Preliminary User's Manual

VR4131™

64/32-Bit Microprocessor

Hardware

μPD30131

Document No. U15350EJ2V0UM00 (2nd edition)
Date Published March 2002 N CP(K)

[MEMO]

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.

Exporting this product or equipment that includes this product may require a governmental license from the U.S.A. for some countries because this product utilizes technologies limited by the export control regulations of the U.S.A.

- **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.**
- **Not all devices/types available in every country. Please check with local NEC representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Corporation. NEC Corporation assumes no responsibility for any errors which may appear in this document.
- NEC Corporation does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from use of a device described herein or any other liability arising from use of such device. No license, either express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Corporation or others.
- Descriptions of circuits, software, and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software, and information in the design of the customer's equipment shall be done under the full responsibility of the customer. NEC Corporation assumes no responsibility for any losses incurred by the customer or third parties arising from the use of these circuits, software, and information.
- While NEC Corporation has been making continuous effort to enhance the reliability of its semiconductor devices, the possibility of defects cannot be eliminated entirely. To minimize risks of damage or injury to persons or property arising from a defect in an NEC semiconductor device, customers must incorporate sufficient safety measures in its design, such as redundancy, fire-containment, and anti-failure features.
- NEC devices are classified into the following three quality grades:
  "Standard", "Special", and "Specific". The Specific quality grade applies only to devices developed based on a customer designated "quality assurance program" for a specific application. The recommended applications of a device depend on its quality grade, as indicated below. Customers must check the quality grade of each device before using it in a particular application.

  Standard: Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots

  Special: Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support)

  Specific: Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems or medical equipment for life support, etc.

  The quality grade of NEC devices is "Standard" unless otherwise specified in NEC's Data Sheets or Data Books. If customers intend to use NEC devices for applications other than those specified for Standard quality grade, they should contact an NEC sales representative in advance.

M5D 98. 12

Regional Information

Some information contained in this document may vary from country to country. Before using any NEC product in your application, please contact the NEC office in your country to obtain a list of authorized representatives and distributors. They will verify:

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and other legal issues may also vary from country to country.

NEC Electronics Inc. (U.S.)
Santa Clara, California
Tel: 408-588-6000
800-366-9782
Fax: 408-588-6130
800-729-9288

NEC do Brasil S.A.
Electron Devices Division
Guarulhos-SP, Brasil
Tel: 11-6462-6810
Fax: 11-6462-6829

NEC Electronics (Europe) GmbH
Duesseldorf, Germany
Tel: 0211-65 03 01
Fax: 0211-65 03 327

- Branch The Netherlands

Eindhoven, The Netherlands
Tel: 040-244 58 45
Fax: 040-244 45 80

- Branch Sweden

Taeby, Sweden
Tel: 08-63 80 820
Fax: 08-63 80 388

NEC Electronics (France) S.A.
Vélizy-Villacoublay, France
Tel: 01-3067-58-00
Fax: 01-3067-58-99

NEC Electronics (France) S.A.
Representación en España
Madrid, Spain
Tel: 091-504-27-87
Fax: 091-504-28-60

NEC Electronics Italiana S.R.L.
Milano, Italy
Tel: 02-66 75 41
Fax: 02-66 75 42 99

NEC Electronics (UK) Ltd.
Milton Keynes, UK
Tel: 01908-691-133
Fax: 01908-670-290

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318
Fax: 2886-9022/9044

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-528-0303
Fax: 02-528-4411

NEC Electronics Shanghai, Ltd.
Shanghai, P.R. China
Tel: 021-6841-1138
Fax: 021-6841-1137

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-2719-2377
Fax: 02-2719-5951

NEC Electronics Singapore Pte. Ltd.
Novena Square, Singapore
Tel: 253-8311
Fax: 250-3583

J02.3

PREFACE

Readers

This manual is intended for users who wish to gain an understanding of the functions of the VR4131 in order to design and develop its application systems and programs.

Purpose

This manual is intended to give users an understanding of the hardware of the VR4131 described in the **Organization** below.

Organization

Two manuals are available for the VR4131: Hardware User's Manual (this manual) and Architecture User's Manual (**VR4100 Series™ Architecture User's Manual**).

Hardware User's Manual	Architecture User's Manual
• Pin functions	• Pipeline operation
• Physical address space	• Cache and memory management system
• Function of coprocessor 0	• Exception handling
• Initialization interface	• Interrupts
• Peripheral units	• Instruction sets

How to Read This Manual

It is assumed that the readers of this manual have general knowledge of electrical engineering, logic circuits, and microcontrollers.

◊ To understand the overall functions of the VR4131
→ Read this manual in the order of the **CONTENTS**.
◊ To learn details of the instruction sets of the VR4131
→ Refer to **VR4100 Series Architecture User's Manual** available separately.
◊ To learn the electrical specifications of the VR4131
→ Refer to the **Data Sheet** available separately.

Conventions

Data significance:	Higher digits on the left and lower digits on the right
Active low representation:	XXX# (# suffixed to pin and signal names)
Note:	Footnote for item marked with **Note** in the text
Caution:	Information requiring particular attention
Remark:	Supplementary information
Numerical representation:	Binary/Decimal ... XXXX
	Hexadecimal ... 0xXXXX

Prefix indicating the power of 2 (address space, memory capacity):

K (kilo)	2^{10} = 1,024
M (mega)	2^{20} = $1{,}024^{2}$
G (giga)	2^{30} = $1{,}024^{3}$
T (tera)	2^{40} = $1{,}024^{4}$
P (peta)	2^{50} = $1{,}024^{5}$
E (exa)	2^{60} = $1{,}024^{6}$

Related Documents

When using this manual, also read the following documents.

The related documents indicated in this publication may include preliminary versions. However preliminary versions are not marked as such.

Documents Related to Devices

Document Name	Document No.
VR4131 Hardware User's Manual	This Manual
μPD30131 (VR4131) Data Sheet	To be prepared
VR4100 Series Architecture Use's Manual	To be prepared
VRC4173™ User's Manual	U14579E
μPD31173 (VRC4173) Data Sheet	U15338E

Application Note

Document Name	Document No.
VR Series™ Application Note Programming Guide	U10710E

CONTENTS

LIST OF FIGURES (1/3)

Figure No. Title Page

LIST OF FIGURES (2/3)

Figure No. Title Page

LIST OF FIGURES (3/3)

LIST OF TABLES (1/3)

LIST OF TABLES (2/3)

Table No. Title Page

LIST OF TABLES (3/3)

CHAPTER 1 INTRODUCTION

This chapter gives an outline of the VR4131 (µPD30131), which is a 64-/32-bit RISC microprocessor.

1.1 Features

The VR4131, which is a high-performance 64-/32-bit microprocessor employing the RISC (reduced instruction set computer) architecture developed by MIPS™, is one of the RISC microprocessor VR Series products manufactured by NEC.

The VR4131 accommodates the ultra low power consumption VR4130™ CPU core provided with cache memory, a high-speed product-sum operation unit, and an address management unit. The VR4131 also has interface units for the peripheral circuits required for battery-driven portable information equipment, such as a DMA interface, a serial interface, an IrDA interface, a real-time clock interface, interface for memories such as synchronous DRAM and Page ROM, an NS16550-compatible serial interface, and a 3-wire clocked serial interface. This processor has a 32-bit external memory bus width and supports the PCI bus interface conforming to Rev2.1 as the interface for an external device that requires the performance level of a color LCD controller.

The VR4131 is compatible with the instruction set architecture (ISA) of MIPS I, MIPS II, MIPS III, and MIPS16. However, it does not support LL, LLD, SC, SCD, and floating-point instructions.

Note that this processor does not incorporate a secondary cache, multi-processor processing function, or floating point arithmetic function.

The features of the VR4131 are described below.

- Employs 64-bit RISC VR4130 CPU Core in 2-way superscalar architecture
- Internal 64-bit data processing
- Two optimized 6-stage pipelines allowing two instructions to be executed simultaneously
- Conforms to MIPS I, II, III instruction sets (without the floating point instructions and the LL, LLD, SC, and SCD instructions, but with the addition of the MACC instruction)
- Supports MIPS16 instructions
- Supports high-speed product-sum operations (MACC instruction) to implement the execution of applications at high speed
- On-chip 2-way set associative cache memory Instruction cache: 16 KB
  Data cache: 16 KB
- Translation lookaside buffer (TLB) for virtual address management
- Address space Physical address space: 32 bits
  Virtual address space: 40 bits
- On-chip peripheral units ideal for portable equipment
  - Memory controller (supports ROM, synchronous DRAM (SDRAM), and flash memory)
  - Supports PCI bus interface conforming to PCI Rev2.1
  - Supports interface with companion chip VRC4173
  - Controller complying with IrDA 1.1 (FIR)
  - DMA controller
  - 3-wire clocked serial interface
  - Debug serial interfaces
  - Interrupt controller
  - General-purpose ports

○ Effective power management features, which include the following five operating modes:
- Fullspeed mode: Normal operating mode in which all clocks operate
- Standby mode: All internal clocks stop except for timer or interrupt-related clocks
- Suspend mode: Bus clock and all internal clocks stop except for timer or interrupt-related clocks
- Exsuspend mode: Suspend mode in which the PLL operation is also stopped
- Hibernate mode: All clocks generated by the CPU core stop

○ External clock input: 32.768 kHz, 18.432 MHz (for internal CPU core and peripheral unit operation), 48 MHz (dedicated for IrDA interface)

○ Clock supply management function for each on-chip peripheral unit to implement low-power consumption

○ Operating supply voltage: $V_{DD}1$ = 1.40 to 1.65 V (internal), $V_{DD}3$ = 3.0 to 3.6 V (external)

1.2 Ordering Information

Part Number	Package	Internal Maximum Operation Frequency
μPD30131F1-200-GA2	224-pin plastic FBGA (16 × 16)	200 MHz

1.3 64-Bit Architecture

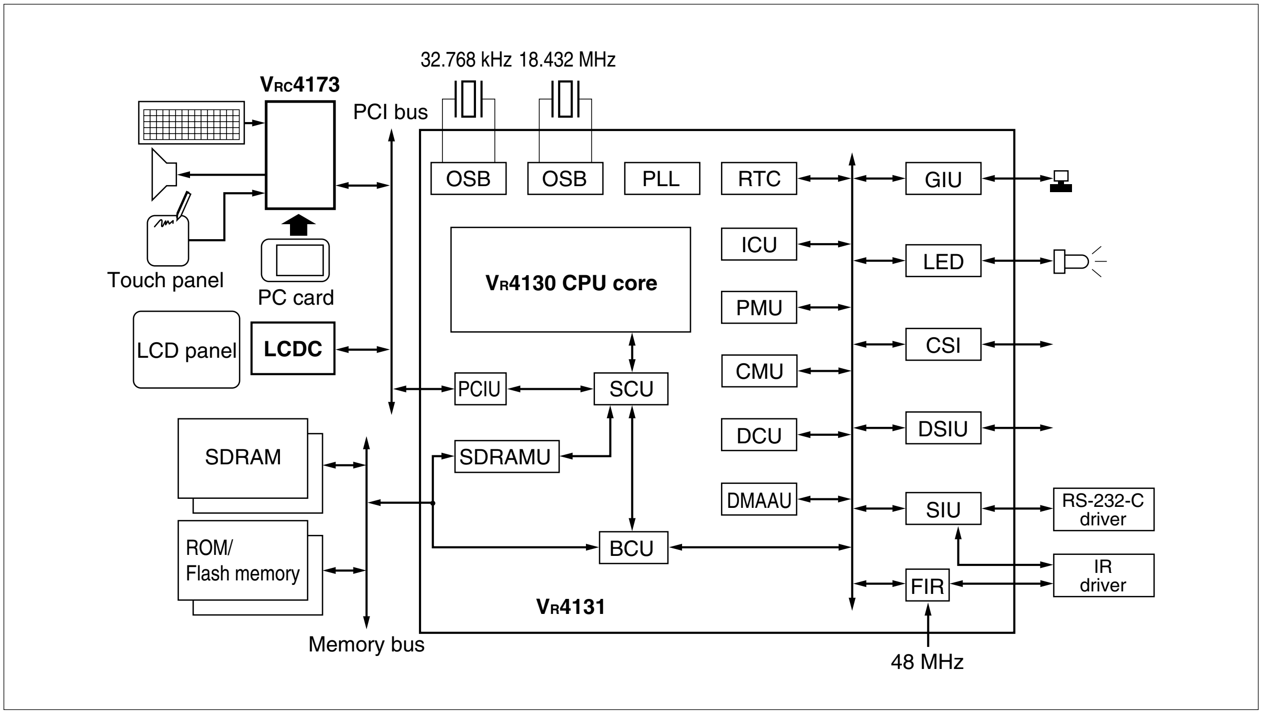
The VR4131 is a high-performance 64-bit microprocessor. However, it can run 32-bit applications even if it operates as a 64-bit microprocessor.

1.4 VR4131 Processor

The VR4131 consists of the VR4130 CPU core and 17 peripheral units. It can connect external controllers directly.

Figure 1-1 shows an internal block diagram of the VR4131 processor and an example of connection to external blocks.

Figure 1-1. VR4131 Internal Block Diagram and Example of Connection to External Blocks



1.4.1 Internal block structure

The following is an outline of the peripheral units. For the CPU core, refer to **1.5 VR4130 CPU Core**.

(1) Bus control unit (BCU)

In the VR4131, the BCU transfers data between the VR4130 CPU core and the SysAD bus. It also controls the ROM (flash memory or mask ROM) connected to the system bus and transfers data to the above devices through the ADD bus and DATA bus.

(2) Real-time clock unit (RTC)

The RTC is provided with an accurate counter that operates on a 32.768 kHz clock pulse supplied from the clock generator. It is also provided with several counters and compare registers for controlling various interrupts.

(3) Interrupt control unit (ICU)

The ICU controls interrupt requests that are generated by internal or external sources of the VR4131, and informs the VR4130 CPU core of an interrupt request.

(4) Power management unit (PMU)

The PMU outputs signals necessary to control the power of the entire system including the VR4131. The signals are used to control the PLL of the VR4130 CPU core and the internal clocks (PClock, VTClock, and MasterOut) in low-power modes.

(5) Direct memory access address unit (DMAAU)

The DMAAU controls the address of the following three different DMA transfers.

- DMA between PCI bus and memory bus
- DMA between on-chip CPU I/O and memory bus
- DMA between general-purpose I/O device and memory

(6) Direct memory access control unit (DCU)

The DCU controls the arbitration of three different DMA transfers.

(7) Clock mask unit (CMU)

The CMU controls the way the clocks (VTClock and MasterOut) are supplied from the VR4130 CPU core to internal peripheral units.

(8) General-purpose I/O unit (GIU)

The GIU controls 35 general-purpose I/O pins.

(9) SysAD control unit (SCU)

The SCU controls bus arbitration so that two or more masters can be connected on the SysAD bus.

(10) Synchronous DRAM interface unit (SDRAMU)

The SDRAMU controls the synchronous DRAM connected to the system bus and transfers data to the above devices through the ADD bus and DATA bus.

(11) PCI bus interface unit (PCIU)

The PCIU conforming to Rev2.1 controls the PCI bus and enables connection with a device provided with the PCI interface such as an LCD controller. It supports power control via the CLKRUN signal.

(12) Debug serial interface unit (DSIU)

The DSIU is a serial interface for debugging. It supports a maximum transfer rate of 115 kbps.

(13) Serial interface unit (SIU)

The SIU conforms to the RS-232C specification and is compatible with NS16550. It supports a maximum transfer rate of 1.15 Mbps. Also available is an IrDA serial interface supporting a maximum transfer rate of 4 Mbps through the FIR unit, but this interface and the RS-232C interface are mutually exclusive.

(14) Fast IrDA interface unit (FIR)

The FIR unit, which conforms to IrDA 1.1, performs 0.5 to 4 Mbps IrDA communication. This unit operates based on a dedicated 48 MHz clock input.

(15) Clocked serial interface unit (CSI)

The CSI unit controls a 3-wire clocked serial interface.

(16) Light emitting diode unit (LED)

The LED unit is used to control the lighting of an external LED.

1.4.2 I/O registers

The I/O registers are used for peripheral unit control. Lists of registers for peripheral units are as follows.

Table 1-1. BCU Registers

Register Symbol	Function	Physical Address
BCUCNTREG1	BCU control register 1	0x0F00 0000
ROMSIZEREG	ROM size register	0x0F00 0004
ROMSPEEDREG	BCU access cycle change register	0x0F00 0006
IO0SPEEDREG	I/O access cycle change register 0	0x0F00 0008
IO1SPEEDREG	I/O access cycle change register 1	0x0F00 000A
REVIDREG	Peripheral unit revision ID register	0x0B00 0010
CLKSPEEDREG	Clock select register	0x0B00 0014
BCUCNTREG3	BCU control register 3	0x0B00 0016

Table 1-2. DMAAU Registers

Register Symbol	Function	Physical Address
CSIIBALREG	CSI reception DMA base address lower register	0x0F00 0020
CSIIBAHREG	CSI reception DMA base address higher register	0x0F00 0022
CSIIALREG	CSI reception DMA address lower register	0x0F00 0024
CSIIAHREG	CSI reception DMA address higher register	0x0F00 0026
CSIOBALREG	CSI transmission DMA base address lower register	0x0F00 0028
CSIOBAHREG	CSI transmission DMA base address higher register	0x0F00 002A
CSIOALREG	CSI transmission DMA address lower register	0x0F00 002C
CSIOAHREG	CSI transmission DMA address higher register	0x0F00 002E
FIRBALREG	FIR DMA base address lower register	0x0F00 0030
FIRBAHREG	FIR DMA base address higher register	0x0F00 0032
FIRALREG	FIR DMA address lower register	0x0F00 0034
FIRAHREG	FIR DMA address higher register	0x0F00 0036
RAMBALREG	RAM base address lower address between I/O space and RAM	0x0F00 01E0
RAMBAHREG	RAM base address higher address between I/O space and RAM	0x0F00 01E2
RAMALREG	RAM address lower address between I/O space and RAM	0x0F00 01E4
RAMAHREG	RAM address higher address between I/O space and RAM	0x0F00 01E6
IOBALREG	I/O base address lower address between I/O space and RAM	0x0F00 01E8
IOBAHREG	I/O base address higher address between I/O space and RAM	0x0F00 01EA
IOALREG	I/O address lower address between I/O space and RAM	0x0F00 01EC
IOAHREG	I/O address higher address between I/O space and RAM	0x0F00 01EE

Table 1-3. DCU Registers

Register Symbol	Function	Physical Address
DMARSTREG	DMA reset register	0x0B00 0040
DMAIDLEREG	DMA sequencer status register	0x0B00 0042
DMASENREG	DMA sequencer enable register	0x0B00 0044
DMAMSKREG	DMA mask register	0x0B00 0046
DMAREQREG	DMA request register	0x0B00 0048
TDREG	Transfer direction setting register	0x0B00 004A
DMAABITREG	DMA arbitration protocol selection register	0x0F00 004C
CONTROLREG	DMA control register	0x0F00 004E
BASSCNTLREG	DMA transfer byte size lower register	0x0F00 0050
BASSCNTHREG	DMA transfer bye size higher register	0x0F00 0052
CURRENTCNTLREG	DMA remaining transfer byte size lower register	0x0F00 0054
CURRENTCNTHREG	DMA remaining transfer byte size higher register	0x0F00 0056
TCINTREG	Terminal count interrupt request register	0x0F00 0058

Table 1-4. CMU Register

Register Symbol	Function	Physical Address
CMUCLKMSK	CMU clock mask register	0x0F00 0060

Table 1-5. ICU Registers

Register Symbol	Function	Physical Address
SYSINT1REG	System interrupt register 1 (level 1)	0x0B00 0080
GIUINTLREG	GIU interrupt lower register (level 2)	0x0B00 0088
DSIUINTREG	DSIU interrupt register (level 2)	0x0B00 008A
MSYSINT1REG	System interrupt mask register 1 (level 1)	0x0B00 008C
MGIUINTLREG	GIU interrupt mask lower register (level 2)	0x0B00 0094
MDSIUINTREG	DSIU interrupt mask register (level 2)	0x0B00 0096
NMIREG	Battery interrupt select register	0x0B00 0098
SOFTINTREG	Software interrupt register	0x0B00 009A
SYSINT2REG	System interrupt register 2 (level 1)	0x0F00 00A0
GIUINTHREG	GIU interrupt higher register (level 2)	0x0F00 00A2
FIRINTREG	FIR interrupt register (level 2)	0x0F00 00A4
MSYSINT2REG	System interrupt mask register 2 (level 1)	0x0F00 00A6
MGIUINTHREG	GIU interrupt mask higher register (level 2)	0x0F00 00A8
MFIRINTREG	FIR interrupt mask register (level 2)	0x0F00 00AA
PCIINTREG	PCI interrupt register (level 2)	0x0F00 00AC
SCUINTREG	SCU interrupt register (level 2)	0x0F00 00AE
CSIINTREG	CSI interrupt register (level 2)	0x0F00 00B0
MPCIINTREG	PCI interrupt mask register (level 2)	0x0F00 00B2
MSCUINTREG	SCU interrupt mask register (level 2)	0x0F00 00B4
MCSIINTREG	CSI interrupt mask register (level 2)	0x0F00 00B6
BCUINTREG	BCU interrupt register (level 2)	0x0F00 00B8
MBCUINTREG	BCU interrupt mask register (level 2)	0x0F00 00BA

Table 1-6. PMU Registers

Register Symbol	Function	Physical Address
PMUINTREG	PMU interrupt/status register	0x0F00 00C0
PMUCNTREG	PMU control register	0x0F00 00C2
PMUINT2REG	PMU interrupt/status register 2	0x0F00 00C4
PMUCNT2REG	PMU control register 2	0x0F00 00C6
PMUWAITREG	PMU wait count register	0x0F00 00C8
PMUCLKDIVREG	PMU Div mode register	0x0F00 00CC
PMUINTRCLKDIVREG	PMU INT clock Div mode register	0x0F00 00CE

Table 1-7. RTC Registers

Register Symbol	Function	Physical Address
ETIMELREG	Elapsed Time timer lower register	0x0F00 0100
ETIMEMREG	Elapsed Time timer lower register	0x0F00 0102
ETIMEHREG	Elapsed Time timer lower register	0x0F00 0104
ECMPLREG	Elapsed Time timer compare lower register	0x0F00 0108
ECMPMREG	Elapsed Time timer compare middle register	0x0F00 010A
ECMPHREG	Elapsed Time timer compare higher register	0X0F00 010C
RTCL1LREG	RTC Long1 timer lower register	0x0F00 0110
RTCL1HREG	RTC Long1 timer higher register	0x0F00 0112
RTCL1CNTLREG	RTC Long1 timer count lower register	0x0F00 0114
RTCL1CNTHREG	RTC Long1 timer count higher register	0x0F00 0116
RTCL2LREG	RTC Long2 timer lower register	0x0F00 0118
RTCL2HREG	RTC Long2 timer higher register	0x0F00 011A
RTCL2CNTLREG	RTC Long2 timer count lower register	0x0F00 011C
RTCL2CNTHREG	RTC Long2 timer count higher register	0x0F00 011E
TCLKLREG	TClock counter lower register	0x0F00 0120
TCLKHREG	TClock counter higher register	0x0F00 0122
TCLKCNTLREG	TClock counter count lower register	0x0F00 0124
TCLKCNTHREG	TClock counter count higher register	0x0F00 0126
RTCINTREG	RTC interrupt register	0x0F00 013E

Table 1-8. GIU Registers

Register Symbol	Function	Physical Address
GIUIOSELL	GPIO input/output select lower register	0x0F00 0140
GIUIOSELH	GPIO input/output select higher register	0x0F00 0142
GIUPIODL	GPIO port input/output data lower register	0x0F00 0144
GIUPIODH	GPIO port input/output data higher register	0x0F00 0146
GIUINTSTATL	GPIO interrupt status lower register	0x0F00 0148
GIUINTSTATH	GPIO interrupt status higher register	0x0F00 014A
GIUINTENL	GPIO interrupt enable lower register	0x0F00 014C
GIUINTENH	GPIO interrupt enable higher register	0x0F00 014E
GIUINTTYPL	GPIO interrupt trigger select lower register	0x0F00 0150
GIUINTTYPH	GPIO interrupt trigger select higher register	0x0F00 0152
GIUINTALSELL	GPIO interrupt level select lower register	0x0F00 0154
GIUINTALSELH	GPIO interrupt level select higher register	0x0F00 0156
GIUINTHTSELL	GPIO interrupt hold select lower register	0x0F00 0158
GIUINTHTSELH	GPIO interrupt hold select higher register	0x0F00 015A
GIUPODATEN	GPIO output data enable register	0x0F00 015C
GIUPODATL	GPIO output data lower register	0x0F00 015E

Table 1-9. SCU Registers

Register Symbol	Function	Physical Address
TIMOUTCNTREG	Timeout value setting register	0x0F00 1000
TIMOUTCOUNTREG	Timeout counter register	0x0F00 1002
ERRLADDRESSREG	Error lower address register	0x0F00 1004
ERRHADDRESSREG	Error higher address register	0x0F00 1006
SCUINTRREG	SCU interrupt request register	0x0F00 1008

Table 1-10. SDRAMU Registers

Register Symbol	Function	Physical Address
SDRAMMODEREG	SDRAM mode register	0x0F00 0400
SDRAMCNTREG	SDRAM control register	0x0F00 0402
BCURFCNTREG	BCU refresh control register	0x0F00 0404
BCURFCOUNTREG	BCU refresh cycle count register	0x0F00 0406
RAMSIZEREG	DRAM size register	0x0F00 0408

Table 1-11. PCI Internal Registers

Register Symbol	Function	Physical Address
PCIMMAW1REG	Master transaction PCI memory space address conversion register 1	0x0F00 0C00
PCIMMAW2REG	Master transaction PCI memory space address conversion register 2	0x0F00 0C04
PCITAW1REG	Target transaction internal memory space conversion register 1	0x0F00 0C08
PCITAW2REG	Target transaction internal memory space conversion register 2	0x0F00 0C0C
PCIMIOAWREG	Master transaction PCI I/O space address conversion register	0x0F00 0C10
PCICONFDREG	Configuration access address register	0x0F00 0C14
PCICONFAREG	Configuration access data register	0x0F00 0C18
PCIMAILREG	Mailbox register	0x0F00 0C1C
BUSERRADREG	Bus error generation address save register	0x0F00 0C24
INTCNTSTAREG	Interrupt status indicate and control register	0x0F00 0C28
PCIEXACCREG	Exclusive access indicate and control register	0x0F00 0C2C
PCIRECONTREG	Retry count register	0x0F00 0C30
PCIENREG	Configuration register setting end register	0x0F00 0C34
PCICLKSELREG	PCI bus operation clock select register	0x0F00 0C38
PCITRDYVREG	TRDY signal select register	0x0F00 0C3C
PCICLKRUNREG	CLK_RUN signal select register	0x0F00 0C60

Table 1-12. PCI Configuration Header Registers

Register Symbol	Function	Physical Address
VENDORIDREG	Vendor ID register	0x0F00 0D00
DEVICEIDREG	Device ID register	
COMMANDREG	PCI interface setting command register	0x0F00 0D04
STATUSREG	PCI interface status register	
REVIDREG	Device revision register	0x0F00 0D08
CLASSREG	Device class register	
CACHELSREG	System cache line size register	0x0F00 0D0C
LATTIMEREG	Timer count value setting register	
MAILBAREG	Base address register of Mailbox register	0x0F00 0D10
PCIMBA1REG	PCI memory space base address register 1	0x0F00 0D14
PCIMBA2REG	PCI memory space base address register 2	0x0F00 0D18
INTLINEREG	PCI interrupt line register	0x0F00 0D3C
INTPINREG	PCI interrupt pin register	
RETVALREG	Retry limit value setting register	0x0F00 0D40
PCIAPCNTREG	PCI arbitration protocol select register	

Table 1-13. DSIU Registers

Register Symbol	Function	LCR7 Bit	Physical Address
DSIURB	Receive buffer register (read)	0	0x0F00 0820
DSIUTH	Transmission hold register (write)		
DSIUDLL	Division ratio lower register	1	
DSIUIE	Interrupt enable register	0	0x0F00 0821
DSIUDLM	Division ratio higher register	1	
DSIUIID	Interrupt indication register (read)	—	0x0F00 0822
DSIUFC	FIFO control register (write)	—	
DSIULC	Line control register	—	0x0F00 0823
DSIUMC	Modem control register	—	0x0F00 0824
DSIULS	Line status register	—	0x0F00 0825
DSIUMS	Modem status register	—	0x0F00 0826
DSIUSC	Scratch register	—	0x0F00 0827
SIURESET	SIU reset register	—	0x0F00 0809

Remark LCR7 is the bit 7 of DSIULC register.

Table 1-14. LED Registers

Register Symbol	Function	Physical Address
LEDHTSREG	LED on time setting register	0x0F00 0180
LEDLTSREG	LED off time setting register	0x0F00 0182
LEDCNTREG	LED control register	0x0F00 0188
LEDASTCREG	LED auto stop time count register	0x0F00 018A
LEDINTREG	LED interrupt register	0x0F00 018C

Table 1-15. SIU Registers

Register Symbol	Function	LCR 7	Physical Address
SIURB	Receive buffer register (read)	0	0x0F00 0800
SIUTH	Transmission hold register (write)		
SIUDLL	Division ratio lower register	1	
SIUIE	Interrupt enable register	0	0x0F00 0801
SIUDLM	Division ratio higher register	1	
SIUIID	Interrupt indication register (read)	–	0x0F00 0802
SIUFC	FIFO control register (write)	–	
SIULC	Line control register	–	0x0F00 0803
SIUMC	Modem control register	–	0x0F00 0804
SIULS	Line status register	–	0x0F00 0805
SIUMS	Modem status register	–	0x0F00 0806
SIUSC	Scratch register	–	0x0F00 0807
SIUIRSEL	Serial communication select register	–	0x0F00 0808
SIURESET	SIU reset register	–	0x0F00 0809
SIUCSEL	SIU echo-back control register	–	0x0F00 080A

Remark LCR7 is bit 7 of the SIULC register.

Table 1-16. CSI Registers

Register Symbol	Function	Physical Address
CSI_MODEREG	CSI mode register	0x0F00 01A0
CSI_CLKSELREG	CSI clock select register	0x0F00 01A1
CSI_SIRBREG	CSI receive data buffer register	0x0F00 01A2
CSI_SOTBREG	CSI transmit data buffer register	0x0F00 01A4
CSI_SIRBEREG	CSI receive data buffer register (read emulation)	0x0F00 01A6
CSI_SOTBFREG	CSI first transmit data buffer register	0x0F00 01A8
CSI_SIOREG	CSI shift register	0x0F00 01AA
CSI_CNTREG	CSI control register	0x0F00 01B0
CSI_INTREG	CSI interrupt register	0x0F00 01B2
CSI_IFIFOVREG	CSI reception FIFO valid register	0x0F00 01B4
CSI_OFIFOVREG	CSI transmission FIFO valid register	0x0F00 01B6
CSI_IFIFOREG	CSI receive register	0x0F00 01B8
CSI_OFIFOREG	CSI transmit register	0x0F00 01BA
CSI_FIFOTRGREG	CSI FIFO trigger level register	0x0F00 01BC

Table 1-17. FIR Registers

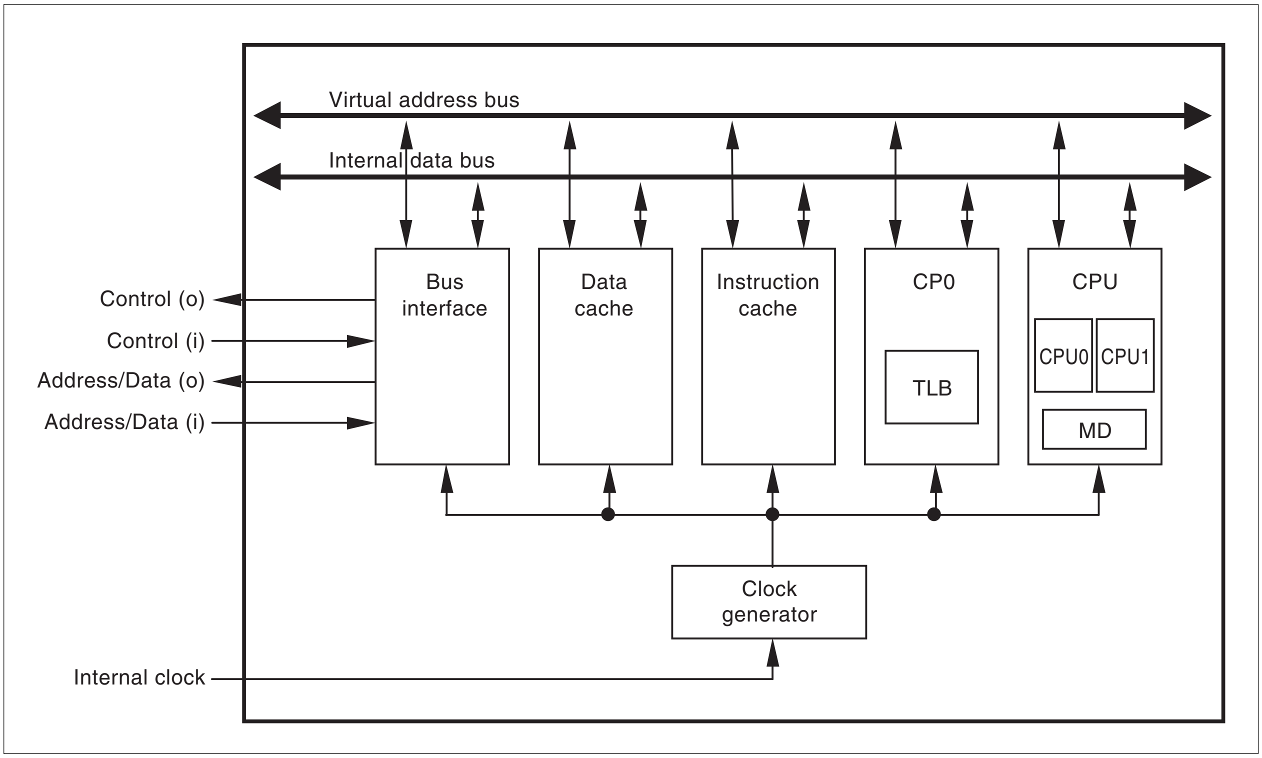
Register Symbol	Function	Physical Address
FRSTR	FIR reset register	0x0F00 0840
DPINTR	DMA page interrupt register	0x0F00 0842
DPCNTR	DMA page control register	0x0F00 0844
TDR	Transmit data register	0x0F00 0850
RDR	Receive data register	0x0F00 0852
IMR	Interrupt mask register	0x0F00 0854
FSR	FIFO setup register	0x0F00 0856
IRSR1	IR setup register 1	0x0F00 0858
CRCSR	CRC setup register	0x0F00 085C
FIRCR	FIR control register	0x0F00 085E
MIRCR	MIR control register	0x0F00 0860
DMACR	DMA control register	0x0F00 0862
DMAER	DMA enable register	0x0F00 0864
TXIR	Transmission indication register	0x0F00 0866
RXIR	Reception indication register	0x0F00 0868
IFR	Interrupt flag register	0x0F00 086A
RXSTS	Reception status register	0x0F00 086C
TXFL	Transmit frame length register	0x0F00 086E
MRXF	Maximum receive frame length register	0x0F00 0870
RXFL	Receive frame length register	0x0F00 0874

1.5 VR4130 CPU Core

Figure 1-2 shows the internal block diagram of the VR4130 CPU core.

The VR4130 core employs 2-way superscalar architecture. In addition to the conventional high-performance integer operation units, this CPU core has a full-associative format translation lookaside buffer (TLB), which has 32 entries that provide mapping to 2-page pairs for one entry. Moreover, it also has instruction and data caches, and a bus interface.

Figure 1-2. VR4130 CPU Core Internal Block Diagram



1.5.1 Internal block configuration

(1) CPU

CPU is a block that performs integer calculations. This block includes a 64-bit integer data path, and product-sum operator.

(2) Coprocessor 0 (CP0)

CP0 incorporates a memory management unit (MMU) and exception handling function. The MMU checks whether there is an access between different memory segments (user, supervisor, and kernel) by executing address conversion. The translation lookaside buffer (TLB) converts virtual addresses to physical addresses.

(3) Instruction cache

The instruction cache employs 2-way set associative, virtual index, and physical tag formats. Its capacity is 16 KB.

(4) Data cache

The data cache employs 2-way set associative, virtual index, physical tag, and writeback. Its capacity is 16 KB.

(5) Bus interface

The bus interface controls data transmission/reception between the VR4130 CPU core and the BCU, which is one of the peripheral units. The bus interface consists of two 32-bit multiplexed address/data buses (one for input, and the other for output), clock signals, interrupt request signals, and various other control signals.

(6) Clock generator

The following clock inputs are oscillated and supplied to internal units.

- 32.768 kHz clock for RTC unit. Oscillating a 32.768 kHz crystal resonator input via an internal oscillator to supply to the RTC unit.
- 18.432 MHz clock for serial interface and the VR4131's reference operating clock. Oscillating an 18.432 MHz crystal resonator input via an internal oscillator, and then multiplying it by a phase-locked loop (PLL) to generate a pipeline clock (PClock). The internal bus clock (TClock and VTClock) is generated from PClock and supplied to peripheral units.

1.5.2 CPU registers

The VR4130 CPU core has the following registers:

- General-purpose registers (GPR): 64 bits × 32

In addition, the processor provides the following special registers:

- PC: Program counter (64 bit)
- HI register: Contains the integer multiply and divide higher doubleword result (64 bits)
- LO register: Contains the integer multiply and divide lower doubleword result (64 bits)

Two of the general-purpose registers are assigned the following functions:

- r0 is fixed to 0, and can be used as the target register for any instruction whose result is to be discarded. r0 can also be used as a source register when a zero value is needed.
- r31 is the link register used by link instructions such as JAL (jump and link) instructions. This register can be used for other instructions. However, be careful that use of the register by a link instruction will not coincide with use of the register for other operations.

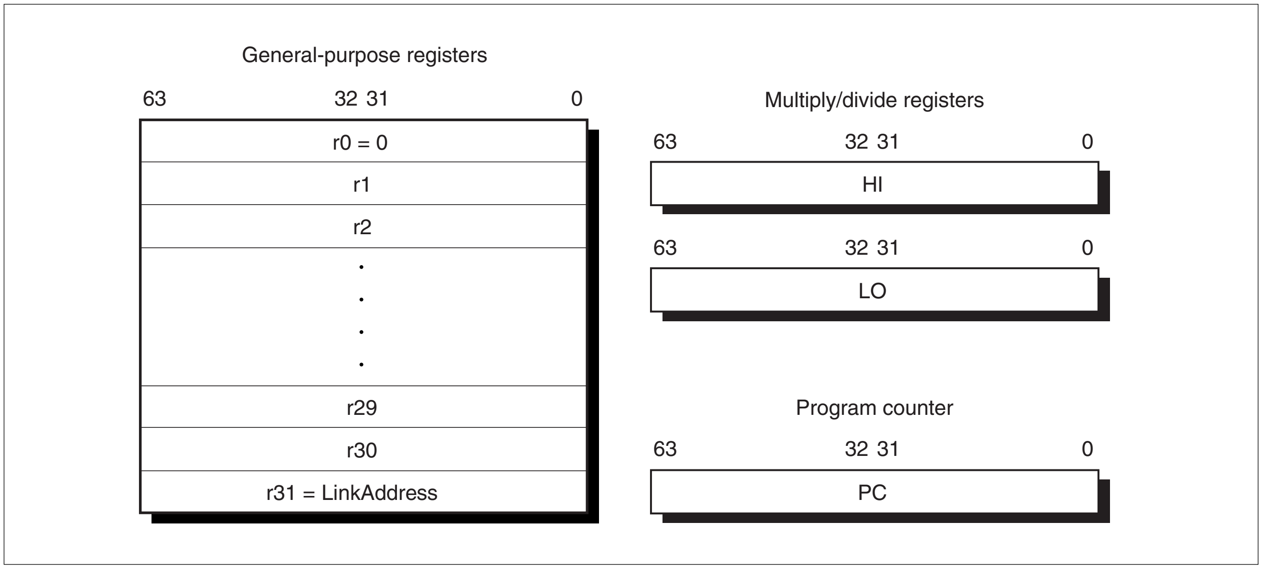
The register group is provided within the CP0 (system control coprocessor), to process exceptions and to manage addresses.

CPU registers can operate as either 32-bit or 64-bit registers, depending on the VR4122 processor operation mode.

The operation of the CPU register differs depending on what instructions are executed: 32-bit instructions or MIPS16 instructions. For details, refer to **VR4100 Series Architecture User's Manual** available separately**.**

Figure 1-3 shows the CPU registers.

Figure 1-3. VR4131 CPU Registers



The VR4131 has no program status word (PSW) register as such; this is covered by the status and cause registers incorporated within the system control coprocessor (CP0).

For details of CP0 registers, refer to **1.5.5 System control coprocessor (CP0).**

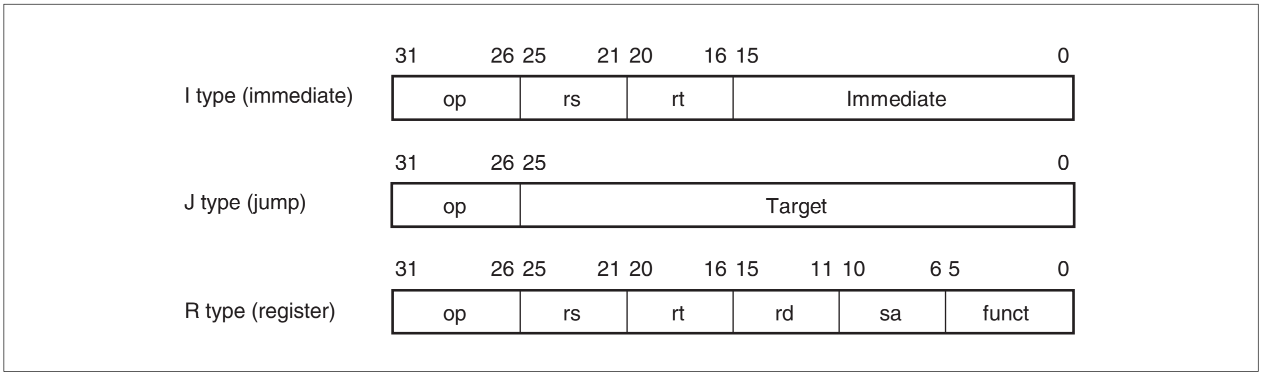
1.5.3 CPU instruction set overview

There are two types of CPU instructions: 32-bit length instructions (MIPS III) and 16-bit length instructions (MIPS16).

(1) MIPS III instructions

All the CPU instructions are 32-bit length when executing MIPS III instructions, and they are classified into three instruction formats as shown in Figure 1-4: immediate (I type), jump (J type), and register (R type). For the field of each instruction format, refer to **VR4100 Series Architecture User's Manual** available separately.

Figure 1-4. CPU Instruction Formats (32-Bit Length Instruction)



The instruction set can be further divided into the following five groupings:

(a) Load and store instructions move data between the memory and the general-purpose registers. They are all immediate (I-type) instructions, since the only addressing mode supported is base register plus 16-bit, signed immediate offset.

(b) Computational instructions perform arithmetic, logical, shift, and multiply and divide operations on values in registers. They include R-type (in which both the operands and the result are stored in registers) and I-type (in which one operand is a 16-bit signed immediate value) formats.

(c) Jump and branch instructions change the control flow of a program. Jumps are made either to an absolute address formed by combining a 26-bit target address with the higher bits of the program counter (J-type format) or register-specified address (R-type format). The format of the branch instructions is I type. Branches have 16-bit offsets relative to the program counter. JAL instructions save their return address in register 31.

(d) System control coprocessor (CP0) instructions perform operations on CP0 registers to control the memory-management and exception-handling facilities of the processor.

(e) Special instructions perform system calls and breakpoint exceptions, or cause a branch to the general exception-handling vector based upon the result of a comparison. These instructions occur in both R-type and I-type formats.

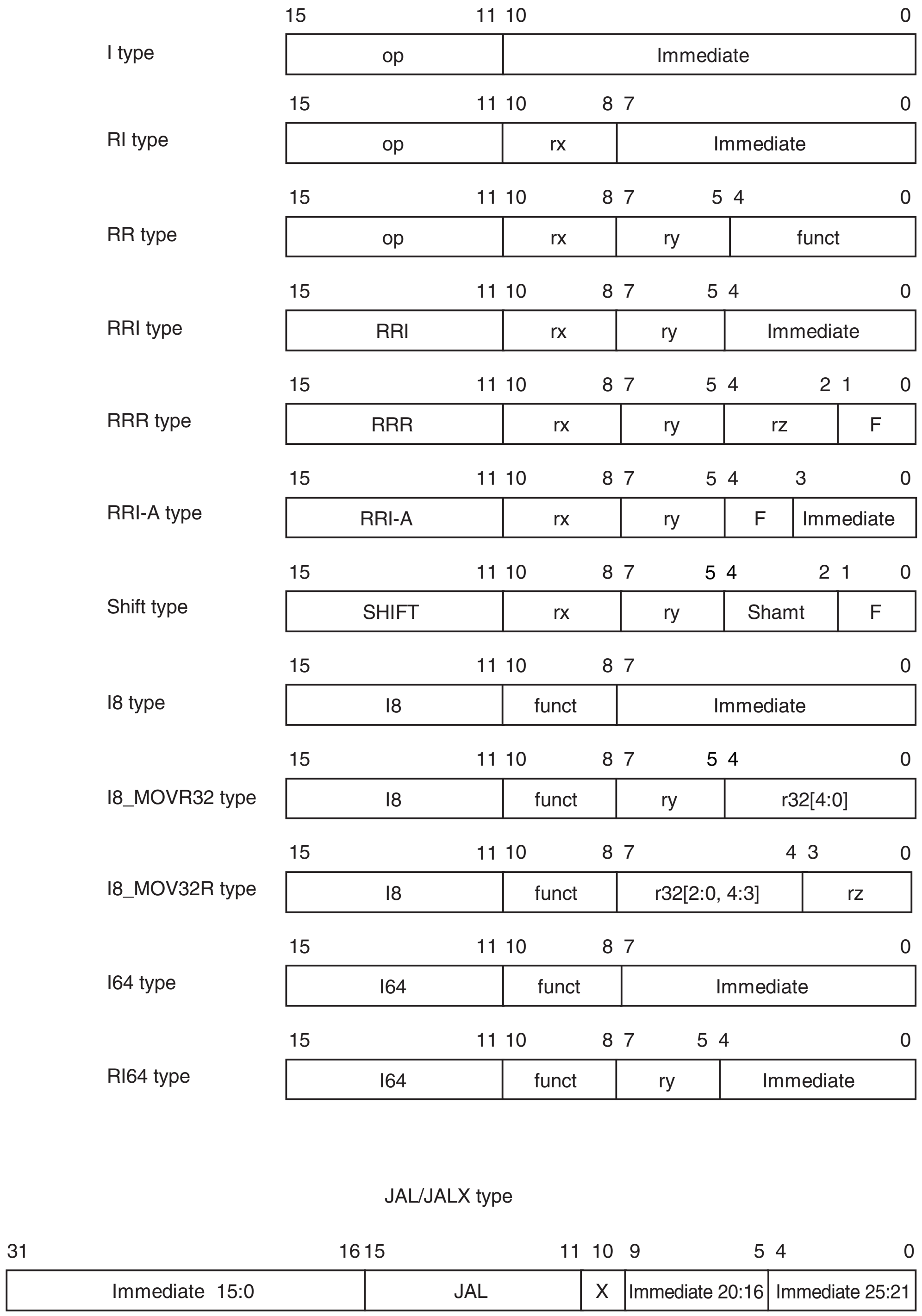
For the operation of each instruction, refer to **VR4100 Series Architecture User’s Manual** available separately**.**

(2) MIPS16 instructions

All the CPU instructions except for JAL and JALX are 16-bit length when executing MIPS16 instructions, and they are classified into thirteen instruction formats as shown in Figure 1-5.

For the field of each instruction format, refer to **VR4100 Series Architecture User’s Manual** available separately**.**

Figure 1-5. CPU Instruction Formats (16-bit length instruction)



The instruction set can be further divided into the following four groupings:

(a) Load and store instructions move data between memory and general-purpose registers. They include RRI, RI, I8, and RI64 types.
(b) Computational instructions perform arithmetic, logical, shift, and multiply and divide operations on values in registers. They include RI-, RRI-A, I8, RI64, I64, RR, RRR, I8_MOVR32, and I8_MOV32R types.
(c) Jump and branch instructions change the control flow of a program. They include JAL/JALX, RR, RI, I8, and I types.
(d) Special instructions are Break and Extend instructions. The Break instruction transfers control to an exception handler. The Extend instruction extends the immediate field of the next instruction. They are RR and I types. When extending the immediate field of the next instruction by using the Extend instruction, one cycle is needed for executing the Extend instruction, and another cycle is needed for executing the next instruction.

For more details of each instruction's operation, refer to **VR4100 Series Architecture User's Manual** available separately.

1.5.4 Data formats and addressing

The VR4131 uses following four data formats:

- Doubleword (64 bits)
- Word (32 bits)
- Halfword (16 bits)
- Byte (8 bits)

In the CPU core, if the data format is any one of halfword, word, or doubleword, the byte ordering can be set as either big endian or little endian.

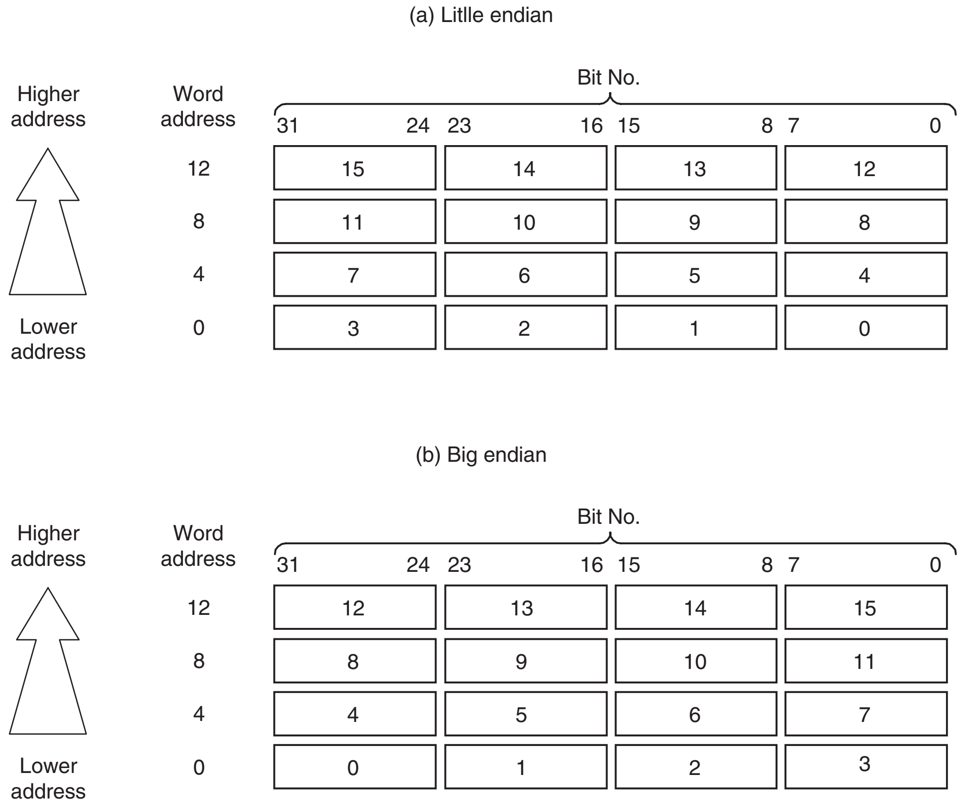
During operation in big-endian mode, however, data transfer to the internal I/O (register) space of the VR4131 or to the PCI bus is performed with data converted to little endian. Therefore, the following restrictions apply for access to these address spaces.

(1) When 3-byte access is executed, data is undefined.
→ Do not perform 3-byte access.
(2) When 8-byte access is executed, the order of higher byte and lower byte is reversed.
(3) Access by the LWR, LWL, LDR, or LDL instruction causes erroneous data to be loaded.
→ Do not use the LWR, LWL, LDR, and LDL instructions.

Endianness refers to the location of byte 0 within a multi-byte data structure. Figures 1-6 and 1-7 show this configuration.

When configured as a little-endian system, byte 0 is always the least-significant (rightmost) byte, which is compatible with iAPX™ and DEC VAX™ conventions.

Figure 1-6. Little-Endian Byte Ordering in Word Data



(a) Litlle endian

Word address	31 – 24	23 – 16	15 – 8	7 – 0
12	15	14	13	12
8	11	10	9	8
4	7	6	5	4
0	3	2	1	0

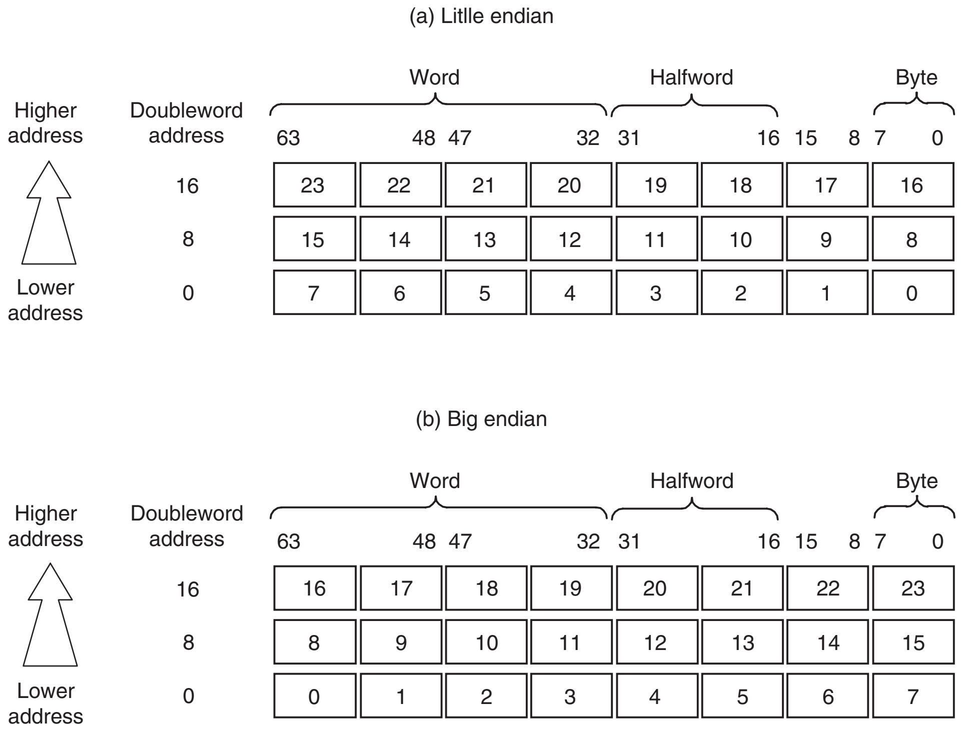
(b) Big endian

Word address	31 – 24	23 – 16	15 – 8	7 – 0
12	12	13	14	15
8	8	9	10	11
4	4	5	6	7
0	0	1	2	3

Remarks 1. The lowest byte is the lowest address.

2. The address of word data is specified by the lowest byte’s address.

Figure 1-7. Little-Endian Byte Ordering in Doubleword Data



(a) Litlle endian

	Word				Halfword			Byte
Doubleword address	63	48	47	32	31	16	15 8	7 0
16	23	22	21	20	19	18	17	16
8	15	14	13	12	11	10	9	8
0	7	6	5	4	3	2	1	0

(b) Big endian

	Word				Halfword			Byte
Doubleword address	63	48	47	32	31	16	15 8	7 0
16	16	17	18	19	20	21	22	23
8	8	9	10	11	12	13	14	15
0	0	1	2	3	4	5	6	7

Remarks 1. The lowest byte is the lowest address.

2. The address of word data is specified by the lowest byte's address.

The VR4130 CPU core uses the following byte boundaries for halfword, word, and doubleword accesses:

- Halfword: An even byte boundary (0, 2, 4...)
- Word: A byte boundary divisible by four (0, 4, 8...)
- Doubleword: A byte boundary divisible by eight (0, 8, 16...)

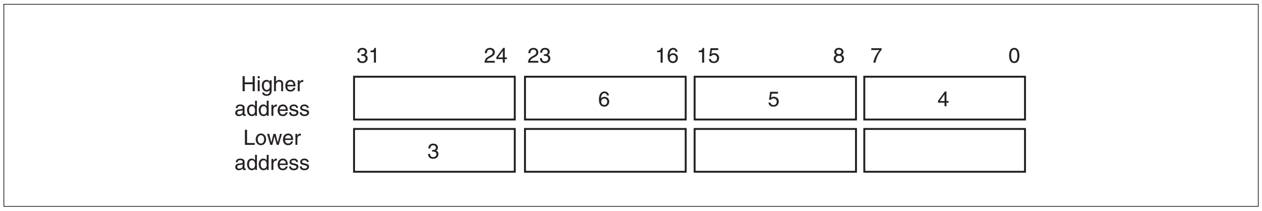
The following special instructions are used to load and store data that are not aligned on 4-byte (word) or 8-byte (doubleword) boundaries:

LWL LWR SWL SWR
LDL LDR SDL SDR

These instructions are used in pairs to provide access to misaligned data. Accessing misaligned data requires one additional instruction cycle (1 Pcycle) over that required for accessing aligned data.

Figure 1-8 shows the access of a misaligned word that has byte address 3.

Figure 1-8. Misaligned Word Accessing (Little-Endian)



1.5.5 System control coprocessor (CP0)

MIPS ISA defines 4 types of coprocessors (CP0 to CP3).

- CP0 translates virtual addresses to physical addresses, switches the operating mode (kernel, supervisor, or user mode), and manages exceptions. It also controls the cache subsystem to analyze a cause and to return from the error state.
- CP1 is reserved for floating-point instructions.
- CP2 is reserved for future definition by MIPS.
- CP3 is no longer defined. CP3 instructions are reserved for future extensions.

Figure 1-9 shows the definitions of the CP0 register, and Table 1-17 shows simple descriptions of each register. For detailed descriptions of the registers related to the virtual memory system, refer to **CHAPTER 3 MEMORY MANAGEMENT SYSTEM**. For detailed descriptions of the registers related to exception handling, refer to **CHAPTER 4 EXCEPTION PROCESSING**.

Figure 1-9. CP0 Registers

Register No.	Register name	Register No.	Register name
0	Index[Note 1]	16	Config[Note 1]
1	Random[Note 1]	17	LLAddr[Note 1]
2	EntryLo0[Note 1]	18	WatchLo[Note 2]
3	EntryLo1[Note 1]	19	WatchHi[Note 2]
4	Context[Note 2]	20	XContext[Note 2]
5	PageMask[Note 1]	21	RFU
6	Wired[Note 1]	22	RFU
7	RFU	23	RFU
8	BadVAddr[Note 1]	24	RFU
9	Count[Note 2]	25	RFU
10	EntryHi[Note 1]	26	Parity error[Note 2]
11	Compare[Note 2]	27	Cache error[Note 2]
12	Status[Note 2]	28	TagLo[Note 1]
13	Cause[Note 2]	29	TagHi[Note 1]
14	EPC[Note 2]	30	ErrorEPC[Note 2]
15	PRId[Note 1]	31	RFU

Notes **1.** For memory management

2. For exception handling

Remark RFU: Reserved for future use

Table 1-18. System Control Coprocessor (CP0) Register Definitions

Register Number	Register Name	Description
0	Index	Programmable pointer to TLB array
1	Random	Pseudo-random pointer to TLB array (read only)
2	EntryLo0	Lower side of TLB entry for even PFN
3	EntryLo1	Lower side of TLB entry for odd PFN
4	Context	Pointer to kernel virtual PTE in 32-bit mode
5	PageMask	Page size specification
6	Wired	Number of wired TLB entries
7	–	Reserved for future use
8	BadVAddr	Virtual address where the most recent error occurred
9	Count	Timer count
10	EntryHi	Higher side of TLB entry (including ASID)
11	Compare	Timer compare
12	Status	Status register
13	Cause	Cause of last exception
14	EPC	Exception program counter
15	PRId	Processor revision identifier
16	Config	Memory mode system specification
17	LLAddr	Reserved for future use
18	WatchLo	Memory reference trap address lower bits
19	WatchHi	Memory reference trap address higher bits
20	XContext	Pointer to kernel virtual PTE in 64-bit mode
21 to 25	–	Reserved for future use
26	Parity error[Note]	Cache parity bits
27	Cache error[Note]	Index and status of cache error
28	TagLo	Cache tag register (low)
29	TagHi	Cache tag register (high)
30	ErrorEPC	Error exception program counter
31	–	Reserved for future use

Note This register is defined to maintain compatibility with the VR4100™. This register is not used in the VR4131 hardware.

1.5.6 Floating-point unit (FPU)

The VR4131 does not support the floating-point unit (FPU). A coprocessor unusable exception will occur if any FPU instructions are executed. If necessary, FPU instructions should be emulated by software in an exception handler.

1.6 CPU Core Memory Management System

The VR4131 has a 32-bit physical addressing range of 4 GB. However, since it is rare for systems to implement a physical memory space as large as that memory space, the CPU provides a logical expansion of memory space by translating addresses composed in the large virtual address space into available physical memory addresses.

The VR4131 supports the following two addressing modes:

- 32-bit mode, in which the virtual address space is divided into 2 GB for user processing and 2 GB for the kernel.
- 64-bit mode, in which the virtual address space is expanded to 1 TB (2^{40} bytes) of user virtual address space.

A detailed description of these address spaces is given in **CHAPTER 6 MEMORY MANAGEMENT SYSTEM**.

1.6.1 Translation lookaside buffer (TLB)

Virtual memory mapping is performed using the translation lookaside buffer (TLB). The TLB converts virtual addresses to physical addresses. It runs by a full-associative method and has 32 entries, each mapping a pair of two consecutive pages. The page size is variable between 1 KB and 256 KB, in powers of 4.

(1) Joint TLB (JTLB)

The JTLB holds both instruction and data addresses.

For fast virtual-to-physical address decoding, the VR4131 uses a large, fully associative TLB (joint TLB) that translates 64 virtual pages to their corresponding physical addresses. The TLB is organized as 32 pairs of even-odd entries, and maps a virtual address and address space identifier (ASID) into the 4 GB physical address space.

The page size can be configured, on a per-entry basis, to map a page size of 1 KB to 256 KB. A CP0 register stores the size of the page to be mapped, and that size is entered into the TLB when a new entry is written. Thus, operating systems can provide special purpose maps; for example, a typical frame buffer can be memory-mapped using only one TLB entry.

Translating a virtual address to a physical address begins by comparing the virtual address from the processor with the physical addresses in the TLB; there is a match when the virtual page number (VPN) of the address is the same as the VPN field of the entry, and either the global (G) bit of the TLB entry is set, or the ASID field of the virtual address is the same as the ASID field of the TLB entry.

This match is referred to as a TLB hit. If there is no match, a TLB miss exception is taken by the processor and software is allowed to refill the TLB from a page table of virtual/physical addresses in memory.

1.6.2 Operating modes

The VR4122 has the following three operating modes:

- User mode
- Supervisor mode
- Kernel mode

The manner in which memory addresses are translated or mapped depends on these operating modes. Refer to **CHAPTER 3 MEMORY MANAGEMENT SYSTEM** for details.

1.6.3 Cache

The VR4131 chip incorporates instruction and data caches, which are independent of each other. This configuration enables high-performance pipeline operations. Both caches have a 64-bit data bus, enabling a one-clock access. These buses can be accessed in parallel. The instruction cache of the VR4131 has a storage capacity of 16 KB, while the data cache has a capacity of 16 KB.

For details, refer to **VR4100 Series Architecture User's Manual** available separately.

1.7 Instruction Pipeline

The VR4131 uses a 6-step instruction pipeline in the MIPS III instruction mode. Therefore, at least 6 cycles are required to execute each instruction. Because a dual pipeline is employed, when the pipeline processing is operating smoothly, it indicates that 12 instructions are being simultaneously executed.

The VR4131 uses a 7-step instruction pipeline in the MIPS 16 instruction mode. Therefore, at least 7 cycles are required to execute each instruction. Because a dual pipeline is employed, when the pipeline processing is operating smoothly, it indicates that 14 instructions are being simultaneously executed.

For details, refer to **VR4100 Series Architecture User's Manuals**, available separately.

1.8 Clock Interface

The VR4131 has the following 10 clocks.

- **CLKX1, CLKX2 (input)**
  These are 18.432 MHz resonator inputs, and are used to generate operation clocks (PClock) for the CPU core. They are also used for the CMU, DSIU, and SIU.
- **RTCX1, RTCX2 (input)**
  These are 32.768 kHz resonator inputs, which are used by the PMU, RTC, DSU, LED, touch panel interface, and keyboard interface. They are also used as the main operation clock for some of the GPIO pins. Only this clock continues to operate when the system is in Hibernate mode.
- **FIRCLK (input)**
  This is a 48 MHz clock input, and used for FIR.
- **PClock (internal)**
  This clock is used to control the pipeline used in the VR4130 CPU core, and for units relating to the pipeline. This clock is generated from the clock input of CLKX1 and CLKX2 pins. Its frequency is determined by CLKSEL(2:0) pins.

- **PCICLK (output)**
  PCICLK supplies the controller on the PCI bus with the clock. The initial value is 1/2 the VTClock frequency. The PCICLK output can be changed by setting PCICLKSELREG.
- **VTClock (internal)**
  This is the reference clock of the ROM, I/O, and SDRAM access. The initial frequency is about 33 MHz. The division ratio to PClock can be changed to any of 1/1, 1/2, 1/3, 1/4, 1/5, or 1/6 after setting the PMUTCLKDIVREG register and restarting the processor. However, depending on their relationship with the maximum frequency of PClock, it may not be possible to set certain division ratios.
- **SCLK (output)**
  SCLK supplies the CLK pins of SDRAM with the clock. It has the same frequency as VTClock and operates only when accessing SDRAM.
- **MasterOut (internal)**
  This is the clock output from the VR4130 CPU Core and is used for interrupt control. The contents of the CP0 counter register are incremented in synchronization with this clock.
  The frequency of MasterOut is 1/4 of TClock.
- **TClock**
  This is the reference clock for on-chip peripheral operations. The initial frequency is about 16 MHz. The division ratio to VTClock can be changed to either 1/2 or 1/4 after setting the PMUTCLKDIVREG register and restarting the processor.
- **Intrclk**
  This is the interrupt detection/control clock of on-chip peripherals. The initial frequency is about 9.216 MHz. The division ratio to CLKX1 (18.432 MHz) can be changed to any of 1/2, 1/4, or 1/8 after setting the PMUINTRDIVREG register and restarting the processor.

The following tables show the CLKSEL(2:0) pin setting and frequency of each clock.

Table 1-19. CLKSEL Pin Setting and Frequency of Each Clock

CLKSEL(2:0)	PClock	VTClock[Note 1]		PCICLK[Note 2] (At 1/2 of VTClock)
		MIN.	MAX.	
111	RFU	RFU	RFU	RFU
110	199.1 MHz	33.2 MHz	99.5 MHz	49.75 MHz
101	181.0 MHz	30.2 MHz	90.5 MHz	45.25 MHz
100	165.9 MHz	27.6 MHz	82.9 MHz	41.45 MHz
011	153.1 MHz	30.6 MHz	76.6 MHz	38.3 MHz
010	132.7 MHz	26.5 MHz	66.4 MHz	33.2 MHz
001	99.5 MHz	33.2 MHz	49.8 MHz	29.9 MHz
000	RFU	RFU	RFU	RFU

Notes 1. This pin is set to the MIN. value during RTC reset. After reset, its frequency can be changed.

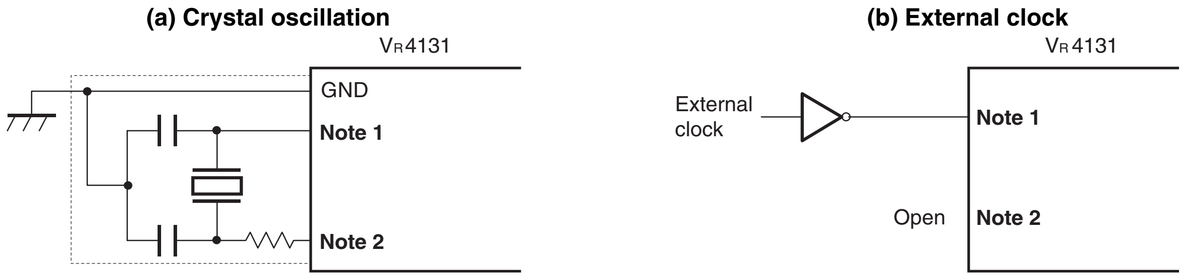
2. During RTC reset, this pin is set to output at 1/2 the VTClock frequency.

Table 1-20. CLKSEL Pin Setting and Frequency of VTClock and SCLK

CLKSEL(2:0)	PClock	VTClock, SCLK				
		Div2	Div3	Div4	Div5	Div6
111	RFU	RFU	RFU	RFU	RFU	RFU
110	199.1 MHz	99.5 MHz	66.4 MHz	49.8 MHz	39.8 MHz	33.2 MHz
101	181.0 MHz	90.5 MHz	60.3 MHz	45.2 MHz	36.2 MHz	30.2 MHz
100	165.9 MHz	82.9 MHz	55.3 MHz	41.5 MHz	33.2 MHz	27.6 MHz
011	153.1 MHz	76.6 MHz	51.0 MHz	38.3 MHz	30.6 MHz	RFU
010	132.7 MHz	66.4 MHz	44.2 MHz	33.2 Mhz	26.5 MHz	RFU
001	99.5 MHz	49.8 MHz	33.2 MHz	RFU	RFU	RFU
000	RFU	RFU	RFU	RFU	RFU	RFU

Figure 1-10 shows the external circuits of the clock oscillator.

Figure 1-10. External Circuits of Clock Oscillator



(a) Crystal oscillation

(b) External clock

Notes 1. CLKX1, RTCX1

2. CLKX2, RTCX2

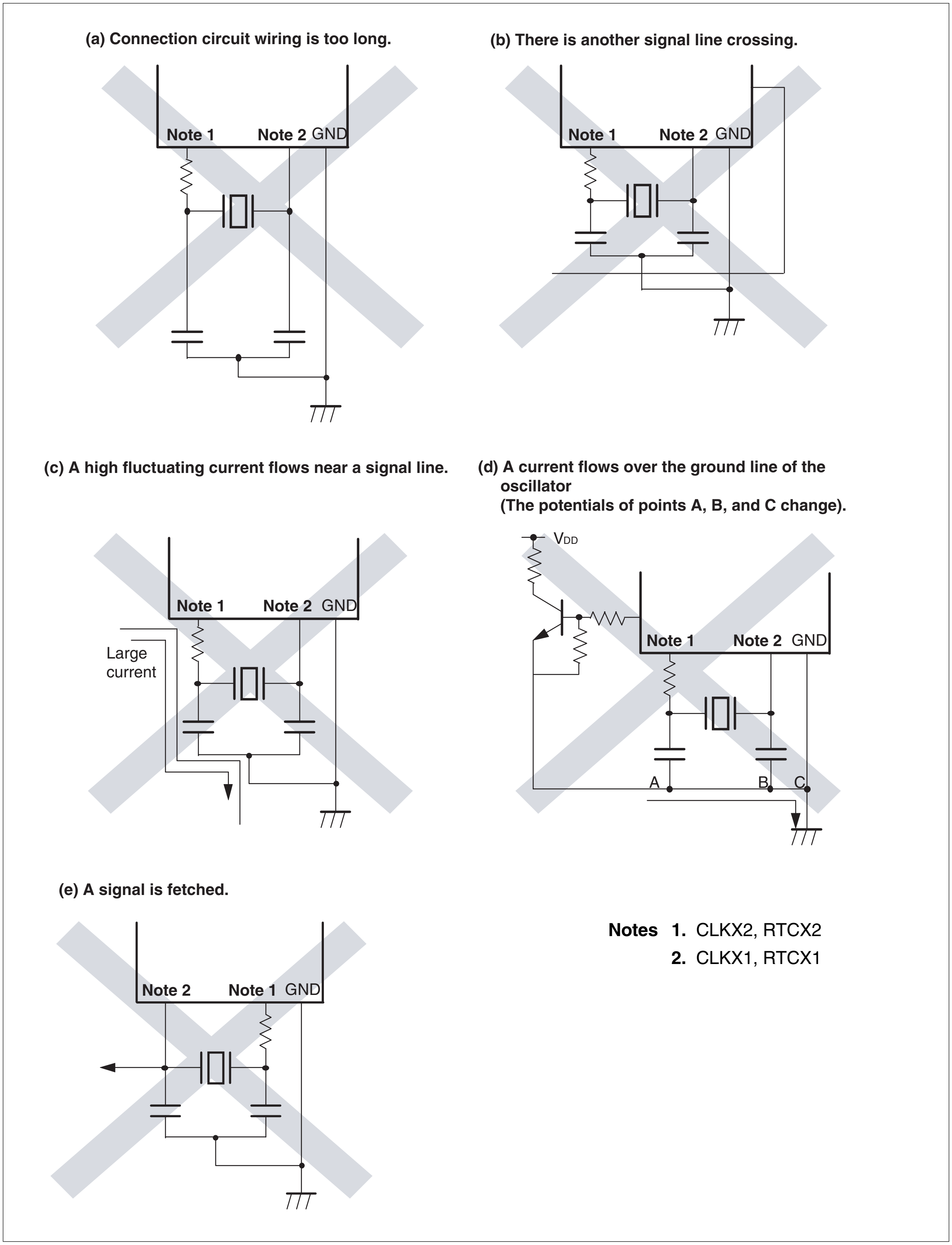
Cautions 1. When using the clock oscillator, wire as follows in the area enclosed by the broken line in the above figures to avoid an adverse effect from wiring capacitance.

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as GND. Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

2. Ensure that no load such as wiring capacity is applied to the CLKX2 or RTCX2 pin when inputting an external clock.

Figure 1-11 shows examples of the incorrect connection circuit of the resonator.

Figure 1-11. Incorrect Connection Circuits of Resonator



(a) Connection circuit wiring is too long.

(b) There is another signal line crossing.

(c) A high fluctuating current flows near a signal line.

(d) A current flows over the ground line of the oscillator (The potentials of points A, B, and C change).

(e) A signal is fetched.

Notes 1. CLKX2, RTCX2

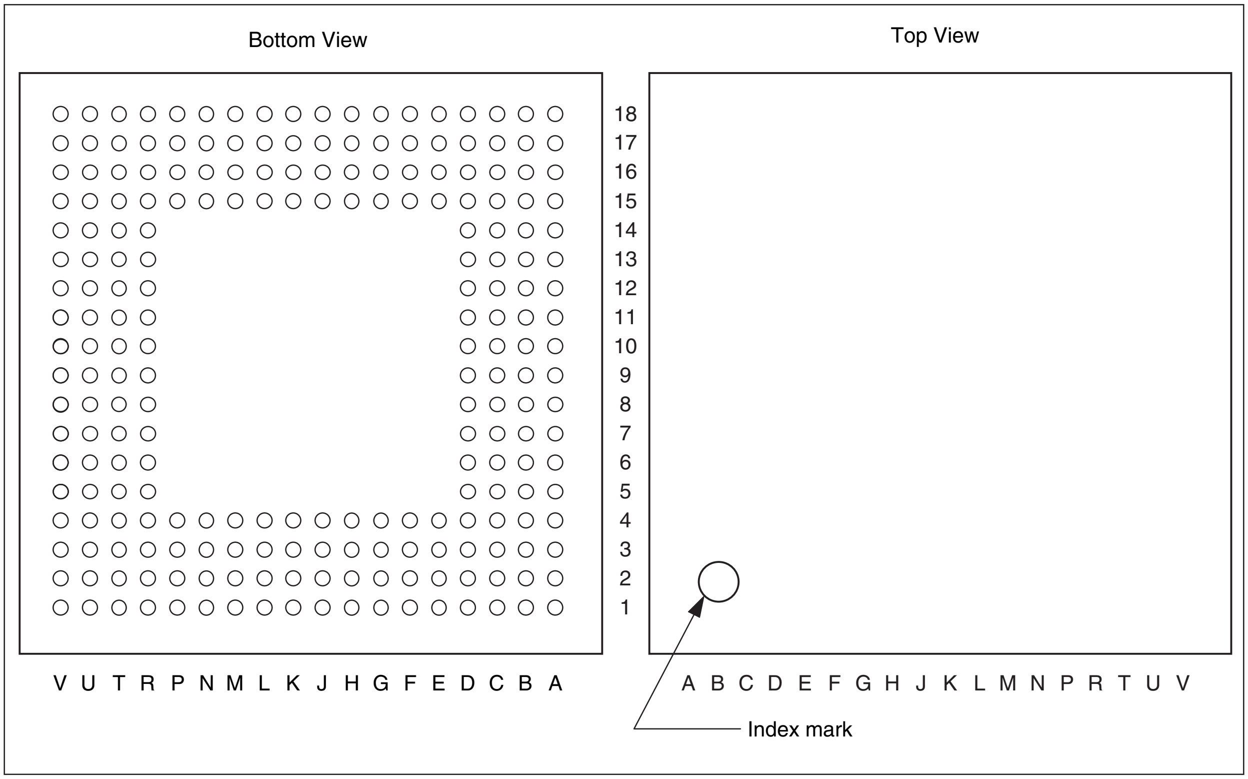
2. CLKX1, RTCX1

CHAPTER 2 PIN FUNCTIONS

Caution CHAPTER2 describes revision 1.2 or earlier.
When using revision 2.0 or later, also read APPENDIX A DIFFERENCES BETWEEN REVISION 1.2 AND REVISION 2.0 OR LATER.

2.1 Pin Configuration

- 224-pin plastic FBGA (16 × 16 mm)


Bottom View
Top View
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
V U T R P N M L K J H G F E D C B A
A B C D E F G H J K L M N P R T U V
Index mark

Pin No.	Power Supply	Pin Name	Pin No.	Power Supply	Pin Name	Pin No.	Power Supply	Pin Name
A1	3.3 V	CLKOUT	C14	3.3 V	AD3	H3	3.3 V	ADD13
A2	1.5 V	V_{DD}PD	C15	3.3 V	CBE2	H4	3.3 V	DATA5
A3	1.5 V	V_{DD}P	C16	3.3 V	DEVSEL#	H15	3.3 V	POWER
A4	3.3 V	CLKX1	C17	3.3 V	PAR	H16	3.3 V	GND3
A5	3.3 V	CLKX2	C18	3.3 V	FRAME#	H17	3.3 V	N.C.
A6	3.3 V	PCLK	D1	3.3 V	ADD17	H18	3.3 V	FIRCLK
A7	3.3 V	AD28	D2	3.3 V	ADD21	J1	3.3 V	DATA8
A8	3.3 V	AD23	D3	3.3 V	ADD22	J2	3.3 V	DATA7
A9	3.3 V	AD19	D4	1.5 V	V_{DD}1	J3	3.3 V	ADD11
A10	3.3 V	AD16	D5	3.3 V	GND3	J4	3.3 V	ADD12
A11	3.3 V	AD13	D6	3.3 V	CGND	J15	3.3 V	POWERON
A12	3.3 V	AD12	D7	3.3 V	AD29	J16	3.3 V	MPOWER
A13	3.3 V	AD10	D8	3.3 V	AD24	J17	1.5 V	GND1
A14	3.3 V	AD6	D9	3.3 V	AD20	J18	3.3 V	IRDOUT#
A15	3.3 V	AD2	D10	3.3 V	AD15	K1	3.3 V	DATA10
A16	3.3 V	RST#	D11	1.5 V	GND1	K2	3.3 V	DATA9
A17	3.3 V	CBE1	D12	3.3 V	AD8	K3	3.3 V	ADD10
A18	3.3 V	IRDY#	D13	3.3 V	AD4	K4	3.3 V	DATA11
B1	3.3 V	ADD23	D14	3.3 V	AD0	K15	3.3 V	GND3
B2	3.3 V	V_{DD}3	D15	3.3 V	GND3	K16	1.5 V	V_{DD}1
B3	1.5 V	GNDP	D16	3.3 V	GND3	K17	3.3 V	MMUEN
B4	3.3 V	CV_{DD}	D17	3.3 V	PERR#	K18	3.3 V	V_{DD}3
B5	3.3 V	RTCX1	D18	3.3 V	STOP#	L1	3.3 V	DATA13
B6	3.3 V	AD30	E1	3.3 V	ADD15	L2	3.3 V	DATA12
B7	3.3 V	AD25	E2	3.3 V	ADD18	L3	3.3 V	GND3
B8	3.3 V	AD22	E3	3.3 V	ADD16	L4	3.3 V	ADD9
B9	3.3 V	AD17	E4	3.3 V	ADD19	L15	3.3 V	RTCRST#
B10	3.3 V	AD14	E15	3.3 V	GND3	L16	3.3 V	RSTSW#
B11	3.3 V	V_{DD}3	E16	3.3 V	GND3	L17	3.3 V	N.C.
B12	3.3 V	AD9	E17	3.3 V	REQ1#	L18	3.3 V	IRDIN
B13	3.3 V	AD5	E18	3.3 V	CLKRUN	M1	3.3 V	DATA15
B14	3.3 V	AD1	F1	3.3 V	GND3	M2	3.3 V	DATA14
B15	3.3 V	CBE3	F2	3.3 V	DATA1	M3	3.3 V	DATA17/GPIO17
B16	3.3 V	CBE0	F3	3.3 V	DATA2	M4	3.3 V	ADD8
B17	3.3 V	V_{DD}3	F4	3.3 V	DATA0	M15	3.3 V	DDIN/GPIO34
B18	3.3 V	TRDY#	F15	3.3 V	REQ0#	M16	3.3 V	LEDOUT#
C1	3.3 V	ADD20	F16	3.3 V	REQ2#	M17	3.3 V	V_{DD}3
C2	3.3 V	ADD24	F17	3.3 V	GNT0#	M18	3.3 V	FIRDIN#/SEL
C3	1.5 V	GNDPD	F18	3.3 V	GNT2#	N1	3.3 V	ADD7
C4	3.3 V	AD27	G1	1.5 V	GND1	N2	3.3 V	DATA16/GPIO16
C5	3.3 V	GND3	G2	3.3 V	ADD14	N3	3.3 V	ADD6
C6	3.3 V	RTCX2	G3	3.3 V	V_{DD}3	N4	3.3 V	DATA18/GPIO18
C7	3.3 V	AD31	G4	3.3 V	DATA3	N15	3.3 V	DRTS#/MIPS16EN/GPIO33
C8	3.3 V	AD26	G15	3.3 V	SERR#	N16	3.3 V	DDOUT/DBUS32/GPIO32
C9	3.3 V	AD21	G16	3.3 V	GNT1#	N17	3.3 V	I.C. (Open)
C10	3.3 V	AD18	G17	3.3 V	BIGENDIAN	N18	3.3 V	HLDRQ#
C11	1.5 V	V_{DD}1	G18	3.3 V	LOCK#	P1	3.3 V	DATA20/GPIO20
C12	3.3 V	AD11	H1	3.3 V	DATA6	P2	3.3 V	DATA19/GPIO19
C13	3.3 V	AD7	H2	3.3 V	DATA4	P3	3.3 V	GND3

Remark # indicates active low.

Pin No.	Power Supply	Pin Name	Pin No.	Power Supply	Pin Name	Pin No.	Power Supply	Pin Name
P4	3.3 V	GND3	T4	3.3 V	DATA27/GPIO27	U12	1.5 V	GND1
P15	3.3 V	RTS#/CLKSEL1	T5	3.3 V	DATA31/GPIO31	U13	3.3 V	SIN
P16	3.3 V	DCTS#/GPIO35	T6	3.3 V	CAS	U14	3.3 V	GPIO3
P17	3.3 V	TxD/CLKSEL2	T7	3.3 V	SWR#	U15	3.3 V	GPIO7
P18	3.3 V	HLDAK#	T8	3.3 V	CKE0	U16	3.3 V	GPIO8
R1	3.3 V	DATA21/GPIO21	T9	3.3 V	ROMCS0#	U17	3.3 V	$V_{DD}3$
R2	3.3 V	ADD5	T10	3.3 V	IOCS0#	U18	3.3 V	DCD#/GPIO15
R3	3.3 V	$V_{DD}3$	T11	1.5 V	$V_{DD}1$	V1	3.3 V	DATA25/GPIO25
R4	3.3 V	DQM1	T12	3.3 V	GPIO0	V2	3.3 V	DATA26/GPIO26
R5	3.3 V	DATA29/GPIO29	T13	3.3 V	GPIO4	V3	3.3 V	DATA28/GPIO28
R6	3.3 V	WR#	T14	3.3 V	GND3	V4	3.3 V	DATA30/GPIO30
R7	3.3 V	RAS	T15	3.3 V	IORDY	V5	3.3 V	SCLK
R8	3.3 V	DQM3	T16	3.3 V	GPIO10	V6	3.3 V	$V_{DD}3$
R9	3.3 V	CS1#	T17	3.3 V	GPIO13	V7	3.3 V	DQM0
R10	3.3 V	SPOWER	T18	3.3 V	DTR#/CLKSEL0	V8	3.3 V	CKE1
R11	3.3 V	$V_{DD}3$	U1	3.3 V	DATA24/GPIO24	V9	3.3 V	CS2#/ROMCS2#
R12	3.3 V	SOUT	U2	3.3 V	$V_{DD}3$	V10	3.3 V	ROMCS1#
R13	3.3 V	GPIO2	U3	3.3 V	ADD3	V11	3.3 V	GND3
R14	3.3 V	GND3	U4	3.3 V	ADD2	V12	3.3 V	SECLK
R15	3.3 V	BATTINH/BATTINT#	U5	3.3 V	ADD1	V13	3.3 V	GPIO1
R16	3.3 V	DSR#	U6	3.3 V	RD#	V14	3.3 V	GPIO5
R17	3.3 V	CTS#	U7	3.3 V	GND3	V15	3.3 V	GPIO6/SYSDIR
R18	3.3 V	RxD	U8	3.3 V	DQM2	V16	3.3 V	GPIO9
T1	3.3 V	DATA23/GPIO23	U9	3.3 V	CS0#	V17	3.3 V	GPIO11
T2	3.3 V	DATA22/GPIO22	U10	3.3 V	CS3#/ROMCS3#	V18	3.3 V	GPIO12
T3	3.3 V	ADD4	U11	3.3 V	IOCS1#			

Remark # indicates active low.

Pin Identification

AD(0:31):	Address/data bus
ADD(1:24):	Address bus
BATTINH:	Battery inhibit
BATTINT#:	Battery interrupt request
BIGENDIAN:	Big endian
CAS:	Column address strobe
CBE(0:3):	Command/byte enable
CGND:	GND for oscillator
CKE(0:1):	Clock enable
CLKSEL(0:2):	Clock select
CLKOUT:	Clock output
CLKRUN:	Clock run
CLKX1:	Clock X1
CLKX2:	Clock X2
CS(0:3)#:	Chip select
CTS#:	Clear to send
CVDD:	VDD for oscillator
DATA(0:31):	Data bus
DBUS32:	Data bus 32
DCD#:	Data carrier detect
DCTS#:	Debug serial clear to send
DDIN:	Debug serial data input
DDOUT:	Debug serial data output
DEVSEL#:	Device select
DQM(0:3):	Data input/output
DRTS#:	Debug serial request to send
DSR#:	Data set ready
DTR#:	Data terminal ready
FIRCLK:	FIR clock
FIRDIN#:	FIR data input
FRAME#:	Cycle frame
GND1, GND3:	Ground
GNDP, GNDPD:	GND for PLL
GNT(0:2)#:	Grant
GPIO(0:13, 15:35):	General purpose I/O
HLDAK#:	Hold acknowledge
HLDRQ#:	Hold request
I.C.:	Internally connected
IOCS(0:1)#:	I/O chip select
IORDY:	I/O ready
IRDY#:	Initiator ready
IRDIN:	IrDA data input
IRDOUT#:	IrDA data output
LEDOUT#:	LED output
LOCK#:	Lock
MIPS16EN:	MIPS16 enable
MMUEN:	MMU Enable
MPOWER:	Main power
N.C.:	No connection
PAR:	Parity
PCLK:	PCI clock
PERR#:	Parity error
POWER:	Power switch
POWERON:	Power on state
RAS:	Row address strobe
RD#:	Read
REQ(0:2)#:	Request
ROMCS(0:3)#:	ROM chip select
RST#:	Reset
RSTSW#:	Reset switch
RTCRST#:	Real-time clock reset
RTCX1:	Real-time clock X1
RTCX2:	Real-time clock X2
RTS#:	Request to send
RxD:	Receive data
SCLK:	SDRAM clock
SECLK:	Clocked serial clock
SEL:	IrDA module select
SERR#:	System error
SIN:	Clocked serial input
SOUT:	Clocked serial output
SPOWER:	SDRAM power control
STOP#:	Target assert stop
SWR#:	SDRAM write
SYSDIR:	System bus buffer direction
TRDY#:	Target ready
TxD:	Transmit data
VDD1, VDD3:	Power supply voltage
VDDP, VDDPD:	VDD for PLL
WR#:	Write

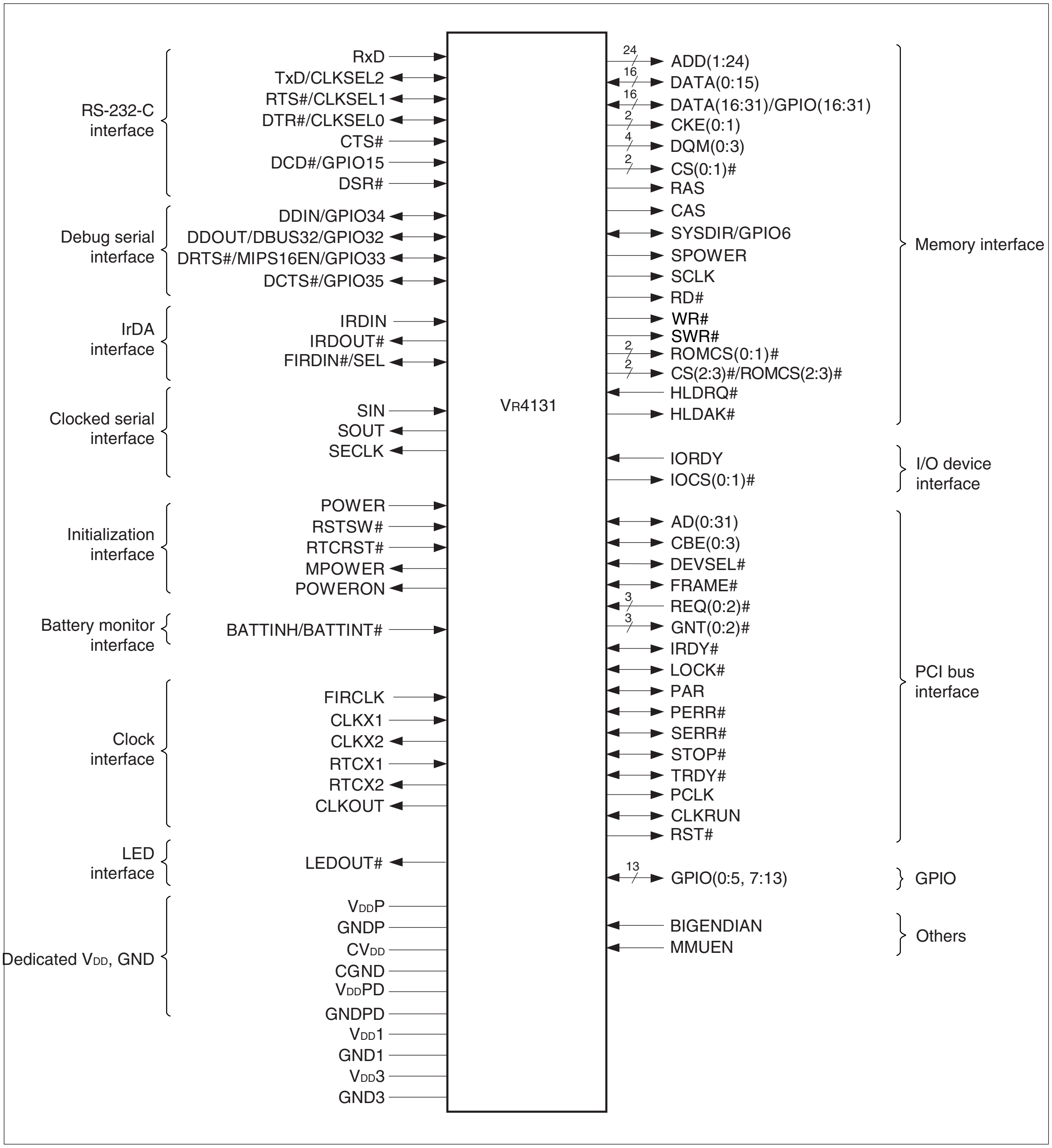
Remark # indicates active low.

2.2 Pin Function Description

The functional classifications of the VR4131 pins are listed below.

Remark # indicates active low.

Figure 2-1. VR4131 Signal Classification



2.2.1 Memory interface signals

The memory interface signals are used to connect the VR4131 with SDRAM and ROM in the system.

Table 2-1. Memory Interface Signals (1/2)

Signal	I/O	Function
SCLK	Output	Operation clock for SDRAM
ADD(1:24)	Output	Higher 24 bits of the 25-bit address bus These signals are used to specify addresses for the VR4131, SDRAM, ROM, and I/O space.
DATA(0:15)	I/O	16-bit data bus These signals are used to transmit data between the VR4131 and the SDRAM, ROM, or I/O space.
DATA(16:31)/ GPIO(16:31)	I/O	Functions may differ depending on the DBUS32 pin setting. • **When DBUS32 = 1** These signals function as the higher 16 bits of the 32-bit data bus. They are used to transmit data between the VR4131 and the DRAM or ROM. • **When DBUS32 = 0** These signals function as general I/O ports.
CKE(0:1)	Output	Clock enable signals for SDRAM
		CKE signal supports the following banks. SDRAM bank / CKE / Data bus 32 / Data bus 16 Bank 3 / CKE(1) / CS(3)#/ROMCS(3)# / DQM3 Bank 2 / CKE(0) / CS(2)#/ROMCS(2)# / DQM2 Bank 1 / CKE(1) / CS(1)# / CS1# Bank 0 / CKE(0) / CS(0)# / CS0#
DQM3	Output	The function differs depending on the setting of the DBUS32 pin and the connected device. • **When DBUS32 = 1 and** SDRAM is accessed: This is the byte enable signal for DATA(24:31) of the 32-bit data bus. A 32-bit external I/O device is accessed: This is the byte enable signal for DATA(24:31) of the 32-bit data bus. • **When DBUS32 = 0 and** SDRAM is accessed: This is the CS signal for SDRAM. This signal becomes active when a command is issued for the SDRAM connected to the highest address.
DQM2	Output	The function differs depending on the setting of the DBUS32 pin and the connected device. • **When DBUS32 = 1 and** SDRAM is accessed: This is the byte enable signal for DATA(16:23) of the 32-bit data bus. A 32-bit external I/O device is accessed: This is the byte enable signal for DATA(16:23) of the 32-bit data bus. • **When DBUS32 = 0 and** SDRAM is accessed: This is the CS signal for SDRAM. This signal becomes active when a command is issued for the SDRAM connected to the second highest address.
DQM1	Output	The function differs depending on the connected device. When SDRAM is accessed: This is the byte enable signal for DATA(8:15). When a 32-bit external I/O device is accessed: This is the byte enable signal for DATA(8:15). When a 16-bit external I/O device is accessed: This is the ADD(0) signal of the address bus. When a 16-bit external I/O device is accessed (little endian): This is the ADD0 signal of the address bus. When a 16-bit external I/O device is accessed (big endian): This is the high-byte enable signal of the I/O bus.

Table 2-1. Memory Interface Signals (2/2)

Signal	I/O	Function
DQM0	Output	The function differs depending on the connected device. When SDRAM is accessed: This is the byte enable signal for DATA(0:7). When a 32-bit external I/O device is accessed: This is the byte enable signal for DATA(0:7). When a 16-bit external I/O device is accessed: This is the high-byte enable signal of the I/O bus. When a 16-bit external I/O device is accessed (little endian): This is the high-byte enable signal of the I/O bus. When a 16-bit external I/O device is accessed (big endian): This is the ADD0 signal of the address bus.
CS(0:1)#	Output	Chip select signal for SDRAM
RAS	Output	RAS signal for SDRAM
CAS	Output	CAS signal for SDRAM
SYSDIR/GPIO6	I/O	Direction signal for SDRAM If not used as the SYSDIR signal, this signal can be used as a GPIO pin.
SPOWER	Output	Power control signal for SDRAM
RD#	Output	This signal becomes active when operating the read access of data from the I/O space and ROM.
WR#	Output	This signal becomes active when writing data to the I/O space.
SWR#	Output	This signal becomes active when writing data to SDRAM.
ROMCS(0:1)#	Output	Chip select signals for ROM
CS(2:3)#/ ROMCS(2:3)#	Output	Chip select signals for an extended SDRAM or extended ROM • **When using extended SDRAM** These signals function as CS(2:3)#. • **When using extended ROM** These signals function as ROMCS(2:3)#.
HLDRQ#	Input	Bus mastership request signal for system bus and DRAM sent from the external bus master
HLDAK#	Output	Bus mastership enable signal for system bus and DRAM sent to the external bus master

2.2.2 I/O device interface signals

The I/O device interface signals are used to control the 16-bit device in I/O space using the memory interface of the VR4131.

Table 2-2. I/O Device Interface Signals

Signal	I/O	Function
IOCS(0:1)#	Output	Device chip select signals These signals become active when the VR4131 accesses the I/O device using the ADD bus or DATA bus.
IORDY	Input	Device ready signal Make this signal active in condition that the I/O device allows to be accessed from the VR4131.

2.2.3 Clock interface signals

These signals are used to supply clocks.

Table 2-3. Clock Interface Signals

Signal	I/O	Function
RTCX1	Input	This is the 32.768 kHz oscillator's input pin. It is connected to one side of a crystal resonator.
RTCX2	Output	This is the 32.768 kHz oscillator's output pin. It is connected to one side of a crystal resonator.
CLKX1	Input	This is the 18.432 MHz oscillator's input pin. It is connected to one side of a crystal resonator.
CLKX2	Output	This is the 18.432 MHz oscillator's output pin. It is connected to one side of a crystal resonator.
FIRCLK	Input	This is the 48 MHz clock input pin. This signal inputs a clock when FIR is used.
CLKOUT	Output	This is the clock output to supply an external device. A 9.216 MHz clock is output in the non-Hibernate mode. The clock output stops at Low during Hibernate.

The operating frequency of the CPU core can be set by the TxD/CLKSEL2, RTS0#/CLKSEL1, and DTR0#/CLKSEL0 signals.

For details of these signals, refer to **2.2.6 RS-232C interface signals**.

2.2.4 Battery monitor interface signals

These signals indicate when an external circuit is able to provide enough power for system operations.

Table 2-4. Battery Monitor Interface Signals

Signal	I/O	Function
BATTINH/ BATTINT#	Input	Functions may differ depending on how the MPOWER pin is set. • **When MPOWER = 0** BATTINH function This signal enables/disables activation at power-on. 1: Activation enabled 0: Activation disabled • **When MPOWER = 1** BATTINT# function This is an interrupt signal that is output when the remaining power is low during normal operations. The external agent checks the remaining battery power. Activate the signal at this pin if voltage sufficient for operations cannot be supplied.

2.2.5 Initialization interface signals

These signals are used when an external circuit initializes the processor operation parameters.

Table 2-5. Initialization Interface Signals

Signal	I/O	Function
MPOWER	Output	This signal indicates the VR4131 is operating. This signal is inactive in Hibernate mode.
POWERON	Output	This signal indicates the VR4131 is ready to operate. It becomes active when a power-on factor is detected and becomes inactive when the BATTINH/BATTINT# signal check operation is completed.
POWER	Input	This is the VR4131 activation signal.
RSTSW#	Input	This is the VR4131 reset signal.
RTCRST#	Input	This signal resets the RTC. When power is first supplied to a device, the external circuit must assert the pin at this pin for about 2 s.

2.2.6 RS-232C interface signals

The RS-232C interface signals control the transmission of data between the RS-232C controller and the VR4131. The RS-232C function and the IrDA function are not used together. These signals have alternate functions as the initialization setting signals set after RTC reset.

Table 2-6. RS-232C Interface Signals

Signal	I/O	Function
RxD	Input	This is a receive data signal. It is used when the RS-232C controller sends serial data to the VR4131.
CTS#	Input	This is a transmit enable signal. Assert this signal when the RS-232C controller is ready to receive transmission of serial data.
DCD#/GPIO15	Input	This is a carrier detection signal. Assert this signal when valid serial data is being received. It is also used as a power-on factor for the VR4131. When this pin is not used for DCD# signal, this pin can be used as an interrupt detection input for the GIU.
DSR#	Input	This is the data set ready signal. Assert this signal when the RS-232C controller is ready to receive/transmit serial data between the controller and the VR4131.
TxD/ CLKSEL2, RTS#/ CLKSEL1, DTR#/ CLKSEL0	I/O	This function differs depending on the operating status. • **During normal operation (output)** Signals used for serial communication TxD signal : This is a transmit data signal. It is used when the VR4131 sends serial data to the RS-232C controller. RTS# signal : This is a transmit request signal. This signal is asserted when the VR4131 is ready to receive serial data from the RS-232C controller. DTR# signal : This is a terminal equipment ready signal. This signal is asserted when the VR4131 is ready to transmit or receive serial data. • **During RTC reset (input)** Signals (CLKSEL(2:0)) used to set the CPU core operation frequency, BUSCLK frequency, and internal bus clock frequency. These signals are sampled when the RTCRST# signal changes from low level to high level. For the relationships between the CLKSEL pin setting and each clock frequency refer to **Table 2-7 Setting of CLKSEL and Frequency of PClock, TClock, VTClock, and MasterOut (Default Value)**.

Table 2-7. Setting of CLKSEL and Frequency of PClock, TClock, VTClock, and MasterOut (Default Value)

CLKSEL(2:0)	PClock (MHz)	VTClock (MHz)	TClock (MHz)	MasterOut (MHz)
111	RFU	RFU	RFU	RFU
110	199.1	33.2	16.6	4.15
101	181.0	30.2	15.1	3.775
100	165.9	33.2	16.6	4.15
011	153.1	30.6	15.3	3.825
010	132.7	33.2	16.6	4.15
001	99.5	33.2	16.6	4.15
000	RFU	RFU	RFU	RFU

2.2.7 Debug serial interface signals

The debug serial interface signals control the transmission of the debug serial data of the VR4131. Some signals have alternate functions as the initialization setting signals set after RTC reset.

Table 2-8. Debug Serial Interface Signals

Signal	I/O	Function
DDIN/GPIO34	I/O	Debug serial reception data input signal. This signal can be used as a general-purpose output port when not being used as the DDIN signal.
DCTS#/GPIO35	I/O	Transmit enable signal. Make this signal active when the RS-232C controller can accept the serial data transfer. This signal can be used as a general-purpose output port when not being used as the DCTS# signal.
DDOUT/ DBUS32/GPIO32	I/O	Functions may differ depending on the operational status • **During normal operation (output)** This signal functions as the debug serial transfer data signal. • **During RTC reset (input)** This signal functions as the data bus width switching signal. When the RTCRST# signal changes from low to high, this signal is sampled. 1: Data bus is used in 32-bit width 0: Data bus is used in 16-bit width This signal can be used as a general-purpose output port when not being used as the DDOUT signal.
DRTS#/ MIPS16EN/ GPIO33	I/O	Functions may differ depending on the operational status • **During normal operation (output)** This signal functions as the debug serial transfer request signal. • **During RTC reset (input)** This signal functions as the MIPS16 instruction enable signal. When the RTCRST# signal changes from low to high, this signal is sampled. 1: MIPS16 instructions enabled 0: MIPS16 instructions disabled This signal can be used as a general-purpose output port when not being used as the DRTS signal.

2.2.8 IrDA interface signals

The IrDA interface signals control the transmission of the IrDA serial data of the VR4131. The IrDA function and the RS-232C function described previously are not used together.

Table 2-9. IrDA Interface Signals

Signal	I/O	Function
IRDIN	I/O	This is an IrDA serial data input signal. It is used when the serial data is transferred from the IrDA controller to the VR4131. Both FIR and SIR can be used. If the IrDA controller used is a Hewlett Packard company product, however, this signal should be used only for SIR.
FIRDIN#/SEL	I/O	This function differs according to the IrDA controller to be used (for how to switch a controller, refer to **20.2.13 SIUIRSEL**). • **Hewlett Packard company's controller or SHARP semiconductor controller** FIRDIN#: It is an FIR receive data input signal. • **TEMIC semiconductor controller** SEL: It is a signal output for the FIR/SIR switching.
IRDOUT#	Output	This is the IrDA serial data output signal. It is used when the serial data is transferred from the VR4131 to the IrDA controller.

2.2.9 Clocked serial signals

The clocked serial signals are 3-wire clocked serial signals of the VR4131.

Table 2-10. Clocked Serial Signals

Signal	I/O	Function
SIN	Input	Clocked serial input signal
SOUT	Output	Clocked serial output signal
SECLK	Output	Synchronous clock output for the clocked serial

2.2.10 General-purpose I/O signals

These are general-purpose I/O pins of the VR4131. Normally, 21 of the 35 general-purpose I/O pins are used as alternate-function pins.

Table 2-11. General-Purpose I/O Signals

Signal	I/O	Function
GPIO(0:3)	I/O	Maskable activation factor input signals. These signals can be used as GPIO ports after activation.
GPIO(4:5)	I/O	General-purpose I/O pins.
SYSDIR/GPIO6	I/O	General-purpose I/O pins. These pins can also be used to input/output the buffer control signals of SDRAM by setting the appropriate register after activation.
GPIO(7:8)	I/O	General-purpose I/O pins.
GPIO(9:12)	I/O	Maskable activation factor input signals. These signals can be used as general-purpose I/O ports after activation.
GPIO(13)	I/O	General-purpose I/O pin. This signal is recommended to be used as a VRC4173 interrupt.
DCD#/GPIO15	Input	Refer to **2.2.6 RS-232C interface signals**.
DATA(16:31)/ GPIO(16:31)	I/O	Refer to **2.2.1 Memory interface signals**.
DDOUT/DBUS32/GPIO32	I/O	Refer to **2.2.7 Debug serial interface signals**.
DRTS#/MIPS16EN/GPIO33	I/O	Refer to **2.2.7 Debug serial interface signals**.
DDIN/GPIO34	I/O	Refer to **2.2.7 Debug serial interface signals**.
DCTS#/GPIO35	I/O	Refer to **2.2.7 Debug serial interface signals**.

2.2.11 LED interface signal

Table 2-12. LED Interface Signal

Signal	I/O	Function
LEDOUT#	O	This is an output signal for lighting LEDs.

2.2.12 PCI Like bus interface signals

These are the signals that comply with the VR4131 PCI bus. This PCI Like bus interface is compliant with PCI revision 2.1, thus allowing connection of up to three master PCI devices.

Table 2-13. PCI Like Bus Interface Signals

Signal	I/O	Function
AD(0:31)	I/O	This is a 32-bit address bus and data bus. In the address phase, addresses are output, and in the data phase, data is output.
CBE(0:3)	I/O	These are the bus-command/byte-enable signals. In the address phase, bus commands are output, and in the data phase, they operate as the byte-enable signals.
DEVSEL#	I/O	This signal is asserted when the target is accessed and continues being asserted until the completion of the transaction.
FRAME#	I/O	This signal is asserted when the initiator starts the transaction. It also remains asserted throughout burst transfer.
REQ(0:2)#	Input	These signals are asserted when the master sends a request to the VR4131 for the right to use the bus.
GNT(0:2)#	Output	These signals are asserted when the VR4131 grants permission to use the bus to the device making the request with the REQ# signal.
IRDY#	I/O	This signal is asserted when the initiator is in the data transfer enabled state.
LOCK#	I/O	This signal indicates a resource lock.
PAR	I/O	This signal outputs L if the number of “1” bits from the 36 AD(0:31) and CBE(0:3)# signals is even, and H if the number is odd.
PERR#	I/O	This signal is asserted when a parity error occurs following a parity check by the data-read initiator in the read cycle or the data-write target in the write cycle.
SERR#	I/O	This signal is asserted when an error that is critical to the system occurs.
STOP#	I/O	This signal is asserted when the target requires the initiator to abort the transaction.
TRDY#	I/O	This signal is asserted when the target is in the transfer-enabled state.
PCLK	Output	This is the PCI bus reference clock.
CLKRUN	I/O	This signal controls the clock for power management.
RST#	Output	This is the PCI bus reset signal.

2.2.13 Dedicated V_{DD} and GND signals

Table 2-14. Dedicated V_{DD} and GND Signals

Signal Name	Power Supply	Function
V_{DD}P	1.5 V	Dedicated V_{DD} for the PLL analog unit
GNDP	1.5 V	Dedicated GND for the PLL analog unit
V_{DD}PD	1.5 V	Dedicated V_{DD} for the PLL digital unit. Its function is identical to V_{DD}1.
GNDPD	1.5 V	Dedicated GND for the PLL digital unit. Its function is identical to GND1.
CV_{DD}	3.3 V	Dedicated V_{DD} for the oscillator
CGND	3.3 V	Dedicated GND for the oscillator
V_{DD}1	1.5 V	Normal 1.5 V V_{DD}
GND1	1.5 V	GND for normal 1.5 V system
V_{DD}3	3.3 V	Normal 3.3 V V_{DD}
GND3	3.3 V	GND for normal 3.3 V system

2.2.14 Other signals

Table 2-15. Other Signals

Signal Name	I/O	Function
BIGENDIAN	Input	Selects big endian. 1: Big endian (pull up[Note]) 0: Little endian (pull down[Note])
MMUEN	Input	Selects MMUEN function. 1: MMU enabled (pull up[Note]) 0: MMU disabled (pull down[Note])

Note When pulling up, the pin must be connected to V_{DD}3 via a resistor. When pulling down, the pin may be directly connected to GND3.

2.3 Pin Status

2.3.1 Pin status in specific states

Table 2-16 lists the pin status after the VR4131 is reset or when it is in the power mode.

Table 2-16. Pin Status in Specific States (1/4)

Pin Name	When Reset by RTCRST	In Hibernate Mode or During HALTimer Shutdown	When Reset by RSTSW	In Suspend Mode	During Bus Hold
AD(31:0)	0	0	0	Hold	**Note 1**
ADD(24:1)	0	0	0	**Note 2**	Hi-Z
BATTINH/BATTINT#	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
BIGENDIAN	Hi-Z	0	0	0	Hold
CAS	0	0	0	**Note 2**	Hi-Z
CBE(3:0)	0	0	0	Hold	**Note 1**
CKE(1:0)	0	0	0	**Note 2**	Hi-Z
CLKOUT	0	0	CLK	CLK	CLK
CLKRUN	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
CS(1:0)#	Hi-Z	1	1	**Note 2**	Hi-Z
CS2#/ROMCS2#	Hi-Z	**Note 3**	1	**Note 2**	**Note 4**
CS3#/ROMCS3#	Hi-Z	**Note 5**	1	**Note 2**	**Note 6**
CTS#	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
DATA(15:0)	0	0	0	**Note 2**	Hi-Z

Notes 1. Normal operation.

2. Maintains the previous status.
For the pin status during the bus hold period, however, refer to the During Bus Hold column.

3. Depends on the status of the BCUCNTREG3 register's EXT_ROMCS0 bit and the DBUS32 pin.
When the EXT_ROMCS0 bit is 0 and the DBUS32 pin = 1: High level
If a combination other than above: High impedance

4. Depends on the status of the BCUCNTREG3 register's EXT_ROMCS0 bit and the DBUS32 pin.
When the EXT_ROMCS0 bit is 0 and the DBUS32 pin = 1: High impedance
If a combination other than above: High level

5. Depends on the status of the BCUCNTREG3 register's EXT_ROMCS1 bit and the DBUS32 pin.
When the EXT_ROMCS1 bit is 0 and DBUS32 = 1: High level
If a combination other than above: high impedance

6. Depends on the status of the BCUCNTREG3 register's EXT_ROMCS1 bit and the DBUS32 pin.
When the EXT_ROMCS1 bit is 0 and DBUS32 = 1: High impedance
If a combination other than above: high level

Remark 0: Low level, 1: High level, Hi-Z: High impedance, Hold: Maintains the status of the preceding Fullspeed mode

Table 2-16. Pin Status upon Specific States (2/4)

Pin Name	When Reset by RTCRST	In Hibernate Mode or During HALTimer Shutdown	When Reset by RSTSW	In Suspend Mode	During Bus Hold
DATA(31:16)/GPIO(31:16)	**Note 1**	**Note 1**	**Note 1**	**Note 2**	**Note 3**
DCD#/GPIO15	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
DCTS#/GPIO35	Hi-Z	**Note 4**	**Note 4**	Hold	Hold
DDIN/GPIO34	Hi-Z	**Note 5**	**Note 5**	Hold	Hold
DDOUT/DBUS32/GPIO32	Hi-Z	**Note 6**	**Note 6**	Hold	Hold
DEVSEL#	Hi-Z	Hi-Z	Hi-Z	Hold	**Note 7**
DQM(3:0)	Hi-Z	0	0	**Note 2**	Hi-Z
DRTS#/MIPS16EN/GPIO33	Hi-Z	**Note 8**	**Note 8**	Hold	Hold
DSR#	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
DTR#/CLKSEL0	Hi-Z	1	1	Hold	Hold
FIRCLK	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
FIRDIN#/SEL	Hi-Z	Hi-Z	Hi-Z	Hold	Hold
FRAME#	Hi-Z	Hi-Z	Hi-Z	Hold	**Note 7**

Notes

1. When the DBUS32 bit is 1: Low level
When the DBUS32 bit is 0: High impedance

2. Maintain the previous status.
For the pin status during the bus hold period, however, refer to the During Bus Hold column.

3. When the DBUS32 bit is 1: High impedance
When the DBUS32 bit is 0: Maintains the previous status

4. Depends on the status of the GIUPODATEN register’s PIOEN35 bit.
When the PIOEN35 bit is 0: High impedance
When the PIOEN35 bit is 1: Maintains the status of the preceding Fullspeed mode

5. Depends on the status of the GIUPODATEN register’s PIOEN34 bit.
When the PIOEN34 bit is 0: High impedance
When the PIOEN34 bit is 1: Maintains the status of the preceding Fullspeed mode

6. Depends on the status of the GIUPODATEN register’s PIOEN32 bit.
When the PIOEN32 bit is 0: High level
When the PIOEN32 bit is 1: Maintains the status of the preceding Fullspeed mode

7. Normal operation

8. Depends on the status of the GIUPODATEN register’s PIOEN33 bit.
When the PIOEN33 bit is 0: High level
When the PIOEN33 bit is 1: Maintains the status of the preceding Fullspeed mode

Remark 0: Low level, 1: High level, Hi-Z: High impedance, Hold: Maintains the status of the preceding Fullspeed mode

Table 2-16. Pin Status upon Specific States (3/4)

Pin Name	When Reset by RTCRST	In Hibernate Mode or During HALTimer Shutdown	When Reset by RSTSW	In Suspend Mode	During Bus Hold
GNT(2:0)#	Hi-Z	Hi-Z	Hi-Z	Hold	Hold
GPIO(5:0)	Hi-Z	Hi-Z	Hi-Z	Hold	Hold
GPIO6/SYSDIR	0	**Note 1**	**Note 1**	**Note 2**	**Note 3**
GPIO(13:7)	Hi-Z	Hi-Z	Hi-Z	Hold	Hold
IOCS(1:0)#	Hi-Z	Hi-Z	1	1	1
IORDY	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
IRDIN	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
IRDOUT#	0	0	0	0	0
IRDY#	Hi-Z	Hi-Z	Hi-Z	Hold	**Note 4**
LEDOUT#	Hi-Z	1	1	1	**Note 4**
LOCK#	Hi-Z	Hi-Z	Hi-Z	Hi-Z	**Note 4**
MPOWER	0	0	1	1	1
MMUEN	Hi-Z	0	0	0	Hold
PAR	0	0	**Note 5**	Hold	**Note 4**
PCLK	0	0	0	Hold	**Note 4**
PERR#	Hi-Z	Hi-Z	Hi-Z	Hi-Z	**Note 4**
POWER	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
POWERON	0	0	0	0	0
RAS	0	0	0	**Note 2**	Hi-Z
RD#	Hi-Z	Hi-Z	1	**Note 2**	Hi-Z
REQ(2:0)#	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
ROMCS(1:0)#	Hi-Z	Hi-Z	1	1	1
RST#	0	0	0	Hold	**Note 6**
RSTSW#	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
RTCRST#	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z

Notes 1. Depends on the setting of the BCUCNTREG3 register's SYSDIR_EN bit.
When the SYSDIR_EN bit is 1: Low level
When the SYSDIR_EN bit is 0: High impedance

2. Maintains the previous status.
For the pin status during the bus hold period, however, refer to the During Bus Hold column.

3. Depends on the setting of the BCUCNTREG3 register's SYSDIR_EN bit.
When the SYSDIR_EN bit is 1: High impedance
When the SYSDIR_EN bit is 0: Maintains the previous status

4. Normal operation

5. Undefined. Drive either a low or high level.

6. Maintains the previous status.

Remark 0: Low level, 1: High level, Hi-Z: High impedance, Hold: Maintains the status of the preceding Fullspeed mode

Table 2-16. Pin Status upon Specific States (4/4)

Pin Name	When Reset by RTCRST	In Hibernate Mode or During HALTimer Shutdown	When Reset by RSTSW	In Suspend Mode	During Bus Hold
RTS#/CLKSEL1	Hi-Z	1	1	1	Hold
RxD	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
SCLK	0	0	**Note 1**	**Note 2**	Hi-Z
SECLK	0	0	1	Hold	Hold
SERR#	Hi-Z	Hi-Z	Hi-Z	Hi-Z	**Note 3**
SIN	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
SOUT	0	0	0	1	1
SPOWER	0	1	1	1	1
STOP#	Hi-Z	Hi-Z	Hi-Z	Hi-Z	**Note 3**
SWR#	Hi-Z	0	0	**Note 2**	Hi-Z
HLDRQ#	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
HLDAK#	Hi-Z	Hi-Z	Hi-Z	**Note 4**	0
TRDY#	Hi-Z	Hi-Z	Hi-Z	Hold	**Note 3**
TxD/CLKSEL2	Hi-Z	1	1	1	Hold
WR#	Hi-Z	Hi-Z	1	**Note 3**	Hi-Z

Notes **1.** Outputs clock.

2. Maintains the previous status.
For the pin status during the bus hold period, however, refer to the During Bus Hold column.

3. Normal operation

4. Depends on the setting of the BCUCNTREG1 register's HLDEN bit.
When the HLDEN bit is 1: Normal operation as HLDAK# pin
When the HLDEN bit is 0: High impedance

Remark 0: Low level, 1: High level, Hi-Z: High impedance, Hold: Maintains the status of the preceding Fullspeed mode

2.3.2 Pin Handling and I/O Circuit Types

Table 2-17. Pin Handling and I/O Circuit Types (1/2)

Pin Name	Pin Handling	Recommended Connection of Unused Pins	Drive Capability	I/O Circuit Type
AD(31:0)	–	Leave open	–	A
ADD(24:1)	–	–	**Note 1**	A
BATTINH/BATTINT#	–	–	–	B
BIGENDIAN	**Note 2**	–	–	A
CAS	–	–	–	A
CBE(3:0)	–	Leave open	–	A
CKE(1:0)	–	–	120 pF	A
CLKOUT	–	Leave open	–	A
CLKRUN	Pull up	Connect to V_{DD} via a resistor	–	A
CS(1:0)#	–	–	120 pF	A
CS(3:2)#/ROMCS(3:2)#	–	Leave open	120 pF	A
CTS#	–	Connect to V_{DD} or GND	–	A
DATA(15:0)	–	–	120 pF	A
DATA(31:16)/GPIO(31:16)	–	Connect V_{DD} or GND via a resistor	120 pF	A
DCD#/GPIO15	–	–	–	B
DCTS#/GPIO35	–	Connect V_{DD} or GND via a resistor	–	A
DDIN/GPIO34	–	Connect V_{DD} or GND via a resistor	–	A
DDOUT/DBUS32/GPIO32	Pull up/pull down	–	–	A
DEVSEL#	Pull up	Connect to V_{DD} via a resistor	–	A
DQM(3:0)	–	–	120 pF	A
DRTS#/MIPS16EN/GPIO33	Pull up/pull down	–	–	A
DSR#	–	Connect to V_{DD} or GND	–	A
DTR#/CLKSEL0	Pull up/pull down	–	–	A
FIRCLK	–	Connect to GND	–	A
FIRDIN#/SEL	–	Connect V_{DD} or GND via a resistor	–	A
FRAME#	Pull up	Connect to V_{DD} via a resistor	–	A
GNT(2:0)#	–	Leave open	–	A
GPIO(0:2)	–	Connect V_{DD} or GND via a resistor	–	B
GPIO3	–	–	–	B
GPIO(4:5)	–	Connect V_{DD} or GND via a resistor	–	B
GPIO6/SYSDIR	–	Leave open	–	B
GPIO(7:13)	–	Connect V_{DD} or GND via a resistor	–	B
IOCS(1:0)#	–	Leave open	–	A

Notes 1. The drive capability of ADD (4:1) is 120 pF and is 40 pF for the rest.

2. Pull the pin up or down. For pull-up processing, the pin must be connected to $V_{DD}3$ via a resistor. For pull-down, the pin can be directly to GND3.

Remarks 1. No external processing is required for the pins for which no instruction is described (–) in the Pin Handling column.

2. For pins for which any instruction is specified in the Recommended Connection of Unused Pins column (–), follow the instruction provided in the Pin Handling column.

Table 2-17. Pin Handling and I/O Circuit Types (2/2)

Pin Name	Pin Handling	Recommended Connection of Unused Pins	Drive Capability	I/O Circuit Type
IORDY	–	Connect V_{DD} or GND via a resistor	–	A
IRDIN	–	Connect V_{DD} or GND via a resistor	–	A
IRDOUT#	–	Leave open	–	A
IRDY#	Pull up	Connect to V_{DD} via a resistor	–	A
LEDOUT	Pull up	–	–	A
LOCK#	Pull up	Connect to V_{DD} via a resistor	–	A
MPOWER	–	–	–	A
MMUEN	**Note**	–	–	A
PAR	–	Leave open	–	A
PCLK	–	Leave open	–	A
PERR#	Pull up	Connect to V_{DD} via a resistor	–	A
POWER	–	–	–	B
POWERON	–	–	–	A
RAS	–	–	–	A
RD#	–	–	–	A
REQ(2:0)#	–	Connect to V_{DD}	–	A
ROMCS(1:0)#	–	–	120 pF	A
RST#	–	Leave open	–	A
RSTSW#	–	–	–	B
RTCRST#	–	–	–	B
RTS#/CLKSEL1	Pull up/pull down	–	–	A
RxD	–	Connect to V_{DD} or GND	–	A
SCLK	–	–	120 pF	A
SECLK	–	Leave open	–	A
SERR#	Pull up	Connect to V_{DD} via a resistor	–	A
SIN	–	Connect to V_{DD} or GND	–	A
SOUT	–	Leave open	–	A
SPOWER	–	–	–	A
STOP#	Pull up	Connect to V_{DD} via a resistor	–	A
SWR#	–	–	–	A
HLDRQ#	–	Leave open	–	A
HLDAK#	Pull down	–	–	A
TRDY#	Pull up	Connect to V_{DD} via a resistor	–	A
TxD/CLKSEL2	Pull up/pull down	–	–	A
WR#	–	Leave open	–	A

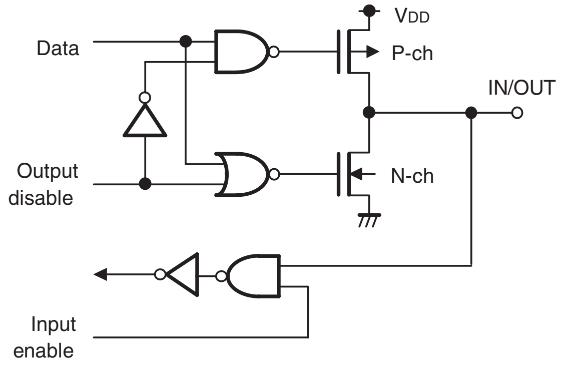
Note Pull up or pull down the pin. For pull up processing, the pin must be connected to VDD3 via a resistor. For pull down, the pin can be directly to GND3.

Remarks 1. No external processing is required for the pins to which no instruction is described (–) in the Pin Handling column.

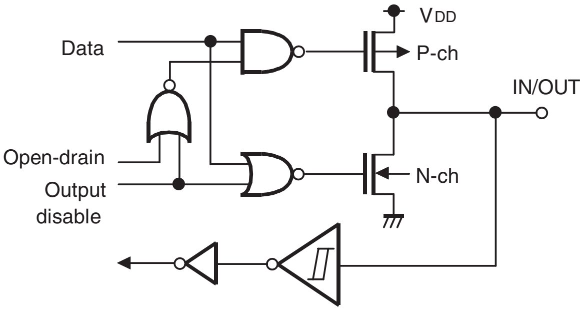
2. For pins for which no instruction is specified in the Recommended Connection of Unused Pins column (–), follow the instruction provided in the Pin Handling column.

2.3.3 Pin I/O circuits

Type A



Type B



CHAPTER 3 MEMORY MANAGEMENT SYSTEM

The VR4131 provides a memory management unit (MMU) that uses a translation lookaside buffer (TLB) to translate virtual addresses into physical addresses. This chapter describes the physical addresses. For the operation of the TLB and the memory mapping method used to translate virtual addresses into physical addresses, refer to the **VR4100 Series Architecture User's Manual** available separately. For the CP0 registers used as the software interface with the TLB, refer to **CHAPTER 5 CP0 REGISTERS**.

3.1 Physical Address Space

Because the VR4130 core uses a 32-bit address, the processor physical address space encompasses 4 GB. The VR4131 uses this 4 GB physical address space as shown in Figure 3-1.

Figure 3-1. VR4131 Physical Address Space

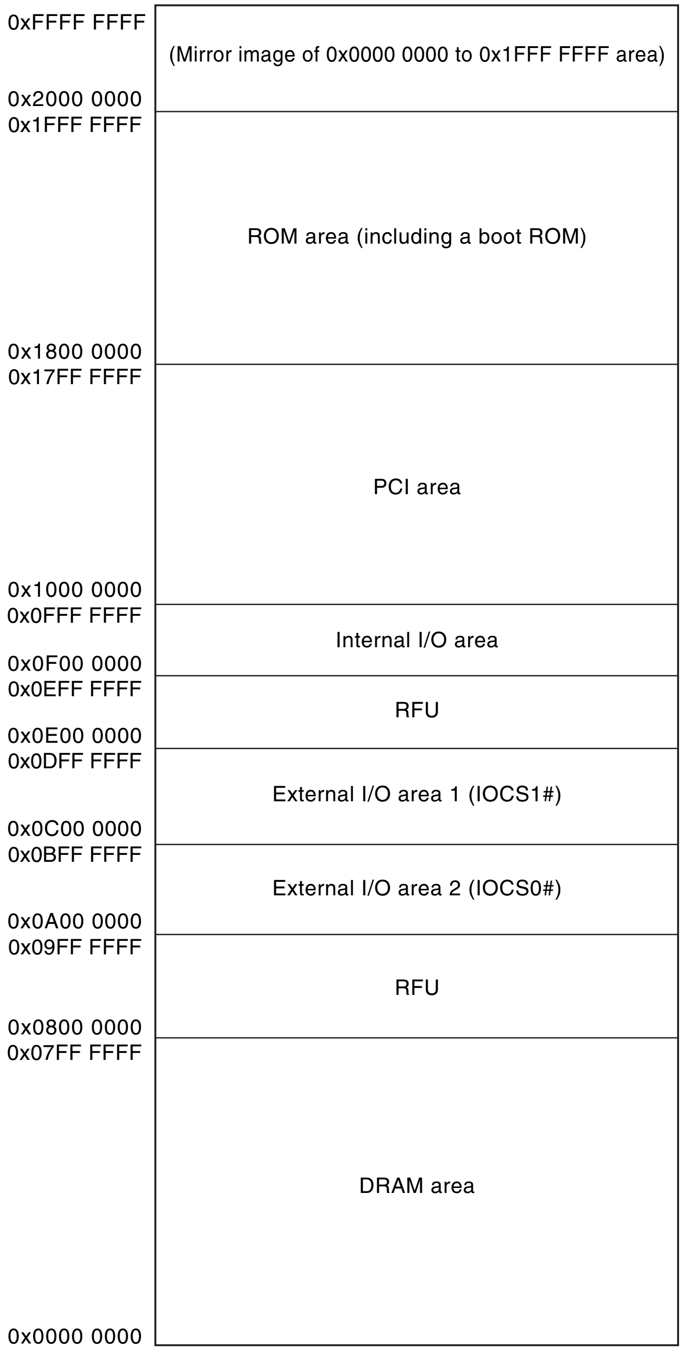


Table 3-1. VR4131 Physical Address Space

Physical Address	Space	Capacity (Bytes)
0xFFFF FFFF to 0x2000 0000	Mirror image of 0x1FFF FFFF to 0x0000 0000	3.5 G
0x1FFF FFFF to 0x1800 0000	ROM space	128 M
0x17FF FFFF to 0x1000 0000	PCI space	128 M
0x0FFF FFFF to 0x0F00 0000	Internal I/O space	16 M
0x0EFF FFFF to 0x0E00 0000	RFU	16 M
0x0DFF FFFF to 0x0C00 0000	External I/O space 1 (IOCS1#)	32 M
0x0BFF FFFF to 0x0A00 0000	External I/O space 2 (IOCS0#)	32 M
0x09FF FFFF to 0x0800 0000	RFU	32 M
0x07FF FFFF to 0x0000 0000	DRAM space	128 M

3.1.1 ROM address space

The VR4131's ROM space is divided into four banks. These four banks are allocated according to the bit width of the data bus as described below.

- In 16-bit bus mode: All banks are allocated in ordinary memory space.
- In 32-bit bus mode: Banks 0 and 1 are allocated in ordinary memory space and banks 2 and 3 are allocated in expansion space.

(1) Capacity of ROM space

The ROM address space varies in size depending on the data bus's bit width and the capacity of the ROM being used.

- The data bus bit width is set via the DBUS32 pin.
- The ROM capacity can be set using the SIZE bit of the ROMSIZEREG register.

(a) Capacity setting via ROMSIZEREG register

The capacity can be selected for each bank from any of 4 MB, 8 MB, 16 MB, 32 MB, or 64 MB.

(2) Mapping of physical address

The area for each bank is determined based on the data bus bit width and the bank capacity, as described below.

(a) In 16-bit bus mode (DBUS32 = 0)

Bank 3 (CS3#/ROMCS3#): 0x1FFF FFFF to 0x1FFF FFFF – (Bank 3 capacity)

Bank 2 (CS2#/ROMCS2#): 0x1FFF FFFF – (Bank 3 capacity + 1 byte) to 0x1FFF FFFF – (Bank 3 capacity + Bank 2 capacity)

Bank 1 (ROMCS1#): 0x1FFF FFFF – (Bank 3 capacity + Bank 2 capacity + 1 byte) to 0x1FFF FFFF – (Bank 3 capacity + Bank 2 capacity + Bank 1 capacity)

Bank 0 (ROMCS0#): 0x1FFF FFFF – (Bank 3 capacity + Bank 2 capacity + Bank 1 capacity + 1 byte) to 0x1FFF FFFF – (Bank 3 capacity + Bank 2 capacity + Bank 1 capacity + Bank 0 capacity)

Physical addresses in the ROM address space are indicated below for both memory spaces, when using 32 Mbit ROM or 64 Mbit ROM.

Table 3-2. ROM Address Example (When Using 16-Bit Data Bus)

Physical Address	ADD(24:1) Pin	When Using 32 Mb ROM (SIZE3 = SIZE2 = SIZE1 = SIZE0 = 1)	When Using 64 Mb ROM (SIZE3 = SIZE2 = SIZE1 = SIZE0 = 2)
0x1FFF FFFF to 0x1FC0 0000	0x1FF FFFF to 0x1C0 0000	Bank 3 (CS3#/ROMCS3#)	Bank 3 (CS3#/ROMCS3#)
0x1FBF FFFF to 0x1F80 0000	0x1BF FFFF to 0x180 0000	Bank 2 (CS2#/ROMCS2#)	
0x1F7F FFFF to 0x1F40 0000	0x17F FFFF to 0x140 0000	Bank 1 (ROMCS1#)	Bank 2 (CS2#/ROMCS2#)
0x1F3F FFFF to 0x1F00 0000	0x13F FFFF to 0x100 0000	Bank 0 (ROMCS0#)	
0x1EFF FFFF to 0x1E80 0000	0x0FF FFFF to 0x080 0000	RFU	Bank 1 (ROMCS1#)
0x1E7F FFFF to 0x1E00 0000	0x07F FFFF to 0x000 0000		Bank 0 (ROMCS0#)
0x1DFF FFFF to 0x1800 0000	0x1FF FFFF to 0x1C0 0000		RFU

Remark The lowest bit of the values in the ADD (24:1) pin column is insignificant.

(b) In 32-bit bus mode (DBUS32 = 1)

Bank 1 (ROMCS1#): 0x1FFF FFFF to 0x1FFF FFFF – (Bank 1 capacity)

Bank 0 (ROMCS0#): 0x1FFF FFFF – (Bank 1 capacity + 1 byte) to 0x1FFF FFFF – (Bank 1 capacity + Bank 0 capacity)

Bank 3 (CS3#/ROMCS3#): 0x1FFF FFFF – (Bank 1 capacity + Bank 0 capacity + 1 byte) to 0x1FFF FFFF – (Bank 1 capacity + Bank 0 capacity + Bank 3 capacity)

Bank 2 (CS2#/ROMCS2#): 0x1FFF FFFF – (Bank 1 capacity + Bank 0 capacity + Bank 3 capacity + 1 byte) to 0x1FFF FFFF – (Bank 1 capacity + Bank 0 capacity + Bank 3 capacity + Bank 2 capacity)

Physical addresses in the ROM address space are indicated in the following table when using 32 Mb ROM (banks 3 and 2), or 32 Mb or 64 Mb ROM (banks 1 and 0).

Table 3-3. Example of ROM Addresses When Using 32 Mb Expansion ROM (When Using 32-Bit Data Bus)

Physical Address	ADD(24:1) Pin[Note 1]	When Using 32 Mb ROM (SIZE3 = SIZE2 = SIZE1 = SIZE0 = 2)	When Using 64 Mb ROM (SIZE1 = SIZE0 = 3, SIZE3 = SIZE2 = 2)
0x1FFF FFFF to 0x1F80 0000	0x3FF FFFF to 0x380 0000	Bank 1 (ROMCS1#)	Bank 1 (ROMCS1#)
0x1F7F FFFF to 0x1F00 0000	0x37F FFFF to 0x300 0000	Bank 0 (ROMCS0#)	
0x1EFF FFFF to 0x1E80 0000	0x2FF FFFF to 0x280 0000	Bank 3 (CS3#/ROMCS3#)[Note 2]	Bank 0 (ROMCS0#)
0x1E7F FFFF to 0x1E00 0000	0x27F FFFF to 0x200 0000	Bank 2 (CS2#/ROMCS2#)[Note 2]	
0x1DFF FFFF to 0x1D80 0000	0x1FF FFFF to 0x180 0000	RFU	Bank 3 (CS3#/ROMCS3#)[Note 2]
0x1D7F FFFF to 0x1D00 0000	0x17F FFFF to 0x100 0000		Bank 2 (CS2#/ROMCS2#)[Note 2]
0x1CFF FFFF to 0x1800 0000	0x0FF FFFF to 0x000 0000		RFU

Notes 1. When ROM is accessed using the 32-bit data bus, the ADD(1) pin functions as the ADD(25) pin.

2. Can be used exclusively from the expansion DRAM.

Remark The lowest two bits of the values in the ADD (24:1) pin column are insignificant.

Physical addresses in the ROM address space are indicated in the following table when using 64 Mb ROM (banks 3 and 2), or 32 Mb or 64 Mb ROM (bank 1 and 0).

Table 3-4. Example of ROM Addresses When Using 64 Mb Expansion ROM (When Using 32-Bit Data Bus)

Physical Address	ADD(24:1) Pin[Note 1]	When Using 32 Mb ROM (SIZE1 = SIZE0 = 2, SIZE3 = SIZE2 = 3)	When Using 64 Mb ROM (SIZE3 = SIZE2 = SIZE1 = SIZE0 = 3)
0x1FFF FFFF to 0x1F80 0000	0x3FF FFFF to 0x380 0000	Bank 1 (ROMCS1#)	Bank 1 (ROMCS1#)
0x1F7F FFFF to 0x1F00 0000	0x37F FFFF to 0x300 0000	Bank 0 (ROMCS0#)	
0x1EFF FFFF to 0x1E00 0000	0x2FF FFFF to 0x200 0000	Bank 3 (CS3#ROMCS3#)[Note 2]	Bank 0 (ROMCS0#)
0x1DFF FFFF to 0x1D00 0000	0x1FF FFFF to 0x100 0000	Bank 2 (CS2#ROMCS2#)[Note 2]	Bank 3 (CS3#ROMCS3#)[Note 2]
0x1CFF FFFF to 0x1C00 0000	0x0FF FFFF to 0x000 0000	RFU	Bank 2 (CS2#ROMCS2#)[Note 2]
0x1BFF FFFF to 0x1800 0000	–		RFU

Notes 1. When ROM is accessed using the 32-bit data bus, the ADD(1) pin functions as the ADD(25) pin.

2. Can be used exclusively from the expansion DRAM.

Remark The lowest two bits of the values in the ADD (24:1) pin column are insignificant.

Physical addresses in the ROM address space are indicated below for when using 64 Mb ROM or 128 Mb ROM (all banks).

Table 3-5. Example of ROM Addresses When Using 128 Mb Expansion ROM (When Using 32-Bit Data Bus)

Physical Address	ADD(24:1) Pin[Note 1]	When Using 64 Mb ROM (SIZE3 = SIZE2 = SIZE1 = SIZE0 = 3)	When Using 128 Mb ROM (SIZE3 = SIZE2 = SIZE1 = SIZE0 = 4)
0x1FFF FFFF to 0x1F00 0000	0x3FF FFFF to 0x300 0000	Bank 1 (ROMCS1#)	Bank 1 (ROMCS1#)
0x1EFF FFFF to 0x1E00 0000	0x2FF FFFF to 0x200 0000	Bank 0 (ROMCS0#)	
0x1DFF FFFF to 0x1D00 0000	0x1FF FFFF to 0x100 0000	Bank 3 (CS3#/ROMCS3#)[Note 2]	Bank 0 (ROMCS0#)
0x1CFF FFFF to 0x1C00 0000	0xFF FFFF to 0x000 0000	Bank 2 (CS2#/ROMCS2#)[Note 2]	
0x1BFF FFFF to 0x1A00 0000	0x3FF FFFF to 0x200 0000	RFU	Bank 3 (CS3#/ROMCS3#)[Note 2]
0x19FF FFFF to 0x1800 0000	0x1FF FFFF to 0x000 0000		Bank 2 (CS2#/ROMCS2#)[Note 2]

Notes 1. When ROM is accessed using the 32-bit data bus, the ADD(1) pin functions as the ADD(25) pin.

2. Can be used exclusively from the expansion DRAM.

Remark The lowest two bits of the values in the ADD (24:1) pin column are insignificant.

3.1.2 PCI bus address space

The VR4131 has an internal I/O space. The following tables show each address space.

- PCI bus I/O space
  This corresponds to the PCI I/O space.

- PCI bus memory space
  This corresponds to the PCI memory space.

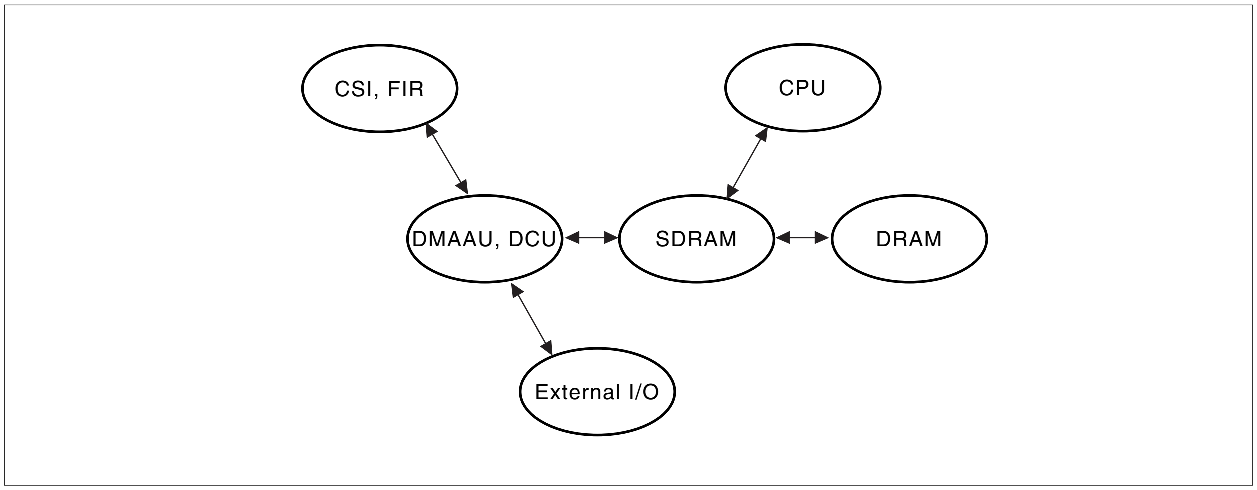
3.1.3 DMA

The VR4131 provides a DCU (DMA control unit) and DMAAU (DMA address unit) for controlling DMA transfer.

(1) Transfer flow

Figure 3-2 shows the types of DMA transfer.

Figure 3-2. DMA Transfer Types



(2) Transfer units

Unit	Transfer Unit
CSI	16 bits x 2 times
FIR	8 bits

(3) Register settings

Make the following settings to perform DMA transfer.

(a) DMA transfer of CSI

• Transmission

<1> Set the DMASEN bit of the DMASENREG register of the DCU to 1 to enable DMA sequencer operation.

<2> Set the DMAMSKCOUT bit of the DMAMSKREG register of the DCU to 1 to enable CSI-transmission DMA transfer.

<3> Set the T_FIFOE and T_DMAEN bits of the CSI_CNTREG register of the CSI to 1 to enable the transmit FIFO and transmission DMA of the CSI.

<4> The transmit data from the memory area set to the CSIOBALREG and CSIOBAHREG registers of the CSI is DMA transferred.

• Reception

<1> Set the DMASEN bit of the DMASENREG register of the DCU to 1 to enable DMA sequencer operation.

<2> Set the DMAMSKCIN bit of the DMAMSKREG register of the DCU to 1 to enable CSI-reception DMA transfer.

<3> Set the R_FIFOE and R_DMAEN bits of the CSI_CNTREG register of the CSI to 1 to enable the receive FIFO and reception DMA of the CSI.

<4> When the CSI_SIRBREG register of the CSI is read, the receive data is DMA transferred and written to the memory area set to the CSIIALREG and CSIIAHREG registers.

(b) FIR

<1> Set the DMASEN bit of the DMASENREG register of the DCU to 1 to enable the DMA sequencer operation.

<2> Set the IRDA_EN bit of the IRSR1 register of the FIR to 1 and set the DMAMSKFOUT bit of the DMAMSKREG register of the DCU to 1 to enable FIR-transmission DMA.

<3> Set the transfer direction for the DMA channel with the FIR bit of the TDREG register of the DCU.

<4> Set the start address of the memory area to the FIRBALREG and FIRBAHREG registers of the DMAAU.

<5> Set the TX_EN bit (transmission) and the RX_EN bit (reception) of the CRCSR register of the FIR to 1 to enable transmission and reception.

<6> Transmission starts when the DMA_EN bit of the DMAER register of the FIR is set to 1 to enable DMA operation.

3.1.4 Internal I/O space

Table 3-6 shows the interval I/O space of VR4131.

Table 3-6. Internal I/O Space 1

Physical Address	Internal I/O
0x0F00 13FF to 0x0F00 1020	RFU
0x0F00 101F to 0x0F00 1000	SCU
0x0F00 0FFF to 0x0F00 0E00	RFU
0x0F00 0DFF to 0x0F00 0C00	PCI
0x0F00 0BFF to 0x0F00 0880	RFU
0x0F00 087F to 0x0F00 0860	FIR2
0x0F00 085F to 0x0F00 0840	FIR1
0x0F00 083F to 0x0F00 0820	DSIU (16550 simple version)
0x0F00 081F to 0x0F00 0800	SIU (equivalent to 16550)
0x0F00 07FF to 0x0F00 0420	RFU
0x0F00 041F to 0x0F00 0400	SDRAMU
0x0F00 03FF to 0x0F00 0200	RFU
0x0F00 01FF to 0x0F00 01E0	DMAAU2
0x0F00 01DF to 0x0F00 01C0	CSI2
0x0F00 01BF to 0x0F00 01A0	CSI1
0x0F00 019F to 0x0F00 0180	LED
0x0F00 017F to 0x0F00 0160	GIU2
0x0F00 015F to 0x0F00 0140	GIU1
0x0F00 013F to 0x0F00 0120	RTC2
0x0F00 011F to 0x0F00 0100	RTC1
0x0F00 00FF to 0x0F00 00E0	PMU2
0x0F00 00DF to 0x0F00 00C0	PMU1
0x0F00 00BF to 0x0F00 00A0	ICU2
0x0F00 009F to 0x0F00 0080	ICU1
0x0F00 007F to 0x0F00 0060	CMU
0x0F00 005F to 0x0F00 0040	DCU
0x0F00 003F to 0x0F00 0020	DMAAU1
0x0F00 001F to 0x0F00 0000	BCU

3.1.5 External I/O address space

The VR4131 has two external I/O spaces. The following table shows each address space.

Table 3-7. External I/O Space

Physical Address	External I/O
0x0DFF FFFF to 0x0C00 0000	External I/O space 1 (IOCS1#)
0x0BFF FFFF to 0x0A00 0000	External I/O space 2 (IOCS0#)

3.1.6 DRAM address space

The VR4131's DRAM space is divided into four banks. These four banks are allocated according to the bit width of the data bus as described below.

- In 16-bit bus mode: All banks are allocated in ordinary memory space.
- In 32-bit bus mode: Banks 0 and 1 are allocated in ordinary memory space and banks 2 and 3 are allocated in expansion space.

(1) Capacity of DRAM space

The DRAM address space varies in size depending on the data bus's bit width and the capacity of the DRAM being used.

- The data bus bit width is set via the DBUS32 pin.
- The DRAM capacity can be set using the SIZE bit of the RAMSIZEREG register.

(a) Capacity setting via RAMSIZEREG register

The capacity can be selected for each bank from any of 8 MB, 16 MB, 32 MB, or 64 MB.

(2) Mapping of physical address

The area for each bank is determined based on the data bus bit width and the bank capacity, as described below.

(a) In 16-bit bus mode (DBUS32 = 0)

Bank 3 (CS3#/ROMCS3#): 0x0000 0000 + (Bank 3 capacity + Bank 2 capacity + Bank 1 capacity + Bank 0 capacity) to 0x0000 0000 + (Bank 3 capacity + Bank 1 capacity + Bank 0 capacity + 1 byte)

Bank 2 (CS2#/ROMCS2#): 0x0000 0000 + (Bank 2 capacity + Bank 1 capacity + Bank 0 capacity) to 0x0000 0000 + (Bank 1 capacity + Bank 0 capacity + 1 byte)

Bank 1 (CS1#): 0x0000 0000 + (Bank 1 capacity + Bank 0 capacity) to 0x0000 0000 + (Bank 0 capacity + 1 byte)

Bank 0 (CS0#): 0x0000 0000 + (Bank 0 capacity) to 0x0000 0000

Physical addresses in the DRAM address space are indicated below, when either 32 MB or 8 MB quantity is set.

For details of ADD(24:10) pin connection, refer to **16.2.6 DRAM connection**.

Table 3-8. DRAM Address Example (When Using 16-Bit Data Bus)

Physical Address	When Each Bank Is Set to 8 MB (SIZE3 = SIZE2 = SIZE1 = SIZE0 = 2)	When Each Bank Is Set to 32 MB (SIZE3 = SIZE2 = SIZE1 = SIZE0 = 4)
0x07FF FFFF to 0x0600 0000	RFU	Bank 3 (CS3#/ROMCS3#)
0x05FF FFFF to 0x0400 0000		Bank 2 (CS2#/ROMCS2#)
0x03FF FFFF to 0x0200 0000		Bank 1 (CS1#)
0x01FF FFFF to 0x0180 0000	Bank 3 (CS3#/ROMCS3#)	Bank 0 (CS0#)
0x017F FFFF to 0x0100 0000	Bank 2 (CS2#/ROMCS2#)	
0x00FF FFFF to 0x0080 0000	Bank 1 (CS1#)	
0x007F FFFF to 0x0000 0000	Bank 0 (CS0#)	

Caution When setting the quantity of each SDRAM space bank, be sure to set either a larger quantity to the lower banks or the same quantity to all banks.
SIZE0 (2:0) ≥ SIZE1 (2:0) ≥ SIZE2 (2:0) ≥ SIZE3 (2:0)

(b) In 32-bit bus mode (DBUS32 = 1)

Bank 3 (CS3#/ROMCS3#): 0x0000 0000 + (Bank 3 capacity + Bank 2 capacity + Bank 1 capacity + Bank 0 capacity) to 0x0000 0000 + (Bank 3 capacity + Bank 1 capacity + Bank 0 capacity + 1 byte)

Bank 2 (CS2#/ROMCS2#): 0x0000 0000 + (Bank 2 capacity + Bank 1 capacity + Bank 0 capacity) to 0x0000 0000 + (Bank 1 capacity + Bank 0 capacity + 1 byte)

Bank 1 (CS1#): 0x0000 0000 + (Bank 1 capacity + Bank 0 capacity) to 0x0000 0000 + (Bank 0 capacity + 1 byte)

Bank 0 (CS0#): 0x0000 0000 + (Bank 0 capacity) to 0x0000 0000

Physical addresses in the DRAM address space are indicated in the following table when any of the 8 MB, 16 MB, 32 MB, or 64 MB quantity is set.

For details of ADD(24:10) pin connection, refer to **16.2.6 DRAM connection**.

Table 3-9. DRAM Address Example When Using 16 Mb Expansion DRAM (When Using 32-Bit Data Bus)

Physical Address	When Each Bank Is Set as: Bank 3 = 8 MB, Bank 2 = 16 MB, Bank 1 = 32 MB, Bank 0 = 64 MB, (SIZE3 = 2, SIZE2 = 3, SIZE1 = 4, SIZE0 = 5)	When Each Bank Is Set to 64 MB (SIZE1 = SIZE0 = 5)
0x07FF FFFF to 0x0780 0000	RFU	Bank 1 (CS1#)
0x077F FFFF to 0x0700 0000	Bank 3 (CS3#/ROMCS3#)[Note]	
0x06FF FFFF to 0x0600 0000	Bank 2 (CS2#/ROMCS2#)[Note]	
0x05FF FFFF to 0x0400 0000	Bank 1 (CS1#)	
0x03FF FFFF to 0x0000 0000	Bank 0 (CS0#)	Bank 0 (CS0#)

Note Can be used exclusively from the expansion ROM.

CHAPTER 4 EXCEPTION PROCESSING

This chapter describes the types of exceptions and the exception processing flow. For the hardware used for exception processing, refer to **CHAPTER 5 CP0 REGISTERS**. For the details of each exception, refer to the separate **VR4100 Series Architecture User's Manual**.

4.1 Overview of Exceptions

4.1.1 Exception operation

If both the EXL and ERL bits of the Status register are 0, either the user mode, supervisor mode, or kernel mode is selected according to the value of the KSU bit of the Status register. If either the EXL bit or ERL bit is 1, the processor enters the kernel mode.

When an exception occurs in the processor, the EXL bit is set to 1 and the system enters the kernel mode. After the data has been saved, the exception handler normally resets the EXL bit to 0. The exception handler also resets the EXL bit to 1 so as not to lose the data due to another exception while recovering the saved data.

Once the exception processing has been completed, the EXL bit is reset to 0. For details, refer to the description of the ERET instruction in the separate **VR4100 Series Architecture User's Manual**.

4.1.2 Exception vector address

When an exception occurs, the exception is branched to the exception vector addresses (value consisting of base address plus vector offset) listed in Tables 4-1 and 4-2. Two sets of vector addresses are provided, one for the 32-bit mode and one for the 64-bit mode. Which vector is to be used is judged based on whether the user, supervisor, and kernel mode uses the 32-bit space or 64-bit space, which depends on the UX, SX, and KX bits of the Status register.

Table 4-1. 64-Bit Mode Exception Vector Base Addresses

	Vector Base Address (Virtual)	Vector Offset
Cold reset Soft reset NMI	0xFFFF FFFF BFC0 0000 (BEV bit is automatically set to 1)	0x0000
TLB refill (EXL = 0)	0xFFFF FFFF 8000 0000 (BEV = 0) 0xFFFF FFFF BFC0 0200 (BEV = 1)	0x0000
XTLB refill (EXL = 0)		0x0080
Other exceptions		0x0180

Table 4-2. 32-Bit Mode Exception Vector Base Addresses

	Vector Base Address (Virtual)	Vector Offset
Cold reset Soft reset NMI	0xBFC0 0000 (BEV bit is automatically set to 1)	0x0000
TLB refill (EXL = 0)	0x8000 0000 (BEV = 0) 0xBFC0 0200 (BEV = 1)	0x0000
XTLB refill (EXL = 0)		0x0080
Other exceptions		0x0180

4.1.3 Priority of exceptions

While more than one exception can occur for a single instruction, only the exception with the highest priority is reported. Table 4-3 lists the priorities.

Table 4-3. Exception Priority Order

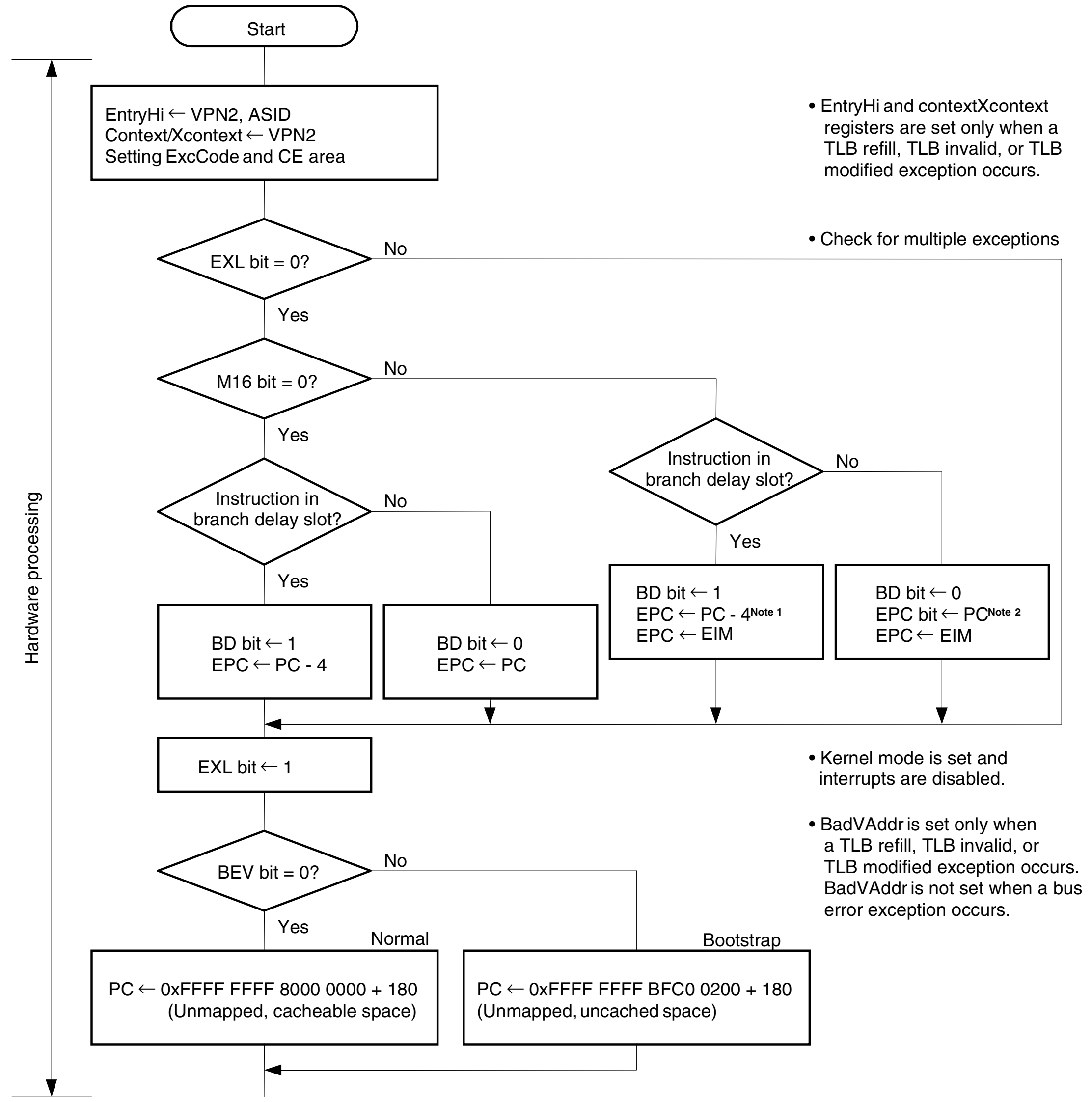
High	Cold reset
▲	Soft reset
	NMI
	Address error (instruction fetch)
	TLB/XTLB refill (instruction fetch)
	TLB invalid (instruction fetch)
	Bus error (instruction fetch)
	System call
	Breakpoint
	Coprocessor unusable
	Reserved instruction
	Trap
	Integer overflow
	Address error (data access)
	TLB/XTLB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

4.2 Exception Processing and Servicing Flowcharts

The remainder of this chapter contains flowcharts for the following exception processing and guidelines for their handlers:

- Common exceptions and guidelines for their exception handlers
- TLB/XTLB refill exceptions and guidelines for exception handlers
- Cold reset, Soft reset and NMI exceptions, and guidelines to their handlers.

Figure 4-1. Common Exception Processing (1/2)

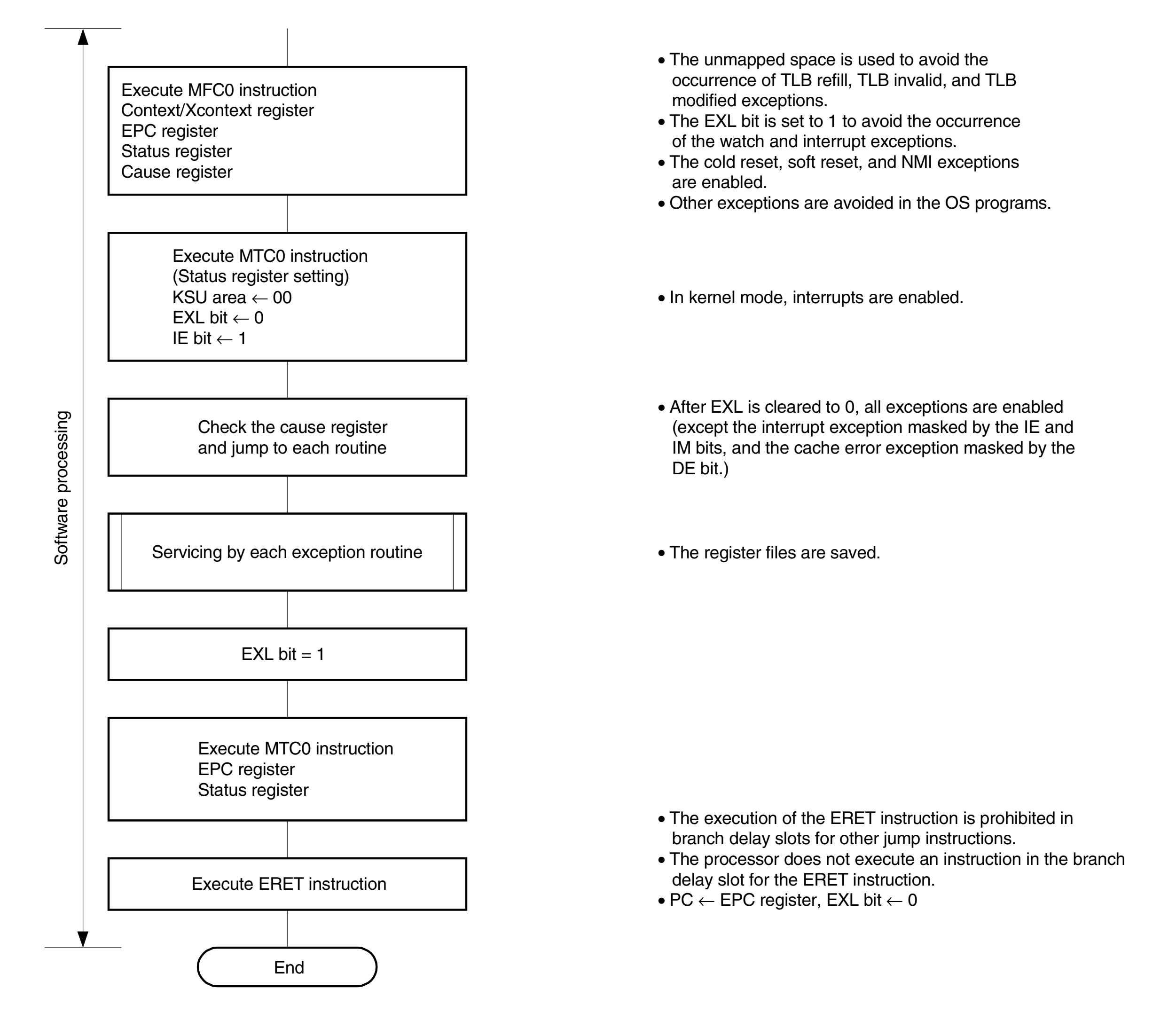

Start
EntryHi ← VPN2, ASID
Context/Xcontext ← VPN2
Setting ExcCode and CE area
• EntryHi and contextXcontext registers are set only when a TLB refill, TLB invalid, or TLB modified exception occurs.
EXL bit = 0?
No
Yes
• Check for multiple exceptions
M16 bit = 0?
No
Yes
Instruction in branch delay slot?
No
Yes
Instruction in branch delay slot?
No
Yes
Hardware processing
BD bit ← 1
EPC ← PC - 4
BD bit ← 0
EPC ← PC
BD bit ← 1
EPC ← PC - 4Note 1
EPC ← EIM
BD bit ← 0
EPC bit ← PCNote 2
EPC ← EIM
EXL bit ← 1
• Kernel mode is set and interrupts are disabled.
• BadVAddr is set only when a TLB refill, TLB invalid, or TLB modified exception occurs. BadVAddr is not set when a bus error exception occurs.
BEV bit = 0?
No
Yes
Normal
Bootstrap
PC ← 0xFFFF FFFF 8000 0000 + 180
(Unmapped, cacheable space)
PC ← 0xFFFF FFFF BFC0 0200 + 180
(Unmapped, uncached space)

Notes 1. PC – 2 when the JR or JALR instruction of the MIPS16 instructions is applied.

2. PC – 2 when the Extend instruction of the MIPS16 instructions is applied.

Remark Interrupts can be masked by setting the IE or IM bit.
The watch exception can be set to pending by setting the EXL bit to 1.

Figure 4-1. Common Exception Processing (2/2)



- The unmapped space is used to avoid the occurrence of TLB refill, TLB invalid, and TLB modified exceptions.
- The EXL bit is set to 1 to avoid the occurrence of the watch and interrupt exceptions.
- The cold reset, soft reset, and NMI exceptions are enabled.
- Other exceptions are avoided in the OS programs.

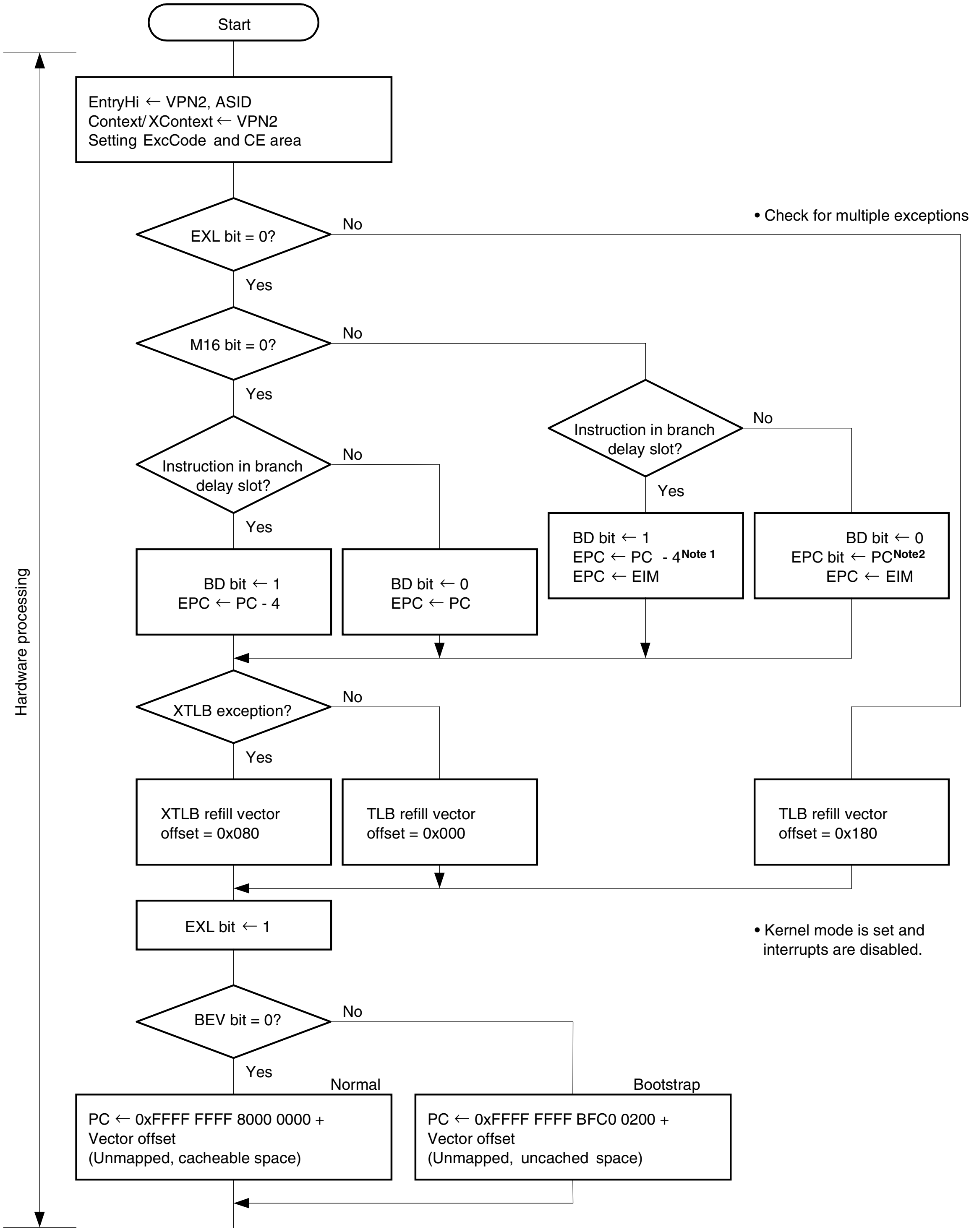
- In kernel mode, interrupts are enabled.

- After EXL is cleared to 0, all exceptions are enabled (except the interrupt exception masked by the IE and IM bits, and the cache error exception masked by the DE bit.)

- The register files are saved.

- The execution of the ERET instruction is prohibited in branch delay slots for other jump instructions.
- The processor does not execute an instruction in the branch delay slot for the ERET instruction.
- PC ← EPC register, EXL bit ← 0

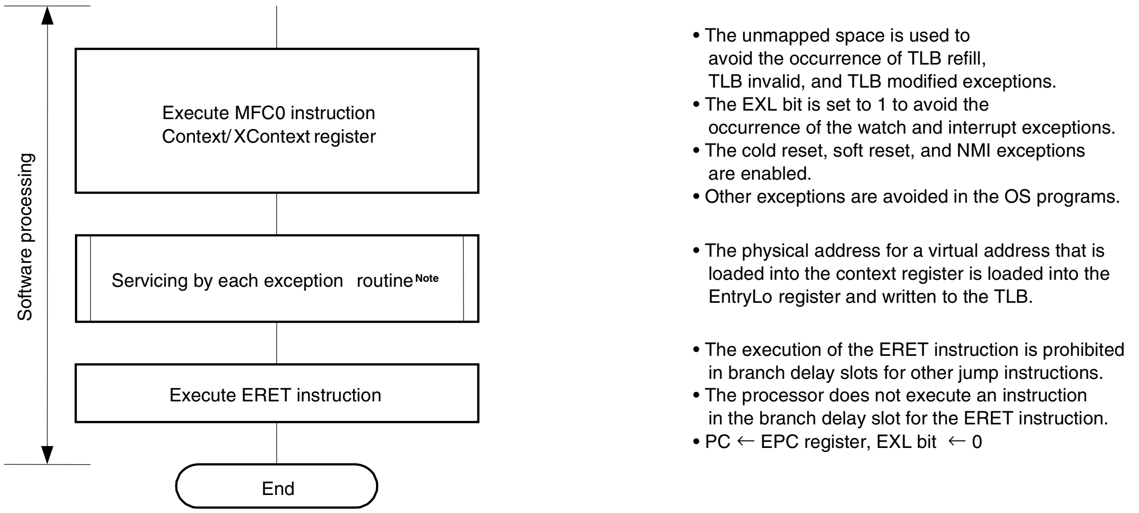
Figure 4-2. TLB/XTLB Refill Exception Processing (1/2)



Notes 1. PC – 2 when the JR or JALR instruction of the MIPS16 instructions is applied.

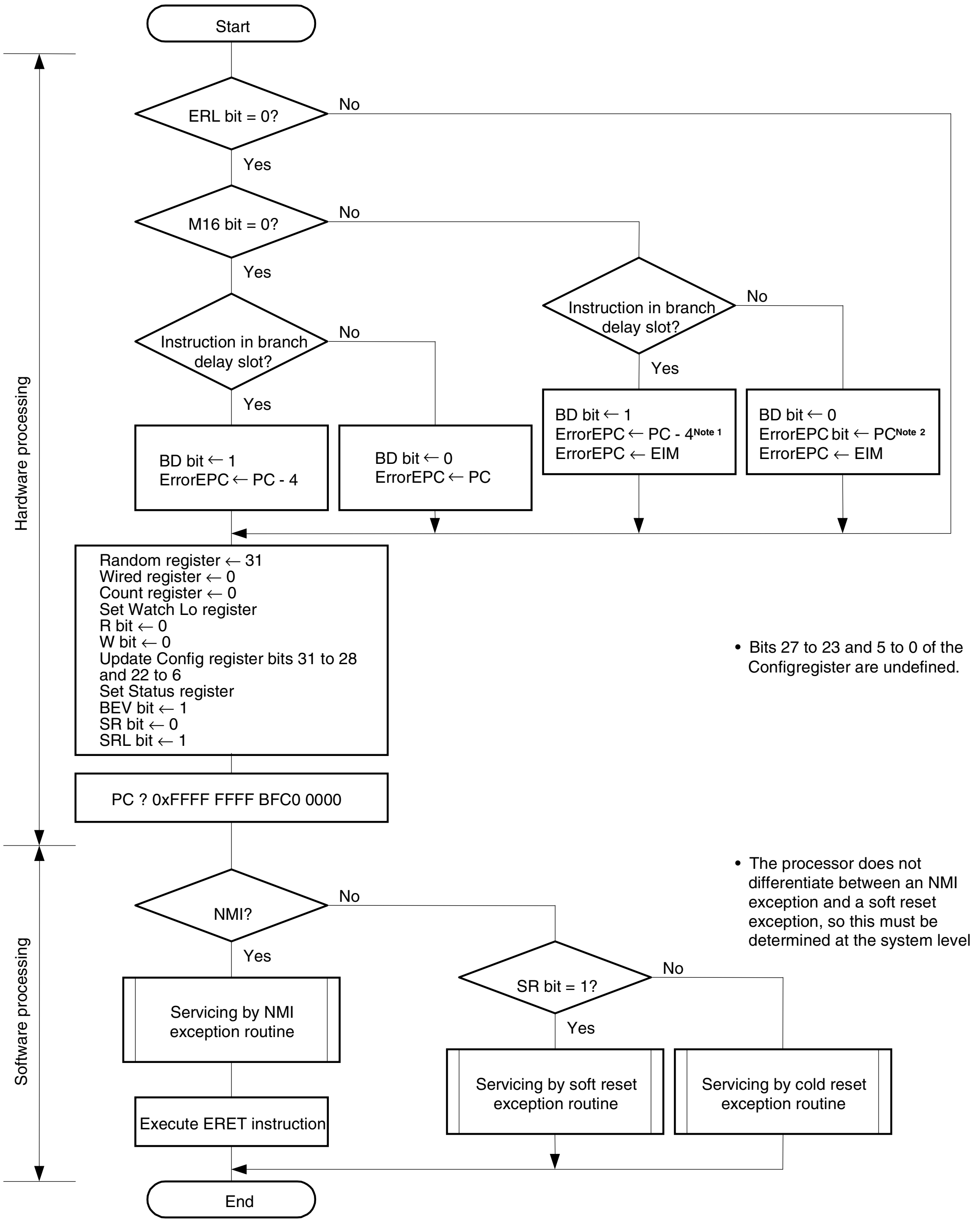
2. PC – 2 when the Extend instruction of the MIPS16 instructions is applied.

Figure 4-2. TLB/XTLB Refill Exception Processing (2/2)


Software processing
Execute MFC0 instruction
Context/ XContext register
Servicing by each exception routine[Note]
Execute ERET instruction
End
• The unmapped space is used to avoid the occurrence of TLB refill, TLB invalid, and TLB modified exceptions.
• The EXL bit is set to 1 to avoid the occurrence of the watch and interrupt exceptions.
• The cold reset, soft reset, and NMI exceptions are enabled.
• Other exceptions are avoided in the OS programs.
• The physical address for a virtual address that is loaded into the context register is loaded into the EntryLo register and written to the TLB.
• The execution of the ERET instruction is prohibited in branch delay slots for other jump instructions.
• The processor does not execute an instruction in the branch delay slot for the ERET instruction.
• PC ← EPC register, EXL bit ← 0

Note As long as data and instruction addresses exist in the mapping space, another TLB refill exception may occur. In such a case, EXL = 1 is set, causing a jump to the common exception vector. In this case, the common exception handler handles the TLB miss, the ERET instruction returns control to the user program, then a TLB refill exception is generated again.

Figure 4-3. Cold Reset Exception Handling



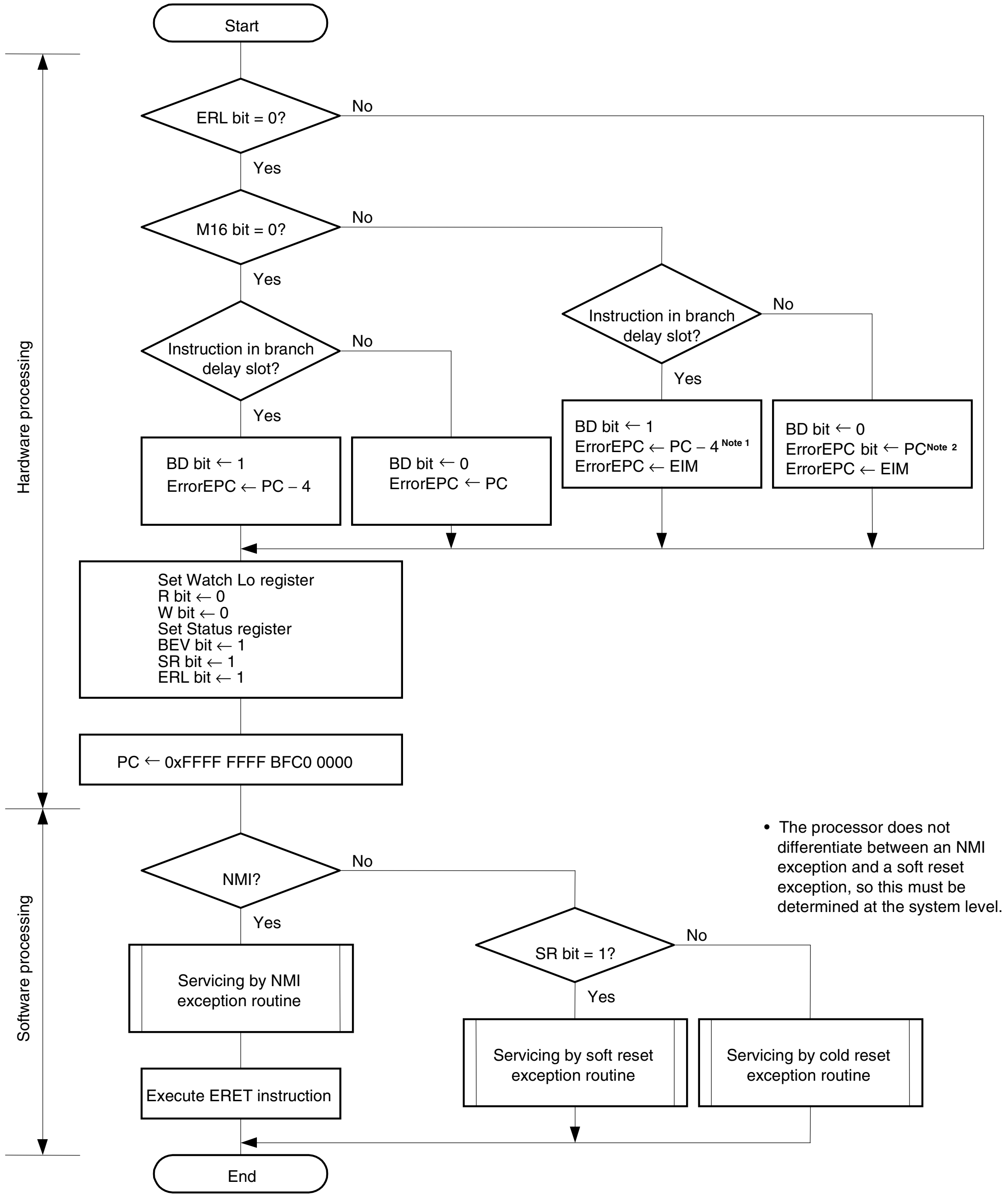
- Bits 27 to 23 and 5 to 0 of the Configregister are undefined.

- The processor does not differentiate between an NMI exception and a soft reset exception, so this must be determined at the system level

Notes 1. PC – 2 when the JR or JALR instruction of the MIPS16 instructions is applied.

2. PC – 2 when the Extend instruction of the MIPS16 instructions is applied.

Figure 4-4. Soft Reset and NMI Exception Handling



- The processor does not differentiate between an NMI exception and a soft reset exception, so this must be determined at the system level.

Notes 1. PC – 2 when the JR or JALR instruction of the MIPS16 instructions is applied.

2. PC – 2 when the Extend instruction of the MIPS16 instructions is applied.

CHAPTER 5 CP0 REGISTERS

The CP0 registers support memory management, address translation, exception processing, and privileged operations. This chapter describes the CP0 registers in detail.

5.1 Details of CP0 Registers

Table 5-1 lists the CP0 registers.

Table 5-1. CP0 Registers

Register No.	Register Name	Description	Category
0	Index	Programmable pointer to TLB array	Memory management
1	Random	Pseudo-random pointer to TLB array (read-only)	Memory management
2	EntryLo0	Lower side of TLB entry for even PFN	Memory management
3	EntryLo1	Lower side of TLB entry for odd PFN	Memory management
4	Context	Pointer to kernel virtual PTE in 32-bit mode	Exception processing
5	PageMask	Specification of page size	Memory management
6	Wired	Number of wired TLB entries	Memory management
7	–	RFU	–
8	BadVAddr	Display of virtual address where most recent error occurred	Exception processing
9	Count	Timer count	Exception processing
10	EntryHi	Higher side of TLB entry (including ASID)	Memory management
11	Compare	Timer comparison value	Exception processing
12	Status	Operation status setting	Exception processing
13	Cause	Display of cause of last exception	Exception processing
14	EPC	Exception program counter	Exception processing
15	PRId	Processor revision ID	Memory management
16	Config	Memory system mode setting	Memory management
17	LLAddr	RFU	Memory management
18	WatchLo	Memory reference trap address low bits	Exception processing
19	WatchHi	Memory reference trap address high bits	Exception processing
20	XContext	Pointer to kernel virtual PTE in 64-bit mode	Exception processing
21-25	–	RFU	–
26	Parity Error[Note]	Cache parity bits	Exception processing
27	Cache Error[Note]	Cache Error and Status register	Exception processing
28	TagLo	Cache Tag register (low)	Memory management
29	TagHi	Cache Tag register (high)	Memory management
30	ErrorEPC	Error exception program counter	Exception processing
31	–	RFU	–

Note This register is defined to maintain compatibility with the VR4100. This register is not used in the VR4131 hardware.

The registers are described in detail below. The number in parentheses indicates the register number.

5.1.1 Index register (0)

The index register is a 32-bit, read/write register containing 5 bits to index an entry in the TLB. The most-significant bit of the register shows the success or failure of a TLB probe (TLBP) instruction.

The index register shows a TLB entry that is a target of the TLB read (TLBR) or TLB write index (TLBWI) instruction.

The contents of the index register become undefined at reset, therefore initialize this register by software.

Figure 5-1. Index Register

31	30 5	4 0
P	0	Index
1	26	5

P: Indicates whether probing is successful or not. It is set to 1 if the latest TLBP instruction fails. It is cleared to 0 when the TLBP instruction is successful.

Index: Specifies an index to a TLB entry that is a target of the TLBR or TLBWI instruction.

0: RFU. Write 0 in a write operation. When this field is read, 0 is returned.

5.1.2 Random register (1)

The random register is a read-only register. The lower 5 bits are used in referencing a TLB entry. This register is decremented each time an instruction is executed. The values that can be set in the register are as follows:

- The lower limit is the content of the wired register.
- The upper limit is 31.

The random register specifies the entry in the TLB that is affected by the TLBWR instruction. The register can be read to verify proper operation of the processor.

The random register is set to the value of the upper limit upon cold reset. This register is also set to the upper limit when the wired register is written. Figure 5-2 shows the format of the random register.

Figure 5-2. Random Register

31 5	4 0
0	Random
27	5

Random: TLB random index

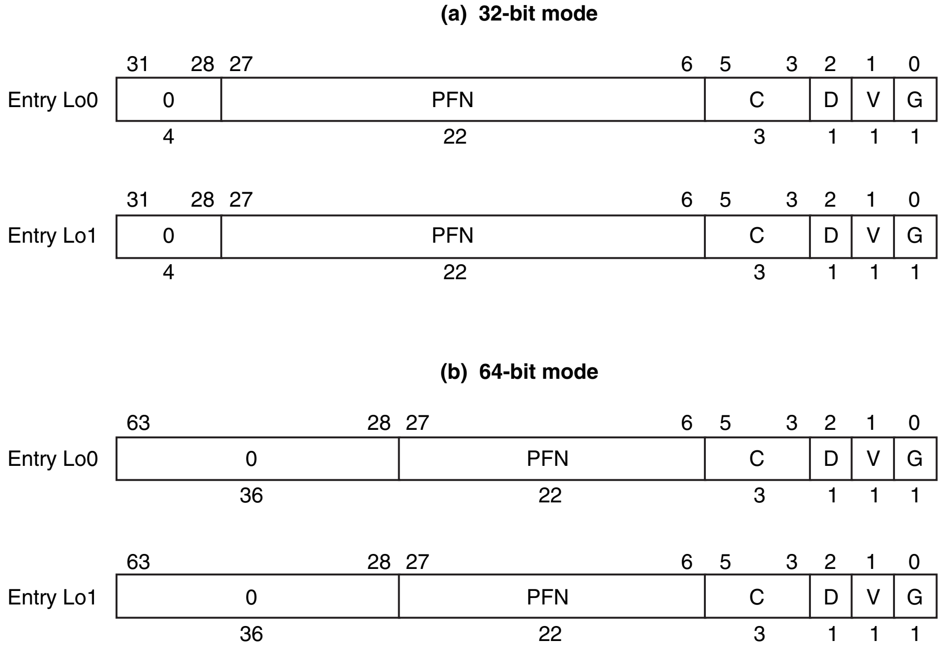
0: RFU. Write 0 in a write operation. When this field is read, 0 is returned.

5.1.3 EntryLo0 (2) and EntryLo1 (3) registers

The EntryLo register consists of two registers that have identical formats: EntryLo0, used for even virtual pages and EntryLo1, used for odd virtual pages. The EntryLo0 and EntryLo1 registers are both read-/write-accessible. They are used to access the lower bits of the on-chip TLB. When a TLB read/write operation is carried out, the EntryLo0 and EntryLo1 registers hold the contents of the lower 32 bits of TLB entries at even and odd addresses, respectively.

The contents of the EntryLo0 and EntryLo1 registers become undefined at reset, therefore initialize these registers by software.

Figure 5-3. EntryLo0 and EntryLo1 Registers



(a) 32-bit mode

(b) 64-bit mode

PFN: Page frame number; higher bits of the physical address.

C: Specifies the TLB page attribute (refer to Table 5-2).

D: Dirty. If this bit is set to 1, the page is marked as dirty and, therefore, writable. This bit is actually a write-protect bit that software can use to prevent alteration of data.

V: Valid. If this bit is set to 1, it indicates that the TLB entry is valid; if an entry with this bit 0 is hit, a TLB Invalid exception (TLBL or TLBS) occurs.

G: Global. If this bit is set in both entryLo0 and entryLo1, then the processor ignores the ASID during TLB lookup.

0: RFU. Write 0 in a write operation. When this field is read, 0 is returned.

The coherency attribute (C) bits are used to specify whether to use the cache in referencing a page. When the cache is used, whether the page attribute is “cached” or “uncached” is selected by algorithm.

Table 5-2 lists the page attributes selected according to the value in the C bits.

Table 5-2. Cache Algorithm

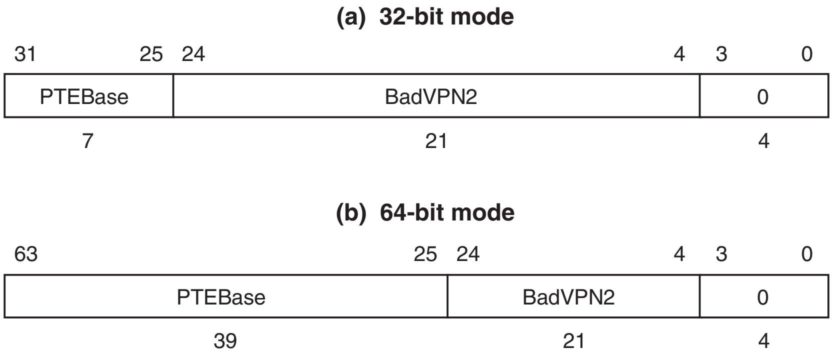
C Bit Value	Cache Algorithm
0	Cached
1	Cached
2	Uncached
3	Cached
4	Cached
5	Cached
6	Cached
7	Cached

5.1.4 Context register (4)

The Context register is a read/write register containing the pointer to an entry in the page table entry (PTE) array on the memory; this array is a table that stores virtual-to-physical address translations. When there is a TLB miss, the operating system loads the unsuccessfully translated entry from the PTE array to the TLB. The Context register is used by the TLB refill exception handler for loading TLB entries.

The Context register duplicates some of the information provided in the BadVAddr register, but the information is arranged in a form that is more useful for a software TLB exception handler. Figure 5-4 shows the format of the context register.

Figure 5-4. Context Register Format



PTEBase: The PTEBase field is a base address of the PTE entry table.

BadVPN2: This field holds the value (VPN2) obtained by halving the virtual page number of the most recent virtual address for which translation failed.

0: RFU. Write 0 in a write operation. When this field is read, 0 is read.

The PTEBase field is used by software as the pointer to the base address of the PTE table in the current user address space.

The 21-bit BadVPN2 field contains bits 31 to 11 of the virtual address that caused the TLB miss; bit 10 is excluded because a single TLB entry maps to an even-odd page pair. For a 1 KB page size, this format can directly address the pair-table of 8-byte PTEs. When the page size is 4 KB or more, shifting or masking this value produces the correct PTE reference address.

5.1.5 PageMask register (5)

The PageMask register is a read/write register used for reading from or writing to the TLB; it holds a comparison mask that sets the page size for each TLB entry, as shown in Table 5-3. Page sizes must be from 1 KB to 256 KB.

TLB read and write instructions use this register as either a source or a destination; bits 18 to 11 that are targets of comparison are masked during address translation.

The contents of the PageMask register become undefined at reset, therefore initialize this register by software.

Figure 5-5. PageMask Register

31 19	18 11	10 0
0	MASK	0
13	8	11

MASK: Page comparison mask, which determines the virtual page size for the corresponding entry.

0: RFU. Write 0 in a write operation. When this field is read, 0 is returned.

Table 5-3 lists the mask pattern for each page size. If the mask pattern is one not listed below, the TLB behaves unexpectedly.

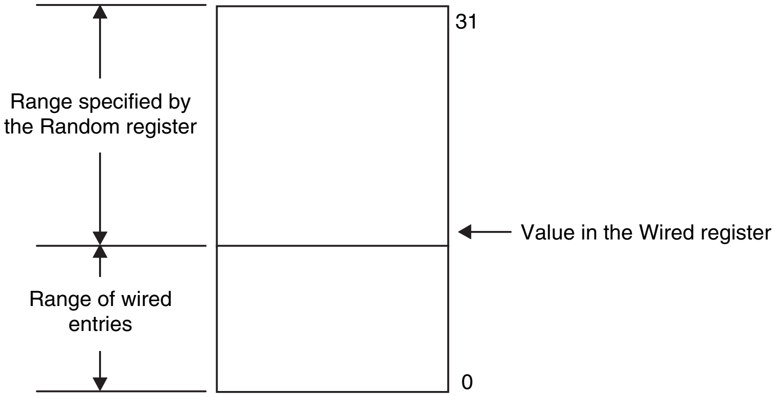
Table 5-3. Mask Values and Page Sizes

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

5.1.6 Wired register (6)

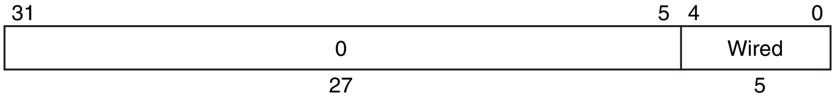
The Wired register is a read/write register that specifies the lower boundary of the random entry of the TLB as shown in Figure 5-6. Wired entries cannot be overwritten by a TLBWR instruction. They can, however, be overwritten by a TLBWI instruction. Random entries can be overwritten by both instructions.

Figure 5-6. Positions Indicated by Wired Register


31
Range specified by the Random register
Value in the Wired register
Range of wired entries
0

The Wired register is set to 0 upon cold reset. Writing this register also sets the Random register to the value of its upper limit (see **5.1.2 Random register (1)**). Figure 5-7 shows the format of the wired register.

Figure 5-7. Wired Register



Wired: TLB wired boundary

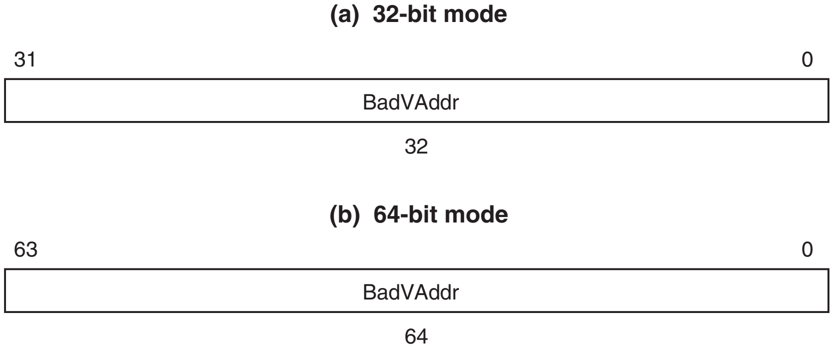
0: RFU. Write 0. When this field is read, 0 is returned.

5.1.7 Bad Virtual Address (BadVAddr) register (8)

The Bad Virtual Address (BadVAddr) register is a read-only register that saves the most recent virtual address that failed to have a valid translation, or that had an addressing error. Figure 5-8 shows the format of the BadVAddr register.

Caution This register saves no information after a bus error exception, because it is not an address error exception.

Figure 5-8. BadVAddr Register Format



(a) 32-bit mode

(b) 64-bit mode

BadVAddr: Most recent virtual address for which an addressing error occurred, or for which address translation failed.

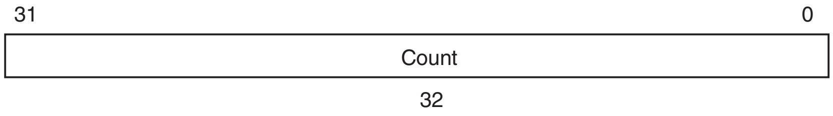
5.1.8 Count register (9)

The Count register can be read and written and acts as a timer. It is incremented in synchronization with the frequencies of the MasterOut clock (refer to **1.8 Clock Interface**), regardless of whether instructions are being executed, retired, or any forward progress is actually made through the pipeline.

This register is a free-running type. When the register reaches all ones, it rolls over to zero and continues counting. This register is used for a self-diagnostic test, system initialization, or the establishment of inter-process synchronization.

Figure 5-9 shows the format of the count register.

Figure 5-9. Count Register Format



Count: Up-to-date count value that is compared with the value of the Compare register.

5.1.9 EntryHi register (10)

The EntryHi register is write-accessible. It is used to access the higher bits in the on-chip TLB. The EntryHi register holds the higher bits of a TLB entry for TLB read and write operations. If a TLB refill, TLB invalid, or TLB modified exception occurs, the EntryHi register holds the higher bits of the TLB entry. The EntryHi register is also set with the virtual page number (VPN2) for a virtual address where an exception occurred and the ASID. See the separate **VR4100 Series Architecture User's Manual** for details of TLB exceptions.

The ASID is used to read from or write to the ASID field of the TLB entry. It is also checked with the ASID of the TLB entry as the ASID of the virtual address during address translation.

The EntryHi register is accessed by the TLBP, TLBWR, TLBWI, and TLBR instructions.

The contents of the EntryHi register become undefined at reset, therefore initialize this register by software.

Figure 5-10. EntryHi Register

(a) 32-bit mode

31 11	10 8	7 0
VPN2	0	ASID
21	3	8

(b) 64-bit mode

63 62	61 40	39 11	10 8	7 0
R	Fill	VPN2	0	ASID
2	22	29	3	8

VPN2: Virtual page number divided by two (mapping to two pages)

ASID: Address space ID. An 8-bit ASID field that lets multiple processes share the TLB; each process has a distinct mapping of otherwise identical virtual page numbers.

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

Fill: RFU. Ignored on write. When read, 0 is returned.

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

5.1.10 Compare register (11)

The Compare register causes a timer interrupt; it maintains a stable value that does not change on its own.

When the value of the Count register (refer to **5.1.8 Count register (9)**) equals the value of the Compare register, the IP7 bit in the cause register is set. This causes an interrupt as soon as the interrupt is enabled.

Writing a value to the Compare register, as a side effect, clears the timer interrupt request.

For diagnostic purposes, the Compare register is a read/write register. Normally, this register should be only used for a write. Figure 5-11 shows the format of the Compare register.

Figure 5-11. Compare Register Format

31 0
Compare
32

Compare: Value that is compared with the count value of the Count register.

5.1.11 Status register (12)

The Status register is a read/write register that contains the operating mode, interrupt enabling, and the diagnostic states of the processor. Figure 5-12 shows the format of the status register.

Figure 5-12. Status Register Format (1/2)

31	30 29	28	27 26	25	24 16	15 8	7	6	5	4 3	2	1	0
XX	0	CU0	0	RE	DS	IM	KX	SX	UX	KSU	ERL	EXL	IE
1	2	1	2	1	9	8	1	1	1	2	1	1	1

XX: Enables/disables the use of expanded instructions (MIPSIV, MIPS32, MIPS64)
1 → Enabled
0 → Disabled

CU0: Enables/disables the use of the coprocessor (1 → Enabled, 0 → Disabled).
CP0 can be used by the kernel at all times.

RE: Enables/disables reversing of the endian setting in user mode (0 → Disabled, 1 → Enabled).

Caution This bit must be set to 0 for the VR4131.

DS: Diagnostic status field (refer to **Figure 5-13**).

IM: Interrupt mask field used to enable/disable external/internal and software interrupts (0 → Disabled, 1 → Enabled). This field consists of 8 bits that are used to control eight interrupts. The bits are assigned to interrupts as follows:

IM7: Masks a timer interrupt.
IM(6:2): Mask ordinary interrupts (Int(4:0)[Note]). However, Int(4:3)[Note] never occur in the VR4131.
IM(1:0): Mask software interrupts.

Note Int(4:0) are internal signals of the CPU core. For details about connection to the on-chip peripheral units of the VR4131, refer to **CHAPTER 11 ICU (INTERRUPT CONTROL UNIT)**.

Figure 5-12. Status Register Format (2/2)

KX: Enables 64-bit addressing in kernel mode (0 → 32-bit, 1 → 64-bit). If this bit is set, an XTLB refill exception occurs if a TLB miss occurs in the kernel mode address space.
In addition, 64-bit operations are always valid in kernel mode.

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

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

KSU: Sets and indicates the operating mode (10 → User, 01 → Supervisor, 00 → Kernel).

ERL: Sets and indicates the error level (0 → Normal, 1 → Error).

EXL: Sets and indicates the exception level (0 → Normal, 1 → Exception).

IE: Sets and indicates interrupt enabling/disabling (0 → Disabled, 1 → Enabled).

0: RFU. Write 0 in a write operation. When this bit is read, 0 is read.

Figure 5-13 shows the details of the diagnostic status (DS) field. All DS field bits are read/writable.

Figure 5-13. Status Register Diagnostic Status Field

24	23	22	21	20	19	18	17	16
0		BEV	0	SR	0	CH	CE	DE
2		1	1	1	1	1	1	1

BEV: Specifies the base address of a TLB refill exception vector and common exception vector (0 → Normal, 1 → Bootstrap).

SR: Causes a soft reset or NMI exception (0 → Did not occur, 1 → Occurred).

CH: CP0 condition bit (0 → False, 1 → True). This bit can be read and written by software only; it cannot be accessed by hardware.

CE, DE: These are prepared to maintain compatibility with the VR4100, and are not used in the VR4131 hardware.

0: RFU. Write 0 in a write operation. When this field is read, 0 is read.

The status register has the following fields where the modes and access status are set.

(1) Interrupt enable

Interrupts are enabled when all of the following conditions are true:

- IE is set to 1.
- EXL is cleared to 0.
- ERL is cleared to 0.
- The appropriate bit of the IM is set to 1.

(2) Operating modes

The following Status register bit settings are required for user, kernel, and supervisor modes.

- The processor is in user mode when KSU = 10, EXL = 0, and ERL = 0.
- The processor is in supervisor mode when KSU = 01, EXL = 0, and ERL = 0.
- The processor is in kernel mode when KSU = 00, EXL = 1, or ERL = 1.

(3) 32- and 64-bit modes

The following Status register bit settings select 32- or 64-bit operation for user, kernel, and supervisor operating modes. Enabling 64-bit operation permits the execution of 64-bit opcodes and translation of 64-bit addresses. 64-bit operation for user, kernel and supervisor modes can be set independently.

- 64-bit addressing for kernel mode is enabled when KX bit = 1. 64-bit operations are always valid in kernel mode.
- 64-bit addressing and operations are enabled for supervisor mode when SX bit = 1.
- 64-bit addressing and operations are enabled for user mode when UX bit = 1.

(4) Kernel address space accesses

Access to the kernel address space is allowed when the processor is in kernel mode.

(5) Supervisor address space accesses

Access to the supervisor address space is allowed when the processor is in supervisor or kernel mode.

(6) User address space accesses

Access to the user address space is allowed in any of the three operating modes.

(7) Status after reset

The contents of the status register are undefined after cold resets, except for the following bits in the diagnostic status field.

- SR is cleared to 0.
- ERL and BEV are set to 1.
- SR is 0 after cold reset, and is 1 after soft reset or NMI interrupt.

Remark Cold reset and soft reset are CPU core reset (see **6.3 Reset of CPU Core**). For the reset of all the VR4131 including peripheral units, refer to **CHAPTER 6 INITIALIZATION INTERFACE** and **CHAPTER 12 PMU (POWER MANAGEMENT UNIT)**.

5.1.12 Cause register (13)

The Cause register is a 32-bit read/write register that holds the cause of the most recent exception. A 5-bit ExcCode (exception code area) indicates one of the causes (refer to **Table 5-4**). Other bits hold the detailed information of the specific exception. All bits in the Cause register, with the exception of the IP1 and IP0 bits, are read-only; IP1 and IP0 are used for software interrupts. Figure 5-14 shows the fields of this register; Table 5-4 describes the Cause register codes.

Figure 5-14. Cause Register Format

31	30	29 28	27 16	15 8	7	6 2	1 0
BD	0	CE	0	IP(7:0)	0	ExcCode	0
1	1	2	12	8	1	5	2

BD: Indicates whether the most recent exception occurred in the branch delay slot (1 → In delay slot, 0 → Normal).

CE: Indicates the coprocessor number in which a coprocessor unusable exception occurred. This field will remain undefined for as long as no exception occurs.

IP: Indicates whether an interrupt is pending (1 → Interrupt pending, 0 → No interrupt pending).

IP7: A timer interrupt.

IP(6:2): Ordinary interrupts (Int(4:0)[Note]). However, Int(4:3)[Note] never occurs in the VR4131.

IP(1:0): Software interrupts. Only these bits cause an interrupt exception, when they are set to 1 by software.

Note Int(4:0) are internal signals of the CPU core. For details about connection to the on-chip peripheral units of the VR4131, refer to **CHAPTER 11 ICU (INTERRUPT CONTROL UNIT)**.

ExcCode: Exception code field (refer to **Table 5-4** for details).

0: RFU. Write 0 in a write operation. When this field is read, 0 is read.

Table 5-4. Cause Register Exception Code Field

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

The VR4131 has eight interrupt request sources, IP7 to IP0. For a detailed description of interrupts, refer to the separate **VR4100 Series Architecture User's Manual**.

(1) IP7

This bit indicates whether there is a timer interrupt request.
It is set when the values of Count register and Compare register match.

(2) IP6 to IP2

IP6 to IP2 reflect the state of the interrupt request signal of the CPU core.

(3) IP1 and IP0

These bits are used to set/clear a software interrupt request.

5.1.13 Exception Program Counter (EPC) register (14)

The Exception Program Counter (EPC) register is a read/write register that contains the address at which processing resumes after an exception has been processed. The contents of this register change depending on whether execution of MIPS16 instructions is enabled or disabled. Setting the MIPS16EN pin during RTC reset specifies whether execution of the MIPS16 instructions is enabled or disabled.

Tables 5-5 and 5-6 show the contents of the EPC register when MIPS16 instruction execution is disabled and enabled, respectively.

Table 5-5. When MIPS16 Instruction Execution Is Disabled

Instruction That Caused Exception	Contents of EPC Register
Branch delay slot of branch or jump instruction	Virtual address of immediately preceding branch or jump instruction
Other than above	Virtual address of the instruction that caused the exception

Table 5-6. When MIPS16 Instruction Execution Is Enabled

(a) When MIPS III Instruction Is Executed

Instruction That Caused Exception	Contents of EPC Register	EIM
Branch delay slot of branch or jump instruction	Virtual address of immediately preceding branch or jump instruction	0
Other than above	Virtual address of the instruction that caused the exception	0

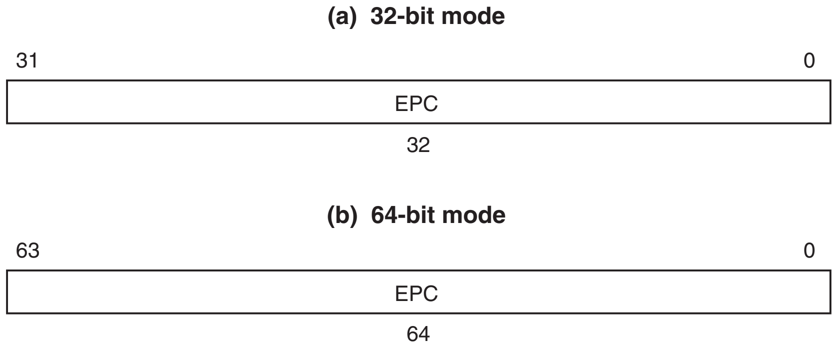
(b) When MIPS16 Instruction Is Executed

Instruction That Caused Exception	Contents of EPC Register	EIM
Branch delay slot of jump instruction or latter half of Extend instruction	Virtual address of immediately preceding jump or Extend instruction	1
Other than above	Virtual address of the instruction that caused the exception	1

The EXL bit in the status register is set to 1 to keep the processor from overwriting the address of the exception-causing instruction contained in the EPC register in the event of another exception.

Figure 5-15 shows the EPC register format when MIPS16ISA is disabled, and Figure 5-16 shows the EPC register format when MIPS16ISA is enabled.

Figure 5-15. EPC Register Format (When MIPS16ISA Is Disabled)

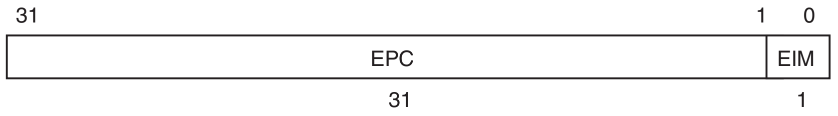

(a) 32-bit mode

(b) 64-bit mode

EPC: Address for a program to be restarted after exception processing.

Figure 5-16. EPC Register Format (When MIPS16ISA Is Enabled)

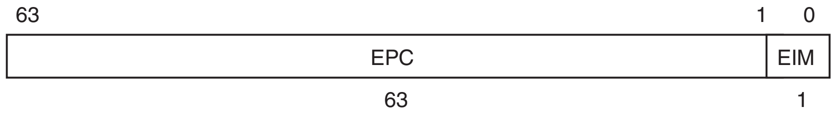
(a) 32-bit mode



EPC: Exceptional virtual address (31:1).

EIM: ISA mode at which an exception occurs.
(1: on MIPS16 instruction, 0: on MIPS III instruction.)

(b) 64-bit mode



EPC: Exceptional virtual address (63:1).

EIM: ISA mode at which an exception occurs.
(1: on MIPS16 instruction, 0: on MIPS III instruction.)

5.1.14 Processor Revision Identifier (PRId) register (15)

The 32-bit, read-only Processor Revision Identifier (PRId) register contains information identifying the implementation and revision level of the CPU and CP0. Figure 5-17 shows the format of the PRId register.

Figure 5-17. PRId Register

31 16	15 8	7 0
0	Imp	Rev
16	8	8

Imp: CPU core processor ID number, 0x0C for the VR4131.

Rev: CPU core processor revision number

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

The processor revision number is stored as a value in the form y.x, where y is a major revision number in bits 7 to 4 and x is a minor revision number in bits 3 to 0.

The processor revision number can distinguish some of the VR4131 CPU core revisions, however there is no guarantee that changes to the CPU core will necessarily be reflected in the PRId register, or that changes to the revision number necessarily reflect real CPU core changes. Therefore, create a program that does not depend on the processor revision number area.

5.1.15 Config register (16)

The Config register indicates/sets various statuses of the processors on the VR4131.

Some configuration options, as defined by the EC and BE fields, are set by the hardware during a cold reset and are included in the Config register as read-only status bits for the software to access. Other configuration options are read/write (AD, EP, and K0 fields) and controlled by software; on a cold reset these fields are undefined.

Since only a subset of the VR4000 Series™ options are available in the VR4131, some bits are set to constants (e.g., bits 14:13) that were variable in the VR4000 Series. The config register should be initialized by software before caches are used. Figure 5-18 shows the format of the Config register.

The contents of the Config register become undefined at reset, therefore initialize this register by software.

Figure 5-18. Config Register Format (1/2)

31	30 28	27 24	23	22 21	20	19 18	17	16	15	14 13	12	11 9	8 6	5	4	3	2 0
IS	EC	EP	AD	0	M16	0	1	BP	BE	10	CS	IC	DC	IB	DB	0	K0
1	3	4	1	2	1	2	1	1	1	2	1	3	3	1	1	1	3

IS: Sets the instruction streaming function. After reset, this bit is cleared to 0.
0 → ON
1 → OFF

Figure 5-18. Config Register Format (2/2)

EC: System interface clock (VTClock) frequency ratio (read only)
0 → RFU
1 → Processor clock frequency divided by 2
2 → RFU
3 → Processor clock frequency divided by 3
4 → Processor clock frequency divided by 4
5 → Processor clock frequency divided by 5
6 → Processor clock frequency divided by 6
7 → RFU

EP: Sets the transfer pattern of write-back data
0 → DD: 1 word/1 cycle
Others → RFU

AD: Sets accelerate data mode
0 → VR4000 Series compatible mode
1 → RFU

M16: Indicates MIPS16 ISA mode enable (read only)
0 → MIPS16 instructions cannot be executed
1 → MIPS16 instructions can be executed

BP: Sets the branch forecast. After reset, this bit is cleared to 0.
0 → ON
1 → OFF

BE: BigEndianMem (Indicates endian)
0 → Little endian
1 → RFU

CS: Indicates cache size mode (n = IC, DC). Fixed to 1 in the VR4131.
0 → RFU
1 → $2^{(n+10)}$ bytes

IC: Indicates the instruction cache size. In the VR4131, $2^{(IC+10)}$ bytes.
5 → 16 KB
Others → RFU

DC: Indicates the data cache size. In the VR4131, $2^{(DC+10)}$ bytes.
4 → 16 KB
Others → RFU

IB: Sets the refill size of the instruction cache
0 → 4 words
1 → 8 words

DB: Sets the data cache refill size.
0 → 4 words
1 → 8 words

K0: Sets kseg0 coherency algorithm
010 → Uncached
Others → Cached

1: 1 is returned when read.

0: 0 is returned when read.

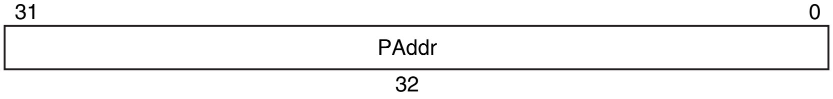
Caution Be sure to set the EP field and the AD bit to 0. If they are set to any other values, the processor may behave unexpectedly.

5.1.16 Load Linked Address (LLAddr) register (17)

The Load Linked Address (LLAddr) register is available for read/write and it is used only for diagnostic purposes. This register is defined in the VR4131, to be compatible with the VR4000™ and VR4400™ and serves no function during normal operation.

The contents of the LLADDr register become undefined at reset, therefore, initialize this register by software.

Figure 5-19. LLAddr Register



PAddr: 32-bit physical address

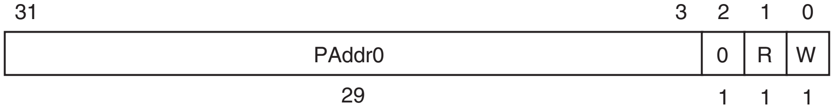
5.1.17 WatchLo (18) and WatchHi (19) registers

The VR4131 processor provides a debugging feature to detect references to a selected physical address; load and store instructions to the location specified by the WatchLo and WatchHi registers cause a watch exception.

Figures 5-20 and 5-21 show the format of the WatchLo and WatchHi registers.

The contents of these registers become undefined at reset, therefore initialize these registers by software.

Figure 5-20. WatchLo Register Format



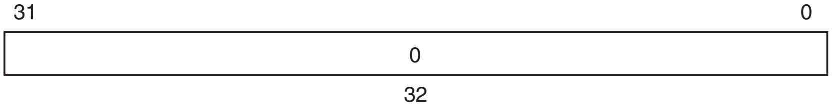
PAddr0: Specifies physical address bits 31 to 3.

R: Sets watch exception detection when a load instruction is executed.
0 → Detected (default)
1 → Not detected

W: Sets watch exception detection when a store instruction is executed.
0 → Detected (default)
1 → Not detected

0: RFU. Write 0 in a write operation. When this field is read, 0 is read.

Figure 5-21. WatchHi Register Format



0: RFU. Write 0 in a write operation. When this field is read, 0 is read.

5.1.18 XContext register (20)

The read/write XContext register contains a pointer to an entry in the page table entry (PTE) array, an operating system data structure that stores virtual-to-physical address translations. If a TLB miss occurs, the operating system loads the untranslated data from the PTE into the TLB to handle the error with software.

The XContext register is used by the XTLB refill exception handler to load TLB entries in 64-bit addressing mode.

The XContext register duplicates some of the information provided in the BadVAddr register, and puts it in a form useful for the XTLB exception handler.

This register is included solely for operating system use. The operating system sets the PTEBase field in the register, as needed.

Figure 5-22 shows the format of the XContext register.

Figure 5-22. XContext Register Format

63 35	34 33	32 4	3 0
PTEBase	R	BadVPN2	0
29	2	29	4

PTEBase: The PTEBase field is a base address of the page table entry.

R: Space type (00 → User, 01→ Supervisor, 11 → Kernel). The setting of this field matches virtual address bits 63 and 62.

BadVPN2: This field holds the value (VPN2) obtained by halving the virtual page number of the most recent virtual address for which translation failed.

0: RFU. Write 0 in a write operation. When this field is read, 0 is read.

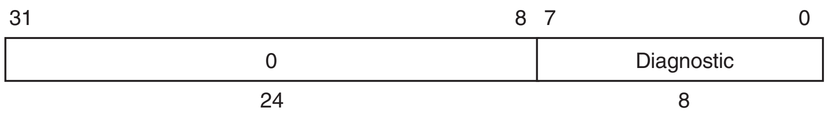
The 29-bit BadVPN2 field has bits 39 to 11 of the virtual address that caused the TLB refill; bit 10 is excluded because a single TLB entry maps to an even-odd page pair. For a 1 KB page size, this format may be used directly to address the pair-table of 8-byte PTEs. For 4 KB or more page and PTE sizes, shifting or masking this value produces the appropriate address.

5.1.19 Parity Error register (26)

The Parity Error (PErr) register is a readable/writable register. This register is defined to maintain software-compatibility with the VR4100, and is not used in hardware because the VR4131 has no parity.

Figure 5-23 shows the format of the PErr register.

Figure 5-23. PErr Register Format



Diagnostic: 8-bit self diagnostic field.

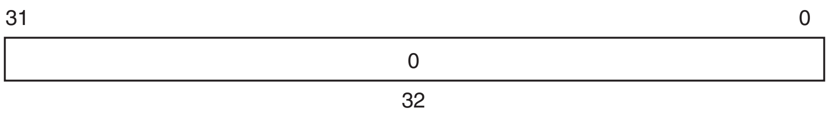
0: RFU. Write 0 in a write operation. When this field is read, 0 is read.

5.1.20 Cache Error (CacheErr) register (27)

The Cache Eerror (CacheErr) register is a readable/writable register. This register is defined to maintain software compatibility with the VR4100, and is not used in hardware because the VR4131 has no parity.

Figure 5-24 shows the format of the CacheErr register.

Figure 5-24. CacheErr Register Format



0: RFU. Write 0 in a write operation. When this field is read, 0 is read.

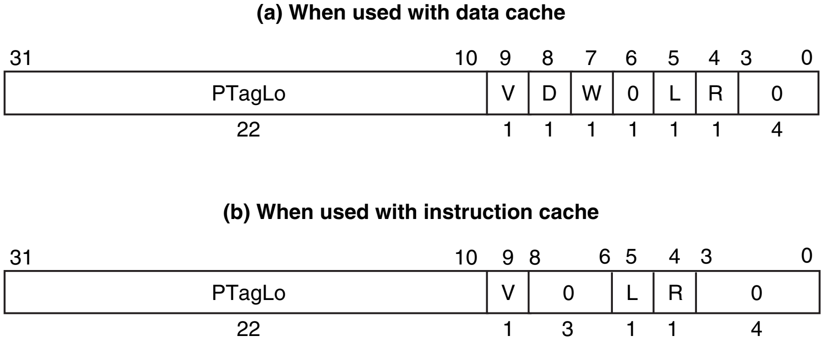
5.1.21 Cache Tag registers (TagLo (28) and TagHi (29))

The TagLo and TagHi registers are 32-bit read/write registers that hold the cache tag during cache initialization, cache diagnostics, or cache error processing. The tag registers are written by the CACHE and MTC0 instructions.

Figures 5-25 and 5-26 show the format of these registers.

The contents of these registers become undefined at reset, therefore, initialize these registers by software.

Figure 5-25. TagLo Register


(a) When used with data cache
31
10
9
8
7
6
5
4
3
0
PTagLo
V
D
W
0
L
R
0
22
1
1
1
1
1
1
4
(b) When used with instruction cache
31
10
9
8
6
5
4
3
0
PTagLo
V
0
L
R
0
22
1
3
1
1
4

PTagLo: Specifies physical address bits 31 to 10.

V: Valid bit

D: Dirty bit. However, this bit is defined to be compatible with the VR4000 Series processors, and does not indicate the status of cache memory in spite of its readability and writability. This bit cannot change the status of cache memory. The valve written to this bit is ignored, and the read value is the same as the V bit value.

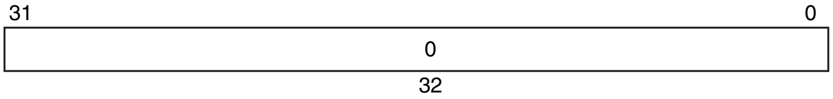
W: Write-back bit (set if cache line has been updated)

L: LOCK bit. When the LOCK bit is set, the cache will not be refilled even if a cache miss occurs.

R: LRU bit

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

Figure 5-26. TagHi Register



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

5.1.22 Error Exception Program Counter ErrorEPC register (30)

The Error Exception Program Counter (ErrorEPC) register is similar to the EPC register. It is used to store the program counter value at which the cold reset, soft reset, or NMI exception has been serviced.

The read/write ErrorEPC register contains the virtual address at which instruction processing can resume after servicing an error. The contents of this register change depending on whether execution of MIPS16 instructions is enabled or disabled. Setting the MIPS16EN pin during RTC reset specifies whether the execution of MIPS16 instructions is enabled or disabled.

Tables 5-7 and 5-8 show the contents of the ErrorEPC register when MIPS16 instruction execution is disabled and enabled, respectively.

Table 5-7. When MIPS16 Instruction Execution Is Disabled

Instruction That Caused Exception	Contents of ErrorEPC Register
Branch delay slot of branch or jump instruction	Virtual address of immediately preceding branch or jump instruction
Other than above	Virtual address of the instruction that caused the exception

Table 5-8. When MIPS16 Instruction Execution Is Enabled

(a) When MIPS III Instruction Is Executed

Instruction That Caused Exception	Contents of EPC Register	ErIM
Branch delay slot of branch or jump instruction	Virtual address of immediately preceding branch or jump instruction	0
Other than above	Virtual address of the instruction that caused the exception	0

(b) When MIPS16 Instruction Is Executed

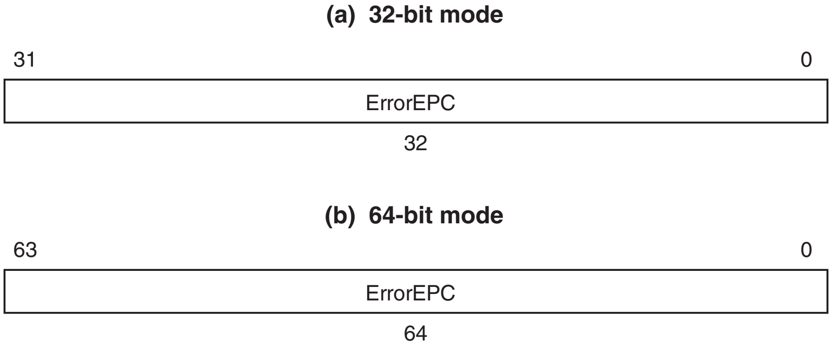
Instruction That Caused Exception	Contents of EPC Register	ErIM
Branch delay slot of jump instruction or latter half of Extend instruction	Virtual address of immediately preceding jump or Extend instruction	1
Other than above	Virtual address of the instruction that caused the exception	1

The contents of the ErrorEPC register do not change when the ERL bit of the status register is set to 1. This prevents the processor when other exceptions occur from overwriting the address of the instruction in this register which causes an error exception.

There is no branch delay slot indication for the ErrorEPC register.

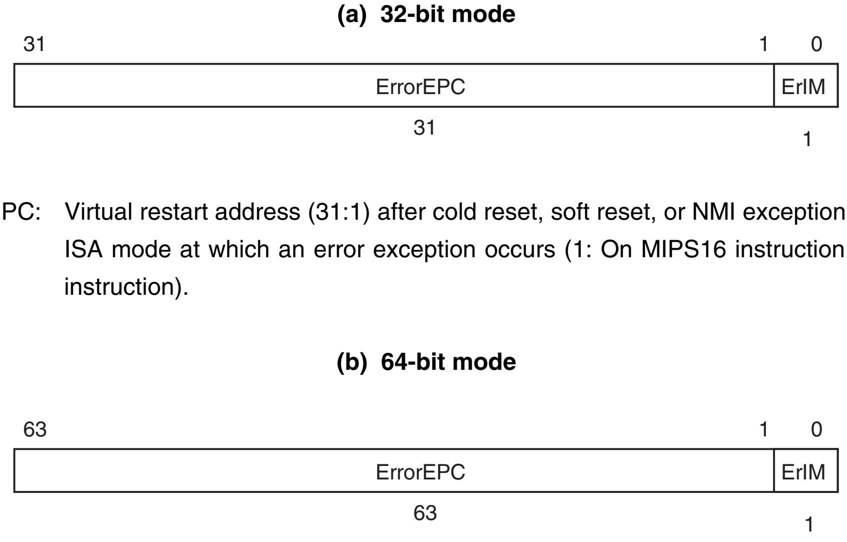
Figure 5-27 shows the format of the ErrorEPC register when the MIPS16ISA is disabled. Figure 5-28 shows the format of the ErrorEPC register when the MIPS16ISA is enabled.

Figure 5-27. ErrorEPC Register Format (When MIPS16ISA Is Disabled)



ErrorEPC: Program counter that indicates the restart address after cold reset, soft reset, or NMI exception.

Figure 5-28. ErrorEPC Register Format (When MIPS16ISA Is Enabled)



ErrorEPC: Virtual restart address (31:1) after cold reset, soft reset, or NMI exception.

ErIM: ISA mode at which an error exception occurs (1: On MIPS16 instruction, 0: On MIPS III instruction).

ErrorEPC: Virtual restart address (63:1) after Cold reset, Soft reset, or NMI exception.

ErIM: ISA mode at which an error exception occurs (1: On MIPS16 instruction, 0: On MIPS III instruction).

CHAPTER 6 INITIALIZATION INTERFACE

This chapter describes the initialization interface and processor modes. It also explains the reset signal descriptions and types, signal- and timing-related dependence, and the initialization sequence during each mode that can be selected by the user.

Remark # that follows signal names indicates active low.

6.1 Reset Function

There are four ways to reset the VR4131. Each is summarized below.

After reset, the processor, as the master of the system bus, executes the cold reset exception sequence and starts accessing the reset vectors of the ROM space. Only a few internal statuses are initialized when the VR4131 is reset, so the processor must be completely initialized by software.

6.1.1 RTC reset

The following shows the RTC reset sequence.

<1> Activate the RTCRST# pin at power application.

<2> After the power supply has become stable at 3.0 V or above, wait until the 32.768 kHz oscillator starts oscillating (approx. 2 seconds), then deassert the RTCRST# pin. The RTC unit then starts counting.

<3> After one RTC cycle, the states of the following pins are read.

- DDOUT/DBUS32/GPIO32 (switching data bus width)
- DRTS#/MIPS16EN/GPIO33 (enabling the use of MIPS16 instructions)
- TxD/CLKSEL2, RTS#/CLKSEL1, DTR#/CLKSEL0 (setting the operating frequency of the CPU core, BUSCLK frequency, internal bus clock frequency, etc.)

<4> When any of the POWER, DCD#, or GPIO3 pins becomes active[Note], the VR4131 activates the POWERON pin, checks that the status of the activated pin is stable for approximately 32 ms, and performs a startup check using the BATTINH/BATTINT# signals.

Note After RTCRST, GPIO3 is active low.

<5> If the startup check is successful, the VR4131 asserts the MPOWER pin and deasserts the POWERON pin.

<6> When the startup standby time (approx. 350 ms) has elapsed, the VR4131 starts the PLL oscillation and all the clocks start (after oscillation has started, however, approximately 16 ms are required as an oscillation stabilization period until the PLL oscillation is stabilized.

<7> When the PLL oscillation stabilization period has elapsed, the RST# pin outputs a low level. A low level continues to be output from RST# until the BMAS bit of the COMMANDREG register of PCIU is set.

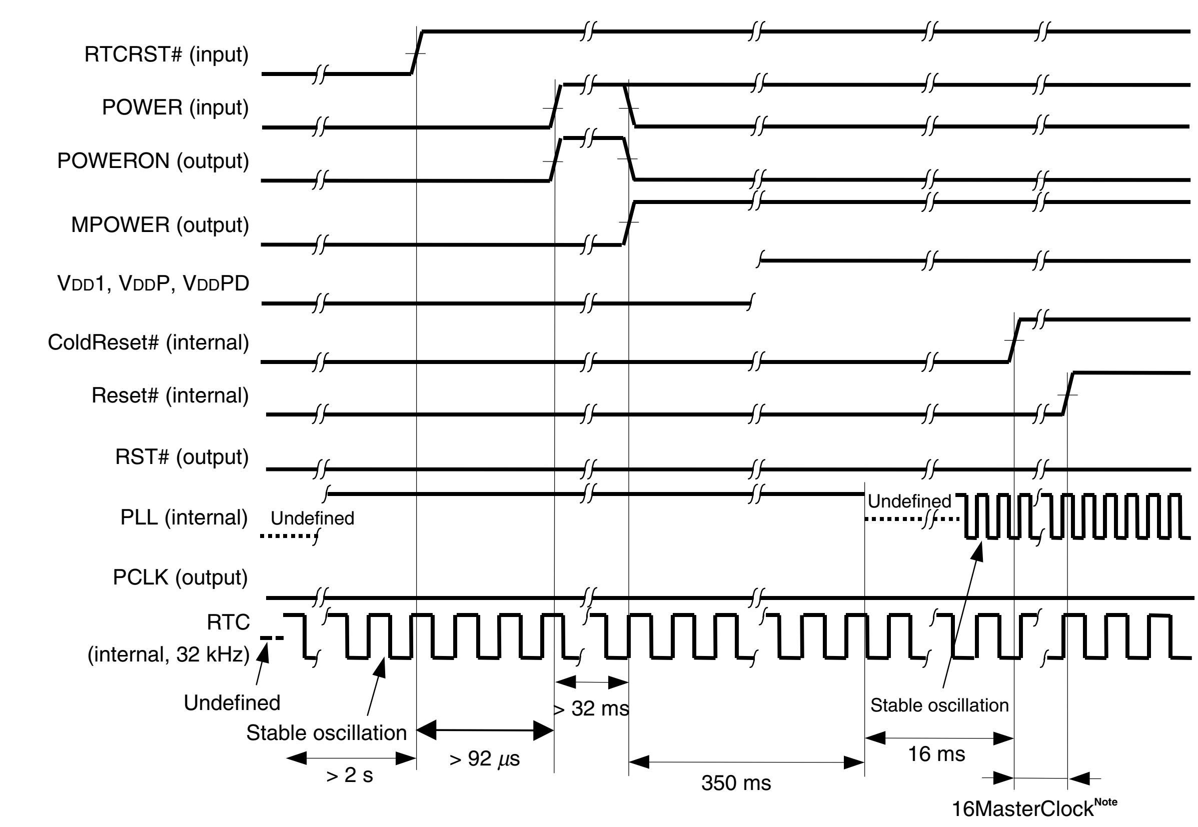
<8> After reset, the processor operates as a master of the system bus, executes the cold reset exception sequence, and starts accessing the reset vector in the ROM space. The VR4131 initializes the internal statuses in only a very limited area by a reset, so the processor must be completely initialized by software.

During an RTC reset, supplying voltage to the 1.5 V power-supply system (VDD1, VDDP, VDDPD) can be stopped to reduce the leakage current. The following operation will not be affected by supplying voltage of 1.5 V to these power supplies within the period from when the MPOWER pin becomes active to when PLL starts oscillation.

After Power-on, the processor's pin statuses are undefined from when RTCRST# becomes active until the 32.768 kHz clock oscillator starts oscillation. The pin statuses after oscillation starts are described **2.3.1**.

An RTC reset resets all the peripheral units including the RTC unit and the CPU core. It does not save any of the status information and it completely initializes the processor's internal state. Since the DRAM is not switched to self refresh mode, the contents of DRAM after an RTC reset are not at all guaranteed.

Figure 6-1. RTC Reset



Note MasterClock is the basic clock used in the CPU core.

When configuring a system that includes the VR4131, it is recommended that at least one or two 3.3 V power supplies be prepared besides the power supply to the VR4131.

MPOWER and SPOWER are used in the VR4131 as the power supply control signals for external devices.

Power supply control for SDRAM is described below.

(1) When using SDRAM

Two or more 3.3 V power supplies systems should be available besides the power supply to the VR4131. A power supply configuration example is shown below.

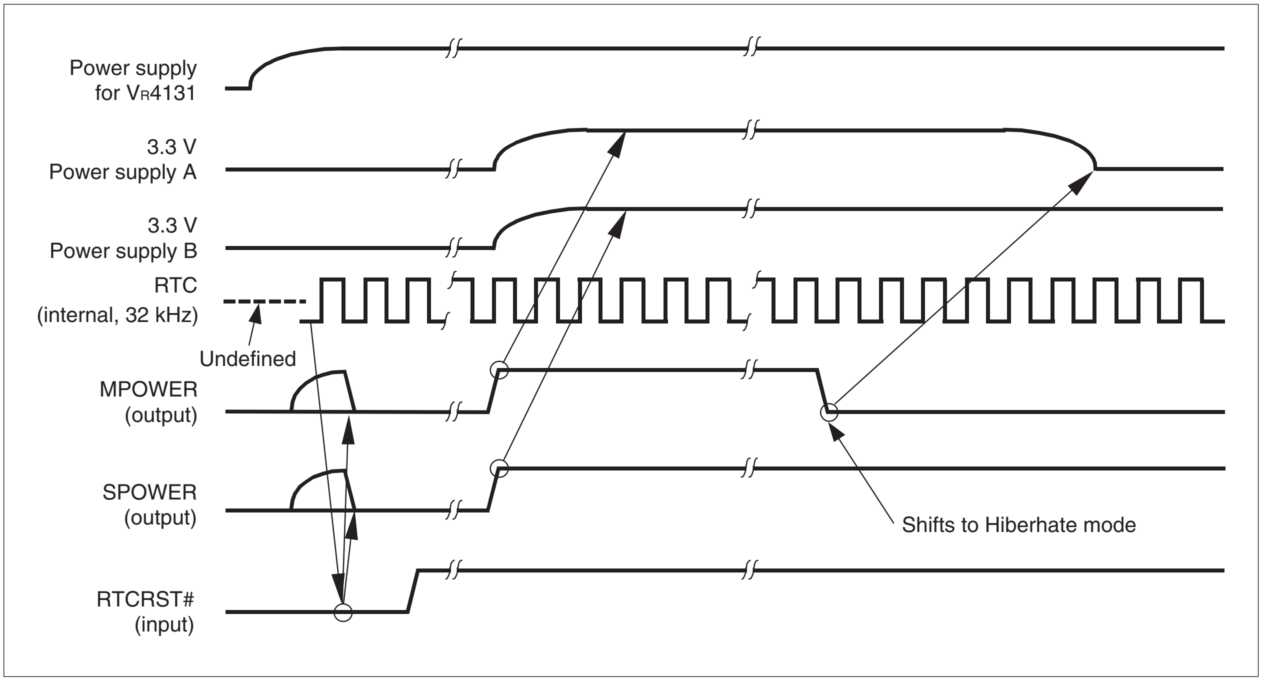
Example

Power supply for VR4131
    Application: Power supply for VR4131 operation and DRAM data retention
    Target devices for power supply: VR4131 (3.3 V power supply)
3.3 V power supply A
    Application: Power consumption reduction
    Target devices for power supply: ROM, External LSI
    Power supply control pin: MPOWER
3.3 V power supply B
    Application: Power consumption reduction and device protection
    Target devices for power supply: SDRAM
    Power supply control pin: SPOWER

The power supply control timing when using SDRAM is shown below.

When using SDRAM, once the RTC reset sequence begins, the VR4131 sets the SPOWER pin to low level and specifies suspending the power supply to SDRAM. After that, synchronization enables the MPOWER pin’s transition to high level, at which time the SPOWER pin also goes to high level. From then on, the SPOWER pin is held at high level even when changing to shut-down or Hibernate mode so as to continue supplying power to SDRAM.

Figure 6-2. Power Supply Control Timing When Using SDRAM

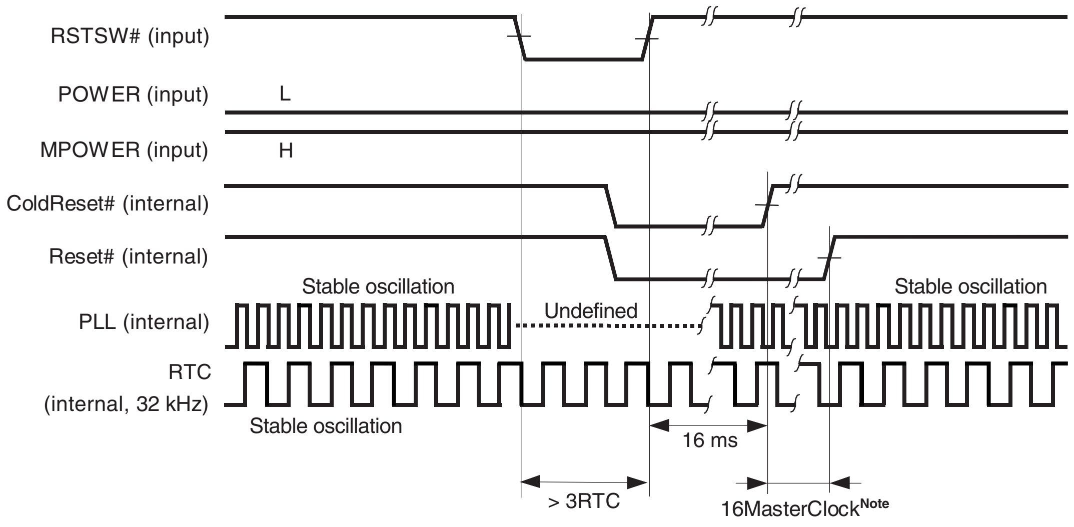


6.1.2 RSTSW

After the RSTSW# pin becomes active and then becomes inactive 100 μs later, the VR4131 starts PLL oscillation and starts all clocks (a period of about 16 ms following the start of PLL oscillation is required for stabilization of PLL oscillation).

A reset by RSTSW# initializes the entire internal state of all the peripheral units and CPU core except for the RTC timer and the PMU.

Figure 6-3. RSTSW



Note MasterClock is the basic clock used in the CPU core.

Caution If the RSTSW# signal becomes active at the same time the CPU transits to the Hibernate mode, the CPU may be activated without asserting the POWER signal after the MPOWER signal becomes inactive.

6.1.3 Software shutdown

When the software executes the HIBERNATE instruction, the V_R4131 sets the DRAM to self refresh mode and sets the MPOWER pin as inactive, then enters reset status. Recovery from reset status occurs when the POWER pin is asserted, when a WakeUp timer interrupt occurs, when the DCD# pin is asserted, or when the GPIO(0:3) [Note 1], GPIO(9:12)[Note 2] pins are asserted.

During a software shutdown, supplying voltage to the 1.5 V power-supply systems (V_{DD}1, V_{DD}P, V_{DD}PD) can be stopped to reduce the leakage current. The following operation will not be affected by supplying voltage within the range of the 1.5 V power supply voltage to these power supplies within the period from when the MPOWER pin becomes active to when PLL starts oscillation.

A reset by software shutdown initializes the entire internal state except for the RTC timer and the PMU.

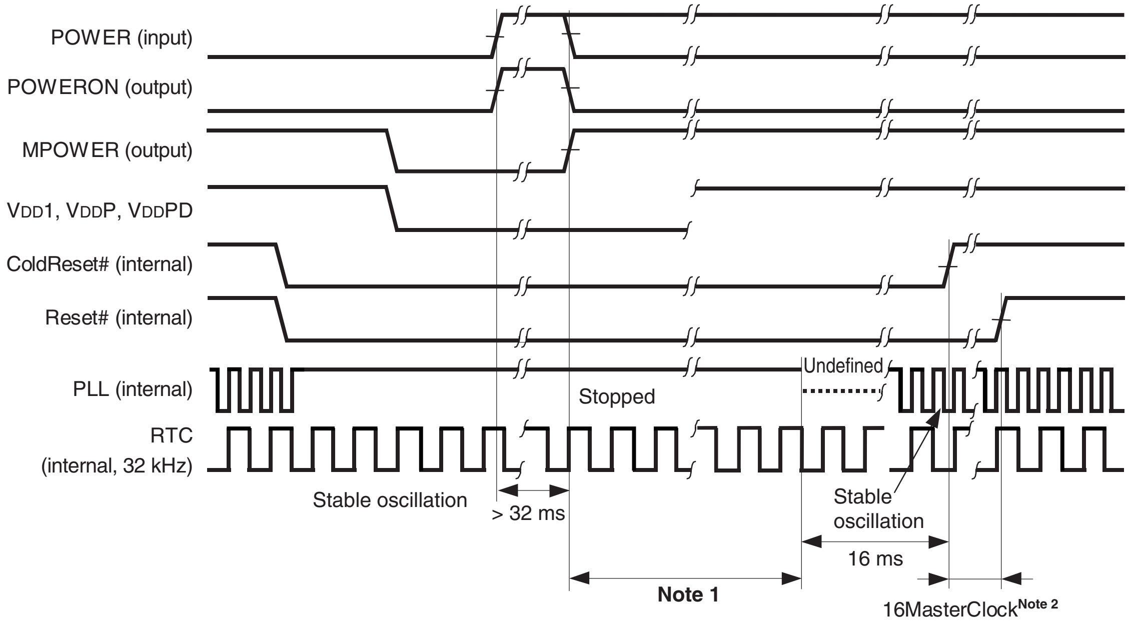
Notes 1. The startup enable and the active level setting using the GPI0 (0:3) pins are set by the PMUNTREG register.

2. The startup enable and the active level setting using the GPI0 (9:12) pins are set by the PMUCNTREG register.

After a reset, the processor becomes the system bus master and it begins the cold reset exception sequence to access the reset vectors in the ROM space. Since only part of the internal status is reset when a reset occurs in the VR4131, the processor should be completely initialized by software.

When switching from Fullspeed mode to Hibernate mode, the VR4131 enters self refresh mode in order to protect data in DRAM. When recovering from shutdown, a vector address for initialization is fetched after the lapse of the activation standby time and PLL oscillation stabilization time following the output of a high level from the MPOWER pin.

Figure 6-4. Software Shutdown



Notes 1. Wait time for activation. It can be changed by setting PMUWAITREG (refer to **12.2.5 PMUWAITREG (0x0B00 00C8)**).

2. MasterClock is the basic clock used in the CPU core.

6.1.4 HALTimer shutdown

After an RTC reset is canceled, if the HALTimer is not canceled by software within about 4 seconds (the HALTIMERRST bit of the PMUCNTREG register is not set to 1), the VR4131 enters reset status (refer to **12.1.2 Shutdown control**). Recovery from reset status occurs when the POWER pin is asserted or when a WakeUp timer interrupt occurs[Note].

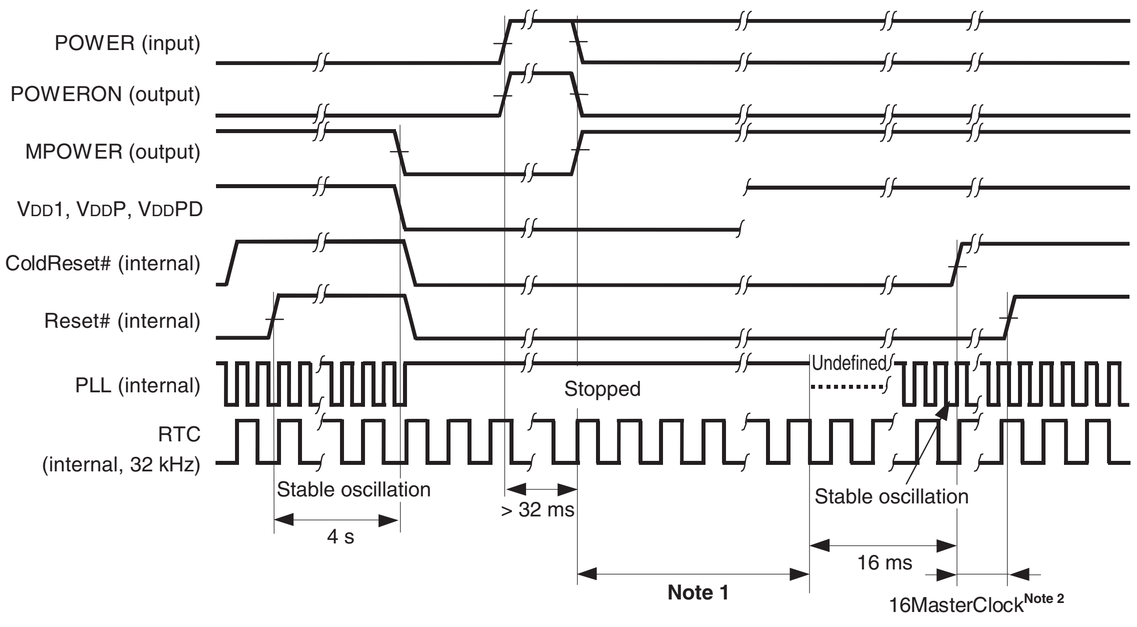
During a HALTimer shutdown, supplying voltage to the 1.5 V power-supply systems (VDD1, VDDP, VDDPD) can be stopped to reduce the leakage current. The following operation will not be affected by supplying voltage within the range of the 1.5 V power supply voltage to these power supplies within the period from when the MPOWER pin becomes active to when PLL starts oscillation.

A reset by HALTimer initializes the entire internal state except for the RTC timer and the PMU.

After a reset, the processor becomes the system bus master and it begins the cold reset exception sequence to access the reset vectors in the ROM space. Since only part of the internal status is reset when a reset occurs in the VR4131, the processor should be completely initialized by software.

Note Recovery using the GPIO (0:3) or GPIO (9:12) pins is not guaranteed.

Figure 6-5. HALTimer Shutdown



Notes 1. Wait time for activation. It can be changed by setting PMUWAITREG (refer to **12.2.5 PMUWAITREG (0x0B00 00C8)**).

2. MasterClock is the basic clock used inside the CPU core.

6.2 Power-On Sequence

The factors that cause the VR4131 to switch from Hibernate mode or shutdown mode to Fullspeed mode are called power-on factors. There are four power-on factors: assertion of the POWER pin, assertion of the DCD# pin, alarm from the WakeUp timer, and assertion of the GPIO pins (GPIO(0:3), GPIO(9:12)). When an activation factor occurs, the VR4131 asserts the POWERON pin, then provides notification to external circuits that the VR4131 is ready for power-on. After checking whether the pin status has become stable for 32 ms after the POWERON pin is asserted, the VR4122 checks the state of the BATTINH/BATTINT# pin. If the BATTINH/BATTINT# pin's state is low, the POWERON pin is deasserted one RTC clock after the BATTINH/BATTINT# pin check is completed, then the VR4131 is not activated. If the BATTINH/BATTINT# pin's state is high, the POWERON pin is deasserted three RTC clocks after the BATTINH/BATTINT# pin check is completed, then the MPOWER pin is asserted and the VR4131 is activated.

While the MPOWER pin is inactive, supplying voltage to the 1.5 V power supplies (VDD1, VDDP, VDDPD) can be stopped to reduce the leakage current. The following operation will not be affected by supplying voltage within the range of the 1.5 V power supply voltage to these power supplies within the period from when the MPOWER pin becomes active to when PLL starts oscillation.

Figure 6-6 shows a timing chart of VR4131 activation and Figure 6-7 shows a timing chart of when activation fails due to the BATTINH/BATTINT# pin's "low" state.

Figure 6-6. VR4131 Activation Sequence (When Activating)

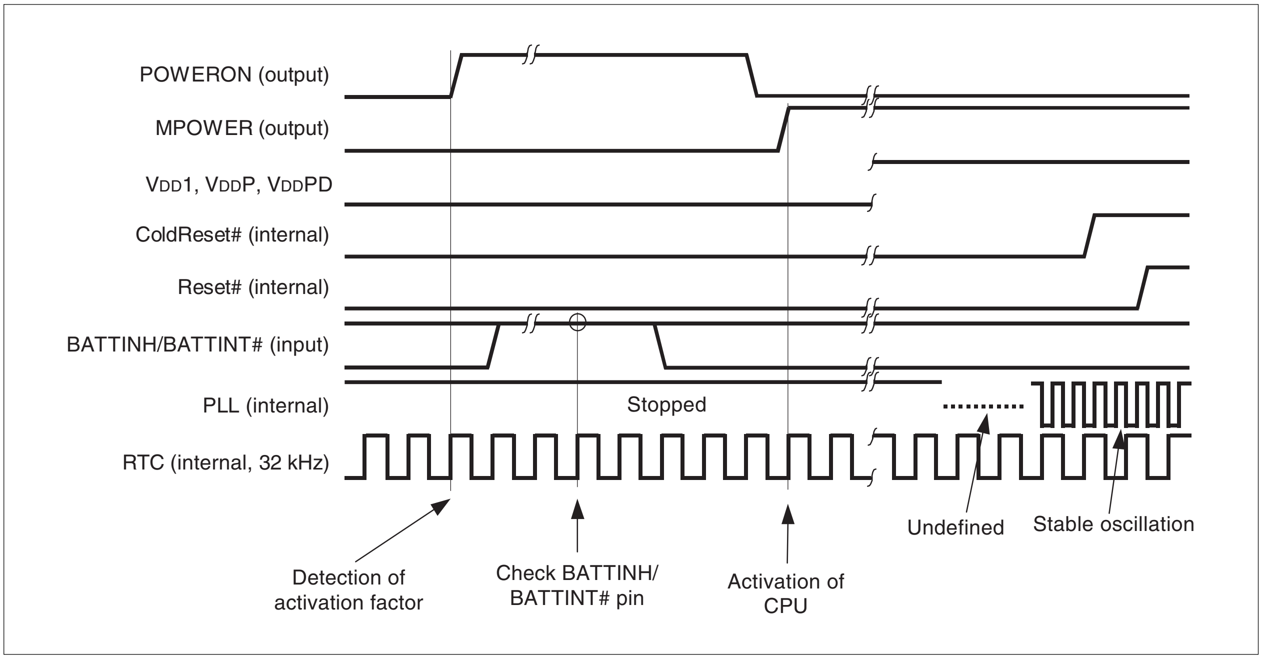
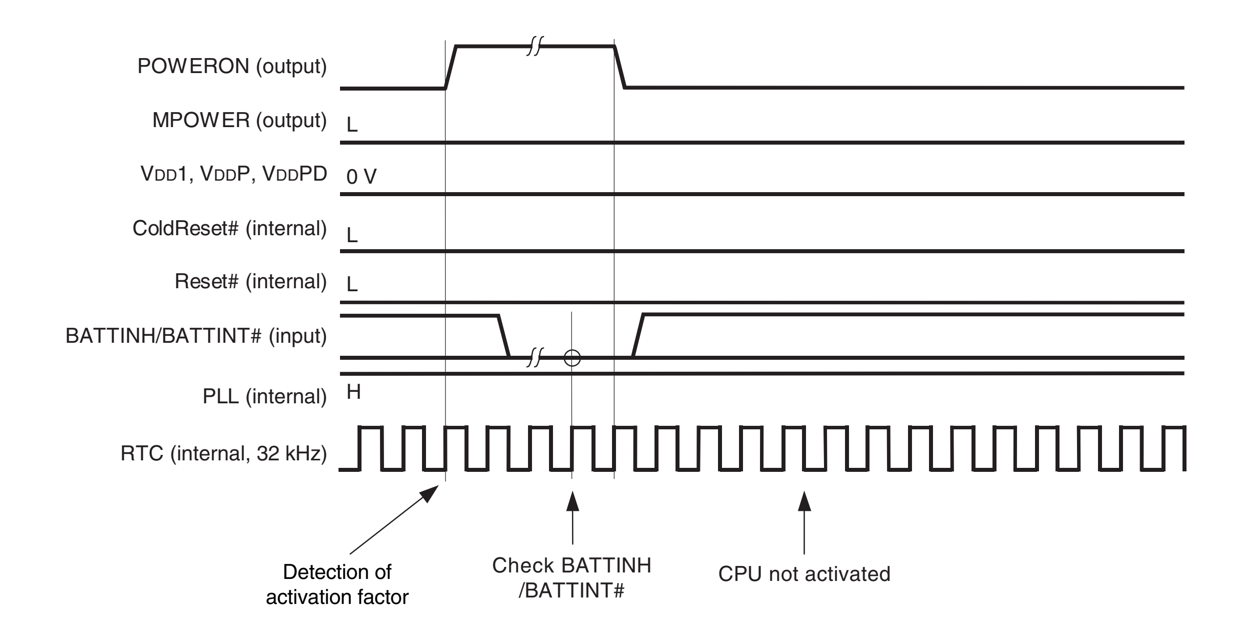


Figure 6-7. VR4131 Activation Sequence (When Not Activating)



6.3 Reset of CPU Core

This section describes the reset sequence of the VR4130 CPU core. For details about factors of reset or reset of the whole VR4131, refer to **6.1 Reset Function** and **CHAPTER 12 PMU (POWER MANAGEMENT UNIT)**.

6.3.1 Cold reset

In the VR4131, a cold reset sequence is executed in the CPU core in the following cases:

- RTC reset
- RSTSW reset
- Software shutdown
- HALTimer shutdown
- Battery low shutdown

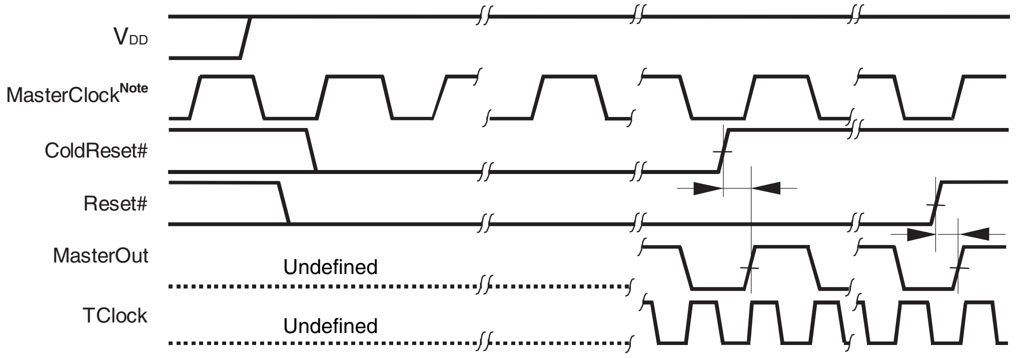
A cold reset completely initializes the CPU core, as follows.

- The SR bit of the status register is cleared to 0.
- The ERL and BEV bits of the status register are set to 1.
- The R and W bits of the WatchLo resister are cleared to 0.
- The upper limit value (31) is set in the random register.
- The wired register is initialized to 0.
- The count resister is initialized to 0.
- The IS, EC, BP, IB, and DB bits of the config register are set to 0 and the IC and DC bits are set to 4. The other bits are undefined.
- The values of the other registers are undefined.

Once power to the processor is established, the ColdReset# (internal) and the Reset# (internal) signals are asserted and a cold reset is started. After approximately 16 ms from assertion, the ColdReset# signal is deasserted synchronously with MasterOut (internal). Then the Reset# signal is deasserted synchronously with MasterOut, and the cold reset is completed.

Upon reset, the CPU core becomes bus master and drives the SysAD bus (internal). After the Reset# signal is deasserted, the CPU core branches to the reset exception vector and begins executing the reset exception code.

Figure 6-8. Cold Reset



Note MasterClock is the basic clock used inside the CPU core.

6.3.2 Soft reset

Caution In the VR4131, a soft reset never occurs.

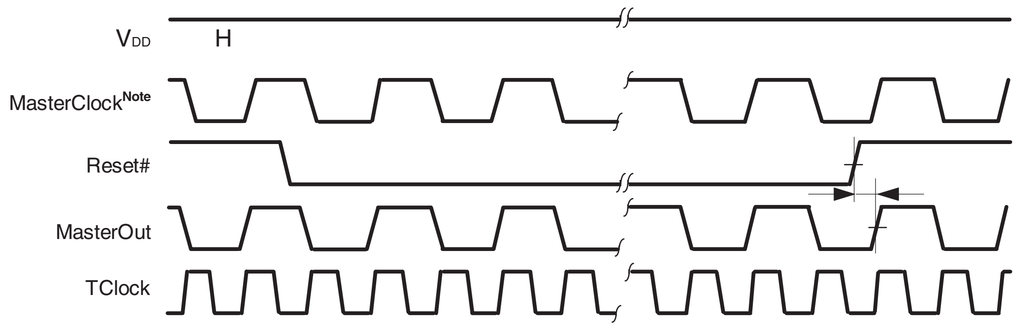
A soft reset initializes the CPU core without affecting the clocks; in other words, a soft reset is a logic reset. In a soft reset, the CPU core retains as much state information as possible; all state information except for the following is retained:

- The BEV, SR, and ERL bits of the status register are set to 1.
- The Count register is initialized to 0.
- The IP7 bit of the cause register is cleared to 0.
- Any interrupts generated on the SysAD bus are cleared.
- Non maskable interrupt is cleared.
- The config register is initialized.

A soft reset is started by assertion of the Reset# signal, and is completed at the deassertion of the Reset# signal. In general, data in the CPU core is preserved for debugging purpose.

Upon reset, the CPU core becomes bus master and drives the SysAD bus (internal). After Reset# is deasserted, the CPU core branches to the reset exception vector and begins executing the reset exception code.

Figure 6-9. Soft Reset



Note MasterClock is the basic clock used inside the CPU core.

6.4 VR4131 Processor Modes

The VR4131 supports various modes, which can be selected by the user. The CPU core mode is set each time a write occurs in the status register and config register. The on-chip peripheral unit mode is set by writing to the I/O register.

This section describes the CPU core's operation modes. For operation modes of on-chip peripheral units, refer to the chapters describing the various units.

6.4.1 Power modes

The VR4131 supports five power modes: Fullspeed mode, Standby mode, Suspend mode, Exsuspend mode, and Hibernate mode.

(1) Fullspeed mode

This is the normal operation mode.

The VR4131's default status sets operation under Fullspeed mode. After the processor is reset, the VR4122 returns to Fullspeed mode.

(2) Standby mode

When a STANDBY instruction has been executed, the processor can be set to Standby mode. During Standby mode, all of the internal clocks in the CPU core except for the timer and interrupt clocks are held at high level. The peripheral units all operate as they do during Fullspeed mode. This means that DMA operations are enabled during Standby mode.

When the STANDBY instruction completes the WB stage, the VR4131 remains idle until the SysAD internal bus enters the idle state. Next, the clocks in the CPU core are shut down and pipeline operation is stopped. However, the PLL, timer, interrupt clocks, TClock, and MasterOut continue to operate. In Standby mode, the processor is returned to Fullspeed mode if any interrupt occurs, including a timer interrupt that occurs internally.

(3) Suspend mode and Exsuspend mode

When a SUSPEND instruction has been executed, the processor can be set to Suspend mode. In Suspend mode, the processor stalls the pipeline and all of the internal clocks in the CPU core except for PLL timer and interrupt clocks are held at high level. The VR4131 stops supplying TClock to peripheral units. Accordingly, during Suspend mode peripheral units can only be activated by a special interrupt unit (DCD interrupt, etc.). While in this mode, the register and cache contents are retained.

When the SUSPEND instruction completes the WB stage, the VR4131 switches the DRAM to self refresh mode and then waits for the SysAD internal bus to enter the idle state. Next, the clocks in the CPU core are shut down and pipeline operation is stopped. The VR4131 then stops supplying TClock to peripheral units. However, the PLL, timer, interrupt clocks, and MasterOut continue to operate.

The processor remains in Suspend mode until an interrupt is received, at which time it returns to Fullspeed mode.

Setting the PLLOFFEN bit of the PMUCNTREG register to 1 allows the processor to enter the Exsuspend mode. The PLL operation also stops in the Exsuspend mode.

(4) Hibernate mode

When a HIBERNATE instruction has been executed, the processor can be set to Hibernate mode. During Hibernate mode, the processor stops supplying clocks to all units. The register and cache contents are retained and output of TClock and MasterOut is stopped. The processor is in Hibernate mode until the POWER pin is activated or a WakeUp timer interrupt is generated. The processor returns from Hibernate mode to Fullspeed mode by POWER pin activation or WakeUp timer interrupt generation.

In this mode, supplying voltage to the 1.5 V power supplies systems ($V_{DD}1$, $V_{DD}P$, $V_{DD}PD$) can be stopped. When the voltage of the 1.5 V power supplies becomes 0 V, the power dissipation becomes almost 0 W (it is not exactly 0 V because there are a 32.768 kHz oscillator and on-chip peripheral circuits operating at 32.768 kHz).

6.4.2 Privilege mode

The VR4131 supports three system modes: kernel expanded addressing mode, supervisor expanded addressing mode, and user expanded addressing mode. These three modes are described below.

(1) Kernel expanded addressing mode

When the status register's KX bit has been set, an expanded TLB refill exception vector is used when a TLB refill occurs for the kernel address. While in Kernel mode, the MIPS III operation code can always be used, regardless of the KX bit.

(2) Supervisor expanded addressing mode

When the status register's SX bit has been set, the MIPS III operation code can be used when in supervisor mode and an expanded TLB refill exception vector is used when a TLB refill occurs for the supervisor address.

(3) User expanded addressing mode

When the status register's UX bit has been set, the MIPS III operation code can be used when in user mode, and an expanded TLB refill exception vector is used when a TLB refill occurs for the user address. When this bit is cleared, the MIPS I and II operation codes can be used, as can 32-bit virtual addresses.

6.4.3 Reverse endian

When the status register's RE bit has been set, the endian ordering is reversed to adopt the user software's perspective. However, the VR4131 does not support the reverse endian.

6.4.4 Bootstrap exception vector (BEV)

The BEV bit is used to generate an exception during operation testing (diagnostic testing) of the cache and main memory system.

When the Status register's BEV bit has been set, the address of the TLB refill exception vector is changed to the virtual address 0xFFFF FFFF BFC0 0200, the XTLB refill exception vector is changed to the virtual address 0xFFFF FFFF BFC0 0280, and the general exception vector is changed to the address 0xFFFF FFFF BFC0 0380.

When the BEV bit is cleared, the TLB refill exception vector's address becomes 0xFFFF FFFF 8000 0000, the XTLB refill exception vector's address becomes 0xFFFF FFFF 8000 0080, and the general vector's address becomes 0xFFFF FFFF 8000 0180.

6.4.5 Cache error check

The Status register's CE bit has no meaning because the VR4131 does not support cache parity.

6.4.6 Parity error disable

The processor does not issue any cache parity error exceptions regardless of whether or not the Status register's DE bit is set.

6.4.7 Interrupt enable (IE)

When the Status register's IE bit has been cleared, no interrupts can be received except for reset interrupts and nonmaskable interrupts.

CHAPTER 7 BCU (BUS CONTROL UNIT)

7.1 Overview

The BCU exchanges data with the VR4130 CPU core via the internal SysAD bus inside the VR4131. It also controls an LCD controller, ROM (flash memory or mask ROM), and PCMCIA controller connected to the system bus, and transfers or receives data to or from the above devices via the ADD bus and DATA bus.

7.2 Register Set

Table 7-1 lists the registers of the BCU.

Table 7-1. BCU Registers

Address	R/W	Symbol	Function
0x0F00 0000	R/W	BCUCNTREG1	BCU control register 1
0x0F00 0004	R/W	ROMSIZEREG	ROM size register
0x0F00 0006	R/W	ROMSPEEDREG	ROM access cycle change register
0x0F00 0008	R/W	IO0SPEEDREG	I/O access cycle change register 0
0x0F00 000A	R/W	IO1SPEEDREG	I/O access cycle change register 1
0x0F00 0010	R	REVIDREG	Peripheral unit revision ID register
0x0F00 0014	R	CLKSPEEDREG	Clock setting register
0x0F00 0016	R/W	BCUCNTREG3	BCU control register 3

Each of these registers is explained in detail below.

7.2.1 BCUCNTREG1 (0x0F00 0000)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	PAGE SIZE[1]	PAGE SIZE[0]	RFU	PAGE ROM2	RFU	PAGE ROM0
R/W	R	R	R/W	R/W	R	R/W	R	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	ROMW EN2	RFU	ROMW EN0	RFU	HLDEN	RFU	RFU
R/W	R	R/W	R	R/W	R	R/W	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:14	RFU	Reserved. Write 0 to these bits. 0 is returned when they are read.
13:12	PAGESIZE(1:0)	Maximum burst access size for the PageROM to be used. 11: RFU 10: 32 bytes 01: 16 bytes 00: 8 bytes
11	RFU	Reserved. Write 0 to this bit. 0 is returned when it is read.
10	PAGEROM2	Enables PageROM access to the ROM space of bank 3 or 2 (in 16-bit mode) or bank 3, 2, or 1 (in 32-bit mode). 1: PageROM bus access 0: Normal ROM bus access
9	RFU	Reserved. Write 0 to this bit. 0 is returned when it is read.

Remark In 16-bit mode: Bank 3 (selected by ROMCS3#)
In 32-bit mode: Bank 1 (selected by ROMCS1#)

(2/2)

Bit	Name	Function
8	PAGEROM0	Enables PageROM access to the ROM space of bank 1 or 0 (in 16-bit mode) or bank 0 (in 32-bit mode). 1: PageROM bus access 0: Normal ROM bus access
7	RFU	Reserved. Write 0 to this bit. 0 is returned when it is read.
6	ROMWEN2	Enables flash memory write cycle and issues bus cycles dedicated to flash memory register read to the ROM space of bank 3 or 2 (in 16-bit mode) or bank 3, 2, or 1 (in 32-bit mode). 1: Enable (not affected by PAGEROM2 bit) 0: Disable
5	RFU	Reserved. Write 0 to this bit. 0 is returned when it is read.
4	ROMWEN0	Enables flash memory write cycle and issues bus cycles dedicated to flash memory register read to the ROM space of bank 1 or 0 (in 16-bit mode) or bank 0 (in 32-bit mode). 1: Enable (not affected by PAGEROM0 bit) 0: Disable
3	RFU	Reserved. Write 0 to this bit. 0 is returned when it is read.
2	HLDEN	Enables bus hold function. 1: Enables bus hold function In version 2.0 or later, when N-wire is not used, the HLDRQ# pin and HLDAK# pin can be used for the Hold Request function and Hold Acknowledge function, respectively. 0: Disables bus hold function
1:0	RFU	Reserved. Write 0 to these bits. 0 is returned when they are read.

Remark In 16-bit mode: Bank 3 (selected by ROMCS3#)
In 32-bit mode: Bank 1 (selected by ROMCS1#)

This register sets parameters such as a bus cycle of the bus interface and the type of memory to be used. When setting this register, also set BCUCNTREG3.

For details about the allocation of the ROM area bank, refer to **3.3.1 ROM address space**.

7.2.2 ROMSIZEREG (0x0F00 0004)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	RFU	SIZE32	SIZE31	SIZE30	RFU	SIZE22	SIZE21	SIZE20
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
RTCRST	0	0	0	1	0	0	0	1
After reset	0	0	0	1	0	0	0	1

Bit	7	6	5	4	3	2	1	0
Name	RFU	SIZE12	SIZE11	SIZE10	RFU	SIZE02	SIZE01	SIZE00
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
RTCRST	0	0	0	1	0	0	0	1
After reset	0	0	0	1	0	0	0	1

<table>
<tr><th>Bit</th><th>Name</th><th>Function</th></tr>
<tr><td>15</td><td>RFU</td><td>Reserved. Write 0 to this bit. 0 is returned when it is read.</td></tr>
<tr><td>14:12</td><td>SIZE3(2:0)</td><td>Selects the ROM capacity of bank 3 (in 16-bit mode) or bank 1 (in 32-bit mode)
<table>
<tr><th>SIZE3(2:0)</th><th>In 16-bit mode (MB)</th><th>In 32-bit mode (MB)</th></tr>
<tr><td>111</td><td>RFU</td><td>RFU</td></tr>
<tr><td>110</td><td>RFU</td><td>RFU</td></tr>
<tr><td>101</td><td>64</td><td>64</td></tr>
<tr><td>100</td><td>32</td><td>32</td></tr>
<tr><td>011</td><td>16</td><td>16</td></tr>
<tr><td>010</td><td>8</td><td>8</td></tr>
<tr><td>001</td><td>4</td><td>4</td></tr>
<tr><td>000</td><td>RFU</td><td>RFU</td></tr>
</table></td></tr>
<tr><td>11</td><td>RFU</td><td>Reserved. Write 0 to this bit. 0 is returned when it is read.</td></tr>
<tr><td>10:8</td><td>SIZE2(2:0)</td><td>Reserved. Selects the ROM capacity of bank 2 (in 16-bit mode) or bank 0 (in 32-bit mode).
<table>
<tr><th>SIZE2(2:0)</th><th>In 16-bit mode (MB)</th><th>In 32-bit mode (MB)</th></tr>
<tr><td>111</td><td>RFU</td><td>RFU</td></tr>
<tr><td>110</td><td>RFU</td><td>RFU</td></tr>
<tr><td>101</td><td>64</td><td>64</td></tr>
<tr><td>100</td><td>32</td><td>32</td></tr>
<tr><td>011</td><td>16</td><td>16</td></tr>
<tr><td>010</td><td>8</td><td>8</td></tr>
<tr><td>001</td><td>4</td><td>4</td></tr>
<tr><td>000</td><td>RFU</td><td>RFU</td></tr>
</table></td></tr>
</table>

(2/2)

<table>
<tr><th>Bit</th><th>Name</th><th>Function</th></tr>
<tr><td>7</td><td>RFU</td><td>Reserved. Write 0 to this bit. 0 is returned when it is read.</td></tr>
<tr><td>6:4</td><td>SIZE1(2:0)</td><td>Selects the ROM capacity of bank 1 (in 16-bit mode) or bank 3 (in 32-bit mode).
<table>
<tr><th>SIZE1(2:0)</th><th>In 16-bit mode
(MB)</th><th>In 32-bit mode
(MB)</th></tr>
<tr><td>111</td><td>RFU</td><td>RFU</td></tr>
<tr><td>110</td><td>RFU</td><td>RFU</td></tr>
<tr><td>101</td><td>64</td><td>64</td></tr>
<tr><td>100</td><td>32</td><td>32</td></tr>
<tr><td>011</td><td>16</td><td>16</td></tr>
<tr><td>010</td><td>8</td><td>8</td></tr>
<tr><td>001</td><td>4</td><td>4</td></tr>
<tr><td>000</td><td>RFU</td><td>RFU</td></tr>
</table></td></tr>
<tr><td>3</td><td>RFU</td><td>Reserved. Write 0 to this bit. 0 is returned when it is read.</td></tr>
<tr><td>2:0</td><td>SIZE0(2:0)</td><td>Selects the ROM capacity of bank 0 (in 16-bit mode) or bank 2 (in 32-bit mode).
<table>
<tr><th>SIZE0(2:0)</th><th>In 16-bit mode
(MB)</th><th>In 32-bit mode
(MB)</th></tr>
<tr><td>111</td><td>RFU</td><td>RFU</td></tr>
<tr><td>110</td><td>RFU</td><td>RFU</td></tr>
<tr><td>101</td><td>64</td><td>64</td></tr>
<tr><td>100</td><td>32</td><td>32</td></tr>
<tr><td>011</td><td>16</td><td>16</td></tr>
<tr><td>010</td><td>8</td><td>8</td></tr>
<tr><td>001</td><td>4</td><td>4</td></tr>
<tr><td>000</td><td>RFU</td><td>RFU</td></tr>
</table></td></tr>
</table>

This register is used to select the capacity of each bank in the ROM space.

7.2.3 ROMSPEEDREG (0x0F00 0006)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	rom4 _wait[1]	rom4 _wait[0]	RFU	RFU	RFU	RFU
R/W	R	R	R/W	R/W	R	R	R	R
RTCRST	0	0	1	1	0	0	0	0
After reset	0	0	1	1	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	rom2 _wait[3]	rom2 _wait[2]	rom2 _wait[1]	rom2 _wait[0]
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	1	1	1	1
After reset	0	0	0	0	1	1	1	1

Bit	Name	Function
15:14	RFU	Reserved. Write 0 to these bits. 0 is returned when they are read.
13:12	rom4_wait[1:0]	Selects a page access time of the ROM. 11: 5vtclk 10: 4vtclk 01: 3vtclk 00: 2vtclk
11:4	RFU	Reserved. Write 0 to these bits. 0 is returned when they are read.
3:0	rom2_wait[3:0]	Selects an access time of the ROM. 1111: 18vtclk 1110: 17vtclk 1101: 16vtclk 1100: 15vtclk 1011: 14vtclk 1010: 13vtclk 1001: 12vtclk 1000: 11vtclk 0111: 10vtclk 0110: 9vtclk 0101: 8vtclk 0100: 7vtclk 0011: 6vtclk 0010: 5vtclk 0001: 4vtclk 0000: 3vtclk

7.2.4 IO0SPEEDREG (0x0F00 0008)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	io0_5 _wait[1]	io0_5 _wait[0]	io0_3 _wait[3]	io0_3 _wait[2]	io0_3 _wait[1]	io0_3 _wait[0]
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	1	1	1	1	1	1
After reset	0	0	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
Name	io0_2 _wait[3]	io0_2 _wait[2]	io0_2 _wait[1]	io0_2 _wait[0]	io0_1 _wait[3]	io0_1 _wait[2]	io0_1 _wait[1]	io0_1 _wait[0]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	1	1	1	1	1	1	1	1
After reset	1	1	1	1	1	1	1	1

Bit	Name	Function
15:14	RFU	Reserved. Write 0 to these bits. 0 is returned when they are read.
13:12	io0_5_wait[1:0]	Hold time of the address and chip select signals from the rising of the read or write signal. 11: 4vtclk 10: 3vtclk 01: 2vtclk 00: 1vtclk
11:8	io0_3_wait[3:0]	Time from the ready signal of the IORDY signal to the rising of the read or write signal.
7:4	io0_2_wait[3:0]	Time from the falling of the read or write signal to the start of the IORDY signal watch.
3:0	io0_1_wait[3:0]	Time from the falling of the IOCS signal to the falling of the read or write signal.

This register sets the access parameters of external I/O space 2 (IOCS0#).

	io0_3_wait	io0_2_wait	io0_1_wait
1111	18	14	16
1110	17	13	15
1101	16	12	14
1100	15	11	13
1011	14	10	12
1010	13	9	11
1001	12	8	10
1000	11	7	9
0111	10	6	8
0110	9	5	7
0101	8	4	6
0100	7	3	5
0011	6	2	4
0010	5	1	3
0001	4	0	2
0000	3	–1	1

7.2.5 IO1SPEEDREG (0x0F00 000A)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	io1_5 _wait[1]	io1_5 _wait[0]	io1_3 _wait[3]	io1_3 _wait[2]	io1_3 _wait[1]	io1_3 _wait[0]
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	1	1	1	1	1	1
After reset	0	0	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
Name	io1_2 _wait[3]	io1_2 _wait[2]	io1_2 _wait[1]	io1_2 _wait[0]	io1_1 _wait[3]	io1_1 _wait[2]	io1_1 _wait[1]	io1_1 _wait[0]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	1	1	1	1	1	1	1	1
After reset	1	1	1	1	1	1	1	1

Bit	Name	Function
15:14	RFU	Reserved. Write 0 to these bits. 0 is returned when they are read.
13:12	io1_5_wait[1:0]	Hold time of the address and chip select signals from the rising of the read or write signal. 11: 4vtclk 10: 3vtclk 01: 2vtclk 00: 1vtclk
11:8	io1_3_wait[3:0]	Time from the ready signal of the IORDY signal to the rising of the read or write signal.
7:4	io1_2_wait[3:0]	Time from the falling of the read or write signal to the start of the IORDY signal watch.
3:0	io1_1_wait[3:0]	Time from the falling of the IOCS signal to the falling of the read or write signal.

This register sets the access parameters of external I/O space 1 (IOCS1#).

	io1_3_wait	io1_2_wait	io1_1_wait
1111	18	14	16
1110	17	13	15
1101	16	12	14
1100	15	11	13
1011	14	10	12
1010	13	9	11
1001	12	8	10
1000	11	7	9
0111	10	6	8
0110	9	5	7
0101	8	4	6
0100	7	3	5
0011	6	2	4
0010	5	1	3
0001	4	0	2
0000	3	–1	1

7.2.6 REVIDREG (0x0F00 0010)

Bit	15	14	13	12	11	10	9	8
Name	RID3	RID2	RID1	RID0	MJREV3	MJREV2	MJREV1	MJREV0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	1	0	0	0	0	0	0
After reset	0	1	0	0	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	MNREV3	MNREV2	MNREV1	MNREV0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15:12	RID(3:0)	Processor revision ID. 0x05 for the VR4131.
11:8	MJREV(3:0)	Major revision number
7:4	RFU	Reserved. Write 0 to these bits. 0 is returned when they are read.
3:0	MNREV(3:0)	Minor revision number

Note The setting of this bit differs depending on the period of shipping.

This register indicates the revision numbers of the peripheral units of the VR4131.

The revision numbers of the peripheral units are indicated in the form of yx, where y is the major revision number and x is the minor revision number.

If the CPU core or peripheral units are changed, however, it is not always reflected on REVIDREG. Moreover, if the revision number is changed, the actual CPU core or peripheral units are not always changed. Therefore, develop a program that does not depend on the value of REVDIREG.

7.2.7 CLKSPEEDREG (0x0F00 0014)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	TDIVMODE	RFU	VTDIV MODE2	VTDIV MODE1	VTDIV MODE0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	Undefined	0	Undefined	Undefined	Undefined
After reset	0	0	0	Undefined	0	Undefined	Undefined	Undefined

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	CLKSP4	CLKSP3	CLKSP2	CLKSP1	CLKSP0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	Undefined	Undefined	Undefined	Undefined	Undefined
After reset	0	0	0	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Name	Function
15:13	RFU	Reserved. Write 0 to these bits. 0 is returned when they are read.
12	TDIVMODE	Sets the division ratio of the operating clock (TCLK) of the internal peripherals to the peripheral interface clock (VTClock). 0: 1/2 1: 1/4
11	RFU	Reserved. Write 0 to this bit. 0 is returned when it is read.
10:8	VTDIVMODE(2:0)	Sets the division ratio of the operating clock (VTClock) of the SDRAM/ROM/external I/O to the frequency of the CPU core operating clock (PClock). 110: 1/6 101: 1/5 100: 1/4 011: 1/3 010: 1/2 Others: RFU
7:5	RFU	Reserved. Write 0 to these bits. 0 is returned when they are read.
4:0	CLKSP(4:0)	Value used to calculate the frequency of the operating clock (PClock) of the CPU core.

The value of the CLKSPEEDREG is changed by the CLKSEL(2:0) pin after RTCRST or reset, or the setting value of PMUTCLKDIVREG.

The frequency of each operating clock can be calculated as follows:

PClock = (18.432 MHz/CLKSP(4:0)) × 108
VTClock = PClock/(Division ratio specified by VTDIVMODE(2:0))
TClock = VTClock/(Division ratio specified by TDIVMODE)

7.2.8 BCUCNTREG3 (0x0F00 0016)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	EXT_ ROMCS1	EXT_ ROMCS0	RFU	RFU	RFU	RFU
R/W	R	R	R/W	R/W	R	R	R	R
RTCRST	0	0	1	1	1	0	0	0
After reset	0	0	Hold	Hold	1	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	IO32	RFU	RFU	RFU	SYSDIR_ EN	RFU	LCDSEL1	LCDSEL0
R/W	R/W	R	R	R	R/W	R	R/W	R/W
RTCRST	0	0	0	0	1	0	0	0
After reset	Hold	0	0	0	Hold	0	Hold	Hold

Bit	Name	Function
15:14	RFU	Reserved. Write 0 to these bits. 0 is returned when they are read.
13:12	EXT_ROMCS(1:0)	Allocates bank 3 or 2 in 32-bit data bus mode. 11: Bank 3 = ROM, bank 2 = ROM 10: Bank 3 = ROM, bank 2 = DRAM 01: RFU 00: Bank 3 = DRAM, bank 2 = DRAM
11	RFU	Reserved. Write 1 to this bit. 1 is returned when it is read.
10:8	RFU	Reserved. Write 0 to these bits. 0 is returned when they are read.
7	IO32	Data bus size of the I/O space in the 32-bit data bus mode. 1: 32 bits 0: 16 bits
6:4	RFU	Reserved. Write 0 to these bits. 0 is returned when they are read.
3	SYSDIR_EN	Selects the function of the GPIO06_SYSDIR pin. 1: GPIO06_SYSDIR pin is used as SYSDIR. 0: GPIO06_SYSDIR pin is used as GPIO06.
2	RFU	Reserved. Write 0 to this bit. 0 is returned when it is read.
1	LCDSEL1	Connection position of the device allocated to physical addresses 0x0DFF FFFF to 0x0C00 0000. 1: Not via load mitigation buffer 0: Via load mitigation buffer
0	LCDSEL0	Connection position of the device allocated to physical addresses 0x0BFF FFFF to 0x0A00 0000. 1: Not via load mitigation buffer 0: Via load mitigation buffer

Bits EXT_ROMCS (1:0) and IO32 can be set only in the 32-bit data bus mode (DBUS32 = 1).

In the VR4131, it is recommended to insert a load mitigation buffer in the DATA bus and ADD bus when SDRAM is used. If a buffer is inserted, the direction of the buffer can be controlled depending on whether the buffer for which a memory device is inserted by LCDSEL is set.

7.3 Connecting Address Pins

The VR4131 supplies the physical address that the CPU core outputs to ROM/SDRAM/external I/O via the ADD bus. The following table shows the correspondence between the physical address bit output to the ADD bus and the address bit of the external device.

Table 7-2. Correspondence Between ADD Bus and Address Bits of External Device

ADD Pin Connected Device	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01
SDRAM (row) DATA(15:0)	18	17	24	22	21	20	19	16	15	14	13	12	11	10	09	Note 2	Note 2	Note 2	Note 2	Note 2	Note 2	Note 2	Note 2	Note 2
SDRAM (column) DATA(15:0)	18	17	24	22	Note 1	20	23	08	07	06	05	04	03	02	01	Note 2	Note 2	Note 2	Note 2	Note 2	Note 2	Note 2	Note 2	Note 2
SDRAM (row) DATA(31:0)	19	18	25	23	22	21	20	17	16	15	14	13	12	11	10	Note 2	Note 2	Note 2	Note 2	Note 2	Note 2	Note 2	Note 2	Note 2
SDRAM (column) DATA (31:0)	19	18	25	23	Note 1	21	24	09	08	07	06	05	04	03	02	Note 2	Note 2	Note 2	Note 2	Note 2	Note 2	Note 2	Note 2	Note 2
ROM DATA(15:0)	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01
ROM DATA(31:0)	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	25
External I/O	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01

Notes 1. CMD

2. Undefined value (1 or 0)

7.3.1 ROM connection

Table 7-3 shows an example of connecting the VR4131 to ROM.

Table 7-3. Example of ROM Connection and Address Output by VR4131 (1/2)

(a) 16-bit data bus mode (DBUS32 = 0)

ROM Address Pin	With 32 Mb ROM (2 Mb × 16)		With 64 Mb ROM (4 Mb × 16)		With 128 Mb ROM (8 Mb × 16)	
	ADD pin	adr	ADD pin	adr	ADD pin	adr
A22					ADD23	adr23
A21			ADD22	adr22	ADD22	ade22
A20	ADD21	adr21	ADD21	adr21	ADD21	adr21
A19	ADD20	adr20	ADD20	adr20	ADD20	adr20
A18	ADD19	adr19	ADD19	adr19	ADD19	adr19
A17	ADD18	adr18	ADD18	adr18	ADD18	adr18
A16	ADD17	adr17	ADD17	adr17	ADD17	adr17
A15	ADD16	adr16	ADD16	adr16	ADD16	adr16
A14	ADD15	adr15	ADD15	adr15	ADD15	adr15
A13	ADD14	adr14	ADD14	adr14	ADD14	adr14
A12	ADD13	adr13	ADD13	adr13	ADD13	adr13
A11	ADD12	adr12	ADD12	adr12	ADD12	adr12
A10	ADD11	adr11	ADD11	adr11	ADD11	adr11
A9	ADD10	adr10	ADD10	adr10	ADD10	adr10
A8	ADD9	adr9	ADD9	adr9	ADD9	adr9
A7	ADD8	adr8	ADD8	adr8	ADD8	adr8
A6	ADD7	adr7	ADD7	adr7	ADD7	adr7
A5	ADD6	adr6	ADD6	adr6	ADD6	adr6
A4	ADD5	adr5	ADD5	adr5	ADD5	adr5
A3	ADD4	adr4	ADD4	adr4	ADD4	adr4
A2	ADD3	adr3	ADD3	adr3	ADD3	adr3
A1	ADD2	adr2	ADD2	adr2	ADD2	adr2
A0	ADD1	adr1	ADD1	adr1	ADD1	adr1

Remark adr(23:1): Physical address bits of CPU core or DMAAU

Table 7-3. Example of ROM Connection and Address Output by VR4131 (2/2)

(b) 32-bit data bus mode (DBUS32 = 1)

ROM Address Pin	With 32 Mb ROM (2 Mb × 16)		With 64 Mb ROM (4 Mb × 16)		With 128 Mb ROM (8 Mb × 16)		With 256 Mb ROM (16 Mb × 16)	
	ADD pin	adr	ADD pin	adr	ADD pin	adr	ADD pin	adr
A23							ADD1	adr25
A22					ADD24	adr24	ADD24	adr24
A21			ADD23	adr23	ADD23	adr23	ADD23	adr23
A20	ADD22	adr22	ADD22	adr22	ADD22	adr22	ADD22	adr22
A19	ADD21	adr21	ADD21	adr21	ADD21	adr21	ADD21	adr21
A18	ADD20	adr20	ADD20	adr20	ADD20	adr20	ADD20	adr20
A17	ADD19	adr19	ADD19	adr19	ADD19	adr19	ADD19	adr19
A16	ADD18	adr18	ADD18	adr18	ADD18	adr18	ADD18	adr18
A15	ADD17	adr17	ADD17	adr17	ADD17	adr17	ADD17	adr17
A14	ADD16	adr16	ADD16	adr16	ADD16	adr16	ADD16	adr16
A13	ADD15	adr15	ADD15	adr15	ADD15	adr15	ADD15	adr15
A12	ADD14	adr14	ADD14	adr14	ADD14	adr14	ADD14	adr14
A11	ADD13	adr13	ADD13	adr13	ADD13	adr13	ADD13	adr13
A10	ADD12	adr12	ADD12	adr12	ADD12	adr12	ADD12	adr12
A9	ADD11	adr11	ADD11	adr11	ADD11	adr11	ADD11	adr11
A8	ADD10	adr10	ADD10	adr10	ADD10	adr10	ADD10	adr10
A7	ADD9	adr9	ADD9	adr9	ADD9	adr9	ADD9	adr9
A6	ADD8	adr8	ADD8	adr8	ADD8	adr8	ADD8	adr8
A5	ADD7	adr7	ADD7	adr7	ADD7	adr7	ADD7	adr7
A4	ADD6	adr6	ADD6	adr6	ADD6	adr6	ADD6	adr6
A3	ADD5	adr5	ADD5	adr5	ADD5	adr5	ADD5	adr5
A2	ADD4	adr4	ADD4	adr4	ADD4	adr4	ADD4	adr4
A1	ADD3	adr3	ADD3	adr3	ADD3	adr3	ADD3	adr3
A0	ADD2	adr2	ADD2	adr2	ADD2	adr2	ADD2	adr2

Remark adr(25:2): Physical address bits of CPU core or DMAAU

7.4 Notes on Using BCU

7.4.1 Bus mode of CPU core

The VR4131 is designed on the assumption that the CPU core is set in the following mode:

- Write-back data rate: D
- AD mode: VR4x00-compatible mode

Therefore, be sure to set the config register of the CPU core as follows:

- EP area: 0000
- AD bit: 0

7.4.2 Access size

The data size when the VR4131 accesses each address space is limited. The access size of each access space is shown below.

Table 7-4. Access Data Size for Each Access Space

Address Space	R/W	Access Size (Bytes)							**Note**
		32	16	8	4	3	2	1	
ROM/PageROM	R	○	○	○	○	○	○	○	
Flash memory	W	×	×	×	Δ	×	Δ	×	1
Flash memory	R	○	○	○	○	○	○	○	
External I/O space	R/W	○	○	○	○	×	○	○	
Internal I/O (1)	R/W	×	○	○	○	×	○	○	2
Internal I/O (2)	R/W	×	×	○	○	×	○	×	3
Internal I/O (3)	R/W	×	×	×	○	×	○	×	4
Internal I/O (4)	R/W	×	×	×	○	×	×	×	5
DRAM	R/W	○	○	○	○	○	○	○	
PCI space	R/W	○	○	○	○	○	○	○	

Notes 1. Make sure that the access size for writing the flash memory is the same as the bus width of the data bus as follows:

32-bit data bus mode: 4 bytes
16-bit data bus mode: 2 bytes

2. CSIU, FIR, DSIU, SIU

3. BCU, DMAAU, DCU, CMU, PMU, RTC, GIU, LED

4. SDRAMU, SCU, PCIU (PCICLKRUNREG register)

5. PCIU (except the PCICLKRUNREG register)

Remark ○, Δ: Can be accessed, ×: Cannot be accessed

7.4.3 ROM interface

This section explains the setting and usage of the ROM when it is used with the VR4131.

(1) Selecting ROM, PageROM, or flash memory

The VR4131 supports three ROM modes for the ROM interface: normal ROM, PageROM, and flash memory. The mode of a ROM bank is selected by using the ROMWEN and PAGEROM bits of BCUCNTREG1.

Table 7-5. ROM Mode Setting and Accessible Device

Mode	Setting		Accessible Device		
	ROMWEN2/0	PAGEROM2/0	Memory Read	Flash Memory Register Read	Flash Memory Write
Normal ROM	0	0	Normal ROM PageROM Flash memory	None	None
PageROM	0	1	PageROM	None	None
Flash memory	1	don’t care	Normal ROM PageROM Flash memory	Flash memory	Flash memory

Remark The normal ROM mode is the default.

(2) Setting of access speed

The VR4131 can change the access speed when it operates in the normal ROM or PageROM mode. For details, refer to **7.5.1 ROM access**.

7.4.4 Flash memory interface

(1) Notes on each mode

The flash memory interface has the following two modes:

- Normal ROM mode (dedicated to memory read)
- Flash memory read (supports write/register read)

The points to be noted in each mode are described below.

(a) Normal ROM mode

- Prohibits writing.
  Even when writing is executed, the WR# pin is not asserted active.
- Reading the flash memory register is prohibited.
  The normal ROM mode issues a bus cycle suitable for reading memory.
  Because the AC characteristics of the flash memory differ when reading registers and when reading memory, correct data cannot be obtained even if the flash memory register is read in this mode.

(b) Flash memory mode

- Be sure to access in 2- or 4-byte units (depending on the data bus width) when writing to the flash memory.

(2) Example of write sequence to flash memory

Here is an example of a write sequence to the flash memory.

Caution The operations in this example are not checked on the actual system.

1. Using a GPIO as an output port, apply a write voltage (V_{PP}) to the flash memory. If an internal GPIO of the VR4131 cannot be used, use an external output port to control the write voltage.
2. Set the VR4131 in the flash memory mode (by setting the ROMWEN bit of BCUCNTREG1 to 1).
3. Wait until the write voltage to the flash memory has stabilized.
4. Issue a write command to the flash memory from the VR4131.
5. Write data to the flash memory from the VR4131.
6. Wait until the write end signal of the flash memory (ry/by) has stabilized.
7. Wait until the write end signal of the flash memory reports completion of writing. Completion of writing the flash memory can be detected by an interrupt triggered by the write end signal of the flash memory (ry/by), or by polling a register of the flash memory.
8. Read the flash memory register.
   - If writing has been successful, proceed to step 9.
   - If writing has failed, proceed to step 12.
9. To write new data to the flash memory, return to step 4.
   To end writing the flash memory, proceed to step 10.
10. Compare the data written to the flash memory with the original data.
    - If the data matches, proceed to step 11.
    - If the data does not match when writing again, proceed to step 11 and then back to step 1. To abort the processing, execute from 11.
11. Drop the write voltage (V_{PP}) of the flash memory, clear the flash memory mode, and complete the processing.
12. Clear the error information of the flash memory register.
    - If the write voltage is too low when writing the data again, return to step 1. Otherwise, return to step 4.
    - To complete the processing, perform processing of step 11.

(3) Memory capacity to be used

Set the memory (ROM and DRAM) capacity to be used.

(4) Using extended memory when data bus size is 32 bits

The extended memory (ROM and DRAM) is enabled by setting appropriate values to EXT_ROMCS(3:2) of BCUCNTREG3 if the data bus size is 32 bits. However, the physical address and bank are changed according to the data bus mode as shown in sections **3.1.1 ROM address space** and **3.1.6 DRAM address space**, so be careful when setting.

7.4.5 External I/O Interface

(1) Access size

Access an external I/O in 1 byte, 2 bytes, 4 bytes, 8 bytes, 16 bytes, or 32 bytes.

(2) Data bus size

If setting the IO bit of BCUCNTREG3 to 1 in the 32-bit data bus mode (DBUS32 = 1), the VR4131 expands the data bus of an external I/O interface to 32 bits (the default is 16 bits). Figures 7-1 and 7-2 show examples of the connection with the 16-bit I/O device and 32-bit I/O device, respectively.

Figure 7-1. Connection with 16-Bit I/O Device Interface

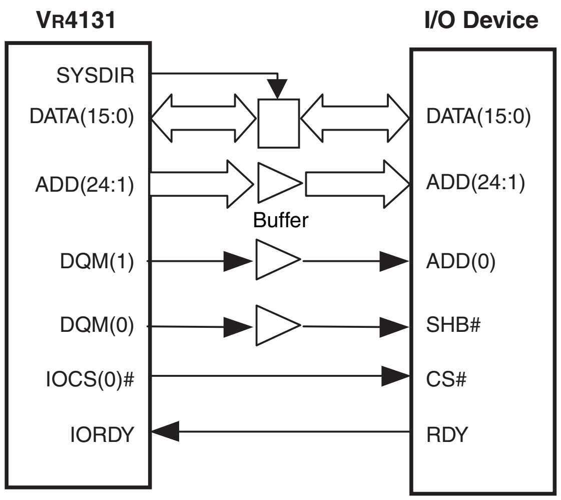
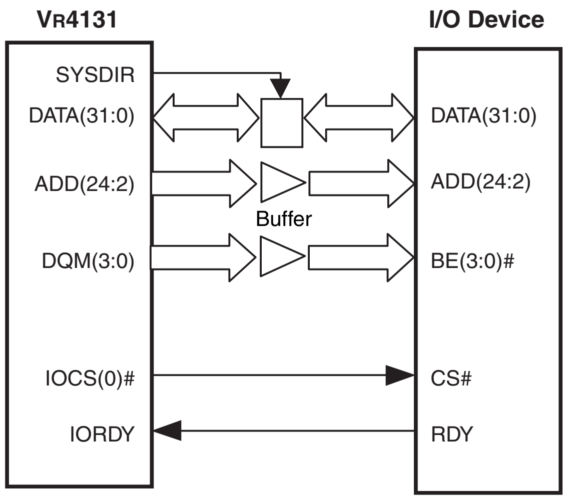


Figure 7-2. Connection with 32-Bit I/O Device Interface



7.4.6 Load mitigation buffer when SDRAM is used

When SDRAM is connected to the VR4131, the load capacitance must be reduced because the address/data bus operates at high speed when the SDRAM is accessed. To reduce the load capacitance, the VR4131 recommends the connection of the address/data bus via the load mitigation buffer when accessing an external I/O space.

The VR4131 uses the SYSDIR/GPIO6 to control the load mitigation buffer. The SYSDIR/GPIO6 pin functions as an SYSDIR signal when setting SYSDIR_EN of BCUCNTREG3 to 1. Otherwise, the pin functions as a GPIO6 signal when setting it to 0.

(1) Space for which load mitigation buffer can be used

The load mitigation buffer can be used for the address/data bus accessing of the following address spaces:

- External I/O space 1 (0x0DFF FFFF through 0x0C00 0000)
- External I/O space 2 (0x0BFF FFFF through 0x0A00 0000)

Whether to use the buffer or not can be selected for the upper or lower address space, individually. This setting is specified using LCDSEL(1:0) bits of BCUCNTREG3.

(2) Controlling data bus buffer

The data bus buffer control signals are used as follows:

- SYSDIR/GPIO6: Controls the direction of the buffer used for the data bus (for details, refer to **(3)** below).
  1: External I/O device → CPU
  0: CPU → external I/O device

Remark In the VR4131, direction control of the buffer used for the address bus is not performed. In a system which uses a bus hold function, configure the system so that the external master can directly drive the SDRAM addresses (ADD (24:10)).

(3) Buffer control timing for data bus

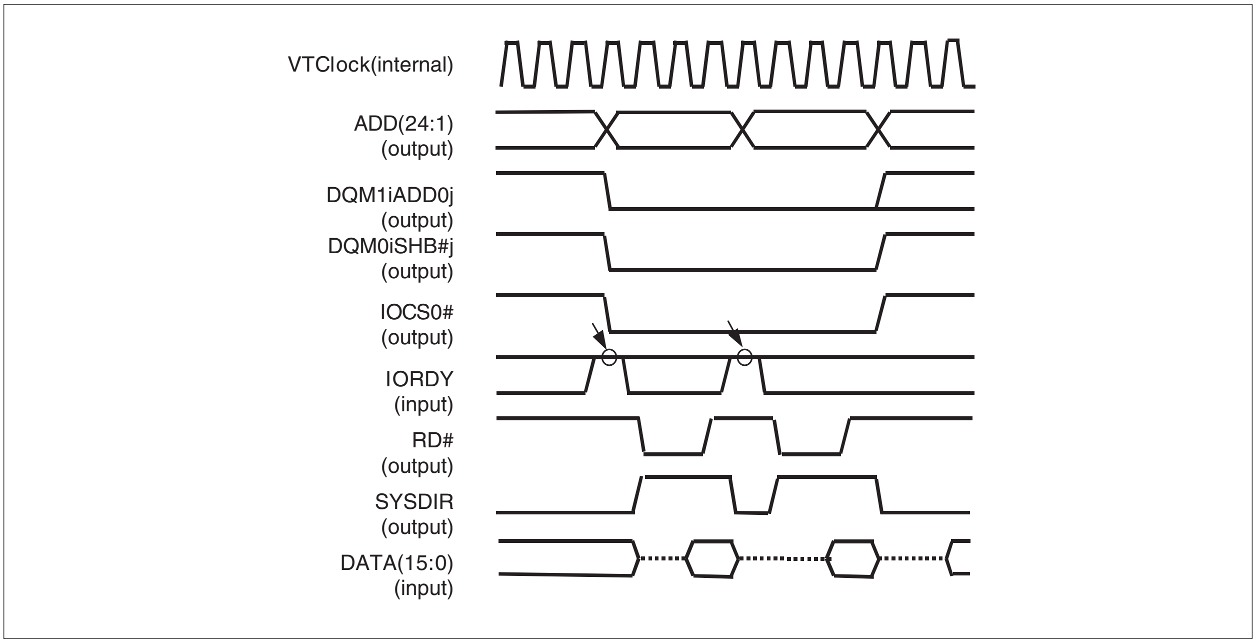
The SYSDIR signal becomes high level during a read access to ROM or the external I/O space. It changes as follows, depending on the access type:

- High impedance: During bus hold
- High level: While the following space is accessed for read[Note]
  External I/O space 1 and LCDSEL1 = “0”
  External I/O space 2 and LCDSEL0 = “0”
- Low level: While a space other than the above is accessed

Even while the CPU is in standby mode, the buffer is controlled as long as the system bus operates.

Note The SYSDIR signal becomes high level in synchronization with the falling edge of the RD# signal, and becomes low level one VTClock after the falling edge of the RD# signal (refer to **Figure 7-3**).

Figure 7-3. Control Timing of SYSDIR Signal (4-Byte Read Access to I/O Space via 16-Bit Bus) (io0_1_wait = 0000, io0_3_wait = 0000, io0_5_wait = 00, No Wait)



(4) Notes on connecting load mitigation buffer

- Disable control of the load mitigation buffer by the CKE signal.
  Although in the VR4121 it was recommended to control the load mitigation buffer by the CKE signal, the VR4131 disables this function.
- The target to be accessed through the load mitigation buffer is the external I/O space only.
  ROM and SDRAM must be configured so that the direct access can be made from the VR4131.

7.5 Bus Operation

This section explains the operations of the bus controlled by the BCU.

The operating clocks (VTClock and internal) of the BCU are shown in the timing chart of each bus operation. The frequency of VTClock is 26 to 100 MHz, depending on the setting of the CLKSEL(2:0) pins.

7.5.1 ROM access

The VR4131 supports the following three modes of ROM access. These modes can be selected by using the PAGEROM and ROMWEN bits of BCUCNTREG1.

- Normal ROM read mode (ROMWEN, PAGEROM = 00)
- PageROM read mode (ROMWEN, PAGEROM = 01)
- Flash memory mode (ROMWEN = 1)

(1) Normal ROM read mode

To use this mode, it is necessary that ROMWEN = 0 and PAGEROM = 0.

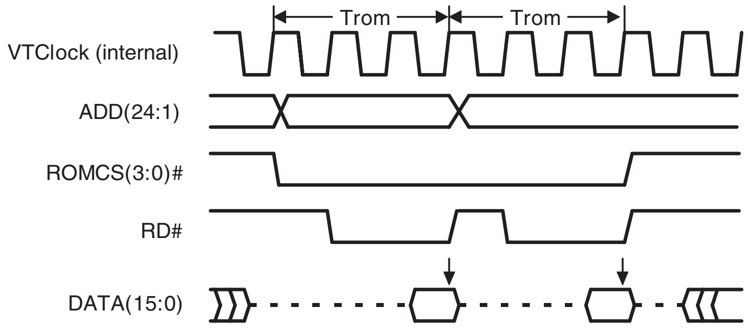
The access time of the normal ROM read cycle (Trom) can be selected as shown in Table 7-6, by setting the rom2_wait(2:0) bit of the ROMSPEEDREG register.

Figures 7-4 and 7-5 are the timing charts illustrating a read operation in 4-byte units when rom2_wait(2:0) are set to 000. If the ROM is accessed in units of 4 bytes or more, the Trom cycle continues by the necessary access size.

Table 7-6. Access Time in Normal ROM Read Mode

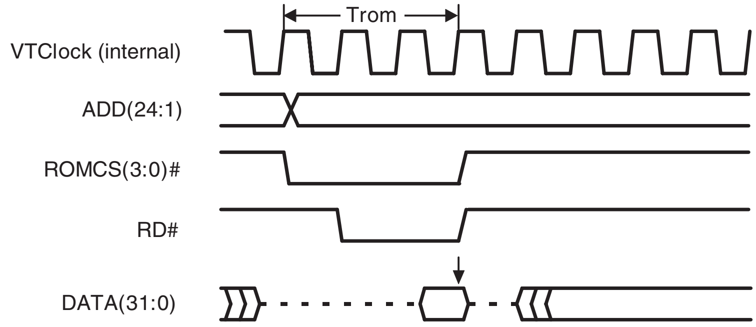
rom2_wait(2:0)	Trom (VTClock)
000	3
001	4
010	5
011	6
100	7
101	8
110	9
111	10

Figure 7-4. ROM 4-Byte Read (16-Bit Mode, rom2_wait(2:0) = 000)



Remark The dotted line indicates a high-impedance state.

Figure 7-5. ROM 4-Byte Read (32-Bit Mode, rom2_wait(2:0) = 000)



Remark The dotted line indicates a high-impedance state.

The DATA bus is sampled at the rising edge of the VTClock that follows the Trom cycle.
The following seven bus operations of the normal ROM can be executed:

1-byte read, 2-byte read, 3-byte read,
1-word read, 2-word read, 4-word read, 8-word read (1 word = 4 bytes)

(2) PageROM read mode

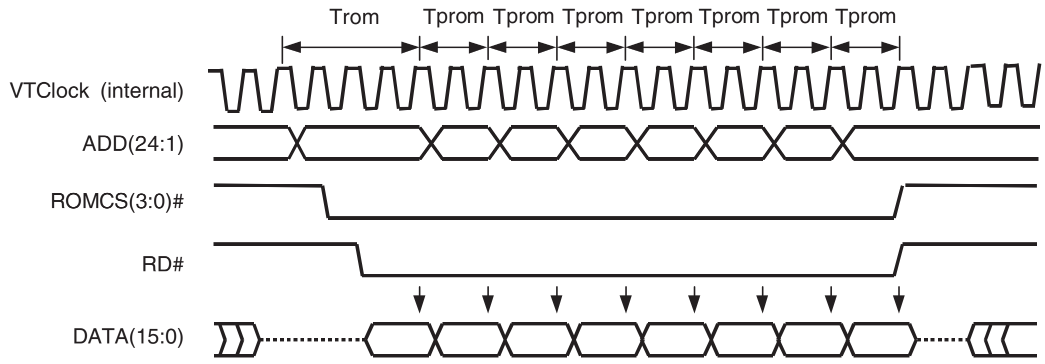
To use this mode, it is necessary that ROMWEN = 0 and PAGEROM = 1.
The access time of the PageROM read cycle (Tprom) can be selected, by setting the rom2_wait(2:0) bit and the rom4_wait(1:0) bit of the ROMSPEEDREG register.
Figures 7-6 and 7-7 show the timing charts illustrating a read operation in 4-word (16-byte) units when rom2_wait(2:0) = 001 and rom4_wait(1:0) = 00. The ROMCS# and RD# pins are kept low during the Trom cycle.

Table 7-7. Access Time in PageROM Read Mode

rom2_wait(2:0)	Trom (VTClock)
000	3
001	4
010	5
011	6
100	7
101	8
110	9
111	10

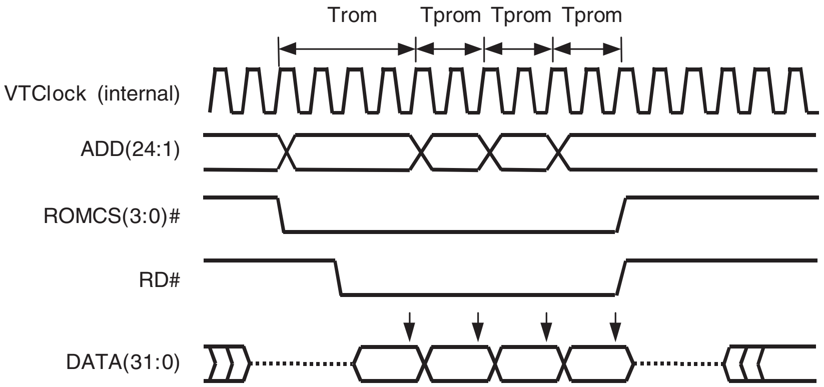
rom4_wait(1:0)	Tprom (VTClock)
00	2
01	3
10	4
11	5

Figure 7-6. PageROM 4-Word Read (16-Bit Mode, rom2_wait(2:0) = 001, rom4_wait(1:0) = 00)



Remark The dotted line indicates a high-impedance state.

Figure 7-7. PageROM 4-Word Read (32-Bit Mode, rom2_wait(2:0) = 001, rom4_wait(1:0) = 00)



Remark The dotted line indicates a high-impedance state.

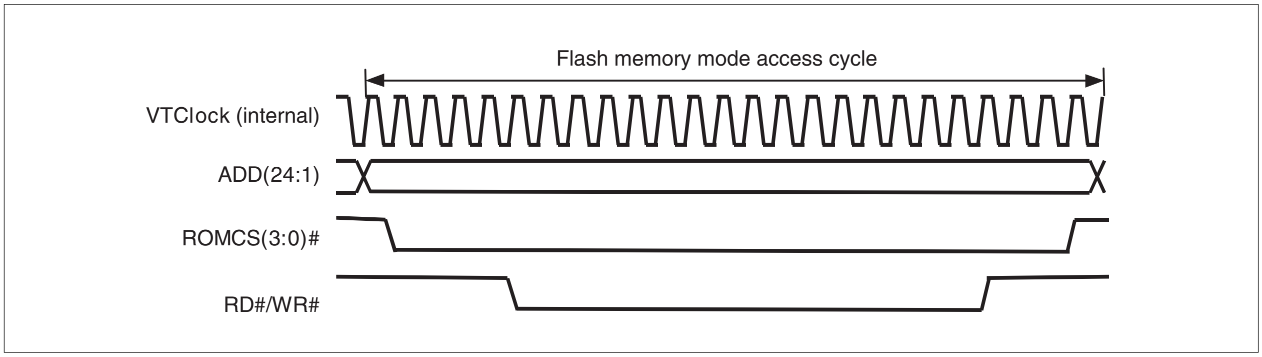
(3) Flash memory mode

To use this mode, it is necessary that ROMWEN = 1.

This mode is to satisfy writing the flash memory and the access sequence of the flash memory register. In this mode, the flash memory can be read.

The access time in this mode is fixed.

Figure 7-8. 2-Byte Access of Flash Memory Mode



7.5.2 I/O space access

The following table shows the relationship between DQM[0], DQM[1] and the data bus during 16-bit accessing of the VR4131 I/O space.

(1) Little-endian mode

DQM[0] (SHB#)	DQM[1] (ADD0)	Write		Read	
		DATA[15:8]	DATA[7:0]	DATA[15:8]	DATA[7:0]
0	0	○	○	Δ	Δ
1	0	×	○	–	Δ
0	1	○	×	Δ	–
1 Note	1 Note	× Note	× Note	– Note	– Note

Note No output with this combination.

Remark ○: Indicates that the data bus outputs valid data.
×: Indicates that the data bus outputs invalid data.
Δ: Indicates the data that the data bus samples.
–: Indicates the data that the data bus does not sample.

(2) Big-endian mode

DQM[0] (ADD0)	DQM[1] (SHB#)	Write		Read	
		DATA[15:8]	DATA[7:0]	DATA[15:8]	DATA[7:0]
0	0	○	○	Δ	Δ
1	0	○	×	Δ	–
0	1	×	○	–	Δ
1 Note	1 Note	× Note	× Note	– Note	– Note

Note No output with this combination.

Remark ○: Indicates that the data bus outputs valid data.
×: Indicates that the data bus outputs invalid data.
Δ: Indicates the data that the data bus samples.
–: Indicates the data that the data bus does not sample.

Figure 7-9. 4-Byte Access to I/O Space via 16-Bit Bus
(io0_1_wait = 0000, io0_3_wait = 0000, io0_5_wait = 00, No Wait)

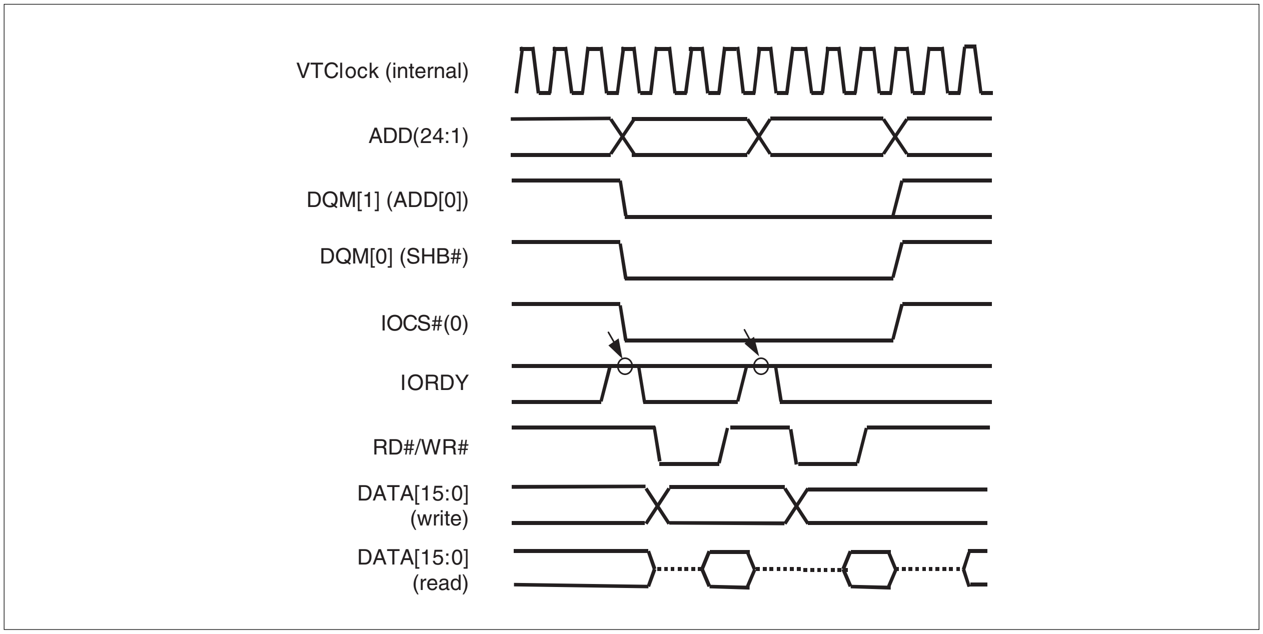


Figure 7-10. 4-Byte Access to I/O Space via 32-Bit Bus
(io0_1_wait = 0000, io0_3_wait = 0000, io0_5_wait = 00, No Wait)

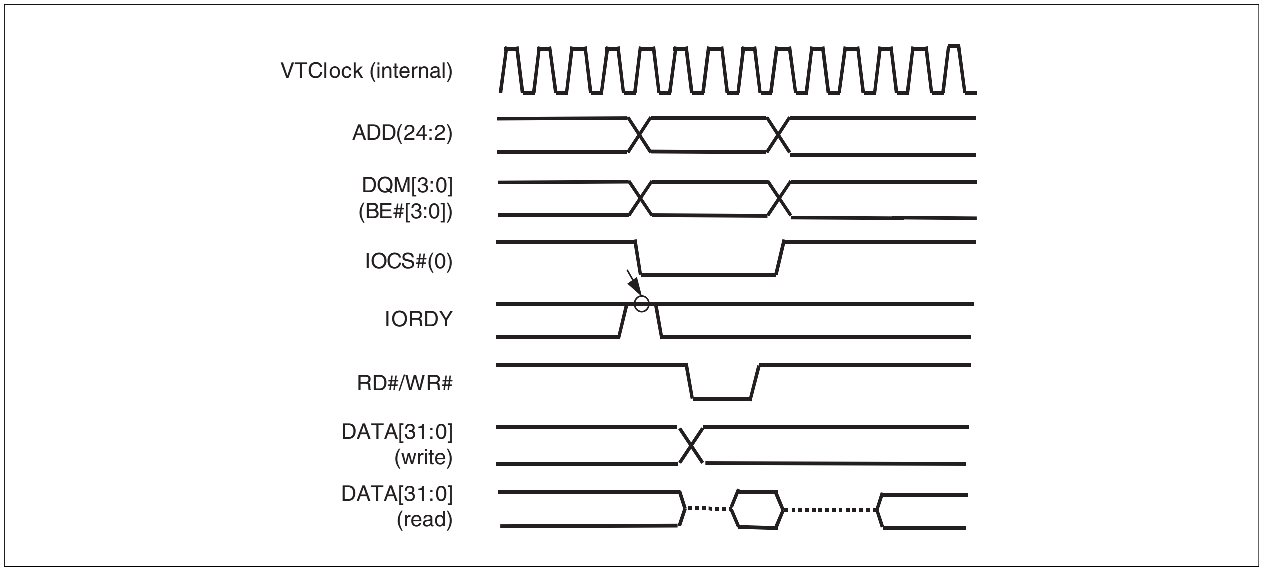


Figure 7-11. Setting Access Time by S/W for 4-Byte Access to I/O Space via 32-Bit Bus (io0_1_wait = 0010, io0_2_wait = 0010, io0_3_wait = 0010, io0_5_wait = 10, No Wait)

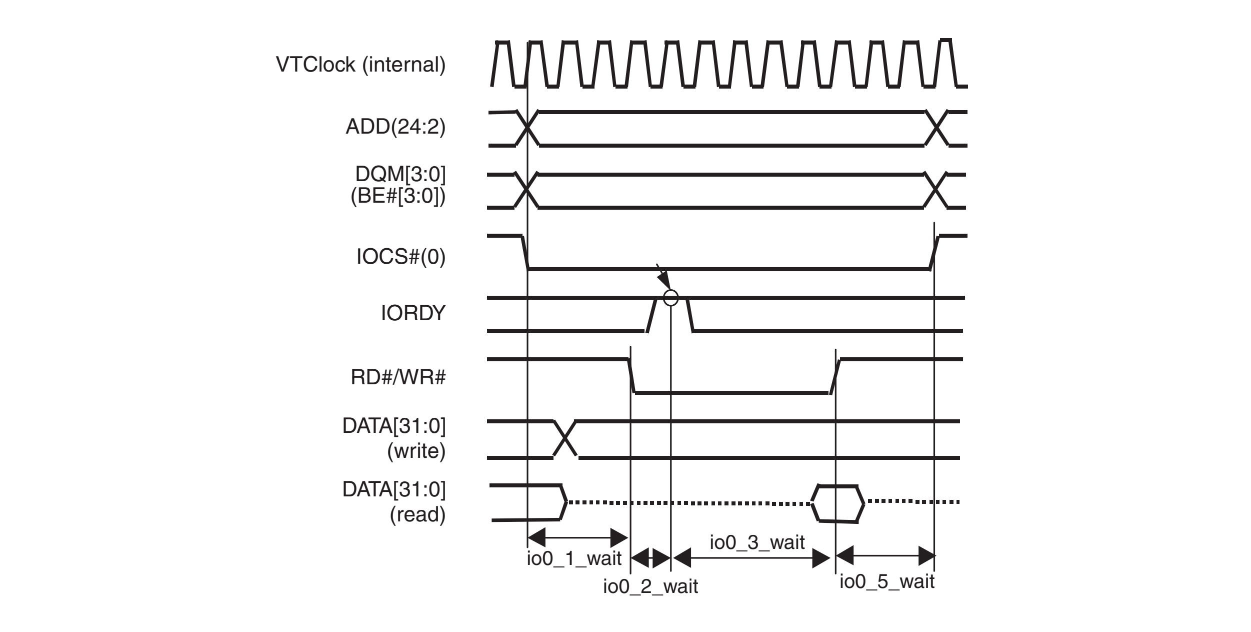
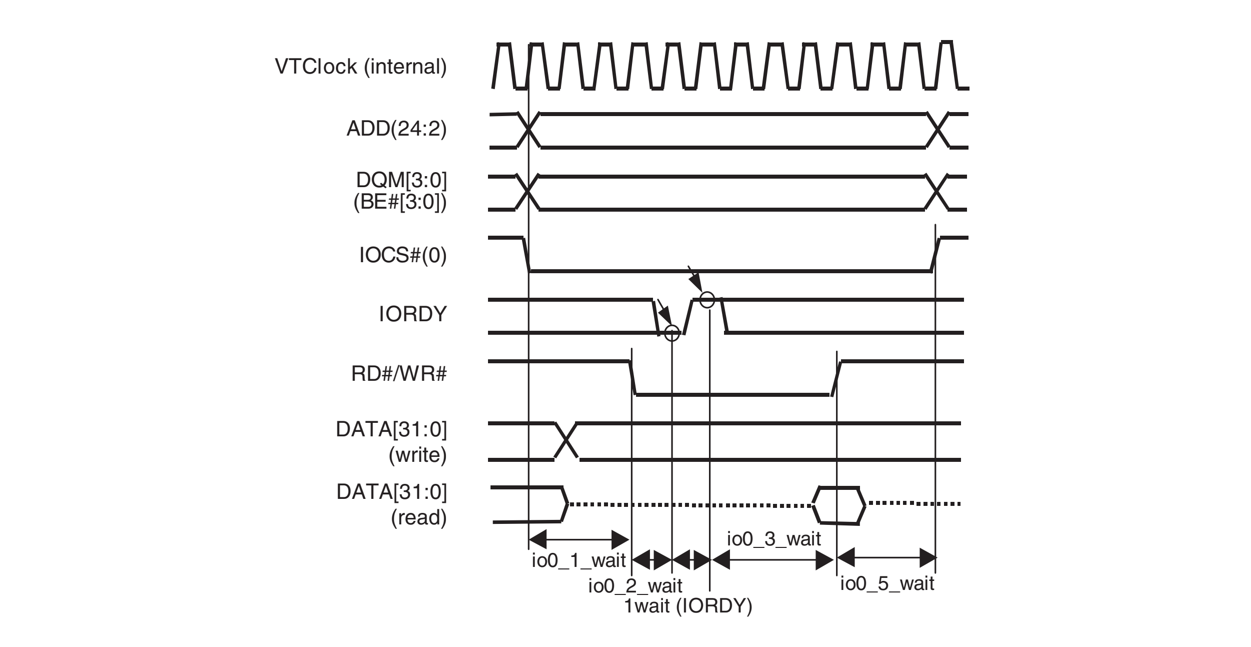


Figure 7-12. Setting Access Time by S/W for 4-Byte Access to I/O Space via 32-Bit Bus (io0_1_wait = 0010, io0_2_wait = 0010, io0_3_wait = 0010, io0_5_wait = 10, 1 Wait)



7.5.3 Bus hold

The bus hold function, which enables an external master in the I/O space to access SDRAM directly, is specified below.

(1) Control method

(a) HLDRQ# and HLDAK# pins

- HLDRQ#: Used for bus hold request
- HLDAK#: Used for bus hold acknowledge

(b) Switching functions by software

The bus hold function is enabled by setting the HLDEN bit of BCUCNTREG1 to 1.

(2) Supporting modes

The bus hold function supports the Fullspeed and Suspend modes. The Hibernate mode is not supported.

(3) Bus hold target pins

The bus hold function is applied to the following pins. For details, refer to **2.3.1 Pin statuses upon specific states**.

ADD(24:1), CAS, CKE(1:0), CS(1:0)#, CS(3:2)#/ROMCS(3:2)#, DATA(15:0),
DATA(31:16)/GPIO(31:16), DQM(3:0), SYSDIR/GPIO6, RAS, RD#, SCLK, SWR#,
HLDRQ#, HLDAK#, WR#

(4) Cautions when using bus hold function

Cautions to be observed when using the bus hold function are specified below.

- When the bus hold function is used, be sure to pull down (prohibit Nwire) the HLDAK# pin.

- The HLDAK# pin enters a high impedance state after reset (RTCRST# and RSTSW#) or at power down. Therefore, the external master must ignore the HLDAK# pin until the bus hold function is enabled by software.

- When the RSTSW# signal is input during a bus hold, the HLDAK# signal is forcibly made high level and the bus mastership is taken from the master by the VR4131. For a system using the bus hold function, it is recommended to generate a signal that is exclusive to the bus hold request signal inside the external bus master without inputting the RSTSW# signal directly to the VR4131, and input the signal to the VR4131 as the RSTSW# signal.

- When using the bus hold function in the Suspend mode, cancel a bus hold following the issuance of the self-refresh command on the external master side. The VR4131 cannot execute a refresh to SDRAM because VTClock is stopped.

CHAPTER 8 DMAAU (DMA ADDRESS UNIT)

8.1 General

The DMAAU controls the DMA addresses for the CSI, IrDA 4 Mbps communication module (hereafter called FIR), I/O space and RAM.

The DMA start address of each DMA channel can be specified to any physical addresses 0x0000 0000 to 0x01FF FFFE. The DMA space of FIR can be selected from either a 2 KB block that starts at the address which is generated by masking the lower 10 bits of the DMA start address with 0 or a 4 KB block that starts at the address which is generated by masking the lower 11 bits of the DMA start address with 0. The DMA space between the I/O space and RAM can be selected from 4 bytes to 265 KB.

Caution DMA operations are not guaranteed if an address overlaps with another DMA buffer.

After a DMA start address is set to the DMA base address register, the VR4131 performs DMA transfer using the registers of DMAAU as below.

8.1.1 DMA of CSI or FIR (when 2 KB block is selected)

(1) When the DMA start address is included in the first page of the DMA space

<1> The VR4131 starts a DMA transfer after writing the start address to the DMA address register.

<2> When the DMA transfer reaches the first page boundary, the VR4131 adds 1 KB to the contents of the DMA base address register, writes the value to the DMA address register, and continues the DMA transfer.

<3> When the DMA transfer reaches the second page boundary, the VR4131 writes the contents of the DMA base address register to the DMA address register and continues the DMA transfer.

<4> The VR4131 repeats <2> and <3> until all the data is transferred.

(2) When the DMA start address is included in the second page of the DMA space

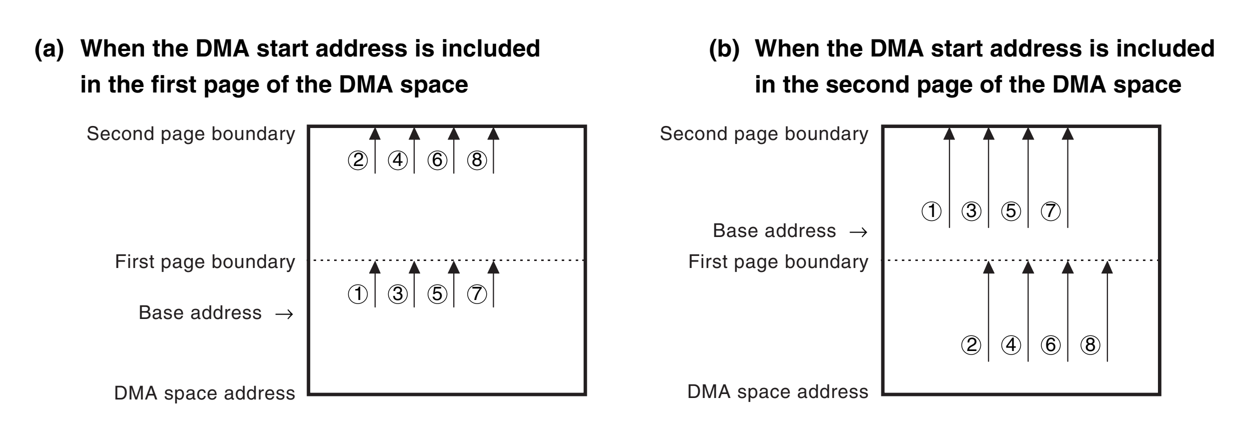
<1> The VR4131 starts a DMA transfer after writing the start address to the DMA address register.

<2> When the DMA transfer reaches the second page boundary, the VR4131 subtracts 1 KB from the contents of the DMA base address register, writes the value to the DMA address register, and continues the DMA transfer.

<3> When the DMA transfer reaches the first page boundary, the VR4131 writes the contents of the DMA base address register to the DMA address register and continues the DMA transfer.

<4> The VR4131 repeats <2> and <3> until all the data is transferred.

Figure 8-1. DMA Space Used in DMA Transfers



(a) When the DMA start address is included in the first page of the DMA space

(b) When the DMA start address is included in the second page of the DMA space

8.1.2 DMA between I/O space and RAM

<1> Write the number of transfer bytes in the BASSCNTLREG and BASSCNTHREG registers and set the values of the CURRENTCNTLREG and CURRENTCNTHREG registers to 0.

<2> Write the start address to the DMA address register and execute DMA transfer. At this time, the DMA address register is incremented from the transfer start address by the number of bytes that have been transferred.

8.1.3 Operation in big-endian mode

Even if the CPU is operating in big-endian mode, DMA transfer operates the same as in little endian mode except for FIR. DMA transfer of FIR is executed in big-endian format.

8.2 Register Set

The DMAAU registers are listed below.

Table 8-1. DMAAU Registers

Address	R/W	Register Symbols	Function
0x0F00 0020	R/W	CSIIBALREG	DMA base address lower register for receiving CSI
0x0F00 0022	R/W	CSIIBAHREG	DMA base address higher register for receiving CSI
0x0F00 0024	R/W	CSIIALREG	DMA address lower register for the receiving CSI
0x0F00 0026	R	CSIIAHREG	DMA address higher register for the receiving CSI
0x0F00 0028	R/W	CSIOBALREG	DMA base address lower register for transmitting CSI
0x0F00 002A	R/W	CSIOBAHREG	DMA base address higher register for transmitting CSI
0x0F00 002C	R/W	CSIOALREG	DMA address lower register for transmitting CSI
0x0F00 002E	R	CSIOAHREG	DMA address higher register for transmitting CSI
0x0F00 0030	R/W	FIRBALREG	DMA base address lower register for FIR
0x0F00 0032	R/W	FIRBAHREG	DMA base address higher register for FIR
0x0F00 0034	R/W	FIRALREG	DMA address lower register for FIR
0x0F00 0036	R	FIRAHREG	DMA address higher register for FIR
0x0F00 01E0	R/W	RAMBALREG	RAM base address lower address between I/O space and RAM
0x0F00 01E2	R/W	RAMBAHREG	RAM base address higher address between I/O space and RAM
0x0F00 01E4	R/W	RAMALREG	RAM address lower address between I/O space and RAM
0x0F00 01E6	R/W	RAMAHREG	RAM address higher address between I/O space and RAM
0x0F00 01E8	R/W	IOBALREG	I/O base address lower address between I/O space and RAM
0x0F00 01EA	R/W	IOBAHREG	I/O base address higher address between I/O space and RAM
0x0F00 01EC	R/W	IOALREG	I/O address lower address between I/O space and RAM
0x0F00 01EE	R/W	IOAHREG	I/O address higher address between I/O space and RAM

These registers are described in detail below.

8.2.1 DMA base address register for receiving CSI

(1) CSIIBALREG (0x0F00 0020)

Bit	15	14	13	12	11	10	9	8
Name	CSIIBA15	CSIIBA14	CSIIBA13	CSIIBA12	CSIIBA11	CSIIBA10	CSIIBA9	CSIIBA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	1	1	1	1	1	0	0	0
After reset	1	1	1	1	1	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	CSIIBA7	CSIIBA6	CSIIBA5	CSIIBA4	CSIIBA3	CSIIBA2	CSIIBA1	CSIIBA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:2	CSIIBA(15:2)	Sets the DMA base address for receiving CSI
1:0	CSIIBA(1:0)	0 is returned after a read.

(2) CSIIBAHREG (0x0F00 0022)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	CSIIBA26	CSIIBA25	CSIIBA24
R/W	R	R	R	R	R	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	1
After reset	0	0	0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
Name	CSIIBA23	CSIIBA22	CSIIBA21	CSIIBA20	CSIIBA19	CSIIBA18	CSIIBA17	CSIIBA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	1	1	1	1	1	1	1	1
After reset	1	1	1	1	1	1	1	1

Bit	Name	Function
15:11	RFU	Reserved. Write 0. 0 is returned after a read.
10:0	CSIIBA(26:16)	Bits 26:16 of the DMA base address for receiving CSI.

CSIIBALREG and CSIIBAHREG set the base addresses of the DMA channel for receiving CSI. The addresses set to these registers become DMA transfer start addresses.

The DMA channel for receiving CSI is retained in SDRAM as a 2 KB buffer that starts at the address which is generated by masking the lower 10 bits of the address set by these registers.

8.2.2 DMA address register for receiving CSI

(1) CSIIALREG (0x0F00 0024)

Bit	15	14	13	12	11	10	9	8
Name	CSIIA15	CSIIA14	CSIIA13	CSIIA12	CSIIA11	CSIIA10	CSIIA9	CSIIA8
R/W	R	R	R	R	R	R/W	R/W	R/W
RTCRST	1	1	1	1	1	0	0	0
After reset	1	1	1	1	1	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	CSIIA7	CSIIA6	CSIIA5	CSIIA4	CSIIA3	CSIIA2	CSIIA1	CSIIA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:2	CSIIA(15:2)	Next DMA address to be accessed for CSI input channel
1:0	CSIIA(1:0)	0 is returned after a read

(2) CSIIAHREG (0x0F00 0026)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	CSIIA26	CSIIA25	CSIIA24
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	1
After reset	0	0	0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
Name	CSIIA23	CSIIA22	CSIIA21	CSIIA20	CSIIA19	CSIIA18	CSIIA17	CSIIA16
R/W	R	R	R	R	R	R	R	R
RTCRST	1	1	1	1	1	1	1	1
After reset	1	1	1	1	1	1	1	1

Bit	Name	Function
15:11	RFU	Reserved. Write 0. 0 is returned after a read
10:0	CSIIA(26:16)	Bits 26:16 of the next DMA address to be accessed for the CSI input channel

8.2.3 DMA base address register for transmitting CSI

(1) CSIOBALREG (0x0F00 0028)

Bit	15	14	13	12	11	10	9	8
Name	CSIOBA15	CSIOBA14	CSIOBA13	CSIOBA12	CSIOBA11	CSIOBA10	CSIOBA9	CSIOBA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	1	1	1	1	1	0	0	0
After reset	1	1	1	1	1	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	CSIOBA7	CSIOBA6	CSIOBA5	CSIOBA4	CSIOBA3	CSIOBA2	CSIOBA1	CSIOBA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:0	CSIOBA(15:0)	Bits 15:0 of the DMA base address for transmitting CSI

(2) CSIOBAHREG (0x0F00 002A)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	CSIOBA26	CSIOBA25	CSIOBA24
R/W	R	R	R	R	R	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	1
After reset	0	0	0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
Name	CSIOBA23	CSIOBA22	CSIOBA21	CSIOBA20	CSIOBA19	CSIOBA18	CSIOBA17	CSIOBA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	1	1	1	1	1	1	1	1
After reset	1	1	1	1	1	1	1	1

Bit	Name	Function
15:11	RFU	Reserved. Write 0. 0 is returned after a read.
10:0	CSIOBA(26:16)	Bits 26:16 of the DMA base address for transmitting CSI.

CSIOBALREG and CSIOBAHREG set the base addresses of the DMA channel for transmitting CSI. The addresses set to these registers become DMA transfer start addresses.

The DMA channel for transmitting CSI is retained in SDRAM as a 2 KB buffer that starts at the address which is generated by masking the lower 10 bits of the address set by these resisters to 0.

8.2.4 DMA address register for transmitting CSI

(1) CSIOALREG (0x0F00 002C)

Bit	15	14	13	12	11	10	9	8
Name	CSIOA15	CSIOA14	CSIOA13	CSIOA12	CSIOA11	CSIOA10	CSIOA9	CSIOA8
R/W	R	R	R	R	R	R/W	R/W	R/W
RTCRST	1	1	1	1	1	0	0	0
After reset	1	1	1	1	1	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	CSIOA7	CSIOA6	CSIOA5	CSIOA4	CSIOA3	CSIOA2	CSIOA1	CSIOA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:2	CSIOA(15:2)	Next DMA address to be accessed for CSI transmitting channel
1:0	CSIOA(1:0)	0 is returned after a read

(2) CSIOAHREG (0x0F00 002E)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	CSIOA26	CSIOA25	CSIOA24
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	1
After reset	0	0	0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
Name	CSIOA23	CSIOA22	CSIOA21	CSIOA20	CSIOA19	CSIOA18	CSIOA17	CSIOA16
R/W	R	R	R	R	R	R	R	R
RTCRST	1	1	1	1	1	1	1	1
After reset	1	1	1	1	1	1	1	1

Bit	Name	Function
15:11	RFU	Reserved. Write 0. 0 is returned after a read.
10:0	CSIOA(26:16)	Bits 26:16 of the next DMA address to be accessed for CSI transmitting channel

8.2.5 DMA base address register for FIR

(1) FIRBALREG (0x0F00 0030)

Bit	15	14	13	12	11	10	9	8
Name	FIRBA15	FIRBA14	FIRBA13	FIRBA12	FIRBA11	FIRBA10	FIRBA9	FIRBA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	1	1	1	1	1	0	0	0
After reset	1	1	1	1	1	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	FIRBA7	FIRBA6	FIRBA5	FIRBA4	FIRBA3	FIRBA2	FIRBA1	FIRBA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:0	FIRBA(15:0)	Sets the DMA base address for FIR

(2) FIRBAHREG (0x0F00 0032)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	FIRBA26	FIRBA25	FIRBA24
R/W	R	R	R	R	R	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	1
After reset	0	0	0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
Name	FIRBA23	FIRBA22	FIRBA21	FIRBA20	FIRBA19	FIRBA18	FIRBA17	FIRBA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	1	1	1	1	1	1	1	1
After reset	1	1	1	1	1	1	1	1

Bit	Name	Function
15:11	RFU	Reserved. Write 0. 0 is returned after a read.
10:0	FIRBA(26:16)	Sets the DMA base address for FIR.

FIRBALREG and FIRBAHREG set the base addresses of the DMA channel for FIR.

The addresses set to these registers become DMA transfer start addresses.

When the FIREX bit of CONTROLREG is 0, the FIR DMA channel is retained in SDRAM as a 2 KB buffer that starts at the address that is generated by masking the lower 10 bits of the address set by these resisters with 0.

When the FIREX bit of CONTROLREG is 1, the FIR DMA channel is retained in SDRAM as a 2 KB buffer that starts at the address that is generated by masking the lower 11 bits of the address set by these registers with 0.

8.2.6 FIR DMA address registers

(1) FIRALREG (0x0F00 0034)

Bit	15	14	13	12	11	10	9	8
Name	FIRA15	FIRA14	FIRA13	FIRA12	FIRA11	FIRA10	FIRA9	FIRA8
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	1	1	1	1	1	0	0	0
After reset	1	1	1	1	1	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	FIRA7	FIRA6	FIRA5	FIRA4	FIRA3	FIRA2	FIRA1	FIRA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:0	FIRA(15:0)	Next DMA address to be accessed for FIR channel

(2) FIRAHREG (0x0F00 0036)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	FIRA26	FIRA25	FIRA24
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	1
After reset	0	0	0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
Name	FIRA23	FIRA22	FIRA21	FIRA20	FIRA19	FIRA18	FIRA17	FIRA16
R/W	R	R	R	R	R	R	R	R
RTCRST	1	1	1	1	1	1	1	1
After reset	1	1	1	1	1	1	1	1

Bit	Name	Function
15:11	RFU	Reserved. Write 0. 0 is returned after a read.
10:0	FIRA(26:16)	Next DMA address to be accessed for FIR channel

8.2.7 DMA base address register for RAM space

(1) RAMBALREG (0x0F00 01E0)

Bit	15	14	13	12	11	10	9	8
Name	RAMBA15	RAMBA14	RAMBA13	RAMBA12	RAMBA11	RAMBA10	RAMBA9	RAMBA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	1	1	1	1	1	0	0	0
After reset	1	1	1	1	1	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RAMBA7	RAMBA6	RAMBA5	RAMBA4	RAMBA3	RAMBA2	RFU	RFU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:2	RAMBA(15:2)	Sets the RAM base address for the DMA transfer between the I/O space and RAM space.
1:0	RFU	Reserved. Write 0. 0 is returned after a read.

(2) RAMBAHREG (0x0F00 01E2)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RAMBA26	RAMBA25	RAMBA24
R/W	R	R	R	R	R	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	1
After reset	0	0	0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
Name	RAMBA23	RAMBA22	RAMBA21	RAMBA20	RAMBA19	RAMBA18	RAMBA17	RAMBA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	1	1	1	1	1	1	1	1
After reset	1	1	1	1	1	1	1	1

Bit	Name	Function
15:11	RFU	Reserved. Write 0. 0 is returned after a read.
10:0	RAMBA(26:16)	Sets the RAM base address for the DMA transfer between the I/O space and RAM space

8.2.8 DMA address register for RAM space

(1) RAMALREG (0x0F00 01E4)

Bit	15	14	13	12	11	10	9	8
Name	RAMA15	RAMA14	RAMA13	RAMA12	RAMA11	RAMA10	RAMA9	RAMA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	1	1	1	1	1	0	0	0
After reset	1	1	1	1	1	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RAMA7	RAMA6	RAMA5	RAMA4	RAMA3	RAMA2	RFU	RFU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:2	RAMA(15:2)	Next DMA address to be accessed for RAM of the DMA transfer between the I/O space and RAM
1:0	RFU	Reserved. Write 0. 0 is returned after a read.

(2) RAMAHREG (0x0F00 01E6)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RAMA26	RAMA25	RAMA24
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	1
After reset	0	0	0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
Name	RAMA23	RAMA22	RAMA21	RAMA20	RAMA19	RAMA18	RAMA17	RAMA16
R/W	R	R	R	R	R	R	R/W	R/W
RTCRST	1	1	1	1	1	1	1	1
After reset	1	1	1	1	1	1	1	1

Bit	Name	Function
15:11	RFU	Reserved. Write 0. 0 is returned after a read.
10:0	RAMA(26:16)	Next DMA address to be accessed for RAM of the DMA transfer between the I/O space and RAM

8.2.9 DMA base address register for I/O space

(1) IOBALREG (0x0F00 01E8)

Bit	15	14	13	12	11	10	9	8
Name	IOBA15	IOBA14	IOBA13	IOBA12	IOBA11	IOBA10	IOBA9	IOBA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	IOBA7	IOBA6	IOBA5	IOBA4	IOBA3	IOBA2	RFU	RFU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:2	IOBA(15:2)	Sets the I/O space base address for the DMA transfer between the I/O space and RAM
1:0	RFU	Reserved. Write 0. 0 is returned after a read.

(2) IOBAHREG (0x0F00 01EA)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	IOBA26	IOBA25	IOBA24
R/W	R	R	R	R	R	R/W	R/W	R/W
RTCRST	0	0	0	0	1	0	1	0
After reset	0	0	0	0	1	0	1	0

Bit	7	6	5	4	3	2	1	0
Name	IOBA23	IOBA22	IOBA21	IOBA20	IOBA19	IOBA18	IOBA17	IOBA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:12	RFU	Reserved. Write 0. 0 is returned after a read.
11	RFU	Reserved. Write 1. 1 is returned after a read.
10:0	IOBA(26:16)	Sets the I/O space base address for the DMA transfer between the I/O space and RAM

8.2.10 DMA address register for I/O space

(1) IOALREG (0x0F00 01EC)

Bit	15	14	13	12	11	10	9	8
Name	IOA15	IOA14	IOA13	IOA12	IOA11	IOA10	IOA9	IOA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	IOA7	IOA6	IOA5	IOA4	IOA3	IOA2	RFU	RFU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:2	IOA(15:2)	Next address to be accessed for the I/O space of the DMA transfer between the I/O space and RAM
1:0	RFU	Reserved. Write 0. 0 is returned after a read

(2) IOAHREG (0x0F00 01EE)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	IOA26	IOA25	IOA24
R/W	R	R	R	R	R	R/W	R/W	R/W
RTCRST	0	0	0	0	1	0	1	0
After reset	0	0	0	0	1	0	1	0

Bit	7	6	5	4	3	2	1	0
Name	IOA23	IOA22	IOA21	IOA20	IOA19	IOA18	IOA17	IOA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	1	1	1	1	1	1	1	1
After reset	1	1	1	1	1	1	1	1

Bit	Name	Function
15:12	RFU	Reserved. Write 0. 0 is returned after a read.
11	RFU	Reserved. Write 1. 1 is returned after a read.
10:0	IOA(26:16)	Next address to be accessed for the I/O space of the DMA transfer between the I/O space and RAM

CHAPTER 9 DCU (DMA CONTROL UNIT)

9.1 General

The DCU is used for DMA control. It controls DMA requests from each on-chip peripheral I/O unit and enables/disables the DMA operation.

9.2 DMA Priority Control

When a conflict occurs between DMA requests sent from on-chip peripheral I/O units, the DCU handles DMA requests in the following priority order. This priority order can be changed using DMAABITREG.

Table 9-1. DMA Default Priority Levels

Priority Order	Type of DMA Operation
High	FIR transmission/reception
↑	CSI reception
↓	CSI transmission
Low	I/O space and RAM space

9.3 Register Set

The DCU register set is described in the following table.

Table 9-2. DCU Registers

Address	R/W	Register Symbol	Function
0x0F00 0040	R/W	DMARSTREG	DMA reset register
0x0F00 0042	R	DMAIDLEREG	DMA sequencer status register
0x0F00 0044	R/W	DMASENREG	DMA sequencer enable register
0x0F00 0046	R/W	DMAMSKREG	DMA mask register
0x0F00 0048	R/W	DMAREQREG	DMA request register
0x0F00 004A	R/W	TDREG	Transfer direction setting register
0x0F00 004C	R/W	DMAABITREG	DMA arbitration protocol selection register
0x0F00 004E	R/W	CONTROLREG	DMA control register
0x0F00 0050	R/W	BASSCNTLREG	DMA transfer byte size lower register
0x0F00 0052	R/W	BASSCNTHREG	DMA transfer byte size higher register
0x0F00 0054	R/W	CURRENTCNTLREG	DMA remaining transfer byte size lower register
0x0F00 0056	R/W	CURRENTCNTHREG	DMA remaining transfer byte size higher register
0x0F00 0058	R/W	TCINTREG	Terminal count interrupt request

These registers are described in detail below.

9.3.1 DMARSTREG (0x0F00 0040)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	DMARST
R/W	R	R	R	R	R	R	R	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:1	RFU	Reserved. Write 0. 0 is returned after a read.
0	DMARST	Resets DMA controller. 0: Reset 1: Normal

This register is used to reset the DMA controller.

9.3.2 DMAIDLEREG (0x0F00 0042)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	DMAISTAT
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	1
After reset	0	0	0	0	0	0	0	1

Bit	Name	Function
15:1	RFU	Reserved. Write 0. 0 is returned after a read.
0	DMAISTAT	Displays DMA sequencer status. 1: Idle status 0: Sequencer busy

This register is used to display the DMA sequencer status.

9.3.3 DMASENREG (0x0F00 0044)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	DMASEN
R/W	R	R	R	R	R	R	R	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:1	RFU	Reserved. Write 0. 0 is returned after a read.
0	DMASEN	Enables DMA sequencer. 1: Enable 0: Disable

This register enables or disables the operation of the DMA sequencer (CSI transmission, CSI reception, FIR transmission/reception, I/O space ⇔ RAM space).

9.3.4 DMAMSKREG (0x0F00 0046)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	DMAMSK IOR	DMAMSK COUT	DMAMSK CIN	DMAMSK FOUT
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:4	RFU	Reserved. Write 0. 0 is returned after a read.
3	DMAMSKIOR	Enables the I/O space and RAM transfer. 1: Enable 0: Disable
2	DMAMSKCOUT	Enables the CSI transmission DMA transfer. 1: Enable 0: Disable
1	DMAMSKCIN	Enables the CSI reception DMA transfer. 1: Enable 0: Disable
0	DMAMSKFOUT	Enables the FIR transmission DMA transfer. 1: Enable 0: Disable

This register is used to set each DMA channel transfer to enable/disable.

9.3.5 DMAREQREG (0x0F00 0048)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	DRQIOR	DRQCOUT	DRQCIN	DRQFOUT
R/W	R	R	R	R	R/W	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:4	RFU	Reserved. Write 0. 0 is returned after a read.
3	DRQIOR	Requests the I/O space and RAM transfer. 1: Request 0: Halt
2	DRQCOUT	CSI transmission DMA transfer 1: Available 0: Not available
1	DRQCIN	CSI reception DMA transfer 1: Available 0: Not available
0	DRQFOUT	FIR transmission DMA transfer 1: Available 0: Not available

This register is used to indicate whether or not there are any DMA transfer requests.

The DMA start to the I/O space and RAM space is performed by software. Setting the DRQIOR bit to 1 starts DMA transfer. This operation is terminated when the transfer of the set data size is completed (TC) and auto-initialization is set to disable. In this case, the DRQIOR bit becomes 0 automatically. Setting the DRQIOR bit to 0 terminates DMA transfer forcibly after the completion of the data transfer being executed.

9.3.6 TDREG (0x0F00 004A)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	IORAM	FIR
R/W	R	R	R	R	R	R	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:2	RFU	Reserved. Write 0. 0 is returned after a read.
1	IORAM	Transfer direction of the DMA channel between the I/O space and RAM. 1: External I/O → memory 0: Memory → external I/O
0	FIR	Transfer direction of the DMA channel for FIR transmission 1: Internal I/O → memory 0: Memory → internal I/O

This register is used to set the data transfer direction of DMA channel.

9.3.7 DMAABITREG (0x0F00 004C)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	DMAPRI3	DMAPRI2	DMAPRI1	DMAPRI0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:4	RFU	Reserved. Write 0. 0 is returned after a read.
3:0	DMAPRI(3:0)	Selects the DMA arbitration protocol. 1000: Priority to I/O space and RAM space transfer 0100: Priority to CSI transmission 0010: Priority to CSI reception 0001: FIR priority Others: Fixed priority order 1) FIR 2) CSI reception 3) CSI transmission 4) DMA to the I/O space

This register is used to set the DMA arbitration protocol.

9.3.8 CONTROLREG (0x0F00 004E)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	FIREX	DMABLKS1	DMABLKS0	AUTOINIT
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:4	RFU	Reserved. Write 0. 0 is returned after a read.
3	FIREX	FIRDMA space extended bit 1: 2 KB 2 pages 0: 1 KB 2 pages
2:1	DMABLKS(1:0)	Block size during DMA transfer 11: RFU 10: 32 bytes 01: 16 bytes 00: 4 bytes
0	AUTOINIT	Enables the automatic initialization. 1: Enable 0: Disable

This register controls the DMA.

When auto-initialization is enabled, the data of RAMALREG/RAMAHREG is automatically set from RAMBALREG/RAMBAHREG and DMA transfer is performed continuously at the completion of the DMA transfer (TC) between the I/O space and RAM space.

9.3.9 BASSCNTLREG (0x0F00 0050)

Bit	15	14	13	12	11	10	9	8
Name	DMABS15	DMABS14	DMABS13	DMABS12	DMABS11	DMABS10	DMABS9	DMABS8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	DMABS7	DMABS6	DMABS5	DMABS4	DMABS3	DMABS2	RFU	RFU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:2	DMABS(15:2)	Sets the number of transfer bytes for the I/O space and RAM transfer.
1:0	RFU	Reserved. Write 0. 0 is returned after a read.

This register is used to set the number of DMA transfer bytes for the I/O space and RAM space.

9.3.10 BASSCNTHREG (0x0F00 0052)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	DMABS17	DMABS16
R/W	R	R	R	R	R	R	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:2	RFU	Reserved. Write 0. 0 is returned after a read.
1:0	DMABS(17:16)	Sets the number of transfer bytes for the I/O space and RAM transfer.

Caution The value must be set in block units during transfer.

9.3.11 CURRENTCNTLREG (0x0F00 0054)

Bit	15	14	13	12	11	10	9	8
Name	DMA RBS15	DMA RBS14	DMA RBS13	DMA RBS12	DMA RBS11	DMA RBS10	DMA RBS9	DMA RBS8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	DMA RBS7	DMA RBS6	DMA RBS5	DMA RBS4	DMA RBS3	DMA RBS2	RFU	RFU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:2	DMARBS(15:2)	Sets the number of transfer blocks for the remaining of the I/O space and RAM transfer.
1:0	RFU	Reserved. Write 0. 0 is returned after a read.

9.3.12 CURRENTCNTHREG (0x0F00 0056)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	DMA RBS17	DMA RBS16
R/W	R	R	R	R	R	R	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:2	RFU	Reserved. Write 0. 0 is returned after a read.
1:0	DMARBS(17:16)	Sets the number of transfer blocks for the remaining of the I/O space and RAM transfer.

9.3.13 TCINTREG (0x0F00 0058)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	TCINT
R/W	R	R	R	R	R	R	R	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:1	RFU	Reserved. Write 0. 0 is returned after a read.
0	TCINT	Interrupt request for the terminal count 1: Available 0: Not available

This register is used to indicate the interrupt request for the terminal count.

This interrupt request is transmitted as BCUINT to the ICU.

CHAPTER 10 CMU (CLOCK MASK UNIT)

10.1 General

When internal clocks are supplied from the CPU to each unit, the CMU controls whether the clocks are masked or not. This masking method enables power consumption to be reduced in units that are not used.

The units for which CMU is used are the SIU, DSIU, FIR, CSI, and PCIU units.

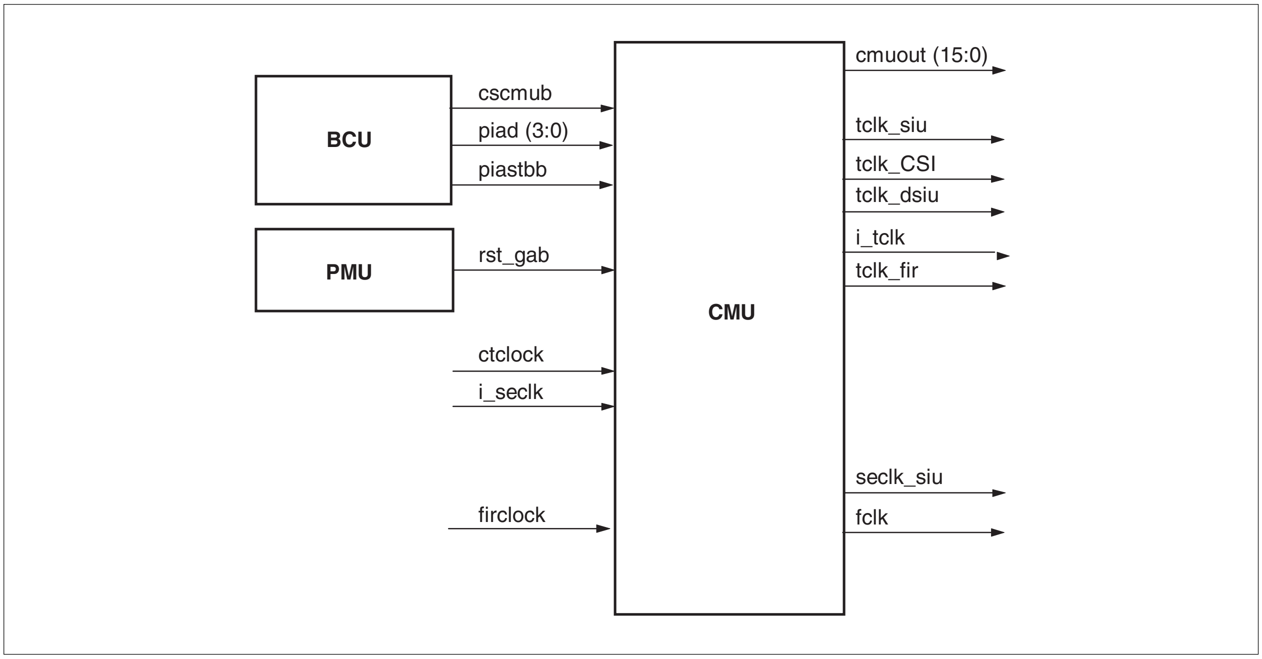
The basic functions are described below.

- Control of TClock supplied to SIU, DSIU, FIR, and CSI
- Control of internal clock (18.432 MHz) supplied to SIU and DSIU
- Control of PCLK supplied to PCI interface
- Control of VTClock supplied to PCIU
- Control of internal clock (48 MHz) supplied to FIR
- Control of Intrclk divide

All the clock supplies are masked (0) in the initial state of the CMU. No clock is supplied unless the CPU writes 1 to the register.

The following figure shows the CMU-related blocks.

Figure 10-1. CMU Peripheral Block Diagram



10.2 Register Set

The CMU register is shown below.

Table 10-1. CMU Register

Address	R/W	Register Symbol	Function
0x0F00 0060	R/W	CMUCLKMSK	CMU clock mask register

This register is described in detail below.

10.2.1 CMUCLKMSK (0x0F00 0060)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	MSKPCIU	MSKSCSI	MSKDSIU	MSKFFIR	RFU	MSKSSIU
R/W	R	R	R/W	R/W	R/W	R/W	R	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	MSKPCIU	MSKCSI	RFU	MSKFIR	RFU	RFU	MSKSIU	RFU
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:14	RFU	Reserved. Write 0 to these bits. 0 is returned after a read.
13	MSKPCIU	Control of PCLK supplied to PCI interface 1: Supply 0: Mask
12	MSKSCSI	Control of 18.432 MHz clock supplied to CSI 1: Supply 0: Mask
11	MSKDSIU	Control of 18.432 MHz clock supplied to DSIU 1: Supply 0: Mask
10	MSKFFIR	Control of 48 MHz clock supplied to CSI 1: Supply 0: Mask
9	RFU	Reserved. Write 0. 0 is returned after a read.
8	MSKSSIU	Control of 18.432 MHz clock supplied to SIU 1: Supply 0: Mask
7	MSKPCIU	Control of VTClock supplied to PCIU 1: Supply 0: Mask
6	MSKCSI	Control of TClock supplied to CSI 1: Supply 0: Mask
5	RFU	Reserved. Write 0. 0 is returned after a read.
4	MSKFIR	Control o TClock supplied to FIR 1: Supply 0: Mask
3:2	RFU	Reserved. Write 0. 0 is returned after a read.
1	MSKSIU	Control of TClock supplied to SIU and DSIU 1: Supply 0: Mask
0	RFU	Reserved. Write 0. 0 is returned after a read.

This register is used to mask the clocks that are supplied to the SIU, DSIU, FIR, CSI, and PCIU units.

The MSKPCIU bit must be set to 1 before accessing the PCIU register.

CHAPTER 11 ICU (INTERRUPT CONTROL UNIT)

11.1 General

The ICU collects interrupt requests from the various on-chip peripheral units and generates interrupt request signals (Int0, Int1, Int2, and NMI) to the CPU core.

The functions of the ICU's internal blocks are briefly described below.

- ADDECICU … Decodes read/write addresses from the CPU that are used for ICU registers.
- REGICU … This includes a register for interrupt masking.
  The initial value is 0 (mask). No interrupt signal is supplied to CPU core unless the CPU writes (1) to this register.
- OUTICU … This block collects interrupt requests after masking them, and generates an interrupt request signal to output to the CPU core.
  In Suspend mode, it also controls the masking of interrupt requests, output of the int_all signal that is the general interrupt source signal, and the memdrv output timing signal that is used when returning from Suspend mode.

To report interrupt requests to the CPU core, the following signals are used:

NMI: For battint_intr
Switching between NMI and Int0 is enabled according to a register's settings.
Because NMI's interrupt masking cannot be controlled by means of software, switch to Int0 to mask battint_Intr.

Int2: For rtc_long2_intr

Int1: For rtc_long1_intr
The dedicated signals should be used because the interval timer requires more responsiveness than other interrupt sources do.

Int0: All other interrupts

For details of the interrupt sources, refer to **11.2 Register Set**.

How an interrupt request is reported to the CPU core is shown below.

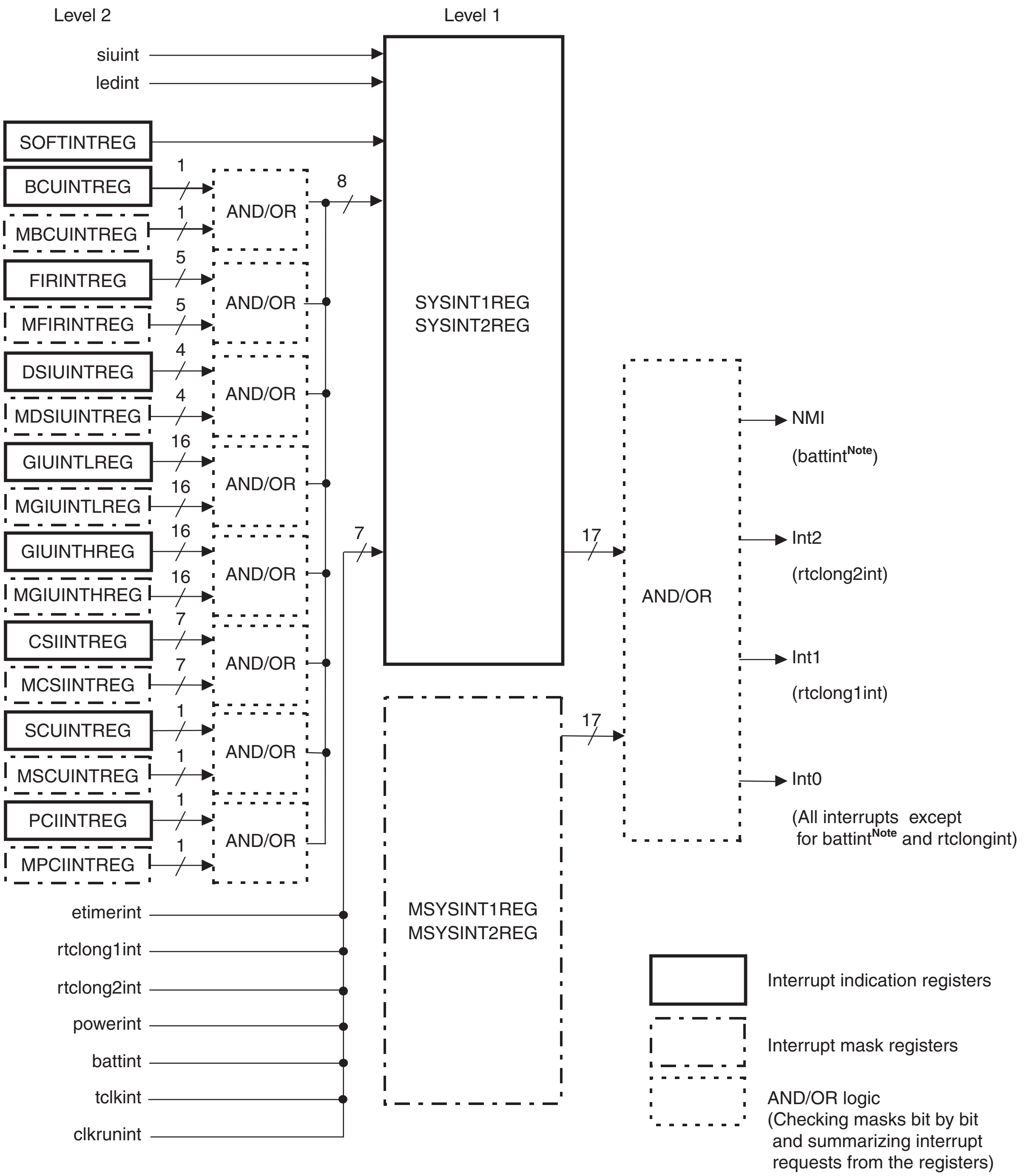
If an interrupt request occurs in the peripheral units, the corresponding bit in the interrupt indication register of level 2 (xxxINTREG) is set to 1. The interrupt indication register is ANDed bitwise with the corresponding interrupt mask register of level 2 (MxxxINTREG). If the generated interrupt request is enabled (set to 1) in the mask register, the interrupt request is reported to the system interrupt register of level 1 (SYSINT1REG) and the corresponding bit is set to 1. At this time, the interrupt requests from the same register of level 2 are reported to SYSINT1REG as a single interrupt request.

Interrupt requests from some units directly set their corresponding bits in SYSINT1REG.

SYSINT1REG is ANDed bitwise with the interrupt mask register of level 1 (MSYSINT1REG). If the interrupt request is enabled by MSYSINT1REG (set to 1), a corresponding interrupt request signal is output from the ICU to the CPU core. battint_intr is connected to the NMI or Int0 signal of the CPU core (selected by setting of NMIREG). rtc_long1_intr and rtc_long2_intr are connected to the Int1 signal and the Int2 signal of the CPU core, respectively. The other interrupt requests are connected to the Int0 signal of the CPU core as a single interrupt request.

The following figure shows an outline of interrupt control in the ICU.

Figure 11-1. Interrupt Control Outline



Note Which of NMI or Int0 is used for battint is selected by setting NMIREG. When NMI use is selected, this interrupt cannot be masked in the CPU core, but masks can be set by MSYSINT1REG in the ICU.

11.2 Register Set

The ICU registers are listed below.

Table 11-1. ICU Registers

Address	R/W	Register Symbols	Function
0x0F00 0080	R	SYSINT1REG	System interrupt register 1 (level 1)
0x0F00 0088	R	GIUINTLREG	GIU interrupt lower register (level 2)
0x0F00 008A	R	DSIUINTREG	DSIU interrupt register (level 2)
0x0F00 008C	R/W	MSYSINT1REG	Mask system interrupt register 1 (level 1)
0x0F00 0094	R/W	MGIUINTLREG	Mask GIU interrupt lower register (level 2)
0x0F00 0096	R/W	MDSIUINTREG	Mask DSIU interrupt register (level 2)
0x0F00 0098	R/W	NMIREG	Battery interrupt selection register
0x0F00 009A	R/W	SOFTINTREG	Software interrupt register
0x0F00 00A0	R	SYSINT2REG	System interrupt register 2 (level 1)
0x0F00 00A2	R	GIUINTHREG	GIU interrupt higher register (level 2)
0x0F00 00A4	R	FIRINTREG	FIR interrupt register (level 2)
0x0F00 00A6	R/W	MSYSINT2REG	Mask system interrupt register 2 (level 1)
0x0F00 00A8	R/W	MGIUINTHREG	Mask GIU interrupt higher register (level 2)
0x0F00 00AA	R/W	MFIRINTREG	Mask FIR interrupt register (level 2)
0x0F00 00AC	R	PCIINTREG	PCI interrupt register (level 2)
0x0F00 00AE	R	SCUINTREG	SCU interrupt register (level 2)
0x0F00 00B0	R	CSIINTREG	CSI interrupt register (level 2)
0x0F00 00B2	R/W	MPCIINTREG	Mask PCI interrupt register (level 2)
0x0F00 00B4	R/W	MSCUINTREG	Mask SCU interrupt register (level 2)
0x0F00 00B6	R/W	MCSIINTREG	Mask CSI interrupt register (level 2)
0x0F00 00B8	R	BCUINTREG	BCU interrupt register (level 2)
0x0F00 00BA	R/W	MBCUINTREG	Mask BCU interrupt register (level 2)

These registers are described in detail below.

11.2.1 SYSINT1REG (0x0F00 0080)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	CLKRUN INTR	SOFT INTR	RFU	SIUINTR	GIUINTR
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	ETIMER INTR	RTCL1 INTR	POWER INTR	BATINTR
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:13	RFU	Reserved. Write 0. 0 is returned after a read.
12	CLKRUNINTR	CLKRUN interrupt request 1: Occurred 0: Normal
11	SOFTINTR	Software interrupt request (occurs by setting the SOFTINTREG) 1: Occurred 0: Normal
10	RFU	Reserved. Write 0. 0 is returned after a read.
9	SIUINTR	SIU interrupt request 1: Occurred 0: Normal
8	GIUINTR	GIU (higher/lower) interrupt request 1: Occurred 0: Normal
7:4	RFU	Reserved. Write 0. 0 is returned after a read.
3	ETIMERINTR	Elapsed Time timer interrupt request 1: Occurred 0: Normal
2	RTCL1INTR	RTC Long1 timer interrupt request 1: Occurred 0: Normal
1	POWERINTR	Power switch interrupt request 1: Occurred 0: Normal
0	BATINTR	Battery interrupt request 1: Occurred 0: Normal

This register indicates whether various interrupt requests occur in the V_R4131 system.

11.2.2 GIUINTLREG (0x0F00 0088)

Bit	15	14	13	12	11	10	9	8
Name	INTS15	RFU	INTS13	INTS12	INTS11	INTS10	INTS9	INTS8
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	INTS7	INTS6	INTS5	INTS4	INTS3	INTS2	INTS1	INTS0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15	INTS15	Inputs interrupt request to GPIO(15) pin. 1: Occurred 0: Normal
14	RFU	Reserved. Write 0. 0 is returned after a read.
13:0	INTS(13:0)	Inputs interrupt request to GPIO(13:0) pins. 1: Occurred 0: Normal

This register indicates whether various GIU-related interrupt requests occur.

11.2.3 DSIUINTREG (0x0F00 008A)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	INTDSIU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:12	RFU	Reserved. Write 0. 0 is returned after a read.
11	INTDSIU	Interrupt request from DSIU 1: Occurred 0: Normal
10:0	RFU	Reserved. Write 0. 0 is returned after a read.

This register indicates whether various DSIU-related interrupt requests occur.

11.2.4 MSYSINT1REG (0x0F00 008C)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	CLKRUN INTR	SOFT INTR	RFU	SIUINTR	GIUINTR
R/W	R	R	R	R/W	R/W	R	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	ETIMER INTR	RTCL1 INTR	POWER INTR	BATINTR
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:13	RFU	Reserved. Write 0. 0 is returned after a read.
12	CLKRUNINTR	CLKRUN interrupt enable 1: Enable 0: Disable
11	SOFTINTR	Software interrupt (occurs by setting the SOFTINTREG) enable 1: Enable 0: Disable
10	RFU	Reserved. Write 0. 0 is returned after a read.
9	SIUINTR	SIU interrupt enable 1: Enable 0: Disable
8	GIUINTR	GIU (lower) interrupt enable 1: Enable 0: Disable
7:4	RFU	Reserved. Write 0. 0 is returned after a read.
3	ETIMERINTR	Elapsed Time timer interrupt enable 1: Enable 0: Disable
2	RTCL1INTR	RTC Long1 timer interrupt enable 1: Enable 0: Disable
1	POWERINTR	PowerSW interrupt enable 1: Enable 0: Disable
0	BATINTR	Battery interrupt enable 1: Enable 0: Disable

This register is used to mask various interrupt requests that occur in the VR4131 system.

11.2.5 MGIUINTLREG (0x0F00 0094)

Bit	15	14	13	12	11	10	9	8
Name	INTS15	RFU	INTS13	INTS12	INTS11	INTS10	INTS9	INTS8
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	INTS7	INTS6	INTS5	INTS4	INTS3	INTS2	INTS1	INTS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15	INTS15	Interrupt input enable to GPIO(15) pin 1: Enable 0: Disable
14	RFU	Reserved. Write 0. 0 is returned after a read.
13:0	INT(13:0)	Interrupt input enable to GPIO(13:0) pins 1: Enable 0: Disable

This register is used to mask various GIU-related interrupt requests.

11.2.6 MDSIUINTREG (0x0F00 0096)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	INTDSIU	RFU	RFU	RFU
R/W	R	R	R	R	R/W	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:12	RFU	Reserved. Write 0. 0 is returned after a read.
11	INTDSIU	DSIU interrupt enable 1: Enable 0: Disable
10:0	RFU	Reserved. Write 0. 0 is returned after a read.

This register is used to mask various DSIU-related interrupt requests.

11.2.7 NMIREG (0x0F00 0098)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	NMIOR INT
R/W	R	R	R	R	R	R	R	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:1	RFU	Reserved. Write 0. 0 is returned after a read.
0	NMIORINT	Reports low battery detect interrupt request. 1: Int0 0: NMI

This register is used to set the type of interrupt request used to notify the VR4130 CPU core when a low battery detect interrupt request has occurred.

11.2.8 SOFTINTREG (0x0F00 009A)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	SOFT INTR3	SOFT INTR2	SOFT INTR1	SOFT INTR0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:4	RFU	Reserved. Write 0. 0 is returned after a read.
3:0	SOFTINTR(3:0)	Software interrupt request 1: Set 0: Clear

This register is used to generate software interrupts. Each bit can be set separately, and can generate four types of interrupt requests.

11.2.9 SYSINT2REG (0x0F00 00A0)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	BCU INTR	CSI INTR
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	SCU INTR	PCI INTR	DSIU INTR	FIR INTR	TCLK INTR	RFU	LED INTR	RTCL2 INTR
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:10	RFU	Reserved. Write 0. 0 is returned after a read.
9	BCUINTR	BCU interrupt request 1: Occurred 0: Normal
8	CSIINTR	CSI interrupt request 1: Occurred 0: Normal
7	SCUINTR	SCU interrupt request 1: Occurred 0: Normal
6	PCIINTR	PCI interrupt request 1: Occurred 0: Normal
5	DSIUINTR	DSIU interrupt request 1: Occurred 0: Normal
4	FIRINTR	FIR interrupt request 1: Occurred 0: Normal
3	TCLKINTR	TClock counter interrupt request 1: Occurred 0: Normal
2	RFU	Reserved. Write 0. 0 is returned after a read
1	LEDINTR	LED interrupt request 1: Occurred 0: Normal
0	RTCL2INTR	RTC Long2 timer interrupt request 1: Occurred 0: Normal

This register indicates whether various interrupt requests occur in the V_R4131 system.

11.2.10 GIUINTHREG (0x0F00 00A2)

Bit	15	14	13	12	11	10	9	8
Name	INTS31	INTS30	INTS29	INTS28	INTS27	INTS26	INTS25	INTS24
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	INTS23	INTS22	INTS21	INTS20	INTS19	INTS18	INTS17	INTS16
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:0	INTS(31:16)	Inputs interrupt request to GPIO(31:16) pins 1: Occurred 0: Normal

This register indicates whether various GIU-related interrupt requests occur.

11.2.11 FIRINTREG (0x0F00 00A4)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	FIRINT	FDPINT4	FDPINT3	FDPINT2	FDPINT1
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:5	RFU	Reserved. Write 0. 0 is returned after a read.
4	FIRINT	Interrupt request from FIR unit 1: Occurred 0: Normal
3	FDPINT4	FIR reception DMA buffer 2-page boundary interrupt request 1: Occurred 0: Normal
2	FDPINT3	FIR transmission DMA buffer 2-page boundary interrupt request 1: Occurred 0: Normal
1	FDPINT2	FIR reception DMA buffer 1-page boundary interrupt request 1: Occurred 0: Normal
0	FDPINT1	FIR transmission DMA buffer 1-page boundary interrupt request 1: Occurred 0: Normal

This register indicates whether various FIR-related interrupt requests occur.

When FDPINT4 or FDPINT3 is set to 1, the VR4131 stops the DMA requests. When FDPINT2 or FDPINT1 is set to 1 while the FDPCNT bit of the DPCNTR register (0x0F00 0844) is set to 1 (DMA buffer 1 page interrupt request is generated), the VR4131 stops the DMA requests.

11.2.12 MSYSINT2REG (0x0F00 00A6)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	BCU INTR	CSI INTR
R/W	R	R	R	R	R	R	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	SCU INTR	PCI INTR	DSIU INTR	FIR INTR	TCLK INTR	RFU	LEDINTR	RTCL2 INTR
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:10	RFU	Reserved. Write 0. 0 is returned after a read.
9	BCUINTR	BCU interrupt enable 1: Enable 0: Disable
8	CSIINTR	CSI interrupt enable 1: Enable 0: Disable
7	SCUINTR	SCU interrupt enable 1: Enable 0: Disable
6	PCIINTR	PCI interrupt enable 1: Enable 0: Disable
5	DSIUINTR	DSIU interrupt enable 1: Enable 0: Disable
4	FIRINTR	FIR interrupt enable 1: Enable 0: Disable
3	TCLKINTR	VTClock counter interrupt enable 1: Enable 0: Disable
2	RFU	Reserved. Write 0. 0 is returned after a read.
1	LEDINTR	LED interrupt enable 1: Enable 0: Disable
0	RTCL2INTR	RTC Long2 timer interrupt enable 1: Enable 0: Disable

This register is used to mask various interrupt requests in the VR4131 system.

11.2.13 MGIUINTHREG (0x0F00 00A8)

Bit	15	14	13	12	11	10	9	8
Name	INTS31	INTS30	INTS29	INTS28	INTS27	INTS26	INTS25	INTS24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	INTS23	INTS22	INTS21	INTS20	INTS19	INTS18	INTS17	INTS16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:0	INTS(31:16)	Interrupt input enable to GPIO(31:16) pins 1: Enable 0: Disable

This register is used to mask various GIU-related interrupt requests.

11.2.14 MFIRINTREG (0x0F00 00AA)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	FIRINT	FDPINT4	FDPINT3	FDPINT2	FDPINT1
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:5	RFU	Reserved. Write 0 to these bits. 0 is returned after a read.
4	FIRINT	FIR unit interrupt enable 1: Enable 0: Disable
3	FDPINT4	FIR reception DMA buffer 2-page boundary interrupt enable 1: Enable 0: Disable
2	FDPINT3	FIR transmission DMA buffer 2-page boundary interrupt enable 1: Enable 0: Disable
1	FDPINT2	FIR reception DMA buffer 1-page boundary interrupt enable 1: Enable 0: Disable
0	FDPINT1	FIR transmission DMA buffer 1-page boundary interrupt enable 1: Enable 0: Disable

This register is used to mask various FIR-related interrupt requests.

11.2.15 PCIINTREG (0x0F00 00AC)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	PCIINT0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:1	RFU	Reserved. Write 0. 0 is returned after a read.
0	PCIINT0	Interrupt request from PCI macro 1: Occurred 0: Normal

This register indicates whether various PCI-related interrupt requests occur.

11.2.16 SCUINTREG (0x0F00 00AE)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	SCUINT0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:1	RFU	Reserved. Write 0. 0 is returned after a read.
0	SCUINT0	SCU interrupt request 1: Occurred 0: Normal

This register indicates whether various SCU-related interrupt requests occur.

11.2.17 CSIINTREG (0x0F00 00B0)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	TRPAGE2	TRPAGE1	TREND	TREMPTY	RCPAGE2	RCPAGE1	RCOVER
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:7	RFU	Reserved. Write 0. 0 is returned after a read.
6	TRPAGE2	DMA transmit 2-page interrupt 1: Occurred 0: Normal
5	TRPAGE1	DMA transmit 1-page interrupt 1: Occurred 0: Normal
4	TREND	Transfer interrupt per 1 data 1: Occurred 0: Normal
3	TREMPTY	FIFO transmit empty interrupt 1: Occurred 0: Normal
2	RCPAGE2	DMA receive 2-page interrupt 1: Occurred 0: Normal
1	RCPAGE1	DMA receive 1-page interrupt 1: Occurred 0: Normal
0	RCOVER	FIFO receive overrun interrupt 1: Occurred 0: Normal

This register indicates whether various CSI-related interrupt requests occur.

11.2.18 MPCIINTREG (0x0F00 00B2)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	PCIINT0
R/W	R	R	R	R	R	R	R	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:1	RFU	Reserved. Write 0. 0 is returned after a read.
0	PCIINT0	PCI interrupt enable 1: Enable 0: Disable

This register is used to mask various PCI-related interrupt requests.

11.2.19 MSCUINTREG (0x0F00 00B4)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	SCUINT0
R/W	R	R	R	R	R	R	R	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:1	RFU	Reserved. Write 0. 0 is returned after a read.
0	SCUINT0	SCU interrupt enable 1: Enable 0: Disable

This register is used to mask various SCU-related interrupt requests.

11.2.20 MCSIINTREG (0x0F00 00B6)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	TRPAGE2	TRPAGE1	TREND	TREMPTY	RCPAGE2	RCPAGE1	RCOVER
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:7	RFU	Reserved. Write 0. 0 is returned after a read.
6	TRPAGE2	DMA transmit 2-page interrupt enable 1: Enable 0: Disable
5	TRPAGE1	DMA transmit 1-page interrupt enable 1: Enable 0: Disable
4	TREND	Transfer interrupt mask per 1 enable 1: Enable 0: Disable
3	TREMPTY	FIFO transmit empty interrupt enable 1: Enable 0: Disable
2	RCPAGE2	DMA receive 2-page interrupt enable 1: Enable 0: Disable
1	RCPAGE1	DMA receive 1-page interrupt enable 1: Enable 0: Disable
0	RCOVER	FIFO receive overrun interrupt enable 1: Enable 0: Disable

This register is used to set various CSI-related interrupt request masks.

11.2.21 BCUINTREG (0x0F00 00B8)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	BCUINTR
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:1	RFU	Reserved. Write 0. 0 is returned after a read.
0	BCUINTR	BCU interrupt request 1: Occurred 0: Normal

This register indicates whether various BCU-related interrupt requests occur.

11.2.22 MBCUINTREG (0x0F00 00BA)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	MBCUINTR
R/W	R	R	R	R	R	R	R	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:1	RFU	Reserved. Write 0. 0 is returned after a read.
0	MBCUINTR	BCU interrupt enable 1: Enable 0: Disable

This register is used to set various BCU interrupt request masks.

11.3 Notes for Register Setting

There is no register setting flow in relation to the ICU.

With regard to each interrupt mask register, the initial setting is 0 (mask) after reset. Therefore, enough masks must be cleared to provide sufficient interrupts for the CPU's start-up processing. This is always necessary for battery interrupt requests (battint_intr).

The battery interrupt request is assigned to NMI for the initial setting (NMIREG bit 0 = 0). A "1" must be written to the NMIREG register to switch this setting to Int0.

The software interrupt request (soft_intr) is output to Int0 by setting 1 to the SOFTINTREG register. Writing a 0 clears the interrupt.

CHAPTER 12 PMU (POWER MANAGEMENT UNIT)

12.1 General

The PMU performs the following power management within the VR4131 and controls the power supply throughout the system that includes the VR4131.

- Reset control
- Shutdown control
- Power-on control
- Low power consumption mode control

The PMU also performs settings to use the GPIO(12:9) and GPIO(3:0) signals as a startup factor.

12.1.1 Reset control

The operations of the RTC, peripheral units, CPU core, and PMUINTREG bit settings during a reset are listed below.

Table 12-1. Bit Operations During Reset

Reset Type	RTC	Peripheral Units	CPU Core	PMUINTREG
RTC reset	Reset	Reset	Cold reset	RTCRST=1
RSTSW reset	Active	Reset	Cold reset	RSTSW=1

(1) RTC reset

When the RTCRST# signal becomes active, the PMU resets all peripheral units including the RTC unit. It also makes the ccoldresetb and creset signals (internal) active and resets the CPU core.

In addition, the RTCRST bit in PMUINTREG is set to 1. After the CPU is restarted, the RTCRST bit must be checked and cleared to 0 by software.

For details of the timing of a RTC reset, refer to **6.1.1 RTC reset**.

(2) RSTSW reset

When the RSTSW# signal becomes active, the PMU resets all peripheral units except for the RTC and PMU. It also makes the ccoldresetb and creset signals (internal) active and resets the CPU core.

In addition, the RSTSW bit in PMUINTREG is set to 1. After the CPU is restarted, the RSTSW bit must be checked and cleared to 0 by software.

For details of the timing of a RSTSW reset, refer to **6.1.2 RSTSW**.

12.1.2 Shutdown control

The operations of the RTC, peripheral units, and CPU core, and the bits set by the PMUINTREG and PMUCNTREG registers are listed below.

Table 12-2. Bit Operations During Shutdown

Shutdown Type	RTC	Peripheral Units	CPU Core	Set Bit
HALTimer shutdown	Active	Reset	Cold reset	HALTIMERRST of PMUCNTREG register = 1
Software shutdown	Active	Reset	Cold reset	–
BATTINH shutdown	Active	Reset	Cold reset	BATTINH of PMUINTREG register = 1

(1) HALTimer shutdown

After the CPU is activated (following the mode change from shutdown or Hibernate mode to Fullspeed mode), the software must write 1 to PMUCNTREG's HALTIMERRST bit within about four seconds to clear the HALTimer.

If the HALTimer is not cleared within about four seconds after the CPU is activated, the PMU resets all peripheral units except for RTC and PMU. Next, it asserts the ccoldresetb and creset signals (internal) and resets the CPU core.

In addition, the TIMOUTRST bit in PMUINTREG is set to 1. After the CPU is restarted, the TIMOUTRST bit must be checked and cleared to 0 by software.

For details of the timing of HALTimer shutdown, refer to **6.1.4 HALTimer shutdown**.

(2) Software shutdown

When the HIBERNATE instruction is executed, the PMU checks for currently pending interrupts. If there are no pending interrupts, it stops the CPU clock. It then resets all peripheral units except for RTC and PMU.

The PMU register contents do not change.

For details of the timing of software shutdown, refer to **6.1.3 Software shutdown**.

(3) BATTINH shutdown

If the BATTINH/BATTINT# signal is low when the CPU is going to start, PMU stops CPU activation, and resets all peripheral units except for RTC and PMU. Next, it asserts the ccoldresetb and creset signals (internal) and resets the CPU core.

In addition, the BATTINH bit in PMUINTREG is set to 1. After the CPU is restarted, the BATTINH bit must be checked and cleared to 0 by software.

For details of the timing of a BATTINH shutdown, refer to **12.1.3 Power-on control** below.

12.1.3 Power-on control

The causes of CPU activation (mode change from shutdown or Hibernate mode to Fullspeed mode) are called startup factors. There are eleven startup factors: a power switch interrupt (POWER), eight types of GPIO activation interrupts (GPIO(3:0), (12:9)), a DCD interrupt (DCD#), and an Elapsed Time timer interrupt.

The BATTINH/BATTINT# pin check is a factor that prevents CPU activation.

The period (power-on wait time), in which the POWERON pin is active at power-on, can be specified by using PMUWAITREG. After RTCRST, by which the CPU is initialized, the period is set to 343.78 ms. The power-on wait time can be specified when activation is caused by sources other than RTCRST.

During CPU activation, supplying voltage to the 1.5 V power supplies ($V_{DD}1$, $V_{DD}P$, $V_{DD}PD$) can be stopped to reduce the leakage current. This means that these power supplies may become 0 V while the MPOWER pin is inactive. The following operation will not be affected by supplying voltage of 1.5 V or more to these power supplies within the period from when the MPOWER pin becomes active to when PLL starts oscillation.

Caution When the CPU switches to Hibernate mode by execution of the HIBERNATE instruction and a startup factor occurs simultaneously, the CPU may be activated without the POWERON signal being active after the MPOWER signal is once inactive. Moreover, if the RSTSW# signal, which is not a startup factor from Hibernate mode, is active at the same time as the CPU switches to Hibernate mode, the CPU may be activated without the POWERON signal being active after the MPOWER signal is inactive once.

(1) Activation via power switch interrupt

When the POWER signal becomes active (if the rising edge of POWER is detected by the PMU), the PMU makes the POWERON signal active and provides external notification that the CPU is being activated. After making the POWERON signal active, the PMU checks the BATTINH/BATTINT# signal and then makes the POWERON signal inactive.

If the BATTINH/BATTINT# signal is high, the PMU cancels peripheral unit reset, then starts the Cold Reset sequence to activate the CPU core.

If the BATTINH/BATTINT# signal is low, the PMU sets PMUINTREG's BATTINH bit to 1 and then performs another shutdown. After the CPU is restarted, the BATTINH bit must be checked and cleared by software.

Activation via power switch interrupt never sets PMUINTREG's POWERSWINTR bit to 1.

Figure 12-1. Activation via Power Switch Interrupt (BATTINH/BATTINT# = 1)

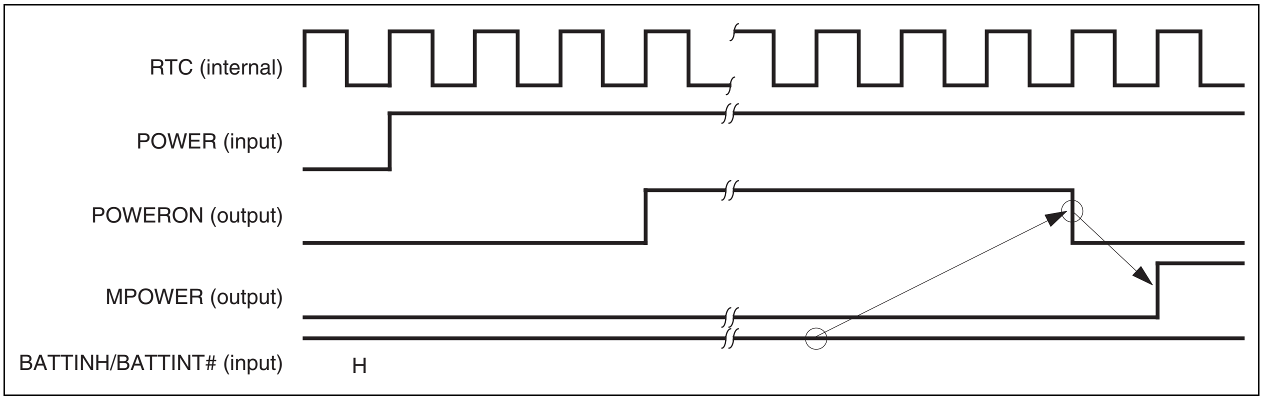
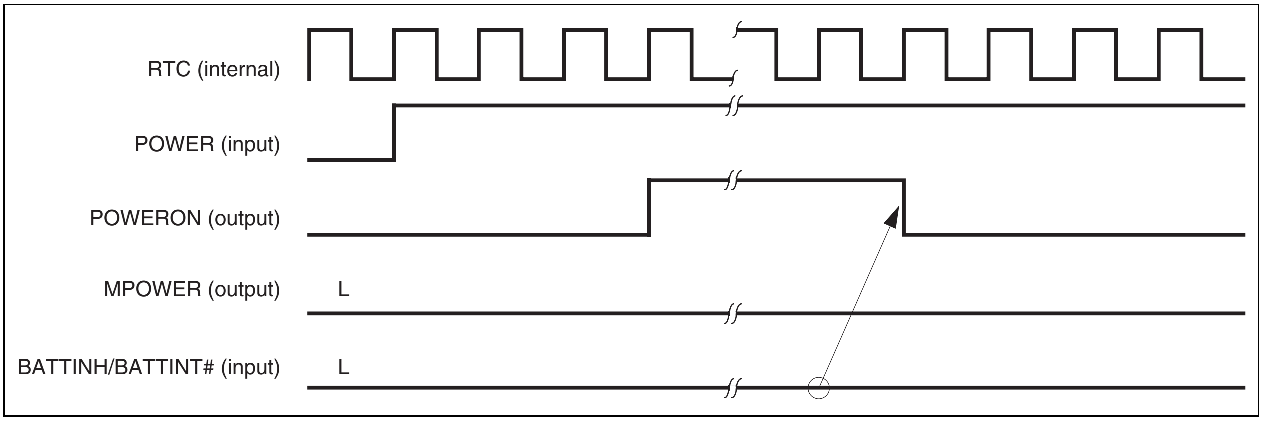


Figure 12-2. Activation via Power Switch Interrupt (BATTINH/BATTINT# = 0)



(2) Activation via GPIO activation interrupt

When the GPIO(3:0) and GPIO(12:9) signals become active, the PMU checks the activation interrupt enable bit for GPIO(3:0) and GPIO(12:9). If GPIO(3:0) and GPIO(12:9) activation interrupts are enabled, the PMU makes the POWERON signal active and provides external notification that the CPU is being activated (since the GPIO(2:0) and GPIO(12:9) activation enable interrupt bit is cleared after an RTC is reset, the GPIO(2:0) and GPIO(12:9) signals cannot be used for activation immediately after an RTC reset. However, activation can occur at the falling edge of the GPIO3 signal immediately after an RTC reset for GPIO3 only). The PMU makes the POWERON signal active, then checks the BATTINH/BATTINT# signal and makes the POWERON signal inactive.

When the BATTINH/BATTINT# signal is high, the PMU cancels peripheral unit reset, then starts the cold reset sequence to activate the CPU core.

When the BATTINH/BATTINT# signal is low, the PMU sets PMUINTREG's BATTINH bit to 1 and then performs another shutdown. After the CPU is restarted, the BATTINH bit must be checked and cleared by means of software.

The CPU sets the corresponding GPIOINTR bit in the PMUINTREG or PMUINT2REG to 1 regardless of whether activation succeeds or fails.

Caution The changes in the GPIO signal are ignored while POWER signal is active.

Figure 12-3. Activation via GPIO Activation Interrupt (BATTINH/BATTINT# = 1)

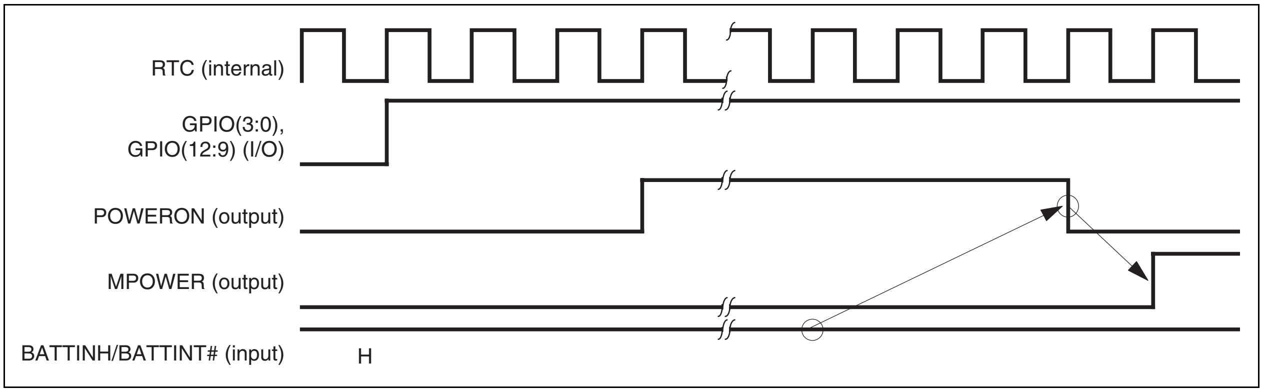
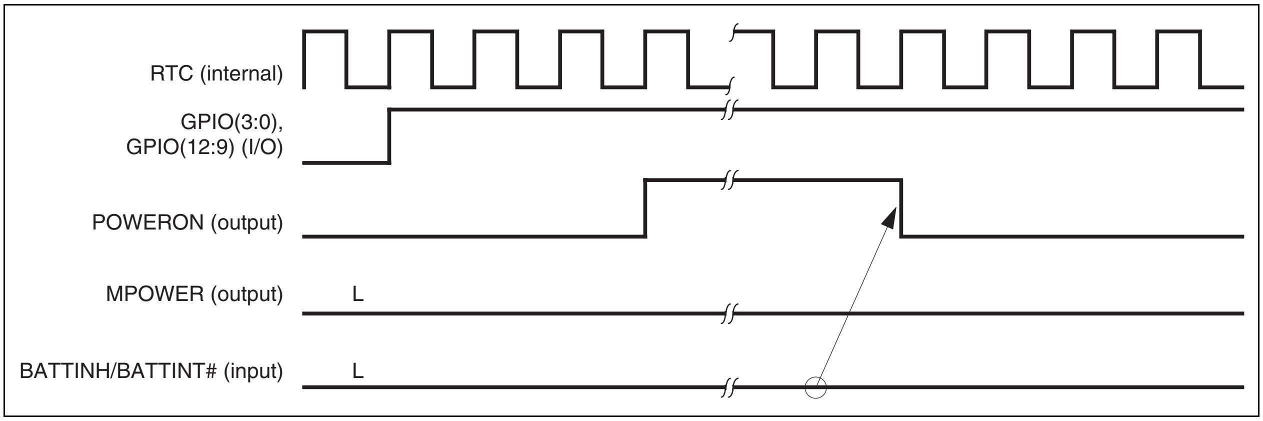


Figure 12-4. Activation via GPIO Activation Interrupt (BATTINH/BATTINT# = 0)



(3) Activation via DCD interrupt

When the DCD# signal becomes active (it means, the falling edge of DCD# signal is detected by PMU), the PMU makes the POWERON signal active and provides external notification that the CPU is being started. After making the POWERON signal active, the PMU checks the BATTINH/BATTINT# signal and then makes the POWERON signal inactive.

If the BATTINH/BATTINT# signal is high, the PMU cancels peripheral unit reset, then starts the cold reset sequence to activate the CPU core.

If the BATTINH/BATTINT# signal is low, the PMU sets PMUINTREG's BATTINH bit to 1 and then performs another shutdown. After the CPU is restarted, the BATTINH bit must be checked and cleared by means of software.

The PMUINTREG's DCDST bit in PMU does not indicate whether a DCD interrupt has occurred, but instead reflects the current status of the DCD# pin.

Cautions 1. When POWERON is active, the PMU cannot recognize changes in the DCD# signal. If the DCD# state when POWERON is active is different from the DCD# state when POWERON is inactive, the change in the DCD# signal is detected only after POWERON becomes inactive. However, if the DCD# state when POWERSW is active is the same as the DCD# state when POWERON is inactive, any changes in the DCD# signal that occur while POWERON is active are not detected.

2. The changes in the DCD# signal are ignored while the POWER signal is active.

Figure 12-5. Activation via DCD Interrupt (BATTINH/BATTINT# = 1)

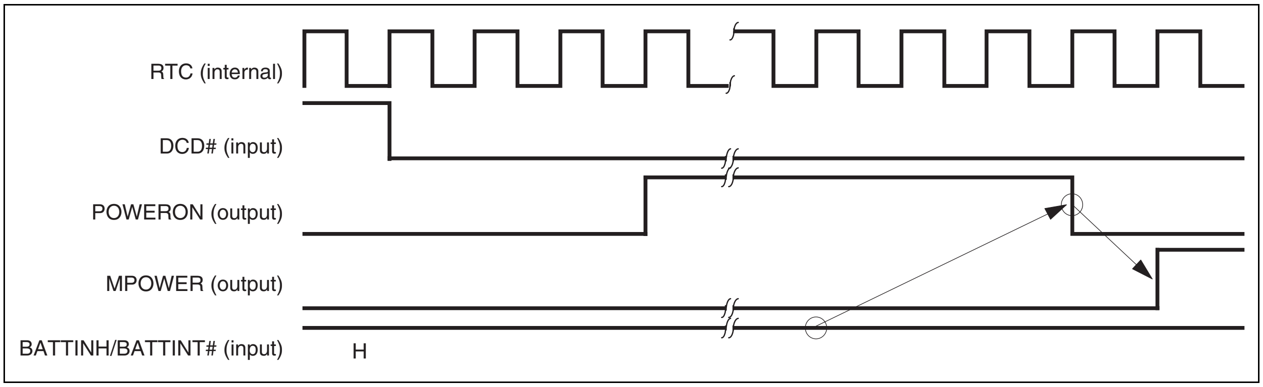
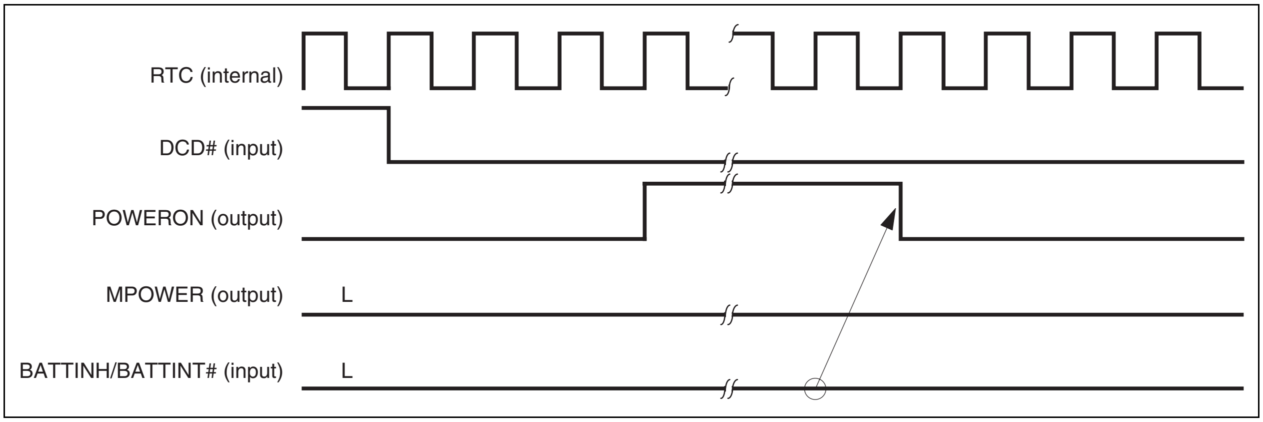


Figure 12-6. Activation via DCD Interrupt (BATTINH/BATTINT# = 0)



(4) Activation via Elapsed Time timer interrupt

When the alarm interrupt (alarm_intr) signal generated from the Elapsed Time timer becomes active, the PMU makes the POWERON signal active and provides external notification that the CPU is being activated. After making the POWERON signal active, the PMU checks the BATTINH/BATTINT# signal and then makes the POWERON signal inactive.

If the BATTINH/BATTINT# signal is high, the PMU cancels peripheral unit reset, then starts the cold reset sequence to activate the CPU core.

If the BATTINH/BATTINT# signal is low, the PMU sets PMUINTREG's BATTINH bit to 1 and then performs another shutdown. After the CPU is restarted, the BATTINH bit must be checked and cleared by means of software.

Caution The Elapsed Time timer interrupt is ignored while the POWER signal is active. After the POWER signal becomes inactive, it is notified to PMU.

Figure 12-7. Activation via Elapsed Timer Interrupt (BATTINH/BATTINT# = 1)

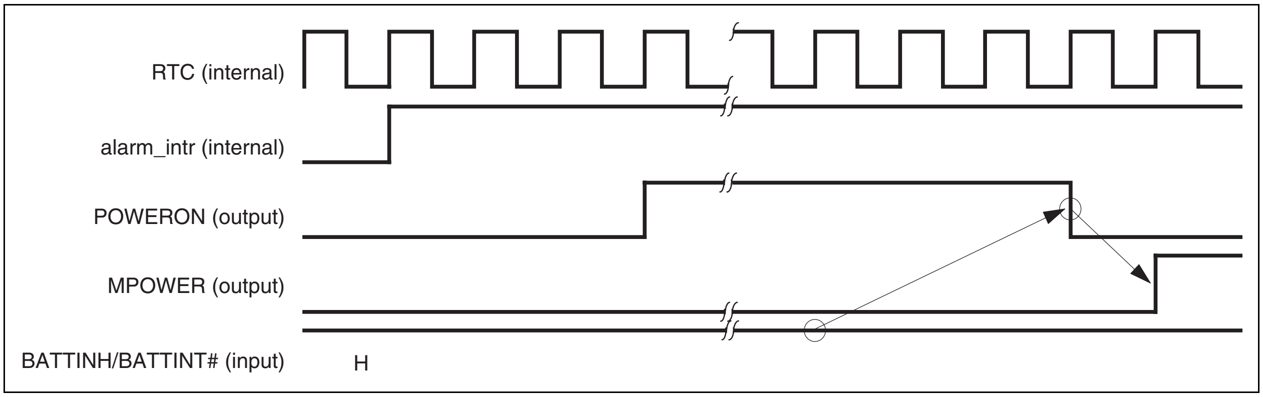
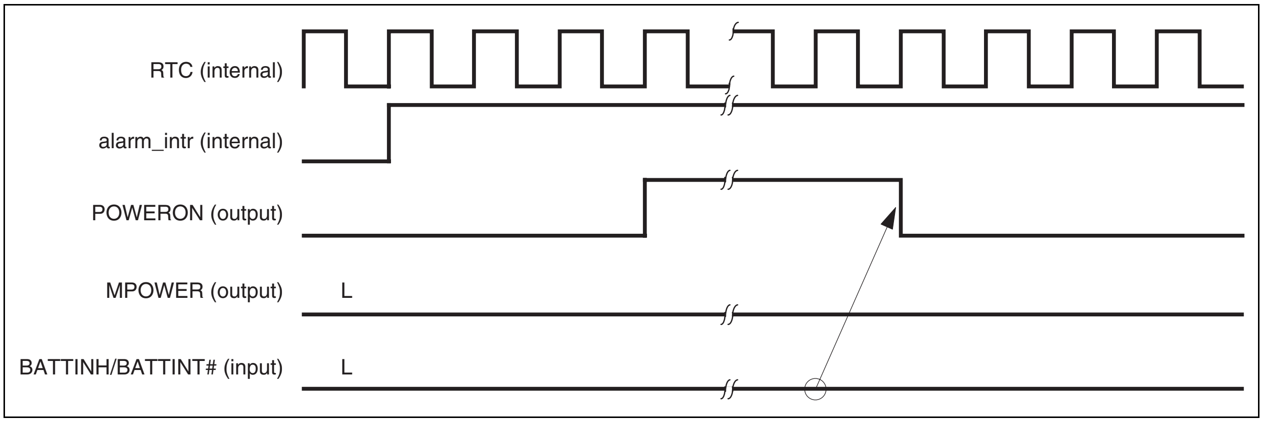


Figure 12-8. Activation via Elapsed Timer Interrupt (BATTINH/BATTINT# = 0)



12.1.4 Power mode

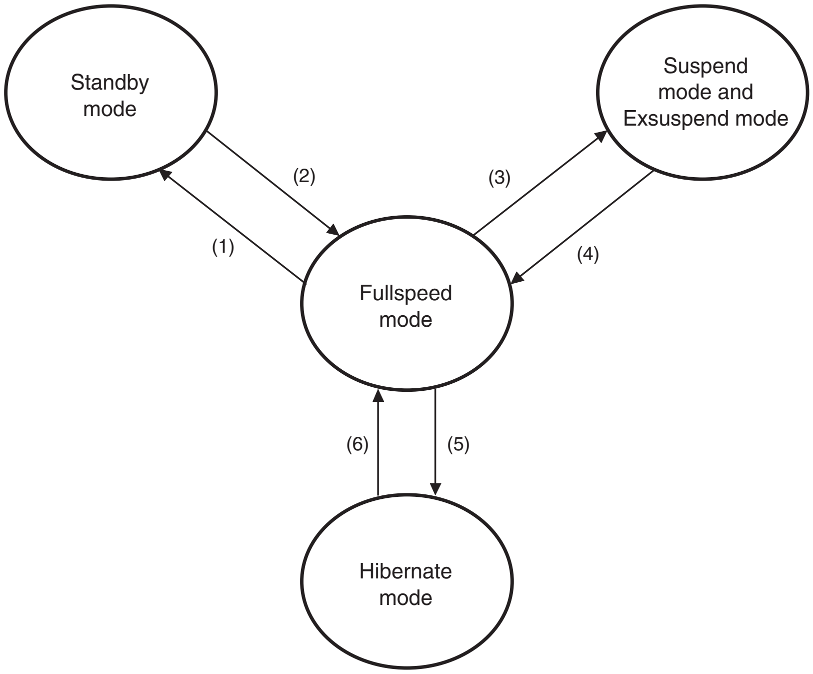
The VR4131 supports the following five power modes.

- Fullspeed mode
- Standby mode
- Suspend mode
- Exsuspend mode
- Hibernate mode

To set Standby, Suspend, Exsuspend, or Hibernate mode from Fullspeed mode, execute the STANDBY, SUSPEND, or HIBERNATE instruction, respectively. RTCRST is always valid in every mode, and initializes units in the CPU including the RTC. However, the CPU is not restarted by RTCRST.

The following figure shows the power mode state transitions.

Figure 12-9. Power Mode State Transition



(1)	(2)	(3)	(4)	(5)	(6)
STANDBY instruction, pipeline flash, SysAD idle , and PClock high	All interrupts	SUSPEND instruction, pipeline flash, SysAD idle, PClock high, VTClock high, and DRAM self refresh	POWER RSTSW Elapsed Time RTC Long1 RTC Long2 GPIO(13:0) DCD# (GPIO, SIU) BATTINTR	HIBERNATE instruction, pipeline flash, SysAD idle, PClock high, VTClock high, MasterOut high, and DRAM self refresh	POWER Elapsed Time DCD# GPIO(12:9) GPIO(3:0)

Table 12-3. Power Mode

Mode	PLL	Internal Peripheral Unit				CPU Core
		RTC	ICU	DCU	Other	
Fullspeed	On	On	On	On	Selectable[Note]	On
Standby	On	On	On	On	Selectable[Note]	Off
Suspend	On	On	On	Off	Off	Off
PLL-off Suspend	Off	On	On	Off	Off	Off
Hibernate	Off	On	Off	Off	Off	Off
Off	Off	Off	Off	Off	Off	Off

Note Refer to **CHAPTER 10 CMU (CLOCK MASK UNIT)** for details.

(1) Fullspeed mode

In Fullspeed mode, all internal clocks and the bus clock operate. In this mode, all the functions of the VR4131 can be executed.

(2) Standby mode

In Standby mode, all internal clocks, other than those provided for the on-chip peripheral units and the internal timer/interrupt unit of the CPU core, are fixed to high level.

To switch to Standby mode from Fullspeed mode, first execute the STANDBY instruction. The VR4131 waits until the SysAD bus (internal) enters the idle status after the completion of the WB stage of the STANDBY instruction. Then, the internal clock is shut down, and the pipeline stops. The PLL, timer/interrupt clock, internal bus clocks (VTClock, Intrclk), and RTC continue to operate.

In Standby mode, the processor returns to Fullspeed mode when an interrupt occurs.

(3) Suspend mode and Exsuspend mode

In Suspend mode, all internal clocks (including VTClock) other than those supplied to the RTC/ICU/PMU on-chip peripheral units and the internal timer/interrupt unit of the CPU core are fixed to high level.

To switch to Suspend mode from Fullspeed mode, first execute the SUSPEND instruction. The VR4131 waits until the SysAD bus (internal) enters the idle status after the completion of the WB stage of the SUSPEND instruction and DRAM has entered self-refresh mode. Then, the internal clocks (including VTClock) are shut down, and the pipeline stops. The PLL, timer/interrupt clock, Intrclk, and RTC continue to operate.

If the SUSPEND instruction is executed during DMA transfer, the DMA transfer is suspended, and the operation is undefined.

In Suspend mode, the processor returns to Fullspeed mode when an interrupt request from the peripheral units or any resets occur.

Setting the PLLOFFEN bit of the PMUCNTREG register allows the processor to enter the Exsuspend mode. The PLL operation also stops in the Exsuspend mode.

(4) Hibernate mode

In Hibernate mode, all the clocks supplied to on-chip peripheral units other than RTC/ICU/PMU and to the CPU core are fixed to high level.

To switch to Hibernate mode from Fullspeed mode, first execute the HIBERNATE instruction. The VR4131 waits until the SysAD bus (internal) enters the idle status after the completion of the WB stage of the HIBERNATE instruction, DRAM has entered self-refresh mode, and the MPOWER pin has been made inactive. Then, the internal clocks (including VTClock and Intrclk) are shut down, and the pipeline stops. The PLL also stops, but the RTC continues to operate.

In Hibernate mode, the processor returns to Fullspeed mode when it is alarmed from the RTC, the power-on switch is pressed, the DCD# pin is active, or GPIO pin inputs an interrupt. At this time, the contents of caches, registers, and the TLB in the CPU core are undefined. The peripheral registers are also undefined except PMU, LED, and a part of RTC, BCU, and GIU. (For details about the contents of the registers after Hibernate mode, refer to the status of the registers after reset.)

- GPIO pins as startup factors

  The GPIO(12:9) and GPIO(3:0) pins can be used not only as general-purpose ports and interrupt request inputs but also as startup factors from Hibernate mode. In addition, the GPIO3 pin can be used as a startup factor that follows an RTC reset. The selection of these pins as startup factors, as well as the settings for triggering, levels, etc., are set via the PMU rather than via the GIU or ICU (refer to **12.2 Register Set**).

  The settings for using GPIO pins for activation after a reset or from another power mode are listed below.

Table 12-4. Activation via GPIO Pins

Pin	RTC Reset		Recovery from Hibernate Mode		Recovery (Interrupt Request) from Suspend or Standby Mode, or Interrupt Request in Fullspeed Mode	
	Synchronization Clock	Trigger	Synchronization Clock	Trigger	Synchronization Clock	Trigger
GPIO(12:9)	–	–	RTC	Rising edge, falling edge	Intrclk	Set via GIU
GPIO3	RTC	Falling edge	RTC	Rising edge, falling edge	Intrclk	Set via GIU
GPIO(2:0)	–	–	RTC	Rising edge, falling edge	Intrclk	Set via GIU

12.2 Register Set

The PMU registers are listed below.

Table 12-5. PMU Registers

Address	R/W	Register Symbol	Function
0x0F00 00C0	R/W	PMUINTREG	PMU interrupt/status register
0x0F00 00C2	R/W	PMUCNTREG	PMU control register
0x0F00 00C4	R/W	PMUINT2REG	PMU interrupt/status register 2
0x0F00 00C6	R/W	PMUCNT2REG	PMU control register 2
0x0F00 00C8	R/W	PMUWAITREG	PMU wait counter register
0x0F00 00CC	R/W	PMUTCLKDIVREG	PMU TCLK, VTCLK Div mode register
0x0F00 00CE	R/W	PMUINTRCLKDIVREG	PMU INTRCLK Div mode register

Each register is described in detail below.

12.2.1 PMUINTREG (0x0F00 00C0)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	GPIO3 INTR	GPIO2 INTR	GPIO1 INTR	GPIO0 INTR	CLKRUN INTR	DCDST	RTCINTR	BATTINH
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note 1**	**Note 1**	**Note 1**	**Note 1**	**Note 1**	**Note 1**	**Note 1**	**Note 1**

Bit	7	6	5	4	3	2	1	0
Name	memo1	memo0	TIMOUT RST	RTCRST	RSTSW	RFU	BATTINTR	POWER SWINTR
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W
RTCRST	0	0	0	1	**Note 1**	0	0	0
After reset	**Note 1**	**Note 1**	**Note 1**	**Note 1**	**Note 2**	**Note 1**	**Note 1**	**Note 1**

Bit	Name	Function
15	GPIO3INTR	GPIO3 activation interrupt request detection. Cleared to 0 when 1 is written. 1: Detected 0: Not detected
14	GPIO2INTR	GPIO2 activation interrupt request detection. Cleared to 0 when 1 is written. 1: Detected 0: Not detected
13	GPIO1INTR	GPIO1 activation interrupt request detection. Cleared to 0 when 1 is written. 1: Detected 0: Not detected
12	GPIO0INTR	GPIO0 activation interrupt request detection. Cleared to 0 when 1 is written. 1: Detected 0: Not detected
11	CLKRUNINTR	CLKRUN interrupt request detection. Cleared to 0 when 1 is written. 1: Detected 0: Not detected
10	DCDST	DCD# pin state 1: High 0: Low
9	RTCINTR	Elapsed Time timer interrupt request detection. Cleared to 0 when 1 is written. 1: Detected 0: Not detected
8	BATTINH	BATTINH/BATTINT# signal low level detection during power-on. Cleared to 0 when 1 is written. 1: Detected 0: Not detected

Notes 1. Holds the value before reset.

2. Reset after restoring from Hibernate mode: Holds the value before reset.
RSTSW reset or reset by SOFTRST bit: 1

(2/2)

Bit	Name	Function
7:6	memo(1:0)	Memo. These bits are readable/writable, and can be used by users freely.
5	TIMOUTRST	HALTimer reset detection. Cleared to 0 when 1 is written. 1: Detected 0: Not detected
4	RTCRST	RTC reset detection. Cleared to 0 when 1 is written. 1: Detected 0: Not detected
3	RSTSW	RESET switch interrupt request detection. Cleared to 0 when 1 is written. 1: Detected 0: Not detected
2	RFU	Reserved. Write 0. 0 is returned after a read.
1	BATTINTR	Battery low detection during normal operation. Cleared to 0 when 1 is written. 1: Detected 0: Not detected
0	POWERSWINTR	POWER switch interrupt request detection. Cleared to 0 when 1 is written. 1: Detected 0: Not detected

This register is used to set or indicate whether the CPU detects a startup factor and reset.

It also indicates the status of the DCD# pin.

The BATTINTR bit is set to 1 when the BATTINH/BATTINT# signal becomes low and a battery-low interrupt request occurs in modes other than Hibernate mode (MPOWER = 1).

The POWERSWINTR bit is set to 1 when the POWER signal becomes high and a power switch interrupt request occurs in modes other than Hibernate mode (MPOWER = 1). However, this bit is not set to 1 when the POWER signal becomes high in Hibernate mode (MPOWER = 0).

The CLKRUNINTR bit is set to 1 when the CLKRUN signal becomes low and the CLKRUN interrupt request occurs in Suspend mode. Even if the CLKRUN pin becomes low in modes other than Suspend mode, the CLKRUN interrupt request does not occur.

12.2.2 PMUCNTREG (0x0F00 00C2)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	GPIO3MSK	GPIO2MSK	GPIO1MSK	GPIO0MSK	GPIO3TRG	GPIO2TRG	GPIO1TRG	GPIO0TRG
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	1	0	0	0	1	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	STANDBY	RFU	RFU	RFU	PLLOFFEN	HALTIMER RST	RFU	RFU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
RTCRST	0	0	0	0	0	0	1	0
After reset	0	0	0	0	0	0	1	0

Bit	Name	Function
15	GPIO3MSK	GPIO3 activation enable 1: Enable 0: Disable
14	GPIO2MSK	GPIO2 activation enable 1: Enable 0: Disable
13	GPIO1MSK	GPIO1 activation enable 1: Enable 0: Disable
12	GPIO0MSK	GPIO0 activation enable 1: Enable 0: Disable
11	GPIO3TRG	Sets the GPIO3 activation interrupt request detection trigger. 1: Falling edge 0: Rising edge
10	GPIO2TRG	Sets the GPIO2 activation interrupt request detection trigger. 1: Falling edge 0: Rising edge
9	GPIO1TRG	Sets the GPIO1 activation interrupt request detection trigger. 1: Falling edge 0: Rising edge
8	GPIO0TRG	Sets the GPIO0 activation interrupt request detection trigger. 1: Falling edge 0: Rising edge
7	STANDBY	Standby mode setting. This setting is performed only for software, and does not affect hardware in any way. 1: Standby mode 0: Normal mode

Note Holds the value before reset.

(2/2)

Bit	Name	Function
6:4	RFU	Reserved. Write 0. 0 is returned after a read.
3	PLLOFFEN	PLL halt enable during Suspend mode 1: PLL halt (Exsuspend mode) 0: PLL operation
2	HALTIMERRST	HALTimer reset 1: Reset 0: Set
1	RFU	Reserved. Write 1. 1 is returned after a read.
0	RFU	Reserved. Write 0. 0 is returned after a read.

This register is used to set CPU shutdown and overall system management operations.

The PLLOFFEN bit sets the PLL operation in Suspend mode.

The HALTIMERRST bit must be reset (write 1) within about four seconds after power-on[Note]. Resetting the HALTIMERRST bit enables the VR4131 to recognize that it has been normally activated. If the HALTIMERRST bit is not reset within about four seconds after power-on, the VR4131 regards the program as having not been normally executed (possibly runaway) and it automatically shuts down.

Each GPIO(3:0)MSK bit sets enable or disable for activation from Hibernate mode by GPIO(3:0) interruption. The GPIO3MSK is set to 1 by RTCRST, and the other bits are cleared to 0 (disable). Accordingly, the GPIO(2:0) interrupt can not be used for activation immediately after the RTCRST. The GPIO activation interrupt is valid only when the CPU operation mode is in Hibernate mode.

Note When 1 is written to the HALTIMERRST bit, HALTimer is reset, however, it may not be cleared if 0 is written again after 1 is written to the HALTIMERRST bit. Therefore it is prohibited to write 0 again after writing 1 to the HALTIMERRST bit. (Write 1.)

12.2.3 PMUINT2REG (0x0F00 00C4)

Bit	15	14	13	12	11	10	9	8
Name	GPIO12 INTR	GPIO11 INTR	GPIO10 INTR	GPIO9 INTR	RFU	RFU	RFU	RFU
R/W	R/W	R/W	R/W	R/W	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15	GPIO12INTR	GPIO12 activation interrupt request detection. Cleared to 0 when 1 is written. 1: Detected 0: Not detected
14	GPIO11INTR	GPIO11 activation interrupt request detection. Cleared to 0 when 1 is written. 1: Detected 0: Not detected
13	GPIO10INTR	GPIO10 activation interrupt request detection. Cleared to 0 when 1 is written. 1: Detected 0: Not detected
12	GPIO9INTR	GPIO9 activation interrupt request detection. Cleared to 0 when 1 is written. 1: Detected 0: Not detected
11:0	RFU	Reserved. Write 0. 0 is returned after a read.

This register is used to specify whether the GPIO(12:9) interrupts are detected as startup factors.

12.2.4 PMUCNT2REG (0x0F00 00C6)

Bit	15	14	13	12	11	10	9	8
Name	GPIO12 MSK	GPIO11 MSK	GPIO10 MSK	GPIO9 MSK	GPIO12 TRG	GPIO11 TRG	GPIO10 TRG	GPIO9 TRG
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	SOFTRST	RFU	RFU	RFU	RFU
R/W	R	R	R	R/W	R	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15	GPIO12MSK	GPIO12 activation enable 1: Enable 0: Disable
14	GPIO11MSK	GPIO11 activation enable 1: Enable 0: Disable
13	GPIO10MSK	GPIO10 activation enable 1: Enable 0: Disable
12	GPIO9MSK	GPIO9 activation enable 1: Enable 0: Disable
11	GPIO12TRG	Sets the GPIO12 activation interrupt request detection trigger 1: Falling edge 0: Rising edge
10	GPIO11TRG	Sets the GPIO11 activation interrupt request detection trigger 1: Falling edge 0: Rising edge
9	GPIO10TRG	Sets the GPIO10 activation interrupt request detection trigger 1: Falling edge 0: Rising edge
8	GPIO9TRG	Sets the GPIO9 activation interrupt request detection trigger 1: Falling edge 0: Rising edge
7:5	RFU	Reserved. Write 0. 0 is returned after a read.
4	SOFTRST	Software reset switch setting 1: Setting 0: Normal
3:0	RFU	Reserved. Write 0. 0 is returned after a read.

Note Holds the value before reset.

This register performs various settings for activation by the GPIO(12:9) interrupt and sets the software reset switch.

The GPIO(12:9)MSK bits are used to set enable/disable for activation from Hibernate mode when the corresponding interrupt (GPIO(12:9)) occurs. All mask bits are cleared to 0 (disable) by RTCRST. Therefore, GPIO(12:9) interrupts cannot be used for activation immediately after the RTCRST reset. Additionally, the GPIO activation interrupt is valid only when the CPU operation mode is Hibernate mode.

The SOFTRST bit specifies reset by software. By writing 1 to this bit, the CPU and peripherals are reset. This reset is the same operation as when RSTSW is pressed.

12.2.5 PMUWAITREG (0x0F00 00C8)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	WCOUNT 13	WCOUNT 12	WCOUNT 11	WCOUNT 10	WCOUNT 9	WCOUNT 8
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	1	0	1	1	0	0
After reset	0	0	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	WCOUNT 7	WCOUNT 6	WCOUNT 5	WCOUNT 3	WCOUNT 3	WCOUNT 2	WCOUNT 1	WCOUNT 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15:14	RFU	Reserved. Write 0. 0 is returned after a read.
13:0	WCOUNT (13:0)	Activation wait time timer count value Activation wait time = (WCOUNT(13:0) + 1) × (1/32.768) ms

Note Holds the value before reset.

This register sets the activation wait time at CPU activation.

This register is set to 0x2C00 (343.78 ms) after an RTC reset. Therefore, a 343.78 ms wait time is always inserted as the activation wait time when the CPU is activated immediately after an RTC reset. The activation wait time can be changed by setting this register for the CPU activation from Hibernate mode, etc.

When this register is set to 0x0, 0x1, 0x2, 0x3, or 0x4, the operation is not guaranteed. Be sure to set a value of 0x5 or greater to this register.

12.2.6 PMUTCLKDIVREG (0x0F00 00CC)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	TDIV
R/W	R	R	R	R	R	R	R	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	**Note**

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	VTDIV2	VTDIV1	VTDIV0
R/W	R	R	R	R	R	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	**Note**	**Note**	**Note**

Bit	Name	Function
15:9	RFU	Reserved. Write 0. 0 is returned after a read.
8	TDIV	TCLK division ratio setting 1: 1/4 VTClock 0: 1/2 VTClock
7:3	RFU	Reserved. Write 0. 0 is returned after a read.
2:0	VTDIV(2:0)	VTCLK division ratio setting 111: RFU 110: 1/6 PCLOCK 101: 1/5 PCLOCK 100: 1/4 PCLOCK 011: 1/3 PCLOCK 010: 1/2 PCLOCK 001: RFU 000: The same mode as the CLKSEL(2:0) setting during RTC reset

Note Holds the value before reset.

This register is used to set the CPU core's DIV mode. The DIV mode setting determines the division rate of the VTClock in relation to the pipeline clock (PClock) frequency.

Since the contents of this register are cleared to 0 during RTC reset, the DIV mode setting immediately after an RTC reset is set as shown in Table 12-6, based on the CLKSEL(2:0) status.

Once the DIV mode has been set via this register, the setting does not become valid immediately in the processor's operations. It becomes valid after a reset other than an RTC reset occurs.

The relationship between CLKSEL settings and various clock frequency values is shown in Table 16-6 below.

Table 12-6. CLKSEL(2:0) Pins and Frequency of PClock, VTClock, TClock and MasterOut (Default Value)

CLKSEL(2:0)	PClock (MHz)	VTClock (MHz)	TClock (MHz)	MasterOut (MHz)
111	RFU	RFU	RFU	RFU
110	199.1	33.2	16.6	4.15
101	181.0	30.2	15.1	3.775
100	165.9	33.2	16.6	4.15
011	153.1	30.6	15.3	3.825
010	132.7	33.2	16.6	4.15
001	99.5	33.2	16.6	4.15
000	RFU	RFU	RFU	RFU

12.2.7 PMUINTRCLKDIVREG (0x0F00 00CE)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	IDIV1	IDIV0
R/W	R	R	R	R	R	R	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	**Note**	**Note**

Bit	Name	Function
15:2	RFU	Reserved. Write 0. 0 is returned after a read.
1:0	IDIV(1:0)	intrclk clock division ratio setting. 11: RFU 10: 1/4 01: 1/8 00: 1/2

Note Holds the value before reset.

This register is used to set the division ratio of the divider that determines the intrclk clock frequency. The IDIV mode setting determines the division ratio of the intrclk clock frequency in relation to the seclk frequency (18.432 MHz).

Since the contents of this register are cleared to 0 during RTC reset, the intrclk clock division ratio immediately after an RTC reset is set to 1/2 of seclk.

Once the IDIV mode has been set via this register, the setting does not become valid immediately in the processor's operations. It becomes valid after a reset other than an RTC reset occurs.

CHAPTER 13 RTC (REALTIME CLOCK UNIT)

13.1 General

The RTC unit has a total of four timers, including the following three types.

- RTC Long timer .... This is a 24-bit programmable counter that counts down using 32.768 kHz frequency. The RTC unit includes two RTC Long timers. Cycle interrupts occur for up to 512 seconds.

- TClock counter .... This is a 25-bit programmable counter that counts down using VTClock cycles. Cycle interrupts occur for up to 1 to 2 seconds by setting the CLKSEL bit. This counter is used for performance evaluation.

- Elapsed Time timer.. This is a 48-bit counter that counts up using 32.768 kHz frequency. It counts up to 272 years before returning to zero. It includes 48-bit comparators (ECMPHREG, ECMPLREG, and ECMPMREG) and 48-bit alarm time registers (ETIMELREG, ETIMEMREG, and ETIMEHREG) to enable interrupts to occur at specified times.

13.2 Register Set

The following table shows the RTC registers.

Table 13-1. RTC Registers

Address	R/W	Register Symbol	Function
0x0F00 0100	R/W	ETIMELREG	Elapsed Time timer lower register
0x0F00 0102	R/W	ETIMEMREG	Elapsed Time timer middle register
0x0F00 0104	R/W	ETIMEHREG	Elapsed Time timer higher register
0x0F00 0108	R/W	ECMPLREG	Elapsed Time timer lower compare register
0x0F00 010A	R/W	ECMPMREG	Elapsed Time timer middle compare register
0x0F00 010C	R/W	ECMPHREG	Elapsed Time timer higher compare register
0x0F00 0110	R/W	RTCL1LREG	RTC Long1 timer lower register
0x0F00 0112	R/W	RTCL1HREG	RTC Long1 timer higher register
0x0F00 0114	R	RTCL1CNTLREG	RTC Long1 timer lower count register
0x0F00 0116	R	RTCL1CNTHREG	RTC Long1 timer higher count register
0x0F00 0118	R/W	RTCL2LREG	RTC Long2 timer lower register
0x0F00 011A	R/W	RTCL2HREG	RTC Long2 timer higher register
0x0F00 011C	R	RTCL2CNTLREG	RTC Long2 timer lower count register
0x0F00 011E	R	RTCL2CNTHREG	RTC Long2 timer higher count register
0x0F00 0120	R/W	TCLKLREG	TClock counter lower register
0x0F00 0122	R/W	TCLKHREG	TClock counter higher register
0x0F00 0124	R	TCLKCNTLREG	TClock counter lower count register
0x0F00 0126	R	TCLKCNTHREG	TClock counter higher count register
0x0F00 013E	R/W	RTCINTREG	RTC interrupt register

Each register is described in detail below.

13.2.1 Elapsed Time timer registers

(1) ETIMELREG (0x0F00 0100)

Bit	15	14	13	12	11	10	9	8
Name	ETIME15	ETIME14	ETIME13	ETIME12	ETIME11	ETIME10	ETIME9	ETIME8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	ETIME7	ETIME6	ETIME5	ETIME4	ETIME3	ETIME2	ETIME1	ETIME0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15:0	ETIME(15:0)	Elapsed Time timer bit 15:0

Note Continues counting.

(2) ETIMEMREG (0x0F00 0102)

Bit	15	14	13	12	11	10	9	8
Name	ETIME31	ETIME30	ETIME29	ETIME28	ETIME27	ETIME26	ETIME25	ETIME24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	ETIME23	ETIME22	ETIME21	ETIME20	ETIME19	ETIME18	ETIME17	ETIME16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15:0	ETIME(31:16)	Elapsed Time timer bit 31:16

Note Continues counting.

(3) ETIMEHREG (0x0F00 0104)

Bit	15	14	13	12	11	10	9	8
Name	ETIME47	ETIME46	ETIME45	ETIME44	ETIME43	ETIME42	ETIME41	ETIME40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	ETIME39	ETIME38	ETIME37	ETIME36	ETIME35	ETIME34	ETIME33	ETIME32
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15:0	ETIME(47:32)	Elapsed Time timer bit 47:32

Note Continues counting

These registers indicate the Elapsed Time timer’s value. They count up using a 32.768 kHz frequency. When a match occurs with the Elapsed Time compare registers, an alarm (Elapsed Time interrupt) occurs (and the count-up continues). The setting is valid once values have been written to all registers (ETIMELREG, ETIMEMREG, and ETIMEHREG).

These registers have no buffers for read. Therefore, an illegal data may be read if the counter value changes during a read operation. When using a read data, be sure to read a value twice and check that the two read values are the same.

When setting these registers again, wait until at least 100 μs (32.768 kHz clock × 3) have elapsed.

13.2.2 Elapsed Time timer compare registers

(1) ECMPLREG (0x0F00 0108)

Bit	15	14	13	12	11	10	9	8
Name	ECMP15	ECMP14	ECMP13	ECMP12	ECMP11	ECMP10	ECMP9	ECMP8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	ECMP7	ECMP6	ECMP5	ECMP4	ECMP3	ECMP2	ECMP1	ECMP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15:0	ECMP(15:0)	Value to be compared with Elapsed Time timer bit 15:0

Note Value before reset is retained.

(2) ECMPMREG (0x0F00 010A)

Bit	15	14	13	12	11	10	9	8
Name	ECMP31	ECMP30	ECMP29	ECMP28	ECMP27	ECMP26	ECMP25	ECMP24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	ECMP23	ECMP22	ECMP21	ECMP20	ECMP19	ECMP18	ECMP17	ECMP16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15:0	ECMP(31:16)	Value to be compared with Elapsed Time bit 31:16

Note Value before reset is retained.

(3) ECMPHREG (0x0F00 010C)

Bit	15	14	13	12	11	10	9	8
Name	ECMP47	ECMP46	ECMP45	ECMP44	ECMP43	ECMP42	ECMP41	ECMP40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	1	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	ECMP39	ECMP38	ECMP37	ECMP36	ECMP35	ECMP34	ECMP33	ECMP32
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15:0	ECMP(47:32)	Value to be compared with Elapsed Time timer bit 47:32

Note Value before reset is retained.

Set the values to be compared with values in the Elapsed Time timer registers to these registers.
The setting is valid once values have been written to all registers (ECMPLREG, ECMPMREG, and ECMPHREG).
When setting these registers again, wait until at least 100 μs (32.768 kHz clock × 3) have elapsed.

13.2.3 RTC Long1 timer registers

(1) RTCL1LREG (0x0F00 0110)

Bit	15	14	13	12	11	10	9	8
Name	RTCL1P15	RTCL1P14	RTCL1P13	RTCL1P12	RTCL1P11	RTCL1P10	RTCL1P9	RTCL1P8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	RTCL1P7	RTCL1P6	RTCL1P5	RTCL1P4	RTCL1P3	RTCL1P2	RTCL1P1	RTCL1P0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15:0	RTCL1P(15:0)	Bit 15:0 for RTC Long1 timer cycle

Note Value before reset is retained.

(2) RTCL1HREG (0x0F00 0112)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	RTCL1P23	RTCL1P22	RTCL1P21	RTCL1P20	RTCL1P19	RTCL1P18	RTCL1P17	RTCL1P16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15:8	RFU	Reserved. Write 0. 0 is returned after a read.
7:0	RTCL1P(23:16)	Bit 23:16 for RTC Long1 timer cycle

Note Value before reset is retained.

Set the RTC Long1 timer cycle to these registers. The RTC Long1 timer begins its countdown at the value written to these registers.

The setting is valid once values have been written to both registers (RTCL1LREG and RTCL1HREG).

When setting these registers again, wait until at least 100 μs (32.768 kHz clock × 3) have elapsed.

Cautions 1. The RTC Long1 timer is stopped when zeros are written to all bits.

2. Any combined setting of “RTCL1HREG = 0x0000” and “RTCL1LREG = 0x0001, 0x0002, 0x0003, 0x0004” is prohibited.

13.2.4 RTC Long1 timer count registers

(1) RTCL1CNTLREG (0x0F00 0114)

Bit	15	14	13	12	11	10	9	8
Name	RTCL1C15	RTCL1C14	RTCL1C13	RTCL1C12	RTCL1C11	RTCL1C10	RTCL1C9	RTCL1C8
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	RTCL1C7	RTCL1C6	RTCL1C5	RTCL1C4	RTCL1C3	RTCL1C2	RTCL1C1	RTCL1C0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15:0	RTCL1C(15:0)	RTC Long1 timer bit 15:0

Note Continues counting.

(2) RTCL1CNTHREG (0x0F00 0116)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	RTCL1C23	RTCL1C22	RTCL1C21	RTCL1C20	RTCL1C19	RTCL1C18	RTCL1C17	RTCL1C16
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15:8	RFU	Reserved. Write 0. 0 is returned after a read.
7:0	RTCL1C(23:16)	RTC Long1 timer bit 23:16

Note Continues counting.

These registers indicate the RTC Long1 timer's values. The countdown uses a 32.768 kHz frequency and begins at the value set to the RTC Long1 timer registers. An RTC Long1 interrupt occurs when the counter reaches 0x00 0001 (at which point the counter returns to the start value and continues counting).

These registers have no buffers for read. Therefore, an illegal data may be read if the counter value changes during a read operation. When using a read data, be sure to read a value twice and check that the two read values are the same.

13.2.5 RTC Long2 timer registers

(1) RTCL2LREG (0x0F00 0118)

Bit	15	14	13	12	11	10	9	8
Name	RTCL2P15	RTCL2P14	RTCL2P13	RTCL2P12	RTCL2P11	RTCL2P10	RTCL2P9	RTCL2P8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	RTCL2P7	RTCL2P6	RTCL2P5	RTCL2P4	RTCL2P3	RTCL2P2	RTCL2P1	RTCL2P0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15:0	RTCL2P(15:0)	Bit 15:0 for RTC Long2 timer cycle

Note Value before reset is retained.

(2) RTCL2HREG (0x0F00 011A)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	RTCL2P23	RTCL2P22	RTCL2P21	RTCL2P20	RTCL2P19	RTCL2P18	RTCL2P17	RTCL2P16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15:8	RFU	Reserved. Write 0. 0 is returned after a read.
7:0	RTCL2P(23:16)	Bit 23:16 for RTC Long2 timer cycle

Note Value before reset is retained.

Set the RTC Long2 timer cycle to these registers. The RTC Long2 timer begins its countdown at the value written to these registers.

The setting is valid once values have been written to both registers (RTCL2LREG and RTCL2HREG).

When setting these registers again, wait until at least 100 μs (32.768 kHz clock × 3) have elapsed.

Cautions 1. The RTC Long2 timer is stopped when zeros are written to all bits.

2. Any combined setting of “RTCL2HREG = 0x0000” and “RTCL2LREG = 0x0001, 0x0002, 0x0003, 0x0004” is prohibited.

13.2.6 RTC Long2 timer count registers

(1) RTCL2CNTLREG (0x0F00 011C)

Bit	15	14	13	12	11	10	9	8
Name	RTCL2C15	RTCL2C14	RTCL2C13	RTCL2C12	RTCL2C11	RTCL2C10	RTCL2C9	RTCL2C8
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	RTCL2C7	RTCL2C6	RTCL2C5	RTCL2C4	RTCL2C3	RTCL2C2	RTCL2C1	RTCL2C0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15:0	RTCL2C(15:0)	RTC Long2 timer counter bit 15:0

Note Continues counting.

(2) RTCL2CNTHREG (0x0F00 011E)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	RTCL2C23	RTCL2C22	RTCL2C21	RTCL2C20	RTCL2C19	RTCL2C18	RTCL2C17	RTCL2C16
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15:8	RFU	Reserved. Write 0 to these bits. 0 is returned after a read.
7:0	RTCL2C(23:16)	RTC Long2 timer counter bit 23:16

Note Continues counting.

These registers indicate the RTC Long2 timer’s values. The countdown uses a 32.768 kHz frequency and begins at the value set to the RTC Long2 timer registers. An RTC Long2 interrupt occurs when the counter reaches 0x00 0001 (at which point the counter returns to the start value and continues counting).

These registers have no buffers for read. Therefore, an illegal data may be read if the counter value changes during a read operation. When using a read data, be sure to read a value twice and check that the two read values are the same.

13.2.7 TClock counter registers

(1) TCLKLREG (0x0F00 0120)

Bit	15	14	13	12	11	10	9	8
Name	TCLKP15	TCLKP14	TCLKP13	TCLKP12	TCLKP11	TCLKP10	TCLKP9	TCLKP8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	TCLKP7	TCLKP6	TCLKP5	TCLKP4	TCLKP3	TCLKP2	TCLKP1	TCLKP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:0	TCLKP(15:0)	Bit 15:0 for TClock counter cycle

(2) TCLKHREG (0x0F00 0122)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	TCLKP24
R/W	R	R	R	R	R	R	R	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	TCLKP23	TCLKP22	TCLKP21	TCLKP20	TCLKP19	TCLKP18	TCLKP17	TCLKP16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:9	RFU	Reserved. Write 0. 0 is returned after a read.
8:0	TCLKP(24:16)	Bit 24:16 for TClock counter cycle

Set the TClock counter cycle to these registers. The TClock counter begins its countdown at the value written to these registers.

The setting is valid once values have been written to both registers (TCLKLREG and TCLKHREG).

Caution The TClock counter is stopped when zeros are written to all bits.

13.2.8 TClock counter count registers

(1) TCLKCNTLREG (0x0F00 0124)

Bit	15	14	13	12	11	10	9	8
Name	TCLKC15	TCLKC14	TCLKC13	TCLKC12	TCLKC11	TCLKC10	TCLKC9	TCLKC8
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	TCLKC7	TCLKC6	TCLKC5	TCLKC4	TCLKC3	TCLKC2	TCLKC1	TCLKC0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:0	TCLKC(15:0)	TClock counter bit 15:0

(2) TCLKCNTHREG (0x0F00 0126)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	TCLKC24
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	TCLKC23	TCLKC22	TCLKC21	TCLKC20	TCLKC19	TCLKC18	TCLKC17	TCLKC16
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:9	RFU	Reserved. Write 0. 0 is returned after a read.
8:0	TCLKC(24:16)	TClock counter bit 24:16

These registers indicate the TClock counter's values. The countdown uses a VTClock cycle and begins its countdown at the value written to the TClock counter registers. A TClock counter interrupt occurs when the counter reaches 0x000 0001 (at which point the counter returns to the start value and continues counting).

These registers have no buffers for read. Therefore, an illegal data may be read if the counter value changes during a read operation. When using a read data, be sure to read a value twice and check that the two read values of the higher 9 bits are the same.

13.2.9 RTC interrupt register

(1) RTCINTREG (0x0F00 013E)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RTCINTR3	RTCINTR2	RTCINTR1	RTCINTR0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	**Note**	**Note**	**Note**

Bit	Name	Function
15:4	RFU	Reserved. Write 0. 0 is returned after a read.
3	RTCINTR3	TClock counter interrupt request. Cleared to 0 when 1 is written. 1: Occurred 0: Normal
2	RTCINTR2	RTC Long2 timer interrupt request. Cleared to 0 when 1 is written. 1: Occurred 0: Normal
1	RTCINTR1	RTC Long1 timer interrupt request. Cleared to 0 when 1 is written. 1: Occurred 0: Normal
0	RTCINTR0	Elapsed Time timer interrupt request. Cleared to 0 when 1 is written. 1: Occurred 0: Normal

Note Value before reset is retained.

This register is used to set/indicate the occurrences of RTC interrupt requests.

CHAPTER 14 GIU (GENERAL-PURPOSE I/O UNIT)

14.1 Outline

The GIU controls the GPIO and DCD# pins. The GPIO pins are general-purpose ports for which input and output are available. An interrupt request signal input function can be assigned to GPIO with input signal change (rising edge or falling edge of signal), low level, or high level used as the trigger.

Table 14-1 shows the types of clock and input buffers used for the interrupt request detection of the GPIO pins.

Table 14-1. GPIO Pins

Pin	Interrupt Request Detection Clock (Internal)	Input Buffer Type
GPIO(31:16)	Intrclk	Normal
GPIO15(DCD#)	Intrclk	Normal
GPIO(13:0)	Intrclk	Schmitt

Caution The function in GPIO15 is fixed to the DCD# input signal.
It cannot be used as a general-purpose I/O pin.

When not used for an interrupt request, the registers corresponding to these pins can be written to output a low-level or high-level signal. Each register can be read to check the state of the signal currently being input to the corresponding pin.

The GPIO pins can be used as transition factors from the Standby, Suspend, or Hibernate mode to the Fullspeed mode. When the same setting as for generating an interrupt request is made the operation returns from the Standby mode or Suspend mode when the GPIO(31:0) pin becomes active. When these pins are specified as startup factors by the PMU registers and the GPIO(12:9) and GPIO(3:0) pins become active, the operation returns from Hibernate mode.

Interrupt requests are acknowledged from the SIU as well as the GIU for GPIO15(DCD#).

Mask requests as required.

14.2 Register Set

The GIU registers are listed below.

Table 14-2. GIU Registers

Address	R/W	Register Symbol	Function
0x0F00 0140	R/W	GIUIOSELL	GPIO input/output setting lower register
0x0F00 0142	R/W	GIUIOSELH	GPIO input/output setting higher register
0x0F00 0144	R/W	GIUPIODL	GPIO input/output data lower register
0x0F00 0146	R/W	GIUPIODH	GPIO input/output data higher register
0x0F00 0148	R/W	GIUINTSTATL	GPIO interrupt lower register
0x0F00 014A	R/W	GIUINTSTATH	GPIO interrupt higher register
0x0F00 014C	R/W	GIUINTENL	GPIO interrupt enable lower register
0x0F00 014E	R/W	GIUINTENH	GPIO interrupt enable higher register
0x0F00 0150	R/W	GIUINTTYPL	GPIO interrupt trigger setting lower register
0x0F00 0152	R/W	GIUINTTYPH	GPIO interrupt trigger setting higher register
0x0F00 0154	R/W	GIUINTALSELL	GPIO interrupt level setting lower register
0x0F00 0156	R/W	GIUINTALSELH	GPIO interrupt level setting higher register
0x0F00 0158	R/W	GIUINTHTSELL	GPIO interrupt hold lower register
0x0F00 015A	R/W	GIUINTHTSELH	GPIO interrupt hold higher register
0x0F00 015C	R/W	GIUPODATEN	GPIO output enable register
0x0F00 015E	R/W	GIUPODATL	GPIO output data lower register

14.2.1 GIUIOSELL (0x0F00 0140)

Bit	15	14	13	12	11	10	9	8
Name	IOS15	RFU	IOS13	IOS12	IOS11	IOS10	IOS9	IOS8
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	IOS7	IOS6	IOS5	IOS4	IOS3	IOS2	IOS1	IOS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15	IOS15	Selects GPIO15(DCD#) pin input/output. 1: RFU 0: Input
14	RFU	Reserved. Write 0. 0 is returned after a read.
13:0	IOS(13:0)	Selects GPIO(13:0) pin input/output. 1: Output 0: Input

This register sets the I/O mode of the GPIO(15:0) pins. The correspondence is 1 bit per pin.

When the IOS bit is set to 1, the corresponding GPIO pin is set to output and the value that has been written to the corresponding PIOD bit in the GIUPIODL register is output.

When this bit is set to 0, the corresponding GPIO pin is set to input.

Caution Since GPIO15(DCD#) is fixed as input, the IOS15 bit cannot be set to output.

14.2.2 GIUIOSELH (0x0F00 0142)

Bit	15	14	13	12	11	10	9	8
Name	IOS31	IOS30	IOS29	IOS28	IOS27	IOS26	IOS25	IOS24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	IOS23	IOS22	IOS21	IOS20	IOS19	IOS18	IOS17	IOS16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:0	IOS(31:16)	Selects GPIO(31:16) pin input/output. 1: Output 0: Input

This register sets the I/O mode of the GPIO(31:16) pins. The correspondence is 1 bit per pin.

When the IOS bit is set to 1, the corresponding GPIO pin is set to output and the value that has been written to the corresponding PIOD bit in the GIUPIODH register is output.

When this bit is set to 0, the corresponding GPIO pin is set to input.

14.2.3 GIUPIODL (0x0F00 0144)

Bit	15	14	13	12	11	10	9	8
Name	PIOD15	RFU	PIOD13	PIOD12	PIOD11	PIOD10	PIOD9	PIOD8
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	PIOD7	PIOD6	PIOD5	PIOD4	PIOD3	PIOD2	PIOD1	PIOD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15	PIOD15	Specifies GPIO15(DCD#) pin output data. 1: RFU 0: Low level
14	RFU	Reserved. Write 0. 0 is returned after a read.
13:0	PIOD(13:0)	Specifies GPIO(13:0) pin output data. 1: High level 0: Low level

This register reads and/or writes the GPIO(15:0) pins. The correspondence is 1 bit per pin.

When 1 is set to the corresponding IOS bit in the GIUIOSELL register, the data written to the PIOD bit is output via the corresponding GPIO pin.

When the value of the corresponding IOS bit in the GIUIOSELL register is 0, writing a value to the PIOD bit does not affect the GPIO pin (the write data is ignored).

When the value of the corresponding bit in the GIUIOSELL register is 0, reading the PIOD bit enables the corresponding GPIO pin's state to be read.

Caution Since the GPIO15(DCD#) pin is fixed as input, data cannot be written to PIOD15.

14.2.4 GIUPIODH (0x0F00 0146)

Bit	15	14	13	12	11	10	9	8
Name	PIOD31	PIOD30	PIOD29	PIOD28	PIOD27	PIOD26	PIOD25	PIOD24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	PIOD23	PIOD22	PIOD21	PIOD20	PIOD19	PIOD18	PIOD17	PIOD16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:0	PIOD(31:16)	Specifies GPIO(31:16) pin output data. 1: High level 0: Low level

This register reads and/or writes the GPIO(31:16) pins. The correspondence is 1 bit per pin.

When 1 is set to the corresponding IOS bit in the GIUIOSELH register, the data written to the PIOD bit is output via the corresponding GPIO pin.

When the value of the corresponding IOS bit in the GIUIOSELH register is 0, writing a value to the PIOD bit does not affect the GPIO pin (the write data is ignored).

When the value of the IOS bit in the GIUIOSELH register is 0, reading the PIOD bit enables the corresponding GPIO pin’s state to be read.

14.2.5 GIUINTSTATL (0x0F00 0148)

Bit	15	14	13	12	11	10	9	8
Name	INTS15	RFU	INTS13	INTS12	INTS11	INTS10	INTS9	INTS8
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	INTS7	INTS6	INTS5	INTS4	INTS3	INTS2	INTS1	INTS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15	INTS15	Interrupt request to GPIO(15) pin. Cleared to 0 when 1 is written. 1: Interrupt occurred 0: No interrupt
14	RFU	Reserved. Write 0. 0 is returned after a read.
13:0	INTS(13:0)	Interrupt request to GPIO(13:0) pin. Cleared to 0 when 1 is written. 1: Interrupt occurred 0: No interrupt

This register indicates the interrupt request status of the GPIO(15:0) pins. The correspondence is 1 bit per pin.

1 is set to the corresponding INTS bit when the signal input to the GPIO pin meets the condition set via the GIUINTTYPL register or the GIUINTALSELL register. Even if the corresponding bit is set to 1, however, no interrupt occurs when the corresponding bit in the GIUINTENL register is set to 0 (disable interrupts).

If a GIPO pin is low as the default status, it is judged that the condition is met, and the corresponding bit is set to 1.

When using this register, it should be cleared to 0 once after the GIUINTTYPL and GIUINTALSELL registers are set to enable interrupts.

Caution The function in GPIO15 is fixed as the DCD# input signal.

14.2.6 GIUINTSTATH (0x0F00 014A)

Bit	15	14	13	12	11	10	9	8
Name	INTS31	INTS30	INTS29	INTS28	INTS27	INTS26	INTS25	INTS24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	INTS23	INTS22	INTS21	INTS20	INTS19	INTS18	INTS17	INTS16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:0	INTS(31:16)	Interrupt request to GPIO(31:16) pin. Cleared to 0 when 1 is written. 1: Interrupt occurred 0: No interrupt

This register indicates the interrupt request status of the GPIO pins. The correspondence is 1 bit per pin.

The corresponding INTS bit is set to 1 when the signal input to the GPIO pin meets the condition set via the GIUINTTYPH register or GIUINTALSELH register.

Even if the corresponding bit is set to 1, however, no interrupt occurs when the corresponding bit in the GIUINTENH register is set to 0 (disable interrupts).

If a GPIO pin is low as the default status, it is judged that the condition is met, and the corresponding bit is set to 1.

When using this register, it should be cleared to 0 once after the GIUINTTYPH and GIUINTALSELH registers are set to enable interrupts.

14.2.7 GIUINTENL (0x0F00 014C)

Bit	15	14	13	12	11	10	9	8
Name	INTE15	RFU	INTE13	INTE12	INTE11	INTE10	INTE9	INTE8
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	INTE7	INTE6	INTE5	INTE4	INTE3	INTE2	INTE1	INTE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15	INTE15	Interrupt enable to GPIO(15) pin. 1: Enable 0: Disable
14	RFU	Reserved. Write 0. 0 is returned after a read.
13:0	INTE(13:0)	Interrupt enable to GPIO(13:0) pins. 1: Enable 0: Disable

This register sets the GPIO(15:0) pins to interrupt enable. The correspondence is 1 bit per pin.
When 1 is set to the corresponding INTE bit, interrupts are enabled for the corresponding GPIO pins.

Caution The function in GPIO15 is fixed as the DCD# input signal.

14.2.8 GIUINTENH (0x0F00 014E)

Bit	15	14	13	12	11	10	9	8
Name	INTE31	INTE30	INTE29	INTE28	INTE27	INTE26	INTE25	INTE24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	INTE23	INTE22	INTE21	INTE20	INTE19	INTE18	INTE17	INTE16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:0	INTE(31:16)	Interrupt enable for GPIO(31:16) pins. 1: Enable 0: Disable

This register sets the GPIO(31:16) pins to the interrupt enabled status. The INTE(31:16) bits correspond to the GPIO(31:16) pins.

When the corresponding INTE bit is set to 1, interrupts are enabled for the corresponding GPIO pins.

14.2.9 GIUINTTYPL (0x0F00 0150)

Bit	15	14	13	12	11	10	9	8
Name	INTT15	RFU	INTT13	INTT12	INTT11	INTT10	INTT9	INTT8
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	INTT7	INTT6	INTT5	INTT4	INTT3	INTT2	INTT1	INTT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15	INTT15	Interrupt request detection trigger to GPIO(15) pin.
14	RFU	Reserved. Write 0. 0 is returned after a read.
13:0	INTT(13:0)	Interrupt request detection trigger for GPIO(13:0) pins. 1: Edge 0: Level

This register sets the trigger to detect an interrupt request for the GPIO(15:0) pins. The correspondence is 1 bit per pin.

When the INTT bit is set to 1, an interrupt request signal for the corresponding GPIO pin is detected as an interrupt request when the signal state changes from high to low or low to high.

When 0 is set, the level set to the corresponding INTL bit in the GIUINTALSELL register is detected as an interrupt request.

Caution The function in GPIO15 is fixed as the DCD# input signal.

14.2.10 GIUINTTYPH (0x0F00 0152)

Bit	15	14	13	12	11	10	9	8
Name	INTT31	INTT30	INTT29	INTT28	INTT27	INTT26	INTT25	INTT24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	INTT23	INTT22	INTT21	INTT20	INTT19	INTT18	INTT17	INTT16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:0	INTT(31:16)	Interrupt request detection trigger 1: Edge 0: Level

This register sets the trigger to detect an interrupt request for the GPIO(31:16) pins. The correspondence is 1 bit per pin.

When the INTT bit is set to 1, an interrupt request signal for the corresponding GPIO pin is detected as an interrupt request when the signal state changes from high to low or low to high.

When 0 is set, the level set to the corresponding INTL bit in the GIUINTALSELH register is detected as an interrupt request.

14.2.11 GIUINTALSELL (0x0F00 0154)

Bit	15	14	13	12	11	10	9	8
Name	INTL15	RFU	INTL13	INTL12	INTL11	INTL10	INTL9	INTL8
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	INTL7	INTL6	INTL5	INTL4	INTL3	INTL2	INTL1	INTL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15	INTL15	Interrupt request detection level to GPIO(15)
14	RFU	Reserved. Write 0. 0 is returned after a read.
13:0	INTL(13:0)	Interrupt request detection level to GPIO(13:0) 1: High level 0: Low level

This register sets the level for the detection of interrupt requests to the GPIO(15:0) pins. The correspondence is 1 bit per pin.

When selecting an edge trigger via the GIUINTTYPL register, the contents of this register are invalid.

To use this register, be sure to set the level trigger using the GIUINTTYPL register.

Caution The function in GPIO15 is fixed as the DCD# input signal.

14.2.12 GIUINTALSELH (0x0F00 0156)

Bit	15	14	13	12	11	10	9	8
Name	INTL31	INTL30	INTL29	INTL28	INTL27	INTL26	INTL25	INTL24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	INTL23	INTL22	INTL21	INTL20	INTL19	INTL18	INTL17	INTL16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:0	INTL(31:16)	Interrupt request detection level 1: High level 0: Low level

This register sets the level for the detection of interrupt requests to the GPIO(31:16) pins. The correspondence is 1 bit per pin.

When selecting an edge trigger via the GIUINTTYPH register, the contents of this register are invalid.

To use this register, be sure to set the level trigger using the GIUINTTYPH register.

14.2.13 GIUINTHTSELL (0x0F00 0158)

Bit	15	14	13	12	11	10	9	8
Name	INTH15	RFU	INTH13	INTH12	INTH11	INTH10	INTH9	INTH8
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	INTH7	INTH6	INTH5	INTH4	INTH3	INTH2	INTH1	INTH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15	INTH15	Sets GPIO(15) pin interrupt request signal hold.
14	RFU	Reserved. Write 0. 0 is returned after a read.
13:0	INTH(13:0)	Sets GPIO(13:0) pin interrupt request signal hold. 1: Hold 0: Through

This register sets whether or not interrupt signals to the GPIO(15:0) pins should be held. The correspondence is 1 bit per pin.

When the INTH bit is set to 1, any interrupt signal input to the corresponding GPIO pin is held, and when this bit is set to 0, an interrupt signal input to the corresponding GPIO pin is not held. Any held interrupt signal is cleared when the corresponding INTS bit in the GIUINTSTATL register is set to 1.

INTH(15:0) are not affected by GIUINTENL register.

If the INTH bit is set to 1 (hold) while the INTE bit of the GIUINTENL is set to 0 (interrupt disable), any change in the pin state is retained as change data. Therefore, an interrupt occurs when the INTE bit is set to 1 (interrupt enable).

Caution The function in GPIO15 is fixed as the DCD# input signal.

14.2.14 GIUINTHTSELH (0x0F00 015A)

Bit	15	14	13	12	11	10	9	8
Name	INTH31	INTH30	INTH29	INTH28	INTH27	INTH26	INTH25	INTH24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	INTH23	INTH22	INTH21	INTH20	INTH19	INTH18	INTH17	INTH16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:0	INTH(31:16)	Sets GPIO(31:16) pin interrupt request signal hold. 1: Hold 0: Through

This register sets whether or not interrupt signals to the GPIO(31:16) pins should be held. The correspondence is 1 bit per pin.

When the INTH bit is set to 1, any interrupt signal input to the corresponding GPIO pin is held, and when this bit is set to 0, an interrupt signal input to the corresponding GPIO pin is not held. Any held interrupt signal is cleared when the corresponding INTS bit in the GIUINTSTATH register is set to 1.

INTH(31:16) are not affected by the GIUINTENH register.

If the INTH bit is set to 1 (hold) while the INTE bit of the GIUINTENH is set to 0 (interrupt disable), any change in the pin state is retained as change data. Therefore, an interrupt occurs when the INTE bit is set to 1 (interrupt enable).

The relationship between the GPIO interrupt enable/disable and hold settings is as follows.

Table 14-3. Correspondence Between Interrupt Mask and Interrupt Hold

Interrupt Trigger	Setting of GIUINTHTSEL Register	Setting of GIUINTEN Register	Hold in GIU	Notation to ICU
Level	Hold	Masked	Held	Not reported
		Not masked	Held	Reported
		Masked → canceled	Held	Reported
	Through	Masked	Not held	Not reported
		Not masked	Not held	Reported
		Masked → canceled	Not held	Not reported
Edge	Hold	Masked	Held	Not reported
		Not masked	Held	Reported
		Masked → canceled	Held	Reported
	Through	Masked	Not held	Not reported
		Not masked	Setting prohibited	Setting prohibited
		Masked → canceled	Not held	Not reported

14.2.15 GIUPODATEN (0x0F00 015C)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	PIOEN35	PIOEN34	PIOEN33	PIOEN32
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15:4	RFU	Reserved. Write 0. 0 is returned after a read.
3:0	PIOEN(35:32)	GPIO(35:32) pin output enable 1: Enable 0: Disable

Note Value before reset is retained.

This register sets the GPIO(35:32) pins to interrupt enable. The correspondence is 1 bit per pin.

When the PIOEN bit is set to 1, the corresponding GPIO pin becomes output and the value written to the PIOD bit corresponding to the GIUPODATL register is output. When 0 is set, the value of the GIUPODATL register does not affect the GPIO pin.

The GPIO(35:32) pins support output-only.

Use of the GPIO(35:32) pins is mutually exclusive to use of the debug serial interface. When using these pins as debug serial interface pins, set the PIOEN bit to 0. In addition, the GPIO(33:32) pins function alternately as initial setting pins and they change to the input state during RTC reset.

The following table shows the correspondence with alternative function pins.

Table 14-4. Correspondence Between GPIO(35:32) and Alternate Function Pins

GPIO Pin	PIOEN Bit	Alternate Function Pin
GPIO35	PIOEN35	DCTS#
GPIO34	PIOEN34	DDIN
GPIO33	PIOEN33	DRTS#/MIPS16EN
GPIO32	PIOEN32	DDOUT/DBUS32

14.2.16 GIUPODATL (0x0F00 015E)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	PIOD35	PIOD34	PIOD33	PIOD32
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15:4	RFU	Reserved. Write 0. 0 is returned after a read.
3:0	PIOD(35:32)	GPIO(35:32) pin output data 1: High level 0: Low level

Note Value before reset is retained.

This register sets the GPIO(35:32) pins to the output level. The correspondence is 1 bit per pin.

When the corresponding PIOEN bit of the GIUPODATEN register is set to 1, data written to the PIOD bit is output to the GPIO pin. Even if a value is written to the PIOD bit when the PIOEN bit is set to 0, the GPIO pins are not affected. (Written data is ignored.)

The set value can be read by reading the PIOD bit.

CHAPTER 15 SCU (SysAD CONTROL UNIT)

15.1 Outline

The SCU performs bus arbitration so that multiple masters can be connected on the SysAD bus. The main specifications of the SCU are as follows:

- Bus arbitration from multiple bus masters
- Detection of access to the invalid address space
- Timeout detection
- Discard of a detected invalid bus cycle and issuance of a dummy cycle

15.2 Register Set

Table 15-1. SCU Registers

Address	R/W	Register Symbol	Function
0x0F00 1000	R/W	TIMOUTCNTREG	Timeout value setting register
0x0F00 1002	R/W	TIMOUTCOUNTREG	Timeout counter register
0x0F00 1004	R/W	ERRLADDRESSREG	Error lower address register
0x0F00 1006	R/W	ERRHADDRESSREG	Error higher address register
0x0F00 1008	R/W	SCUINTRREG	SCU interrupt request register

Each register is described in detail below.

15.2.1 TIMOUTCNTREG (0x0F00 1000)

Bit	15	14	13	12	11	10	9	8
Name	TIMOUT15	TIMOUT14	TIMOUT13	TIMOUT12	TIMOUT11	TIMOUT10	TIMOUT9	TIMOUT8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	TIMOUT7	TIMOUT6	TIMOUT5	TIMOUT4	TIMOUT3	TIMOUT2	TIMOUT1	TIMOUTE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:1	TIMOUT(15:1)	Sets the value of the timeout detection.
0	TIMOUTE	Enables the timeout detection. 1: Enable 0: Disable

15.2.2 TIMOUTCOUNTREG (0x0F00 1002)

Bit	15	14	13	12	11	10	9	8
Name	TIMCNT15	TIMCNT14	TIMCNT13	TIMCNT12	TIMCNT11	TIMCNT10	TIMCNT9	TIMCNT8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	TIMCNT7	TIMCNT6	TIMCNT5	TIMCNT4	TIMCNT3	TIMCNT2	TIMCNT1	TIMCNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:0	TIMCNT(15:0)	Sets the value of a timeout counter. This counter synchronizes with VTClock. A timeout is detected when TIMCNT(15:1) matches the TIMOUT(15:1) bit of the TIMOUTCNTREG register.

15.2.3 ERRLADDRESSREG (0x0F00 1004)

Bit	15	14	13	12	11	10	9	8
Name	ERRADR15	ERRADR14	ERRADR13	ERRADR12	ERRADR11	ERRADR10	ERRADR9	ERRADR8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	ERRADR7	ERRADR6	ERRADR5	ERRADR4	ERRADR3	ERRADR2	ERRADR1	ERRADR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:0	ERRADR(15:0)	Holds the lower 16 bits of the physical address of the last CPU core timeout or accesses the reserved space.

15.2.4 ERRHADDRESSREG (0x0F00 1006)

Bit	15	14	13	12	11	10	9	8
Name	ERRADR31	ERRADR30	ERRADR29	ERRADR28	ERRADR27	ERRADR26	ERRADR25	ERRADR24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	ERRADR23	ERRADR22	ERRADR21	ERRADR20	ERRADR19	ERRADR18	ERRADR17	ERRADR16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:0	ERRADR(31:16)	Holds the higher 16 bits of the physical address of the last CPU core timeout or accesses the reserved space.

15.2.5 SCUINTREG (0x0F00 1008)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	TIMOERR	RSVERR
R/W	R	R	R	R	R	R	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:6	RFU	Reserved. Write 0. 0 is returned after a read.
5:2	RFU	Reserved. Write 0. Value is undefined after a read.
1	TIMOERR	CPU core timeout error
0	RSVERR	CPU core access error to the reserved space

15.2.6 Notification of illegal access

(1) Types of illegal access

The VR4131 notifies the CPU core of the occurrence of an illegal access in the following cases using a non-maskable interrupt request (bus error exception) at reading or a maskable interrupt request (Int0) by SCUINTRREG at writing.

- Bus deadlock

  When the ready signal is not returned during issuance of the processor read request to the external I/O space and PCI space and a bus time-out occurs, the VR4122 determines that it is a deadlock and acknowledges the occurrence of an illegal access. The interval is determined by the value of the TIMOUTCNTREG setting.

- Reserved address space for future

  When the processor accesses the following physical addresses, the VR4122 acknowledges the occurrence of an illegal access.

  0x0EFF FFFF to 0x0E00 0000
  0x09FF FFFF to 0x0800 0000

(2) Notification of illegal access

The following table shows how the CPU core is notified:

Table 15-2. Notification of Illegal Access

Access Type	Notification of Illegal Access
Processor read request	Bus error acknowledgement via SysCmd bus Interrupt exception (Int0) acknowledgement
Processor write request	Interrupt exception (Int0) acknowledgement (RSVERR only)

CHAPTER 16 SDRAMU (SDRAM CONTROL UNIT)

16.1 General

The SDRAMU is a unit that arbitrates between the SysAD bus and SDRAM.

16.2 Register Set

The following table shows the SDRAMU registers.

Table 16-1. SDRAM Registers

Address	R/W	Register Symbol	Function
0x0F00 0400	R/W	SDRAMMODEREG	SDRAM mode register
0x0F00 0402	R/W	SDRAMCNTREG	SDRAM control register
0x0F00 0404	R/W	BCURFCNTREG	BCU refresh control register
0x0F00 0406	R/W	BCURFCOUNTREG	BCU refresh cycle count register
0x0F00 0408	R/W	RAMSIZEREG	DRAM size register

These registers are described in detail below.

16.2.1 SDRAMMODEREG (0x0F00 0400)

Bit	15	14	13	12	11	10	9	8
Name	SCLK	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R/W	R	R	R	R	R	R	R
RTCRST	1	0	0	0	0	0	0	0
After reset	1	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	LTMODE2	LTMODE1	LTMODE0	WT	BL2	BL1	BL0
R/W	R	R/W	R/W	R/W	R	R	R	R
RTCRST	0	0	1	1	1	0	0	1
After reset	0	0	1	1	1	0	0	1

Bit	Name	Function
15	SCLK	Sets the SCLK output of the ADD25/SCLK pin. 1: SCLK is always output. 0: SCLK is output only when accessing SDRAM.
14:7	RFU	Reserved. Write 0. 0 is returned after a read.
6:4	LTMODE(2:0)	Sets the /CAS latency of SDRAM. 111: RFU 110: RFU 101: RFU 100: RFU 011: 3 010: 2 001: RFU 000: RFU
3	WT	Indicates the address ordering. 1: Interleave 0: RFU
2:0	BL(2:0)	Indicates the burst length. 111: RFU : 010: RFU 001: 2 000: RFU

This register sets or indicates the operation mode when SDRAM is used.

16.2.2 SDRAMCNTREG (0x0F00 0402)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	TRC3	TRC2	TRC1	TRC0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	1	0	1	0	0	1
After reset	0	0	1	0	1	0	0	1

Bit	7	6	5	4	3	2	1	0
Name	RFU	TDAL2	TDAL1	TDAL0	RFU	TRCD2	TRCD1	TRCD0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
RTCRST	0	1	0	0	0	1	0	0
After reset	0	1	0	0	0	1	0	0

Bit	Name	Function
15:12	RFU	Reserved. Write 0. 0 is returned after a read.
11:8	TRC(3:0)	Sets the minimum interval of the following cycles: Bank active → Bank active/Self-refresh/CBR refresh, self-refresh → Bank active/Self-refresh/CBR refresh, and CBR refresh → Bank active/Self-refresh/CBR refresh. 1111: RFU : 1010: RFU 1001: 9VTClock 1000: 8VTClock 0111: 7VTClock 0110: 6VTClock 0101: 5VTClock 0100: 4VTClock 0011: 3VTClock 0010: RFU : 0000: RFU
7	RFU	Reserved. Write 0. 0 is returned after a read.
6:4	TDAL(2:0)	Sets the minimum interval from one clock after the last data output of the write command with auto-precharge to the bank active. 111: RFU 110: RFU 101: RFU 100: 4VTClock 011: 3VTClock 010: 2VTClock 001: RFU 000: RFU
3	RFU	Reserved. Write 0. 0 is returned after a read.

(2/2)

Bit	Name	Function
2:0	TRCD(2:0)	Sets the minimum interval of Bank active → Read/Read with auto-precharge/Write/ write with auto-precharge 111: RFU 110: RFU 101: RFU 100: 4VTClock 011: 3VTClock 010: 2VTClock 001: RFU 000: RFU

Set the value of TRC(3:0) according to TRC/TRC1 of the SDRAM to be used.
Set the value of TDAL(2:0) according to TDAL of the SDRAM to be used.
Set the value of TRCD(2:0) according to TRCD/TRAS of the SDRAM to be used.

16.2.3 BCURFCNTREG (0x0F00 0404)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	BRF13	BRF12	BRF11	BRF10	BRF9	BRF8
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	1	0
After reset	0	0	0	0	0	0	1	0

Bit	7	6	5	4	3	2	1	0
Name	BRF7	BRF6	BRF5	BRF4	BRF3	BRF2	BRF1	BRF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:14	RFU	Write 0. 0 is returned after a read.
13:0	BRF(13:0)	DRAM refresh cycle count (VTClock count)

The refresh interval is calculated as follows:

Refresh interval = BRF(13:0) × VTClock

Therefore calculate the setting value based on the DRAM refresh cycle count and bus access cycle (each address space/bus hold cycle) that are used. Refer to the **CLKSPEEDREG register** for the frequency of VTClock.

16.2.4 BCURFCOUNTREG (0x0F00 0406)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	BRFC13	BRFC12	BRFC11	BRFC10	BRFC9	BRFC8
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	BRFC7	BRFC6	BRFC5	BRFC4	BRFC3	BRFC2	BRFC1	BRFC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	1
After reset	0	0	0	0	0	0	0	1

Bit	Name	Function
15:14	RFU	Reserved. Write 0. 0 is returned after a read.
13:0	BRFC(13:0)	Current DRAM refresh cycle count

This register indicates the current DRAM refresh cycle count.

When the value becomes 0x0000, this register generates the refresh cycle request and sets the value of the BCURFCNTREG register. The counter operates whether the refresh cycle occurs or not.

This register decrements in synchronization with VTClock. Therefore the refresh cycle can be calculated by the setting value of the VTDIVMODE(2:0) bit in the CLKSPEEDREG register.

16.2.5 RAMSIZEREG (0x0F00 0408)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	RFU	SIZE32	SIZE31	SIZE30	RFU	SIZE22	SIZE21	SIZE20
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
RTCRST	0	**Note**	**Note**	**Note**	0	**Note**	**Note**	**Note**
After reset	0	**Note**	**Note**	**Note**	0	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	RFU	SIZE12	SIZE11	SIZE10	RFU	SIZE02	SIZE01	SIZE00
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
RTCRST	0	**Note**	**Note**	**Note**	0	**Note**	**Note**	**Note**
After reset	0	**Note**	**Note**	**Note**	0	**Note**	**Note**	**Note**

<table>
<tr><th>Bit</th><th>Name</th><th>Function</th></tr>
<tr><td>15</td><td>RFU</td><td>Reserved. Write 0. 0 is returned after a read.</td></tr>
<tr><td>14:12</td><td>SIZE3(2:0)</td><td>Sets the RAM size of bank 3 in 16-bit mode or bank 3 in 32-bit mode.
<table>
<tr><th>SIZE3(2:0)</th><th>16-bit mode
(MB)</th><th>32-bit mode
(MB)</th></tr>
<tr><td>111</td><td>RFU</td><td>RFU</td></tr>
<tr><td>110</td><td>RFU</td><td>RFU</td></tr>
<tr><td>101</td><td>RFU</td><td>64</td></tr>
<tr><td>100</td><td>32</td><td>32</td></tr>
<tr><td>011</td><td>16</td><td>16</td></tr>
<tr><td>010</td><td>8</td><td>8</td></tr>
<tr><td>001</td><td>RFU</td><td>RFU</td></tr>
<tr><td>000</td><td>RFU</td><td>RFU</td></tr>
</table></td></tr>
<tr><td>11</td><td>RFU</td><td>Reserved. Write 0. 0 is returned after a read.</td></tr>
<tr><td>10:8</td><td>SIZE2(2:0)</td><td>Sets the RAM size of bank 2 in 16-bit mode or bank 2 in 32-bit mode.
<table>
<tr><th>SIZE2(2:0)</th><th>16-bit mode
(MB)</th><th>32-bit mode
(MB)</th></tr>
<tr><td>111</td><td>RFU</td><td>RFU</td></tr>
<tr><td>110</td><td>RFU</td><td>RFU</td></tr>
<tr><td>101</td><td>RFU</td><td>64</td></tr>
<tr><td>100</td><td>32</td><td>32</td></tr>
<tr><td>011</td><td>16</td><td>16</td></tr>
<tr><td>010</td><td>8</td><td>8</td></tr>
<tr><td>001</td><td>RFU</td><td>RFU</td></tr>
<tr><td>000</td><td>RFU</td><td>RFU</td></tr>
</table></td></tr>
</table>

Note According to the bit width of the data bus and RAM type, values are set as follows (n = 0 to 3):

In 16-bit mode: SIZEn(2:0) = 100
In 32-bit mode: SIZEn(2:0) = 101

(2/2)

<table>
<tr><th>Bit</th><th>Name</th><th>Function</th></tr>
<tr><td>7</td><td>RFU</td><td>Reserved. Write 0. 0 is returned after a read.</td></tr>
<tr><td>6:4</td><td>SIZE1(2:0)</td><td>Sets the RAM size of bank 1 in 16-bit mode or bank 1 in 32-bit mode.
<table>
<tr><th>SIZE1(2:0)</th><th>16-bit mode
(MB)</th><th>32-bit mode
(MB)</th></tr>
<tr><td>111</td><td>RFU</td><td>RFU</td></tr>
<tr><td>110</td><td>RFU</td><td>RFU</td></tr>
<tr><td>101</td><td>RFU</td><td>64</td></tr>
<tr><td>100</td><td>32</td><td>32</td></tr>
<tr><td>011</td><td>16</td><td>16</td></tr>
<tr><td>010</td><td>8</td><td>8</td></tr>
<tr><td>001</td><td>RFU</td><td>RFU</td></tr>
<tr><td>000</td><td>RFU</td><td>RFU</td></tr>
</table></td></tr>
<tr><td>3</td><td>RFU</td><td>Reserved. Write 0. 0 is returned after a read.</td></tr>
<tr><td>2:0</td><td>SIZE0(2:0)</td><td>Sets the RAM size of bank 0 in 16-bit mode or bank 0 in 32-bit mode.
<table>
<tr><th>SIZE0(2:0)</th><th>16-bit mode
(MB)</th><th>32-bit mode
(MB)</th></tr>
<tr><td>111</td><td>RFU</td><td>RFU</td></tr>
<tr><td>110</td><td>RFU</td><td>RFU</td></tr>
<tr><td>101</td><td>RFU</td><td>64</td></tr>
<tr><td>100</td><td>32</td><td>32</td></tr>
<tr><td>011</td><td>16</td><td>16</td></tr>
<tr><td>010</td><td>8</td><td>8</td></tr>
<tr><td>001</td><td>RFU</td><td>RFU</td></tr>
<tr><td>000</td><td>RFU</td><td>RFU</td></tr>
</table></td></tr>
</table>

This register sets the size of each bank of the DRAM space.
The size should be set with a larger size to the lower banks or the same size to all banks.

SIZE0(2:0) ≥ SIZE1(2:0) ≥ SIZE2(2:0) ≥ SIZE3(2:0)

16.2.6 DRAM connection

The following table shows an example of the connection between the VR4131 and DRAM.

From the ADD pins (ADD(24:10)) of the VR4131 the row address (RAS signal) and column address (CAS signal) are output to DRAM (address multiplex mode supported) according to the RAS signal (RAS#) or CAS signal (CAS#). The physical addresses for each output address are as follows.

Table 16-2. Example of SDRAM Connection and Addresses Output from the VR4131

SDRAM Address Pin	ADD Pin	16-Bit Data Bus Mode (DBUS32 = 0)		32-Bit Data Bus Mode (DBUS32 = 1)	
		Row	Column	Row	Column
BA1	ADD24	adr18	adr18	adr19	adr19
BA0	ADD23	adr17	adr17	adr18	adr18
A12	ADD22	adr24	adr24	adr25	adr25
A11	ADD21	adr22	adr22	adr23	adr23
A10	ADD20	adr21	CMD	adr22	CMD
A9	ADD19	adr20	adr20	adr21	adr21
A8	ADD18	adr19	adr23	adr20	adr24
A7	ADD17	adr16	adr8	adr17	adr9
A6	ADD16	adr15	adr7	adr16	adr8
A5	ADD15	adr14	adr6	adr15	adr7
A4	ADD14	adr13	adr5	adr14	adr6
A3	ADD13	adr12	adr4	adr13	adr5
A2	ADD12	adr11	adr3	adr12	adr4
A1	ADD11	adr10	adr2	adr11	adr3
A0	ADD10	adr9	adr1	adr10	adr2

Remark adr(25:1): Physical address bits in CPU core or DMAAU

16.2.7 SDRAM interface

The SCU supports bus operations for SDRAM. When accessing SDRAM in the VR4131 the following features are available:

- The CAS latency can be selected from the two types of 2 and 3 (set using the SDRAMMODREG register).
- The burst mode is fixed to interleave.
- The burst length is fixed to 2.
- A command with auto-precharge is issued at the end of each read/write cycle.

The following two bus operations for SDRAM are available.

- Single read/write (1 byte, 2 bytes, 3 bytes, 1 word (= 4 bytes))
- Block read/write (2 words, 4 words, 8 words)

The SDRAM interface specifies the following parameters. These parameters are set using the SDRAMCNTREG register.

- TRCD: Cycle count from the active command to the read/write command.
- TDAL: Cycle count from the last write data to the active command.
- TRC: Cycle count from refresh to refresh or active command, or cycle count from self-refresh to refresh or active command.

(1) Single write cycle

This section describes the timing when two successive 1-word write operations are performed. Commands for the first and second 1-word write are issued at <1> and <2>, and <3> and <4> in Figure 16-1, respectively.

(a) First 1-word write

<1> Active command issued.

<2> Write command with auto-precharge issued. DQM(3:0)# going high at the rise of the next clock after the command was issued causes the write data to be masked.

(b) Second 1-word write

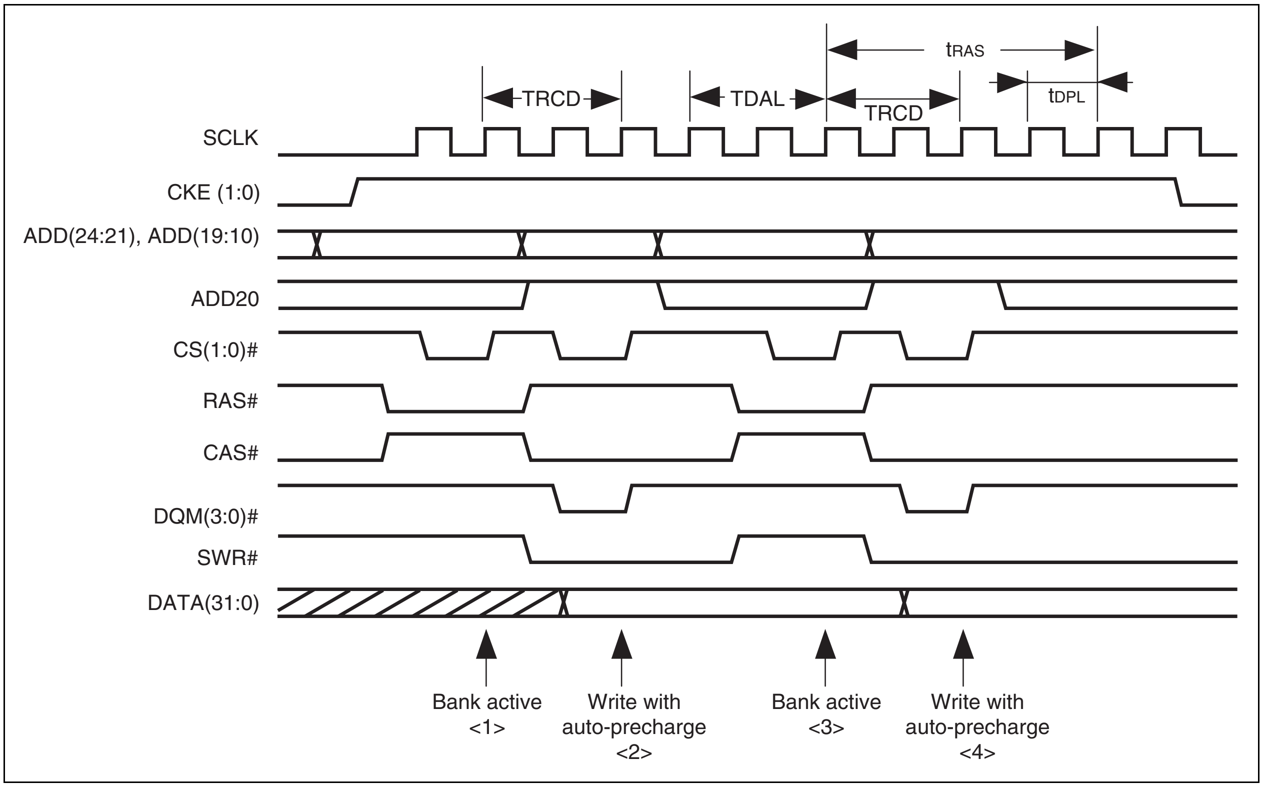
<3> Active command issued.

<4> Command with auto-precharge issued.

(C) Cautions when setting TRCD

Consider not only t_{RCD} of the SDRAM to be connected but the value of t_{RAS} (bank active → precharge) when setting the TRCD value using the SDRAMCNTREG register.

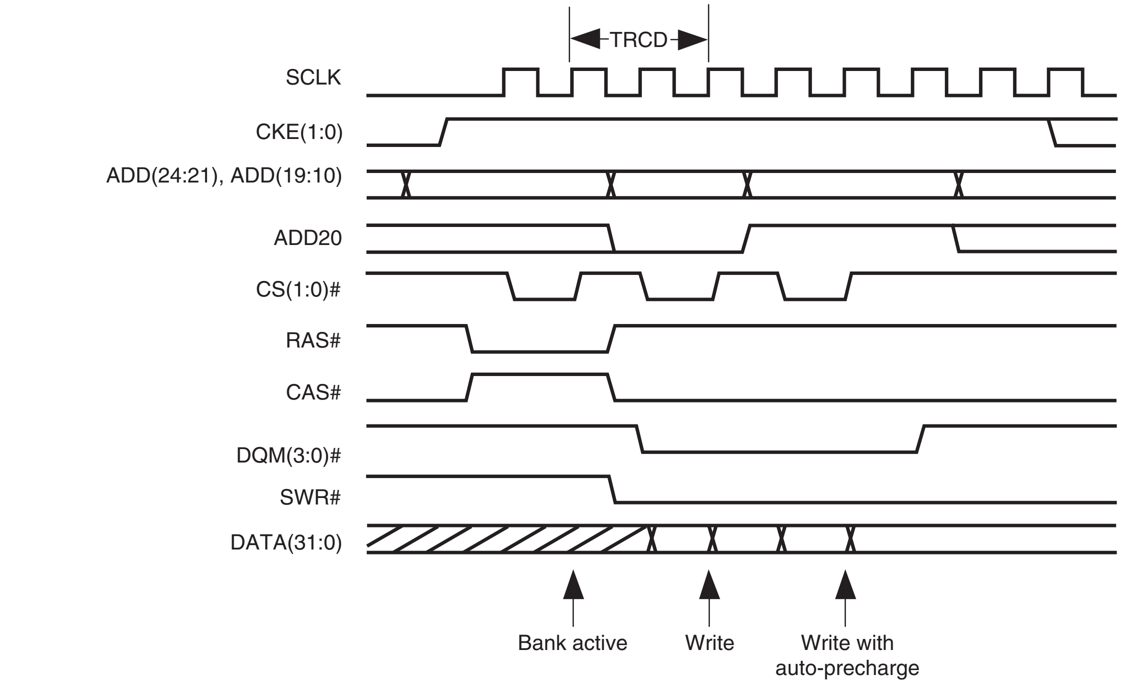
Figure 16-1. Successive Write Cycle in 32-Bit Bus Mode (TRCD(2:0) = 010, TDAL(2:0) = 010)



(2) Block write cycle

This section describes the timing of the block write cycle. The VR4131 issues a write command with auto-precharge at the last bus cycle and performs precharge. For this operation the VR4122 issues a write command between the bank active command and write command with auto-precharge.

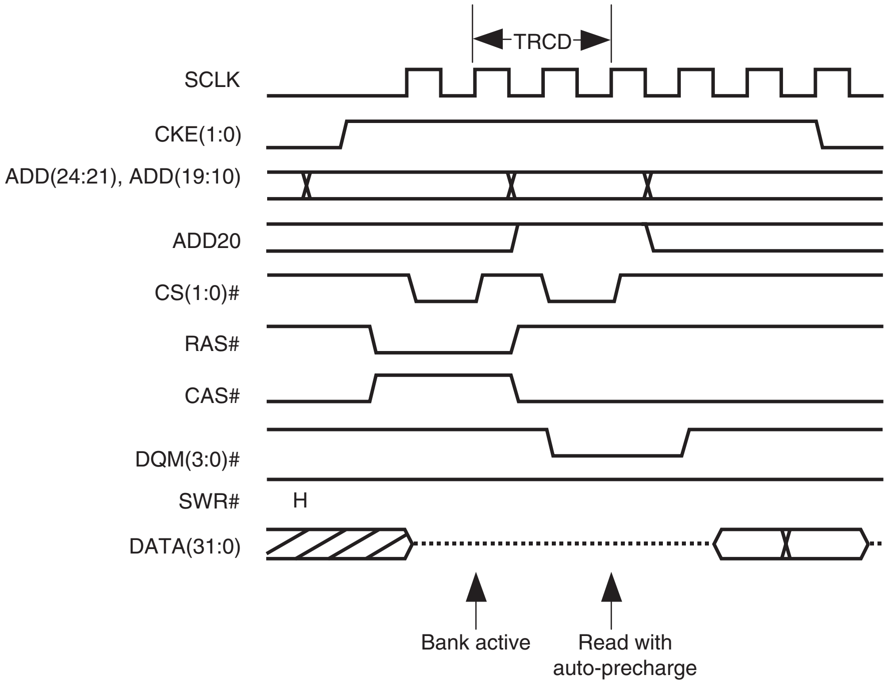
Figure 16-2. Block Write Cycle in 32-Bit Bus Mode (TRCD(2:0) = 010, CAS Latency = 2)



(3) Single read cycle

This section describes the timing of the single read cycle.

Figure 16-3. Single Read Cycle in 32-Bit Bus Mode (TRCD(2:0) = 010, TDAL(2:0) = 010, CAS Latency = 2)

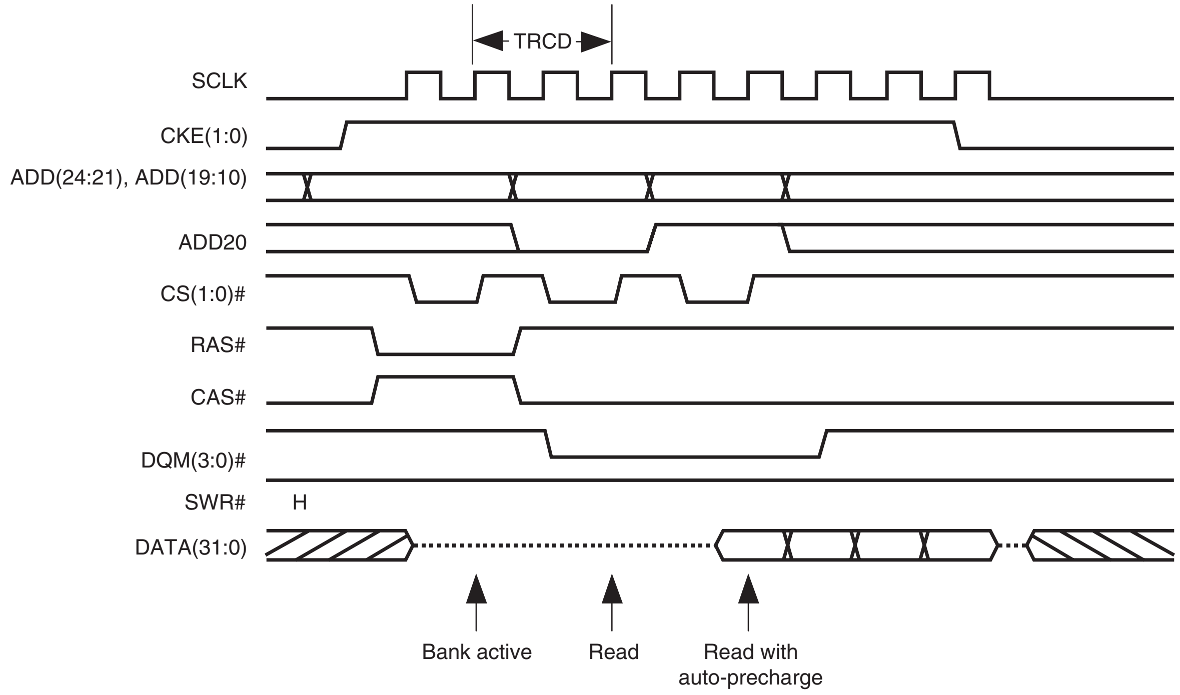


Remark Dotted lines indicate high-impedance.

(4) Block read cycle

This section describes the timing of the block read cycle. The VR4131 issues a read command with auto-precharge at the last bus cycle and performs precharge. For this operation the VR4122 issues a read command between the bank active command and read command with auto-precharge.

Figure 16-4. Block Read Cycle in 32-Bit Bus Mode (TRCD(2:0) = 010, CAS Latency = 2)



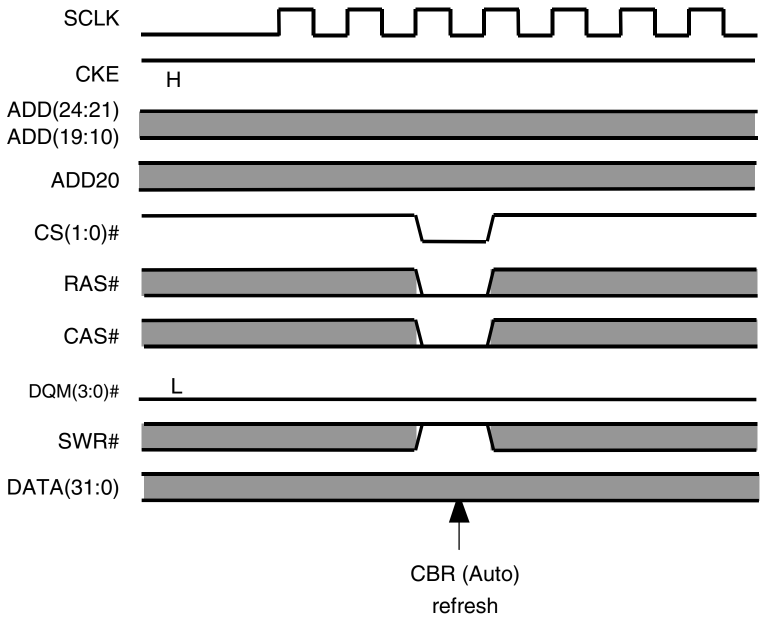
Remark Dotted lines indicate high-impedance.

16.2.8 Refresh

The VR4131 supports CBR refresh and self-refresh.

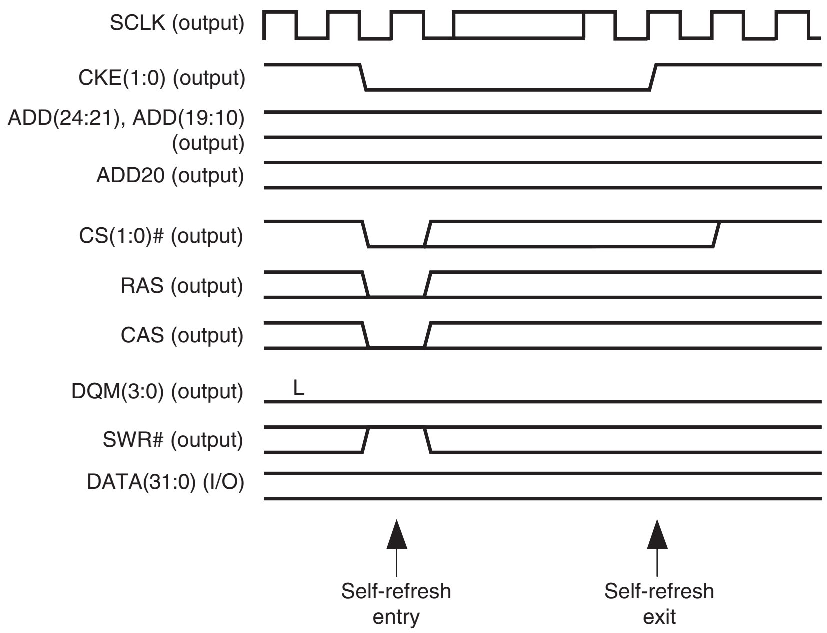
(1) CBR refresh

Figure 16-5. CBR (Auto) Refresh



(2) Self-refresh

Figure 16-6. Self-Refresh (Entry and Exit)



CHAPTER 17 PCIU (PCI CONTROL UNIT)

17.1 Overview

The PCI control unit can connect the internal bus of the VR4131 to the PCI bus. The following describes the bridge specifications.

17.2 Specifications

The features of the PCI interfacing of the PCI control unit are as follows:

- Conforms to PCI2.1 subset
- PCI transaction as a master or target
- PCI bus arbiter (three types of arbitration protocols) incorporated
- Up to three PCI devices can be connected
- Bidirectional 8-word length FIFO provided between the PCI and internal bus
- PCI clock: 33 MHz (maximum transfer rate: 132 MB/s)
- Number of PCI space address windows

As master	PCI memory space:	Two windows
	PCI I/O space:	One window
As slave	DRAM space:	Two windows

- Delayed transaction: If the PCIU operates as the target, the PCIU returns a retry following the reception of a command during read and resumes the cycle after fetching data on internal bus side.
- Posted transaction: If the PCIU operates as the target, the PCIU receives write data during write, holds write data within the PCIU, and then normally completes the cycle on PCI side. A write cycle is subsequently executed on the internal bus side.
- Generates interrupt after assertion of IRDY# if TRDY# remains deasserted for a fixed period. Period is set by the PCITRDYVREG register.

Note that the following functions are not supported.

- High-speed back-to-back transaction
- Interrupt control function
- On-chip CPU cache
- Special cycle
- Stepping
- Applications other than host bridge (such as mounting on add-on board)

17.3 Reset Signals

The RST# signal can be used for PCI interface reset.

The RST# signal is asserted by the following operation.

- Reset by RTCRST#
- Reset by RSTSWB
- HALTimer shutdown
- Software shutdown (HIBERNATE instruction)

When the BMAS bit of the COMMANDREG register is set to 1, the RST# signal can be deasserted.

The period between supplying PCLK by the MSKPPCIU bit of the CMUCLKMSK register and deasserting the RST# signal is recommended to be more than 100 μs.

17.4 Commands

The PCIU supports the following PCI commands.

Table 17-1. PCI Command List

CBEn(3:0)	Command	Master Operation	Target Operation
0000	Reserved	Ignored	Ignored
0001	Reserved	Ignored	Ignored
0010	I/O Read	Valid	Ignored
0011	I/O Write	Valid	Ignored
0100	Reserved	Ignored	Ignored
0101	Reserved	Ignored	Ignored
0110	Memory Read	Valid	Valid
0111	Memory Write	Valid	Valid
1000	Reserved	Ignored	Ignored
1001	Reserved	Ignored	Ignored
1010	Configuration Read	Valid	Ignored
1011	Configuration Write	Valid	Ignored
1100	Memory Read Multiple	Ignored	Aliased to Memory Read
1101	Reserved	Ignored	Ignored
1110	Memory Read Line	Ignored	Aliased to Memory Read
1111	Memory Write and invalidate	Ignored	Aliased to Memory Write

17.5 Conversion Between Big Endian and Little Endian on PCI Bus in Big Endian Mode

17.5.1 When VR4131 is master

Number of SysAD Data Transfer Bytes	SysAD(2:0) (Internal Bus, Address)	SysAD(31:0) (Internal Bus, Data)	CBE (3:0)	AD (31:0) (Data)
1 byte	x00 x01 x10 x11	ABCD	1110 1101 1011 0111	DCBA
2 bytes	x00 x10	ABCD	1100 0011	CDAB
3 bytes	Undefined			
4 bytes	x00	ABCD	0000	ABCD
8 bytes	x00	ABCD EFGH	0000 (2) 0000 (1)	EFGH ABCD
16 bytes	000	ABCD EFGH IJKL MNOP	0000 (1) 0000 (2) 0000 (3) 0000 (4)	ABCD EFGH IJKL MNOP
32 bytes	000	ABCD EFGH IJKL NMOP QRST UVWX YZ01	0000 (1) 0000 (2) 0000 (3) 0000 (4) 0000 (5) 0000 (6) 0000 (7) 0000 (8)	ABCD EFGH IJKL NMOP QRST UVWX YZ01

Remarks 1. x: Don't care

2. The numbers in the CBE(3:0) column indicate the order of the data output to the AD bus.

If the memory on the PCI bus is cache area, the PCI bus is accessed in big endian mode. In other words, SysAD(31:0) (Internal Bus, Data) and AD(31:0) (Data) in the table above are equal.

17.5.2 When VR4131 is target

Number of SysAD Data Transfer Bytes	SysAD(2:0) (Internal Bus, Address)	SysAD(31:0) (Internal Bus, Data)	CBE (3:0)	AD (31:0) (Data)
1 byte	x00 x01 x10 x11	DCBA	1110 1101 1011 0111	ABCD
2 bytes	x00 x10	CDAB	1100 0011	ABCD
3 bytes	Undefined			
4 bytes	x00	ABCD	0000	ABCD
Burst	x00	ABCD	0000[Note]	ABCD

Note Processed as 4 bytes of continuous data.

Remark x: Don't care

17.6 Transaction from Internal Bus to PCI (PCIU: PCI Master)

17.6.1 Address conversion

If an address of the internal bus space is output to the internal bus, the PCIU compares the higher bits of the address with the value preset in PCIMMAW1REG, PCIMMAW2REG, and PCIMIOAWREG. If they match, the higher bits are rewritten to the value preset to PCIMMAW1REG, PCIMMAW2REG, and PCIMIOAWREG, and the rewritten address is output to the PCI bus.

PCIMMAW1REG and PCIMMAW2REG are available for the PCI memory space, while PCIMIOAWREG is available for the PCI I/O space. The bit configuration of these registers is the same, and each register has mask bits (bits 19 to 13). These bits are used to change the address window size.

17.6.2 Example of setting address window

Suppose the 128 MB space of 0x1000 0000 to 0x17FF FFFF is used as an internal I/O space and this space is allocated to 0xA000 0000 to 0xA7FF FFFF of the PCI memory space. If PCIMMAW1REG is used, the setting is as follows:

Example of setting PCIMMAW1REG register (Set value: 0x170F 10A7)

31 24	23 20	19 13	12	11 8	7 0
IBA(31:24)	RFU	MSK(6:0)	WIN1EN	RFU	PCIA(31:24)
00010111	0000	1111000	1	0000	10100111

17.6.3 Command conversion

The command received from the internal bus side is converted as shown in Table 17-2 and transferred to the PCI bus. Only the read/write commands to the internal PCI memory space are valid while the PCIU receives a cycle on the internal bus. If an address on the internal bus hits a PCIMMAW1REG or PCIMMAW2REG of the PCI at this time, the command is converted into one for the memory cycle on the PCI; if the address hits PCIMIOAWREG of the PCI, the command is converted into one for an I/O cycle. If the command accesses the PCICONFDREG in the PCIU, it is converted into one for the configuration cycle. The command is then output to the PCI bus.

Table 17-2. Command Conversion from Internal Bus to PCI Bus

Internal Bus Command	Conversion to PCI Command		
	CBE(3:0)	Command	Note
Memory Read	0110	Memory Read	If an address from the internal bus hits the PCIMMAW1REG or PCIMMAW2REG register
	0010	I/O Read	If an address from the internal bus hits the PCIMIOAWREG register
	1010	Configuration Read	If the PCICONFDREG register is accessed for read
Memory Write	0111	Memory Write	If an address from the internal bus hits the PCIMMAW1REG or PCIMMAW2REG register
	0011	I/O Write	If an address from the internal bus hits the PCIMIOAWREG register
	1011	Configuration Write	If the PCICONFDREG register is accessed for write

17.6.4 Remarks

Observe the following precautions for transactions between the internal bus and PCI.

- When the memory is written or read, the lower 2 bits of the address sent from the internal bus side are fixed to 0 at the timing of converting to a PCI address.
- If the cycle is stopped because of a disconnection while the target on the PCI is being accessed, this PCIU resumes the cycle from the address at which the cycle was stopped.
- If the internal I/O area (0x0F00 0C00 to 0x0F00 0DFF) in the PCIU is included in the address window, a transaction performed in this area is interpreted as being for the register area, and is not transmitted to the PCI bus.
- The configuration access should be made in 1 word (32 bits).

17.7 Transaction from PCI to Internal Bus (PCIU: PCI Target)

17.7.1 Address conversion

The PCIU transfers a transaction to the SDRAM space (0x07FF FFFF to 0x0000 0000) by receiving a memory cycle from other master devices on the PCI, as a PCI target. When transferring a transaction to the SDRAM space from the PCI bus, two windows are used to convert the address. These windows are set by using PCITAW1REG and PCITAW2REG. However, the address on the PCI side is set by using PCIMBA1REG and PCIMBA2REG of the configuration header register.

17.7.2 Command conversion

The commands received from the PCI bus are converted as shown in Table 17-3 and transferred to the internal bus. The PCIU receives only the memory cycle from the PCI side. Because this unit does not execute a burst cycle of eight words or longer, it only transfers data up to the cache line length (eight words) even when it has received Memory Read Multiple. In addition, because this unit does not support caching, it regards Memory Write And Invalidate as being ordinary Memory Write even when it has received this command, and executes cycle transfer.

Even if a command other than the PCI commands listed in Table 17-3 has been received from the PCI bus, the PCIU does not acknowledge the PCI bus.

Table 17-3. Command Conversion from PCI Bus to Internal Bus

PCI Command		Conversion to Internal Command	
CBE(3:0)	Command	Command	Note
0110	Memory Read	Memory Read	–
1100	Memory Read Multiple		Only four words of data are actually transferred when Memory Read Multiple is received from the PCI bus.
1110	Memory Read Line		–
0111	Memory Write	Memory Write	–
1111	Memory Write And Invalidate		Because the PCIU does not support caching, the MWI command is recognized as being ordinary Memory Write.

17.7.3 Delayed transaction

Generally, when the PCIU operates as a target on the PCI bus, the PCIU can post memory write data or immediately transfers the memory read access cycle to the secondary bus by using its internal buffer. When posting is completed or the secondary bus cycle ends at this time, the bus cycle on the primary bus is completed. This type of protocol is called “immediate transaction”.

If the bus is held in the idle status for a long time because access to the secondary side is not ready even after a cycle has been executed on the primary bus, the PCIU stops the cycle and completes the cycle on the primary bus side as soon as the data is ready, taking the latency and use efficiency of the bus into consideration. This is called a “delayed transaction”.

The PCIU executes a delayed transaction when it receives a memory read cycle for the internal bus cycle from the PCI.

<1> A memory read cycle is issued from the PCI bus side to the VR4131.
<2> The PCIU latches an address and a command, and then issues a retry.
<3> The PCIU continues memory read on the internal bus side, and fetches eight words.
<4> If the cycle from the PCI bus side is the same (i.e., if the address and command match), the read data is passed and the cycle is completed. Otherwise, a retry is issued.

17.7.4 Posted transaction

In a memory read cycle, the cycle on the primary bus side is extended by a retry by using delayed transaction to prevent the use efficiency of the bus from dropping. In contrast, the cycle on the primary bus side is completed in the memory write cycle after write data has been transferred (posted) normally to the PCIU.

If a memory write cycle is received from the PCI in the internal cycle, the posted transaction is executed in the following sequence:

<1> The memory write cycle is issued from the PCI bus side to the VR4131.
<2> The PCIU normally completes all data (up to eight words) sent from the PCI master.
<3> The PCI bus side continues transferring the received data to the internal bus side even after the cycle has been completed on the PCI bus side. In the meantime, a retry response is returned to an access from the PCI bus side.

17.7.5 Precautions

Observe the following precautions for transactions between the PCI and internal bus.

- Notes on writing internal bus from the PCI side
  The data written to the internal bus from the PCI is regarded as consecutive byte data. If non-consecutive write data is received, the valid data in SDRAM may or may not be overwritten.
- Parity error
  If an illegal parity is detected in the data cycle, the parity error bit in the configuration space is set and an interrupt occurs if enabled. If an illegal parity is detected in the address cycle, SERR# is asserted. This is a limitation on byte enable from the PCI side.

17.8 Interrupts

PCIU has an interrupt output signal, INT_Z, which is asserted high and held high if any one of the interrupt sources listed in the table below occurs. The interrupt output and the status bit of the corresponding interrupt source are cleared if the CPU reads the INTCNTSTAREG register or writes 1 to the corresponding interrupt source clear bit.

Table 17-4. Interrupt Source List

Interrupt Source	Location of Interrupt Source	Description	Contents of BUSERRADREG
PCI Target Abort	PCI bus	If target abort is received from the target while the PCIU operates on the PCI bus as a master	PCI address
PCI Master Abort	PCI bus	If master abort occurs while the PCIU operates on the PCI bus as a master	PCI address
PCI Retry Limit Reached	PCI bus	If the retry count has reached the value set in RTYVAL(7:0) of the RETVALREG register while the PCIU operates on the PCI bus as a master	PCI address
PCI SERR	PCIU PCI bus	If the PCIU asserts SERR# or if SERR# is asserted by another device	–
Mailbox Access	PCIU	If the PCIMAILREG register is accessed	–
PCI Parity Error	PCI bus	If a parity error is detected in the data phase or if PERR# is asserted by the target while the PCIU operates as a master	PCI address
PCI TRDY Limit Reached	PCI bus	If TRDY# is not asserted for a fixed time after IRDY# has been asserted while the PCIU operates as a master	PCI address
IBus Master Abort	Internal bus	If master abort is generated because a slave does not respond while the PCIU operates on the internal bus as a master	PCI address at the interrupt destination

17.9 Internal Register Set

Table 17-5 lists the internal registers of the PCI.

Table 17-5. PCI Internal Registers

Address	R/W	Register Symbols	Function
0x0F00 0C00	R/W	PCIMMAW1REG	PCI memory space address conversion register 1 for master transaction
0x0F00 0C04	R/W	PCIMMAW2REG	PCI memory space address conversion register 2 for master transaction
0x0F00 0C08	R/W	PCITAW1REG	Internal memory space conversion register 1 for target transaction
0x0F00 0C0C	R/W	PCITAW2REG	Internal memory space conversion register 2 for target transaction
0x0F00 0C10	R/W	PCIMIOAWREG	PCI I/O space address conversion register for master transaction
0x0F00 0C14	R/W	PCICONFDREG	Data register for configuration access
0x0F00 0C18	R/W	PCICONFAREG	Address register for configuration access
0x0F00 0C1C	R/W	PCIMAILREG	Register for Mailbox
0x0F00 0C24	R	BUSERRADREG	Bus error occurrence address storage register
0x0F00 0C28	R/W	INTCNTSTAREG	Interrupt status indication and control register
0x0F00 0C2C	R/W	PCIEXACCREG	Exclusive access indication and control register
0x0F00 0C30	R	PCIRECONTREG	Retry count register
0x0F00 0C34	R/W	PCIENREG	Configuration register setting end register
0x0F00 0C38	R/W	PCICLKSELREG	PCI bus operation clock select register
0x0F00 0C3C	R/W	PCITRDYVREG	TRDY signal setting register
0x0F00 0C60	R/W	PCICLKRUNREG	CLK_RUN signal setting register

Each of these internal registers is detailed below.

17.9.1 PCIMMAW1REG (0x0F00 0C00)

Bit	31	30	29	28	27	26	25	24
Name	IBA31	IBA30	IBA29	IBA28	IBA27	IBA26	IBA25	IBA24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	1	0	0	0	0
After reset	0	0	0	1	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Name	RFU	RFU	RFU	RFU	MSK6	MSK5	MSK4	MSK3
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	1	1	1	1
After reset	0	0	0	0	1	1	1	1

Bit	15	14	13	12	11	10	9	8
Name	MSK2	MSK1	MSK0	WIN1EN	RFU	RFU	RFU	RFU
R/W	R/W	R/W	R/W	R/W	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	PCIA31	PCIA30	PCIA29	PCIA28	PCIA27	PCIA26	PCIA25	PCIA24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
31:24	IBA(31:24)	Internal bus base address. If bits 31 to 24 of the internal address match the value of the IBA bits, an access to the PCI bus is recognized. Make sure that the PCI space set to the IBA bits does not overlap a local resource address, such as that of memory or boot ROM, or a PCI slave window.
23:20	RFU	Reserved. Write 0 to these bits. 0 is returned when these bits are read.
19:13	MSK(6:0)	Address mask. Specify the bits of bits 30 to 24 of the internal address to be masked during address conversion. The window size is from 16 MB to 2 GB. 1: Not masked 0: Masked
12	WIN1EN	PCI access enable. 1: Enabled 0: Disabled
11:8	RFU	Reserved. Write 0 to these bits. 0 is returned when these bits are read.
7:0	PCIA(31:24)	PCI address setting. If the internal address is the same as the address set in the IBA field, it is converted into the address value set in these bits. The address having the higher bits converted is used as the memory space address of the PCI.

During master transaction (CPU → PCI), the contents of this register are used as a table that converts the internal address output by the CPU into a memory space address for the PCI.

17.9.2 PCIMMAW2REG (0x0F00 0C04)

Bit	31	30	29	28	27	26	25	24
Name	IBA31	IBA30	IBA29	IBA28	IBA27	IBA26	IBA25	IBA24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	1	0	0	0	0
After reset	0	0	0	1	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Name	RFU	RFU	RFU	RFU	MSK6	MSK5	MSK4	MSK3
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	1	1	1	1
After reset	0	0	0	0	1	1	1	1

Bit	15	14	13	12	11	10	9	8
Name	MSK2	MSK1	MSK0	WIN2EN	RFU	RFU	RFU	RFU
R/W	R/W	R/W	R/W	R/W	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	PCIA31	PCIA30	PCIA29	PCIA28	PCIA27	PCIA26	PCIA25	PCIA24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
31:24	IBA(31:24)	Internal bus base address. If bits 31 to 24 of the internal address match the value of the IBA bits, an access to the PCI bus is recognized. Make sure that the PCI space set to the IBA does not overlap a local resource address, such as that of memory or boot ROM, or a PCI slave window.
23:20	RFU	Reserved. Write 0 to these bits. 0 is returned when these bits are read.
19:13	MSK(6:0)	Address mask. Specify the bits of bits 30 to 24 of the internal address to be masked during address conversion. The window size is from 16 MB to 2 GB. 1: Not masked 0: Masked
12	WIN2EN	PCI access enable. 1: Enabled 0: Disabled
11:8	RFU	Reserved. Write 0 to these bits. 0 is returned when these bits are read.
7:0	PCIA(31:24)	PCI address setting. If the internal address is the same as the address set in the IBA field, it is converted into the address value set in these bits. The address having the higher bits converted is used as the memory space address of the PCI.

During master transaction (CPU → PCI), the contents of this register are used as a table that converts the internal address output by the CPU into a memory space address for the PCI.

17.9.3 PCITAW1REG (0x0F00 0C08)

Bit	31	30	29	28	27	26	25	24
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Name	RFU	RFU	RFU	RFU	MSK6	MSK5	MSK4	MSK3
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Name	MSK2	MSK1	MSK0	WIN1EN	RFU	ITA10	ITA9	ITA8
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	ITA7	ITA6	ITA5	ITA4	ITA3	ITA2	ITA1	ITA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
31:20	RFU	Reserved. Write 0 to these bits. 0 is returned when these bits are read.
19:13	MSK(6:0)	Address mask. Specify the bits of bits 27 to 21 of the PCI address to be masked during address conversion. The window size is from 2 MB to 256 MB. 1: Not masked 0: Masked
12	WIN1EN	PCI access enable. 1: Enabled 0: Disabled
11	RFU	Reserved. Write 0 to this bit. 0 is returned when this bit is read.
10:0	ITA(10:0)	Internal bus address setting. If bits 31 to 21 of the PCI address are the same as the value set to PMBA(31:21) of the PCIMBA1REG register, they are converted into the value set in this field. The address having the higher bits converted is used as the internal bus address.

This register sets the address value on the internal bus side of the window that converts PCI memory space addresses into internal memory space addresses, and the window size when the PCIU operates on the PCI bus as a target.

17.9.4 PCITAW2REG (0x0F00 0C0C)

Bit	31	30	29	28	27	26	25	24
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Name	RFU	RFU	RFU	RFU	MSK6	MSK5	MSK4	MSK3
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Name	MSK2	MSK1	MSK0	WIN2EN	RFU	ITA10	ITA9	ITA8
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	ITA7	ITA6	ITA5	ITA4	ITA3	ITA2	ITA1	ITA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
31:20	RFU	Reserved. Write 0 to these bits. 0 is returned when these bits are read.
19:13	MSK(6:0)	Address mask. Specify the bits of bits 27 to 21 of the PCI address to be masked during address conversion. The window size is from 2 MB to 256 MB. 1: Not masked 0: Masked
12	WIN2EN	PCI access enable. 1: Enabled 0: Disabled
11	RFU	Reserved. Write 0 to this bit. 0 is returned when this bit is read.
10:0	ITA(10:0)	Internal bus address setting. If bits 31 to 21 of the PCI address are the same as the value set to PMBA(31:21) of the PCIMBA2REG register, they are converted into the value set in this field. The address having the higher bits converted is used as the internal bus address.

This register sets the address value on the internal bus side of the window that converts PCI memory space addresses into internal memory space addresses, and the window size when the PCIU operates on the PCI bus as a target.

17.9.5 PCIMIOAWREG (0x0F00 0C10)

Bit	31	30	29	28	27	26	25	24
Name	IBA31	IBA30	IBA29	IBA28	IBA27	IBA26	IBA25	IBA24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	1	0	0	0	0
After reset	0	0	0	1	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Name	RFU	RFU	RFU	RFU	MSK6	MSK5	MSK4	MSK3
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	1	1	1	1
After reset	0	0	0	0	1	1	1	1

Bit	15	14	13	12	11	10	9	8
Name	MSK2	MSK1	MSK0	WIN1EN	RFU	RFU	RFU	RFU
R/W	R/W	R/W	R/W	R/W	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	PCIIA31	PCIIA30	PCIIA29	PCIIA28	PCIIA27	PCIIA26	PCIIA25	PCIIA24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
31:24	IBA(31:24)	Internal bus base address. If bits 31 to 24 of the internal address match the value of the IBA bits, an access to the PCI bus is recognized. Make sure that the PCI space set to these bits does not overlap a local resource address, such as that of memory or boot ROM, or a PCI slave window.
23:20	RFU	Reserved. Write 0 to these bits. 0 is returned when these bits are read.
19:13	MSK(6:0)	Address mask. Specify the bits of bits 30 to 24 of the internal address to be masked during address conversion. The window size is 16 MB to 2 GB. 1: Not masked 0: Masked
12	WIN1EN	PCI access enable. 1: Enabled 0: Disabled
11:8	RFU	Reserved. Write 0 to these bits. 0 is returned when these bits are read.
7:0	PCIIA(31:24)	PCI I/O address setting. If the internal address is the same as the address set in the IBA field, it is converted into the address value set in these bits. The address with the higher bits converted is used as the I/O space address of the PCI.

During master transaction (CPU → PCI), the contents of this register are used as a table that converts the internal address output by the CPU into a memory space address for the PCI.

17.9.6 PCICONFDREG (0x0F00 0C14)

Bit	31	30	29	28	27	26	25	24
Name	CONFD31	CONFD30	CONFD29	CONFD28	CONFD27	CONFD26	CONFD25	CONFD24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Name	CONFD23	CONFD22	CONFD21	CONFD20	CONFD19	CONFD18	CONFD17	CONFD16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Name	CONFD15	CONFD14	CONFD13	CONFD12	CONFD11	CONFD10	CONFD9	CONFD8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	CONFD7	CONFD6	CONFD5	CONFD4	CONFD3	CONFD2	CONFD1	CONFD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
31:0	CONFD(31:0)	Configuration data

This register is used to set data for accessing the configuration register.

17.9.7 PCICONFAREG (0x0F00 0C18)

Bit	31	30	29	28	27	26	25	24
Name	CONFA31	CONFA30	CONFA29	CONFA28	CONFA27	CONFA26	CONFA25	CONFA24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Name	CONFA23	CONFA22	CONFA21	CONFA20	CONFA19	CONFA18	CONFA17	CONFA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Name	CONFA15	CONFA14	CONFA13	CONFA12	CONFA11	CONFA10	CONFA9	CONFA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	CONFA7	CONFA6	CONFA5	CONFA4	CONFA3	CONFA2	CONFA1	CONFA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
31:0	CONFA(31:0)	Configuration address

The value set in this register becomes the output of the AD(31:0) pins in the address phase of the configuration cycle.

17.9.8 PCIMAILREG (0x0F00 0C1C)

Bit	31	30	29	28	27	26	25	24
Name	MBOX31	MBOX30	MBOX29	MBOX28	MBOX27	MBOX26	MBOX25	MBOX24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Name	MBOX23	MBOX22	MBOX21	MBOX20	MBOX19	MBOX18	MBOX17	MBOX16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Name	MBOX15	MBOX14	MBOX13	MBOX12	MBOX11	MBOX10	MBOX9	MBOX8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	MBOX7	MBOX6	MBOX5	MBOX4	MBOX3	MBOX2	MBOX1	MBOX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
31:0	MBOX(31:0)	Mailbox data

This register is used to transfer messages between the CPU and PCI bus master. It can be read or written from both the internal bus side and PCI bus side. This register is mapped to the internal I/O space and PCI memory space and is used when it is accessed from the internal bus side or PCI bus side. When this register is accessed from the PCI bus side, the MB bit of the interrupt status register is set, and is used as one of the interrupt sources.

The address of this register is set by the MAILBAREG in the configuration header register. Only 21 bits, bits 31 to 11, of this register are decoded. Bit 10 and below are not decoded.

17.9.9 BUSERRADREG (0x0F00 0C24)

Bit	31	30	29	28	27	26	25	24
Name	EA31	EA30	EA29	EA28	EA27	EA26	EA25	EA24
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Name	EA23	EA22	EA21	EA20	EA19	EA18	EA17	EA16
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Name	EA15	EA14	EA13	EA12	EA11	EA10	EA9	EA8
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	EA7	EA6	EA5	EA4	EA3	EA2	EA1	EA0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
31:2	EA(31:2)	Error Address. Indicates the higher 30 bits of the address at which an error occurred. Indicates an internal address in the case of a PCIU master abort; otherwise, indicates a PCI address.
1:0	EA(1:0)	The value read is undefined.

This register indicates the address at which the bus error that has caused an interrupt occurred. All the bits of this register are cleared when they are read. Once a bus error has occurred and a value has been set in this register, it is retained until the register is read or the value is updated because another bus error has occurred.

17.9.10 INTCNTSTAREG (0x0F00 0C28)

(1/3)

Bit	31	30	29	28	27	26	25	24
Name	MABTCLR	TRDYCLR	PARCLR	MBCLR	SERRCLR	RTYCLR	MABCLR	TABCLR
R/W	W	W	W	W	W	W	W	W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Name	MABTMSK	TRDYMSK	PARMSK	MBMSK	SERRMSK	RTYMSK	MABMSK	TABMSK
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	1	1	1	1	1	1	1	1
After reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
Name	IBAMABT	TRDYRCH	PAR	MB	PCISERR	RTYRCH	MABORT	TABORT
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
31	MABTCLR	Clears IBAMABT interrupt. 1: Cleared 0: Not
30	TRDYCLR	Clears PCITRDY limit interrupt. 1: Cleared 0: Not
29	PARCLR	Clears PCI parity error interrupt. 1: Cleared 0: Not
28	MBCLR	Clears PCI Mailbox access interrupt. 1: Cleared 0: Not
27	SERRCLR	Clears PCI SERR INT interrupt. 1: Cleared 0: Not

(2/3)

Bit	Name	Function
26	RTYCLR	Clears PCI Retry Limit interrupt. 1: Cleared 0: Not
25	MABCLR	Clears PCI Master Abort interrupt. 1: Cleared 0: Not
23:16	RFU	Reserved. Write 0 to these bits. 0 is returned when these bits are read.
24	TABCLR	Clears PCI Target Abort interrupt. 1: Cleared 0: Not
15	MABTMSK	Masks IBAMABT interrupt output. 1: Masked 0: Not masked
14	TRDYMSK	Masks PCI TRDY interrupt output. 1: Masked 0: Not masked
13	PARMSK	Enables PCI parity error interrupt. 1: Enabled 0: Disabled
12	MBMSK	Masks PCI Mailbox access interrupt output. 1: Masked 0: Not masked
11	SERRMSK	Masks PCI SERR INT interrupt output. 1: Masked 0: Not masked
10	RTYMSK	Masks PCI Retry Limit interrupt output. 1: Masked 0: Not masked
9	MABMSK	Masks PCI Master Abort interrupt output. 1: Masked 0: Not masked
8	TABMSK	Masks PCI Target Abort interrupt output. 1: Masked 0: Not masked
7	IBAMABT	Aborts master of internal bus. 1: Aborts master 0: Does not abort master
6	TRDYRCH	PCI TRDY Limit Reached. The change of the TRDY# signal for the duration of the number of clocks set by the PCITRDYVREG after IRDY# has been asserted when the PCIU operates as a PCI master. 1: Not asserted 0: Asserted
5	PAR	Detects PCI Parity Error. 1: Parity error detected 0: Parity error not detected

(3/3)

Bit	Name	Function
4	MB	Accesses PCI Mailbox. 1: Accessed 0: Not accessed
3	PCISERR	Change of SERR#. 1: Asserted 0: Not asserted
2	RTYRCH	Retry Limit Reached. Indicates whether the PCIU has repeated retry the number of times set by RETVALREG when the PCIU operates as a PCI master. 1: Retry repeated 0: Retry not repeated
1	MABORT	Master Abort 1: Master abort issued 0: Master abort not issued
0	TABORT	Target Abort 1: Target abort received 0: Target abort not received

This register indicates the interrupt status and performs interrupt mask and clear control.

17.9.11 PCIEXACCREG (0x0F00 0C2C)

Bit	31	30	29	28	27	26	25	24
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	UNLOCK	EAREQ
R/W	R	R	R	R	R	R	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
31:2	RFU	Reserved. Write 0 to these bits. 0 is returned when these bits are read.
1	UNLOCK	Lock from device on PCI. 1: Not locked 0: Locked
0	EAREQ	Exclusive Access Request 1: Requests exclusive access by asserting LOCK# if PCI bus is in a condition in which it can lock. 0: Unlocks target.

This register controls exclusive access of the PCIU on the PCI bus, and indicates the status during exclusive access.

This unit can control exclusive access when it operates as the master on the PCI bus. When this unit operates as the target, it is accessed by the CPU core even though exclusive control is performed for access from the PCI bus.

(1) If VR4131 acts as the PCI master

When the VR4131 operates as the PCI master, first set the EAREQ bit of PCIEXACCREG to 1 to request a target for exclusive access. If the accessed target is ready to be locked in the next cycle, the target is locked to the PCIU. If the EAREQ bit is cleared to 0, the cycle currently being executed maintains the lock status. The target is unlocked from the next cycle.

(2) If VR4131 acts as target

When the VR4131 operates as a target, set the UNLOCK bit of the PCIEXACCREG register to 1 to enable locking to the VR4131 from the other master devices. If the PCIU is locked by another master device, it does not response to an access from a device on PCI other than the one that has locked the PCIU.

17.9.12 PCIRECONTREG (0x0F00 0C30)

Bit	31	30	29	28	27	26	25	24
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RTRYCNT7	RTRYCNT6	RTRYCNT5	RTRYCNT4	RTRYCNT3	RTRYCNT2	RTRYCNT1	RTRYCNT0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
31:8	RFU	Reserved. Write 0 to these bits. 0 is returned when these bits are read.
7:0	RTRYCNT(7:0)	Retry count

This register stores the result of counting the number of times the PCIU has generated a retry during a PCI bus transaction.

17.9.13 PCIENREG (0x0F00 0C34)

Bit	31	30	29	28	27	26	25	24
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	CONFIG_ DONE	RFU	RFU
R/W	R	R	R	R	R	R/W	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
31:3	RFU	Reserved. Write 0 to these bits. 0 is returned when these bits are read.
2	CONFIG_DONE	PCI Configuration Done 1: PCI transaction executed normally 0: Retry issued
1:0	RFU	Reserved. Write 0 to these bits. 0 is returned when these bits are read.

This register completes the setting of the configuration register.

The software must set the CONFIG_DONE bit of this register to 1 when the registers on the PCIU have been set. If this bit is set to 1, a PCI transaction is executed normally. If it is 0, a retry is issued whenever a target cycle is received on the PCI bus.

17.9.14 PCICLKSELREG (0x0F00 0C38)

Bit	31	30	29	28	27	26	25	24
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	SEL_CLK1	SEL_CLK0
R/W	R	R	R	R	R	R	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
31:2	RFU	Reserved. Write 0 to these bits. 0 is returned when these bits are read.
1:0	SEL_CLK(1:0)	Selects the PCI bus operation clock. 11: RFU 10: Internal bus clock/1 division 01: Internal bus clock/4 division 00: Internal bus clock/2 division

This register is used to select an operation clock on the PCI bus.

17.9.15 PCITRDYVREG (0x0F00 0C3C)

Bit	31	30	29	28	27	26	25	24
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	TRDYV7	TRDYV6	TRDYV5	TRDYV4	TRDYV3	TRDYV2	TRDYV1	TRDYV0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	1	0	0	0	0
After reset	0	0	0	1	0	0	0	0

Bit	Name	Function
31:8	RFU	Reserved. Write 0 to these bits. 0 is returned when these bits are read.
7:0	TRDYV(7:0)	Wait time limit period from asserting IRDY# to asserting TRDY#.

This register sets the upper-limit value for the time (number of clocks) during which TRDY# is asserted by the target after IRDY# has been asserted, when the PCIU operates on the PCI bus as a master. If TRDY# is not asserted within the time set in this register, the PCIU sets the TRDYRCH bit of INTCNTSTAREG to 1 and asserts an interrupt signal. Upon a reset, the number of clocks is set to 16.

17.9.16 PCICLKRUNREG (0x0F00 0C60)

Bit	15	14	13	12	11	10	9	8
Name	STOPEN	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R/W	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	CLKRUN
R/W	R	R	R	R	R	R	R	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15	STOPEN	Indicates the result of arbitration requesting stoppage of the PCI clock by the CLKRUN signal. 1: PCI clock can be stopped. 0: PCI clock cannot be stopped
14:1	RFU	Reserved. Write 0 to these bits. 0 is returned when these bits are read.
0	CLKRUN	Starts arbitration requesting stoppage of the PCI clock by the CLKRUN signal. 1: PCI clock stoppage request arbitration started 0: PCI clock stoppage request arbitration completed

This register is used to perform arbitration to request stoppage of the PCI clock by using the CLKRUN signal. When the CLKRUN bit of this register is set to 1, the CPU drives the CLKRUN pin high for one cycle. After that, it pulls up the CLKRUN pin and checks the input level.

If a low level is not input to the CLKRUN pin for a duration of four cycles after the CLKRUN pin has been driven high, the CLKRUN bit is cleared to 0. At the same time, the STOPEN bit is set to 1.

If a low level is input to the CLKRUN pin for a duration of four cycles after the CLKRUN pin has been driven high, the CLKRUN bit is cleared to 0. At the same time the STOPEN bit is also cleared to 0.

If a PCI device that cannot stop the clock supply arbitrarily is connected, execute arbitration to request stoppage of the PCI clock by the CLKRUN pin, by using this register, before executing a HIBERNATE or SUSPEND instruction.

The following are examples of the coding.

```
HIBERNATE:
      mfc0    t0,C0_SR
      ori     t0,t0,0x1
      xori    t0,t0,0x1
      mtc0    t0,C0_SR             # Int. disable
      li      t0,0xab000000        # I/O register base address
      li      t1,0x0001
      sh      t1,0xC60(t0)          # stopen = 0, clkrun = 1
chkloop:
      lhu     t2,0x60(t0)          # load check bits
      beq     t1,t2,chkloop
      nop

      beq     t2,zero,NO_SLEEP
      nop

      mfc0    t0,C0_SR
      xori    t0,t0,0x1
      mtc0    t0,C0_SR             # Int. enable

      hibernate
```

17.10 Configuration Header Registers

The PCIU can access the PCI configuration space as a master. This PCIU has two one-word length registers: PCICONFAREG and PCICONFDREG. To access the PCI configuration space, first access PCICONFAREG of the internal registers and set the 32-bit PCI address of the configuration register to be accessed. Then, read or write PCICONFDREG of the internal registers, which becomes a PCI configuration cycle to the set PCI address.

The configuration registers of this PCIU are mapped to the internal I/O space and cannot be accessed from other PCI devices (except PCIMAILREG).

17.10.1 Configuration cycle

The PCIU can access the internal configuration registers of each device on the PCI bus via PCICONFAREG and PCICONFDREG. The configuration address output to the PCI bus at this time is the value written to PCICONFAREG. The format of the address must be correctly set by software, in accordance with the PCI specifications.

To execute a type-0 configuration cycle, use the address format shown in format **(a)** of **Figure 17-1**. Set one of bits 13 to 11 of the device specification field to 1. On the motherboard, these bits are connected to IDSEL3 to IDSEL1.

To execute a type-1 configuration cycle, use the address format shown in format **(b)** of **Figure 17-1**.

Figure 17-1. Configuration Format

(a) Format in Type-0 configuration cycle

31 14	13 11	10 8	7 2	1	0
Reserved	Device specified bit	Function number	Register number	0	Type

Remark Set type = 0

(b) Format in Type-1 configuration cycle

31 24	23 16	15 11	10 8	7 2	1	0
Reserved	Bus number	Device specified bit	Function number	Register number	0	Type

Remark Set type = 1

Bit 1 of the PCICONFAREG register is fixed to 0 and the other bits are set freely by software. The value set becomes the value AD(31:0) in the address phase during the configuration cycle.

Table 17-6. PCI Configuration Header Registers

Physical Address	R/W	Abbreviation	Function
0x0F00 0D00	R	VENDORIDREG	Vendor ID register
		DEVICEIDREG	Device ID register
0x0F00 0D04	R/W	COMMABDREG	Command register that sets PCI interface
		STATUSREG	Status register of PCI interface
0x0F00 0D08	R/W	REVIDREG	Revision register of device
		CLASSREG	Class register of device
0x0F00 0D0C	R/W	CACHELSREG	System cache line size register
		LATTIMEREG	Timer count value setting register
0x0F00 0D10	R/W	MAILBAREG	Base address register of PCIMAILREG
0x0F00 0D14	R/W	PCIMBA1REG	PCI memory space base address register 1
0x0F00 0D18	R/W	PCIMBA2REG	PCI memory space base address register 2
0x0F00 0D3C	R/W	INTLINEREG	PCI interrupt line register
		INTPINREG	PCI interrupt pin register
0x0F00 0D40	R/W	RETVALREG	Retry limit value setting register
		PCIAPCNTREG	PCI arbitration protocol select register

The details of the configuration registers are described below.

17.10.2 VENDORIDREG, DEVICEIDREG (0x0F00 0D00)

Bit	31	30	29	28	27	26	25	24
Name	DID15	DID14	DID13	DID12	DID11	DID10	DID9	DID8
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Name	DID7	DID6	DID5	DID4	DID3	DID2	DID1	DID0
R/W	R	R	R	R	R	R	R	R
RTCRST	1	1	0	1	1	1	1	1
After reset	1	1	0	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
Name	VID15	VID14	VID13	VID12	VID11	VID10	VID9	VID8
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	1	0	0	0	0
After reset	0	0	0	1	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	VID7	VID6	VID5	VID4	VID3	VID2	VID1	VID0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	1	1	0	0	1	1
After reset	0	0	1	1	0	0	1	1

Bit	Name	Function
31:16	DID(15:0)	Device ID. 0x00DF for the V$_R$4131.
15:0	VID(15:0)	Vendor ID. 0x1033 for NEC.

Among 32 bits starting at 0x0F00 0D00, bits 15:0 and 31:16 are used as the VENDORIDREG and DEVICEIDREG registers, respectively.

17.10.3 COMMANDREG, STATUSREG (0x0F00 0D04)

(1/2)

Bit	31	30	29	28	27	26	25	24
Name	DPE	SSE	RMA	RTA	STA	DEVSEL1	DEVSEL0	DPR
R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W
RTCRST	0	0	0	0	0	0	1	0
After reset	0	0	0	0	0	0	1	0

Bit	23	22	21	20	19	18	17	16
Name	FBBC	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	FBBE	SERR_EN
R/W	R	R	R	R	R	R	R	R/W
RTCRST	0	0	0	1	0	0	0	0
After reset	0	0	0	1	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	WAIT_CTL	PER	VGA	MWI	SPC	BMAS	MEMEN	IOEN
R/W	R	R/W	R	R	R	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
31	DPE	Parity error detection 1: Detected 0: Not detected
30	SSE	System error report 1: When SERR# is asserted 0: When SERR# is deasserted
29	RMA	Master abort reception 1: Master abort when the device operates as a master 0: No master abort
28	RTA	Target abort reception bit 1: Target abort is received 0: No target abort
27	STA	Target abort report 1: Target abort occurs 0: No target abort

(2/2)

Bit	Name	Function
26:25	DEVSEL(1:0)	DEVSEL issue timing. The bits of the VR4131 are fixed to 01 (intermediate speed).
24	DPR	Data parity error detection 1: PERR# is asserted or PERR# of other devices is detected. 0: No parity error
23	FBBC	High-speed back-to-back support 1: Supported 0: Not supported
22:10	RFU	Reserved. Write 0 to these bits. 0 is returned when these bits are read.
9	FBBE	High-speed back-to-back enable 1: Enabled 0: Disabled
8	SERR_EN	System error (SERR#) enable 1: Enabled 0: Disabled
7	WAIT_CTL	Wait cycle control 1: Stepping supported 0: Not supported
6	PER	Parity error enable 1: Enabled 0: Disabled
5	VGA	VGA palette snoop 1: Enabled 0: Disabled
4	MWI	Memory write and invalidate enable 1: Enabled 0: Disabled
3	SPC	Special cycle enable 1: Enabled 0: Disabled
2	BMAS	Bus master enable 1: Enabled 0: Disabled
1	MEMEN	Memory space enable 1: Enabled 0: Disabled
0	IOEN	I/O space enable 1: Enabled 0: Disabled

Among 32 bits starting at 0x0F00 0D04, bits 15:0 and 31:16 are used as the COMMANDREG and STATUSREG registers, respectively.

COMMANDREG is used to perform the setting of the PCI interface.

STATUSREG is used to indicate the characteristics of the PCIU as a PCI agent, and to record events that have occurred on the bus. Although this register is a read/write register, write operations are used only for bet resets.

17.10.4 REVIDREG, CLASSREG (0x0F00 0D08)

Bit	31	30	29	28	27	26	25	24
Name	BASECL7	BASECL6	BASECL5	BASECL4	BASECL3	BASECL2	BASECL1	BASECL0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	1	1	0
After reset	0	0	0	0	0	1	1	0

Bit	23	22	21	20	19	18	17	16
Name	SUBCL7	SUBCL6	SUBCL5	SUBCL4	SUBCL3	SUBCL2	SUBCL1	SUBCL0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Name	PROG7	PROG6	PROG5	PROG4	PROG3	PROG2	PROG1	PROG0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RID7	RID6	RID5	RID4	RID3	RID2	RID1	RID0
R/W	R	R	R	R	R	R	R	R
RTCRST	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**
After reset	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
31:24	BASECL(7:0)	Basic class. 0x06 is read for the VR4131.
23:16	SUBCL(7:0)	Subclass. 0x00 is read for the VR4131.
15:8	PROG(7:0)	Programming interface. 0x00 is read for the VR4131.
7:0	RID(7:0)	Revision ID

Note The values differ depending on the revision.

Among 32 bits starting at 0x0F00 0D08, bits 7:0 and 31:8 are used as the REVIDREG and CLASSREG registers, respectively.

This register indicates the type of revision and function of the PCIU.

The values of BASECL(7:0), SUBCL(7:0), and PROG(7:0) that indicate class code are defined by PCISIG (PCI Special Interest Group), and the values shown above are set in the register of the VR4131 to indicate a host bridge.

17.10.5 CACHELSREG, LATTIMEREG (0x0F00 0D0C)

Bit	31	30	29	28	27	26	25	24
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Name	MLTIM7	MLTIM6	MLTIM5	MLTIM4	MLTIM3	MLTIM2	MLTIM1	MLTIM0
R/W	R/W	R/W	R/W	R/W	R/W	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	CLSIZ7	CLSIZ6	CLSIZ5	CLSIZ4	CLSIZ3	CLSIZ2	CLSIZ1	CLSIZ0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	1	0	0
After reset	0	0	0	0	0	1	0	0

Bit	Name	Function
31:16	RFU	Reserved. Write 0 to these bits. 0 is returned when these bits are read.
15:8	MLTIM(7:0)	Master latency timer setting. The lower three bits are fixed to 0.
7:0	CLSIZ(7:0)	Cache line size. 0x04 for the VR4131.

Among 32 bits starting at 0x0F00 0D0C, bits 7:0 and 31:8 are the CACHELSREG and LATTIMEREG registers, respectively.

17.10.6 MAILBAREG (0x0F00 0D10)

Bit	31	30	29	28	27	26	25	24
Name	MBADD31	MBADD30	MBADD29	MBADD28	MBADD27	MBADD26	MBADD25	MBADD24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Name	MBADD23	MBADD22	MBADD21	MBADD20	MBADD19	MBADD18	MBADD17	MBADD16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Name	MBADD15	MBADD14	MBADD13	MBADD12	MBADD11	MBADD10	MBADD9	MBADD8
R/W	R/W	R/W	R/W	R/W	R/W	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	MBADD7	MBADD6	MBADD5	MBADD4	PREF	TYPE1	TYPE0	MSI
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
31:4	MBADD(31:4)	Base address for PCIMAILREG. MBADD(10:4) are fixed to 0. Requests 2 KB space.
3	PREF	Prefetch 1: Enabled 0: Disabled
2:1	TYPE(1:0)	Address range in which memory block can be located 00: Optionally located in 32-bit address space Other: RFU
0	MSI	Memory space indicator 1: RFU 0: For memory address space

This register is used to set a base address of PCIMAILREG in the PCI memory space.

17.10.7 PCIMBA1REG (0x0F00 0D14)

Bit	31	30	29	28	27	26	25	24
Name	PMBA31	PMBA30	PMBA29	PMBA28	PMBA27	PMBA26	PMBA25	PMBA24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Name	PMBA23	PMBA22	PMBA21	PMBA20	PMBA19	PMBA18	PMBA17	PMBA16
R/W	R/W	R/W	R/W	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Name	PMBA15	PMBA14	PMBA13	PMBA12	PMBA11	PMBA10	PMBA9	PMBA8
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	PMBA7	PMBA6	PMBA5	PMBA4	PREF	TYPE1	TYPE0	MSI
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	1	0	0	0
After reset	0	0	0	0	1	0	0	0

Bit	Name	Function
31:4	PMBA(31:4)	Base address for target address window 1. Bits PMBA(20:4) are fixed to 0. Bits PMBA(27:21) reflect bits MSK(6:0) of the PCITAW1REG register^Note.
3	PREF	Prefetch 1: Enabled 0: Disabled
2:1	TYPE(1:0)	Address range in which memory block can be located. 00: Optionally located in 32-bit address space Other: RFU
0	MSI	Memory space indicator 1: RFU 0: For memory address space

Note PMBA(27:21) are ANDed with the MSK(6:0) bits of PCITAW1REG during write. The initialization program of the system reads all the bits after 1 has been written to them, and recognizes the requested size of the device depending on the number of bits of the lower address that are 0, when an address is allocated to the base address register. Therefore, the MSK(6:0) bits of PCITAW1REG must be set before the PCIU recognizes this requested size.

This register is used to set the base address of target address window 1 on the PCI side to transfer a transaction to the internal bus side when the PCIU serves as a target on the PCI bus. The required address space size reflects the set value of the MSK(6:0) bits of the PCITAW1REG register.

17.10.8 PCIMBA2REG (0x0F00 0D18)

Bit	31	30	29	28	27	26	25	24
Name	PMBA31	PMBA30	PMBA29	PMBA28	PMBA27	PMBA26	PMBA25	PMBA24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Name	PMBA23	PMBA22	PMBA21	PMBA20	PMBA19	PMBA18	PMBA17	PMBA16
R/W	R/W	R/W	R/W	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Name	PMBA15	PMBA14	PMBA13	PMBA12	PMBA11	PMBA10	PMBA9	PMBA8
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	PMBA7	PMBA6	PMBA5	PMBA4	PREF	TYPE1	TYPE0	MSI
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	1	0	0	0
After reset	0	0	0	0	1	0	0	0

Bit	Name	Function
31:4	PMBA(31:4)	Base address for target address window 2. Bits PMBA(20:4) are fixed to 0. Bits PMBA(27:21) reflect the MSK(6:0) bits of the PCITAW2REG register[Note].
3	PREF	Prefetch 1: Enabled 0: Disabled
2:1	TYPE(1:0)	Address range in which a memory block can be located. 00: Optionally located in 32-bit address space Others: RFU
0	MSI	Memory space indicator 1: RFU 0: For memory address space

Note PMBA(27:21) are ANDed with the MSK(6:0) bits of the PCITAW2REG register during write. The initialization program of the system reads all the bits after 1 has been written to them, and recognizes the requested size of the device depending on the number of bits of the lower address that are 0, when an address is allocated to the base address register. Therefore, the MSK(6:0) bits of the PCITAW2REG register must be set before the PCIU recognizes this requested size.

This register is used to set the base address of target address window 2 on the PCI side to transfer a transaction to the internal bus side when the PCIU serves as a target on the PCI bus. The required address space size reflects the set value of the MSK(6:0) bits of the PCITAW2REG register.

17.10.9 INTLINEREG, INTPINREG (0x0F00 0D3C)

Bit	31	30	29	28	27	26	25	24
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Name	INTPIN7	INTPIN6	INTPIN5	INTPIN4	INTPIN3	INTPIN2	INTPIN1	INTPIN0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	1
After reset	0	0	0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
Name	INTLINE7	INTLINE6	INTLINE5	INTLINE4	INTLINE3	INTLINE2	INTLINE1	INTLINE0
R/W	R/W	R/W	R/W	R/W	R/W	R/WW	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
31:16	RFU	Reserved. Write 0 to these bits. 0 is returned when these bits are read.
15:8	INTPIN(7:0)	Indicates a PCI interrupt pin. These bits of the V$_R$4122 are fixed to 0x01 (INTA#). 1: Used 0: Not used
7:0	INTLINE(7:0)	Connects PCI interrupt line. 1: Connected 0: Not connected

Among 32 bits starting at 0x0F00 0D3C, bits 7:0 and 31:8 are used as the INTLINEREG and INTPINREG registers, respectively.

This register is a memo register that indicates the PCI interrupt line connection and pin indication.

INTLINE(7:0) constitute a register that can be used to note the system interrupt line to which the interrupt of the PCIU is connected

17.10.10 RETVALREG, PCIAPCNTREG (0x0F00 0D40)

Bit	31	30	29	28	27	26	25	24
Name	RFU	RFU	RFU	RFU	RFU	TKYGNT	PAPC1	PAPC0
R/W	R	R	R	R	R	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Name	RTYVAL7	RTYVAL6	RTYVAL5	RTYVAL4	RTYVAL3	RTYVAL2	RTYVAL1	RTYVAL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
31:27	RFU	Reserved. Write 0 to these bits. 0 is returned when these bits are read.
26	TKYGNT	Take Away GNT mode. 1: Enabled 0: Disabled
25:24	PAPC(1:0)	PCI arbiter priority control. 11: RFU 10: Alternate B 01: Alternate 0 00: Fair
23:16	RFU	Reserved. Write 0 to these bits. 0 is returned when these bits are read.
15:8	RTYVAL(7:0)	Retry limit value. If these bits are 0, no limit is imposed on the number of retries.
7:0	RFU	Reserved. Write 0 to these bits. 0 is returned when these bits are read.

Among 32 bits starting at 0x0F00 0D40, bits 15:0 and 31:16 are used as the RETVALREG and PCIAPCNTREG registers, respectively.

The PCIU has an internal arbiter for the PCI bus. This arbiter arbitrates a total of four requests; one internal request from the PCIU and three external requests (REQ0, REQ1, and REQ2). The following three types of arbitration protocols are available: fair algorithm, the PCIU internal request priority algorithm, and external agent

(PEQ0) priority algorithm. With each algorithm, preemption is enabled or disabled in a cycle (Take Away GNT). The PCIAPCNTREG register is used to specify the operation mode of the arbiter. The priority is changed every cycle.

The PCIU provides the following three arbitration modes:

Fair: Protocol that evenly allocates the priority assigned to the requests of all master devices 0 to 3.
Alternate B: Protocol that allocates a priority, giving weight to the internal requests of the PCIU.
Alternate 0: Protocol that allocates a priority, giving weight to the request (REQ0) from external agent 0

In addition, if a request with a higher priority is issued even while the device granted by the PCIU is asserting a request, granting is immediately deasserted, and the mode (take away grant) which gives a grant to the request with the higher priority is taken. The TKYGNT bit becomes an enable bit.

The other functions of the internal arbiter of PCIU are as follows.

- Bus parking
  The internal arbiter of the PCIU supports “bus parking,” a PCI protocol that prevents the bus from being driven in the high-impedance state because one of the masters gives GNT if the bus is idle and if all the PCI masters, including the PCIU, do not assert a request. The master that executed the transaction immediately before the parking status asserts GNT.

- Frame timer timeout
  The arbiter in the PCIU starts the timer count when GNT is given to the PCI master connected to the PCI bus to monitor the FRAME# signal line. If FRAME# is not asserted for a duration of 16 clocks, GNT is canceled and is given to another master.
  If GNT is canceled after a duration of 16 clocks, and if a request is not asserted by another master, GNT is given to the PCIU, and the PCIU performs bus parking.

17.11 Transaction Transfer

This section explains the specifications related to the internal bus of the PCI and transaction transfer.

17.11.1 Transaction transfer path overview

The type of transaction transfer between the PCI and internal bus and how to implement a transaction are described below.

Table 17-7. Transaction Transfer

Transaction Direction	R/W	Transfer Protocol
PCI → Internal bus	Read	Delayed transaction 1) Receives read command from PCI bus. 2) Returns retry response to PCI bus side. 3) Executes read cycle to internal bus side. 4) Responds to transaction on PCI bus side and transmits read data after receiving read data from internal bus
	Write	Posted transaction 1) Receives write command from PCI bus 2) Receives write data from PCI bus and retains it internally. 3) Ends cycle from PCI bus side 4) Executes write cycle on internal bus side
Internal bus → PCI	Read	Inserts wait cycle 1) Receives read command from internal bus 2) Deasserts slave ready signal of internal bus 3) Executes read cycle on PCI bus side. 4) Asserts slave ready of internal bus and transmits read data after receiving read data from PCI bus.
	Write	Posted transaction 1) Receives write command from internal bus. 2) Receives write data from internal bus and retains it internally. 3) Ends cycle on internal bus side. 4) Executes write cycle to PCI bus side.

17.11.2 Transaction transfer case

The style of each transaction transfer shown in Table 17-7 is described below.

(1) PCI → Internal bus transaction (read)

The read cycle from the PCI bus side is in the form of a delayed transaction. When the PCIU has received a read command for the internal bus on the PCI bus, it retains the address of the command, and completes that cycle with a retry. In the meantime, the read command and address are transferred to the interface on the internal bus side, and a specific read cycle is started on the internal bus side. Before the read cycle on the internal bus side is completed, the interface on the PCI side receives no cycle as a target, but returns a retry response to the cycle for the PCIU. When the read cycle is completed on the internal bus side and all the read data has been fetched, the interface on the PCI side returns the read data to the cycle that has already returned a retry response to complete the cycle normally. At this time, the cycle that returns the read data is identified by checking whether it matches the retained address.

(2) PCI → Internal bus transaction (write)

The write cycle from the PCI bus side is in the form of a posted transaction. When this PCIU has received a transaction for the internal bus on the PCI bus, it retains its address and data after reception, and completes the cycle normally. In the meantime, the address and write data are transferred to the interface on the internal bus side, and a specific write cycle is started on the internal bus side.

(3) Internal bus → PCI bus transaction (read)

In the read cycle from the internal bus side, a wait state is inserted until the PCIU fetches read data from the PCI side. As a result, the internal bus cycle is in the wait status. The PCIU latches the address and requests the PCI side for a read cycle, keeping the slave ready signal on the internal bus side deasserted, when it has received a read command to the PCI bus on the internal bus. After that, it asserts slave ready on completely fetching specific read data, and transfers the read data to the internal bus side.

(4) Internal bus → PCI transaction (write)

The write cycle from the internal bus side is in the form of a posted transaction. When the PCIU has received a write command to the PCI bus on the internal bus, it retains its address and write data after reception, and completes the cycle normally. In the meantime, the address and write data are transferred to the interface on the PCI side, and a specific write cycle is started on the PCI side.

17.12 Abnormal Termination

This section explains the types of abnormal termination related to the PCIU.

17.12.1 From internal bus to PCI bus

(1) Cause on PCI side

The following cases may be listed as abnormal termination caused by the PCI bus for a PCI transaction from the internal bus. The PCIU actions are included in this description.

(a) If no DEVSEL response is issued from the PCI target.

The PCIU deasserts the FRAME# signal, and then the IRDY# signal at the next clock, and terminates the cycle by aborting the master if the DEVSEL# signal is not asserted six clocks after the FRAME# signal has been asserted. At this time the RMA bit of the STATUREG register in the configuration register header is set.

In a memory cycle or I/O cycle, the RMA bit of the STATUSREG register and MABORT bit of the INTCNTSTARTREG register are set. At this time, the PCI address is written to the BUSERRADREG register. No response is issued to the internal bus side (write data is discarded).

In the configuration cycle, dummy read data is returned to the internal bus, assuming that data with all bits set to 1 has been read. If data is written, it is discarded.

(b) If PCI target is silent after DEVSEL response

- If preemption occurs

  The cycle is terminated by a timeout.

  Once the value set in the master latency timer has been counted, the FRAME# signal is deasserted, the IRDY# signal is deasserted at the next clock, and the cycle is terminated. After that, transaction to the PCI side is started from the address at which the transfer was aborted. The regulations related to repetition are in compliance with the processing performed if retry has been received from the PCI target.

- If preemption does not occur

  The PCIU waits until preemption occurs. If the TRDY# signal is not asserted after the PCIU has waited for specified time, however, an interrupt is generated.

  If the TRDY# signal is not asserted after the time set in the PCITRDYVREG register has elapsed after the PCIU asserted the IRDY# signal, the TRDYRCH bit in the INTCNTSTAREG register is set to 1, and an interrupt is generated.

(c) If retry is received from the PCI target

The same access is attempted, with the number of times set in the RETVALREG register assumed as the upper limit.

- If the set number of times has been reached, the access is discarded. The data is also discarded.
- Set the RTYRCH bit of the INTCNTSTAREG register to 1.
- Write this PCI address to the BUSERRADREG register.

(d) If disconnect is received from the PCI target

Access is attempted again from the continuation of transfer, with the number of times set in the RETVALREG register assumed as the upper limit.

- If the set number of times has been reached, the access is discarded. The data is also discarded.

- Set the RTYRCH bit of the INTCNTSTAREG register to 1.
- Write this PCI address to the BUSERRADREG register.

(e) If target abort is received from the PCI target

The cycle is terminated. Transfer is not retried.

- The FRAME# signal is deasserted and the cycle is terminated. Access is not resumed.
- Set RTA bit of the STATUSREG register to 1. Set the TABORT bit of the INTCNTSTAREG register.
- If the SERR_EN bit of the COMMANDREG register is set, assert the SERR# signal.

(f) If parity error is detected in address/command from the PCI master.

Target abort is executed.

- Set the DPE and SSE bits of the STATUSREG register to 1.
- Set the PCISERR bit of the INTCNTSTAREG register to 1.

(2) Causes on internal bus side

The following case is an abnormal termination caused by the internal bus for a PCI transaction from the internal bus. The PCIU actions are included in this description.

(a) If the internal master does not assert/deassert the master ready signal

The internal bus arbiter performs timeout. The transferred data is discarded.

- The internal bus arbiter performs timeout and resets the system.
- This PCIU does not apply any action such as timeout.

17.12.2 From PCI bus to internal bus

(1) Causes on PCI side

The following cases may be listed as abnormal termination caused by the PCI bus for the transaction from a PCI bus to the internal bus. The PCIU actions are included in this description.

(a) If timeout is received from the PCI master

The cycle is terminated normally.

(b) If the PCI master attempts to transfer eight words or more

The PCI master is disconnected when the eight words of data have been transferred, and the cycle is terminated.

CHAPTER 18 DSIU (DEBUG SERIAL INTERFACE UNIT)

18.1 General

The debug serial interface unit (DSIU) is a serial interface for debugging and supports transfer rates of up to 115 kbps. In addition to the DDIN and DDOUT input/output pins, the DSIU supports the DCTS# and DRTS# pins, which are used for flow control.

18.2 Register Set

The DSIU registers are listed below.

Table 18-1. DSIU Registers

Physical Address	LCR7	R/W	Register Symbol	Function
0x0F00 0820	0	R	DSIURB	Receive buffer register (read)
		W	DSIUTH	Transmission hold register (write)
	1	R/W	DSIUDLL	Division rate lower register
0x0F00 0821	0	R/W	DSIUIE	Interrupt enable register
	1	R/W	DSIUDLM	Division rate higher register
0x0F00 0822	–	R	DSIUIID	Interrupt indication register (read)
		W	DSIUFC	FIFO control register (write)
0x0F00 0823	–	R/W	DSIULC	Line control register
0x0F00 0824	–	R/W	DSIUMC	MODEM control register
0x0F00 0825	–	R/W	DSIULS	Line status register
0x0F00 0826	–	R/W	DSIUMS	Modem status register
0x0F00 0827	–	R/W	DSIUSC	Scratch register
0x0F00 0809	–	R/W	SIURESET	Reset register (shared with SIU)

Remark LCR7 is bit 7 of the DSIULC register.

These registers are described in detail below.

18.2.1 DSIURB (0x0F00 0820: LCR7 = 0, Read)

Bit	7	6	5	4	3	2	1	0
Name	RXD7	RXD6	RXD5	RXD4	RXD3	RXD2	RXD1	RXD0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7:0	RXD(7:0)	Serial receive data

This register stores receive data used in serial communications.

To access this register, set the LCR7 bit of the DSIULC register to 0.

18.2.2 DSIUTH (0x0F00 0820: LCR7 = 0, Write)

Bit	7	6	5	4	3	2	1	0
Name	TXD7	TXD6	TXD5	TXD4	TXD3	TXD2	TXD1	TXD0
R/W	W	W	W	W	W	W	W	W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7:0	TXD(7:0)	Serial transmit data

This register stores transmit data used in serial communications.

To access this register, set the LCR7 bit of the DSIULC register to 0.

18.2.3 DSIUDLL (0x0F00 0820: LCR7 = 1)

Bit	7	6	5	4	3	2	1	0
Name	DLL7	DLL6	DLL5	DLL4	DLL3	DLL2	DLL1	DLL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7:0	DLL(7:0)	Baud rate divisor (lower byte)

This register is used to set the divisor (division rate) for the baud rate generator.

The data in this register and the higher DSIUDLM register are together handled as 16-bit data.

To access this register, set the LCR7 bit of the DSIULC register to 1.

18.2.4 DSIUIE (0x0F00 0821: LCR7 = 0)

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	IE3	IE2	IE1	IE0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7:4	RFU	Reserved. Write 0. 0 is returned after read.
3	IE3	Modem status interrupt 1: Enable 0: Disable
2	IE2	Receive status interrupt 1: Enable 0: Disable
1	IE1	Transmit holding register empty interrupt 1: Enable 0: Disable
0	IE0	Receive data presence interrupt or timeout interrupt in FIFO mode 1: Enable 0: Disable

This register is used to specify interrupt enable/disable settings for the five types of interrupts used by the DSIU.

This register enables each interrupt by setting the corresponding bit to 1.

Overall use of interrupt functions can be halted by setting all bits in this register to 0.

When interrupts are disabled, “pending” is not displayed in the IIR0 bit of the DSIUIID register even when the interrupt condition has been met.

Other functions in the DSIU are not affected even when interrupts are disabled. The settings in the DSIULS and DSIUMS registers are also valid.

To access this register, set the LCR7 bit of the DSIULC register to 0.

18.2.5 DSIUDLM (0x0F00 0821: LCR7 = 1)

Bit	7	6	5	4	3	2	1	0
Name	DLM7	DLM6	DLM5	DLM4	DLM3	DLM2	DLM1	DLM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7:0	DLM(7:0)	Baud rate divisor (higher byte)

This register is used to set the divisor (division rate) for the baud rate generator.

The data in this register and the lower DSIUDLL register are together handled as 16-bit data.

To access this register, set the LCR7 bit of the DSIULC register to 1.

The relationship between the baud rates and the DSIUDLL and DSIUDLM registers is as follows.

Table 18-2. Correspondence Between Baud Rates and Divisors

Baud Rate	Divisor	1-Clock Width (μs)
50	23,040	20,000.00
75	15,360	13,333.33
110	10,473	9,090.91
134.5	8,565	7,434.94
150	7,680	6,666.67
300	3,840	3,333.33
600	1,920	1,666.67
1,200	960	833.33
1,800	640	555.56
2,000	576	500.00
2,400	480	416.67
3,600	320	277.78
4,800	240	208.33
7,200	160	138.89
9,600	120	104.17
19,200	60	52.08
38,400	30	26.04
57,600	20	17.36
115,200	10	8.68
128,000	9	7.81
144,000	8	6.94
192,000	6	5.21
230,400	5	4.34
288,000	4	3.47
384,000	3	2.60
576,000	2	1.74
1,152,000	1	0.868

18.2.6 DSIUIID (0x0F00 0822: Read)

Bit	7	6	5	4	3	2	1	0
Name	IIR7	IIR6	RFU	RFU	IIR3	IIR2	IIR1	IIR0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	1
After reset	0	0	0	0	0	0	0	1

Bit	Name	Function
7:6	IIR(7:6)	Becomes 11 when FCR0 = 1
5:4	RFU	Reserved. Write 0. 0 is returned after read.
3	IIR3	Pending character timeout interrupt request (in FIFO mode) 1: No pending interrupt 0: Pending interrupt
2:1	IIR(2:1)	Sets the priority level of pending interrupts. See **Table 18-3**.
0	IIR0	Pending interrupt request 1: No pending interrupt 0: Pending interrupt

This register indicates the priority level for interrupt requests and the existence of pending interrupt requests.

The interrupt request priority order from highest to lowest is: receive line status, receive data existence, character timeout, transmission hold register empty, and modem status.

The contents of the IIR3 bit is valid only in FIFO mode; it is always 0 in 16450 mode.

The IIR2 bit becomes 1 when the IIR3 bit is set to 1.

Table 18-3. Interrupt Set/Reset

DSIUIID Register			Interrupt Set/Reset Function			
Bit 3 **Note**	Bit 2	Bit 1	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Control
0	1	1	Highest (1st)	Receive line status	Overrun error, parity error, framing error, or break interrupt	Read line status register
0	1	0	2nd	Receive data presence	Receive data exists or has reached the trigger level.	Read the receive buffer register or lower trigger level via FIFO.
1	1	0	2nd	Character timeout	During the time period for the four most recent characters, not one character has been read from the receive FIFO nor has a character been input to the receive FIFO. During this period, at least one character has been held in the receive FIFO.	Read receive buffer register
0	0	1	3rd	Transmission hold register empty	Transmit register is empty	Read IIR (if it is the interrupt source) or write to transmission hold register
0	0	0	4th	Modem status	CTS#	Read modem status register

Note In FIFO mode.

18.2.7 DSIUFC (0x0F00 0822: Write)

Bit	7	6	5	4	3	2	1	0
Name	FCR7	FCR6	RFU	RFU	FCR3	FCR2	FCR1	FCR0
R/W	W	W	R	R	W	W	W	W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7:6	FCR(7:6)	Receive FIFO trigger level 11: 14 bytes 10: 8 bytes 01: 4 bytes 00: 0 bytes
5:4	RFU	Reserved. Write 0. 0 is returned after read.
3	FCR3	Switch between 16450 mode and FIFO mode 1: From 16450 mode to FIFO mode 0: From FIFO mode to 16450 mode
2	FCR2	Transmit FIFO clear/counter clear. Cleared to 0 when 1 is written. 1: FIFO clear/counter clear 0: Normal
1	FCR1	Receive FIFO clear/counter clear. Cleared to 0 when 1 is written. 1: FIFO clear/counter clear 0: Normal
0	FCR0	Receive/Transmit FIFO enable 1: Enable 0: Disable

This register is used to control the FIFO.

- **FIFO interrupt modes**

When the receive FIFO is usable and receive interrupts are enabled, receive interrupts are generated as described below.

<1> When the FIFO has reached the specified trigger level, a receive data existence interrupt request is generated to inform the CPU.
This interrupt is cleared when the FIFO falls below the trigger level.

<2> When the FIFO has reached the specified trigger level, the DSIUIID register indicates a receive data existence interrupt request. As with the interrupt above, the interrupt is cleared when the FIFO falls below the trigger level.

<3> Receive line status interrupts are assigned a higher priority level than receive data existence interrupts.

<4> When characters are transferred from the shift register to the receive FIFO, the LSR0 bit is set to 1.
The value of this bit returns to 0 when the FIFO becomes empty.

When the receive FIFO is usable and receive interrupts are enabled, receive FIFO timeout interrupt requests are generated as described below.

<1> The following are the conditions under which FIFO timeout interrupt requests are generated.

- At least one character is being stored in the FIFO.
- The time required for sending four characters has elapsed since the serial reception of the last character. This includes the time for two stop bits in cases where a stop bit has been specified.
- The time required for sending four characters has elapsed since the CPU last accessed the FIFO.

The time between receiving the last character and issuing a timeout interrupt request is a maximum of 160 ms when operating at 300 baud and receiving 12-bit data.

<2> The transfer time for a character is calculated based on the baud rate clock for reception (internal). The delay time is in proportion to the baud rate.

<3> Once a timeout interrupt request has been generated, the timeout interrupt is cleared and the timer is reset as soon as the CPU reads one character from the receive FIFO.

<4> If no timeout interrupt request has been generated, the timer is reset when a new character is received or when the CPU reads the receive FIFO.

When the transmit FIFO is usable and transmit interrupts are enabled, transmit interrupt requests are generated as described below.

<1> When the transmit FIFO becomes empty, a transmission hold register empty interrupt request is generated. This interrupt is cleared when a character is written to the transmission hold register (from one to 16 characters can be written to the transmit FIFO during servicing of this interrupt), or when the DSIUIID register is read.

<2> If there are not at least two bytes of character data in the transmit FIFO between the time last LSR5 in the DSIULS register was 1 (transmit FIFO empty) and the next time LSR5 = 1, an empty transmit FIFO status is reported to the IIR bit after a delay period calculated as “The time for one character − The time for the last stop bit(s)”.
When transmit interrupts are enabled, the first transmit interrupt that is generated after the FCR0 (FIFO enable bit) is overwritten is indicated immediately.

The priority level of the character timeout interrupt and receive FIFO trigger level interrupt is the same as that of the receive data existence interrupt.
The priority level of the transmit FIFO empty interrupt is the same as that of the transmission hold register empty interrupt.
Whether data to be transmitted exists or not in the transmit FIFO and the transmit shift register, can be ascertained by checking bit 6 of the DSIULS register.
Bit 5 in the DSIULS register only indicates the existence of data in the transmit FIFO. Therefore, data may be remaining in the transmit shift register.

- **FIFO polling mode**

When FCR0 = 1 (FIFO is enabled), if the value of any or all of bits 3 to 0 of the DSIUIE register become 0, the DSIU enters FIFO polling mode. Because the transmit block and receive block are controlled separately, the polling mode can be set for either or both blocks.
In this mode, the status of the transmit block and/or receive block can be checked by reading the DSIULS register via a user program.
In the FIFO polling mode, there is no acknowledgement for the trigger level reach or a timeout, but the receive FIFO and transmit FIFO can still store characters as they normally do.

18.2.8 DSIULC (0x0F00 0823)

Bit	7	6	5	4	3	2	1	0
Name	LCR7	LCR6	LCR5	LCR4	LCR3	LCR2	LCR1	LCR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7	LCR7	Register switching at divisor latch access 1: Divisor latch access 0: Receive buffer, transmission hold register, interrupt enable register
6	LCR6	Break control 1: Set break 0: Clear break
5	LCR5	Parity fixing 1: Parity fixed 0: Parity not fixed
4	LCR4	Parity setting 1: Even parity 0: Odd parity
3	LCR3	Parity enable 1: Create parity (during transmission) or check parity (during reception) 0: No parity (during transmission) or no checking (during reception)
2	LCR2	Stop bit setting 1: 1.5 bits (character length is 5 bits) 2 bits (character length is 6, 7, or 8 bits) 0: 1 bit
1:0	LCR(1:0)	Specifies the length of one character (number of bits) 11: 8 bits 10: 7 bits 01: 6 bits 00: 5 bits

This register is used to specify the format for asynchronous data communication and exchange and to set the divisor latch access registers.

The LCR6 bit is used to send the break status to the receive side's UART. When the LCR6 bit = 1, the serial output (TxD) is forcibly set to the spacing (0) state.

The setting of the LCR5 bit becomes valid according to the settings of the LCR4 and LCR3 bits.

18.2.9 DSIUMC (0x0F00 0824)

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	MCR4	MCR3	MCR2	MCR1	MCR0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7:5	RFU	Reserved. Write 0. 0 is returned after read.
4	MCR4	For diagnostic testing (local loopback) 1: Enable 0: Disable
3	MCR3	OUT2 signal (internal) setting 1: Low level 0: High level
2	MCR2	OUT1 signal (internal) setting 1: Low level 0: High level
1	MCR1	RTS# output setting 1: Low level 0: High level
0	MCR0	DTR# output (internal) setting 1: Low level 0: High level

This register is used for interface control with a modem or data set (or a peripheral device that emulates a modem).

The settings of the MCR3, MCR2 and MCR0 bits become valid only when MCR4 is set to 1 (enabling use of local loopback).

- **Local loopback**

  The local loopback can be used to test the transmit/receive data path in the DSIU.

  The following operation (local loopback) is executed inside the DSIU when the MCR4 bit = 1.

  The transmit block's serial output (TxD) enters the marking state (1) and the serial input (RxD) to the receive block is cut off. The transmit shift register's output is looped back to the receive shift register's input.

  The modem control input (CTS#) is cut off and the four modem control outputs (DTR#(internal), RTS#, OUT1 (internal), and OUT2 (internal)) are internally connected to the corresponding modem control inputs.

  The modem control output pins are forcibly set as inactive (high level). During this kind of loopback mode, transmitted data can be immediately and directly received.

  When in loopback mode, both transmit and receive interrupts can be used. The interrupt sources are external to the transmit and receive blocks.

  Although modem control interrupts can be used, the lower 4 bits of the modem control register can be used instead of the four modem control inputs as interrupt sources.

  As usual, each interrupt is controlled by an interrupt enable register.

18.2.10 DSIULS (0x0F00 0825)

Bit	7	6	5	4	3	2	1	0
Name	LSR7	LSR6	LSR5	LSR4	LSR3	LSR2	LSR1	LSR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	1	1	0	0	0	0	0
After reset	0	1	1	0	0	0	0	0

Bit	Name	Function
7	LSR7	Error detection (in FIFO mode) 1: Parity error, framing error, or break is detected. 0: Normal
6	LSR6	Transmit block empty 1: No data in transmission hold register or transmit shift register No data in transmit FIFO (during FIFO mode) 0: Data exists in transmission hold register or transmit shift register. Data exists in transmit FIFO (in FIFO mode).
5	LSR5	Transmission hold register empty 1: Character is transferred to transmit shift register (in 16450 mode). Transmit FIFO is empty (in FIFO mode). 0: Character is stored in transmission hold register (during 16450 mode). Transmit data exists in transmit FIFO (in FIFO mode).
4	LSR4	Break interrupt 1: Break interrupt detected 0: Normal
3	LSR3	Framing error 1: Framing error detected 0: Normal
2	LSR2	Parity error 1: Parity error detected 0: Normal
1	LSR1	Overrun error 1: Overrun error detected 0: Normal
0	LSR0	Receive data ready 1: Receive data exists in FIFO. 0: No receive data in FIFO

The CPU uses this register to obtain information related to data transfers.
When LSR7 and LSR(4:1) are 1, reading this register clears these bits to 0.

Caution The LSR0 bit of the DSIULS register (receive data ready) is set before serial data reception is completed. Therefore, the LSR0 bit may not be cleared if the serial receive data is read from the DSIURB register immediately after the LSR0 bit is set.
Before reading data from the DSIURB register, wait for the stop bit width time to elapse after the LSR0 bit of the DSIULS register bit is set.

The LSR7 bit is valid only in FIFO mode; it always indicates 0 in 16450 mode.

The value of LSR4 becomes 1 when the spacing state (logical 0) is held longer than the time required for transmission of one word of receive data input (start bit + data bits + parity bit + stop bit). In FIFO mode, if a break interrupt is detected for one character in the FIFO, the character is regarded as an error character and the CPU is notified of a break interrupt when that character reaches the highest position in the FIFO. When a break occurs, one “zero” character is sent to the FIFO. The RxD enters a marking state, and when the next valid start bit is received, the next character can be transmitted.

The value of the LSR3 bit becomes 1 when a stop bit of 0 (spacing level) is detected (framing error) following the final data bit or parity bit. In FIFO mode, if a framing error is detected for one character in the FIFO, the character is regarded as an error character and the CPU is notified of a framing error when that character reaches the highest position in the FIFO. When a framing error occurs, the DSIU prepares for further synchronization. The next start bit is assumed to be the cause of the framing error and further data is not accepted until the next start bit has been sampled twice.

The value of the LSR2 bit becomes 1 when the even or odd parity specified in the LCR4 bit for a received character is not satisfied (parity error). In FIFO mode, if a parity error is detected for one character in the FIFO, the character is regarded as an error character and the CPU is notified of a parity error when that character reaches the highest position in the FIFO.

The value of the LSR1 bit becomes 1 when the next character is transmitted to the receive buffer register before the CPU reads this register and the previous character disappears (overrun error). In FIFO mode, if the data exceeds the trigger level as it continues to be transferred to the FIFO, even after the FIFO becomes full an overrun error will not occur until all characters are stored in the shift register. The CPU is acknowledged as soon as an overrun error occurs. The characters in the shift register are overwritten and are not transferred to the FIFO.

18.2.11 DSIUMS (0x0F00 0826)

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	MSR4	RFU	RFU	RFU	MSR0
R/W	R	R	R	R	R	R	R	R/W
RTCRST	0	0	0	Undefined	0	0	0	0
After reset	0	0	0	Undefined	0	0	0	0

Bit	Name	Function
7:5	RFU	Reserved. Write 0. 0 is returned after a read.
4	MSR4	State of CTS# input 1: High level 0: Low level
3:1	RFU	Reserved. Write 0. 0 is returned after a read.
0	MSR0	CTS# signal change 1: Change in CTS# signal 0: No change

This register indicates the current status of the CTS# signal and whether the line status has changed. When the MSR0 bit is set to 1, reading this register clears this bit to 0.

18.2.12 DSIUSC (0x0F00 0827)

Bit	7	6	5	4	3	2	1	0
Name	SCR7	SCR6	SCR5	SCR4	SCR3	SCR2	SCR1	SCR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7:0	SCR(7:0)	General-purpose data

This register is a readable/writable 8-bit register, and can be used freely by users. It does not affect control of the DSIU.

18.2.13 SIURESET (0x0F00 0809)

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	DSIU RESET	SIU RESET
R/W	R	R	R	R	R	R	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7:2	RFU	Reserved. Write 0. 0 is returned after a read.
1	DSIURESET	Resets DSIU. 1: Reset 0: Release
0	SIURESET	Resets SIU. 1: Reset 0: Release

This register is used to reset DSIU and SIU forcibly.

CHAPTER 19 LED (LED CONTROL UNIT)

19.1 General

LEDs are switched on and off at a regular interval. That can be set by program.

19.2 Register Set

The LED registers are listed below.

Table 19-1. LED Registers

Address	R/W	Register Symbol	Function
0x0F00 0180	R/W	LEDHTSREG	LED on time setting register
0x0F00 0182	R/W	LEDLTSREG	LED off time setting register
0x0F00 0188	R/W	LEDCNTREG	LED control register
0x0F00 018A	R/W	LEDASTCREG	LED auto stop time set register
0x0F00 018C	R/W	LEDINTREG	LED interrupt register

These registers are described in detail below.

19.2.1 LEDHTSREG (0x0F00 0180)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	HTS4	HTS3	HTS2	HTS1	HTS0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	1	0	0	0	0
After reset	0	0	0	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15:5	RFU	Reserved. Write 0. 0 is returned after a read.
4:0	HTS(4:0)	Sets LED on time. 11111: 1.9375 seconds : 10000: 1 second : 01000: 0.5 second : 00100: 0.25 second : 00010: 0.125 second 00001: 0.0625 second 00000: Disabled

Note The previous value is retained.

This register is used to set the LED’s ON time (low-level width of LEDOUT#).

The ON time ranges from 0.0625 to 1.9375 seconds and can be set in 0.0625 second units. The initial value is 1 second.

This register must not be changed once the LEDENABLE bit of the LEDCNTREG register has been set to 1. The operation is not guaranteed if a change is made after that point.

19.2.2 LEDLTSREG (0x0F00 0182)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	LTS6	LTS5	LTS4	LTS3	LTS2	LTS1	LTS0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	1	0	0	0	0	0
After reset	0	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15:7	RFU	Reserved. Write 0. 0 is returned after a read.
6:0	LTS(6:0)	Sets LED off time. 1111111: 7.9375 seconds : 1000000: 4 seconds : 0100000: 2 seconds : 0010000: 1 second : 0001000: 0.5 second : 0000100: 0.25 second : 0000010: 0.125 second 0000001: 0.0625 second 0000000: Disabled

Note The previous value is retained.

This register is used to set the LED’s OFF time (high-level width of LEDOUT#).

The OFF time ranges from 0.0625 to 7.9375 seconds and can be set in 0.0625 second units. The initial value is 2 seconds.

This register must not be changed once the LEDENABLE bit of the LEDCNTREG register has been set . The operation is not guaranteed if a change is made after that point.

19.2.3 LEDCNTREG (0x0F00 0188)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	LED STOP	LED ENABLE
R/W	R	R	R	R	R	R	R/W	R/W
RTCRST	0	0	0	0	0	0	1	0
After reset	0	0	0	0	0	0	**Note**	**Note**

Bit	Name	Function
15:2	RFU	Reserved. Write 0. 0 is returned after a read.
1	LEDSTOP	LED blink auto stop setting 1: Performs auto stop. 0: Does not perform auto stop.
0	LEDENABLE	LED blink setting 1: Blink. 0: Does not blink.

Note The previous value is retained.

This register is used to make various LED settings.

Caution When setting up LED activation, make sure that a value other than 0 has been set to LEDHTSREG, LEDLTSREG, and LEDASTCREG in advance. The operation is not guaranteed if 0 is set to these registers.

19.2.4 LEDASTCREG (0x0F00 018A)

Bit	15	14	13	12	11	10	9	8
Name	ASTC15	ASTC14	ASTC13	ASTC12	ASTC11	ASTC10	ASTC9	ASTC8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	1	0	0
After reset	0	0	0	0	0	1	0	0

Bit	7	6	5	4	3	2	1	0
Name	ASTC7	ASTC6	ASTC5	ASTC4	ASTC3	ASTC2	ASTC1	ASTC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	1	0	1	1	0	0	0	0
After reset	1	0	1	1	0	0	0	0

Bit	Name	Function
15:0	ASTC(15:0)	LED auto stop time blink count

This register sets the number of ON/OFF times prior to the automatic stopping of LED activation. The set value is read during a read. The initial setting is 1,200 times in which each time includes one second of ON time and two seconds of OFF time (one hour in total).

The pair of operations in which the LED is switched ON once and OFF once is counted as 1 by this counter. The counter counts down from the set value and an LEDINT interrupt is generated when it reaches 0.

Caution Setting this register is prohibited to 0. The operation is not guaranteed if this register is set to 0.

19.2.5 LEDINTREG (0x0F00 018C)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

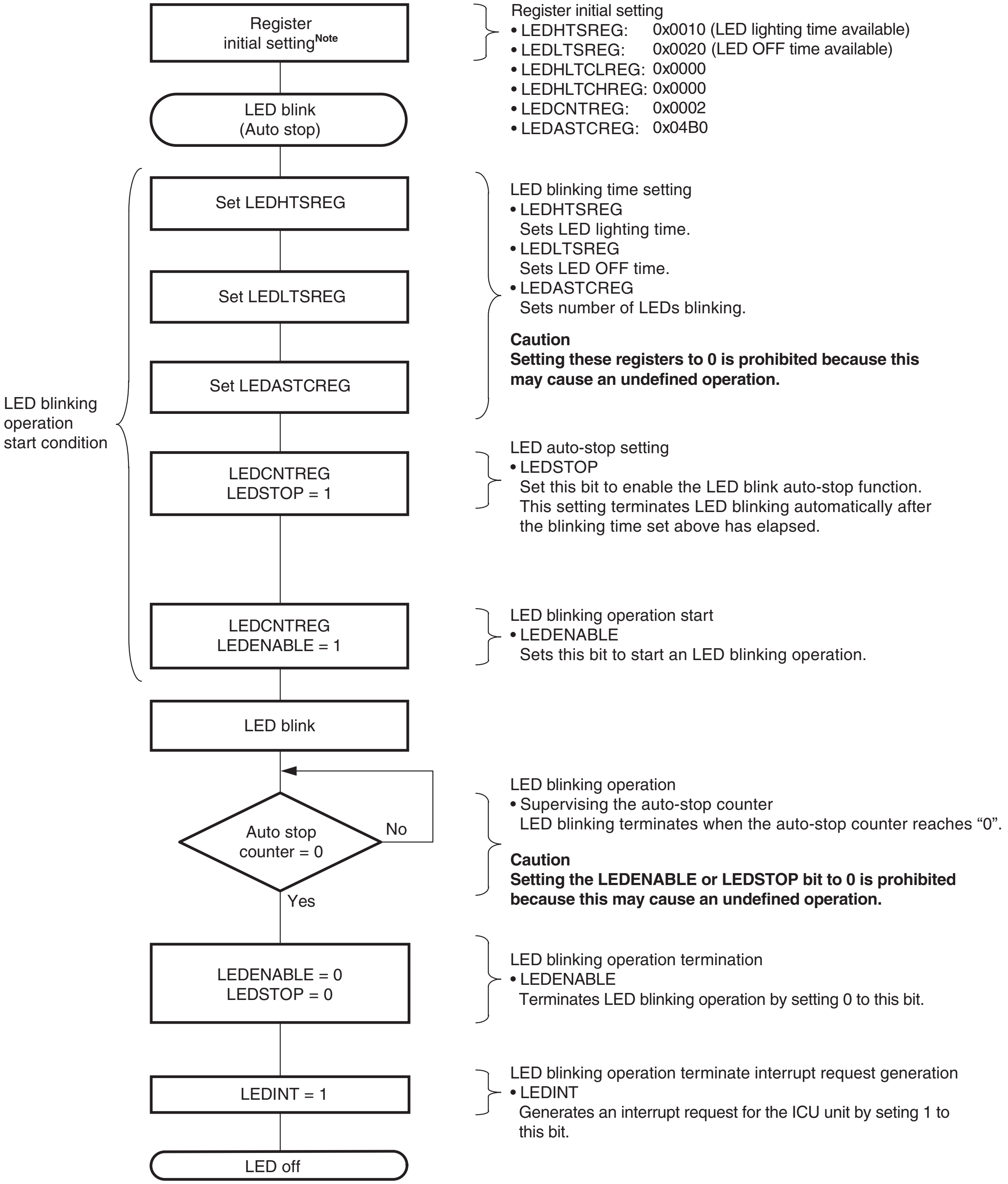
Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	LEDINT
R/W	R	R	R	R	R	R	R	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:1	RFU	Reserved. Write 0. 0 is returned after a read.
0	LEDINT	Auto stop interrupt request. Cleared to 0 when 1 is written. 1: Yes 0: No

This register indicates when an auto stop interrupt request has occurred.

An auto stop interrupt request occurs if the LEDSTOP and LEDENABLE bits of the LEDCNTREG register have already been set to 1 when the LEDASTCREG register is cleared to 0. When this interrupt request is generated, the LEDSTOP and LEDENABLE bits of LEDCNTREG are both cleared to 0.

19.3 Operation Flow



Register initial setting

• LEDHTSREG:	0x0010 (LED lighting time available)
• LEDLTSREG:	0x0020 (LED OFF time available)
• LEDHLTCLREG:	0x0000
• LEDHLTCHREG:	0x0000
• LEDCNTREG:	0x0002
• LEDASTCREG:	0x04B0

LED blinking time setting

- LEDHTSREG
  Sets LED lighting time.
- LEDLTSREG
  Sets LED OFF time.
- LEDASTCREG
  Sets number of LEDs blinking.

Caution
Setting these registers to 0 is prohibited because this may cause an undefined operation.

LED auto-stop setting

- LEDSTOP
  Set this bit to enable the LED blink auto-stop function. This setting terminates LED blinking automatically after the blinking time set above has elapsed.

LED blinking operation start

- LEDENABLE
  Sets this bit to start an LED blinking operation.

LED blinking operation

- Supervising the auto-stop counter
  LED blinking terminates when the auto-stop counter reaches “0”.

Caution
Setting the LEDENABLE or LEDSTOP bit to 0 is prohibited because this may cause an undefined operation.

LED blinking operation termination

- LEDENABLE
  Terminates LED blinking operation by setting 0 to this bit.

LED blinking operation terminate interrupt request generation

- LEDINT
  Generates an interrupt request for the ICU unit by seting 1 to this bit.

Note The initial setting for each register must be performed only when power is supplied to the device for the first time, regardless of whether the LED blinking function is used or not.

CHAPTER 20 SIU (SERIAL INTERFACE UNIT)

20.1 General

The SIU is a serial interface supporting communication that conforms to the RS-232C standard and is equipped with two one-channel interfaces, one for transmission and one for reception. In addition, this unit can support the transfer rate of the infrared communication physical layer standard IrDA 1.1, SIR.

This unit is functionally compatible with the NS16550.

20.2 Register Set

The SIU registers are listed below.

Table 20-1. SIU Registers

Physical Address	LCR7	R/W	Register Symbol	Function
0x0F00 0800	0	R	SIURB	Receive buffer register (read)
		W	SIUTH	Transmission hold register (write)
	1	R/W	SIUDLL	Division rate lower register
0x0F00 0801	0	R/W	SIUIE	Interrupt enable register
	1	R/W	SIUDLM	Division rate higher register
0x0F00 0802	–	R	SIUIID	Interrupt indication register (read)
		W	SIUFC	FIFO control register (write)
0x0F00 0803	–	R/W	SIULC	Line control register
0x0F00 0804	–	R/W	SIUMC	Modem control register
0x0F00 0805	–	R/W	SIULS	Line status register
0x0F00 0806	–	R/W	SIUMS	Modem status register
0x0F00 0807	–	R/W	SIUSC	Scratch register
0x0F00 0808	–	R/W	SIUIRSEL	Serial communication select register
0x0F00 0809	–	R/W	SIURESET	SIU reset register
0x0F00 080A	–	R/W	SIUCSEL	SIU echo-back control register

Remark LCR7 is bit 7 of the SIULC register.

These registers are described in detail below.

20.2.1 SIURB (0x0F00 0800: LCR7 = 0, Read)

Bit	7	6	5	4	3	2	1	0
Name	RXD7	RXD6	RXD5	RXD4	RXD3	RXD2	RXD1	RXD0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7:0	RXD(7:0)	Serial receive data

This register stores receive data used in serial communications.

To access this register, set the LCR7 bit of the SIULC register to 0.

20.2.2 SIUTH (0x0F00 0800: LCR7 = 0, Write)

Bit	7	6	5	4	3	2	1	0
Name	TXD7	TXD6	TXD5	TXD4	TXD3	TXD2	TXD1	TXD0
R/W	W	W	W	W	W	W	W	W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7:0	TXD(7:0)	Serial transmit data

This register stores transmit data used in serial communications.

To access this register, set the LCR7 bit of the SIULC register to 0.

20.2.3 SIUDLL (0x0F00 0800: LCR7 = 1)

Bit	7	6	5	4	3	2	1	0
Name	DLL7	DLL6	DLL5	DLL4	DLL3	DLL2	DLL1	DLL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7:0	DLL(7:0)	Baud rate divisor (lower byte)

This register is used to set the divisor (division rate) for the baud rate generator.

The data in this register and the higher SIUDLM register are together handled as 16-bit data.

To access this register, set the LCR7 bit of the SIULC register to 1.

20.2.4 SIUIE (0x0F00 0801: LCR7 = 0)

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	IE3	IE2	IE1	IE0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7:4	RFU	Reserved. Write 0. 0 is returned after read.
3	IE3	Modem status interrupt 1: Enable 0: Disable
2	IE2	Receive status interrupt 1: Enable 0: Disable
1	IE1	Transmission hold register empty interrupt 1: Enable 0: Disable
0	IE0	Receive data presence interrupt or timeout interrupt in FIFO mode 1: Enable 0: Disable

This register is used to specify interrupt enable/disable settings for the five types of interrupts used by the SIU. This register enables each interrupt by setting the corresponding bit to 1. Overall use of interrupt functions can be halted by setting all bits in this register to 0.

When interrupts are disabled “pending” is not displayed in the IIR0 bit of the SIUIID register even when the interrupt condition has been met.

Other functions in the SIU are not affected even when interrupts are disabled. The settings in the SIULS and SIUMS registers are also valid.

To access this register, set the LCR7 bit of the SIULC register to 0.

20.2.5 SIUDLM (0x0F00 0801: LCR7 = 1)

Bit	7	6	5	4	3	2	1	0
Name	DLM7	DLM6	DLM5	DLM4	DLM3	DLM2	DLM1	DLM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7:0	DLM(7:0)	Baud rate divisor (higher byte)

This register is used to set the divisor (division rate) for the baud rate generator.

The data in this register and the lower SIUDLL register are together handled as 16-bit data.

To access this register, set the LCR7 bit of the SIULC register to 1.

The relationship between the baud rates and the SIUDLL and SIUDLM registers is as follows.

Table 20-2. Correspondence Between Baud Rates and Divisors

Baud Rate	Divisor	1-Clock Cycle (μs)
50	23,040	20,000.00
75	15,360	13,333.33
110	10,473	9,090.91
134.5	8,565	7,434.94
150	7,680	6,666.67
300	3,840	3,333.33
600	1,920	1,666.67
1,200	960	833.33
1,800	640	555.56
2,000	576	500.00
2,400	480	416.67
3,600	320	277.78
4,800	240	208.33
7,200	160	138.89
9,600	120	104.17
19,200	60	52.08
38,400	30	26.04
57,600	20	17.36
115,200	10	8.68
128,000	9	7.81
144,000	8	6.94
192,000	6	5.21
230,400	5	4.34
288,000	4	3.47
384,000	3	2.60
576,000	2	1.74
1,152,000	1	0.868

20.2.6 SIUIID (0x0F00 0802: Read)

Bit	7	6	5	4	3	2	1	0
Name	IIR7	IIR6	RFU	RFU	IIR3	IIR2	IIR1	IIR0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	1
After reset	0	0	0	0	0	0	0	1

Bit	Name	Function
7:6	IIR(7:6)	Becomes 11 when FCR0 = 1
5:4	RFU	Reserved. Write 0. 0 is returned after read.
3	IIR3	Pending character timeout interrupt request (in FIFO mode) 1: No pending interrupt 0: Pending interrupt
2:1	IIR(2:1)	Sets the priority level of pending interrupts. See **Table 20-3**.
0	IIR0	Pending interrupt request 1: No pending interrupt 0: Pending interrupt

This register indicates the priority level for interrupt requests and the existence of pending interrupt requests. The interrupt request priority order from highest to lowest is: receive line status, receive data existence, character timeout, transmission hold register empty, and modem status.

The contents of the IIR3 bit is valid only in FIFO mode; it is always 0 in 16450 mode. The IIR2 bit becomes 1 when the IIR3 bit is set to 1.

Table 20-3. Interrupt Set/Reset

SIUIID Register			Interrupt Set/Reset Function			
Bit 3 **Note**	Bit 2	Bit 1	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Control
0	1	1	Highest (1st)	Receive line status	Overrun error, parity error, framing error, or break interrupt	Read line status register
0	1	0	2nd	Receive data presence	Receive data exists or has reached the trigger level.	Read the receive buffer register or lower trigger level via FIFO.
1	1	0	2nd	Character timeout	During the time period for the four most recent characters, not one character has been read from the receive FIFO nor has a character been input to the receive FIFO. During this period, at least one character has been held in the receive FIFO.	Read receive buffer register
0	0	1	3rd	Transmission hold register empty	Transmit register is empty	Read IIR (if it is the interrupt source) or write to transmission hold register
0	0	0	4th	Modem status	CTS#, DSR#, or DCD#	Read modem status register

Note In FIFO mode.

20.2.7 SIUFC (0x0F00 0802: Write)

Bit	7	6	5	4	3	2	1	0
Name	FCR7	FCR6	RFU	RFU	FCR3	FCR2	FCR1	FCR0
R/W	W	W	R	R	W	W	W	W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7:6	FCR(7:6)	Receive FIFO trigger level 11: 14 bytes 10: 8 bytes 01: 4 bytes 00: 0 bytes
5:4	RFU	Reserved. Write 0. 0 is returned after read.
3	FCR3	Switch between 16450 mode and FIFO mode 1: From 16450 mode to FIFO mode 0: From FIFO mode to 16450 mode
2	FCR2	Transmit FIFO clear/counter clear. Cleared to 0 when 1 is written. 1: FIFO clear/counter clear 0: Normal
1	FCR1	Receive FIFO clear/counter clear. Cleared to 0 when 1 is written. 1: FIFO clear/counter clear 0: Normal
0	FCR0	Receive/Transmit FIFO enable 1: Enable 0: Disable

This register is used to control the FIFO.

- **FIFO interrupt modes**

  When the receive FIFO is usable and receive interrupts are enabled, receive interrupts are generated as described below.

  <1> When the FIFO has reached to the specified trigger level, a receive data existence interrupt request is generated to inform the CPU.
  This interrupt is cleared when the FIFO falls below the trigger level.

  <2> When the FIFO has reached to the specified trigger level, the SIUIID register indicates a receive data existence interrupt request. As with <1> above, the interrupt is cleared when the FIFO falls below the trigger level.

  <3> Receive line status interrupts are assigned a higher priority level than receive data existence interrupts.

  <4> When characters are transferred from the shift register to the receive FIFO, the LSR0 bits is set to 1.
  The value of this bit returns to 0 when the FIFO becomes empty.

  When the receive FIFO is usable and receive interrupts are enabled, receive FIFO timeout interrupt requests are generated as described below.

  <1> The following are the conditions under which FIFO timeout interrupt requests are generated.

  - At least one character is being stored in the FIFO.
  - The time required for sending four characters has elapsed since serial reception of the last character. This includes the time for two stop bits in cases where a stop bit has been specified.
  - The time required for sending four characters has elapsed since the CPU last accessed the FIFO.

  The time between receiving the last character and issuing a timeout interrupt request is a maximum of 160 ms when operating at 300 baud and receiving 12-bit data.

  <2> The transfer time for a character is calculated based on the baud rate clock for reception (internal). The delay time is in proportion to the baud rate.

  <3> Once a timeout interrupt request has been generated, the timeout interrupt is cleared and the timer is reset as soon as the CPU reads one character from the receive FIFO.

  <4> If no timeout interrupt request has been generated, the timer is reset when a new character is received or when the CPU reads the receive FIFO.

When the transmit FIFO is usable and transmit interrupts are enabled, transmit interrupt requests are generated as described below.

<1> When the transmit FIFO becomes empty, a transmission hold register empty interrupt request is generated. This interrupt is cleared when a character is written to the transmission hold register (from one to 16 characters can be written to the transmit FIFO during servicing of this interrupt), or when the SIUIID register is read.

<2> If there are not at least two bytes of character data in the transmit FIFO between the last time LSR5 in the SIULS register was 1 (transmit FIFO empty) and the next time LSR5 = 1, an empty transmit FIFO status is reported to the IIR bit after a delay period calculated as “The time for one character − The time for the last stop bit(s)”.
When transmit interrupts are enabled, the first transmit interrupt that is generated after the FCR0 (FIFO enable bit) is overwritten is indicated immediately.

The priority level of the character timeout interrupt and receive FIFO trigger level interrupt is the same as that of the receive data existence interrupt.
The priority level of the transmit FIFO empty interrupt is the same as that of the transmission hold register empty interrupt.
Whether data to be transmitted exists or not in the transmit FIFO and the transmit shift register, can be ascertained by checking bit 6 of the SIULS register.
Bit 5 in the SIULS register only indicates the existence of data in the transmit FIFO. Therefore, data may be remaining in the transmit shift register.

- **FIFO polling mode**

When FCR0 = 1 (FIFO is enabled), if the value of any or all of bits 3 to 0 of the SIUIE register become 0, the SIU enters FIFO polling mode. Because the transmit block and receive block are controlled separately, polling mode can be set for either or both blocks.
In this mode, the status of the transmit block and/or receive block can be checked by reading the SIULS register via a user program.
In the FIFO polling mode, there is no acknowledgement for the trigger level reach or a timeout, but the receive FIFO and transmit FIFO can still store characters as they normally do.

20.2.8 SIULC (0x0F00 0803)

Bit	7	6	5	4	3	2	1	0
Name	LCR7	LCR6	LCR5	LCR4	LCR3	LCR2	LCR1	LCR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7	LCR7	Register switching at divisor latch access 1: Divisor latch access 0: Receive buffer, transmission hold register, interrupt enable register
6	LCR6	Break control 1: Set break 0: Clear break
5	LCR5	Parity fixing 1: Fixed parity 0: Parity not fixed
4	LCR4	Parity setting 1: Even parity 0: Odd parity
3	LCR3	Parity enable 1: Create parity (during transmission) or check parity (during reception) 0: No parity (during transmission) or no checking (during reception)
2	LCR2	Stop bit setting 1: 1.5 bits (character length is 5 bits) 2 bits (character length is 6, 7, or 8 bits) 0: 1 bit
1:0	LCR(1:0)	Specifies the length of one character (number of bits) 11: 8 bits 10: 7 bits 01: 6 bits 00: 5 bits

This register is used to specify the format for asynchronous data communication and exchange and to set the divisor latch access registers.

The LCR6 bit is used to send the break status to the receive side's UART. When the LCR6 bit = 1, the serial output (TxD) is forcibly set to the spacing (0) state.

The setting of the LCR5 bit becomes valid according to the settings of the LCR4 and LCR3 bits.

20.2.9 SIUMC (0x0F00 0804)

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	MCR4	MCR3	MCR2	MCR1	MCR0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7:5	RFU	Reserved. Write 0. 0 is returned after read.
4	MCR4	For diagnostic testing (local loopback) 1: Enable 0: Disable
3	MCR3	OUT2 signal (internal) setting 1: Low level 0: High level
2	MCR2	OUT1 signal (internal) setting 1: Low level 0: High level
1	MCR1	RTS# output setting 1: Low level 0: High level
0	MCR0	DTR# output setting 1: Low level 0: High level

This register is used for interface control with a modem or data set (or a peripheral device that emulates a modem).

The settings of the MCR3 and MCR2 bits become valid only when the MCR4 bit is set to 1 (enabling use of local loopback).

- **Local loopback**

  The local loopback can be used to test the transmit/receive data path in the SIU.

  The following operation (local loopback) is executed inside the SIU when the MCR4 bit = 1.

  The transmit block's serial output (TxD) enters the marking state (1) and the serial input (RxD) to the receive block is cut off. The transmit shift register's output is looped back to the receive shift register's input.

  The four modem control inputs (DSR#, CTS#, RI (internal), and DCD#) are cut off and the four modem control outputs (DTR#, RTS#, OUT1 (internal), and OUT2 (internal)) are internally connected to the corresponding modem control inputs.

  The modem control output pins are forcibly set as inactive (high level). During this kind of loopback mode, transmitted data can be immediately and directly received.

  When in loopback mode, both transmit and receive interrupts can be used. The interrupt sources are external to the transmit and receive blocks.

  Although modem control interrupts can be used, the lower 4 bits of the modem control register can be used instead of the four modem control inputs as interrupt sources. As usual, each interrupt is controlled by an interrupt enable register.

20.2.10 SIULS (0x0F00 0805)

Bit	7	6	5	4	3	2	1	0
Name	LSR7	LSR6	LSR5	LSR4	LSR3	LSR2	LSR1	LSR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	1	1	0	0	0	0	0
After reset	0	1	1	0	0	0	0	0

Bit	Name	Function
7	LSR7	Error detection (in FIFO mode) 1: Parity error, framing error, or break is detected. 0: Normal
6	LSR6	Transmit block empty 1: No data in transmission hold register or transmit shift register No data in transmit FIFO (in FIFO mode) 0: Data exists in transmission hold register or transmit shift register. Data exists in transmit FIFO (in FIFO mode).
5	LSR5	Transmission hold register empty 1: Character is transferred to transmit shift register (in 16450 mode). Transmit FIFO is empty (in FIFO mode). 0: Character is stored in transmission hold register (in 16450 mode). Transmit data exists in transmit FIFO (in FIFO mode).
4	LSR4	Break interrupt 1: Break interrupt detected 0: Normal
3	LSR3	Framing error 1: Framing error detected 0: Normal
2	LSR2	Parity error 1: Parity error detected 0: Normal
1	LSR1	Overrun error 1: Overrun error detected 0: Normal
0	LSR0	Receive data ready 1: Receive data exists in FIFO. 0: No receive data in FIFO

The CPU uses this register to obtain information related to data transfers.

When bits LSR7 and LSR(4:1) are 1, reading this register clears these bits to 0.

Caution The LSR0 bit of the SIULS register (receive data ready) is set before serial data reception is completed. Therefore, the LSR0 bit may not be cleared if the serial receive data is read from the SIURB register immediately after the LSR0 bit is set.

Before reading data from the SIURB register, wait for the stop bit width time to elapse after the LSR0 bit of the SIULS register bit is set.

The LSR7 bit is valid only in FIFO mode; it always indicates 0 in 16450 mode.

The value of LSR4 becomes 1 when the spacing state (logical 0) is held longer than the time required for transmission of one word of receive data input (start bit + data bits + parity bit + stop bit). In FIFO mode, if a break interrupt is detected for one character in the FIFO, the character is regarded as an error character and the CPU is notified of a break interrupt when that character reaches the highest position in the FIFO. When a break occurs, one “zero” character is sent to the FIFO. The RxD enters a marking state, and when the next valid start bit is received, the next character can be transmitted.

The value of the LSR3 bit becomes 1 when a stop bit of 0 (spacing level) is detected (framing error) following the final data bit or parity bit. In FIFO mode, if a framing error is detected for one character in the FIFO, the character is regarded as an error character and the CPU is notified of a framing error when that character reaches the highest position in the FIFO. When a framing error occurs, the SIU prepares for further synchronization. The next start bit is assumed to be the cause of the framing error and further data is not accepted until the next start bit has been sampled twice.

The value of the LSR2 bit becomes 1 when the even or odd parity specified in the LCR4 bit for a received character is not satisfied (parity error). In FIFO mode, if a parity error is detected for one character in the FIFO, the character is regarded as an error character and the CPU is notified of a parity error when that character reaches the highest position in the FIFO.

The value of the LSR1 bit becomes 1 when the next character is transmitted to the receive buffer register before the CPU reads this register and the previous character disappears (overrun error). In FIFO mode, if the data exceeds the trigger level as it continues to be transferred to the FIFO, even after the FIFO becomes full an overrun error will not occur until all characters are stored in the shift register. The CPU is acknowledged as soon as an overrun error occurs. The characters in the shift register are overwritten and are not transferred to the FIFO.

20.2.11 SIUMS (0x0F00 0806)

Bit	7	6	5	4	3	2	1	0
Name	MSR7	MSR6	MSR5	MSR4	MSR3	MSR2	MSR1	MSR0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	Undefined	Undefined	Undefined	Undefined	0	0	0	0
After reset	Undefined	Undefined	Undefined	Undefined	0	0	0	0

Bit	Name	Function
7	MSR7	State of DCD# signal 1: High level 0: Low level
6	MSR6	State of RI signal (internal) 1: High level 0: Low level
5	MSR5	State of DSR# input 1: High level 0: Low level
4	MSR4	State of CTS# input 1: High level 0: Low level
3	MSR3	DCD# signal change 1: Change in DCD# signal 0: No change
2	MSR2	RI signal (internal) change 1: Change in RI signal (internal) 0: No change
1	MSR1	DSR# signal change 1: Change in DSR# signal 0: No change
0	MSR0	CTS# signal change 1: Change in CTS# signal 0: No change

This register indicates the current status of various control signals that are input to the CPU from a modem or other peripheral device and whether each line status changes.

When the MSR(3:0) bits are set to 1, reading this register clears these bits to 0.

20.2.12 SIUSC (0x0F00 0807)

Bit	7	6	5	4	3	2	1	0
Name	SCR7	SCR6	SCR5	SCR4	SCR3	SCR2	SCR1	SCR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7:0	SCR(7:0)	General-purpose data

This register is a readable/writable 8-bit register, and can be used freely by users. It does not affect control of the SIU.

20.2.13 SIUIRSEL (0x0F00 0808)

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	TMICMODE	TMICTX	IRMSEL1	IRMSEL0	IRUSESEL	SIRSEL
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

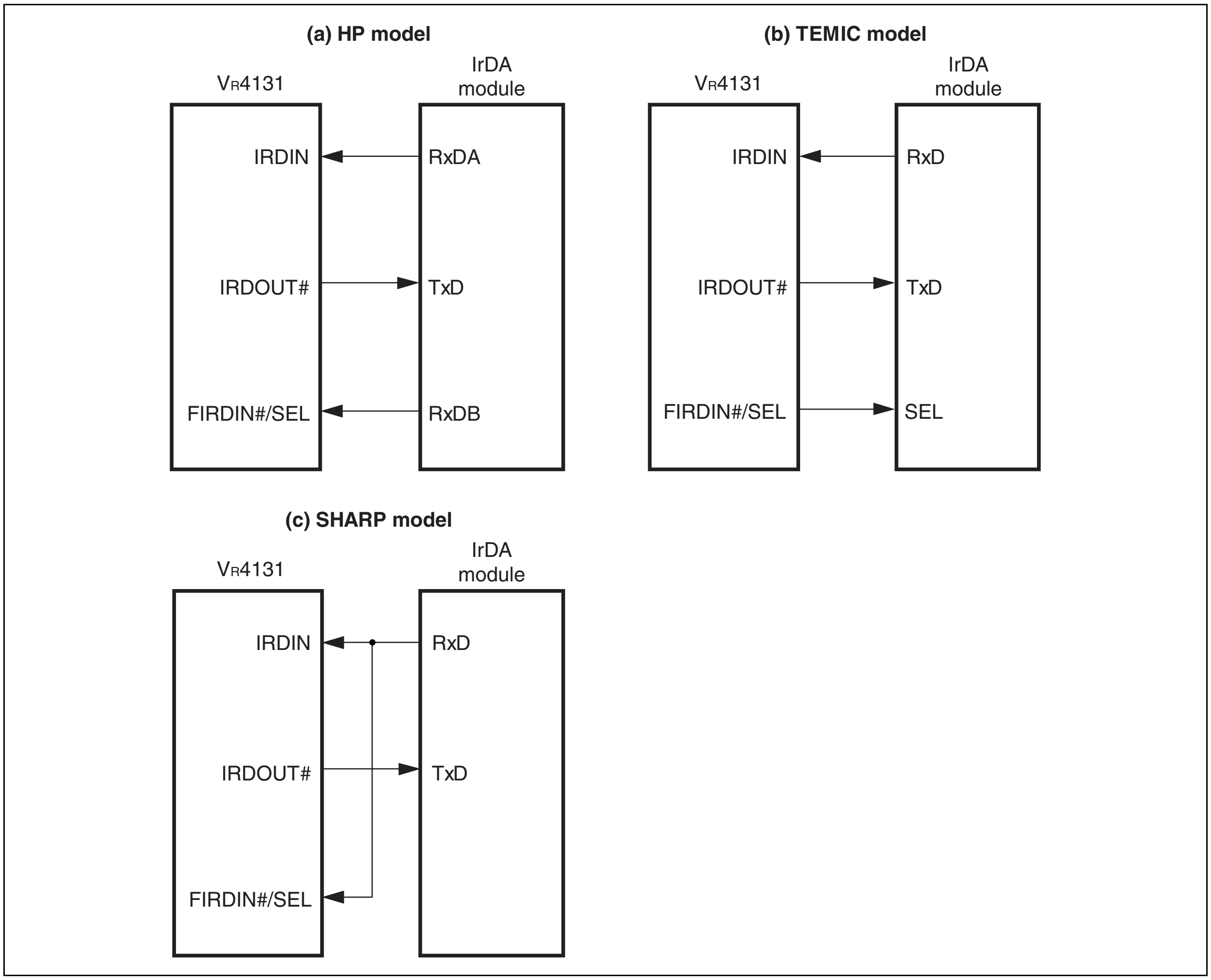
Bit	Name	Function
7:6	RFU	Reserved. Write 0. 0 is returned after read.
5	TMICMODE	If the emitter/receptor used is a TEMIC product, this bit is used. Mode setting of the emitter/receptor module by switching between low and high. Refer to TMICTX bit.
4	TMICTX	If the emitter/receptor used is a TEMIC product, this bit is used. Always set 0 after making the setting 1: Communication at 4 Mbps 0: Communication at 1.15 Mbps or less
3:2	IRMSEL(1:0)	Sets the type of emitter/receptor module to be used 11: RFU 10: HP model (HSDL-1100 is assumed) 01: TEMIC model (TFDS6000 is assumed) 00: SHARP model (RY5FD01D is assumed)
1	IRUSESEL	Selects SIU or FIR for use with IrDA emitter/receptor module 1: FIR uses IrDA module 0: SIU uses IrDA module
0	SIRSEL	Selects whether the SIU uses the IrDA module or the RS-232C interface during communications 1: Use IrDA module 0: Use RS-232C interface

This register is used to set the IrDA module settings, IrDA module access privileges, and the SIU’s communication format (IrDA or serial).

The settings of the TMICMODE and TMICTX bits are valid only when the IRMSEL(1:0) bits are set to 01 (TEMIC model).

Connection examples of the VR4131 and IrDA modules are shown below.

Figure 20-1. Example of Connection Between VR4131 and IrDA Module



20.2.14 SIURESET (0x0F00 0809)

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	DSIU RESET	SIU RESET
R/W	R	R	R	R	R	R	R	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7:1	RFU	Reserved. Write 0. 0 is returned after a read.
1	DSIURESET	Resets DSIU. 1: Reset 0: Release
0	SIURESET	Resets SIU. 1: Reset 0: Release

This register is used to reset the DSIU or the SIU forcibly.

20.2.15 SIUCSEL (0x0F00 080A)

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	CSEL
R/W	R	R	R	R	R	R	R	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7:1	RFU	Reserved. Write 0. 0 is returned after a read.
0	CSEL	This bit is used to specify masking for echo-back prevention. 1: Mask 0: Do not mask

This register is used to specify whether masking for echo-back prevention is performed when using SIR.

CHAPTER 21 CSI (CLOCKED SERIAL CONTROL UNIT)

21.1 Outline

The CSI, which is short for a clocked serial interface, is a synchronous serial interface.

21.2 Register Set

The CSI registers are listed below.

Table 21-1. CSI Registers

Address	R/W	Register Symbol	Function
0x0F00 01A0	R/W	CSI_MODEREG	CSI mode register
0x0F00 01A1	R/W	CSI_CLKSELREG	CSI clock select register
0x0F00 01A2	R	CSI_SIRBREG	CSI receive data buffer register
0x0F00 01A4	R/W	CSI_SOTBREG	CSI transmit data buffer register
0x0F00 01A6	R	CSI_SIRBEREG	CSI receive data buffer register (emulation read)
0x0F00 01A8	R/W	CSI_SOTBFREG	CSI first transmit data buffer register
0x0F00 01AA	R	CSI_SIOREG	CSI shift register
0x0F00 01B0	R/W	CSI_CNTREG	CSI control register
0x0F00 01B2	R/W	CSI_INTREG	CSI interrupt register
0x0F00 01B4	R/W	CSI_IFIFOVREG	CSI receive FIFO valid register
0x0F00 01B6	R/W	CSI_OFIFOVREG	CSI transmit FIFO valid register
0x0F00 01B8	R	CSI_IFIFOREG	CSI receive register
0x0F00 01BA	R/W	CSI_OFIFOREG	CSI transmit register
0x0F00 01BC	R/W	CSI_FIFOTRGREG	CSI FIFO trigger level register

These registers are described in detail below.

21.2.1 CSI_MODEREG (0x0F00 01A0)

Bit	7	6	5	4	3	2	1	0
Name	CSIE	TRMD	CCL	DIR	RFU	AUTO	RFU	CSOT
R/W	R/W	R/W	R/W	R/W	R	R/W	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7	CSIE	Enables CSI unit operation. 1: Enable 0: Disable
6	TRMD	Sets transmit/receive. 1: Transmit and receive 0: Receive
5	CCL	Sets data length. 1: 16 bits 0: 8 bits
4	DIR	Sets communication direction. 1: Transfer from LSB 0: Transfer from MSB
3	RFU	Reserved. Write 0. 0 is returned after a read.
2	AUTO	Sets sequence communication mode. 1: Sequence transfer mode 0: Single transfer mode
1	RFU	Write 0. 0 is returned after a read.
0	CSOT	Indicates communication state. 1: During transfer 0: Idle state

This is an 8-bit register used to control the serial transfer processing of the CSI unit. A reset clears the entire register contents to 0. The CSOT bit only, however, retains 1 for several μs after the MSKCSI bit of the CMUCLKMSK register of CMU is set to 1 following reset release.

The TRMD, CCL, DIR, and AUTO bits can only be overwritten when the CSOT bit is cleared.

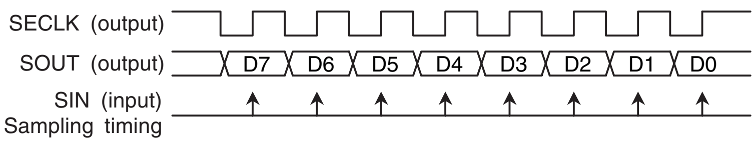
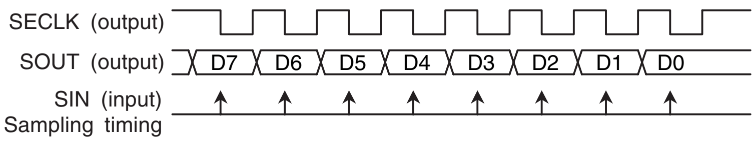
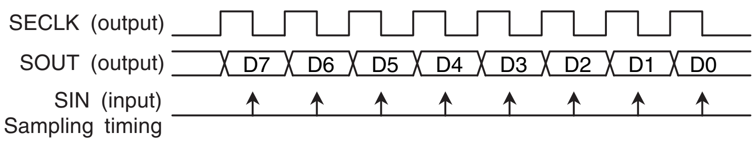
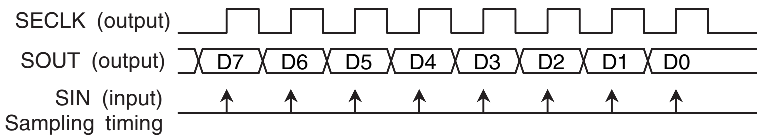
21.2.2 CSI_CLKSELREG (0x0F00 01A1)

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	CKP	DAP	CKS2	CKS1	CKS0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	1	1	0
After reset	0	0	0	0	0	1	1	0

Bit	Name	Function
7:5	RFU	Reserved. Write 0. 0 is returned after a read.
4	CKP	Selects clock phase. (See below.)
3	DAP	Selects data phase. (See below)
2:0	CKS(2:0)	Selects clock. (Only master mode is supported.) 111: RFU 110: 0.288 MHz 101: 0.576 MHz 100: 1.152 MHz 011: 2.304 MHz 010: 4.608 MHz 001: 9.216 MHz 000: RFU

This is an 8-bit register used to control the serial transfer processing of the CSI unit. A reset sets the register contents to 0x06.

The bits can be overwritten only when the CSIE bit is cleared.

CKP	DAP	Operation Mode
0	0	 SECLK (output) SOUT (output) D7 D6 D5 D4 D3 D2 D1 D0 SIN (input) Sampling timing
0	1	 SECLK (output) SOUT (output) D7 D6 D5 D4 D3 D2 D1 D0 SIN (input) Sampling timing
1	0	 SECLK (output) SOUT (output) D7 D6 D5 D4 D3 D2 D1 D0 SIN (input) Sampling timing
1	1	 SECLK (output) SOUT (output) D7 D6 D5 D4 D3 D2 D1 D0 SIN (input) Sampling timing

21.2.3 CSI_SIRBREG (0x0F00 01A2)

Bit	15	14	13	12	11	10	9	8
Name	SIRB15	SIRB14	SIRB13	SIRB12	SIRB11	SIRB10	SIRB9	SIRB8
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	SIRB7	SIRB6	SIRB5	SIRB4	SIRB3	SIRB2	SIRB1	SIRB0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:0	SIRB(15:0)	Receive data

This is a 16-bit register for storing receive data. Data can be read from this register when the TRMD bit of the CSI_MODEREG register is cleared.

Reception is started by reading from this register. It is therefore necessary to disregard the first data when first starting reception.

In sequential transfer mode (when the AUTO bit of the CSI_MODEREG register is 1), a read from this register becomes the trigger for the next reception. To end sequential transfer, abort reading from this register and wait for reception to end (when the CSOT bit of CSI_MODEREG is 0). Following the end of reception, the final data is stored in the CSI_SIOREG register, and the penultimate data is stored in both this register and CSI_SIRBEREG.

The entire register contents are cleared to 0 upon a reset, or when the CSIRST bit of the CSI_CNTREG register is set.

When the CCL bit of the CSI_MODEREG register is 0, only the lower 8 bits of the register are used, whereas when CCL is 1, all 16 bits are used.

21.2.4 CSI_SOTBREG (0x0F00 01A4)

Bit	15	14	13	12	11	10	9	8
Name	SOTB15	SOTB14	SOTB13	SOTB12	SOTB11	SOTB10	SOTB9	SOTB8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	SOTB7	SOTB6	SOTB5	SOTB4	SOTB3	SOTB2	SOTB1	SOTB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:0	SOTB(15:0)	Transmit data

This is a 16-bit register for storing transmit data. Data can be written to this register when the TRMD bit of the CSI_MODEREG register is set. The entire register contents are cleared to 0 upon a reset, or when the CSIE bit of CSI_MODEREG is cleared.

When not using transmit FIFO, transmission is started by writing to this register.

When the AUTO bit of the CSI_MODEREG register is set, it is necessary to write the data to be transmitted first to the CSI_SOTBFREG register, before writing to this register.

When the CCL bit of the CSI_MODEREG is 0, only the lower 8 bits of the register are used, whereas when CCL is 1, all 16 bits are used.

21.2.5 CSI_SIRBEREG (0x0F00 01A6)

Bit	15	14	13	12	11	10	9	8
Name	SIRBE15	SIRBE14	SIRBE13	SIRBE12	SIRBE11	SIRBE10	SIRBE9	SIRBE8
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	SIRBE7	SIRBE6	SIRBE5	SIRBE4	SIRBE3	SIRBE2	SIRBE1	SIRBE0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:0	SIRBE(15:0)	Receive data for read.

This is a 16-bit register for storing receive data. Even if the contents of this register are read, the next reception will not start. Data can be read from this register when the TRMD bit of the CSI_MODEREG register is cleared. The entire register contents are cleared to 0 upon a reset, or when the CSIRST bit of the CSI_CNTREG register is set.

When the CCL bit is 0, only the lower 8 bits of the register are used, whereas when CCL is 1, all 16 bits are used.

21.2.6 CSI_SOTBFREG (0x0F00 01A8)

Bit	15	14	13	12	11	10	9	8
Name	SOTBF15	SOTBF14	SOTBF13	SOTBF12	SOTBF11	SOTBF10	SOTBF9	SOTBF8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	SOTBF7	SOTBF6	SOTBF5	SOTBF4	SOTBF3	SOTBF2	SOTBF1	SOTBF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:0	SOTBF(15:0)	First transmit data

This is a 16-bit register for storing the first transmit data in sequential transfer mode (when the AUTO bit of the CSI_MODEREG register is 1). Set this register ahead of CSI_SOTBREG before starting transmission.

The entire register contents are cleared to 0 upon a reset, or when the CSIRST bit of the CSI_CNTREG register is set.

When the CCL bit is 0, only the lower 8 bits of the register are used, whereas when CCL is 1, all 16 bits are used.

This register can only be written to when the CSOT bit is cleared.

21.2.7 CSI_SIOREG (0x0F00 01AA)

Bit	15	14	13	12	11	10	9	8
Name	SIO15	SIO14	SIO13	SIO12	SIO11	SIO10	SIO9	SIO8
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	SIO7	SIO6	SIO5	SIO4	SIO3	SIO2	SIO1	SIO0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:0	SIO(15:0)	Shift data

This is a 16-bit register for storing serial data converted from parallel data. This register stores the final receive data after the CSOT bit is cleared at the end of reception in sequential transfer mode (when the AUTO bit of the CSI_MODEREG register is 1). The penultimate data is stored in both CSI_SIRBREG and CSI_SIRBEREG.

Even if the contents of this register are read, the next reception will not start.

The entire register contents are cleared to 0 upon a reset, when the CSIRST bit of the CSI_CNTREG register is set, or when the CSIE bit is cleared.

When the CCL bit of the CSI_MODEREG register is 0, only the lower 8 bits of the register are used, whereas when CCL is 1, all 16 bits are used.

This register can only be written to when the CSOT bit of the CSI_MODEREG register is cleared.

21.2.8 CSI_CNTREG (0x0F00 01B0)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	CSIRST	RFU	RFU	T_TRGEN	T_FIFOF	T_FIFOE	T_P1STP	T_DMAEN
R/W	R/W	R	R	R/W	R	R/W	R/W	R/W
RTCRST	1	0	0	0	0	0	0	0
After reset	1	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	SIO_V	SIRB_V	R_FULLS	R_TRGEN	R_FIFOF	R_FIFOE	R_P1STP	R_DMAEN
R/W	R	R	R/W	R/W	R	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15	CSIRST	Resets CSI unit 1: Reset state (default) 0: Normal state (0 is set when setting each register of CSI.)
14:13	RFU	Reserved. Write 0. 0 is returned after a read.
12	T_TRGEN	Enables transmit halt when the transmit FIFO reaches data set to T_TRG(2:0) of CSI_FIFOTRGREG. 1: Enable 0: Disable
11	T_FIFOF	Indicates that the transmit FIFO is FULL. 1: Full 0: Not full
10	T_FIFOE	Enables the use of transmit FIFO. 1: Enable 0: Disable
9	T_P1STP	Enables halt when the transmit DMA reaches the 1-page boundary. 1: Enable 0: Disable
8	T_DMAEN	Enables the use of transmit DMA. 1: Enable 0: Disable
7	SIO_V	Indicates that there is a valid data in SIO. 1: Valid data exists. 0: No valid data exists.
6	SIRB_V	Indicates that there is a valid data in SIRB. 1: Valid data exists. 0: No valid data exists.
5	R_FULLS	Enables halt when the receive FIFO is FULL. 1: Enable 0: Disable

(2/2)

Bit	Name	Function
4	R_TRGEN	Enables receive halt when the receive FIFO reaches data set to R_TRG(2:0) of CSI_FIFOTRGREG. 1: Enable 0: Disable
3	R_FIFOF	Indicates the receive FIFO is FULL. 1: Full 0: Not full
2	R_FIFOE	Enables the use of receive FIFO. 1: Enable 0: Disable
1	R_P1STP	Enables halt when the receive DMA reaches 1-page boundary. 1: Enable 0: Disable
0	R_DMAEN	Enables the use of receive DMA. 1: Enable 0: Disable

This is a 16-bit register used to control the CSI unit. A reset sets the CSIRST bit to 1 and clears all other bits to 0.

When the transmit/receive FIFO is enabled (R_FIFOE = 1, T_FIFOE = 1), the AUTO bit of the CSI_MODEREG register must be set. The operation will be undefined if the transmit/receive FIFO is enabled without the AUTO bit being set.

When transmit DMA and receive DMA are enabled (T_DMAEN = 1, R_DMAEN = 1), the transmit FIFO and transmit/receive FIFO must be enabled at the same time. The operation will be undefined if DMA is enabled without the transmit/receive FIFO being enabled.

21.2.9 CSI_INTREG (0x0F00 01B2)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	CSIEND	RFU	RFU	RFU	RFU	T_P2STP	T_P1STP	T_EMP
R/W	R/W	R	R	R	R	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	R_P2STP	R_P1STP	R_OVER
R/W	R	R	R	R	R	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15	CSIEND	Transmit and receive completion interrupt request This bit indicates that 1 data transmission or reception is completed. This bit is cleared to 0 when 1 is written. 1: Yes 0: No
14:11	RFU	Reserved. Write 0. 0 is returned after a read.
10	T_P2STP	Transmit DMA 2-page interrupt request This bit indicates that DMA exceeds 2 pages on transmission. This bit is cleared to 0 when 1 is written. 1: Yes 0: No
9	T_P1STP	Transmit DMA 1-page interrupt request This bit indicates that DMA exceeds 1 page on transmission. This bit is cleared to 0 when 1 is written. 1: Yes 0: No
8	T_EMP	Transmit FIFO empty interrupt request This bit is cleared to 0 when 1 is written. • T_TRGEN of CSI_CNTREG is 0: 1: Indicates that there is an open space because FIFO is empty on transmission. 0: FIFO is not empty. • T_TRGEN of CSI_CNTREG is 1: 1: Indicates that the transmit trigger level has been reached. 0: The transmit trigger level has not been reached.
7:3	RFU	Reserved. Write 0. 0 is returned after a read.
2	R_P2STP	Receive DMA 2-page interrupt This bit indicates that DMA exceeds 2 pages on reception. This bit is cleared to 0 when 1 is written.
1	R_P1STP	Receive DMA 1-page interrupt This bit indicates that DMA exceeds 1 page on reception. This bit is cleared to 0 when 1 is written.

(2/2)

Bit	Name	Function
0	R_OVER	CSICSB receive FIFO over interrupt request This bit is cleared to 0 when 1 is written. • R_TRGEN of the CSI_CNTREG register is 0: 1: Indicates that FIFO is Full on reception. 0: FIFO is not Full. • R_TRGEN of the CSI_CNTREG register is 1: 1: Indicates that the receive trigger level has been reached. 0: The receive trigger level has not been reached.

This register is used to report that an interrupt request was generated in the CSI unit, or to cancel the interrupt request.

All the bits are cleared to 0 after reset.

21.2.10 CSI_IFIFOVREG (0x0F00 01B4)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	R_F7HV	R_F7LV	R_F6HV	R_F6LV	R_F5HV	R_F5LV	R_F4HV	R_F4LV
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	R_F3HV	R_F3LV	R_F2HV	R_F2LV	R_F1HV	R_F1LV	R_F0HV	R_F0LV
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15	R_F7HV	Indicates that the higher receive FIFO7 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
14	R_F7LV	Indicates that the lower receive FIFO7 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
13	R_F6HV	Indicates that the higher receive FIFO6 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
12	R_F6LV	Indicates that the lower receive FIFO6 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
11	R_F5HV	Indicates that the higher receive FIFO5 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
10	R_F5LV	Indicates that the lower receive FIFO5 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
9	R_F4HV	Indicates that the higher receive FIFO4 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
8	R_F4LV	Indicates that the lower receive FIFO4 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
7	R_F3HV	Indicates that the higher receive FIFO3 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
6	R_F3LV	Indicates that the lower receive FIFO3 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid

(2/2)

Bit	Name	Function
5	R_F2HV	Indicates that the higher receive FIFO2 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
4	R_F2LV	Indicates that the lower receive FIFO2 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
3	R_F1HV	Indicates that the higher receive FIFO1 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
2	R_F1LV	Indicates that the lower receive FIFO1 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
1	R_F0HV	Indicates that the higher receive FIFO0 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
0	R_F0LV	Indicates that the lower receive FIFO0 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid

The CSIU has eight 16-bit FIFO buffers for storing receive data. This register is a valid bit of the FIFO buffer for reception. The contents of this register are valid only when the R_FIFOE bit of the CSI_CNTREG register is set. The entire register contents are cleared to 0 upon a reset, or when the CSIRST bit is set.

21.2.11 CSI_OFIFOVREG (0x0F00 01B6)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	T_F7HV	T_F7LV	T_F6HV	T_F6LV	T_F5HV	T_F5LV	T_F4HV	T_F4LV
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	T_F3HV	T_F3LV	T_F2HV	T_F2LV	T_F1HV	T_F1LV	T_F0HV	T_F0LV
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15	T_F7HV	Indicates that the higher transmit FIFO7 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
14	T_F7LV	Indicates that the lower transmit FIFO7 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
13	T_F6HV	Indicates that the higher transmit FIFO6 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
12	T_F6LV	Indicates that the lower transmit FIFO6 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
11	T_F5HV	Indicates that the higher transmit FIFO5 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
10	T_F5LV	Indicates that the lower transmit FIFO5 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
9	T_F4HV	Indicates that the higher transmit FIFO4 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
8	T_F4LV	Indicates that the lower transmit FIFO4 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
7	T_F3HV	Indicates that the higher transmit FIFO3 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
6	T_F3LV	Indicates that the lower transmit FIFO3 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid

(2/2)

Bit	Name	Function
5	T_F2HV	Indicates that the higher transmit FIFO2 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
4	T_F2LV	Indicates that the lower transmit FIFO2 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
3	T_F1HV	Indicates that the higher transmit FIFO1 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
2	T_F1LV	Indicates that the lower transmit FIFO1 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
1	T_F0HV	Indicates that the higher transmit FIFO0 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid
0	T_F0LV	Indicates that the lower transmit FIFO0 is valid. This bit is cleared when 0 is written. 1: Valid 0: Invalid

The CSIU has eight 16-bit FIFO buffers for storing transmit data. This register is a valid bit of the FIFO buffer for transmission. The contents of this register are valid only when the T_FIFOE bit of the CSI_CNTREG register is set.

The entire register contents are cleared to 0 upon a reset, or when the CSIRST bit of the CSI_CNTRER register is set.

21.2.12 CSI_IFIFOREG (0x0F00 01B8)

Bit	15	14	13	12	11	10	9	8
Name	IFIFO15	IFIFO14	IFIFO13	IFIFO12	IFIFO11	IFIFO10	IFIFO9	IFIFO8
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	IFIFO7	IFIFO6	IFIFO5	IFIFO4	IFIFO3	IFIFO2	IFIFO1	IFIFO0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:0	IFIFO(15:0)	Receive FIFO buffer data

This is a window register for reading data from the receive FIFO buffers. Each time this register is read, the read pointer is moved. The contents of this register are valid only when the R_FIFOE bit of the CSI_CNTREG register is set and the CSI_IFIFOVREG register is not 0.

When the CCL bit of the CSI_MODEREG register is 0, only the lower 8 bits of the register can be used. However, 2 items of data can also be read simultaneously using 16-bit access. When the CCL bit of the CSI_MODEREG register is 1, all 16 bits are used.

21.2.13 CSI_OFIFOREG (0x0F00 01BA)

Bit	15	14	13	12	11	10	9	8
Name	OFIFO15	OFIFO14	OFIFO13	OFIFO12	OFIFO11	OFIFO10	OFIFO9	OFIFO8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	OFIFO7	OFIFO6	OFIFO5	OFIFO4	OFIFO3	OFIFO2	OFIFO1	OFIFO0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:0	OFIFO(15:0)	Transmit FIFO register data

This is a window register for writing data to the transmit FIFO buffers. Each time this register is written to, the write pointer is moved. Writing data to this register is valid only when the T_FIFOE bit of CSI_CNTREG is set.

When the CCL bit of the CSI_MODEREG register is 0, only the lower 8 bits of the register can be used. However, 2 items of data can also be written simultaneously using 16-bit access. When CCL is 1, all 16 bits are used. However, use 16-bit access when CCL is 1.

21.2.14 CSI_FIFOTRGREG (0x0F00 01BC)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	T_TRG2	T_TRG1	T_TRG0
R/W	R	R	R	R	R	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	R_TRG2	R_TRG1	R_TRG0
R/W	R	R	R	R	R	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:11	RFU	Reserved. Write 0. 0 is returned after a read.
10:8	T_TRG(2:0)	Stops transfer of transmit FIFO 111: Halts after transmission of 8 × 16 bits. 110: Halts after transmission of 7 × 16 bits. 101: Halts after transmission of 6 × 16 bits. 100: Halts after transmission of 5 × 16 bits. 011: Halts after transmission of 4 × 16 bits. 010: Halts after transmission of 3 × 16 bits. 001: Halts after transmission of 2 × 16 bits. 000: Halts after transmission of 1 × 16 bits.
7:3	RFU	Reserved. Write 0. 0 is returned after a read.
2:0	R_TRG(2:0)	Stops transfer of transmit FIFO 111: Halts after reception of 8 × 16 bits. 110: Halts after reception of 7 × 16 bits. 101: Halts after reception of 6 × 16 bits. 100: Halts after reception of 5 × 16 bits. 011: Halts after reception of 4 × 16 bits. 010: Halts after reception of 3 × 16 bits. 001: Halts after reception of 2 × 16 bits. 000: Halts after reception of 1 × 16 bits.

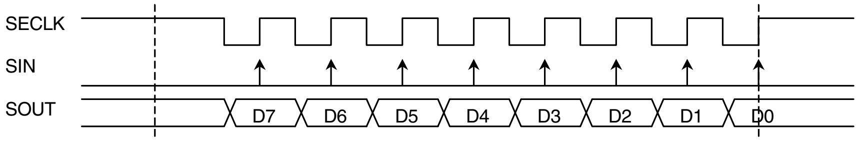
This is a register for stopping transfer after transmission/reception of the set data has finished. This register is valid only when the transmit/receive FIFO buffers are used.

For example, when the R_TRG(2:0) bits of this register are set to 001 and the CCL bit of the CSI_MODEREG register is 0, transfer can be stopped after 4 receptions of 8-bit data.

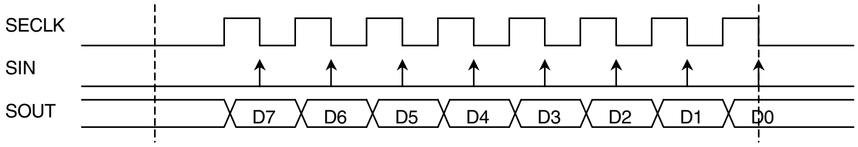
21.3 Clock Phase Selection Timing Chart

- 8-bit data length (CCL = 0)
- MSB first mode (DIR = 0)

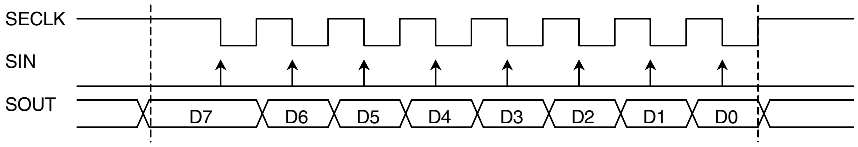
- **When CKP = 0, DAP = 0**



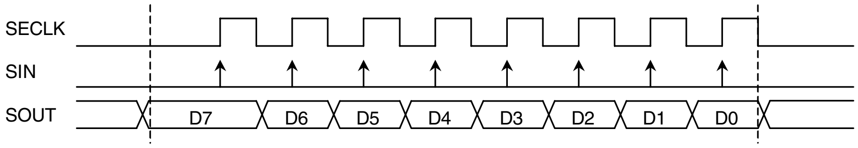
- **When CKP = 1, DAP = 0**



- **When CKP = 0, DAP = 1**



- **When CKP = 1, DAP = 1**



21.4 Details Concerning Single Transfer Mode

Single transfer mode is entered when the AUTO bit of the CSI_MODEREG register is cleared.

The method for starting transmission/reception differs depending on the state of the CSI_MODEREG register's TRMD bit.

- When TRMD = 0: Reception is started by reading data from the CSI_SIRBREG register. Disregard the first data received.
- When TRMD = 1: Transmission/reception is started by writing data to the CSI_SOTBREG register.

CSOT of CSI_MODEREG becomes 1 after the start of transfer, by which time the CSI interrupt source must have been cleared. At the end of transfer, the CSI interrupt becomes active, and CSOT changes to 0. The next data can then be transferred.

Cautions 1. Do not access CSI_MODEREG, CSI_CLKSELREG, CSI_CNTREG, CSI_SIRBREG, CSI_SOTBREG, CSI_SIRBEREG, or CSI_SOTBFREG when the CSOT bit is 1.

2. The transmit/receive FIFO cannot be used in single transfer mode.

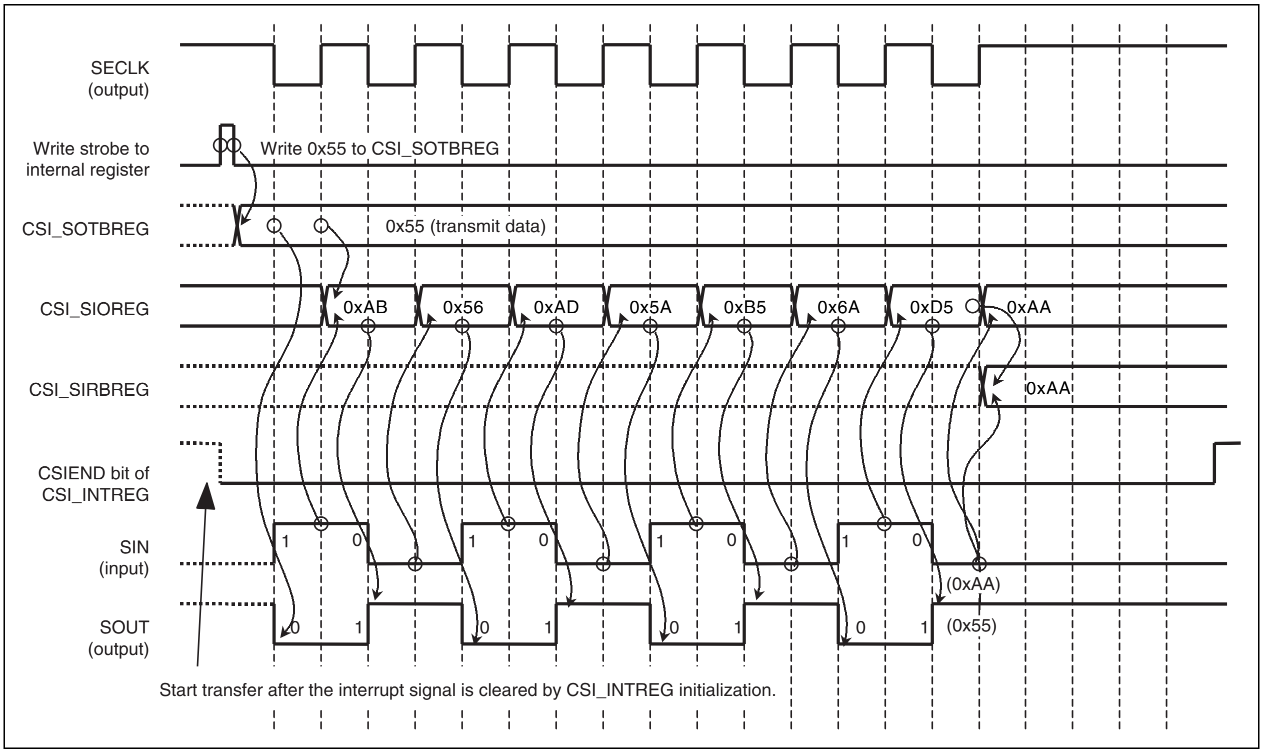
21.4.1 Single transfer mode timing (8-bit transfer)

When CSI_MODEREG's TRMD = 1, CCL = 0, DIR = 1, AUTO = 0, and the CSI_CLKSELREG register's CKP = 0:

When transmit data is written to the CSI_SOTBREG register, after half a clock (SECLK), the SECLK clock is output in 8-clock units. Transmit data is output from SOUT in synchronization with the SECLK clock. The transmit/receive completion interrupt (CSIEND bit of CSI_INTREG) becomes active at the end of transmission.

Because the interrupt signal is not cleared at the start of transfer, be sure to clear it using CSI_INTREG.

Figure 21-1. Single Transfer Mode (8-Bit Transfer)



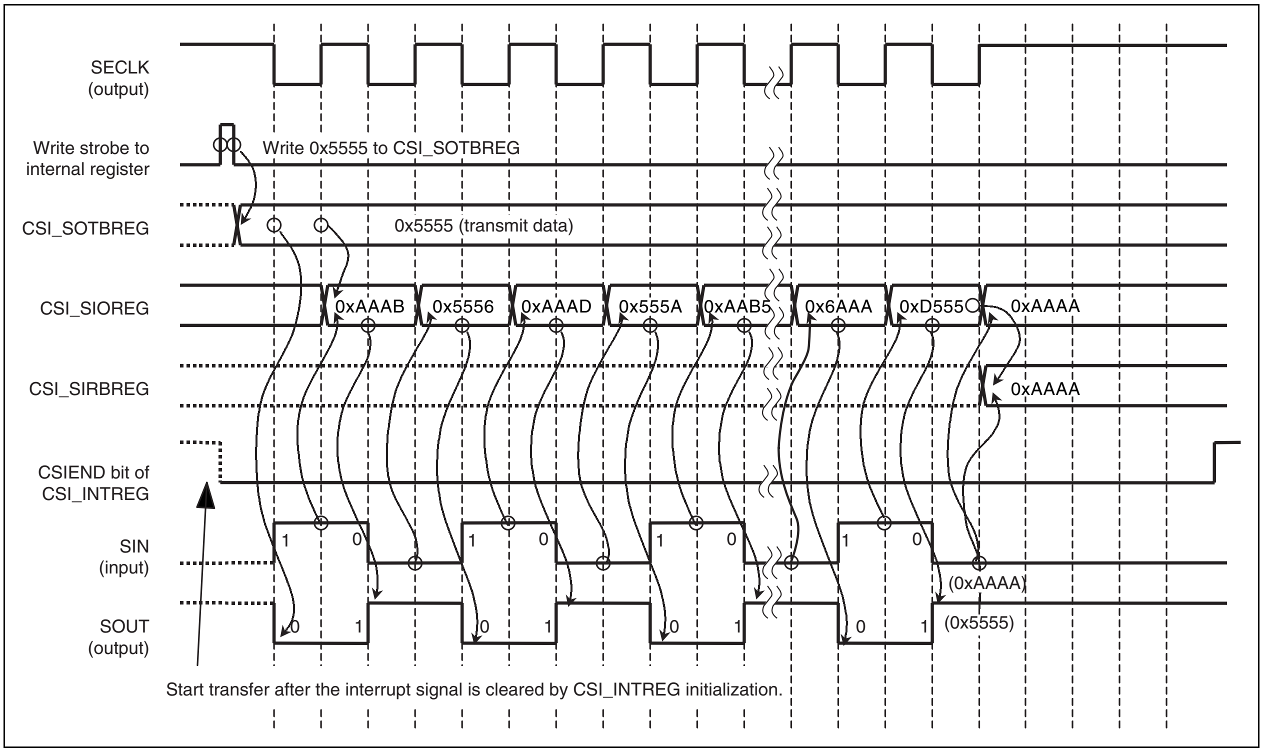
21.4.2 Single transfer mode timing (16-bit transfer)

When CSI_MODEREG's TRMD = 1, CCL = 1, DIR = 1, AUTO = 0, and CSI_CLKSELREG's CKP = 0:

When transmit data is written to the CSI_SOTBREG register, after half a clock (SECLK), the SECLK clock is output in 16-clock units. Transmit data is output from SOUT in synchronization with the SECLK clock. The transmit/receive completion interrupt (CSIEND bit of CSI_INTREG) becomes active at the end of transmission.

Because the interrupt signal is not cleared at the start of transfer, be sure to clear it using CSI_INTREG.

Figure 21-2. Single Transfer Mode (16-Bit Transfer)



21.5 Details Concerning Sequential Transfer Mode

Sequential transfer mode is entered when the AUTO bit of the CSI_MODEREG register is set.
Both the transmit/receive FIFO buffers and transmit/receive DMA can be used in sequential transfer mode.

21.5.1 When transmit/receive FIFO is not used

When the T_FIFOE or R_FIFOE bit of the CSI_CNTREG register is cleared in sequential transfer mode, transmit/receive data must be exchanged in synchronization with the transfer cycle.

(1) When TRMD of CSI_MODEREG is 0:

Reception is started by reading data from the CSI_SIRBREG register.
Reception is performed continuously each time data is read from CSI_SIRBREG while CSOT is 1 following the start of reception. In order to receive data continuously, data must therefore be read from the CSI_SIRBREG register between the start and end of reception of 1 data.
The transmit/receive completion interrupt (CSIEND) is generated every time 1 data is received.
To end reception, abort reading from the CSI_SIRBREG register and wait until 2 data items have been received.
At the end of reception, the CSOT bit changes to 1, the final data is stored in the CSI_SIOREG register, and the penultimate data is stored in the CSI_SIRBREG register.

(2) When TRMD of CSI_MODEREG is 1:

The data to be transmitted first is written to the CSI_SOTBFREG register, and transfer is then started by writing to the CSI_SOTBREG register. CSOT of CSI_MODEREG becomes 1 at the start of transfer. CSI_SOTBFREG data transmission is performed first, followed by data transmission of the CSI_SOTBREG register.
The transmit/receive completion interrupt (CSIEND) is generated every time 1 data has been transmitted.
To transfer data continuously, it is necessary to write the next transmit data to the CSI_SOTBREG register while data of the CSI_SOTBREG is being transmitted.
The CSOT changes to 0 at the end of transfer. Be aware that the CSI_SOTBFREG register must be written to at the end of each transfer.

21.5.2 When transmit/receive FIFO is used

The transmit/receive FIFO can be used by setting the T_FIFOE or R_FIFOE bit of the CSI_CNTREG register. When the transmit/receive FIFO is used, DMA transfer can also be used.

(1) When DMA is not used

The following control operations are necessary when using the transmit/receive FIFO without using DMA in sequential transfer mode.

(a) When TRMD of CSI_MODEREG is 0:

Reception is started by reading data from the CSI_SIRBREG register.
When the R_TRGEN bit of the CSI_CNTREG register is 0 following the start of reception, reception continues as long as space remains in the receive FIFO buffers. In the receive FIFO, data is stored in the order it was received, and can be read from the CSI_IFIFOREG register in that order.
When the R_TRGEN bit is 1, reception is stopped after the amount of data set in the R_TRG(2:0) bits of the CSI_FIFOTRGREG register has been received, and the receive trigger interrupt (R_OVER of CSI_INTREG is 1) is generated.
The transmit/receive completion interrupt (CSIEND of CSI_INTREG) is generated each time 1 data is received.

(b) When TRMD of CSI_MODEREG is 1:

Transmission/reception is started by writing the data to be transmitted to the CSI_OFIFOREG register.

The CSOT bit of CSI_MODEREG register becomes 1 at the start of transfer.

When the T_TRGEN bit of the CSI_CNTREG register is 0 following the start of transmission, transmission continues as long as data remains in the transmit FIFO buffers. Data can be written to the CSI_OFIFOREG register until the transmit FIFO buffer is FULL.

When the T_TRGEN bit is 1, transmission is stopped after the amount of data set in the T_TRG(2:0) bit of the CSI_FIFOTRGREG register has been transmitted, and the transmit trigger interrupt (T_EMP of CSI_INTREG is 1) is generated.

The transmit/receive completion interrupt (CSIEND of CSI_INTREG) is generated each time 1 data is transmitted.

(2) When DMA is used

The following control operations are necessary when using both the transmit/receive FIFO and DMA in sequential transfer mode. Firstly, set the DMAAU and DCU before setting the T_DMAEN and R_DMAEN bits of the CSI_CNTREG register.

(a) When TRMD = 0:

Reception is started by reading data from the CSI_SIRBREG register.

When the R_TRGEN bit of the CSI_CNTREG register is 0 following the start of reception, reception continues as long as space remains in the receive FIFO buffers and the page set for DMA has not been reached. The memory area set by the DMAAU is written to each time receive data is stored in the receive FIFO buffers.

The transmit/receive completion interrupt (CSIEND of CSI_INTREG) is generated each time 1 data is received.

When the R_TRGEN bit is 1, reception is stopped after the amount of data set in the R_TRG(2:0) bits of the CSI_FIFOTRGREG register has been received, and the receive trigger interrupt (R_OVER of CSI_INTREG is 1) is generated.

Reception is also stopped when the page set by DMA is reached, and the receive-page interrupt (R_P1STP or R_P2STP of CSI_INTREG) is generated. In this case, because data that has not been DMA-transferred is sometimes left in the receive FIFO buffers, be sure to check for this kind of data using the CSI_IFIFOVREG register, and read any unsent data found from the CSI_IFIFOREG register.

If reception is terminated due to an interrupt, it will not resume even if the interrupt is cancelled.

(b) When TRMD = 1:

When the CSI_CNTREG register's CSIRST bit is 0, T_FIFOE bit 1, and T_DMAEN bit 1, and the CSIE bit of the CSI_MODEREG register is 1, data transmitted by the DMA transfer from the memory area set by the DMAAU is stored in the transmit FIFO buffers, and transmission is started.

The transmit/receive completion interrupt (CSIEND of CSI_INTREG) is generated every time 1 data has been transmitted.

When the T_TRGEN bit of the CSI_CNTREG register is 0 following the start of transmission, transmission continues as long as data remains in the transmit FIFO buffers.

When the T_TRGEN bit is 1, transmission is stopped after the amount of data set in the T_TRG bit of the CSI_FIFOTRGREG register has been transmitted, and the transmit trigger interrupt (T_EMP of CSI_INTREG is 1) is generated. In this case, data is sometimes left in the transmit FIFO buffers. Transmission of the remaining data can be started by clearing the interrupt (writing 1 to T_EMP of CSI_INTREG).

Transmission is also stopped when the page set by DMA is reached, and the transmit-page interrupt (T_P1STP or T_P2STP of CSI_INTREG) is generated.

If reception is terminated by an interrupt that occurred during reception, it will not resume even if the interrupt is cancelled.

CHAPTER 22 FIR (FAST IrDA INTERFACE UNIT)

22.1 General

This unit supports the FIR (Fast SIR) transfer rate; 0.576 Mbps, 1.152 Mbps, and 4 Mbps in the IrDA 1.1 high-speed infrared communication physical layer standard. SIR (up to 1.152 kbps) is not supported.

22.2 Register Set

The FIR registers are listed below.

Table 22-1. FIR Registers

Address	R/W	Register Symbol	Function
0x0F00 0840	R/W	FRSTR	FIR reset register
0x0F00 0842	R/W	DPINTR	DMA page interrupt register
0x0F00 0844	R/W	DPCNTR	DMA page control register
0x0F00 0850	W	TDR	Transmit data register
0x0F00 0852	R	RDR	Receive data register
0x0F00 0854	R/W	IMR	Interrupt mask register
0x0F00 0856	R/W	FSR	FIFO setup register
0x0F00 0858	R/W	IRSR1	IR setup register 1
0x0F00 085C	R/W	CRCSR	CRC setup register
0x0F00 085E	R/W	FIRCR	FIR control register
0x0F00 0860	R/W	MIRCR	MIR control register
0x0F00 0862	R/W	DMACR	DMA control register
0x0F00 0864	R/W	DMAER	DMA enable register
0x0F00 0866	R	TXIR	Transmission indication register
0x0F00 0868	R	RXIR	Reception indication register
0x0F00 086A	R	IFR	Interrupt flag register
0x0F00 086C	R	RXSTS	Reception status register
0x0F00 086E	R/W	TXFL	Transmit frame length register
0x0F00 0870	R/W	MRXF	Maximum receive frame length register
0x0F00 0874	R	RXFL	Receive frame length register

These registers are described in detail below.

22.2.1 FRSTR (0x0F00 0840)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	FRST
R/W	R	R	R	R	R	R	R	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:1	RFU	Reserved. Write 0. 0 is returned after a read.
0	FRST	FIR reset. Write 0 when releasing reset. 1: Reset 0: Normal

22.2.2 DPINTR (0x0F00 0842)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	FDPINT5	FDPINT4	FDPINT3	FDPINT2	FDPINT1
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:5	RFU	Reserved. Write 0. 0 is returned after a read.
4	FDPINT5	FIR interrupt 1: Generated 0: Normal
3	FDPINT4	Receive DMA 2-page interrupt request. Cleared to 0 when 1 is written. 1: Generated 0: Normal
2	FDPINT3	Transmit DMA 2-page interrupt request. Cleared to 0 when 1 is written. 1: Generated 0: Normal
1	FDPINT2	Receive DMA 1-page interrupt request. Cleared to 0 when 1 is written. 1: Generated 0: Normal
0	FDPINT1	Transmit DMA 1-page interrupt request. Cleared to 0 when 1 is written. 1: Generated 0: Normal

This register indicates the generation of the DMA page interrupts of the FIR unit.

The FDPINT4 bit indicates that the DMA buffer (receive side) has become full (2 pages). DMARQ is stopped when this bit is set to 1.

The FDPINT3 bit indicates that the DMA buffer (transmit side) has become full (2 pages). DMARQ is stopped when this bit is set to 1.

The FDPINT2 bit indicates that the first page of the DMA buffer (receive side) has become full. DMARQ is stopped when this bit is set to 1 while bit 0 of DPCNTR is 1.

The FDPINT1 bit indicates that the first page of the DMA buffer (transmit side) has become full. DMARQ is stopped when this bit is set to 1 while bit 0 of DPCNTR is 1.

Cautions 1. If FDPINT(4:3) are set to 1, the last transfer data is not guaranteed.

2. If FDPINT(2:1) are set to 1 while the FDPCNT bit of the DPCNTR register is set to 1, the last transfer data is not guaranteed.

22.2.3 DPCNTR (0x0F00 0844)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	FDPCNT
R/W	R	R	R	R	R	R	R	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:1	RFU	Reserved. Write 0. 0 is returned after a read.
0	FDPCNT	Sets whether or not to stop the DMA transfer at the first page. 1: 1-page transfer 0: 2-page transfer

Cautions 1. When the FDPCNT bit is set to 1, the last data of the first page is not guaranteed.

2. When the FDPCNT bit is set to 0, the last data of the second page is not guaranteed.

22.2.4 TDR (0x0F00 0850)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	W	W	W	W	W	W	W	W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0
R/W	W	W	W	W	W	W	W	W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:8	RFU	Reserved. Write 0. 0 is returned after a read.
7:0	TDR(7:0)	Transmit data FIFO

This register is used to store the address to which data is written to the transmit data store FIFO.

Up to 64- or 32-byte data (determined by bit 3 of the FSR register) is stored in the transmit data store FIFO.

The transmit data FIFO is used as follows.

(1) Write

Data is written to the transmit data FIFO while the IrDA is operating.

When a write operation is completed, the write pointer of the transmit data FIFO is incremented. However, if data is written when the transmit data FIFO is full, the write pointer is not incremented.

After writing the frame size to the TXFL register while transmission is not busy (start enable), transfer starts when the number of writes to this register exceeds the threshold (start 1).

Even if short of the threshold, reaching the frame size starts transfer (start 2).

(2) Read

After frame transmission is completed, the sequencer reads the transmit data during the data transfer sequence, and the read pointer is incremented.

If read while the transmit FIFO is empty, a transmit underrun error occurs. This stops the current frame transmission and then starts the abort frame transmission. The following frames scheduled to be transmitted next are not transmitted.

22.2.5 RDR (0x0F00 0852)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:8	RFU	Reserved. Write 0. 0 is returned after a read.
7:0	RDR(7:0)	Receive data FIFO

This register is used to store the address from which data is read from the receive data store FIFO.

Up to 64- or 32-byte data (determined by bit 3 of the FSR register) is stored in the receive data store FIFO.

The receive data store FIFO is used as follows.

(1) Write

During frame data reception, the sequencer writes the receive data to the data transfer sequence, and the write pointer is incremented.

If data is written when the unread data in the receive data FIFO reaches the maximum volume, the receive overrun error occurs and the current frame reception is ended. At this time, the write pointer is not incremented.

After the receive data FIFO is cleared, if the number of received frames is less than 7 frames, it is possible to continue frame reception. To receive 8 or more frames, read all the data and statuses that are already received from the receive data FIFO, then clear the receive data FIFO and restart reception.

(2) Read

Data is read from the receive data store FIFO while the IrDA is operating. (A pre-buffer is not incorporated.)

When a read operation is completed, the read pointer of the receive data store FIFO is incremented. However, it is not incremented when the receive FIFO is empty.

When the number of read frames reaches the receive frame size, an interrupt is generated and the VALID bit of the RXSTS register is set.

Cautions 1. If data is read when the receive FIFO is empty (read pointer = write pointer), it may contend with the sequencer's write operation, which cause undefined data.

2. The error generated by a read underrun is not reported in the VR4131.

22.2.6 IMR (0x0F00 0854)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:8	RFU	Reserved. Write 0. 0 is returned after a read.
7:0	IMR(7:0)	These bits are used to enable/disable interrupt output. 1: Enable 0: Disable

This register sets whether to report externally when an interrupt is generated. Each bit corresponds to the equivalent IFR register bit. When interrupt output is enabled and the corresponding bit is 1, interrupt output becomes active.

Caution The IFR register is set irrespective of this register's setting.

22.2.7 FSR (0x0F00 0856)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RX_TH1	RX_TH0	TX_TH1	TX_TH0	F_SIZE	TXF_CLR	RXF_CLR	TX_STOP
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:8	RFU	Reserved. Write 0. 0 is returned after a read.
7:6	RX_TH(1:0)	Specifies the receive FIFO's threshold. RX_TH (1:0) F_SIZE = 0 F_SIZE = 1 11 26 bytes 48 bytes 10 16 bytes 32 bytes 01 4 bytes 8 bytes 00 1 byte 1 byte
5:4	TX_TH(1:0)	Specifies the transmit FIFO's threshold. TX_TH (1:0) F_SIZE = 0 F_SIZE = 1 11 26 bytes 48 bytes 10 16 bytes 32 bytes 01 8 bytes 16 bytes 00 1 byte 1 byte
3	F_SIZE	Specifies the maximum size of transmit/receive FIFO. 1: 64 bytes 0: 32 bytes
2	TXF_CLR	Transmit FIFO clear trigger (read value = 0)
1	RXF_CLR	Receive FIFO clear trigger (read value = 0)
0	TX_STOP	Transmission stop trigger (read value = 0)

This register is used to make various settings for the transmit/receive FIFO.

When the TXF_CLR bit is set, the pointers of the transmit data FIFO and transmit frame size FIFO are initialized.

When the RXF_CLR bit is set, the pointers of the receive data FIFO, receive frame size FIFO, and receive status FIFO are initialized.

When the TX_STOP bit is set, the current frame transmission is stopped and abort frame transmission starts. The following frames scheduled to be transmitted next are not transmitted. Also, setting this bit causes the DMA operation to be stopped, and then a DMA completion interrupt to be output.

Cautions 1. During transmission/reception, the contents of bits 7 to 3 of the FSR register must not be changed (refresh is possible).

2. The data in the FIFO is not cleared even if the TXF_CLR or RXF_CLR bit is set (clearing the pointer).

3. Set the TX_STOP bit to stop DMA operation after data transfer is completed, regardless of whether the DMA operation is transmission or reception. In the case of reception, however, the DMA should be stopped after confirming the command bit of the transferred data.

22.2.8 IRSR1 (0x0F00 0858)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	IRDA_EN	RFU	RFU	RFU	RFU	RFU	IRDA_MD	MIR_MD
R/W	R/W	R	R	R	R	R	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

<table>
<tr><th>Bit</th><th>Name</th><th>Function</th></tr>
<tr><td>15:8</td><td>RFU</td><td>Reserved. Write 0. 0 is returned after a read.</td></tr>
<tr><td>7</td><td>IRDA_EN</td><td>FIR operation enable
1: Enable
0: Disable</td></tr>
<tr><td>6:2</td><td>RFU</td><td>Reserved. Write 0. 0 is returned after a read.</td></tr>
<tr><td>1:0</td><td>IRDA_MD,
MIR_MD</td><td>Speifies IrDA/MIR mode
<table>
<tr><th>IRDA_MD</th><th>MIR_MD</th><th>Operating Mode</th><th>Frequency</th><th>Modulation Method</th></tr>
<tr><td>1</td><td>1</td><td>MIR Half mode</td><td>0.576 MHz</td><td>Bit Stream/Stuff</td></tr>
<tr><td>1</td><td>0</td><td>MIR Full mode</td><td>1.152 MHz</td><td>Bit Stream/Stuff</td></tr>
<tr><td>0</td><td>X</td><td>FIR mode</td><td>8 MHz</td><td>4 PPM</td></tr>
</table></td></tr>
</table>

When the IRDA_EN bit is set, the peripheral main block's reset is released and clock supply starts.

Pulse output level changes according to operation mode changes by IRDA_MD and MIR_MD. The operation mode should be changed after changing the IrDA operation to the disabled state (by setting IRDA_EN = 0). The output level does not change because output latch is reset.

Once the mode is changed, be sure to switch bit inversion of I/O data ON/OFF by setting DATA_INV of the CRCSR register.

Example) Sequence of changing operation mode from FIR mode to MIR Full mode

```
clr1    0x7, IRSR1     ; Disable IrDA operation
set1    0x1, IRSR1     ; Change the mode
set1    0x0, CRCSR     ; Set data inversion
set1    0x7, IRSR1     ; Enable IrDA operation
```

Caution During transmission/reception, this register must not be changed (refresh is possible).

22.2.9 CRCSR (0x0F00 085C)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	TX_EN	RX_EN	4PPM_DIS	DPLL_DIS	RFU	NON_CRC	CRC_INV	DATA_INV
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:8	RFU	Reserved. Write 0. 0 is returned after a read.
7	TX_EN	Masking enable of transmit start enable flag. 1: Enable 0: Disable
6	RX_EN	Receive operation enable. 1: Enable 0: Disable
5	4PPM_DIS	4PPM modulation disable (for debugging). 1: Disable 0: Enable
4	DPLL_DIS	Bit correction disable (for debugging). 1: Disable 0: Enable
3	RFU	Reserved. Write 0. 0 is returned after a read.
2	NON_CRC	Addition of CRC to transmit frame (for debugging). 1: Do not add CRC 0: Add CRC
1	CRC_INV	Inversion of CRC to transmit frame (for debugging). 1: Inverted CRC 0: Normal CRC (not inverted)
0	DATA_INV	Inversion of transmit/receive data. 1: Inverted 0: Normal (not inverted)

The TX_EN bit is used to set whether to mask sequence transition to the transmission enable state entered by writing the TXFL register.

The RX_EN bit is used to set whether to release masking of the receive line and to enable data sampling and receive clock generation.

The 4PPM_DIS bit is used to set whether to disable 4PPM modulation of transmit data.

The DPLL_DIS bit is used to set whether to disable bit correction of receive data.

The NON_CRC bit is used to set whether to transmit with STO after the transmission of transmit frame data.

The CRC_INV bit is used to set whether to invert CRC to create an illegal CRC in the normal routine.

The DATA_INV bit is used to set whether to invert I/O data. Be sure to set as 0 (normal) in FIR and 1 (invert) in MIR.

Caution During transmission/reception, the data in this register must not be changed (refresh is possible).

22.2.10 FIRCR (0x0F00 085E)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	PA_LEN2	PA_LEN1	PA_LEN0	W_PULSE1	W_PULSE0	F_WIDTH2	F_WIDTH1	F_WIDTH0
R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W
RTCRST	1	0	0	0	0	1	0	1
After reset	1	0	0	0	0	1	0	1

<table>
<tr><th>Bit</th><th>Name</th><th>Function</th></tr>
<tr><td>15:8</td><td>RFU</td><td>Reserved. Write 0. 0 is returned after a read.</td></tr>
<tr><td>7:5</td><td>PA_LEN(2:0)</td><td>Specifies the number of PA (preamble).
111: 32
100: 16
011: 4
010: 2
001: 1
Other: RFU</td></tr>
<tr><td>4:3</td><td>W_PULSE(1:0)</td><td>These bits are used to specify the undefined receive pulse width area.
11: 10 to 11 clocks
10: 9 to 10 clocks
01: 8 to 9 clocks
00: 7 to 8 clocks</td></tr>
<tr><td>2:0</td><td>F_WIDTH(2:0)</td><td>These bits are used to specify the FIR pulse modulation width.
<table>
<tr><th>F_WIDTH(2:0)</th><th>Single pulse</th><th>Double pulse</th></tr>
<tr><td>101 (initial value)</td><td>6 clocks</td><td>12 clocks</td></tr>
<tr><td>100</td><td>5 clocks</td><td>11 clocks</td></tr>
<tr><td>011</td><td>4 clocks</td><td>10 clocks</td></tr>
<tr><td>010</td><td>3 clocks</td><td>9 clocks</td></tr>
<tr><td>001</td><td>2 clocks</td><td>8 clocks</td></tr>
<tr><td>000</td><td>1 clock</td><td>7 clocks</td></tr>
<tr><td>Other</td><td>Setting prohibited</td><td>Setting prohibited</td></tr>
</table>
</td></tr>
</table>

This register is used to control FIR operations.

The PA_LEN bits are used to specify the number of PA added to the FIR transmit frame.

The W_PULSE bits are used to specify the undefined receive pulse width area to recognize a single or double pulse. A pulse width within the undefined receive pulse width area is recognized as a single pulse. A pulse width outside the undefined receive pulse width area is recognized as a double pulse.

The F_WIDTH bits specify how many cycles of the reference clock (48 MHz) per width the FIR output pulse should be modulated.

Caution During transmission/reception, the contents of this register must not be changed.

22.2.11 MIRCR (0x0F00 0860)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	STA_LEN2	STA_LEN1	STA_LEN0	M_WIDTH4	M_WIDTH3	M_WIDTH2	M_WIDTH1	M_WIDTH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	1	0	0	1	0	0	1
After reset	0	1	0	0	1	0	0	1

Bit	Name	Function
15:8	RFU	Reserved. Write 0. 0 is returned after a read.
7:5	STA_LEN(2:0)	Specifies the number of STA (start flag). 111: 32 100: 16 011: 4 010: 2 (initial value) 001: 1 Others: RFU
4:0	M_WIDTH(4:0)	Specifies MIR pulse modulation width. 11111: 32 clocks : 10100: 21 clocks : 01001: 10 clocks (initial value) : 00001: 2 clocks 00000: 1 clock

This register is used to control the MIR operation.

The STA_LEN(2:0) bits are used to specify the number of STA added to the MIR transmit frame.

The M_WIDTH(4:0) bits specify how many cycles of the reference clock (48 MHz) per width the MIR output pulse should be modulated.

The nominal pulse width of MIR is 1/4. Therefore, the following setting should normally be made.

MIR Full mode (1.152 MHz) = 01001 (rate 10/42)
MIR Half mode (0.576 MHz) = 10100 (rate 21/83)

Caution During transmission/reception, the contents of this register must not be changed.

22.2.12 DMACR (0x0F00 0862)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	ACES_MD	TRANS_MD	RFU	RFU	RFU	DEMAND2	DEMAND1	DEMAND0
R/W	R/W	R/W	R	R	R	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:8	RFU	Reserved. Write 0. 0 is returned after a read.
7	ACES_MD	Selects the access mode. Write 0 when writing. 0 is returned after a read.
6	TRANS_MD	Specifies the transfer direction. 1: RDR register → Memory 0: Memory → TDR register
5:3	RFU	Reserved. Write 0. 0 is returned after a read.
2:0	DEMAND(2:0)	These bits are used to specify the demand size. 111: RFU 110: 7 101: 6 100: 5 011: 4 010: 3 001: 2 000: RFU

Caution During DMA operation (both the master side and IrDA side), the contents of this register must not be changed.

22.2.13 DMAER (0x0F00 0864)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	DMA_BUSY	DMA_EN
R/W	R	R	R	R	R	R	R	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:2	RFU	Reserved. Write 0. 0 is returned after a read.
1	DMA_BUSY	DMA busy status 1: DMA operation in progress 0: DMA operation not in progress
0	DMA_EN	DMA operation enable trigger 1: Enable 0: Disable

The DMA_BUSY bit is set when the DMA_EN bit is set to 1, and is cleared when bit 0 (TX_STOP) of the FSR register is set.

Even if 0 is written to the DMA_EN bit during DMA operation, DMA operation does not stop and writing becomes invalid.

22.2.14 TXIR (0x0F00 0866)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	TX_BUSY	RFU	LAST_TFL	TX_TH_OV	RFU	TXF_UNDR	TXF_FULL	TXF_EMP
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:8	RFU	Reserved. Write 0. 0 is returned after a read.
7	TX_BUSY	Transmission busy 0: Not in transmission 1: In transmission
6	RFU	Reserved. Write 0. 0 is returned after a read.
5	LAST_TFL	Last transmission frame 0: Normal 1: Last frame
4	TX_TH_OV	Transmission FIFO threshold over 0: Normal 1: Excesses
3	RFU	Reserved. Write 0. 0 is returned after a read.
2	TXF_UNDR	Transmission FIFO underrun 0: Normal 1: Data is read
1	TXF_FULL	Transmission FIFO full 0: Normal 1: FIFO is full
0	TXF_EMP	Transmission FIFO empty 0: Normal 1: FIFO is empty

The TX_BUSY bit is set to 1 during the period between PA (in FIR)/STA (in MIR) transmission and abort transmission.

The LAST_TFL bit indicates whether or not data exists in the transmit frame size FIFO. This bit changes when the STA transmission sequence ends. Its default value is 1.

The TX_TH_OV bit indicates whether or not the data size within the transmit FIFO exceeds the threshold.

The TXF_UNDR bit is set when data is read when no data is in the transmit FIFO.

The TXF_FULL bit is set when there is no writable space in the transmit FIFO.

The TXF_EMP bit is set when there is no more data to be read in the transmit FIFO.

22.2.15 RXIR (0x0F00 0868)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RX_BUSY	END_DATA	LAST_RFL	RX_TH_O	RFU	RFU	RXF_FULL	RXF_EMP
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:8	RFU	Reserved. Write 0. 0 is returned after a read.
7	RX_BUSY	Reception busy 0: Reception in progress 1: Reception no in progress
6	END_DATA	Frame last data 0: Normal 1: Exists
5	LAST_RFL	Last reception frame 0: Normal 1: Result is stored
4	RX_TH_O	Reception FIFO threshold over 0: Normal 1: Exceeds
3:2	RFU	Reserved. Write 0. 0 is returned after a read.
1	RXF_FULL	Reception FIFO full 0: Normal 1: FIFO is full
0	RXF_EMP	Reception FIFO empty 0: Normal 1: FIFO is empty

The RX_BUSY bit is set to 1 during the period between when PA (in FIR) or STA (in MIR) is detected and when reception ends.

The END_DATA bit is set when the last data of the frame that is received completely exists in the FIFO.

The LAST_RFL bit is set when the reception result (frame size and status) of the 7th frame is stored.

The RX_TH_O bit is set when the data size within the receive FIFO exceeds the threshold.

The RXF_FULL bit is set when there is no writable space in the receive FIFO.

The RXF_EMP bit is set when there is no data to be read in the receive FIFO.

This register's initial value is the value immediately after the IrDA operation is enabled or after the receive FIFO is cleared. 0x00 is also read while the operation is stopped.

Caution This register can be read only in IrDA mode.

22.2.16 IFR (0x0F00 086A)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	TX_ABORT	TX_ERR	RX_VALID	DMA_END	RX_END	TX_END	TX_WR_RQ	RX_RD_RQ
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:8	RFU	Reserved. Write 0. 0 is returned after a read.
7	TX_ABORT	Abort frame transmission end interrupt 0: Normal 1: Generated
6	TX_ERR	Transmission error interrupt 0: Normal 1: Generated
5	RX_VALID	Reception result valid interrupt 0: Normal 1: Generated
4	DMA_END	DMA end interrupt 0: Normal 1: Generated
3	RX_END	Reception end interrupt 0: Normal 1: Generated
2	TX_END	Transmission end interrupt 0: Normal 1: Generated
1	TX_WR_RQ	Transmission data write request interrupt 0: Normal 1: Generated
0	RX_RD_RQ	Reception data read request interrupt 0: Normal 1: Generated

This register indicates the generation of FIR interrupt requests.

The TX_ABORT bit is set when the abort frame is transmitted and the following frame's transfer reservation is cancelled.

The TX_ERR bit is set when a transmission error (such as a forcible stop) occurs.

The RX_VALID bit is set when the last data of the frame is read from the receive FIFO and the received status becomes valid.

The DMA_END bit is set when the DMA operation ends.

The RX_END bit is set when the STO is detected for each reception frame.

The TX_END bit is set when the STO is transmitted for each transmission frame.

The TX_WR_RQ bit is set when the transmission data write request occurs.

The RX_RD_RQ bit is set when the reception data read request occurs.

If this register is read, flags set at that time are all cleared.

22.2.17 RXSTS (0x0F00 086C)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	VALID	RFU	RFU	RXF_OV	CRC_ERR	ABORT	MRXF_OV	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:8	RFU	Reserved. Write 0. 0 is returned after a read.
7	VALID	Receive data read status 0: Not completed 1: Completed
6:5	RFU	Reserved. Write 0. 0 is returned after a read.
4	RXF_OV	Receive FIFO overrun error 0: Normal 1: Overrun
3	CRC_ERR	CRC error 0: Normal 1: CRC error
2	ABORT	Abort detection error 0: Normal 1: Detected
1	MRXF_OV	Maximum receive frame size error 0: Normal 1: Overrun
0	RFU	Reserved. Write 0. 0 is returned after a read.

This register indicates the occurrence of various errors during transmission/reception.

The VALID bit is set to 1 when receive data of one frame is read completely.

The RXF_OV bit is set when a receive operation is stopped due to receive FIFO overrun.

The CRC_ERR bit is set when the receive result CRC does not match the expected value.

The ABORT bit is set when the receive operation is stopped by abort frame detection.

The MRXF_OV bit is set when the receive operation is stopped by maximum receive frame size overrun.

Data of up to 7 frames can be stored in the receive status store FIFO. This FIFO is initialized by setting bit 1 of the FSR register.

The receive status FIFO is used as follows.

(1) Write (bits 4 to 1)

The receive status is written to these bits at the same timing of writing data to the receive frame length register. This register shares the write pointer with the receive frame length register.

(2) Write (bit 7)

This bit is set to 1 when the data of receive frame size is read from the FIFO. While this bit is 1, data is recognized as valid.

(3) Read

This register shares the read pointer with the receive frame length register.
The read pointer is incremented by reading the RXFL register after valid data is read from this register.

22.2.18 TXFL (0x0F00 086E)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	TXFL12	TXFL11	TXFL10	TXFL9	TXFL8
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	TXFL7	TXFL6	TXFL5	TXFL4	TXFL3	TXFL2	TXFL1	TXFL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:13	RFU	Reserved. Write 0. 0 is returned after a read.
12:0	TXFL(12:0)	Transmit frame size. 0x1FFF: RFU : 0x0802: RFU 0x0801: 2K(+2) bytes : 0x0001: 2 bytes 0x0000: 1 byte

This register is used to set the address for data write to the transmit frame size store FIFO. Set a value of transmit size −1 to this register. The setting range is 1 to 2K+2 bytes.

Data of up to 7 frames can be stored in the transmit frame size data store FIFO.

The FIFO is initialized by setting bit 2 of the FSR register.

The transmit frame size FIFO is used as follows.

(1) Write

The data transmit size of frames to be transferred is written to this register.

Transmission is enabled when data is written to this register in a state other than the transmission busy state (after FIFO initialization and after transmission completion).

The number of frames specified by this register are transferred continuously (back-to-back transfer).

During single frame transfer, the FIFO should be initialized at each 1-frame transfer completion to restart the transmit operation.

(2) Read

The sequencer reads the transmission size from this register after the STA flag of the transmission frame is transmitted completely. The read pointer is then incremented.

Caution If data exists in the FIFO when the STO transmit sequence is completed, continuous transfer mode is entered. When multiple frames are transferred, be sure to write data to the TXFL register before the STO transmit sequence is completed.

22.2.19 MRXF (0x0F00 0870)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	MRXF12	MRXF11	MRXF10	MRXF9	MRXF8
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	MRXF7	MRXF6	MRXF5	MRXF4	MRXF3	MRXF2	MRXF1	MRXF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:13	RFU	Reserved. Write 0 to these bits. 0 is returned after a read.
12:0	MRXF(12:0)	Specifies receivable maximum frame size. 0x1FFF: RFU : 0x0802: RFU 0x0801: 2K(+2) bytes : 0x0001: 2 bytes 0x0000: 1 byte

The maximum receive frame size is set in this register. When a 1-frame receive data exceeding the set value of this register is transferred to the receive FIFO, an error occurs, even though frame reception is not complete, and the current frame reception is terminated. When an error occurs, bit 1 is set to the receive status register.

If the number of received frames is less than 7 frames after clearance of the receive FIFO, it is possible to continue frame reception.

To receive 8 or more frames, read all the data and frames that are already received from the receive FIFO, then clear the receive FIFO and restart reception.

When receiving data via the DMA operation, set the DMA transfer size value by the following expression:

DMA receivable capacity = Set value × 7 frames

Caution Data exceeding the maximum size is not transferred to the FIFO.

22.2.20 RXFL (0x0F00 0874)

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RXFL12	RXFL11	RXFL10	RXFL9	RXFL8
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RXFL7	RXFL6	RXFL5	RXFL4	RXFL3	RXFL2	RXFL1	RXFL0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15:13	RFU	Reserved. Write 0. 0 is returned after a read.
12:0	RXFL(12:0)	Receive frame size. 0x1FFF: RFU : 0x0802: RFU 0x0801: 2K(+2) bytes : 0x0001: 2 bytes 0x0000: 1 byte

This register is used to set the address for data read from the receive frame size store FIFO. Set a value of receive size –1 to this register. The value can be set from 1 to 2 K+2 bytes (0000h to 0801h).

Data of up to 7 frames can be stored in the receive size frame data store FIFO.

The FIFO is initialized by setting bit 1 of the FSR register.

The receive frame size FIFO is used as follows.

(1) Write

When the frame reception is completed after its data is transferred (even if only 1 byte) to the receive FIFO, the sequencer writes the current transfer data size to this register, and the write pointer is incremented.

When the frame reception is completed before its data is transferred to the receive FIFO, write operation is not performed (lost frame).

(2) Read

Incrementation or the read pointer is enabled by reading valid data from the RXSTS register, and the next data can be read.

Caution If a receive operation ends abnormally, the data size transferred to the receive FIFO at that time is written to this register. After 7 frames of data are stored, the receive line is automatically masked. Accordingly, a frame that cannot store a receive result is not transferred to the FIFO.

The update condition of the read pointer of the receive frame size store FIFO is also valid in the test mode.

CHAPTER 23 CP0 HAZARDS

The VR4130 CPU core avoids contention of its internal resources by causing a pipeline interlock in cases when the contents of the destination register of an instruction are used as a source in the succeeding instruction. Therefore, instructions such as NOP must not be inserted between instructions.

However, interlocks do not occur in operations related to the CP0 registers and the TLB. Therefore, contention of internal resources should be considered when composing a program that manipulates the CP0 registers or the TLB. The CP0 hazards define the number of NOP instructions required to avoid contention of internal resources, or the number of instructions unrelated to contention. This chapter describes the CP0 hazards.

The CP0 hazards of the VR4130 CPU core are as or less stringent than those of the VR4000. Table 23-1 lists the CP0 hazards of the VR4130 CPU core. Code that complies with these hazards will run without modification on the VR4000 Series.

The contents of the CP0 registers or the bits in the “Source” column of this table can be used as a source after they are fixed.

The contents of the CP0 registers or the bits in the “Destination” column of this table can be available as a destination after they are stored.

Based on this table, the number of NOP instructions required between instructions related to the TLB is computed by the following formula, and so is the number of instructions unrelated to contention:

(Destination hazard number of A) – {(Source hazard number of B) + 1}

As an example, to compute the number of instructions required between an MTC0 and a subsequent MFC0 instruction, this is:

(5) – (3 + 1) = 1 instruction

The CP0 hazards do not generate pipeline interlocks. Therefore, the required number of instructions must be controlled by program.

Table 23-1. CP0 Hazards

Operation	Source		Destination	
	Source Name	No. of Cycles	Destination Name	No. of Cycles
MTC0			cpr rd	6
MFC0	cpr rd	4		
TLBR	Index, TLB	3	PageMask, EntryHi, EntryLo0, EntryLo1	6
TLBWI TLBWR	Index or Random, PageMask, EntryHi, EntryLo0, EntryLo1	3	TLB	6
TLBP	PageMask, EntryHi	3	Index	6
ERET	EPC or ErrorEPC, TLB	5	Status.EXL, Status.ERL	6
	Status	5		
CACHE Index Load Tag			TagLo, TagHi, PErr	6
CACHE Index Store Tag	TagLo, TagHi, PErr	4		
CACHE Hit ops.	cache line	4	cache line	6
Coprocessor usable test	Status.CU, Status.KSU, Status.EXL, Status.ERL	2		
Instruction fetch	EntryHi.ASID, Status.KSU, Status.EXL, Status.ERL, Status.RE, Config.K0C	2		
	TLB	2		
Instruction fetch exception			EPC, Status	6
			Cause, BadVAddr, Context, XContext	6
Interrupt signals	Cause.IP, Status.IM, Status.IE, Status.EXL, Status.ERL	2		
Load/Store	EntryHi.ASID, Status.KSU, Status.EXL, Status.ERL, Status.RE, Config.K0C, TLB	4		
	Config.AD, Config.EP	4		
	WatchHi, WatchLo	4		
Load/Store exception			EPC, Status, Cause, BadVAddr, Context, XContext	6

Caution If the setting of the K0 bit in the config register is changed to uncached mode by MTC0, the accessed memory area is switched to an uncached one at the instruction fetch of the second instruction after MTC0.

Remarks 1. The instruction following MTC0 must not be MFC0.

2. The five instructions following MTC0 to status register, via which the KSU bit is changed and the EXL and ERL bits are set, may be executed in the new mode, and not kernel mode. This can be avoided by setting the EXL bit first, waiting for the KSU bit to be set by the kernel, and then later changing KSU.

3. There must be two non-load, non-cache instructions between a store and a CACHE instruction directed to the same primary cache line as the store.

The status during execution of the following instructions for which CP0 hazards must be considered is described below.

(1) MTC0

Destination: The completion of writing to a destination register (CP0) of MTC0.

(2) MFC0

Source: The confirmation of a source register (CP0) of MFC0.

(3) TLBR

Source: The confirmation of the status of TLB and the Index register before the execution of TLBR.

Destination: The completion of writing to a destination register (CP0) of TLBR.

(4) TLBWI, TLBWR

Source: The confirmation of a source register of these instructions and registers used to specify a TLB entry.

Destination: The completion of writing to TLB by these instructions.

(5) TLBP

Source: The confirmation of the PageMask register and the EntryHi register before the execution of TLBP.

Destination: The completion of writing the result of execution of TLBP to the Index register.

(6) ERET

Source: The confirmation of registers containing information necessary for executing ERET.

Destination: The completion of the processor state transition by the execution of ERET.

(7) CACHE index load tag

Destination: The completion of writing the results of execution of this instruction to the related registers.

(8) CACHE index store tag

Source: The confirmation of registers containing information necessary for executing this instruction.

(9) Coprocessor usable test

Source: The confirmation of modes set by the bits of the CP0 registers in the "Source" column.

Examples 1. When accessing the CP0 registers in user mode after the contents of the CU0 bit of the status register are modified, or when executing instructions (TLB instructions, CACHE instructions, or branch instructions) that use the resources of the CP0.

2. When accessing the CP0 registers in the operating mode set by the status register after the KSU, EXL, and ERL bits of the status register are modified.

(10) Instruction fetch

Source: The confirmation of the operating mode and TLB necessary for instruction fetch.

Examples 1. When changing the operating mode from user to kernel and fetching instructions after the KSU, EXL, and ERL bits of the status register are modified.

2. When fetching instructions using the modified TLB entry after TLB modification.

(11) Instruction fetch exception

Destination: The completion of writing to registers containing information related to the exception when an exception occurs on instruction fetch.

(12) Interrupt

Source: The confirmation of registers judging the condition of occurrence of interrupt when an interrupt factor is detected.

(13) Load/store

Source: The confirmation of the operating mode related to the address generation of load/store instructions, TLB entries, the cache mode set in the K0 bit of the config register, and the registers setting the condition of occurrence of a watch exception.

Example When load/store instructions are executed in the kernel field after changing the mode from user to kernel.

(14) Load/store exception

Destination: The completion of writing to registers containing information related to the exception when an exception occurs on load or store operation.

Table 23-2 indicates examples of calculation.

Table 23-2. Calculation Example of CP0 Hazard and Number of Instructions Inserted

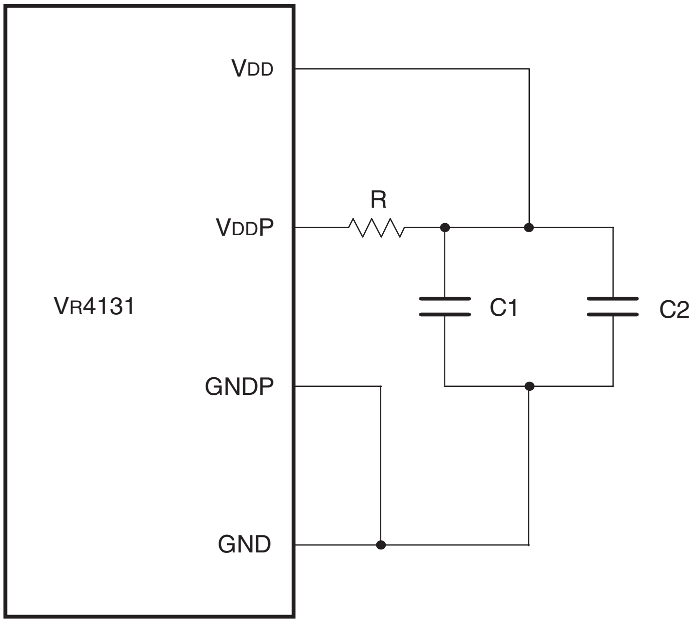
Destination	Source	Contending Internal Resource	Number of Instructions Inserted	Formula
TLBWR/TLBWI	TLBP	TLB Entry	2	6 – (3 + 1)
TLBWR/TLBWI	Load or store using newly modified TLB	TLB Entry	1	6 – (4 + 1)
TLBWR/TLBWI	Instruction fetch using newly modified TLB	TLB Entry	3	6 – (2 + 1)
MTC0, Status [CU]	Coprocessor instruction that requires the setting of CU	Status [CU]	3	6 – (2 + 1)
TLBR	MFC0 EntryHi	EntryHi	1	6 – (4 + 1)
MTC0 EntryLo0	TLBWR/TLBWI	EntryLo0	2	6 – (3 + 1)
TLBP	MFC0 Index	Index	1	6 – (4 + 1)
MTC0 EntryHi	TLBP	EntryHi	2	6 – (3 + 1)
MTC0 EPC	ERET	EPC	0	6 – (5 + 1)
MTC0 Status	ERET	Status	0	6 – (5 + 1)
MTC0 Status [IE]Note	Instruction that causes an interrupt	Status [IE]	0	6 – (5 + 1)

Note The number of hazards is undefined if the instruction execution sequence is changed by exceptions. In such a case, the minimum number of hazards until the IE bit value is confirmed may be the same as the maximum number of hazards until an interrupt request occurs that is pending and enabled.

CHAPTER 24 PLL PASSIVE COMPONENTS

Several passive components need to be connected externally for proper operation of the VR4131.
A connection example of the PLL passive components is shown in Figure 24-1.

Figure 24-1. Example of Connection of PLL Passive Components



Remarks **1.** Capacitors C1 and C2 and resistor R are mounted on the printed circuit board.
2. Since the value for the components depends upon the application system, the optimum values for each system should be decided after repeated experimentation.

It is essential to isolate the wiring for the PLL power supply (VDDP) from the regular power supply (VDD) and ground (GND). An example of the values of each component is shown below.

R = 100 Ω C1 = 0.1 μF C2 = 1.0 μF

Since the optimum values for the filter components depend upon the application and the system noise environment, these values should be considered as starting points for further experimentation within the user-specific application. In addition, a choke (inductor: *L*) can be considered for use as an alternative to a resistor (*R*) for use in filtering the power supply.

APPENDIX A DIFFERENCES BETWEEN REVISION 1.2 AND REVISION 2.0 OR LATER

This appendix describes the pins and functions that have been added to or eliminated from revision 2.0 and subsequent revisions of the VR4131. These descriptions are not applicable if revision 1.2 or an earlier revision is used.

A.1 Changes in Pin Functions

The following pin functions differ between VR4131 revision 1.2 or earlier, and VR4131 revision 2.0 and later.

Table A-1. Comparison of Pin Functions

Pin No.	Pin Name up to Revision 1.2	Pin Name from Revision 2.0
H17	N.C.	BKTGIO#
J18	IRDOUT#	IRDOUT#/JTDO
K17	MMUEN	JTDI/RMODE#
L17	N.C.	JTCK
M17	$V_{DD}3$	JTMS
N17	I.C.	JTRST#
P18	HLDAK#	HLDAK#/NWIREEN

A.1.1 Pin Functions

The signals added to revision 2.0 consist of the debug interface signals. This debug interface is used when using the N-Wire function. Since a number of signals are shared with the FIR pin, the N-Wire function and FIR function are used on a mutually exclusive basis. For a description of the pin functions when used as the FIR pin, refer to **2.2.8 IrDA interface signals**. Table A-2 lists the functions of the pins that have been added to revision 2.0.

Table A-2. Debug Interface Signals (1/2)

Signal Name	I/O	Function
JTCK	I	JTAG Clock This is the N-Wire clock input.
JTMS	I	JTAG Mode Select This is the N-Wire serial transfer mode selection signal.
JTDI/RMODE#	I	JTAG Data Input / Reset Mode This is the RMODE# and JTDI alternate-function pin. When JTRST# is active, this pin functions as RMODE#, and when JTRST# is inactive, it functions as JTDI. When not connecting the debug tool to external, pull up this pin to high level. • RMODE#: Input When JTRST# is active, the reset mode pin is selected. The initial value for debug reset is determined based on the level of this signal. Debug reset consists of a reset applied to the processor. There are two types of debug resets, debug cold reset and debug soft reset. They are the same functions as cold reset input and soft reset input, respectively, from the target system. 0: Enables debug reset and resets the CPU core. 1: Disables debug reset and does not reset the CPU core. • JTDI: Input When JTRST# is inactive, N-Wire serial data input is selected.
JTRST#	I	JTAG Reset This is the N-Wire unit reset signal.
BKTGIO#	I/O	Break Trigger I/O • BKTGIO#: In case of input setting When JTRST# is inactive and BKTGIO# is set to input, the event trigger/break request input pin is selected. When break request input is enabled, setting BKTGIO# to low level breaks user program execution in the normal mode and forcibly places the processor in the debug mode. When BKTGIO# changes to low level in the debug mode, the break request is held until the processor returns to the normal mode. 0: Requests a break and forcibly places the processor in the debug mode. 1: Holds the current status of the processor. • BKTGIO#: In case of output setting When JTRST# is inactive and BKTGIO# is set to output, the event trigger/break output pin is selected. When the processor, while operating in the normal mode, detects an event that matches one of the conditions for a hardware breakpoint (instruction address breakpoint, data access breakpoint), a low level (1 pulse) is output as an event trigger from BKTGIO#, and detection of an event is notified to the external debug tool. Lastly, after the event trigger has been output, all the detected events are notified as one event trigger. When the processor goes into the debug mode, a low level is continuously output and none of the events that have not yet been notified are notified. 0: Detects hardware breakpoint The processor is in the debug mode. 1: The processor is in the normal mode.

Table A-2. Debug Interface Signals (2/2)

Signal Name	I/O	Function
HLDAK#/ NWIREEN	I/O	N-Wire Enable / Hold Acknowledge The function differs depending on the operation status. • After RTC reset (input) NWIREEN: This is the HALTimer shutdown function control signal and N-Wire use enable signal. 1: HALTimer shutdown function disable (HALTimer shutdown is not executed even if HALTimer is not cleared), N-Wire can be used. 0: HALTimer shutdown function enable (If HALTimer is not cleared, shutdown is executed after 4 seconds lapse following power-ON), N-Wire cannot be used. • When HLDEN bit of BCUCNTREG1 register = 1 HLDAK#: This signal enables use of the system bus to the external bus master and the DRAM bus. Refer to **2.2.1 Memory interface signals**.
IRDOUT/JTDO	O	IrDA Data Output / JTAG Data Output • IRDOUT: This is the IrDA serial data output signal. Refer to **2.2.8 IrDA interface signals**. • JTDO: Can be used as the N-Wire serial data output signal.

A.1.2 Pin Status

(1) Status of pins in defined state

Table A-3. Status of Pins in Defined State

Signal Name	During RTCRST	During Hibernate or HALTimer Shutdown	During RSTSW	During Suspend	During Bus Hold
BKTGIO#	**Note 1**	**Note 2**	**Note 1**	**Note 3**	**Note 1**
HLDAK#/NWIREEN	Hi-Z	0	1	Hold	0
IRDOUT#/JTDO	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
JTCK	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
JTDI/RMODE#	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
JTMS	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
JTRST#	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z

Notes 1. The value differs depending on debugger control. It is either 1 or high impedance.

2. The value differs depending on debugger control. It is either 0 or high impedance.

3. The value differs depending on debugger control. It is either the immediately preceding status held as is, or high impedance.

Remark 0: Low level; 1: High level; Hi-Z: High impedance; Hold: Immediately preceding full-speed mode status held as is

(2) Pin handling and I/O circuit types

Table A-4. Pin Connections and I/O Circuit Types

Pin Name	Pin Handling	Recommended Handling When Unused	Drive Performance	I/O Circuit Type
BKTGIO#	**Note**	Connect to V_{DD} via resistor	–	A
HLDAK#/NWIREEN	**Note**	Pull down	–	A
IRDOUT#/JTDO	Pull up	Leave open	–	A
JTCK	Pull up	Connected to V_{DD}	–	B
JTDI/RMODE#	Pull up	Connected to V_{DD}	–	A
JTMS	Pull up	Connected to V_{DD}	–	A
JTRST#	Leave open	Leave open	–	B

Note Refer to Figure A-1.

Remark For the I/O circuit type, refer to **2.3.3 Pin I/O circuit**.

A.2 Functions Added to/Deleted from Revision 2.0

A.2.1 Addition of N-Wire interface function

The N-Wire interface function is added to V_R4131 from revision 2.0.

(1) Features

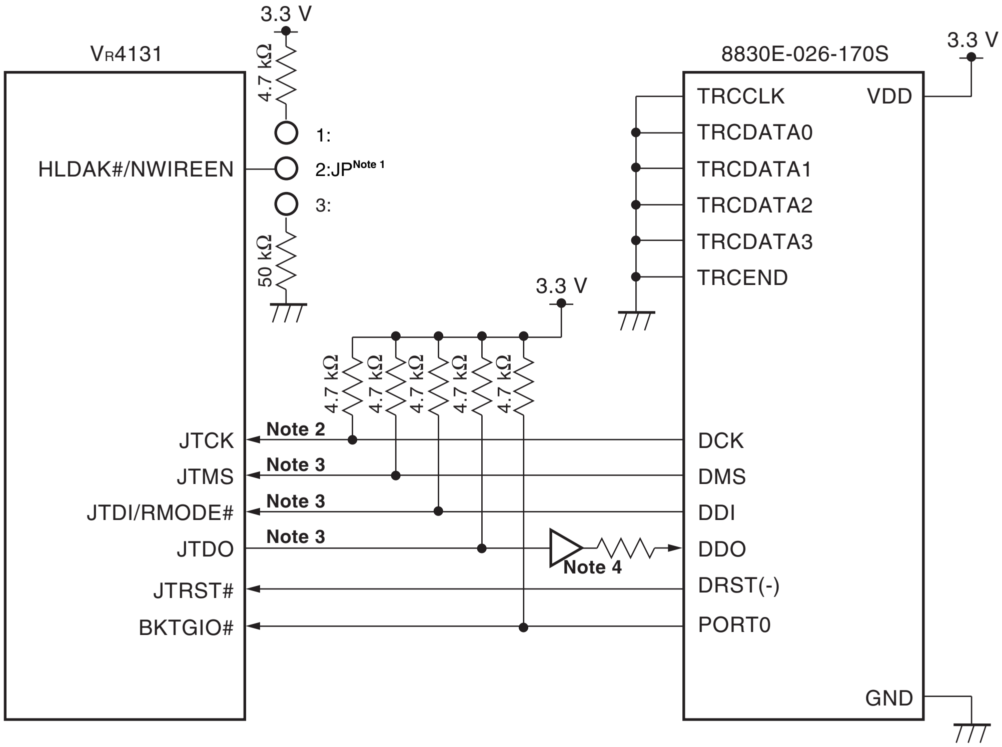
The features of the N-Wire interface are as follows.

- Interface based on JTAG standard
- Forced reset of the CPU core is possible (forced reset function)
- Forced interruption of user program execution is possible (forced break function)
- Interruption of user program execution at any address is possible.
- Execution of user program from any address is possible.
- User resources can be read/written during user program interruption (monitor function).
- Interrupt and reset masking is possible (pin mask function).
- Bus deadlock can be avoided (forced ready function).

(2) Connection circuit example

A sample connection circuit using the 8830E-026-170S connector made by KEL Corporation is shown below.

Figure A-1. Debug Tool Connection Circuit Example



Notes 1. Perform jumper settings as follows.

When using VR4131 revision 1.2 or when using N-Wire IE with VR4131 revision 2.0 and later: Close 1, 2

When not connecting N-Wire IE with VR4131 revision 2.0 and later: Close 2, 3

2. Keep the clock pattern length as short as possible and shield it with GND.

3. Keep the pattern length as short as possible. Make sure that the pattern length does not exceed 100 mm.

4. Use a 3-V buffer (SN74LVC541A made by Texas Instruments Incorporated or TC74LCX541F made by Toshiba Corporation are recommended.)

Remark VDD (B13) of the connector is used only to detect whether power is applied to the target board. However, depending on the tool that is used, this pin may also be used as the power supply for driving DCK or other signals, so directly connect it to the power supply of the target board.

A.2.2 Elimination of MMU disable mode

The MMU disable mode function has been eliminated from VR4131 revision 2.0 and later.

A.3 Functions Added to Revision 2.1

A.3.1 Addition of PCIU internal register

The PCIU internal register shown in Table A-5 has been added to VR4131 revision 2.1 and later.

Table A-5. Register Added to Revision 2.1

Physical Address	R/W	Symbol	Function
0x0F00 0C82	R/W	PCIBYTESWAPREG	Register for byte swap setting in big-endian mode

The details of this register are provided below.

(1) PCIBYTESWAPREG (0x0F00 0C82)

Bit Position	15	14	13	12	11	10	9	8
Bit name	BYTESWAP	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R/W	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit Position	7	6	5	4	3	2	1	0
Bit name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
After reset	0	0	0	0	0	0	0	0

Bit Position	Bit Name	Function
15	BYTESWAP	Specifies whether to perform little endian-big endian conRevision in byte units, regardless of the transfer size, during PCIU target operation in the big endian mode. 1: Unconditional byte swap 0: Swap according to transfer size
14:0	RFU	Reserved. Write 0 to these bits. When these bits are read, 0 is returned.

This register sets whether to perform little endian-big endian conRevision in byte units regardless of the transfer size in the CPU core big-endian mode. When the CPU core is in the little-endian mode or PCIU is operating as the master, the value of this register is ignored.

Next, the big endian-little endian conRevision during PCI bus operation in the big endian mode is described.

(a) When VR4131 = master (BYTESWAP = 1/0)

Number of SysAD Data Transfer Bytes	SysAD(2:0) (Internal Bus, Address)	SysAD(31:0) (Internal Bus, Address)	CBE(3:0)	AD(31:0) (Data)
1 byte	x00 x01 x10 x11	ABCD	1110 1101 1011 0111	DCBA
2 bytes	x00 x10	ABCD	1100 0011	CDAB
3 bytes	Undefined			
4 bytes	x00	ABCD	0000	ABCD
8 bytes	x00	ABCD EFGH	0000 (2) 0000 (1)	EFGH ABCD
16 bytes	000	ABCD EFGH IJKL MNOP	0000 (1) 0000 (2) 0000 (3) 0000 (4)	ABCD EFGH IJKL MNOP
32 bytes	000	ABCD EFGH IJKL MNOP QRST UVWX YZ01	0000 (1) 0000 (2) 0000 (3) 0000 (4) 0000 (5) 0000 (6) 0000 (7) 0000 (8)	ABCD EFGH IJKL MNOP QRST UVWX YZ01

Remarks 1. x: Don't care

2. The numbers in the CBE(3:0) column indicate the order of the data output to the AD bus.

If the memory on the PCI bus consists of cache, the PCI bus is accessed in big endian mode. In other words, SysAD(31:0) (Internal Bus, Data) and AD(31:0) (Data) in the table above are equal.

(b) When VR4131 = target (BYTESWAP = 1)

Number of SysAD Data Transfer Bytes	SysAD(2:0) (Internal Bus, Address)	SysAD(31:0) (Internal Bus, Address)	CBE(3:0)	AD(31:0) (Data)
1 byte	x00 x01 x10 x11	DCBA	1110 1101 1011 0111	ABCD
2 bytes	x00 x10	DCBA	1100 0011	ABCD
3 bytes	x00 x01	DCBA	1000 0001	ABCD
4 bytes	x00	DCBA	0000	ABCD
Burst	x00	DCBA	0000Note	ABCD

Note Processed as 4 bytes of continuous data.

Remark x: Don't care

(c) VR4131 = target (BYTESWAP = 0)

Number of SysAD Data Transfer Bytes	SysAD(2:0) (Internal Bus, Address)	SysAD(31:0) (Internal Bus, Data)	CBE(3:0)	AD(31:0) (Data)
1 byte	x00 x01 x10 x11	DCBA	1110 1101 1011 0111	ABCD
2 bytes	x00 x10	CDAB	1100 0011	ABCD
3 bytes	Undefined			
4 bytes	x00	ABCD	0000	ABCD
Burst	x00	ABCD	0000[Note]	ABCD

Note Processed as 4 bytes of continuous data.

Remark x: Don't care

APPENDIX B INDEX

[MEMO]

NEC

Facsimile Message

From:

____
Name

____
Company

____
Tel. FAX

____
Address

Although NEC has taken all possible steps to ensure that the documentation supplied to our customers is complete, bug free and up-to-date, we readily accept that errors may occur. Despite all the care and precautions we've taken, you may encounter problems in the documentation. Please complete this form whenever you'd like to report errors or suggest improvements to us.

Thank you for your kind support.

North America
NEC Electronics Inc.
Corporate Communications Dept.
Fax: +1-800-729-9288
+1-408-588-6130

Europe
NEC Electronics (Europe) GmbH
Market Communication Dept.
Fax: +49-211-6503-274

South America
NEC do Brasil S.A.
Fax: +55-11-6462-6829

Hong Kong, Philippines, Oceania
NEC Electronics Hong Kong Ltd.
Fax: +852-2886-9022/9044

Korea
NEC Electronics Hong Kong Ltd.
Seoul Branch
Fax: +82-2-528-4411

P.R. China
NEC Electronics Shanghai, Ltd.
Fax: +86-21-6841-1137

Taiwan
NEC Electronics Taiwan Ltd.
Fax: +886-2-2719-5951

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-250-3583

Japan
NEC Semiconductor Technical Hotline
Fax: +81- 44-435-9608

I would like to report the following error/make the following suggestion:

Document title: _______________________________

Document number: ____________________ Page number: __________

____

____

____

If possible, please fax the referenced page or drawing.

Document Rating	Excellent	Good	Acceptable	Poor
Clarity	❏	❏	❏	❏
Technical Accuracy	❏	❏	❏	❏
Organization	❏	❏	❏	❏