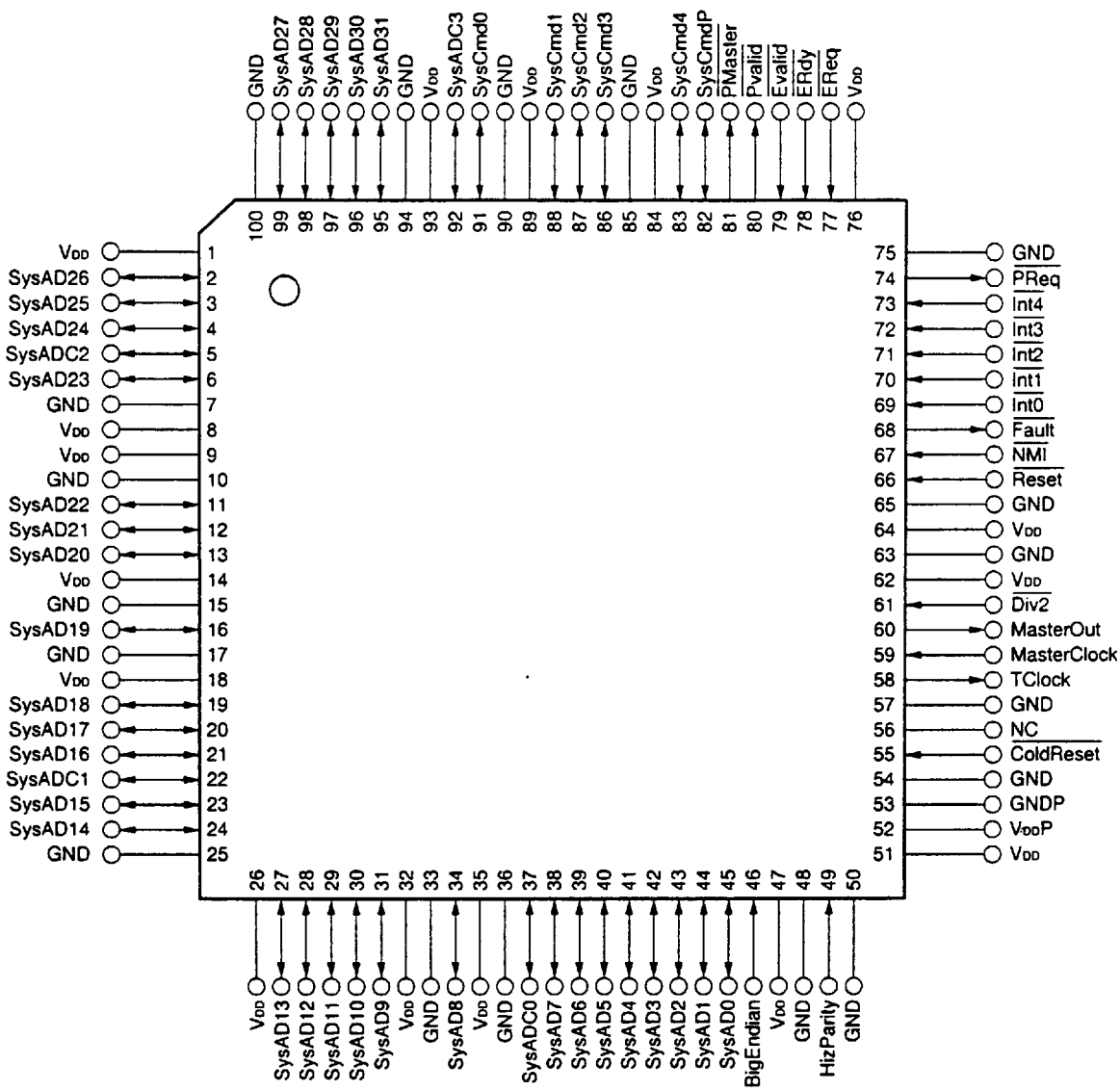
ORDERING INFORMATION

Part Number	Package
μPD30100GC-40-9EU	100-pin plastic TQFP (fine pitch) (14 × 14 mm)

The information contained in this document is being issued in advance of the production cycle for the device. The parameters for the device may change before final production or NEC Corporation, at its own discretion, may withdraw the device prior to its production.

Document No. U10428EJ1V0PM00 (1st edition)
Date Published September 1995 P
Printed in Japan

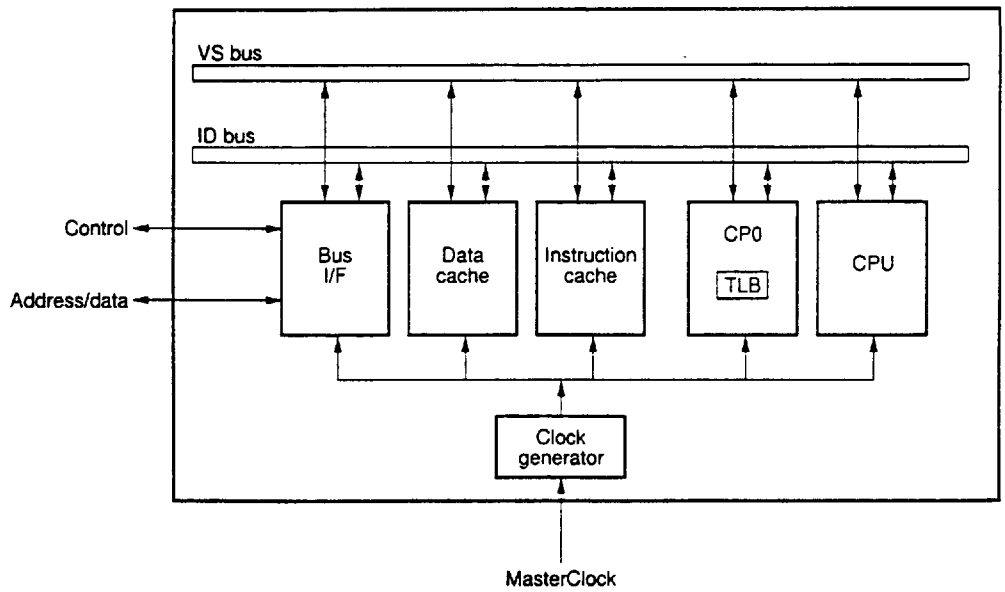
PIN CONFIGURATION



PIN NAMES

BigEndian	:	Big Endian
$\overline{\text{ColdReset}}$	:	Cold Reset
$\overline{\text{Div2}}$	:	Divide by 2
$\overline{\text{ERdy}}$	:	External Ready
$\overline{\text{EReq}}$	:	External Request
$\overline{\text{EValid}}$	:	External Valid
$\overline{\text{Fault}}$	:	Fault
HizParity	:	Hi Impedance Parity
$\overline{\text{Int(4:0)}}$	:	Interrupt Request
MasterClock	:	Master Clock
MasterOut	:	Master Clock Out
$\overline{\text{NMI}}$	:	Non-maskable Interrupt Request
$\overline{\text{PMaster}}$	:	Processor Master
$\overline{\text{PReq}}$	:	Processor Request
$\overline{\text{PValid}}$	:	Processor Valid
$\overline{\text{Reset}}$	:	Reset
SysAD(31:0)	:	System Address/Data Bus
SysADC(3:0)	:	System Address/Data Check
SysCmd (4:0)	:	System Command/Data ID Bus
SysCmdP	:	System Command Parity
TClock	:	Transmit Clock
V_{DD}	:	Power Supply
GND	:	Ground
$V_{DD}P$	:	V_{DD} for PLL
GNDP	:	GND for PLL

INTERNAL BLOCK DIAGRAM



CONTENTS

1. PIN FUNCTIONS

Pin Name	I/O	Function
SysAD (31:0)	I/O	System address/data bus. 32-bit bus for communication between processor and external agent.
SysADC (3:0)	I/O	System address/data check bus. 4-bit bus for checking SysAD bus.
SysCmd (4:0)	I/O	System command/data ID bus. 5-bit bus for communication of commands and data identifiers between processor and external agent.
SysCmdP	I/O	System command/data ID bus parity. 1-bit bus even number parity bit for SysCmd.
$\overline{\text{EValid}}$	Input	External valid. Signal indicating that external agent has transmitted valid address or data onto SysAD bus and valid command or data identifier onto SysCmd bus.
$\overline{\text{PValid}}$	Output	Processor valid. Signal indicating that processor has transmitted valid address or data onto SysAD bus and valid command or data identifier onto SysCmd bus.
$\overline{\text{EReq}}$	Input	External request. Signal used by external agent to request for issuance by system interface.
$\overline{\text{PReq}}$	Output	Processor request. Signal used by processor to request for issuance by system interface.
$\overline{\text{PMaster}}$	Output	Processor master. Signal indicating that processor controls system interface.
$\overline{\text{ERdy}}$	Input	External ready. Signal indicating that external agent can accept processor request.
$\overline{\text{Fault}}$	Output	Fault. Signal indicating that a parity error has occurred in SysCmd bus.
$\overline{\text{Int (4:0)}}$	Input	Interrupt. General-purpose processor interrupt requests whose input statuses can be confirmed by bits 14 through 10 of cause register.
$\overline{\text{NMI}}$	Input	Non-maskable interrupt. Interrupt request that cannot be masked.
$\overline{\text{ColdReset}}$	Input	Cold reset. Signal that initializes internal status of processor.
$\overline{\text{Reset}}$	Input	Reset. Signal that generates reset exception without initializing internal status of processor.
MasterClock	Input	Master clock. Clock input signal to processor.
MasterOut	Output	Master clock output. Output of master clock synchronized with master clock.
TClock	Output	Transmit clock. Transmit clock of system interface.

Pin Name	I/O	Function
BigEndian	Input	Big Endian. Switches endian mode of system interface.
$\overline{\text{Div2}}$	Input	Operation speed of system interface. Switches the frequency divisor for the pipeline clock for the system interface frequency.
HizParity	Input	High impedance parity mode. Determines whether the output of the parity pins (SysADC (3:0), SysCmdP) is set to high impedance and the external bus parity check is stopped.
$V_{DD}P$	–	PLL V_{DD}. Power supply for internal PLL.
GNDP	–	PLL GND. Ground for internal PLL.
V_{DD}	–	Positive power supply pin.
GND	–	Ground pin.

2. INTERNAL BLOCK

(1) Execution unit

Executes integer operation instructions. This unit is provided with the 64-bit register file and data path.

(2) Coprocessor 0 (CP0)

This coprocessor is provided with the following:

- Exception processing unit
- Memory management unit
  The memory management unit converts virtual addresses into physical addresses and checks memory access between different memory segments (kernel, supervisor, and user).

TLB (translation lookaside buffer) converts virtual addresses into physical addresses.
The VR4100 supports five types of page size, 1K byte, 4K bytes, 16K bytes, 64K bytes and 256K bytes, with VSIZE (virtual address) = 40 and PSIZE = (physical address) = 32. TLB has 32 entries. Each entry is mapped to an even/odd page of a page frame number.
The exception processing unit is provided with system control coprocessor registers. For the data format of each register, refer to **3.4 SYSTEM CONTROL COPROCESSOR (CP0)**.

(3) Pipeline control

The pipeline is controlled and appropriate processing is executed in the following cases:

- Occurrence of cache miss
- Multi-cycle instruction
- Occurrence of system exception, etc.

(4) Instruction address

The execution address of the next instruction to be fetched is calculated.
For this purpose, the following units are provided:

- PC incrementer
- Branch address adder
- Conditional branch address selector

(5) Instruction cache

The instruction cache employs the following methods:

- Direct map
- Virtual index address
- Physical tag cache

The capacity of the instruction cache is 2K bytes. Each cache line consists of 4-word data and its word parity, 22-bit tag and its parity bit, and a valid bit.
The cache data interface is 64 bits wide.

(6) Data cache

The following methods are employed:

- Direct map
- Virtual index address
- Physical tag, write back cache

The capacity of the data cache is 1K byte.
Each cache line consists of 4-word data and its byte parity, 22-bit tag and its parity bit, valid bit, write back bit and its parity bit.
Two cache lines correspond to one physical line.
Data is read from the cache in 1 cycle, and is written to the data cache in 2 cycles.

(7) System interface

This interface enables the processor to access external resources to satisfy internal requests.
The system interface consists of a 32-bit multiplexed address/data bus, clock signal, interrupt, and control signal.

(8) Clock generator

Generates the following four clocks from the input clock (MasterClock):

- PClock : Pipeline clock (internal)
- MasterOut : Clock output to external agent in synchronization with MasterClock
- SClock : System interface clock (internal). Two frequencies can be selected at cold reset
- TClock : Output clock as reference for external agent. Same frequency as SClock

The clocks internally generated by the processor can be temporarily stopped by a power mode.
The VR4100 uses a phase locked loop (PLL) to suppress skew between the input clock and internal clock.

3. INTERNAL ARCHITECTURE

3.1 POWER MODES

The VR4100 supports three types of power modes, in addition to Fullspeed mode, to reduce the power dissipation:

- Fullspeed mode : Normal operation mode. All the clocks operate.
- Standby mode : The internal clocks except that related to the timer/interrupt are stopped.
- Suspend mode : The internal clocks except that related to the timer/interrupt, and TClock are stopped.
- Hibernate mode : All the clocks generated by the processor are stopped.

To change the mode from Fullspeed, a dedicated instruction is used. To change the mode from Standby or Suspend to Fullspeed, an interrupt is generated. To change from Hibernate to Fullspeed, execute a cold reset.

The mode cannot be directly changed from the Standby, Suspend, and Hibernate modes to a mode other than Fullspeed. Be sure to change the mode from these three modes to Fullspeed, and then change from Fullspeed to the desired operation mode.

Figure 3-1 and Table 3-1 show how the modes are changed. Table 3-2 shows the internal status of the processor in each mode.

Figure 3-1. Power Mode Status Transitions

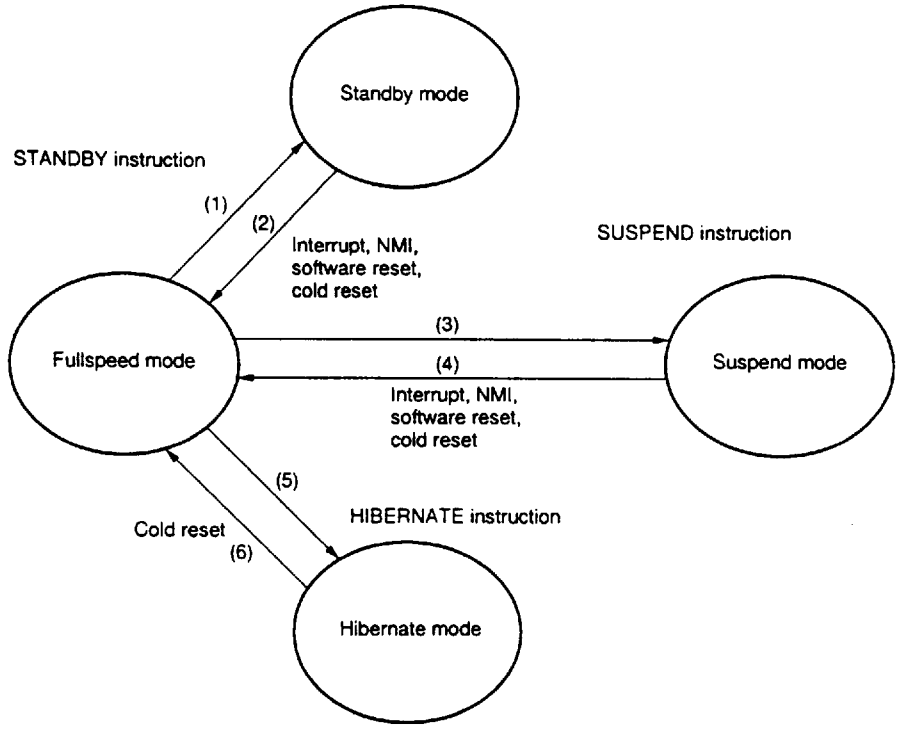


Table 3-1. Changing to Power Modes

No. in Figure 3-1	How Changed	Maximum Time (where PClock is 33 MHz)
(1)	STANDBY instruction	Time until SysAD bus idle + 4 master clock cycles
(2)	Interrupt, NMI, software reset	4 master clock cycles
	Cold reset	PLL lock time[Note] + 16 master clock cycles
(3)	SUSPEND instruction	Time until SysAD bus idle + master clock cycle
(4)	Interrupt, NMI, software reset	4 master clock cycles
	Cold reset	PLL lock time[Note] + 16 master clock cycles
(5)	HIBERNATE instruction	Time until SysAD bus idle + 4 master clock cycles
(6)	Cold reset	PLL lock time[Note] + 16 master clock cycles

Note PLL lock time is approximately 8 ms.

Table 3-2. Processor Status in Power Modes

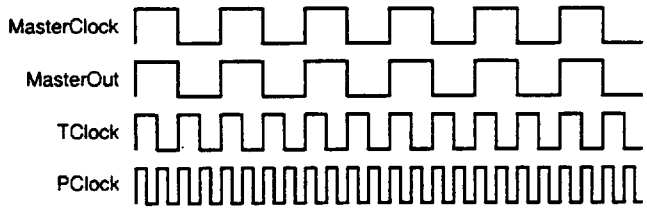
Item / Mode	PLL	MasterOut	Timer/interrupt	TClock	PClock
Fullspeed	ON	ON	ON	ON	ON
Standby	ON	ON	ON	ON	OFF
Suspend	ON	ON	ON	OFF	OFF
Hibernate	OFF	OFF	OFF	OFF	OFF

3.1.1 Fullspeed Mode

This is a normal operation mode and all the internal clocks and functions of the VR4100 can be used in this mode.

Figure 3-2 shows the clock status in the Fullspeed mode.

Figure 3-2. Timing in Fullspeed Mode (when the $\overline{Div2}$ pin is set to low)



3.1.2 Standby Mode

In this mode, all the internal clocks except that related to the timer/interrupt are fixed to high level and stopped.

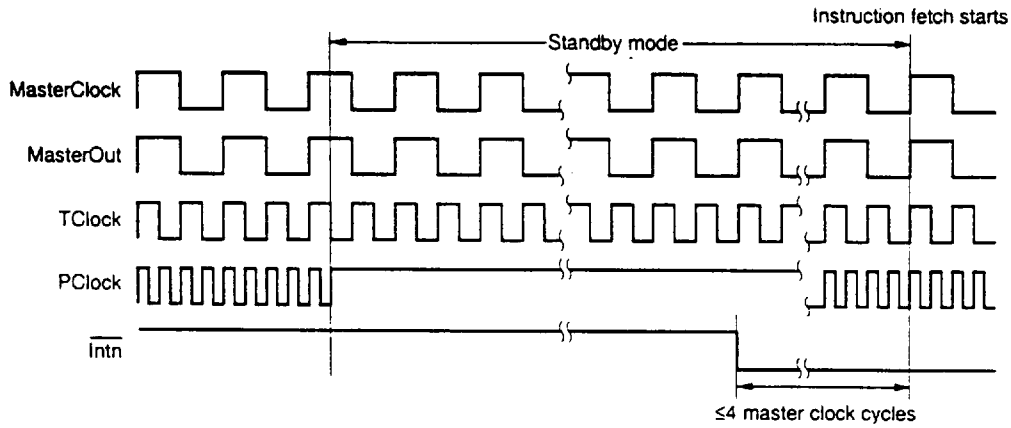
This mode is set by using the STANDBY instruction. If execution of the STANDBY instruction proceeds to the WB stage, the VR4100 waits for the SysAD bus to enter the idle status, then stops the internal clocks, and aborts pipeline operation.

In the Standby mode, the PLL, the clock related to the timer/interrupt, and system interface clocks (TClock and MasterOut) operate normally.

To change the mode from Standby to Fullspeed, either generate an interrupt or NMI or execute a software reset or cold reset.

Figure 3-3 shows the clock status in the Standby mode and the mode changing sequence.

Figure 3-3. Timing in Standby Mode (when the $\overline{\text{Div2}}$ pin is set to low)



3.1.3 Suspend Mode

In this mode, all the internal clocks except that related to the timer/interrupt, and TClock are fixed to the high level and stopped.

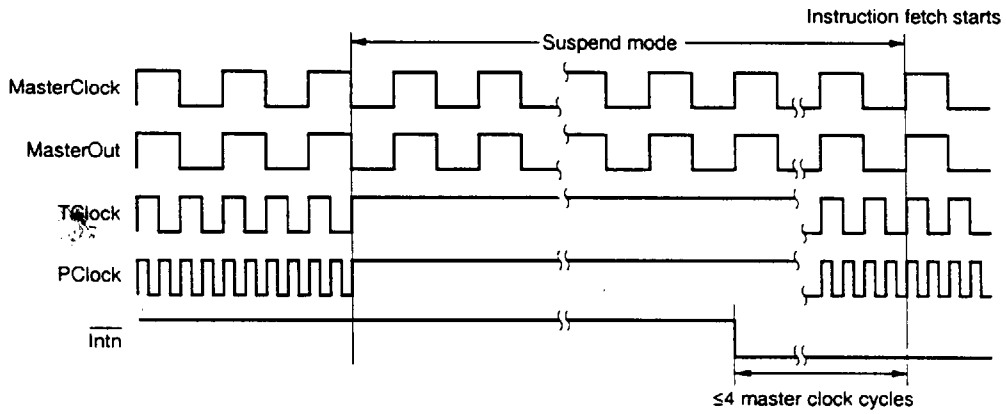
To set this mode, execute the SUSPEND instruction. When execution of the SUSPEND instruction proceeds to the WB stage, the VR4100 waits for the SysAD bus to enter the idle status, then stops the internal clocks and TClock, and aborts pipeline operation.

In the Suspend mode, the clocks related to the timer/interrupt and MasterOut operate normally.

To change the mode from Suspend to Fullspeed, either generate an interrupt or NMI, or execute a software reset or cold reset.

Figure 3-4 shows the clock status in the Suspend mode and the mode changing sequence.

Figure 3-4. Timing in Suspend Mode (when the $\overline{\text{Div2}}$ pin is set to low)



3.1.4 Hibernate Mode

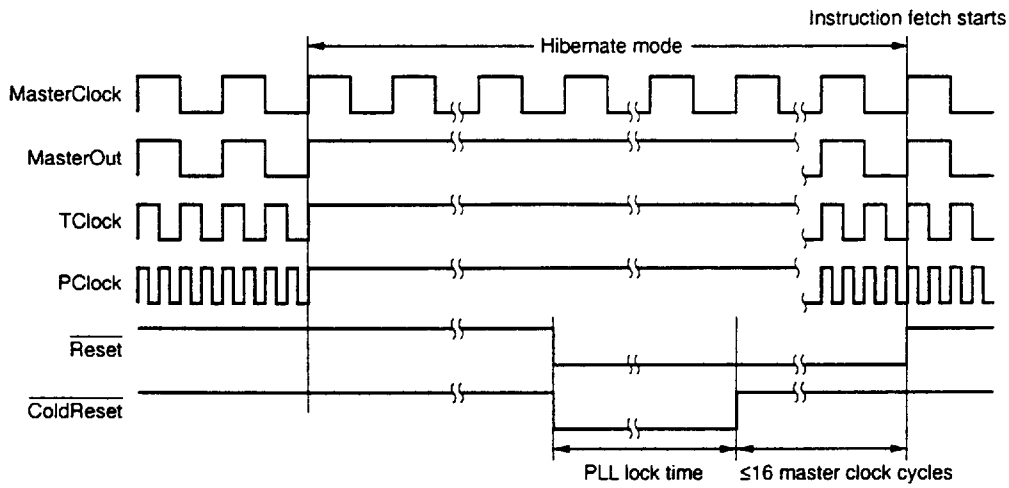
In this mode, all the clocks internally generated by the processor are fixed to the high level and stopped.

To set this mode, execute the HIBERNATE instruction. When execution of the HIBERNATE instruction proceeds to the WB stage, the VR4100 waits for the SysAD bus to enter the idle status, then stops the internal clocks and system interface clocks, and aborts pipeline operation.

To change the mode from Hibernate to Fullspeed, execute a cold reset.

Figure 3-5 shows the clock status in the Hibernate mode and the mode changing sequence.

Figure 3-5. Timing in Hibernate Mode (when the $\overline{Div2}$ pin is set to low)



3.2 PROCESSOR PIPELINE

3.2.1 Pipeline

Each instruction is executed in the following five steps:

(1) IF instruction fetch
(2) RF decode, register fetch, jump/branch
(3) EX execution
(4) DC data cache read
(5) WB write to register file and data cache

The VR4100 is provided with a 5-stage pipeline. Although it takes five clocks to execute each instruction, paralleling is implemented at instruction level. PClock, which is the pipeline clock, operates at a frequency 4 times higher than that of the master clock input.

The figure below shows an operation in the pipeline stage for each instruction type.

Figure 3-6. Operation of Pipeline

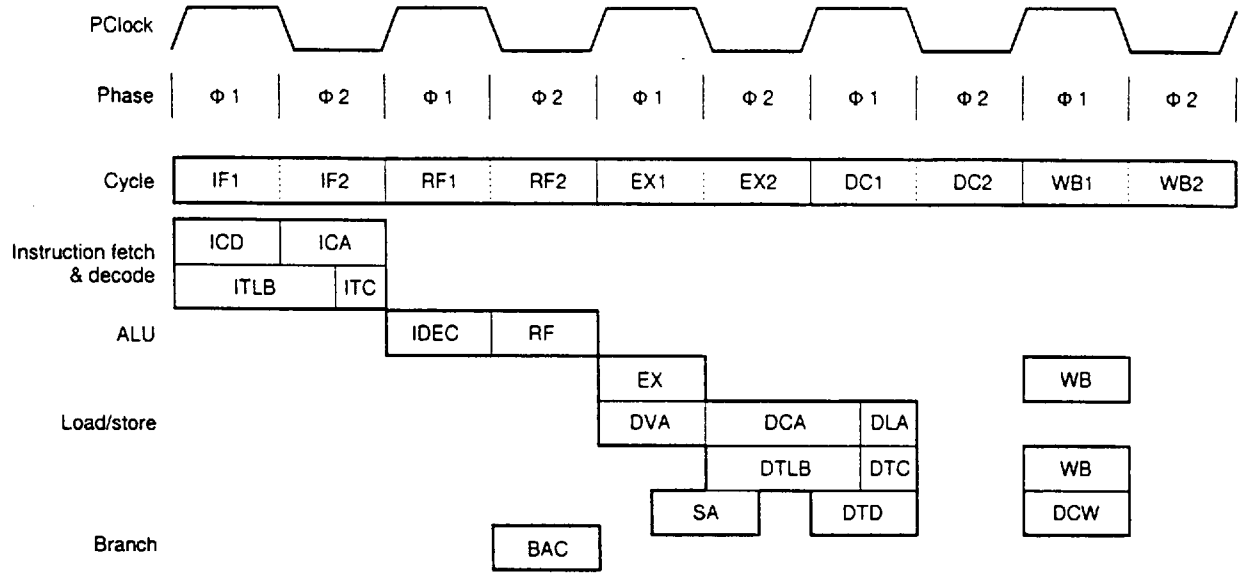


Table 3-5. Pipeline Stages

Cycle	Phase	Mnemonic	Description
IF	Φ1	ICD	Instruction cache address decode
		ITLB	Instruction address conversion
	Φ2	ICA	Instruction cache access
		ITC	Instruction tag check
RF	Φ1	IDEC	Instruction decode
	Φ2	RF	Register operand fetch
		BAC	Branch address calculation
EX	Φ1	EX	Execution stage
		DVA	Data virtual address calculation
		SA	Store align
	Φ2	DCA	Data cache address decode, data cache access
		DTLB	Data address conversion
DC	Φ1	DLA	Data cache load align
		DTC	Data tag check
		DTD	Data transfer to data cache
WB	Φ1	DCW	Data cache write
		WB	Write back to register file

Figure 3-7. Correspondence between Interlock and Exception Condition, and Pipeline Stage

Status \ Stage		IF	RF	EX	DC	WB
Interlock	Stall	–	ITM ICM	–	DTM DCM DCB	–
	Slip	–	LDI MDI SLI CP0	–	–	–
Exception		IAErr	NMI ITLB IPErr INTr IBE SYSC BP CUn RSVD	Trap OVF DAErr	Reset DTLB TMod DPErr WAT DBE	–

3.2.2 Interlock and Exception

The VR4100 performs pipeline processing in the same sequence as the instruction flow when an instruction is fetched, executed, and completed. When an interlock condition and an exception occur, however, the flow of the pipeline is stopped.

Of the interlocks, a stall stops all stages of the pipeline. A sleep causes some stages of the pipeline to stop.

When an exception occurs, execution of the instructions related to the exception, and the instructions that follow are stopped.

The interlock condition and exception in a predetermined pipeline stage are shown below.

Table 3-4. Pipeline Interlocks and Exceptions

Cycle	Interlock		Exception	
IF	–	–	IAErr	Instruction address error exception
RF	ITM	Instruction TLB miss	NMI	External NMI exception
	ICM	Instruction cache miss	SYSC	SYSCALL instruction exception
	LDI	Load data interlock	BP	Breakpoint instruction exception
	MDI	MD busy interlock	RSVD	Reserved instruction exception
	SLI	Store/load interlock	CUn	Coprocessor unusable exception
	CP0	Coprocessor 0 interlock	ITLB	Instruction TLB exception
			IBE	Instruction bus error exception
			IPErr	Instruction parity error exception
			INTr	External interrupt exception
EX	–	–	OVF	Integer overflow exception
			DAErr	Data address error exception
			Trap	Trap instruction exception
DC	DTM	Data TLB miss	Reset	Reset exception
	DCM	Data cache miss	DTLB	Data TLB exception
	DCB	Data cache busy	TMod	Data TLB change exception
			WAT	Watch exception
			DPErr	Data parity error exception
			DBE	Data bus error exception

3.3 EXECUTION UNIT

3.3.1 Overview of Register File

The register file of the VR4100 is provided with thirty-two 64-bit general-purpose registers (GPRs) conforming to MIPS-III. However, because the VR4100 does not support floating-point operation instructions, the thirty-two floating-point general-purpose registers (FGRs), which are provided in the VR4200™ and VR4400™, are not provided in the VR4100.

Of the thirty-two GPRs of the VR4100, the following two registers have special meanings:

- Register r0 : The contents of this register are always 0. To discard the result of an operation, this register can be coded as the target of the instruction. If the value 0 is needed, this register can be used as a source register.
- Register r31 : This is a link register for the JAL and JALR instructions. Therefore, do not use this register with any other instructions.

In addition to the GPRs, the following three special registers are provided:

- 64-bit program counter PC
- Higher bits of 64-bit multiplication/division register, HI
- Lower bits of 64-bit multiplication/division register, LOW

The VR4100 has operating modes that can be selected by the user. Of these modes, two (user and supervisor modes) affect access to the register file.

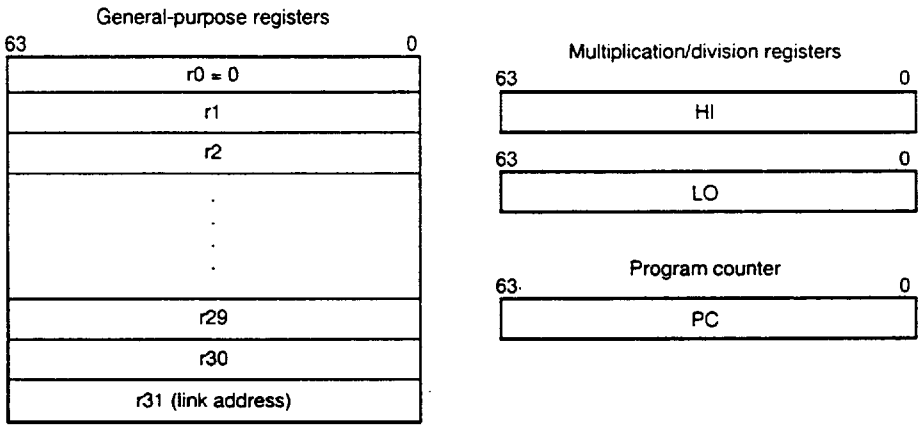
In the user mode, for example, a 32-bit and 64-bit operating modes can be selected by the UX bit in the status

register. When the 32-bit operating mode is selected, the sign-extended value of bit 31 is loaded to the higher 32 bits of the GPRs.

In the kernel mode, 64-bit data can always be operated on even if the 32-bit operating mode is selected by the KX bit in the status register.

Remark The load link bit (LL bit) used for the multi-processor system synchronization instructions (LL and SC) supported by the VR4200 and VR4400 is not provided in the VR4100 (refer to **3.3.2 (2) Deletion of multi-processor instructions**).

Figure 3-8. CPU Registers



3.3.2 Overview of Instruction Set

The instruction set of the VR4100 conforms to the MIPS-I, -II, and -III instruction sets. However, it differs from the instructions set of the other processors in the VR series in the following four points:

(1) Deletion of floating-point (FPU) instructions

Because the VR4100 does not have a floating-point unit, it does not support FPU instructions. If an FPU instruction is executed, therefore, a reserved instruction exception occurs. If it is necessary to use an FPU instruction, emulate it by software in an exception handler.

(2) Deletion of multi-processor instructions

The VR4100 does not support the operating environment of a multi-processor system. If a synchronization support instruction (LL or SC instruction) defined by MIPS-II and -III ISA is executed, therefore, a not-provided instruction exception occurs. For the above reason, the VR4100 is not provided with a load link bit (LL bit).

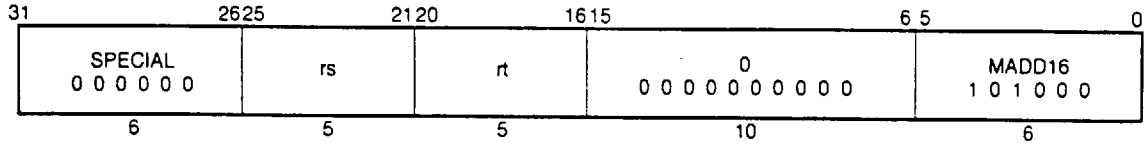
The VR4100 executes all the load/store instructions in the sequence specified in the program. Therefore, the SYNC instruction is treated as a NOP instruction.

(3) Addition of sum-of-products instructions

The VR4100 has a dedicated multiplication core in the CPU and integer sum-of-products instructions to increase the speed of sum-of-products operations. These instructions are not correctly executed by the other processors in the VR series.

The details of the sum-of-products instructions are given below.

(a) MADD16 Multiply and Add 16-bit Integer



Instruction format MADD16 rs, rt

Remark Multiplies the contents of general-purpose register rs by the contents of general-purpose register rt. Both the operands are treated as 2's complements. The bits 62 through 15 of both the operands must be sign-extended values. If the sign is not correctly extended, the result is undefined.
The result of the multiplication is added to the 64-bit value of the special registers HI and LO.
An integer overflow exception does not occur under any circumstances.
The lower word of the 64-bit result is loaded to special register LO, and the higher word is loaded to special register HI.
The hazard cycle that is generated between the MADD16 instruction and other instructions is shown below.

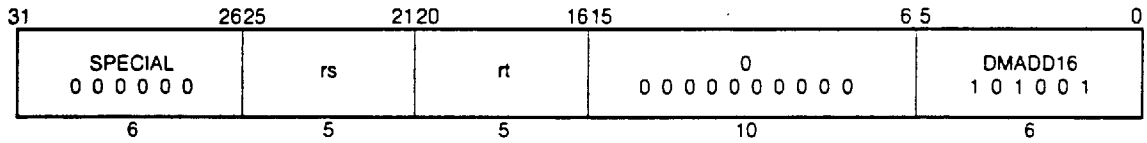
Instruction	Number of Cycles
MULT	1
DMULT	4
DIV	36
DDIV	68

Operation

```
T :  temp1 ← GPR [rs]  *GPR [rt]
     temp2 ← temp1 + (HI31..0 || LO31..0)
     LO ← (temp231) 32 || temp231..0
     HI ← (temp263) 32 || temp263..32
```

Exceptions None

(b) DMADD16 Doubleword Multiply and Add 16-bit Integer



Instruction format MADD16 rs, rt

Remark Multiplies the contents of general-purpose register rs by the contents of general-purpose register rt. Both the operands are treated as 2's complements. The bits 62 through 15 of both the operands must be sign-extended values. If the sign is not correctly extended, the result is undefined.
The result of the multiplication is added to the value of the special register HI. The result of the addition is treated as a signed integer.
An integer overflow exception does not occur under any circumstances.
The 64-bit result is loaded to the special register LO.
This operation is defined in the 64-bit mode and 32-bit kernel mode. If this instruction is executed in the 32-bit user or supervisor mode, a reserved instruction exception occurs.
The hazard cycle that is generated between the DMADD16 instruction and other instructions is shown below.

Instruction	Number of Cycles
MULT	1
DMULT	4
DIV	36
DDIV	68

Operation

```
T :   temp ← GPR [rs]  *GPR [rt]
      temp ← temp + LO
      LO ← temp
```

Exceptions Reserved instruction exception (in 32-bit user or supervisor mode)

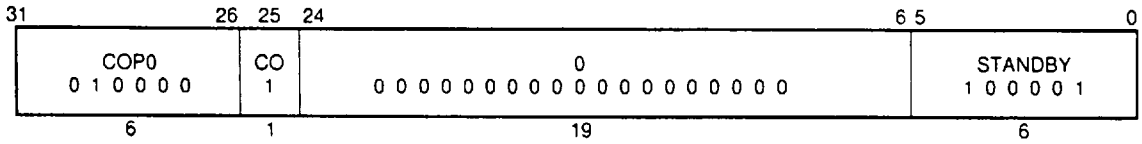
(4) Additional instructions for power modes

The VR4100 supports three power modes to reduce the power dissipation, and has dedicated instructions to set these modes.

Note that these dedicated instructions are not correctly executed by the other processors in the VR series.

Here are the details of the instructions for the power modes:

(a) STANDBY Standby



Instruction format STANDBY

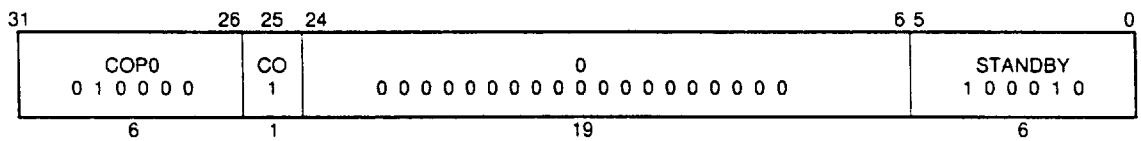
Remark Changes the mode of the processor from Fullspeed to Standby.
When execution of the instruction proceeds to the WB stage, the processor waits for the SysAD bus to enter the idle status, then fixes the internal clock to the high level and stops pipeline operation.
In the Standby mode, the PLL, clock related to the timer/interrupt, and system interface clocks (TClock and MasterOut) operate normally.
To change the mode from Standby to Fullspeed, either generate an interrupt or NMI, or execute a software reset or cold reset.

Operation

```
T :
T + 1 :  Standby Operation ()
```

Exceptions None

(b) SUSPEND Suspend



Instruction format SUSPEND

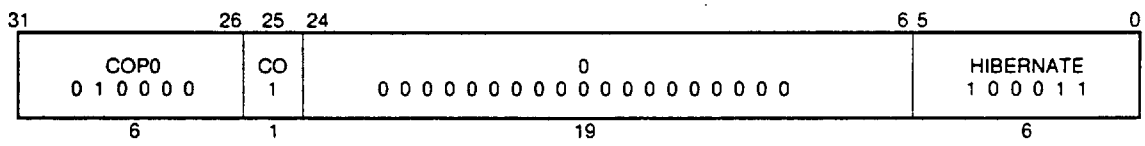
Remark Changes the mode of the processor from Fullspeed to Suspend.
When execution of the instruction proceeds to the WB stage, the processor waits for the SysAD bus to enter the idle status, then fixes the internal clock and TClock to the high level and stops pipeline operation.
In the Suspend mode, the PLL, clock related to the timer/interrupt, and MasterOut operate normally.
To change the mode from Suspend to Fullspeed, either generate an interrupt or NMI, or execute a software reset or cold reset.

Operation

```
T :
T + 1 : Suspend Operation ()
```

Exceptions None

(c) HIBERNATE Hibernate



Instruction format HIBERNATE

Remark Changes the mode of the processor from Fullspeed to Hibernate.
When execution of the instruction proceeds to the WB stage, the processor waits for the SysAD bus to enter the idle status, then fixes the all the clocks to the high level and stops pipeline operation.
To change the mode from Hibernate to Fullspeed, execute a cold reset.

Operation

```
T :
T + 1 : Hibernate Operation ()
```

Exceptions None

3.3.3 Instruction Execution Times

The VR4100 in principle executes one instruction in one cycle.

Some instructions, however, requires two or more cycles to execute.

(1) Data loaded by the load instruction cannot be used in the delay slot. If an instruction that uses the loaded data is located in the delay slot, the pipeline is stalled.

If a store instruction is executed followed by a load instruction or another store instruction, the pipeline is stalled for the duration of the delay slot.

If the condition of a branch instruction is satisfied or if a jump instruction is executed, the instruction at the destination address is executed after the delay slot.

Table 3-5. Number of Delay Slot Cycles

Instruction Category	Required Number of Cycles (PCycle)
Load	1
Store	1
Jump	1
Branch	1

(2) To execute an integer multiplication, division, or sum-of-products instruction, the number of cycles shown in the table below is necessary.

These instructions can be executed in parallel with instructions other than those that access the HI/LO register which stores execution results, and multiplication, division, and sum-of-products instructions.

Table 3-6. Number of Execution Cycles for Integer Multiplication and Division Instructions

Instruction Category	Required Number of Cycles (PCycle)
MULT	1
MULTU	1
DIV	35
DIVU	35
DMULT	4
DMULTU	4
DDIV	67
DDIVU	67
MADD16	1
DMADD16	1

3.4 SYSTEM CONTROL COPROCESSOR (CP0)

The system control coprocessor (CP0) operates as a part of the CPU. The CP0 supports memory management, address conversion, exception processing, and privileged operation. CP0 has the registers shown in Table 3-7 and a 32-entry TLB.

Although the basic configuration of the CP0 register of the VR4100 is the same as that of the VR4200 and VR4400, the bit configuration and the settings differ because the number of entries of the TLB, page size, cache size, physical address space, and system interface differ. For details, refer to the description of each register and **5. DIFFERENCES WITH THE VR4200 AND VR4400.**

3.4.1 System Control Coprocessor Registers

All the CP0 registers of the VR4100 that can be used are listed below. Writing or reading an unused register (RFU) is undefined. In the 32-bit mode, the higher 32 bits of the 64-bit registers are masked.

Table 3-7. CP0 Registers

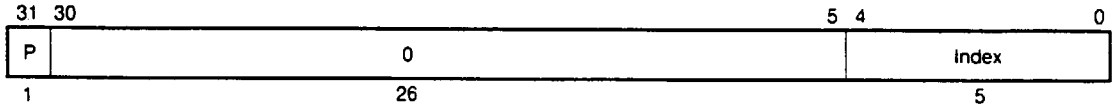
Number	Register	Description
0	Index	Programmable pointer to TLB array
1	Random	Random pointer to TLB array (read-only)
2	Entry Lo0	Second half of TLB entry for even VPN
3	Entry Lo1	Second half of TLB entry for odd VPN
4	Context	Pointer to virtual PTE table of kernel in 32-bit mode
5	Page mask	Specifies page size
6	Wired	Number of wired TLB entries
7	–	RFU (Reserved for Future Use)
8	BadVAddr	Displays virtual address at which error last occurred
9	Count	Timer count
10	Entry Hi	First half of TLB entry (includes ASID)
11	Compare	Timer comparison value
12	Status	Sets operation status
13	Cause	Displays cause of last exception
14	EPC	Exception program counter
15	PRId	Processor revision ID
16	Config	Sets memory system mode
17	LLAddr	Displays address of LL instruction
18	Watch Lo	Lower bits of memory reference trap address
19	Watch Hi	Higher bits of memory reference trap address
20	X context	Pointer to virtual PTE table of kernel in 64-bit mode
21 to 25	–	RFU
26	Parity error	Parity bit of cache
27	Cache error	Cache error and status register
28	Tag Lo	Cache register, low
29	Tag Hi	Cache register, high (reserved register)
30	Error EPC	Error exception program counter
31	–	RFU

3.4.2 Index Register (0)

The index register is a 32-bit read/write register. Five bits of this register are used as an index to the TLB entry. The most significant bit of this register indicates success or failure of the TLB probe (TLBP) instruction.

This register indicates the TLB entry subject to the TLB read (TLBR) or TLB write index (TLBWI) instruction.

Figure 3-9. Index Register



P : Success or failure of probe. If execution of the last TLBP instruction fails, 1 is set to this bit. If the instruction execution is successful, this bit is cleared to 0.

Index : This field specifies the index to the TLB entry subject to the TLBR and TLBWI instructions.

0 : RFU. Write 0 to this field. This field is 0 when read.

3.4.3 Random Register (1)

The random register is a read-only register. The lower 5 bits of this register are used to reference the TLB entry.

The contents of this register are decremented each time an instruction has been executed. The range of the value of this register is as follows:

- The lower limit is indicated by the contents of the wired register.
- The upper limit is 31.

The random register indicates the TLB entry subject to the TLBWR instruction. It can be read to check the operation of the processor.

The random register is set to the upper-limit value at cold reset. It is also set to the upper-limit value when data is written to the wired register.

Figure 3-10. Random Register

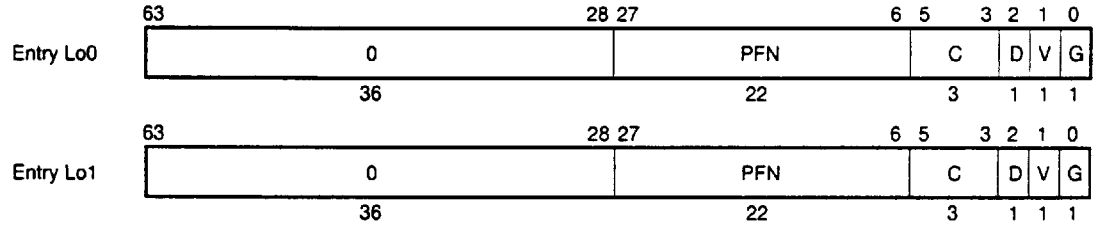
Random : TLB random index

0 : RFU. Write 0 to this field. When it is read, 0 is returned.

3.4.4 Entry Lo0 (2) and Entry Lo1 (3) Registers

Entry Lo consists of two registers: entry Lo0 for even virtual pages, and entry Lo1 for odd virtual pages. Both the entry Lo0 and entry Lo1 registers are read/write registers and are used to access the lower bits of the internal TLB. When a TLB read/write operation is executed, the entry Lo0 and entry Lo1 registers read/write the contents of the lower side of the TLB entry at even and odd page numbers, respectively.

Figure 3-11. Entry Lo0 and Entry Lo1 Registers



PFN : Page frame number. Higher bits of physical address

C : Specifies the page attribute of TLB.
- 010 : Page does not use the cache.
- Others : Page uses the cache.

D : Dirty. When this bit is set to 1, page is marked "Dirty" and is enabled to be written. Actually, this bit is used by software to prevent data from being changed, as a "write protection" bit.

V : Valid. When this bit is set to 1, the TLB entry is valid. If this entry is hit without the V bit set, TLB invalid exception (TLBL or TLBS) occurs.

G : Global. If the global bits of both the entry Lo0 and entry Lo1 registers are set, ASID is ignored when TLB is referenced.

0 : RFU. Write 0 to this field. When it is read, 0 is returned.

3.4.5 Context Register (4)

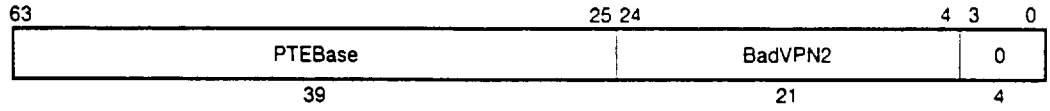
The context register is a read/write register and indicates a pointer to a page table entry (PTE) on memory. This is used by the TLB unmatch exception handler when loading the contents of a new TLB entry.

The BadVPN2 field of this register can only be read. This field stores the bits 31 through 11 of the virtual address that has caused a TLB miss. Because a TLB entry consists of a pair of an odd page and an even page, it does not include bit 10.

If the page size is 1K bytes, the value of this field can be directly used as an address. If the page size is other than 1K bytes, shift and mask the value of this field.

The PTEBase field is a write-only field that can be written only via software. This field indicates the address of the PTE table currently on memory.

Figure 3-12. Context Register



PTEBase : Base address of page entry table

BadVPN2 : Value of virtual address at which translation is invalidated ÷ 2 (bits 31 through 11 of virtual address)

0 : RFU. Write 0 to this field. When it is read, 0 is returned.

3.4.6 Page Mask Register (5)

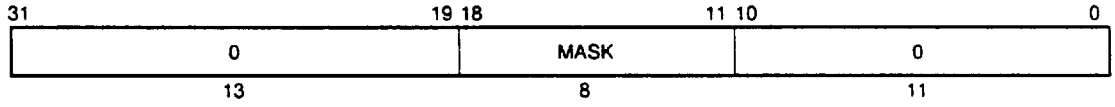
The page mask register is a read/write register that is used to read or write the TLB. When compare mask data is registered to this register, five types of page size of each TLB entry can be set as shown in Table 3-8. To read or write the TLB, this register is used as the destination or source. At the time of address conversion, bit 18 through bit 11 of the item being compared are masked.

Table 3-8 shows the mask pattern of each page size. If the mask pattern is other than those shown, the operation of the TLB is undefined.

Table 3-8. Mask Value and Page Size

Page Size	Bit							
	18	17	16	15	14	13	12	11
1K bytes	0	0	0	0	0	0	0	0
4K bytes	0	0	0	0	0	0	1	1
16K bytes	0	0	0	0	1	1	1	1
64K bytes	0	0	1	1	1	1	1	1
256K bytes	1	1	1	1	1	1	1	1

Figure 3-13. Page Mask Register



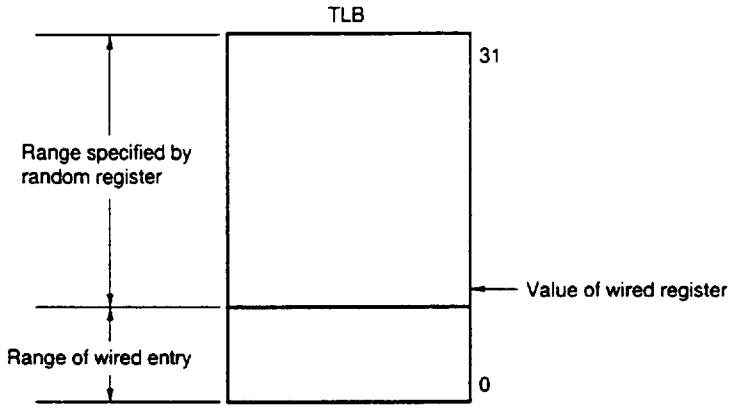
MASK : Page compare mask. Determines the virtual page size of the corresponding entry.

0 : RFU. Write 0 to this field. When it is read, 0 is returned.

3.4.7 Wired Register (6)

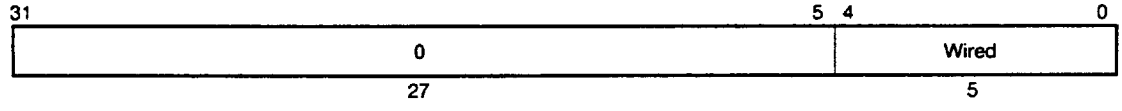
The wired register is a read/write register and indicates the lower-limit value of the random entry of the TLB (*refer to* **Figure 3-14**). The wired entry cannot be updated by the TLBWR instruction, but can be updated by the TLBWI instruction. The random entry can be updated by both the instructions.

Figure 3-14. Position Indicated by Wired Register



The wired register is cleared to 0 at cold reset. When data is written to this register, the random register is set to the upper-limit value (refer to **3.4.3 Random Register (1)**).

Figure 3-15. Wired Register



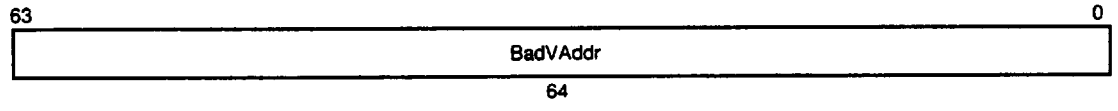
Wired : Specifies the TLB wired boundary.
0 : RFU. Write 0 to this field. When it is read, 0 is returned.

3.4.8 BadVAddr Register (8)

The BadVAddr (Bad Virtual Address) register is a read-only register and holds a virtual address that has performed invalid address translation last, or a virtual address at which an addressing error has occurred.

Caution No information is stored to this register even if a bus error exception has occurred, because it is not an address error exception.

Figure 3-16. BadVAddr Register



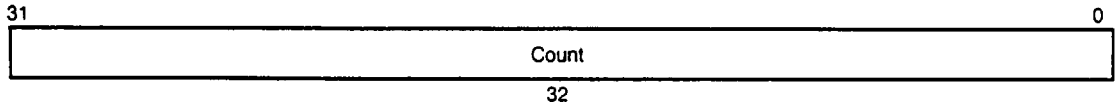
BadVAddr : Virtual address at which an address translation has failed, or at which an addressing error has occurred last.

3.4.9 Count Register (9)

The count register is a read/write register and operates as a timer. The contents of this register are incremented in synchronization with a frequency that is 1/2 that of PClock, regardless of instruction execution.

This register is the free running type. When all the bits of this register are set to 1, they are reset to 0 when the next event occurs, and then the value of the register is incremented again. This register is used for self-diagnosis.

Figure 3-17. Count Register

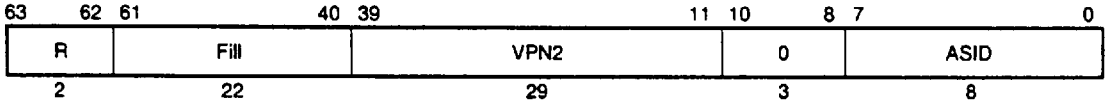


Count: The latest count value. Compares the contents of compare register.

3.4.10 Entry Hi Register (10)

The entry Hi register is a read/write register and is used to access the higher bits of the internal TLB. When TLB read/write operations are executed, the higher bits of the TLB entry are read/written.

Figure 3-18. Entry Hi Register



VPN2 : Value of virtual page number divided by 2 (mapped to two pages).

ASID : Address space ID field. This 8-bit field can use the TLB for multiple processes. The virtual address of each process may overlap with that of the other.

R : Space identifier (00 → user, 01 → supervisor, 11 → kernel). Coincides with the bits 63 and 62 of a virtual address.

Fill : RFU. Write 0 or –1 to this field (sign-extends a virtual page number).

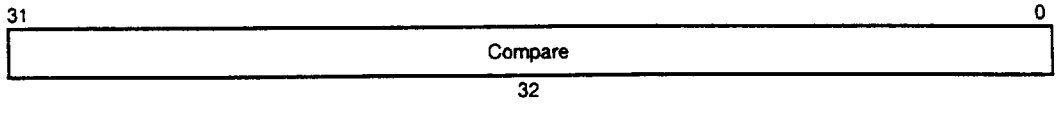
0 : Write 0 to this field. When it is read, 0 is returned.

3.4.11 Compare Register (11)

The compare register generates a timer interrupt. Although it holds a value, it cannot change the value by itself. When the value of the compare register is equal to the value of the count register (refer to **3.4.9**), the IP7 bit of the cause register is set. When the IP7 bit is set, an interrupt occurs in the DF stage next to the stage at which the interrupt has been enabled. When a value is written to the compare register, the timer interrupt request is cleared.

The compare register can be read or written because it is used for self-diagnosis. Usually, however, it is used as a write-only register.

Figure 3-19. Compare Register

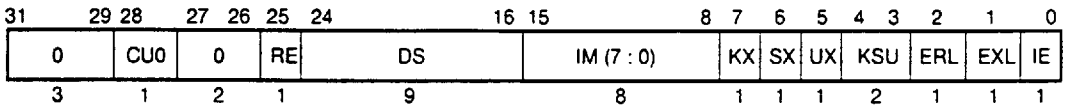


Compare: Value to be compared with the contents of the count register.

3.4.12 Status Register (12)

The status register is a read-write register and holds status information such as on the operating mode, enabling of interrupt, and self-diagnosis of the processor. At reset, the TS and RP bits are reset to 0, and ERL and BEV bits are set to 1. The SR bit is reset to 0 at cold reset, and is set to 1 at NMI/software reset. The contents other than the above are undefined. The TS, BEV, and SR bits are in the DS field.

Figure 3-20. Status Register



CU0 : Enables use of system control coprocessors (1 → can be used, 0 → cannot be used).
CP0 can be always used in the kernel mode, regardless of the CU0 bit.

RE : Enables endian inversion in the user mode (0 → disables, 1 → inverts).

DS : Self-diagnosis status field (for details, refer to **Figure 3-21**).

IM : Interrupt mask. Enables external, internal, coprocessor, and software interrupt (0 → disables, 1 → enables).
This field consists of 8 bits and controls eight interrupts.
The interrupts are allocated to the respective bits as follows:
IM7 : Masks timer interrupt
IM(6:2) : Mask external normal interrupts $\overline{\text{Int}(4:0)}$ and external write request
IM(1:0) : Mask software interrupts and cause register IP (1:0)

KX : Sets an exception vector that is used in case of a TLB miss in the kernel mode (0 → TLB unmatch exception vector, 1 → XTLB unmatch exception vector).
In the kernel mode, the 64-bit operation is always valid.

SX : Enables 64-bit addressing and 64-bit operation in the supervisor mode (0 → 32 bits, 1 → 64 bits).
When this bit is set, and if a TLB miss occurs in the supervisor mode address space, an XTLB unmatch exception occurs.

UX : Enables 64-bit addressing and 64-bit operation in the user mode (0 → 32 bits, 1 → 64 bits).
When this bit is set, and if a TLB miss occurs in the user mode address space, an XTLB unmatch exception occurs.

KSU : Sets/displays the operating mode (10 → user, 01 → supervisor, 00 → kernel).

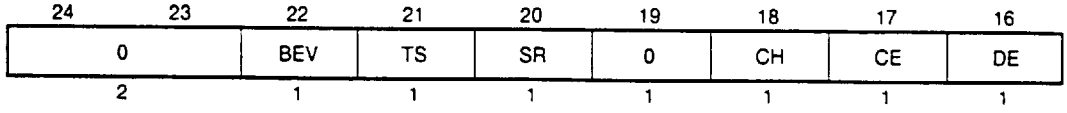
ERL : Sets/displays an error level (0 → normal, 1 → error).

EXL : Sets/displays an exception level (0 → normal, 1 → exception).

IE : Sets/displays enabling of interrupts (0 → disables, 1 → enables).

Figure 3-21 shows the format of the self-diagnosis status (DS) field. All the bits in the DS field, except the TS bit, can be read or written.

Figure 3-21. Self-Diagnosis Status Field



BEV : Specifies the base addresses of the TLB unmatch exception and general-purpose exception vector (0 → normal, 1 → bootstrap).

TS : Generates TLB shutdown (read-only) (0 → does not generate, 1 → generates).
This bit is used to avoid adverse influence on the TLB if two or more TLB entries coincide with the same virtual address. After TLB shutdown, reset and restart the processor. When the TLB entry is invalidated (by clearing the V bit of the entry), TLB shutdown occurs.

SR : Generation of soft reset or NMI (0 → not generated, 1 → generated)

CH : Condition bit of CP0 (0 → false, 1 → true). This bit can be read or written only by software, and cannot be accessed by hardware.

CE : If the CE bit is 1, the contents of the parity error register are used to check the cache or set or change the check bit (for details, refer to **3.4.20**).

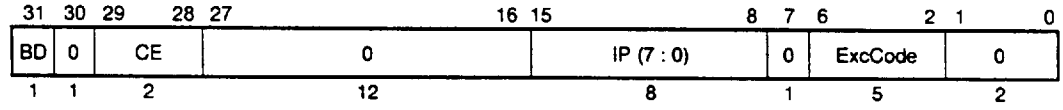
DE : Disables occurrence of the cache error exception (0 → enables, 1 → disables). Invalidates the parity check in the cache and of the system bus.

0 : RFU. Write 0 to this bit. When it is read, 0 is returned.

3.4.13 Cause Register (13)

The cause register is a 32-bit read/write register and holds the cause of the exception that has occurred last. The 5 bits of the exception code field (ExcCode) of this register indicates the cause of the exception (refer to **Table 3-9**). The other fields hold detailed information on specific exceptions. All the bits, except bits IP1 and IP0, are read-only. IP1 and IP0 bits are used to generate a software interrupt.

Figure 3-22. Cause Register



BD : Indicates whether the exception that has occurred last is executed in a branch delay slot (1 → delay slot, 0 → normal).

CE : Indicates the number of the coprocessor in which a coprocessor unusable exception has occurred. If this exception does not occur, the value of this field is undefined.

IP : Indicates a pending interrupt (1 → pending, 0 → no interrupt).
Interrupts are allocated to the respective bits of this field as follows:
IP7 : Timer interrupt
IP (6:2): External normal interrupts $\overline{\text{Int}}$ (4:0) and external write request
IP (1:0): Software interrupts. If only these bits are set to 1 by software, an interrupt exception occurs. After the occurrence of software interrupts, clear the bit to 0.

ExcCode : Exception code field (for details, refer to **Table 3-9**).

0 : RFU. Write 0 to this field. When it is read, 0 is returned.

Table 3-9. Exception Code Field of Cause Register

Exception Code Value	Mnemonic	Description
0	Int	Interrupt exception
1	Mod	TLB change exception
2	TLBL	TLB unmatch exception (load or fetch)
3	TLBS	TLB unmatch exception (store)
4	AdEL	Address error exception (load or instruction fetch)
5	AdES	Address error exception (store)
6	IBE	Bus error exception (instruction fetch)
7	DBE	Bus error exception (load or store of data)
8	Sys	System call exception
9	Bp	Breakpoint exception
10	RI	Reserved instruction exception
11	CpU	Coprocessor unusable exception
12	Ov	Operation overflow exception
13	Tr	Trap exception
14 to 22	–	RFU
23	WATCH	Watch exception
24 to 31	–	RFU

3.4.14 EPC (Exception Program Counter) Register (14)

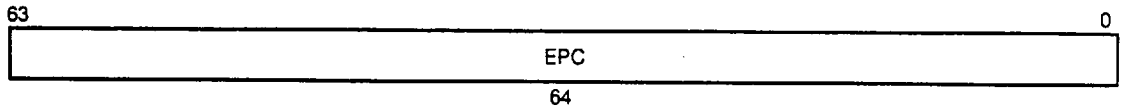
The EPC (exception program counter) register is a read/write register and indicates an address from which program execution is to be resumed after exception processing.

The EPC register indicates either of the following addresses:

- Virtual address of the instruction that has directly caused the exception
- Virtual address of the branch or jump instruction immediately before (if the instruction that has caused the exception is in a branch delay slot, the BD bit of the cause register is set to 1).

If an exception occurs when the EXL bit of the status register is 1, write to the EPC register is stopped. This prevents the processor from overwriting the instruction address stored in the EPC register that has caused the exception when any other exception occurs.

Figure 3-23. EPC Register

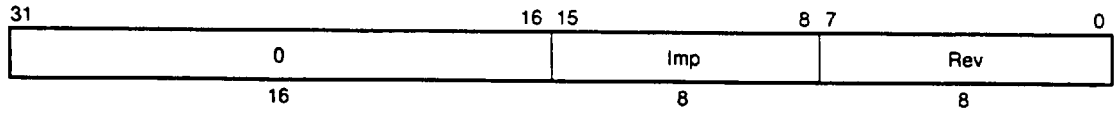


EPC: Address from which program execution is resumed after exception processing

3.4.15 PRId (Processor Revision ID) Register (15)

The PRId (processor revision ID) register is a 32-bit read-only register and indicates the identifiers of the CPU and CP0, and a revision level.

Figure 3-24. PRId Register



Imp: Processor ID number. 0x0C for the VR4100.
Rev: Processor revision number.
0 : RFU. Write 0 to this field. When it is read, 0 is returned.

The processor revision number is a value of yx format. y indicates the major revision number included in bits 7 through 4, and x is the minor revision number included in bits 3 through 0.

The processor revision number identifies the revision of the chip. However, the revision of the chip is not always reflected on the PRId register. Conversely, a change in the revision number does not always reflect the actual change of the chip. Therefore, the user should develop a program that is not dependent upon the processor revision number field.

3.4.16 Config Register (16)

This register displays or sets various statuses of the VR4100.

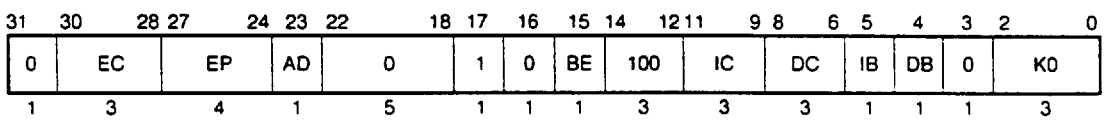
Although consideration is given to compatibility with the config register of the VR4000 and VR4400, the settings of some bits of the config register of the VR4100 are different from those of the VR4000 and VR4400, and some bits of the VR4100's config register are fixed.

The EC and BC fields of the config register are read-only, and are initialized by input from a mode setting pin at cold reset.

The EP, AD, and K0 fields can be read/written via software, and are undefined at cold reset.

Be sure to initialize this register by software before using the cache.

Figure 3-25. Config Register



EC: System interface frequency ratio (read-only). Set by the $\overline{Div2}$ pin at cold reset.
000 → 1/2 PClock
111 → Same as PClock
Others → RFU

EP: Sets the transfer pattern for write back data.
0 → DD : 1 word/1 cycle
1, 2 → RFU
3 → DxDx : 2 words/4 cycles
4, 5 → RFU
6 → DxxDxx : 2 words/6 cycles
7 → RFU
8 → DxxxDxxxx : 2 words/8 cycles
9-15 → RFU

AD: Sets the high-speed data (AD) mode. In the AD mode, data are successively stored in the non-cache area at high speeds (for details, refer to **4.1.4 (12) High-speed data (AD) mode**).
0 → Data mode compatible with VR4000 series
1 → AD mode

BE: BigEndianMem (sets endian)
0 → Little endian
1 → Big endian

IC: Instruction cache size[Note]. 0 is returned when read.

DC: Data cache size[Note]. 0 is returned when read.

Note The MIPS architecture defines the instruction cache size specified by the IC area to be 2^{12+IC} bytes and the data cache size specified by the DC area to be 2^{12+DC} bytes. The cache size of the VR4100, however, does not satisfy neither of these definitions, and therefore, setting of these areas is meaningless.

IB: Instruction cache line size
0 → 16 bytes
1 → RFU

DB: Data cache line size
0 → 16 bytes
1 → RFU

K0: Sets coherency algorithm of Kseg0.
010 → Cache cannot be used.
Others → Cache can be used.

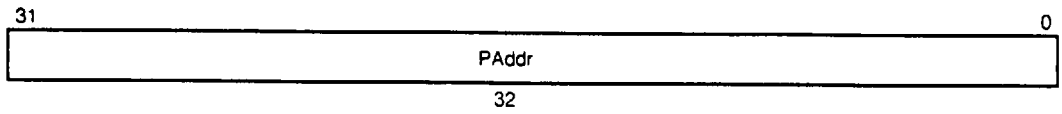
1: 1 is returned when this bit is read.

0: 0 is returned when this bit is read.

3.4.17 LLAddr (Load Linked Address) Register (17)

Although the LLAddr register is a read/write register, it is used for self-diagnosis only. This register of the VR4100 is defined only to maintain compatibility with the VR4000 and VR4400, and is meaningless for normal operation.

Figure 3-26. LLAddr Register

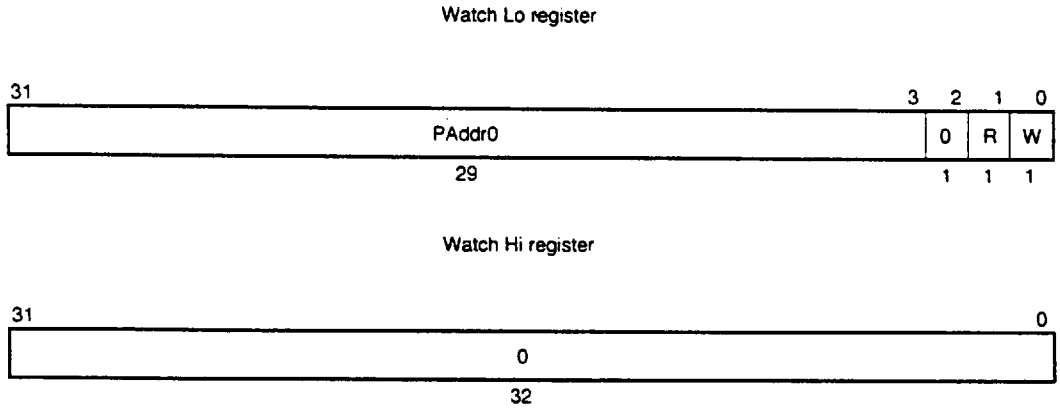


PAddr : 32-bit physical address

3.4.18 Watch Lo (18) and Watch Hi (19) Registers

The VR4100 uses the watch Lo and watch Hi registers to detect a request to reference a physical address. The processor has a debug function to generate a watch exception when the load/store instruction is executed.

Figure 3-27. Watch Lo and Watch Hi Registers



PAddr0 : Specifies bits 31 through 3 of a physical address.
R : When this bit is 1 and when the load instruction is executed, a watch exception occurs.
W : When this bit is 1 and when the store instruction is executed, a watch exception occurs.
0 : RFU. Write 0 to this field. When it is read, 0 is returned.

3.4.19 X Context Register (20)

The X context register is a read/write register and contains a pointer to the page table entry (PTE) in memory. This pointer is used to load the contents of a new TLB entry by the XTLB unmatch exception handler in the 64-bit mode.

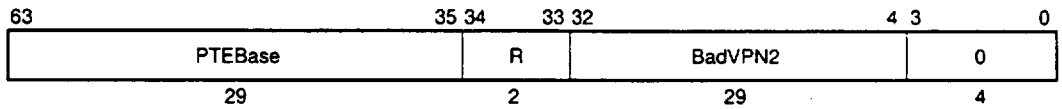
The BadVPN2 area is a read-only area and stores bits 39 through 11 of a virtual address that has caused a TLB miss.

Because a TLB entry consists of an even page and an odd page (i.e., a pair of pages), it does not include bit 10.

If the page size is 1K bytes, the value of the BadVPN2 area can be directly used as an address. If the page size is 256K bytes, shift and mask the value.

The PTEBase area is a read-only area, and used by the operating system. It indicates the address of the current PTE table in memory.

Figure 3-28. X Context Register



PTEBase: Base address of page table entry

R : Space identifier (00 → user, 01 → supervisor, and 11 → kernel). Matches with bits 63 and 62 of virtual address.

0 : RFU. Write 0 to this field. When it is read, 0 is returned.

BadVPN2: Virtual address at which conversion has been invalidated ÷2 (bits 39 through 11 of virtual address)

3.4.20 Parity Error Register (26)

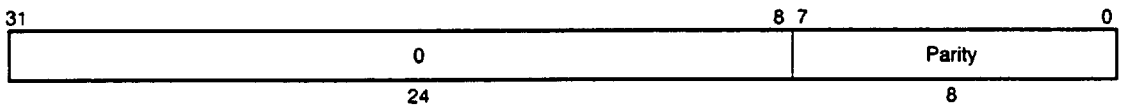
The parity error register is a read/write register. By using the bits of this register, initialization, self-diagnosis, and error processing of the cache are performed.

Data is loaded to the parity error register by the Index_Load_Tag CACHE instruction. The contents of this register are manipulated as follows:

(1) If the CE bit of the status register is set, the contents of the parity error register are written to the cache instead of the parity by the store instruction.

(2) If the Fill CACHE instruction is executed, the contents of the parity error register are used, instead of calculating the instruction parity.

Figure 3-29. Parity Error Register



Parity : 8-bit area that specifies the parity bit to be read from or written to the cache

0 : RFU. Write 0 to this bit. 0 is returned when it is read.

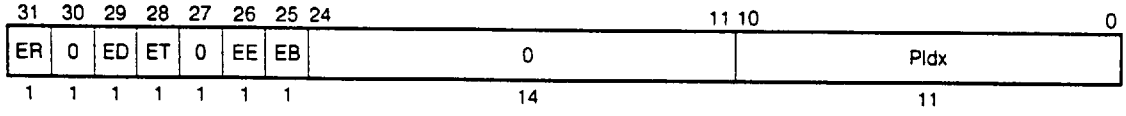
3.4.21 Cache Error Register (27)

The cache error register is a 32-bit read-only register and indicates the parity error status of the cache. A parity error cannot be corrected by hardware.

The cache error register has cache index bits that indicate the cause and nature of the error that has occurred, and status bits.

This register is read if a cache error exception occurs.

Figure 3-30. Cache Error Register



ER : Type of cache (0 → instruction, 1 → data)

ED : Indicates whether a data area error occurred (0 → error did not occur, 1 → error occurred)

ET : Indicates whether a tag area error occurred (0 → error did not occur, 1 → error occurred)

EE : Indicates whether an error occurred on the SysAD bus (0 → error did not occur, 1 → error occurred)

EB : Set to 1 if in addition to an instruction error (indicated by the ER bit), a data error occurred. When this bit is set, it indicates that the data cache requires flashing after processing of an instruction error.

PIdx : Indicates the index of the cache where an error occurs.

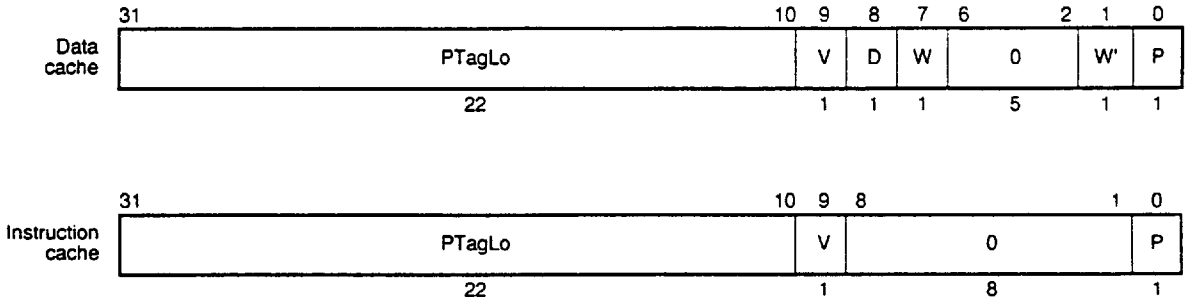
0 : RFU. Write 0 to this bit. 0 is returned when it is read.

3.4.22 Tag Lo (28) and Tag Hi (29) Registers

The Tag Lo and Tag Hi registers are 32-bit read/write registers and hold the cache tag of the cache and parity for initialization, self-diagnosis, and error processing of the cache.

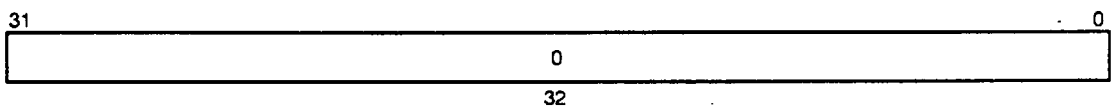
The P area of these registers is not affected when Index Store Tag of the CACHE instruction is executed. Parity is generated during store operation.

Figure 3-31. Tag Lo Register



PTagLo : Bits 31 through 10 of physical address
V : Valid bit
D : Dirty bit. Set when the cache is dirty.
W : Write back bit. Set when data is written to the cache line.
W' : Even parity bit for the write back bit.
P : Even parity bit for the cache tag
0 : RFU. Write 0 to this bit. 0 is returned when it is read.

Figure 3-32. Tag Lo and Hi Registers



0 : RFU. Write 0 to this field. When it is read, 0 is returned.

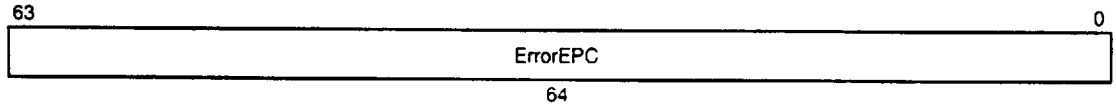
3.4.23 Error EPC Register (30)

The configuration of the error EPC (exception program counter) register is the same as that of the EPC register. However, this register is used for cold reset (exception), soft reset (exception), and storing the contents of the program counter when the NMI exception occurs. The error EPC register is a read/write register and holds a virtual address from which instruction execution is to be resumed after error processing has been completed. This virtual address is either of the following:

- Virtual address of the instruction that has caused the exception
- Virtual address of the branch or jump instruction immediately before if the instruction that has caused the error exception is in the branch delay slot

The error EPC register does not indicate a branch delay slot.

Figure 3-33. Error EPC Register



ErrorEPC: Contains the program re-start address when a parity error occurs, or the program counter at the time of a cold reset exception, soft reset exception, or NMI exception.

3.4.24 System Control Coprocessor (CP0) Instructions

The CP0 instruction executes memory management and exception processing by using the CP0 register. However, the CACHE instruction is executed slightly differently. For the operations of the instructions, refer to **VR4200 User's Manual**.

Table 3-10. System Control Coprocessor (CP0) Instructions

Instruction	Contents
MTC0	Move To Coprocessor0 Loads the contents of a general-purpose register to a specified CP0 register.
MFC0	Move From Coprocessor0 Loads the contents of a CP0 register to a general-purpose register.
DMTC0	Double Move to Coprocessor0 Transfers the 64-bit data of a general-purpose register to a CP0 register 64 bits wide. This instruction is undefined with a CP0 register 32 bits wide.
DMFC0	Double Move From Coprocessor0 Transfers the contents of a CP0 register 64 bits wide to a general-purpose register. This instruction is undefined with a CP0 register 32 bits wide.
ERET	Exception Return Returns from an interrupt, exception, or error trap. Unlike the jump/branch instruction, this instruction does not execute the next instruction. When the processor returns from an error trap, the PC holds the contents of the error EPC register; otherwise, the PC holds the contents of the EPC register.
TLBR	Read Indexed TLB Entry Loads the contents of a TLB entry indicated by the index register to the page mask/entry Hi/entry Lo1/entry Lo0 register.
TLBWI	Write Indexed TLB Entry Loads the contents of the page mask/entry Hi/entry Lo1/entry Lo0 register to a TLB entry indicated by the index register.
TLBWR	Write Random TLB Entry Loads the contents of the page mask/entry Hi/entry Lo1/entry Lo0 register to a TLB entry indicated by the random register.
TLBP	Probe TLB for Matching Entry Loads the address of a TLB entry that matches the contents of the entry Hi register to the index register. If there is no TLB entry that matches the contents of the entry Hi register, the most significant bit of the index register is set.

3.4.25 Virtual Address Space

The VR4100 has two operation modes: 32-bit and 64-bit modes. It also has three operating modes: user, supervisor, and kernel. The figures below show the virtual address spaces in each operating mode.

Figure 3-34. User Mode Address Space

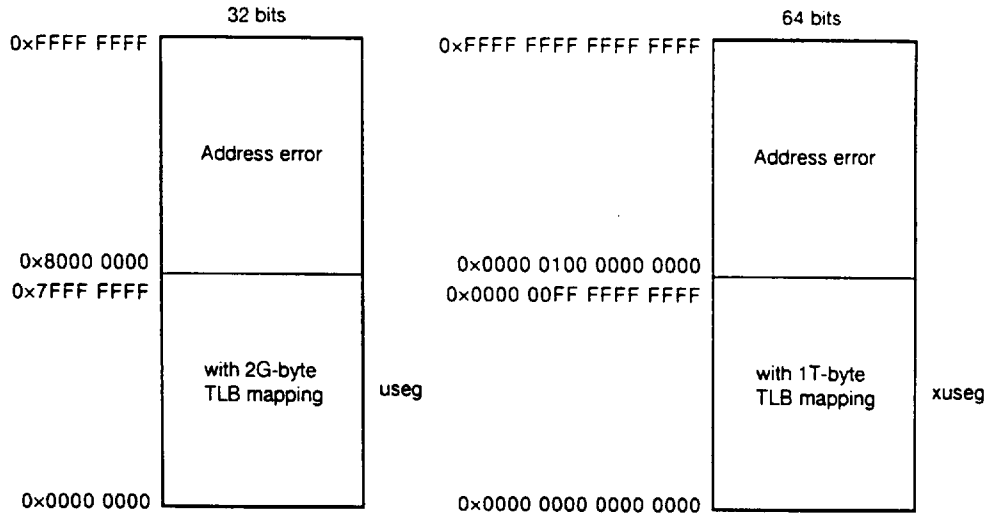
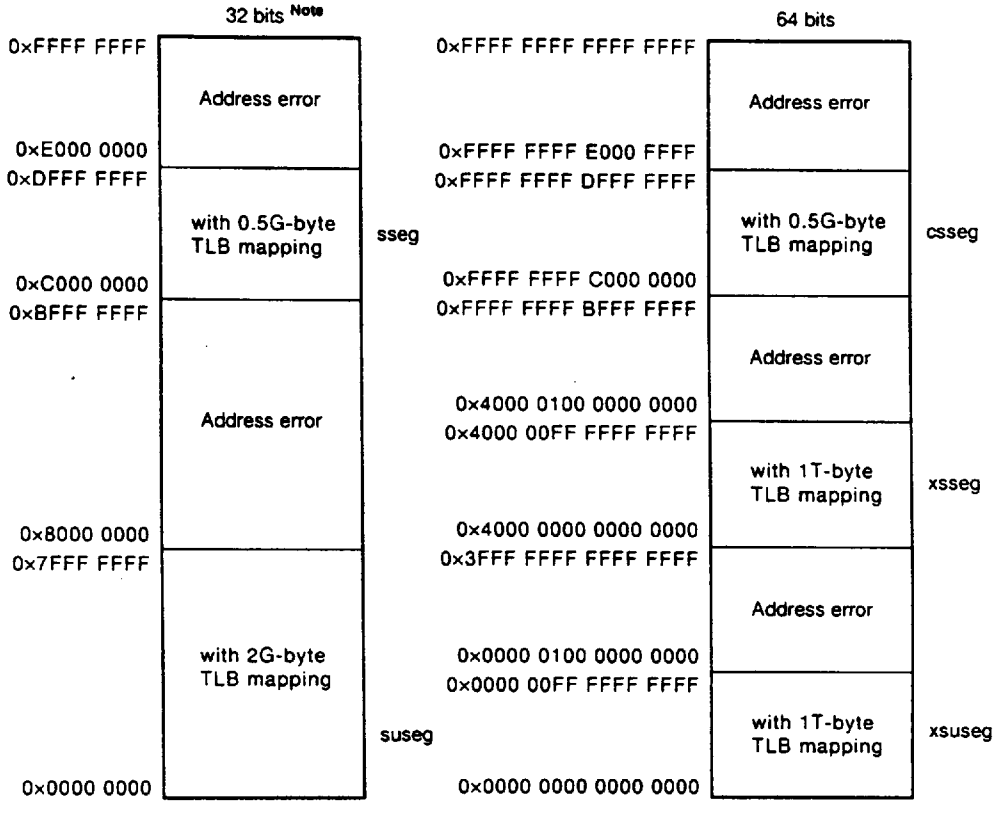
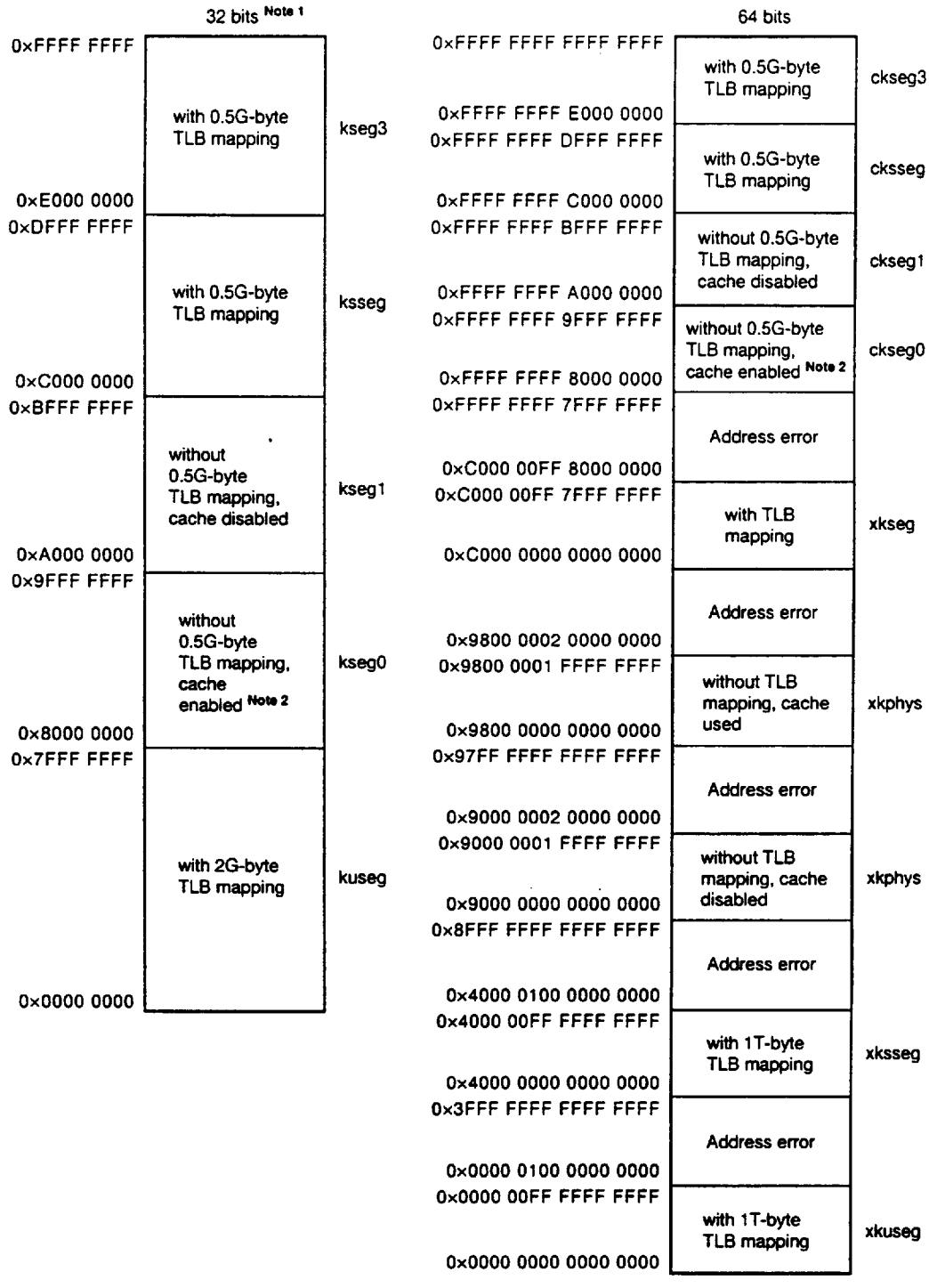


Figure 3-35. Supervisor Mode Address Space



Note In the 32-bit mode, the value of bit 31 is sign-extended to bits 32 through 63. The operation is undefined in case of a failure.

Figure 3-36. Kernel Mode Address Space



Notes 1. In the 32-bit mode, the value of bit 31 is sign-extended to bits 32 through 63. The operation is undefined in case of a failure.

2. The config register K0 area determines whether this is the cache area.

3.4.26 High-speed Translation Lookaside Buffer (TLB)

Mapped virtual addresses are translated into physical addresses by the high-speed translation lookaside buffer (TLB). The VR4100 has a full-associative TLB. This TLB is also called an integrated TLB (JTLB) because it can translate the addresses of an instruction and data with only one hardware unit.

TLB has 32 entries. Each entry simultaneously checks whether it matches a virtual address. Each TLB entry consists of a pair of an odd page and an even page. The VR4100 has five types of TLB page size, 1K byte, 4K bytes, 16 bytes, 64K bytes, and 256K bytes. The page size is specified for each entry by the MASK bit of that entry. For details of the MASK bit, refer to **3.4.6 Page Mask Register (5)**.

If the least significant bit of the virtual page number (VPN) is 1, the TLB entry for the odd number page is used; if VPN is 0, the entry for the even number is used. VPN is defined as follows depending on the TLB page size.

Table 3-11. Bit Correspondence between TLB Page Size and Virtual Addresses

Page Size	VPN	VPN2
1K bytes	Bits 63-10	Bits 63-11
4K bytes	Bits 63-12	Bits 63-13
16K bytes	Bits 63-14	Bits 63-15
64K bytes	Bits 63-16	Bits 63-17
256K bytes	Bits 63-18	Bits 63-19

Remark All addresses shown are virtual addresses.
Mask the higher 32 bits of the virtual address in the 32-bit mode.

A virtual address matches a TLB entry in the following cases:

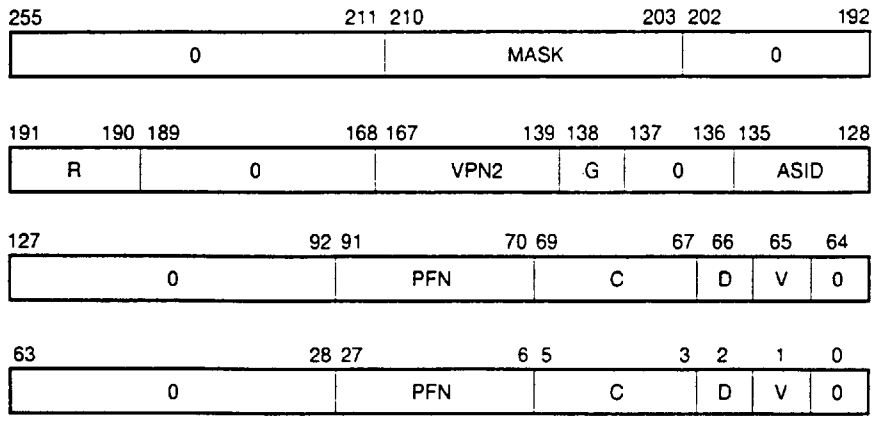
- When the virtual page number (VPN) of the virtual address is equal to the VPN of the entry, and the global bit (G) of the TLB entry is set
- When the virtual page number (VPN) of the virtual address is equal to the VPN of the entry, and the address space identifier (ASID) of the virtual address matches the ASID of the TLB entry

The valid bit (V) of each entry is not used to determined whether the TLB entry matches the virtual address. However, set this bit to translate addresses.

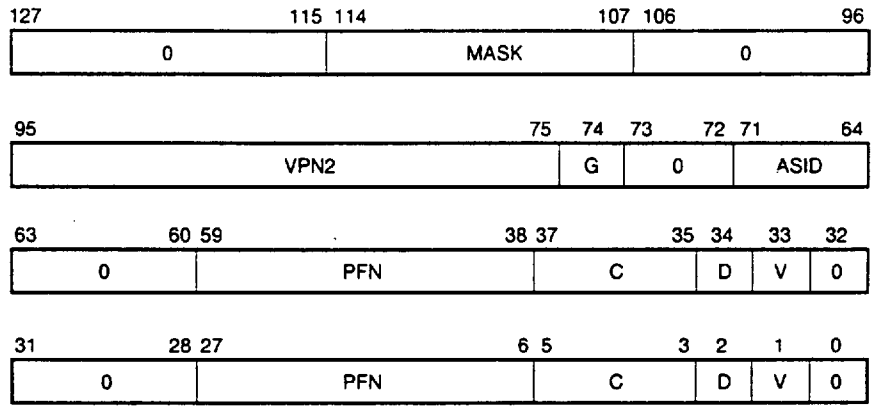
If a matching entry exists
A physical address and access control bits (C, D, and V) are obtained.
If data is stored to a page 0 with the D bit of the access control bit, a TLB change exception occurs.
If the V bit is 0, a TLB invalid exception occurs.
If the C bit is 010, the output physical address is used to access the main memory, not the cache.

If the TLB entry does not match
A TLB unmatch exception occurs

Format of Each TLB Entry in 64-Bit Addressing Mode



Format of Each TLB Entry in 32-Bit Addressing Mode



MASK : Page comparison mask value. Sets the page size of the corresponding entry.

R : Space type (00 → user, 01 → supervisor, 11 → kernel). Used to compare bits 63 and 62 of a virtual address.

VPN2 : Value of a virtual page number divided by 2 (maps two pages).

G : Global. ASID is ignored when TLB is referenced if the global bits of both entry Lo0 and entry Lo1 registers are set.

ASID : Address space ID area. Verified against the ASID area of the entry Hi register when TLB is referenced. TLB can be shared by multiple processes.

PFN : Page frame number (higher bits of physical address).

C : Cache algorithms (page attribute).
010 → Cache is used.
Others → Cache is not used.

D : Dirty (0 → write disabled, 1 → write enabled). Used by software to prevent changing of data as a "write-protect" bit.

V : Valid (0 → valid, 1 → invalid)

0 : RFU. Write 0 to this bit. 0 is returned when it is read.

Micro TLB

As described above, the thirty-two TLBs (JTLBs) can be accessed by software. In addition to these, the VR4100 has four micro TLBs (ITLBs) for instruction address conversion and four micro TLBs (DTLBs) for data address conversion. These TLBs are used as primary TLBs for high-speed address conversion.

Micro TLBs are accessed in the same manner as JTLBs. If a miss occurs in a micro TLB, and while a new TLB entry is transferred from a JTLB to the micro TLB, the pipeline is stalled.

3.4.27 VR4100 Processor Mode

The VR4100 supports several user-selectable modes. Except for the power mode command, BigEndian, $\overline{\text{Div2}}$, and HizParity pins, these modes can be set by writing into the status register (12) and config register (16).

(1) Power mode

The VR4100 supports three types of power modes, in addition to Fullspeed mode, to reduce the power dissipation:

- Fullspeed mode : Normal operation mode. All the clocks operate.
- Standby mode : The internal clocks except that related to the timer/interrupt are stopped.
- Suspend mode : The internal clocks except that related to the timer/interrupt, and TClock are stopped.
- Hibernate mode : All the clocks generated by the processor are stopped.

To change the mode from Fullspeed, a dedicated instruction is used. To change the mode from Standby or Suspend to Fullspeed, an interrupt is generated. To change from Hibernate to Fullspeed, execute a cold reset.

For details, refer to **3.1 POWER MODES**.

(2) High-speed data (AD) mode

When the AD bit of the config register is set, data can be output without having to insert an idle cycle in a data write cycle to the non-cache area of the system interface. For details, refer to **4.1.4 (12) High-speed data (AD) mode**.

(3) High-impedance parity mode

When the HizParity pin is made active, the SysADC bus and SysCmdP pin are fixed to the high-impedance state. Consequently, a parity check is not performed on the system interface.

However, a parity check related to internal cache access is performed.

(4) Reversing endian

When the RE bit of the status register is set, the endian in the user mode is reversed. For details, refer to **VR4200 User's Manual**.

(5) Bootstrap exception vector

The BEV bit of the status register is used when an exception occurs while self-diagnostic test is conducted before checking the normal operations of the cache and main memory systems.

When this bit is set or cleared, the virtual address of the exception vector (in 64-bit mode) are as follows. Also, when in 32-bit mode, the higher 32 bits of the virtual address are masked.

	Vector Base Address	Vector Offset
Cold reset, software reset, NMI	0xFFFF FFFF BFC0 0000 (BEV bit is automatically set to 1.)	0x0000
Cache error	0xFFFF FFFF A000 0000 (BEV = 0) 0xFFFF FFFF BFC0 0200 (BEV = 1)	0x0100
TLB unmatch, EXL = 0	0xFFFF FFFF 8000 0000 (BEV = 0) 0xFFFF FFFF BFC0 0200 (BEV = 1)	0x0000
XTLB unmatch EXL = 0		0x0080
Others		0x0180

The BEV bit is set to 1 when a cold reset, soft reset, or NMI exception occurs.

(6) Cache error check

When the CE bit of the status register is set, the contents of the parity error register are used as the parity instead of the calculated parity. When data is stored to the register, this bit enables the user to directly write the parity bit in the data cache. The parity bit in the instruction cache can be directly written from the parity error register by using the "FILL" CACHE instruction and setting the CE bit. If the CE bit is set, the contents of the TagLo register are written as the tag parity, instead of the calculated tag.

(7) Disable error

When the DE bit of the status register is set, occurrence of an exception due to a parity error is disabled.

(8) Kernel expansion addressing

When the KX bit of the status register is set, and if a TLB miss occurs in the kernel address space, the processor uses the XTLB unmatch exception vector.

In the kernel mode, the 64-bit instruction can always be used regardless of the status of this bit.

(9) Supervisor expansion addressing

When the SX bit of the status register is set:

- The 64-bit instruction is enabled in the supervisor mode.
- The XTLB unmatch exception vector is used if a TLB miss occurs in the supervisor address space.

When this bit is cleared, virtual address translation compatible with MIPS-II is executed.

(10) User expansion addressing

When the UX bit of the status register is set,

- The 64-bit instruction is enabled in the user mode.
- The XTLB unmatch exception vector is used if a TLB miss occurs in the user address space.

When this bit is cleared, virtual address translation compatible with MIPS-II is executed.

(11) Interrupt enable

When the IE bit of the status register is set, all the interrupts except NMI and reset are disabled.

3.4.28 Processor Interrupts

The VR4100 has the following four interrupt exceptions:

(1) NMI (Non-Maskable Interrupt Request) : 1 source
(2) Software interrupt : 2 sources
(3) External interrupt : 5 sources
(4) Timer interrupt : 1 source

(1) NMI (Non-Maskable Interrupt Request)

The NMI request is acknowledged when the $\overline{\text{NMI}}$ pin is made active. This interrupt request can also be set by an external write request via the SysAD bus. In the address cycle of this external write request, set SysAD(6:4) to 0. In the data cycle, SysAD22 serves as a SysAD write enable bit. This interrupt request cannot be masked.

(2) Software interrupt

Bits 9 and 8 of the cause register (13), i.e., bits 1 and 0 of the IP field are used. These bits can be written by software, and cannot be set by hardware. This interrupt request can be masked by bits 9 and 8 (the IM field bits 1 and 0) of the interrupt status register (12).

(3) External interrupt

Interrupt requests are acknowledged when external interrupt request pin $\overline{\text{Int(4:0)}}$ is made active. These interrupt requests can also be set by the external write request via the SysAD bus. In the address cycle of this external write request, clear SysAD(6:4) to 0.
In the data cycle, SysAD(20:16) serve as SysAD(4:0) write enable bits.
SysAD(4:0) are the values to be written to the interrupt register.
When any of these interrupt requests are acknowledged, bits 14 through 10 (IP field bit 6 through bit 2) of the cause register (13) are set.
These interrupt requests can be masked by bits 6 through bit 10 (the IM field bit 6 through bit 2) of the status register (12).

(4) Timer interrupt

Bit 15 of the cause register (13), i.e., bit 7 in the IP field is used.
When the value of the count register (9) is equal to the value of the compare register (11), this bit is set. This interrupt request can be masked by bit 15 (the IM field bit 7) of the status register (12).

3.5 DATA/INSTRUCTION CACHE

The VR4100 has data and instruction caches of direct map type.

- Data cache ··· 1K bytes
  Write back type
- Instruction cache ··· 2K bytes

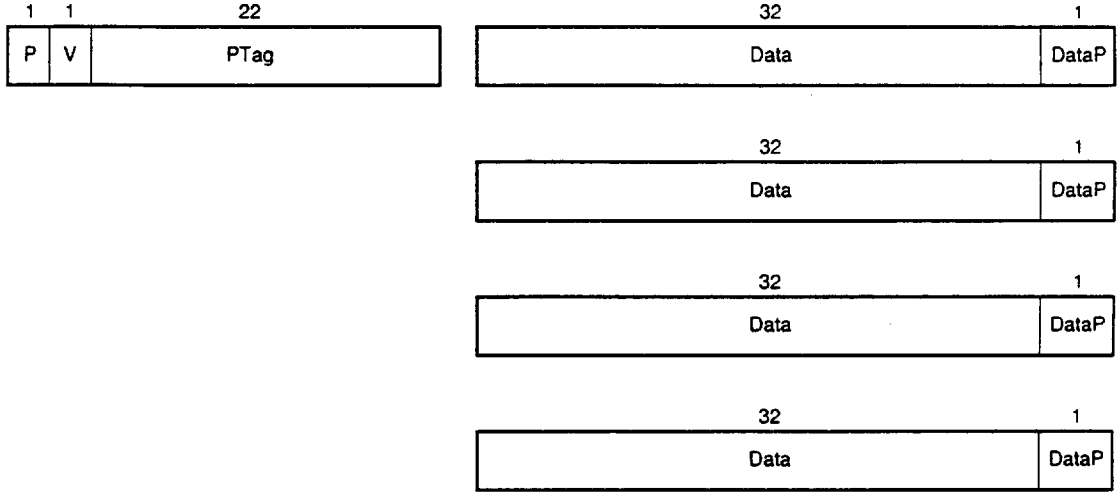
In both cases, the line size is 4 words (16 bytes), with virtual addressing and the physical tag technique.

3.5.1 Cache Configuration

(1) Configuration of instruction cache line

The format of the instruction cache line is shown below.

Figure 3-37. Format of Instruction Cache Line

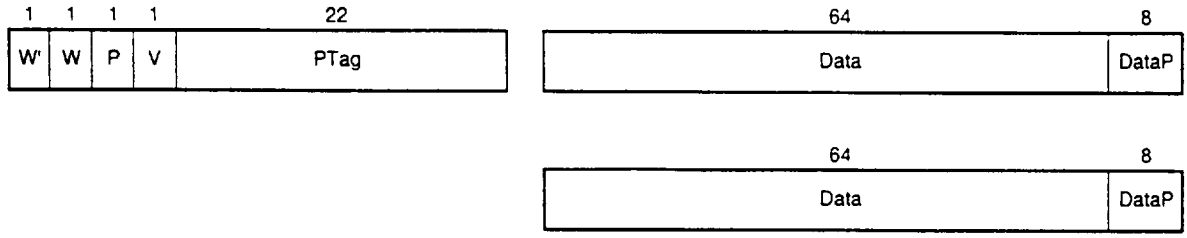


PTag : Physical tag (bits 31 through 10 of physical address)
V : Valid bit
P : Even number tag parity
Data : Cache data
DataP : Even number data parity (1 bit per 1 word of data)

(2) Configuration of data cache line

The format of the data cache line is shown below.

Figure 3-38. Format of Data Cache Line



PTag : Physical tag (bits 31 through 10 of physical address)
V : Valid bit
P : Even number tag parity
W : Write back bit (set at storing to cache line)
W' : Write back bit parity
Data : Cache data
DataP : Even number data parity (1 bit per 1 word of data)

3.5.2 Cache Status

(1) Instruction cache

The instruction cache has the following two cache statuses:

	Cache Status	Information on Cache Line
1	Valid	Valid
0	Invalid	Invalid

(2) Data cache

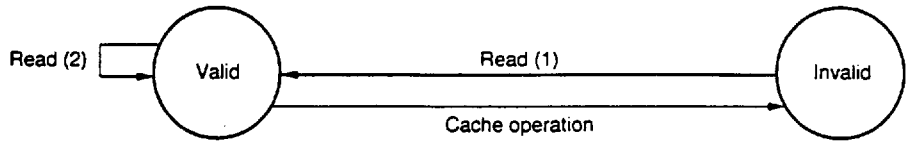
The data cache has the following three cache statuses:

V bit	W bit	Cache Status	Information on Cache Line
0	×	Invalid	Invalid
1	0	Valid Clean	Valid Cache contents match those of main memory
1	1	Valid Dirty	Valid Cache contents do not match those of main memory

(3) Status transitions of the instruction cache

The following figure illustrates the status transitions of the instruction cache.

Figure 3-39. Status Transitions of the Instruction Cache

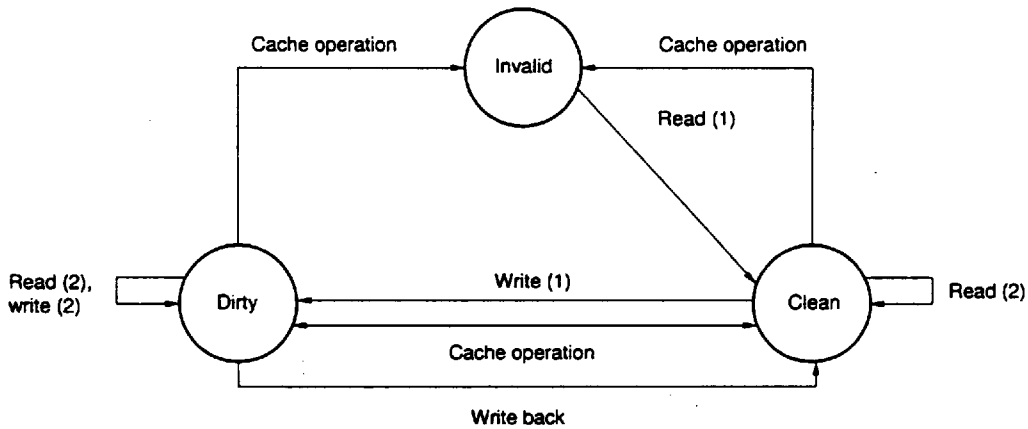


- Read (1) : Read from main memory to cache. Cache status is changed.
- Read (2) : Read from cache to processor. Cache status is not changed.

(4) Status transition of data cache

Figure 3-40 shows the status transition of the data cache.

Figure 3-40. Transition of Data Cache Status



- Read (1): Read from main memory to cache. Cache status is changed.
- Write (1): Write from processor to cache. Cache status is changed.
- Read (2): Read from cache to processor. Cache status is not changed.
- Write (2): Write from processor to cache. Cache status is not changed.

(5) Status transition caused by CACHE instruction

The status of the valid cache line also changes when the processor executes the CACHE instruction. For the CACHE instruction, refer to **3.5.3 CACHE Instruction**.

(6) Cache processing by external agent

The VR4100 does not allow the external agent to manipulate the status or contents of the cache.

3.5.3 CACHE Instruction

The VR4100 cannot directly control the status and data of the cache from the system interface. Instead, the status and data of the cache can be controlled through software by using the CACHE instruction. The CACHE instruction is also used to initialize the cache. The CACHE instruction is a part of the MIPS-III instruction set. For details, refer to **VR4200 User's Manual**. The operations of the CACHE instruction are shown in the following table.

Table 3-12. CACHE Instruction

Operation	Cache	Operation
Index Invalidate	Instruction/data	Invalidates the cache status of the cache block.
Index WriteBack Invalidate	Data	Checks the cache status. If it is dirty, the cache block is written back to the main memory, and the cache status is invalidated.
Index Load Tag	Instruction/data	Reads the tag corresponding to the specified index and loads it to the TagLo register. No error is checked at this time.
Index Store Tag	Instruction/data	Writes the contents of TagLo register to the tag of the cache block corresponding to the specified index.
Create Dirty	Data	If the cache block does not contain a specified address and that block is dirty, writes back the cache block to the main memory. The tag is set to a specified physical address, and the cache status becomes dirty.
Hit Invalidate	Instruction/data	If the cache block contains a specified address, invalidates the cache status.
Hit WriteBack Invalidate	Data	If the cache block contains a specified address, and if that block is dirty, writes back the cache block to the main memory, and invalidates the cache status.
Fill	Instruction	Fills the block of the instruction cache with the data from the main memory.
Hit WriteBack	Data	If the cache block contains a specified address and if that block is dirty, writes back the cache block to the main memory.
Hit WriteBack	Instruction	If the cache block contains a specified address, writes back the cache block to the main memory unconditionally.

3.5.4 Protecting Cache Data

The data of the instruction cache and data cache are protected by parity. The tags corresponding to these data are also protected by parity.

The parity bit is checked each time the cache is accessed to see if an error occurs. If a parity error occurs when the instruction cache, data cache, or main memory is accessed, the processor generates a cache error exception, and the cache error register indicates the cause of the error.

The following table shows how the parity is created and checked when the cache is accessed.

Table 3-13. Data Preservation during Cache Operation

Cache Operation	Operation of Data Cache	Operation of Instruction Cache
Store Hit	Tag remains unchanged and new data is stored. Tag parity is checked, and data parity is created.	Does not take place
Load Hit	Because data and tag parities exist and are not changed, both data and tag are checked.	Same as data cache
Load Miss	If cache line (data cache only) is Valid Dirty, that cache line is written back (refer to **Remark** below). External agent only supplies data parity and does not supply tag parity. Processor checks data parity and generates tag parity. Dirty bit does not include generation of tag parity.	Same as data cache
Store Miss	If cache line (data cache only) is Valid Dirty, that cache line is written back (refer to **Remark** below). External agent only supplies data parity and does not supply tag parity. Processor checks data parity and generates tag parity. Dirty bit does not include generation of tag parity.	Does not take place

Remark Write back procedure:

To write back in case of a store miss, the data tag and tag parity are checked, and the data parity is transferred to address and is written to the write buffer.

If an error is detected in the data field, the write back operation is not completed, and wrong data is written to the main memory. If an error is detected in the tag field, the write back operation is completed. In either case, a cache error exception occurs.

During CACHE operation, the cache data is not checked, but the tag parity is checked. If a tag parity error occurs at this time, a cache error exception occurs, and the operation cannot be completed.

3.6 EXCEPTION PROCESSING

The exception processing mechanism of the VR4100 efficiently processes exception events (such as TLB translation miss, arithmetic overflow, interrupt, and system call). These exceptions interrupt the normal instruction execution flow. The instruction that has generated an exception, the instructions that follow, and the instruction whose execution has been already started are aborted, and execution directly jumps to a handler routine.

To facilitate analysis of an exception, exception processing, and returning of execution to the normal instruction execution flow, the VR4100 keeps minimum the exception-related information to be saved to the coprocessor.

3.6.1 Operation in Case of Exception

If an exception occurs, the VR4100 enters the kernel mode with interrupts disabled, and executes an exception handler from a fixed exception vector address. When execution returns from the exception handler, it is necessary to restore the original information on the program counter, operating mode, and enabling of interrupts. Therefore, save this information when an exception occurs.

When an exception occurs, the EPC register holds the address of the instruction that has generated the exception. If the exception occurs in the delay branch slot, the EPC register holds the address of the instruction immediately before the instruction that has generated the exception. In other words, the EPC register stores an address from which execution is to be started after exception processing has been completed.

The VR4100 also has the following modes related to exception processing.

- Interrupt enable (IE bit : enable, disable)
- Operating modes (KSU bit : user, supervisor, and kernel)
- Exception levels (EXL bit : normal and exception)
- Error levels (ERL bit : normal and error)

An interrupt is enabled when "interrupt enable" (IE = 1) and the exception/error level is "normal" (EXL = 0, ERL = 0).

The operating modes are set as follows:

Exception level : normal ··· mode specified by the KSU bit of the status register
: exception ··· always kernel mode

The exception level is set again to "normal" by the ERET instruction that returns execution from an exception handler. For details, refer to ERET instruction in **VR4200 User's Manual.**

3.6.2 Accuracy of Exception

The instruction that has generated an exception and the instructions that follow can be aborted before the state is changed to another, and can be executed again after the exception processing has been completed. If execution of the instructions that follow the instruction responsible for the occurrence of the exception is aborted, any exception that may occur as a result of executing these instructions is invalidated.

3.6.3 Types of Exceptions

The exceptions processed by the VR4100 are described next.

For details, refer to **VR4200 User's Manual.**

Table 3-14. Types of Exceptions

Exception	Abbreviation	Description
Cold reset	–	Occurs if $\overline{\text{ColdReset}}$ and $\overline{\text{Reset}}$ signals are simultaneously asserted active. Aborts instruction execution and executes a handler on the reset vector. The internal status is undefined, except some bits of the status register.
Soft reset	–	Occurs if $\overline{\text{Reset}}$ signal is asserted active. Aborts instruction execution and executes a handler on the reset vector. The internal status before soft reset is retained.
NMI	–	Non-maskable interrupt request by the external agent.
TLB unmatch	TLBL/TLBS	Occurs if the operating mode is the 32-bit mode and the number of TLB entries matching the referenced address runs short.
Expansion addressing TLB unmatch	TLBL/TLBS	Occurs if the operating mode is 64-bit mode and the number of TLB entries matching the referenced address runs short.
TLB invalid	TLBL/TLBS	Occurs if the TLB entry matching the referenced virtual address is invalid. (V bit = 0)
TLB change	Mod	Occurs if a TLB entry that coincides with virtual address to be accessed is valid but write is disabled (D bit = 0) when the store instruction is executed
Bus error	IBE/DBE	Occurs if an external agent indicates a data error on the SysCmd bus by an external interrupt to the bus interface (bus timeout, bus parity error, or invalid physical memory address or access type)
Address error	AdEL/AdES	Occurs if an attempt is made to execute the LH, SH/LW/SW, LD, or SD instruction to the half word/word/double word not at the half word/word/double word boundary, or to reference a virtual address that cannot be accessed.
Integer overflow	Ov	Occurs if 2's complement overflow occurs as a result of addition or subtraction.
Trap	Tr	Occurs if the condition is true at trap instruction execution.
System call	Sys	Occurs when the SYSCALL instruction is executed.
Breakpoint	Bp	Occurs when the BREAK instruction is executed.
Reserved instruction	RI	Occurs when an instruction whose op code (bits 31-26) is undefined, or the SPECIAL instruction whose op code (bits 5-0) is undefined is executed.
Coprocessor unusable	CpU	Occurs if the coprocessor instruction is executed when the corresponding coprocessor use enable bit is not set.
Interrupt	Int	Occurs when one of the eight interrupt sources becomes active.
Cache error	–	Occurs when a parity error is detected on the internal cache or system interface.
Watch	WATCH	Occurs when an attempt is made to reference the physical address in the watch Lo/Hi register with the load/store instruction.

(1) Exception vector address

When a cold reset, soft reset, or NMI exception occurs, execution always branches to the following vector address:

- In 32-bit mode: 0xBFC0 0000
- In 64-bit mode: 0xFFFF FFFF BFC0 0000

If any other exception occurs, the vector address is the sum of a vector offset and a base address.
The exception vectors and their offset values in the 64-bit and 32-bit modes are shown in the tables below.

Table 3-15. Base Address of Exception Vector in 64-Bit Mode

	Vector Base Address	Vector Offset
Cold reset, soft reset, NMI	0xFFFF FFFF BFC0 0000 (BEV bit is automatically set to 1.)	0x0000
Cache error	0xFFFF FFFF A000 0000 (BEV = 0) 0xFFFF FFFF BFC0 0200 (BEV = 1)	0x0100
TLB unmatch, EXL = 0	0xFFFF FFFF 8000 0000 (BEV = 0)	0x0000
XTLB unmatch, EXL = 0	0xFFFF FFFF BFC0 0200 (BEV = 1)	0x0080
Others		0x0180

Table 3-16. Base Address of Exception Vector in 32-Bit Mode

	Vector Base Address	Vector Offset
Cold reset, soft reset, NMI	0xBFC0 0000 (BEV bit is automatically set to 1.)	0x0000
Cache error	0xA000 0000 (BEV = 0) 0xBFC0 0200 (BEV = 1)	0x0100
TLB unmatch, EXL = 0	0x8000 0000 (BEV = 0	0x0000
XTLB unmatch, EXL = 0	0xBFC0 0200 (BEV = 1)	0x0080
Others		0x0180

(2) Priority of exception

When two or more exceptions occur for one instruction, only one of them is selected. Table 3-17 shows the priority.

Table 3-17. Exception Priority

High ↑	Cold reset
	Soft reset
	NMI
	Address error (instruction fetch)
	TLB/XTLB unmatch (instruction fetch)
	TLB invalid (instruction fetch)
	Cache error (instruction fetch)
	Bus error (instruction fetch)
	System call
	Breakpoint
	Coprocessor unusable
	Reserved instruction
	Trap
	Integer overflow
	Address error (data access)
	TLB/XTLB unmatch (data access)
	TLB invalid (data access)
	TLB change (data write)
	Cache error (data access)
	Watch
	Bus error (data address)
Low	Interrupt (other than NMI)

4. INTERFACE

4.1 SYSTEM INTERFACE

Events generated inside the processor when an access to an external system is requested are called "system events". These system events include the following:

- Instruction fetch caused by instruction cache miss
- Load/store of data caused by data cache miss
- Load/store of uncache field
- Processing caused by execution of CACHE instruction

When a system event occurs, the processor accesses the external system. To process this event, the processor issues a request via system interface. Connect this system interface to an external agent that can control access to the system.

(1) Processor request

The processor requests include the following:

- Read requests
  Gives the external agent a read address.
- Write request
  Gives an address and single data/block data to write the external agent.

(2) External requests

The external requests include the following:

- Read response
  Given by the external agent to transfer data.
- Write request
  Gives the processor an address and word data.

When the external agent receives a read request, it accesses a specified resource, and returns the requested data as a read response after the read request.

The read request of the processor in the status in which a read request has been issued but the data is not returned as a read response is in the "pending" status. If a read request is in the pending status, the processor cannot issue another request. The processor read request ends after the processor has received the last data of the response data from the external agent, and the processor write request ends after the processor has transferred the last word of data.

For an external write request, it is possible only to request a processor interrupt. (Refer to **4.1.4 (7) External write request** for move information.)

Usually, the processor is the master of the system interface. When the processor issues a processor read request, however, the external agent serves as the master.

The protocol of the SysAD bus is described next. The VR4100 always monitors the status of the SysAD bus protocol and can successively issue requests without having to insert an idle cycle between them. The user should design the external agent so that these requests can be efficiently processed.

4.1.1 Sequence

A sequence is a series of requests the processor generates for a system event. The following paragraphs describe the sequence generated by the processor for each system event.

(1) Fetch miss

If a cache miss occurs as a result of a fetch from the instruction cache, the processor acquires a cache line that includes an instruction from the external agent. The processor issues a read request to acquire the cache line, and waits until the external agent returns instruction data as a response to the read request.

(2) Load miss

If a cache miss occurs as a result of loading from the data cache, the processor acquires a cache line that includes data, from the external agent. The processor issues a read request to acquire the cache line, and waits until the external agent returns data as a response to the read request. However, if the contents of the cache line to be replaced are dirty data, the processor writes back the data to the memory, before issuing a request.

(3) Store miss

If a cache miss occurs as a result of storing to the data cache, the processor issues a read request to acquire a cache line to which the store data is to be stored. If the contents of the cache line to be replaced are dirty data, however, the processor writes back the data to the memory, before issuing a request.

It is necessary that the store data in the cache match the contents of the main memory at the end of the program. Therefore, the contents of the cache line are flashed by the CACHE instruction. For the CACHE instruction, refer to **3.5.3 CACHE Instruction**.

(4) Uncache load/store

When the processor loads data from an uncache area, it issues a read request and waits for transfer of response data from the external agent. When the processor performs uncache store, it issues a write request and transfers data to the external agent.

(5) CACHE instruction

The VR4100 has a CACHE instruction that can manipulate the status and contents of the cache.

For the details of the CACHE instruction, refer to **3.5.3 CACHE Instruction**.

4.1.2 Endian

The endian mode of the system interface can be set as follows by using the BigEndian signal at reset. The endian can be changed in the user mode by setting the RE (reverse endian) bit of the status register, which can be set by software, to 1.

Table 4-1. Setting of Endian Mode

BE Bit	Mode
High level	Big endian mode
Low level	Little endian mode

4.1.3 System Interface Signal

The major system interface signals are transferred or received via 32-bit address/data bus SysAD(31:0) and 5-bit command bus SysCmd(4:0). These signals are driven in the following cases:

- When the processor operates as the master and issues a processor request
- When the processor operates as a slave and when the external agent issues an external request or response

Requests from the system interface, such as a write request and read response, consists of a system interface command that indicates the types of the address, data, and request. The address and data are transferred via the SysAD bus, and the system interface command is transferred via the SysCmd bus.

When the processor operates as the master and when the contents of the SysAD bus and SysCmd bus are valid, the processor asserts the $\overline{\text{PValid}}$ signal active. When the processor operates as a slave and if the contents of the SysAD bus and SysCmd bus are valid, the external agent asserts the $\overline{\text{EValid}}$ signal active.

The SysCmd bus defines its contents while the contents of the SysAD bus are valid.

SysCmd4 indicates whether the current cycle is an address cycle or a data cycle. The following table shows the relations between the status of SysCmd4 and the contents indicated by SysCmd(3:0).

SysCmd4	Contents Indicated by SysCmd(3:0)
Address cycle	System interface command
Data cycle	• Whether the current cycle is the last cycle of data transfer • Other information related to data

To learn how to encode the system interface command and the data identifier, refer to **4.1.6 Syntax of System Interface Command.**

4.1.4 Timing of System Interface

The system interface protocol functions to enable a request between the processor and external agent. Specifically, signals are output to the pins of the system interface in cycle units.

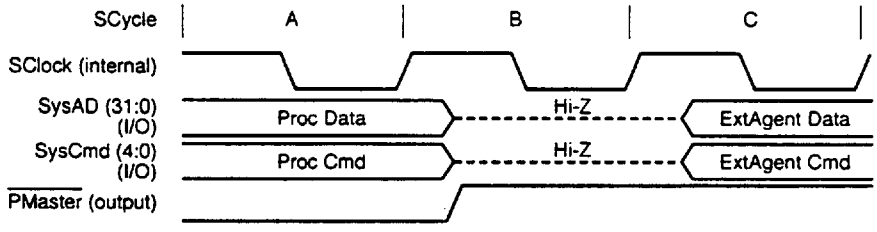
(1) General

The timing of each signal is described next.

(a) $\overline{\text{PMaster}}$ signal (output)

This signal indicates that the processor is in the status of the master of the system interface.

Figure 4-1. Processor Status Transition (Master to Slave)

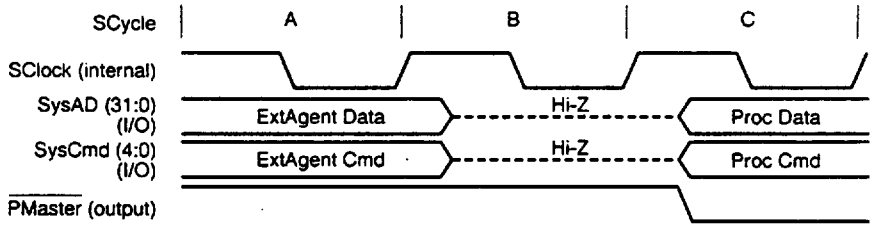


A: The processor drives SysAD(31:0) and SysCmd(4:0).

B: The $\overline{\text{PMaster}}$ signal is asserted inactive, and SysAD(31:0) and SysCmd(4:0) go into a high-impedance state.

C: The external agent outputs data to SysAD(31:0) and SysCmd(4:0).

Figure 4-2. Processor Status Transition (Slave to Master)



A: The external agent drives SysAD(31:0) and SysCmd(4:0).

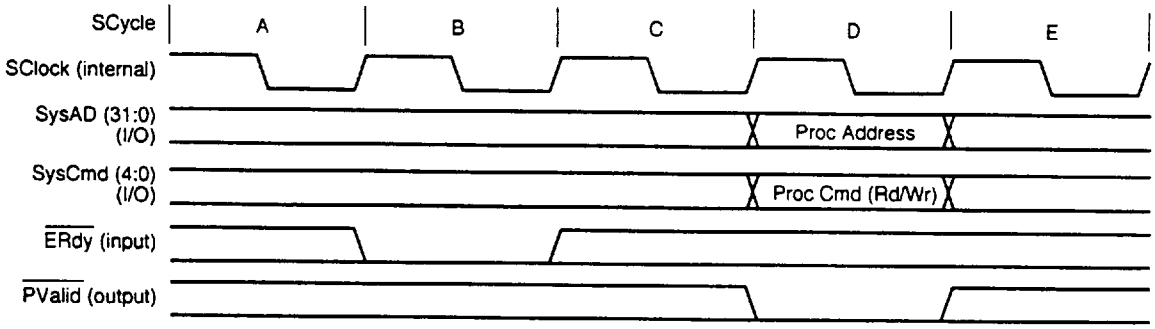
B: SysAD(31:0) and SysCmd(4:0) go into a high-impedance state.

C: The $\overline{\text{PMaster}}$ signal becomes active and the processor drives SysAD(31:0) and SysCmd(4:0).

(b) $\overline{\text{ERdy}}$ signal (input)

This signal indicates that the external agent can accept a processor request.

Figure 4-3. When External Agent Accepts Processor Request



B: The $\overline{\text{ERdy}}$ signal is active.

D: The processor asserts the $\overline{\text{PValid}}$ signal active and drives a read/write command. Receives a request from the processor.

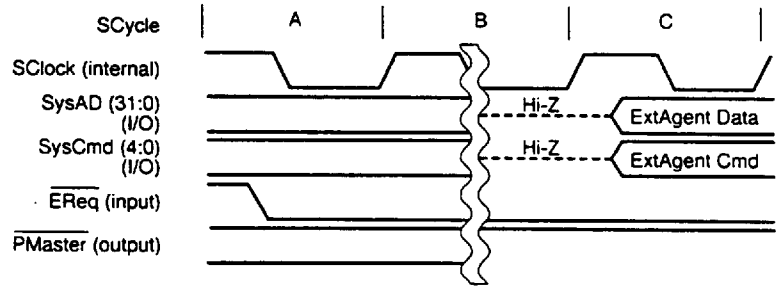
The external agent should receive write data to be output when it has accepted a processor write command. Moreover, it should send read response data at any time.

When the $\overline{\text{ERdy}}$ signal is inactive in cycle B, the processor continues a read/write command output.

(c) $\overline{\text{EReq}}$ signal (input)

This signal is used by the external agent to request for control of the system interface. Adjust the processor as follows so that the external agent can gain control of the bus.

Figure 4-4. Request by External Agent for Bus Control



A: The external agent asserts the $\overline{\text{EReq}}$ signal active.

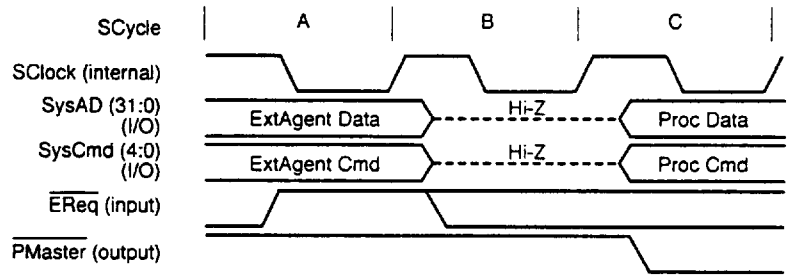
B: The external agent waits until the $\overline{\text{PMaster}}$ signal is asserted inactive.

C: The external agent outputs data to SysAD(31:0) and SysCmd(4:0). While the $\overline{\text{EReq}}$ signal is active, the external agent takes control of the bus.

When the $\overline{\text{EReq}}$ signal is asserted inactive, the external agent returns to the status A.

The external agent should receive one processor request since the $\overline{\text{EReq}}$ signal has been asserted active until the external agent gains control of the bus.
The external agent releases the control over the bus when the $\overline{\text{EReq}}$ signal is asserted inactive.

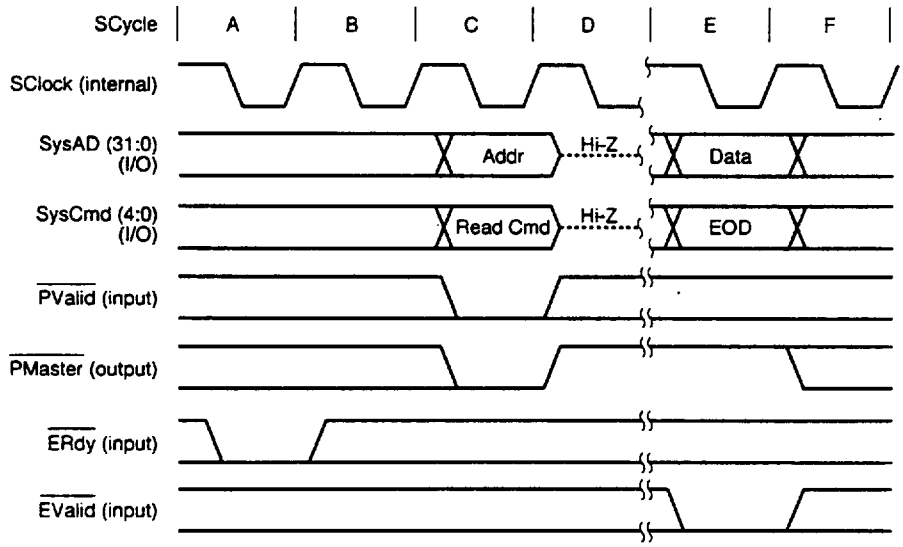
Figure 4-5. External Agent Releases Bus



A: The external agent asserts the $\overline{\text{EReq}}$ signal inactive, and drives SysAD(31:0) and SysCmd(4:0).
B: SysAD(31:0) and SysCmd(4:0) go into a high-impedance state.
C: The processor gains control of the bus.

The external agent can gain control of the bus when the $\overline{\text{EReq}}$ signal is asserted active, but the following processor read request is an exception.

Figure 4-6. Status Transition by Processor Read Request



C: The processor drives a valid read command and the external agent receives this.
D: The $\overline{\text{PMaster}}$ signal is asserted inactive, and SysAD(31:0) and SysCmd(4:0) go into a high-impedance state.
E: The external agent drives the last data of the request data. The external agent gains control of the bus in the cycle between D and E.

(2) Arbitration

Arbitration for transition of bus control is performed by the handshake signals ($\overline{EReq}$, $\overline{PReq}$, and $\overline{PMaster}$ signals) of the system interface.

(a) When processor is master

The external agent gains control of the bus when the processor is the master, as follows:

<1> The external agent asserts the $\overline{EReq}$ signal active.
<2> The processor asserts the $\overline{PMaster}$ signal inactive, releases control of the system interface, and transfers it to the external agent.
<3> The processor returns to the master status after the external request has been completed if the $\overline{EReq}$ signal is inactive.

The external agent must assert the $\overline{EReq}$ signal active until the processor asserts the $\overline{PMaster}$ signal inactive.
As long as the $\overline{EReq}$ signal is active, the external agent can serve as the master.
When the status transition has been completed, each processing is performed without changing the $\overline{EReq}$ signal.

(b) When processor is slave

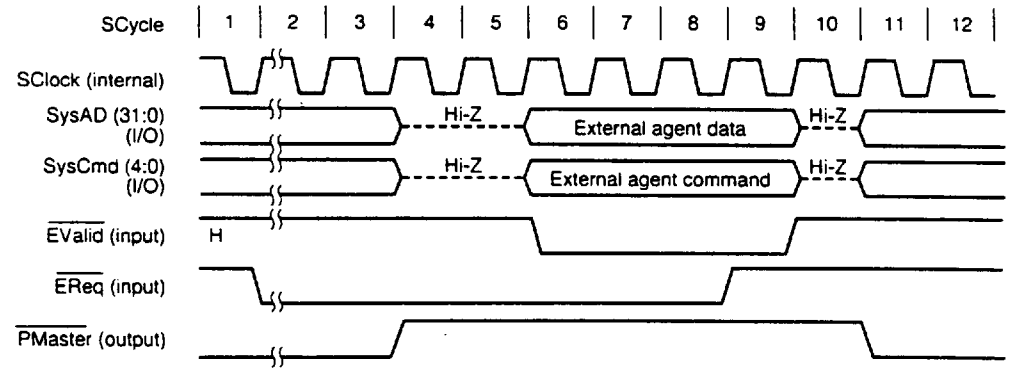
The processor requests for control of the bus when it is in the slave status, as follows:

<1> The processor asserts the $\overline{PReq}$ signal active and requests the external agent for control of the bus.
<2> When the processor learns that the $\overline{EReq}$ signal is inactive, it gains control of the bus, asserts the $\overline{PMaster}$ signal active, and executes commands.
<3> The processor enters the master status two cycles after the $\overline{EReq}$ signal has been asserted inactive.

(c) At reset

At reset (when the $\overline{Reset}$ or $\overline{ColdReset}$ signal is active), the processor serves as the master and the external agent serves as a slave.
Figure 4-7 shows the timing of arbitration of the external request.

Figure 4-7. Arbitration of External Request



(3) Command execution

(a) When the processor is bus master

The processor can issue a command two cycles after the $\overline{\text{ERdy}}$ signal has been asserted active. The command is accepted if the $\overline{\text{ERdy}}$ signal is asserted active for the duration of 1 cycle.

(b) When external agent is bus master

The operation in this case is always a response to a read request.

When the $\overline{\text{EReq}}$ signal is asserted active, the processor serves as the $\overline{\text{PMaster}}$, and the external agent serves as the bus master, processing is started. In this case, the processor can accept any command or data.

When the processor asserts the $\overline{\text{PReq}}$ signal active, the external agent asserts the $\overline{\text{EReq}}$ inactive, and relinquish control of the bus.

(4) Processor write request

(a) Issuance of request

The processor write request is issued by driving a write command and a write address to the SysCmd bus and SysAD bus, respectively, and asserting the $\overline{\text{PValid}}$ signal for one cycle.

(b) Data transfer

Data is transferred by driving the data identifier to the SysCmd bus and the write data to the SysAD bus, and asserting the $\overline{\text{PValid}}$ signal active for the necessary cycles, after a processor write request has been issued.

Data of 1 to 4 bytes is transferred in a single data cycle, and data exceeding 4 bytes is transferred in 4-byte units by repeating the data cycle until the required data has been completely transferred. Data is successively transferred when the write command is issued. EOD is used as the data identifier of the last cycle of data transfer.

The VR4100 can insert an idle cycle between two data cycles so that the external agent can easily receive the data. At this time, the processor drives data at the transfer rate set by the EP bit of the config register.

The address cycle can be extended by the $\overline{\text{ERdy}}$ signal.

Figures 4-8 and 4-9 show the data transfer timing of cache block store of 2 words.

Figure 4-8. Processor Block Write Request (Data Rate: D)

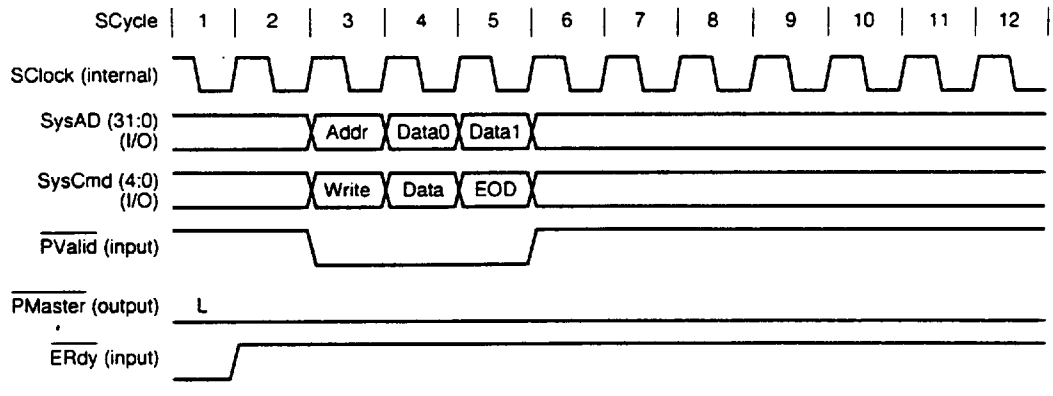
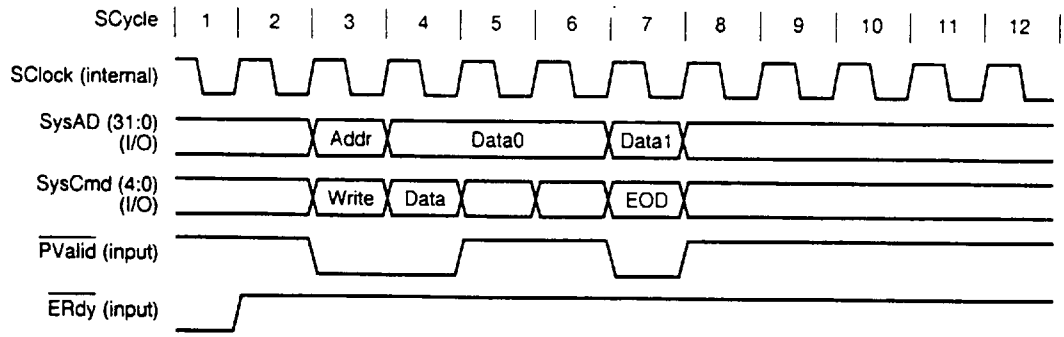


Figure 4-9. Processor Block Write Request (Data Rate: Dxx)



(5) Processor read request

(a) Issuance of request

The processor read request is issued by driving a read command to the SysCmd bus and a read address to the SysAD bus, and asserting the ~~PValid~~ signal active for 1 cycle.

Also, the address cycle can be extended by the ~~ERdy~~ signal.

(b) Reception of response data

The processor enters the read request pending status after it has issued a processor read request, but only one read request is pending. The processor waits for issuance of the subsequent read request until the external read response is returned. After issuing the read request, the processor asserts the ~~PMaster~~ signal inactive, and enters the slave status. The external agent can then return the request data as a read response. The external agent does not need to make a request to gain control of the system interface by asserting the ~~EReq~~ signal active.

In the slave status, the external agent can issue an external write request before it returns a read response. Figure 4-10 shows the timing where non-forced transition to a slave state is made by the processor read request.

Figure 4-10. Non-Forced Transition by Processor Read Request

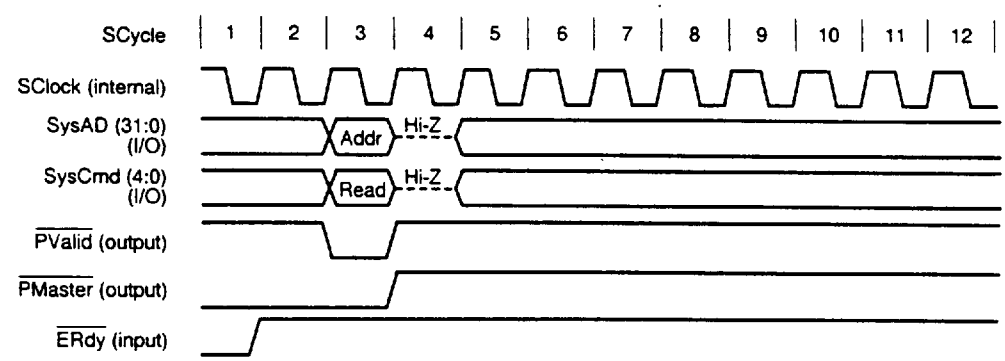
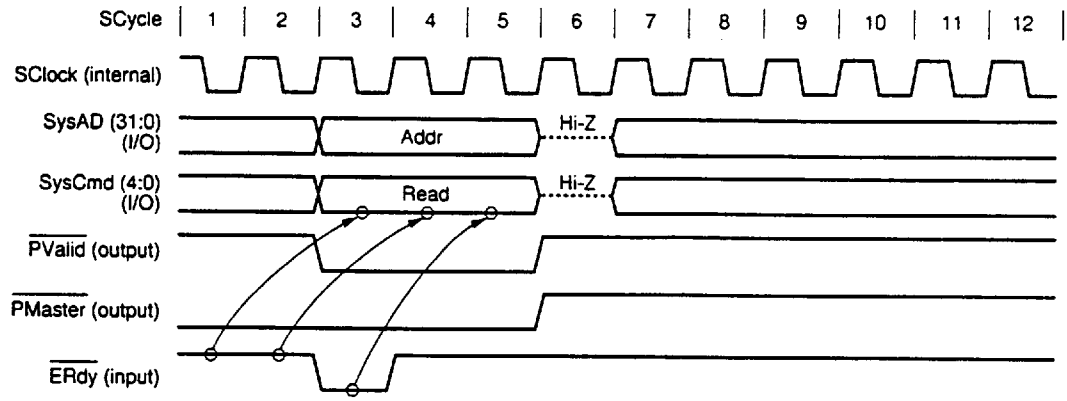


Figure 4-11. Delay of Read Request by $\overline{\text{ERdy}}$ Control



(6) External read response

The external agent should wait until the processor enters the slave status, and return data as a response to the processor read request. To transfer the requested data, only the number of data cycles corresponding to the data size is transferred. When the last data transfer cycle is completed, the read response ends. The external agent then releases control of the system interface, and the processor enters the master status.

To use the SysAD bus when the processor is in the slave status, the processor asserts the $\overline{\text{PReq}}$ signal active, and waits until the $\overline{\text{EReq}}$ signal is asserted inactive.

A data identifier can indicate that an error occurs in the contents of the data in that cycle. However, the external agent must return data of specific size, regardless of whether an error occurs or not. If the response data contains an error, the processor generates a bus error exception.

If a read response is returned when the processor read request is not pending, the operation of the processor cannot be guaranteed.

Figure 4-12 shows the timing of the processor single read request and the subsequent read response, and Figure 4-13 shows the timing of the block read response with the processor in the slave status.

Figure 4-12. Processor Single Read Request and Subsequent Read Response

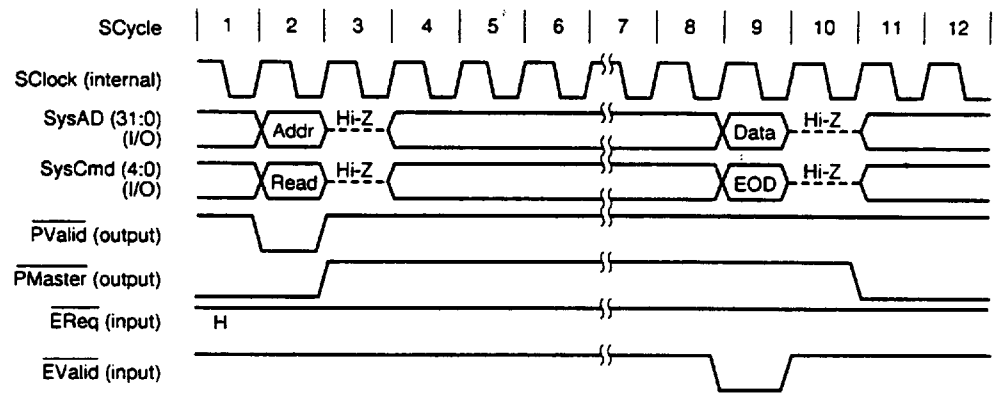
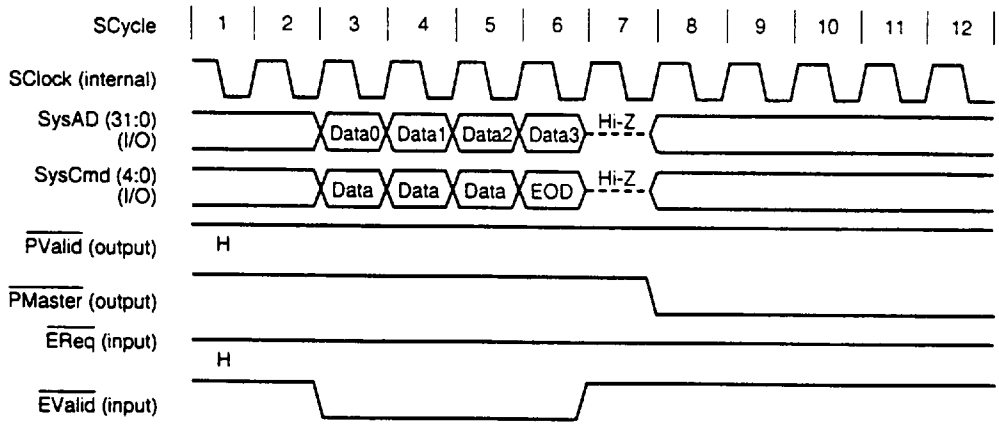


Figure 4-13. Block Read Response in Slave Status



(7) External write request

The external write request is the same as the processor write request except for the following:

- The $\overline{\text{EValid}}$ signal is asserted active instead of the $\overline{\text{PValid}}$ signal.

(a) Issuance of request

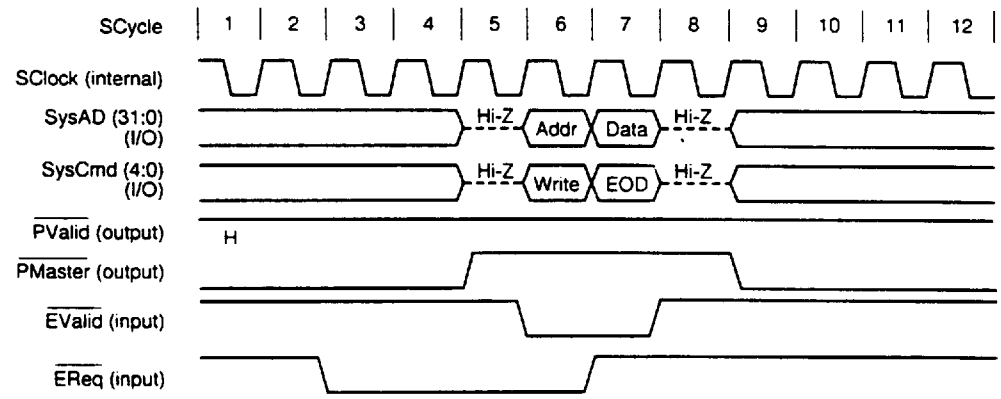
The external write request is issued by driving a write request command to the SysCmd bus and a write address to the SysAD bus, and asserting the $\overline{\text{EValid}}$ signal active for one cycle.

(b) Data transfer

In the cycle next to the one in which the request has been issued, a data identifier and write data are driven to the SysCmd bus and SysAD bus, respectively, and the $\overline{\text{EValid}}$ signal is asserted active for the duration of one cycle. Use EOD as the data identifier in the last cycle of data transfer.
When the last data transfer cycle is completed, the external write request ends.
When the request ends, the external agent asserts the $\overline{\text{EReq}}$ signal inactive, release control of the system interface, and the processor returns to the master status.

Figure 4-14 shows the timing of the external write request that is generated when the processor is in the master status.

Figure 4-14. External Write Request in Master Status



(8) Flow control

The $\overline{\text{ERdy}}$ signal is used by the external agent to control the flow of the processor read/write request. While the $\overline{\text{ERdy}}$ signal is inactive, the processor repeats the current address cycle. This address cycle continues until two cycle after the cycle that asserts the $\overline{\text{ERdy}}$ signal active.

Assert the $\overline{\text{ERdy}}$ signal active for the duration of at least one of cycle. To repeat the address cycle, assert the $\overline{\text{ERdy}}$ signal inactive at least two cycle before the first address cycle.

Figures 4-15 and 4-16 illustrates how to use the $\overline{\text{ERdy}}$ signal.

Figure 4-15. To Delay Processor Read Request

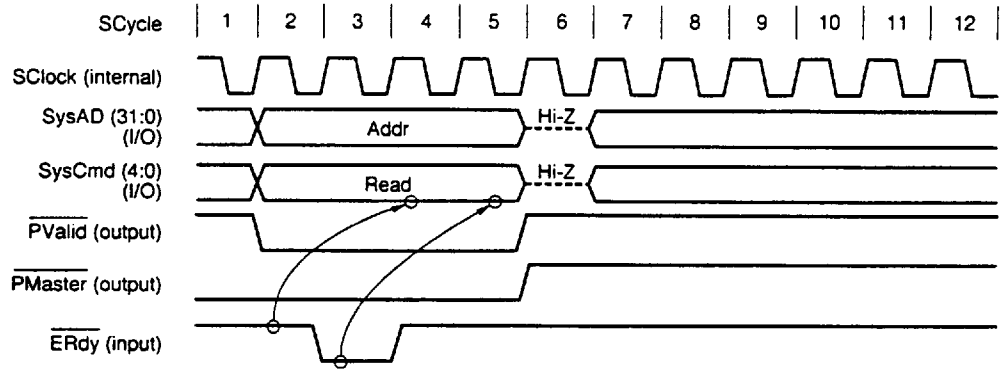
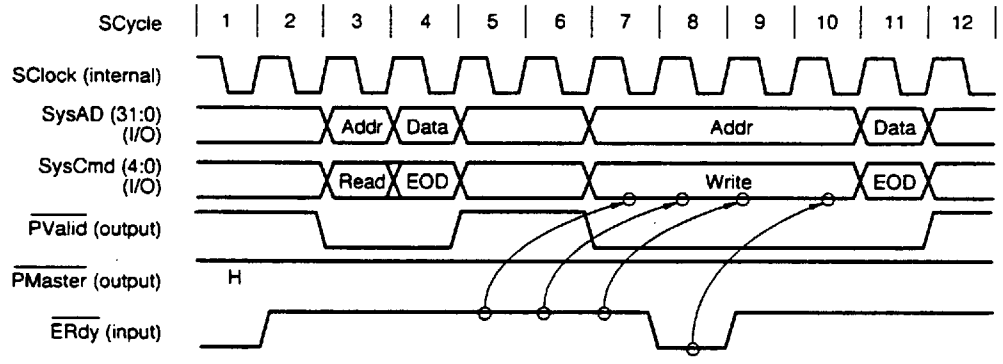


Figure 4-16. Two Processor Write Requests



(9) Control of data transfer rate

The system interface of the VR4100 can transfer word data in one cycle. The rate at which data is transferred to the processor is determined by the data transfer capability of the external agent. The processor decodes the contents of the SysCmd bus, including a data identifier, in a cycle in which the $\overline{\text{EValid}}$ signal is active and the data is valid. The processor continues receiving the data until it detects the last data transfer. The external agent can transfer data to the processor at any transfer rate.

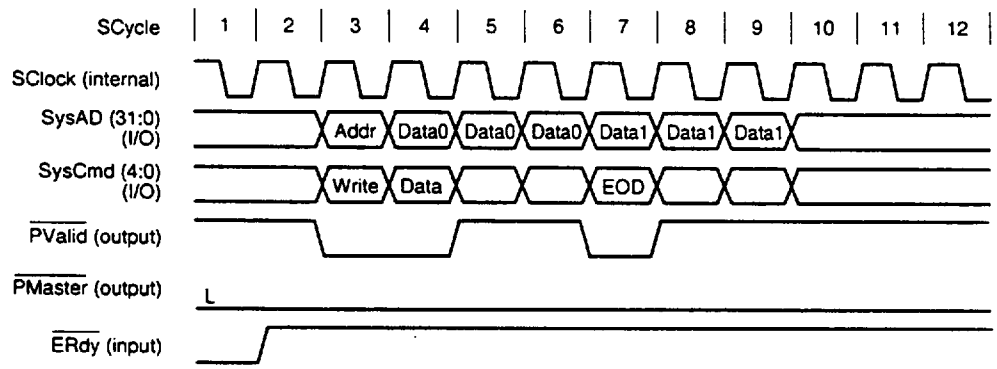
The rate at which data is transferred from the processor to the external agent is set by the EP field of the config register. The pattern of the data transfer rate is expressed by a combination of symbols "D" and "X", where "D" indicates the data transfer cycle, and "X" indicates an unused cycle. This transfer pattern indicates an appropriate data transfer rate by a data cycle and unused cycle.

Example DXX: Transfers 1-word data every 3 cycles

The VR4100 has four data transfer rates: "D", "DX", "DXX", and "DXXX". The processor holds the data output during the period "D" immediately before, during the period of "X", and continues output.

Figure 4-17 shows the timing of the processor block write request when the transfer pattern is DXX.

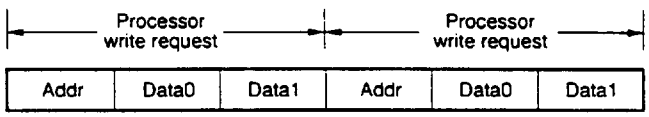
Figure 4-17. Processor Block Write Request When Transfer Pattern Is DXX



(10) Successive processor write requests

Even if another processor write request follows a processor write request as shown in Figure 4-18, a wait cycle is not inserted in between the two requests. The timing is the same as that shown for a processor write request.

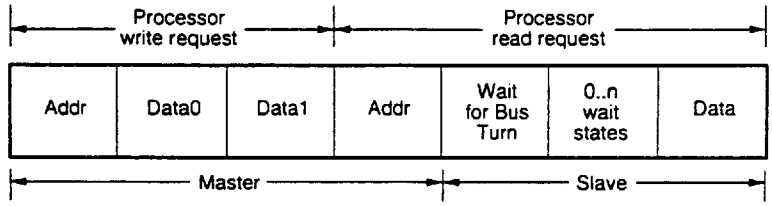
Figure 4-18. Successive Processor Write Requests



(11) Processor write request followed by processor read request

Processor write request followed by processor read request Figure 4-19 shows a processor write request followed by a processor read request. The timing is the same as that shown for a processor write request and processor read request.

Figure 4-19. Processor Write Request Followed by Processor Read Request

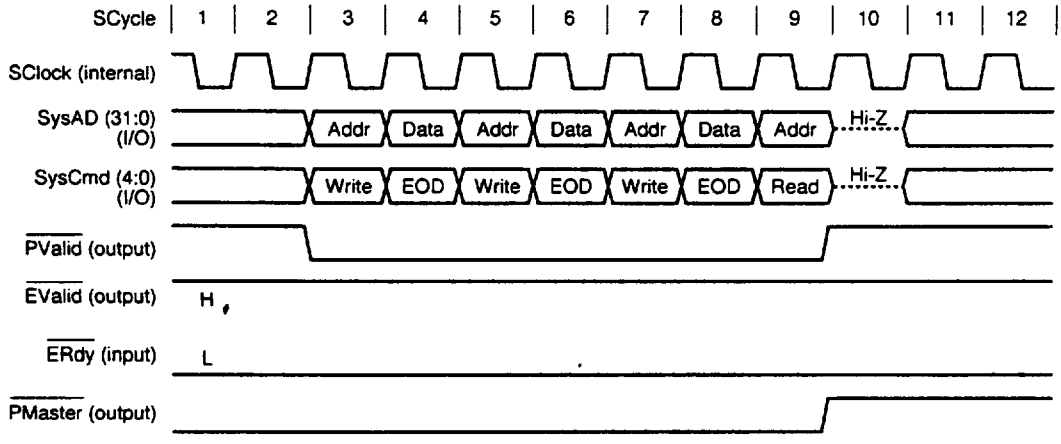


(12) High-speed data (AD) mode

This mode processes successive processor write requests to the non-cache area at high speed. In the AD mode, the processor proceeds to the next cycle without inserting an idle cycle after one cycle has been completed. The AD mode is selected when the AD bit of the config register is set.

Figure 4-20 shows the write cycle in the AD mode.

Figure 4-20. Write Cycle in AD Mode



4.1.5 Plural Drivers on SysAD Bus

Most of the systems using the VR4100 connect only a bidirectional transceiver to the SysAD bus between the processor and external agent. Some systems, however, add a driver and a receiver to the SysAD bus to transfer data on the SysAD bus independently of the processor. The external agent must therefore arbitrate the SysAD bus by using handshake signals. This operation is described next.

<1> Because transfer is performed on the SysAD bus independently of the processor, the external agent requests control of the system interface by using the $\overline{\text{EReq}}$ signal.

<2> After the processor has asserted the $\overline{\text{PMaster}}$ signal inactive, released the system interface, and entered the slave status, the external agent executes transfer on the SysAD bus independently of the processor. While transfer is under execution, do not assert the $\overline{\text{EValid}}$ signal active.

<3> When transfer has been completed, the external agent asserts the $\overline{\text{EValid}}$ signal active to release the system interface, and returns the processor to the master status.

To operate two or more drivers on the SysAD bus in this way, dedicated lines that control the devices other than the processor are necessary.

4.1.6 Syntax of System Interface Command

The command and data identifier of the system interface are coded in 5-bit units, and are driven to the SysCmd bus in the address/data cycle.

- When SysCmd4 is 0
  The current cycle is an address cycle, and SysCmd(3:0) indicates a command.

- **When SysCmd4 is 1**
  The current cycle is a data cycle, and SysCmd(3:0) indicates a data identifier.

The command identifier in each case is shown below.

(1) Address cycle

In the address cycle, SysCmd(4:0) is coded as follows:

Table 4-2. Coding in Address Cycle

Bit	Contents
SysCmd4	0: Address cycle
SysCmd3	Type of request 0: Read 1: Write
SysCmd2	Read/write attribute 0: Single 1: Block
SysCmd (1 : 0)	When SysCmd2 = 0 Data size for single read/write request 0: 1 byte (byte) 1: 2 bytes (half word) 2: 3 bytes 3: 4 bytes (single word)
	When SysCmd2 = 1 Block size for block read/write request 0: 2 words 1: 4 words 2: RFU 3: RFU

(2) Data cycle

In the data cycle, SysCmd(4:0) is coded as follows:

(a) Data identifier for processor request

The processor data identifier is coded as follows:

Table 4-3. Coding of Processor Data Identifier

Bit	Contents
SysCmd4	1: data cycle
SysCmd3	Indicates last data (EOD) 0: EOD 1: Other than EOD
SysCmd2	Indicates response data Fixed to 1 (other than response data)
SysCmd1	Indicates accurate data 0: No error 1: Error
SysCmd0	RFU

(b) Data identifier for external request

The external data identifier is coded as follows:

Table 4-4. Coding of External Data Identifier

Bit	Contents
SysCmd4	1: data cycle
SysCmd3	Indicates last data (EOD) 0: EOD 1: Other than EOD
SysCmd2	Indicates response data 0: Response data 1: Other than response data
SysCmd1	Indicates accurate data 0: No error 1: Error
SysCmd0	Enables data check 0: Check enabled 1: Check disabled

4.1.7 Physical Address

Physical addresses are driven to the SysAD bus in the address cycle.

Addresses for single read/write request are aligned to the size of these data elements.

Table 4-5. Address for Single Read/Write Request

Request	Address
Word	Lower 2 bits: 00
Half word	Lower 1 bit: 0
Byte	Byte address
3 bytes	Byte address

The VR4100 employs subblock ordering for block accessing.

When a block read request is issued, the address is aligned to the requested word. Response data are returned in the following sequence, in response to a processor block read request:

<1> A word including data addressed is returned.

<2> The rest of the word is returned.

For details, refer to **VR4200 User's Manual.**

Block write is always aligned at the beginning of a block.

4.2 PROCESSOR RESET AND INITIALIZATION

The VR4100 has two reset signals: cold reset ($\overline{\text{ColdReset}}$) and soft reset ($\overline{\text{Reset}}$). Setting of the necessary mode is controlled directly by pins and the config register.

4.2.1 Cold Reset

This reset operation initializes all the information of the CPU.

The following registers are set to the values shown at cold reset.

- TS and SR bits of status register ... 0
- ERL and BEV bits of status register ... 1
- Random register ... upper-limit value
- Wired register ... 0

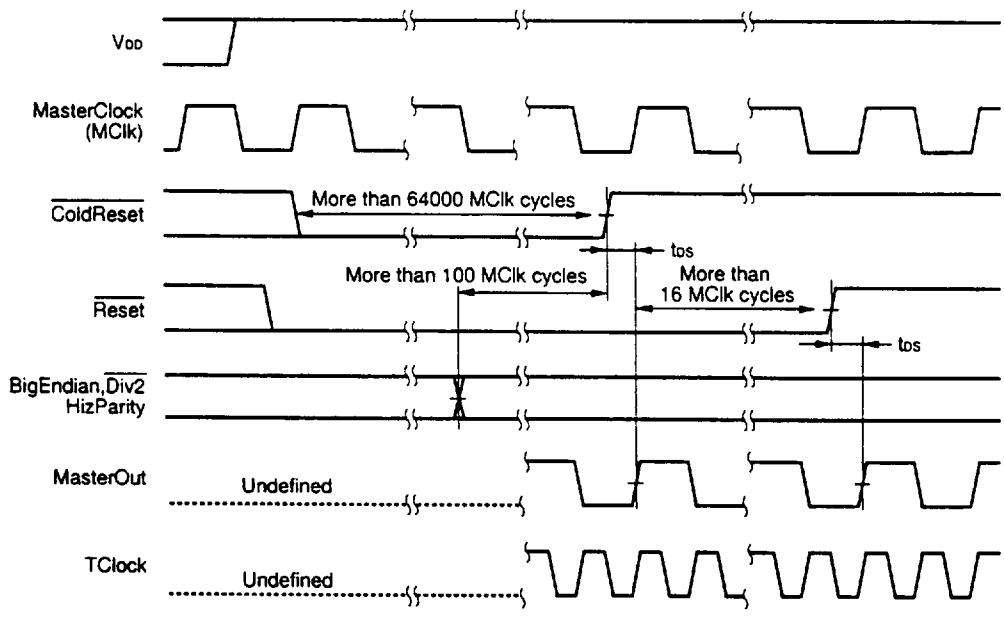
Cold reset is executed when the $\overline{\text{ColdReset}}$ and $\overline{\text{Reset}}$ signals are made active. The $\overline{\text{ColdReset}}$ signal must be active for the duration of the 64000 master clock cycle, during which various processor clocks are created from MasterClock on power application.

The status of the mode setting pins (BigEndian, $\overline{\text{Div2}}$, and HizParity) must become stable 100 master clock cycles or more before the $\overline{\text{ColdReset}}$ signal is made inactive.

The timing in which the $\overline{\text{ColdReset}}$ signal is made inactive must be synchronized with the rising of MasterClock. Keep the Reset signal active at least for 16 cycles after the $\overline{\text{ColdReset}}$ signal has been made inactive.

After the $\overline{\text{Reset}}$ signal has been asserted inactive, the CPU branches to a reset exception vector, and issues a cold reset exception. After cold reset, the CPU serves as the bus master and drives the SysAD bus.

Figure 4-21. Cold Reset Sequence



4.2.2 Soft Reset

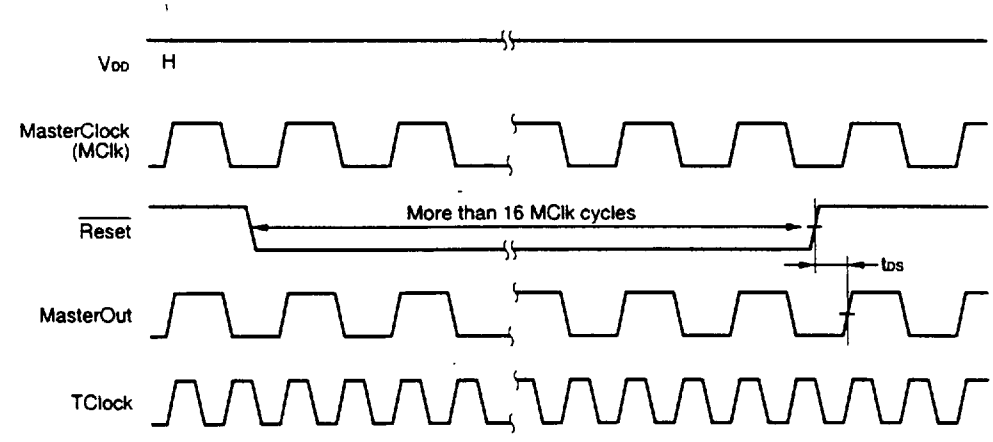
This reset operation can be executed when the $\overline{\text{Reset}}$ signal is asserted active. This reset does not perform initialization, and the status before reset is retained as much as possible. If a multi-cycle instruction is aborted by this reset, the result is undefined.

Assert the $\overline{\text{Reset}}$ signal at least for the duration of 16 cycles. Assert the signal inactive in synchronization with the MasterOut, so that the setup/hold time is satisfied.

After soft reset, TS bit of the status register is cleared to 0, SR, and BEV bits are set to 1.

Immediately after reset has been started, the CPU serves as the bus master, and starts driving data to the SysAD bus. Therefore, if the $\overline{\text{Reset}}$ signal is asserted active during SysAD processing, also reset the external agent to avoid conflict of the bus.

Figure 4-22. Soft Reset Sequence



4.2.3 Non-maskable Interrupt (NMI)

The software cannot distinguish branch to a reset exception vector caused by NMI from soft reset. NMI is acknowledged only when the processor pipeline is executed. In other words, NMI is used to recover the CPU from a software hang-up such as an infinite loop. The CPU cannot be recovered from a hardware hang-up if the external agent does not return a read response.

Because NMI does not cause a drive conflict on the SysAD bus, the external agent does not need to be reset.

4.2.4 Notes on Reset

When the reset signal is asserted, the CPU serves as the bus master and drives the SysAD bus. Therefore, exercise care in combining the reset signal of the other systems. The operation of the CPU is undefined if a bus error occurs immediately before or after, or in the middle of reset.

The VR4100 initializes a part of the status register and random/wired register after reset. Accurately initialize the CPU by software.

The VR4100 is different from the VR4200 in that it sets the basic operation mode by using the config register. However, it sets the endian and system interface division ratio by using external pins (BigEndian and $\overline{\text{Div2}}$) in the same way as the VR4200.

At cold reset, the EP, AD, and K0 bits of the config register are undefined, and the data rate is "D".

These values are not initialized by software reset or NMI.

4.3 CLOCK

The internal clock of the VR4100 is controlled by an internal phase lock loop (PLL) circuit. This PLL circuit synchronizes the internal clock of the VR4100 with the MasterClock (input clock) signal.

Stopping the output of the clock can be controlled in the power mode. For details, refer to **3.1 POWER MODES**.

Figure 4-23 shows the relation between the SysAD bus and each clock.

4.3.1 PClock (Pipeline Clock)

PClock is the basis of the internal operation. The frequency of PClock is 4 times higher than that of the Master Clock.

4.3.2 SClock (System Interface Clock)

SClock is the basis of the operation of the system interface that accesses the external agent. However, this clock does not output to the external agent.

SClock is generated by dividing PClock.

The division ratio to generate the SClock from the PClock is set by the $\overline{\text{Div2}}$ pin at cold reset.

The frequency of the SClock may be the same as that of the PClock, or half that of the PClock.

4.3.3 TClock (Transmit Clock)

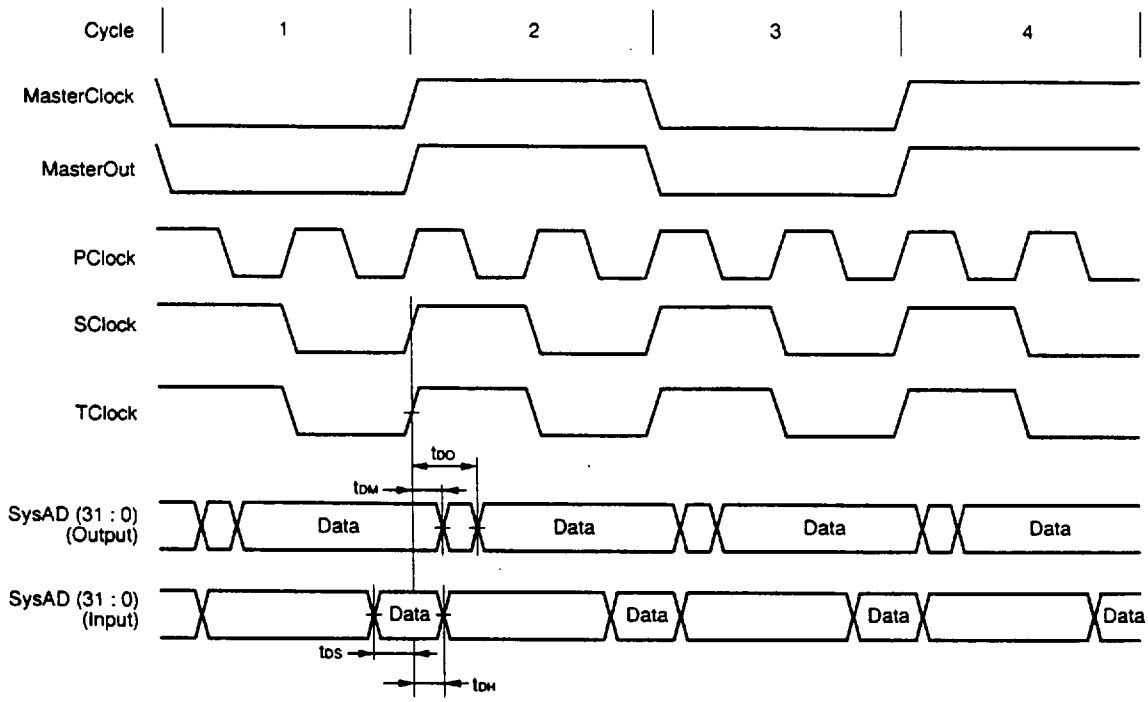
TClock is generated at the same frequency as that of SClock, and is synchronized with SClock.

This clock can be used by the external agent to drive data, and is used as the global clock of the external agent.

4.3.4 MasterOut (master clock output)

MasterOut is generated at the same frequency as MasterClock and is synchronized with MasterClock. This clock is used for the external logic that controls resets or interrupts.

Figure 4-23. Clock of VR4100 (when the $\overline{\text{Div2}}$ pin is set to low)



5. DIFFERENCES WITH THE VR4200 AND VR4400

5.1 CACHE

Item	VR4100	VR4200	VR4400
Cache size	Instruction: 2K bytes Data: 1K bytes	Instruction: 16K bytes Data: 8K bytes	Instruction: 16K bytes Data: 16K bytes
Cache line size	Instruction: 16 bytes Data: 16 bytes	Instruction: 32 bytes Data: 16 bytes	Both instruction and data: 16 or 32 bytes (selectable by software)
Cache configuration	Direct map	Direct map	Direct map
Cache index	Instruction: vAddr (10:0) Data: vAddr (9:0)	Instruction: vAddr (13:0) Data: vAddr (12:0)	vAddr (13:0)
Cache tag	pAddr (31 : 10)	pAddr (32 : 12)	pAddr (35 : 12)
Data cache write mode	Write back mode	Write back mode	Write back mode
Data ordering for block read	Subblock ordering	Subblock ordering	Subblock ordering
Data ordering for block write	Subblock ordering	Sequential ordering	Sequential ordering
Restart on instruction cache miss	When last data is received and written to cache	After all data are received and written to cache	After all data are received and written to cache
Restart on data cache miss	When last data is received and written to cache	When last double word is received	After all data are received and written to cache
Instruction cache parity	1 parity bit per 1 word	1 parity bit per 1 byte	1 parity bit per 1 byte
Data cache parity	1 parity bit per 1 byte	1 parity bit per 1 byte	1 parity bit per 1 byte

5.2 TLB

Item	VR4100	VR4200	VR4400
Instruction virtual address conversion	4-entry ITLB	2-entry ITLB	2-entry ITLB
Data virtual address conversion	4-entry DTLB	JTLB	JTLB
JTLB	32 entries (even/odd page pair)	32 entries (even/odd page pair)	48 entries (even/odd page pair)
	Full associative	Full associative	Full associative
Page size (bytes)	1K, 4K, 16K, 64K, 256K	4K, 16K, 64K, 256K, 1M, 4M, 16M	4K, 16K, 64K, 256K, 1M, 4M, 16M
Coincidence of multiple entries in JTLB	TS bit of status register is set and TLB is disabled until reset	TS bit of status register is set and TLB is disabled until reset	TS bit of status register is set and TLB is disabled until reset
Address size	VSIZE = 40 PSIZE = 32	VSIZE = 40 PSIZE = 33	VSIZE = 40 PSIZE = 36

5.3 PIPELINE

Item	VR4100	VR4200	VR4400
CPU/FPU	FPU not provided	Theoretically separated. Data bus is shared.	Completely separated
ALU delay	1 cycle	1 cycle	1 cycle
Load delay	2 cycles	2 cycles	3 cycles
Branch delay	2 cycles	2 cycles	4 cycles
Store buffer	2 double words	2 double words	2 double words
Non-cache store buffer	1 double word (1 address) Shared with write buffer	2 double words (1 address) Shared with write buffer	1 double word
Integer multiplication	Hardware for integer multiplication/division. Issued in 1 cycle	Adder and shifter are used. Issued in 12 cycles	Hardware for integer multiplication. Issued in 1 cycle
Integer division	Hardware for integer multiplication/division. Issued in 1 cycle	General-purpose adder and shifter are used. Issued in 36 cycles	Integer data bus adder is used. Issued in 69 cycles
Register for integer multiplication	Special registers HI and LO can be used simultaneously	Special registers HI and LO can be used simultaneously	Special registers HI and LO can be used simultaneously
Integer sum-of-products operation	Delay 0	None	None
Register for integer division	Special registers HI and LO can be used 1 cycle before	Special registers HI and LO can be used simultaneously	Special registers HI and LO can be used simultaneously
Hazard for special registers HI and LO	One or more 1-cycle hazards (hazard of 2 cycles for time being)	One or more 1-cycle hazards (hazard of 2 cycles for time being)	Write in early stage of pipeline
Execution cycle of MFHI/MFLO instruction	2 cycles	1 cycle	1 cycle
Execution cycle of SLLV, SRLV, and SRAC instructions	1 cycle	1 cycle	2 cycles
Execution cycle of DSLL, DSRL, DSRA, DSLL32, DSRL32, DSRA32, DSLLV, DSRLCV, DSRAC instructions	1 cycle	1 cycle	2 cycles

5.4 HARDWARE

(1/2)

Item	VR4100	VR4200	VR4400
I/O level	LVTTL (LVCMOS)	LVCMOS	LVCMOS (3 V) TTL compatible
Package	100-pin TQFP	208-pin QFP	179-pin PGA (PC) 447-pin PGA (MC)
JTAG	None	Provided	Provided
Block transfer size	Instruction: 16 bytes Data: 16 bytes	Instruction: 32 bytes Data: 16 bytes	16 or 32 bytes selectable
MasterClock input	1 to 8.25 MHz	40 MHz	100 MHz (MAX.)
SClock division ratio	1, 2	2, 3, 4	2, 3, 4, 6, 8
Non-block write cycle	Once per 2S cycles or once per 4S cycles selectable as maximum transfer quantity	Once per 3S cycles max.	Once per 4S cycles max.
Mode setting	By input from dedicated pin or software	By dedicated pins	By serial input from mode pin
Higher bits of address for read/write	Bits 63-32 are undefined	Bits 63-33 are 0	Bits 63-36 are 0
Write to non-cache area buffer	1-entry write buffer	1-entry write buffer	1-entry non-cache store
SysAD bus	32 bits	64 bits	64 bits
SysCmd bus	5 bits	9 bits	9 bits
SysADC bus	4-bit parity	8-bit parity	8-bit parity
SysAD other than data cycle	Parity	0	Parity
SysCmdP other than data cycle	Parity	0	Parity
Parity error during write back	Cache error exception used	Cache error exception used	Cache error exception used
Error bit during data identification of read response	Bus error occurs if error bit is set per word	Bus error occurs if error bit is set per double word	Bus error occurs if error bit is set per double word
Read data/parity error	Cache error exception used	Cache error exception used	Cache error exception used
Arbitration of system interface	Handshake signal: $\overline{\text{EReq}}$, $\overline{\text{PReq}}$, $\overline{\text{PMaster}}$	Handshake signal: $\overline{\text{ExtRqst}}$, $\overline{\text{Release}}$	Handshake signal: $\overline{\text{ExtRqst}}$, $\overline{\text{Release}}$
Block write	0 cycle between address/data	0 cycle between address/data	1-2 null cycles between address/data
Bus release after read request	No delay	No delay	Delay changes depending on status
SysAD bus in x cycle of write back data	Holds value of last D cycle	Holds value of last D cycle	Undefined
SysAD bus after last D cycle	Not used during subsequent x cycle	Not used during subsequent x cycle	Undefined

(2/2)

Item	VR4100	VR4200	VR4400
External interrupt	$\overline{\text{Int}}$ (4:0), $\overline{\text{NMI}}$	$\overline{\text{Int}}$ (4:0), $\overline{\text{NMI}}$	$\overline{\text{Int}}$ (5:0), $\overline{\text{NMI}}$
Output slew rate	CMOS output buffer	CMOS output buffer	Dynamic feedback control
IOOut output feedback loop output	No pin	No pin	Output slew rate control
IOIn input input	No pin	No pin	Output slew rate control

5.5 POWER MANAGEMENT

Item	VR4100	VR4200	VR4400
Low-power mode	Standby mode, Suspend mode, Hibernate mode	Power-saving mode (1/4 of normal frequency) Power-OFF mode	None

6. ELECTRICAL CHARACTERISTICS (TARGETED VALUES)

Caution The specifications shown below are targeted values only. The mass-produced model does not necessarily satisfy these specifications.

Absolute Maximum Ratings (T_A = 25 °C)

Parameter	Symbol	Condition	Rating	Unit
Supply voltage	V_{DD}		−0.5 to +4.0	V
Input voltage	V_I	$V_{DD} \geq 3.7$ V	−0.5 to +4.0	V
		$V_{DD} < 3.7$ V	−0.5 to V_{DD} + 0.3	V
Operating ambient temperature	T_A		−10 to +70	°C
Storage temperature	T_{Stg}		−65 to +150	°C

Cautions
1. Do not short circuit two or more outputs at the same time.
2. The quality of the product may be degraded if the absolute maximum rating of even one of the above parameters is exceeded, even momentarily. Absolute maximum ratings, therefore, specify the values which if exceeded may physically damage the product. Use the product never exceeding these ratings.
   The specifications and conditions shown in the following DC Characteristics and AC Characteristics are the range within which the product can normally operate and the quality can be guaranteed.

DC Characteristics (T_A = −10 to +70 °C, V_{DD} = 2.7 to 3.6 V)

Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit
High-level output voltage	V_{OH}	I_{OH} = −2 mA	0.8 V_{DD}			V
		I_{OH} = −20 μA	V_{DD}−0.1			
Low-level output voltage	V_{OL}	I_{OL} = 2 mA			0.4	V
		I_{OL} = 20 μA			0.1	
High-level input voltage[Note 1]	V_{OHC}	I_{OH} = −2 mA	0.8 V_{DD}			V
		I_{OH} = −20 μA	V_{DD}−0.1			
Low-level input voltage[Note 1]	V_{OLC}	I_{OL} = 2 mA			0.4	V
		I_{OL} = 20 μA			0.1	
High-level input voltage[Note 2]	V_{IH}		2.0		V_{DD} + 0.3	V
Low-level input voltage[Note 2]	V_{IL}		−0.3		0.3 V_{DD}	V
High-level input voltage[Note 3]	V_{KH}		0.7 V_{DD}		V_{DD} + 0.3	V
Low-level input voltage[Note 3]	V_{KL}		−0.3		0.3 V_{DD}	V
Supply current	I_{DD}	V_{DD} = 3.3 V, Tc = 0 °C		1f	2f	mA
High-level input leakage current	I_{LIH}	V_{DD} = 3.6 V, V_I = 3.6 V			5	μA
Low-level input leakage current	I_{LIL}	V_{DD} = 3.6 V, V_I = 0 V			−5	μA
High-level output leakage current	I_{LOH}	V_{DD} = 3.6 V, V_O = 3.6 V			5	μA
Low-level output leakage current	I_{LOL}	V_{DD} = 3.6 V, V_O = 0 V			−5	μA

Notes 1. Applied to the TClock pin and MasterOut pin.
2. Applied to the pins other than the MasterOut pin.
3. Applied to the MasterClock pin only.

Remarks 1. Tc : case temperature
2. f : PClock frequency (MHz)

Capacitance (T_A = 25 °C, V_{DD} = 0 V)

Parameter	Symbol	Condition	MIN.	MAX.	Unit
Input capacitance	C_{in}	fc = 1 MHz		15	pF
Output capacitance	C_{out}	Pins other than tested pin: 0 V		15	pF

AC Characteristics (T_A = −10 to +70 °C, V_{DD} = 2.7 to 3.6 V)

All the output timing is tested by the load capacity 40 pF.

Master clock and clock parameter

Parameter	Symbol	Condition	MIN.	MAX.	Unit
Master clock high-level width	t_{KKH}		50		ns
Master clock low-level width	t_{KKL}		50		ns
Master clock frequency[Note]			1	8.25	MHz
Master clock cycle	t_{CYK}		121	1000	ns
Clock jitter	t_{KKBY}			±2	ns
Master clock rise time	t_{KR}			5	ns
Master clock fall time	t_{KF}			5	ns

Note The operation of the V_R4100 is guaranteed only when the PLL is enable.

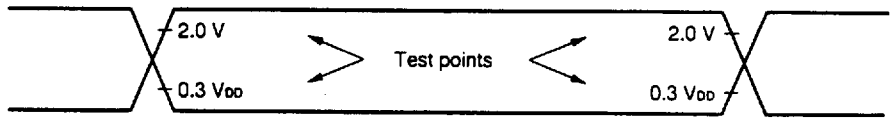
System Interface Parameter

Parameter	Symbol	Condition	MIN.	MAX.	Unit
Data output delay time	t_{DO}		3	15	ns
Data retention time	t_{DM}		3		ns
Data setup delay time	t_{DS}		6		ns
Data hold delay time	t_{DH}		1		ns
MasterOut rise time	t_{MOR}			6	ns
MasterOut fall time	t_{MOF}			6	ns
MasterOut high-level width	t_{MOH}		50		ns
MasterOut low-level width	t_{MOL}		50		ns
TClock rise time	t_{TCR}			3	ns
TClock fall time	t_{TCF}			3	ns
TClock high-level width	t_{TCH}		10		ns
TClock low-level width	t_{TCL}		10		ns

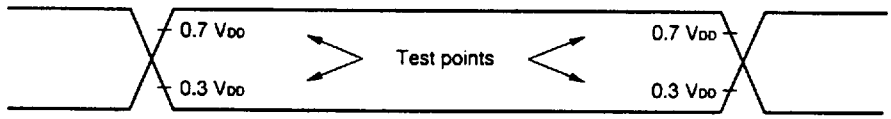
Load Coefficient

Parameter	Symbol	Condition	Rating		Unit
			MIN.	MAX.	
Load coefficient	CLD			5	ns/20 pF

AC Test Input Points (except MasterClock)

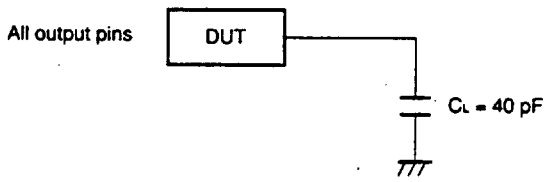


AC Test Input Points (MasterClock)



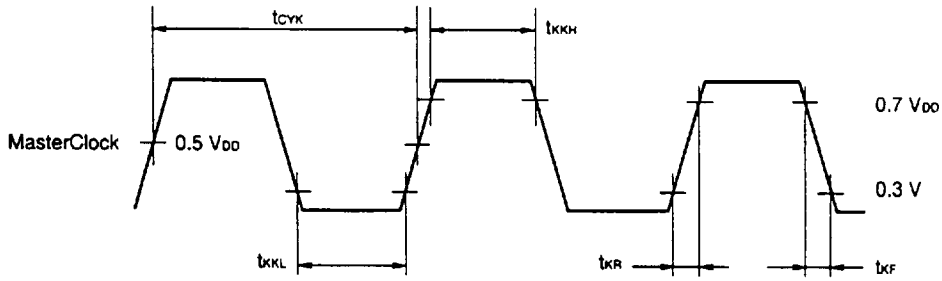
AC Test Output Points

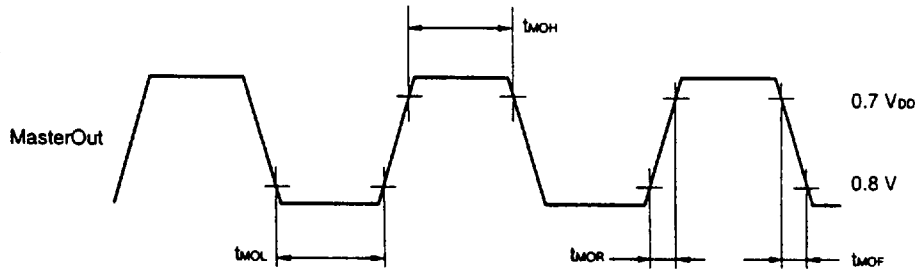
Test Load

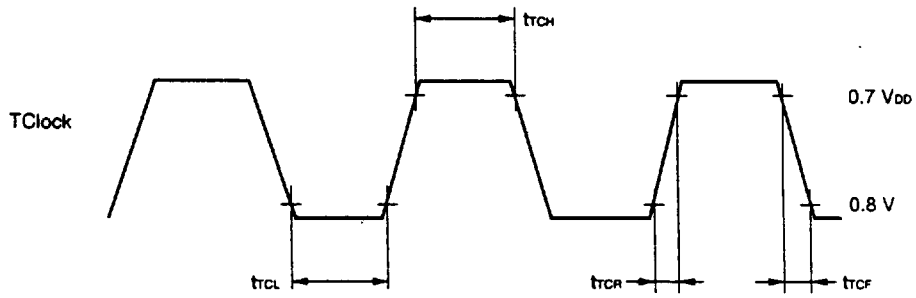


Timing Chart

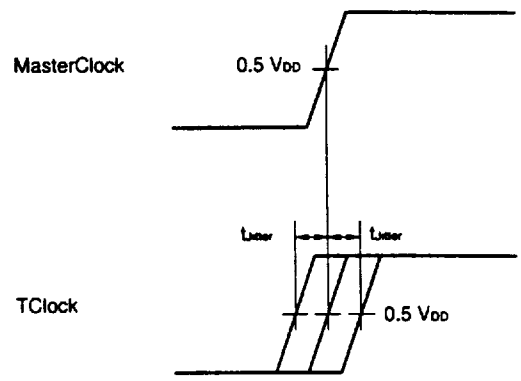
Clock timing



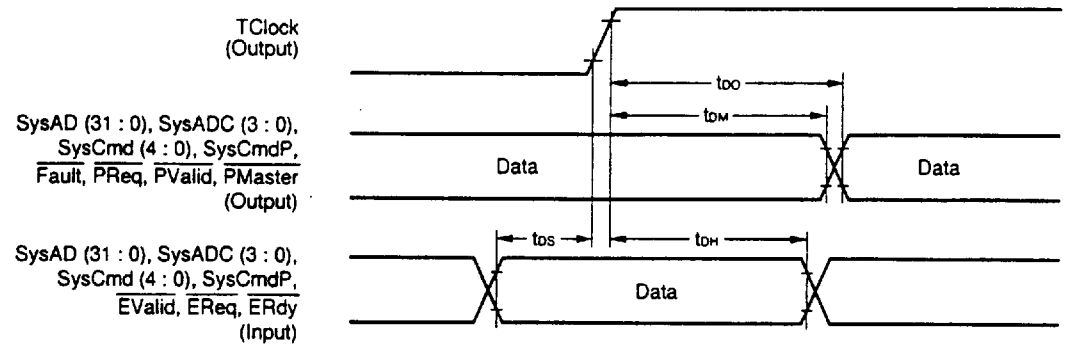




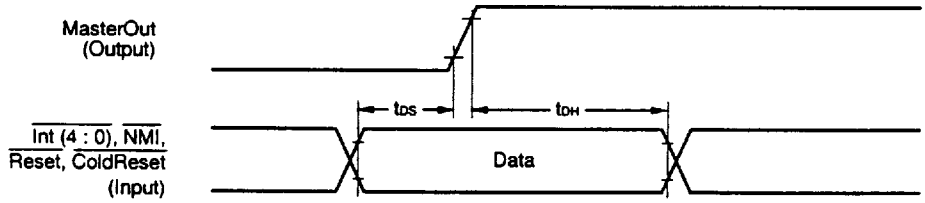
Clock jitter



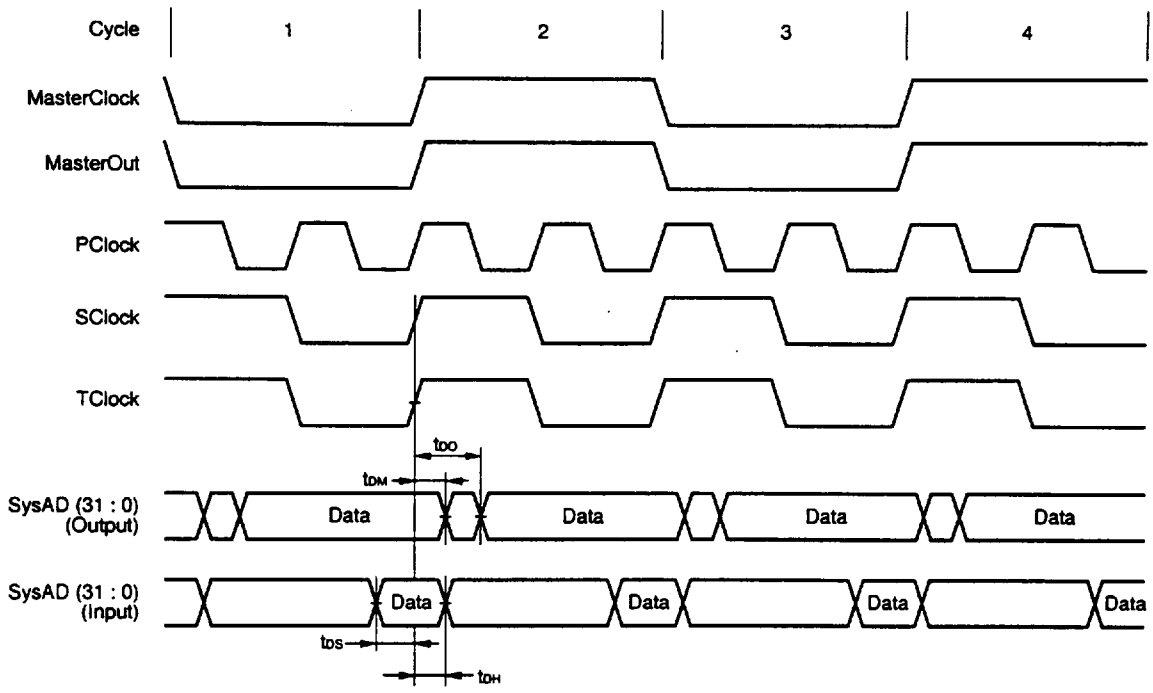
System interface edge timing



Interrupt/reset edge timing

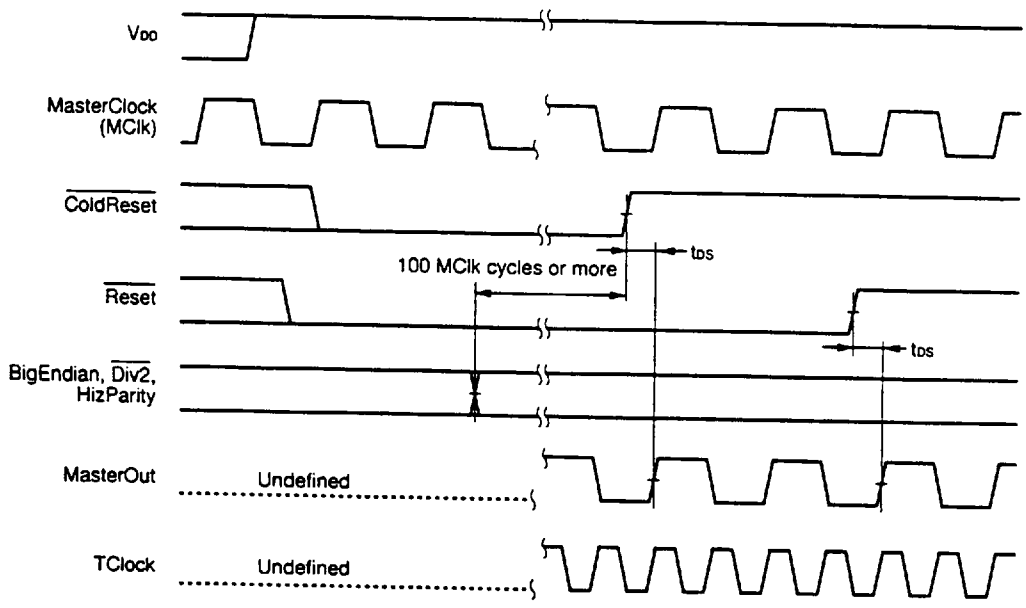


Relation between clock and system bus (when the Div2 pin is set to low)



Remark For the details or t_{DO}, t_{DM}, t_{DS}, and t_{DH}, refer to **System interface edge timing.**

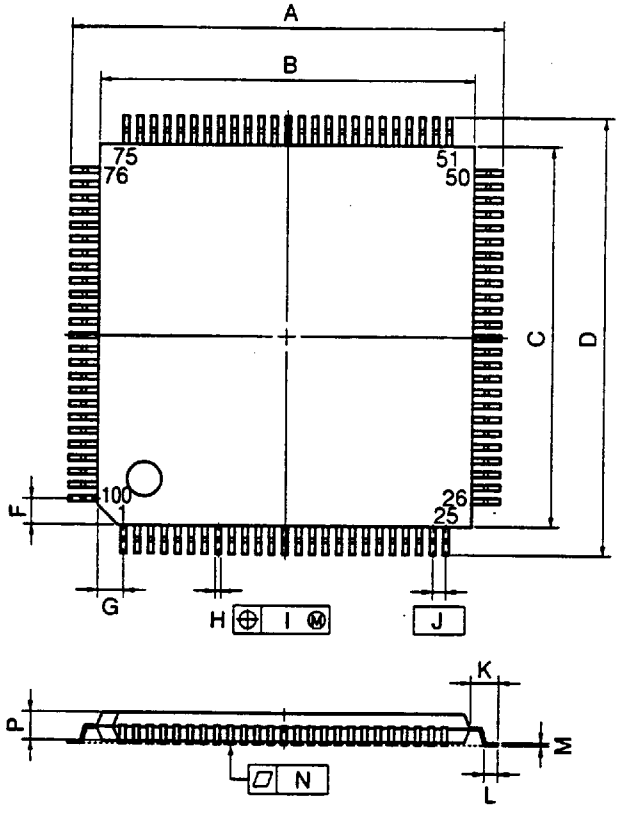
Reset timing

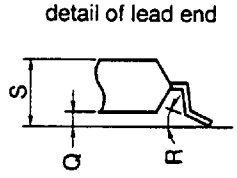


Remark For the datails on tDS, refer to **Interrupt/reset edge timing**.

7. PACKAGE DRAWING

100 PIN PLASTIC TQFP (FINE PITCH) (□14)





NOTE

Each lead centerline is located within 0.10 mm (0.004 inch) of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS	INCHES
A	16.0±0.2	0.630±0.008
B	14.0±0.2	$0.551^{+0.009}_{-0.008}$
C	14.0±0.2	$0.551^{+0.009}_{-0.008}$
D	16.0±0.2	0.630±0.008
F	1.0	0.039
G	1.0	0.039
H	$0.22^{+0.05}_{-0.04}$	0.009±0.002
I	0.10	0.004
J	0.5 (T.P.)	0.020 (T.P.)
K	1.0±0.2	$0.039^{+0.009}_{-0.008}$
L	0.5±0.2	$0.020^{+0.008}_{-0.009}$
M	$0.145^{+0.055}_{-0.045}$	0.006±0.002
N	0.10	0.004
P	1.0±0.1	$0.039^{+0.005}_{-0.004}$
Q	0.1±0.05	0.004±0.002
R	$3°^{+7°}_{-3°}$	$3°^{+7°}_{-3°}$
S	1.27 MAX.	0.050 MAX.

S100GC-50-9EU-1

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note: Strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it once, when it has occurred. Environmental control must be adequate. When it is dry, humidifier should be used. It is recommended to avoid using insulators that easily build static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work bench and floor should be grounded. The operator should be grounded using wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with semiconductor devices on it.

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note: No connection for CMOS device inputs can be cause of malfunction. If no connection is provided to the input pins, it is possible that an internal input level may be generated due to noise, etc., hence causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using a pull-up or pull-down circuitry. Each unused pin should be connected to V_{DD} or GND with a resistor, if it is considered to have a possibility of being an output pin. All handling related to the unused pins must be judged device by device and related specifications governing the devices.

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note: Power-on does not necessarily define initial status of MOS device. Production process of MOS does not define the initial operation status of the device. Immediately after the power source is turned ON, the devices with reset function have not yet been initialized. Hence, power-on does not guarantee out-pin levels, I/O settings or contents of registers. Device is not initialized until the reset signal is received. Reset operation must be executed immediately after power-on for devices having reset function.

Related Documents VR4200 User's Manual (IEU-1392)
Data Sheet (ID-3194)

Note that this document is not designated as 'preliminary', while some of the related documents are preliminary versions.

VR4000, VR4100, VR4200, VR4400, and VR Series are trademarks of NEC Corporation.

The export of this product from Japan is prohibited without governmental license. To export or re-export this product from a country other than Japan may also be prohibited without a license from that country. Please call an NEC sales representative.

This product employs technology which is restricted by the export control regulations of the United States of America. Permission of the United States government might be required in case of exporting this product or products in which this product is installed.

M4 94.11