NEC

User's Manual

VR4181™

64-/32-Bit Microprocessor

Hardware

μPD30181

Document No. U14272EJ3V0UM00 (3rd edition)
Date Published November 2002 NS CP(K)

Printed in Japan

[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 in this document is current as of January, 2002. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales 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 Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics 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 a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".
  The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product 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 and medical equipment for life support, etc.

The quality grade of NEC Electronics products is "Standard" unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact an NEC Electronics sales representative in advance to determine NEC Electronics' willingness to support a given application.

(Note)
(1) "NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.
(2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

M8E 02. 11-1

Regional Information

Some information contained in this document may vary from country to country. Before using any NEC Electronics product in your application, pIease contact the NEC Electronics 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 America, Inc. (U.S.)
Santa Clara, California
Tel: 408-588-6000
800-366-9782
Fax: 408-588-6130
800-729-9288

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

- **Sucursal en España**
  Madrid, Spain
  Tel: 091-504 27 87
  Fax: 091-504 28 60

- **Succursale Française**
  Vélizy-Villacoublay, France
  Tel: 01-30-67 58 00
  Fax: 01-30-67 58 99

- **Filiale Italiana**
  Milano, Italy
  Tel: 02-66 75 41
  Fax: 02-66 75 42 99

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

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

- **United Kingdom Branch**
  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: 6253-8311
Fax: 6250-3583

J02.11

Major Revisions in This Edition (1/5)

Page	Description
Throughout this manual	Separation of the following parts of the previous (the 2nd) edition CHAPTER 3 MIPS III INSTRUCTION SET SUMMARY, CHAPTER 4 MIPS16 INSTRUCTION SET, CHAPTER 5 VR4181 PIPELINE, CHAPTER 6 MEMORY MANAGEMENT SYSTEM (first half), CHAPTER 7 EXCEPTION PROCESSING (second half), CHAPTER 9 CACHE MEMORY, CHAPTER 10 CPU CORE INTERRUPTS, CHAPTER 27 MIPS III INSTRUCTION SET DETAILS, CHAPTER 28 MIPS16 INSTRUCTION SET FORMAT
p. 30	Deletion of modem block in **Figure 1-1. Internal Block Diagram**
p. 34	Modification of description in **1.3.16 LCD interface**
p. 35	Modification of Remark in **1.3.17 Wake-up events**
pp. 38 to 42	Addition of **1.4.2 CPU instruction set overview** and **1.4.3 Data formats and addressing**
p. 43	Modification of description and deletion of figure in **1.4.4 CP0 registers**
pp. 45, 46	Addition of **1.4.9 Power modes** and **1.4.10 Code compatibility**
p. 47	Addition of descriptions in **1.5 Clock Interface**
pp. 48, 49	Addition of **Figure 1-8. External Circuits of Clock Oscillator** and **Figure 1-9. Incorrect Connection Circuits of Resonator**
p. 52	Modification of Note in **2.2.1 System bus interface signals**
p. 53	Modification of descriptions for SYSDIR and SYSEN# and addition of description in Note in **2.2.1 System bus interface signals**
p. 58	Addition of description for IRDIN/RxD2 in **2.2.10 IrDA interface signals**
p. 60	Addition and modification in **2.3 Pin Status in Specific Status**
pp. 63 to 66	Addition of **2.4 Recommended Connection of Unused Pins and I/O Circuit Types** and **2.5 Pin I/O Circuits**
pp. 67 to 90	Addition of **CHAPTER 3 CP0 REGISTERS**
p. 95	Modification of **Table 4-6. DRAM Address Map**
p. 97	Modification of description in **5.1.1 RTC reset**
pp. 97 to 101, 104, 105	Addition of description in Note in **Figure 5-1** through **Figure 5-5**, **Figure 5-8**, and **Figure 5-9**
p. 98	Modification in **Figure 5-2. RSTSW Reset**
p. 101	Modification of description in **5.1.5 HALTimer shutdown**
p. 104	Addition of description in **5.3.1 Cold Reset**
p. 105	Modification of description in **5.3.2 Soft Reset**
pp. 106, 107	Addition of **5.4 Notes on Initialization**
p. 108	Modification in **Figure 6-1. VR4181 Internal Bus Structure**
p. 109	Modification of description in **6.1.2 (3) LCD module (LCD Control Unit)**
p. 111	Modification of description for bit 4 and addition of Caution and Remark in **6.2.1 BCUCNTREG1 (0x0A00 0000)**
p. 113	Modification of descriptions for bits 14 to 12, bits 3 to 0, and Remark in **6.2.3 BCUSPEEDREG (0x0A00 000C)**
p. 114	Modification of **Figure 6-2. ROM Read Cycle and Access Parameters**
p. 117	Deletion of description for Div4 mode and addition of description in Remark in **6.2.6 (2) Peripheral clock (TClock)**

Major Revisions in This Edition (2/5)

Page	Description
p. 119	Modification of description in **6.3.2 Connection to external ROM (x 16) devices**
p. 122	Modification of Remark in **6.3.3 (4) 64 Mbit PageROM**
p. 123	Modification of figure in **6.3.3 (5) 32 Mbit flash memory (when using Intel™ DD28F032)**
pp. 125 to 128, 130	Modification of **Figure 6-3** through **Figure 6-8**
p. 129	Addition of description in **Table 6-2. VR4181 EDO DRAM Capacity**
p. 134	Addition of Caution and modification in Remark in **6.5.2 MEMCFG_REG (0x0A00 0304)**
p. 135	Modification of description for bits 6 to 4 in **6.5.3 MODE_REG (0x0A00 0308)**
p. 136	Modification of Note in **6.5.4 SDTIMINGREG (0x0A00 030C)**
p. 137	Addition of description in **6.6 ISA Bridge**
p. 138	Addition of description in **6.7.1 ISABRGCTL (0x0B00 02C0)**
p. 140	Modification of description for bits 10 and 9 and addition of description in **6.7.3 XISACTL (0x0B00 02C4)**
p. 149	Modification of description for bits 3 and 2 in **7.2.6 AIUDMAMSKREG (0x0A00 0046)**
p. 150	Modification of values at reset in **7.2.7 MICRCLENREG (0x0A00 0658)** and **7.2.8 SPKRCLENREG (0x0A00 065A)**
p. 151	Addition of description for bit 8 in **7.2.9 MICDMACFGREG (0x0A00 065E)**
p. 152	Addition of description for bit 0 in **7.2.10 SPKDMACFGREG (0x0A00 0660)**
p. 153	Addition of description for bits 5 and 4 in **7.2.11 DMAITRQREG (0x0A00 0662)**
p. 156	Modification of description and addition of Caution in **8.1 Overview**
p. 157	Addition of Caution in **Figure 8-1. SCK and SI/SO Relationship**
pp. 157, 158	Addition and modification of descriptions in **8.2.2 SCK phase and CSI transfer timing**
p. 159	Modification of description in **8.2.3 (1) Burst mode**
pp. 161, 162	Addition of Remarks and description in **8.3.1 CSIMODE (0x0B00 0900)**
p. 171	Addition of description in **9.1 Overview**
p. 173	Modification of description in **Table 9-1. ICU Registers**
p. 184	Modification of address and description for bits 2 and 1, and addition of description in **9.2.9 KIUINTREG (0x0B00 0086)**
p. 186	Modification of R/W and addition of description in **9.2.11 MAIUINTREG (0x0B00 0090)**
p. 189	Modification of description in **Figure 10-1. Transition of VR4181 Power Mode**
pp. 190, 191	Addition and modification of descriptions in **10.2.1 Power mode and state transition**
p. 191	Modification of description in **Table 10-2. Operations During Reset**
p. 192	Modification of location of **10.3.3 Deadman's Switch reset**
p. 192	Modification of **Figure 10-2. EDO DRAM Signals on RSTSW Reset (SDRAM Bit = 0)**
p. 192	Modification of description in **10.3.4 (2) Preserving SDRAM data**
p. 193	Modification of description in **Table 10-3. Operations During Shutdown**
p. 194	Modification of description of Caution in **10.5 Power-on Control**
p. 196	Modification of signal name in **10.5.2 Activation via CompactFlash interrupt request**
p. 197	Modification of description in **10.5.3 Activation via GPIO activation interrupt request**

Major Revisions in This Edition (3/5)

Page	Description
p. 198	Modification of description of Cautions in **10.5.4 Activation via DCD interrupt request**
pp. 201 to 204	Modification of descriptions in **10.6.1** through **10.6.4**
pp. 205 to 207	Addition of **10.6.5** through **10.6.8**
p. 209	Modification of description for bit 6 in **10.7.1 PMUINTREG (0x0B00 00A0)**
p. 211	Modification of value at reset for bit 7 in **10.7.2 PMUCNTREG (0x0B00 00A2)**
p. 214	Modification of description for bit 2 to 0 in **10.7.4 PMUDIVREG (0x0B00 00AC)**
p. 215	Modification of description for bit 4 in **10.7.5 DRAMHIBCTL (0x0B00 00B2)**
p. 220	Modification of value at reset for bit 15 in **11.2.2 (3) ECMPHREG (0x0B00 00CC)**
p. 238	Modification and addition of descriptions in **13.1.3 General-purpose registers**
p. 242	Modification of description in **13.2.5 16-bit bus cycles**
p. 254	Modification of R/W for bits 15 to 8 in **13.3.5 GPDATHREG (0x0B00 0308)**
p. 267	Modification of description in **13.3.15 KEYEN (0x0B00 031C)**
p. 270	Modification of description for bit 15 in **13.3.19 PCS1STRA (0x0B00 0326)**
p. 273	Modification of description for bit 7 in **13.3.23 LCDGPMODE (0x0B00 032E)**
p. 275	Addition of Caution in **14.1 General**
p. 283	Modification of location of Note in **Table 14-3. PIUCNTREG Bit Manipulation and States**
p. 286	Modification of description for bits 5 to 0 in **14.3.4 PIUSTBLREG (0x0B00 0128)**
p. 289	Addition of description in **14.3.6 PIUASCNREG (0x0B00 0130)**
p. 290	Modification of description in **Table 14-4. PIUASCNREG Bit Manipulation and States**
p. 291	Addition of description in **14.3.7 PIUAMSKREG (0x0B00 0132)**
p. 292	Modification of values at reset for bits 2 to 0 in **14.3.8 PIUCIVLREG (0x0B00 013E)**
p. 295	Modification of description in **Table 14-7. Mask Clear During Scan Sequencer Operation**
p. 298	Addition of Note in **Figure 14-6. Touch/Release Detection Timing**
p. 298	Modification of **Figure 14-7. A/D Port Scan Timing**
p. 301	Modification of description and addition of Caution in **15.1 General**
pp. 303, 304	Modification of addresses in **15.2.1 SDMADATREG (0x0B00 0160)** and **15.2.2 MDMADATREG (0x0B00 0162)**
p. 308	Modification of values at reset for bits 11, 10 and 5 and addition of Caution in **15.2.6 SCNVC_END (0x0B00 016E)**
p. 314	Modification of values at reset for bits 11, 10 and 5 and addition of Caution in **15.2.12 MCNVC_END (0x0B00 017E)**
pp. 315, 316	Addition of descriptions in **15.3.1 Output (speaker)** and **15.3.2 Input (microphone)**
p. 320	Modification of description in **16.2.6 Interrupts and status reporting**
p. 321	Modification of description in **Table 16-3. KIU Interrupt Registers**
p. 324	Modification of description for bits 1 and 0 in **16.3.3 KIUSCANS (0x0B00 0192)**
p. 325	Modification of descriptions for bits 14 to 10 and bits 4 to 0 in **16.3.4 KIUWKS (0x0B00 0194)**
p. 327	Modification and addition of descriptions for bits 2 to 0 in **16.3.6 KIUINT (0x0B00 0198)**
p. 328	Modification of signal name in **17.1 General**

Major Revisions in This Edition (4/5)

Page	Description
p. 333	Modification of signal names in **Figure 17-1. CompactFlash Interrupt Logic**
p. 333	Modification of description for bit 0 in **17.3.3 CFG_REG_1 (0x0B00 08FE)**
p. 336	Addition of Caution for bit 4 in **17.4.3 PWRRSETDRV (Index: 0x02)**
p. 337	Modification of description for bit 7 in **17.4.4 ITGENCREG (Index: 0x03)**
p. 338	Modification of description and addition of Caution for bit 0 in **17.4.5 CDSTCHGREG (Index: 0x04)**
p. 341	Modification of descriptions for bits 7, 4, 3, and 0 in **17.4.8 IOCTRL_REG (Index: 0x07)**
p. 344	Modification of description for bit 6 in **17.4.14 MEMWIDn_REG (Index: 0x11, 0x19, 0x21, 0x29, 0x31)**
p. 345	Modification of description for bits 7 and 6 and addition of description in **17.4.16 MEMSELn_REG (Index: 0x13, 0x1B, 0x23, 0x2B, 0x33)**
p. 346	Addition of description in **17.4.17 MEMOFFLnREG (Index: 0x14, 0x1C, 0x24, 0x2C, 0x34)**
p. 346	Modification of Remark for bits 5 to 0 in **17.4.18 MEMOFFHnREG (Index: 0x15, 0x1D, 0x25, 0x2D, 0x35)**
p. 348	Modification of description for bit 2 in **17.4.20 GLOCTRLREG (Index: 0x1E)**
p. 349	Modification of description for bits 1 and 0 and addition of description in **17.4.22 VOLTSELREG (Index: 0x2F)**
pp. 350, 351	Addition of **17.5 Memory Mapping of CompactFlash Card**
p. 352	Addition of **17.6 Controlling Bus When CompactFlash Card Is Used**
p. 356	Addition of function for bit 2 in **18.2.3 LEDCNTREG (0x0B00 0248)**
p. 357	Modification of description in **18.2.4 LEDASTCREG (0x0B00 024A)**
p. 359	Modification of figure in **18.3 Operation Flow**
p. 360	Addition of Caution in **19.1 General**
p. 361	Modification of description in **Table 19-1. SIU1 Registers**
pp. 362, 364, 376	Modification of values at reset in **19.3.1** through **19.3.3**, **19.3.5**, and **19.3.12**
p. 365	Addition of description in **Table 19-2. Correspondence between Baud Rates and Divisors**
p. 368	Modification of descriptions for bits 2 to 0 in **19.3.7 SIUFC_1 (0x0C00 0012: Write)**
p. 373	Modification of R/W and addition of description in **19.3.10 SIULS_1 (0x0C00 0015)**
p. 375	Modification of descriptions for bits 7 to 4 in **19.3.11 SIUMS_1 (0x0C00 0016)**
p. 377	Modification of R/W for bit 1 in **19.3.14 SIUACTMSK_1 (0x0C00 001C)**
p. 379	Addition of description and Caution in **20.1 General**
p. 380	Modification of description in **Table 20-1. SIU2 Registers**
pp. 381, 383, 395	Modification of values at reset in **20.3.1** through **20.3.3**, **20.3.5**, and **20.3.12**
p. 384	Addition of description in **Table 20-2. Correspondence between Baud Rates and Divisors**
p. 387	Modification of descriptions for bits 2 to 0 in **20.3.7 SIUFC_2 (0x0C00 0002: Write)**
p. 392	Modification of R/W and addition of description in **20.3.10 SIULS_2 (0x0C00 0005)**
p. 394	Modification of descriptions for bits 7 to 4 in **20.3.11 SIUMS_2 (0x0C00 0006)**
p. 395	Addition of description in **20.3.13 SIUIRSEL_2 (0x0C00 008)**
p. 397	Modification of R/W for bit 1 in **20.3.16 SIUACTMSK_2 (0x0C00 000C)**

Major Revisions in This Edition (5/5)

Page	Description
p. 399	Modification of description in **21.1.1 LCD interface**
p. 401	Modification of bus width in **Figure 21-1. LCD Controller Block Diagram**
p. 406	Modification of description in **21.3.4 Frame buffer memory and FIFO**
p. 420	Addition of Remark in **21.4.11 LCDCFGREG0 (0x0A00 0414)**
p. 428	Addition of Remark in **21.4.22 CPINDCTREG (0x0A00 047E)**
p. 429	Addition of Caution in **21.4.23 CPALDATREG (0x0A00 0480)**
p. 433	Addition of Caution in **Table 23-1. Coprocessor 0 Hazards**
pp. 436 to 438	Addition of **APPENDIX A RESTRICTIONS ON V_R4181**
pp. 439 to 444	Addition of **APPENDIX B INDEX**

The mark ★ shows major revised points.

PREFACE

Readers

This manual targets users who intend to understand the functions of the VR4181 and to design application systems using this microprocessor.

Purpose

This manual introduces the hardware functions of the VR4181 to users, following the organization described below.

Organization

Two manuals are available for the VR4181: Hardware User's Manual (this manual) and Architecture User's Manual common to the VR4100 Series™.

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

How to read this manual

It is assumed that the reader of this manual has general knowledge in the fields of electric engineering, logic circuits, microcomputers, and SDRAMs.

To learn about the overall functions of the VR4181,

→ Read this manual in sequential order.

To learn about instruction sets,

→ Read **VR4100 Series Architecture User's Manual** that is separately available.

To learn about electrical specifications,

→ Refer to **Data Sheet** that is separately available.

Conventions

Data significance:	Higher on left and lower on right
Active low:	XXX# (trailing # after pin and signal names)
Note:	Description of item marked with **Note** in the text
Caution:	Information requiring particular attention
Remark:	Supplementary information
Numeric representation:	binary/decimal ... XXXX
	hexadecimal ... 0xXXXX

Prefixes representing an exponent of 2 (for address space or memory capacity):

K (kilo) …	$2^{10} = 1024$
M (mega) …	$2^{20} = 1024^2$
G (giga) …	$2^{30} = 1024^3$
T (tera) …	$2^{40} = 1024^4$
P (peta) …	$2^{50} = 1024^5$
E (exa) …	$2^{60} = 1024^6$

Related Documents

When using this manual, also refer to the following documents.

Document name	Document number
VR4181 Hardware User's Manual	This manual
μPD30181 (VR4181) Data Sheet	U14273E
VR4100 Series Architecture User's Manual	U15509E
VR Series ™ Programming Guide Application Note	U10710E

The related documents indicated here may include preliminary version. However, preliminary versions are not marked as such.

CONTENTS

LIST OF FIGURES (1/3)

LIST OF FIGURES (2/3)

LIST OF FIGURES (3/3)

LIST OF TABLES (1/2)

LIST OF TABLES (2/2)

CHAPTER 1 INTRODUCTION

This chapter describes the outline of the VR4181 (μPD30181), which is a 64-/32-bit microprocessor.

1.1 Features

The VR4181, which is a high-performance 64-/32-bit microprocessor employing the RISC (reduced instruction set computer) architecture developed by MIPS™, is one of the VR-Series microprocessor products manufactured by NEC Electronics.

The VR4181 contains the VR4110™ CPU core of ultra-low-power consumption with cache memory, high-speed product-sum operation unit, and memory management unit. It also has interface units for peripheral circuits such as LCD controller, CompactFlash controller, DMA controller, keyboard interface, serial interface, IrDA interface, touch panel interface, real-time clock, A/D converter and D/A converter required for the battery-driven portable information equipment. The features of the VR4181 are described below.

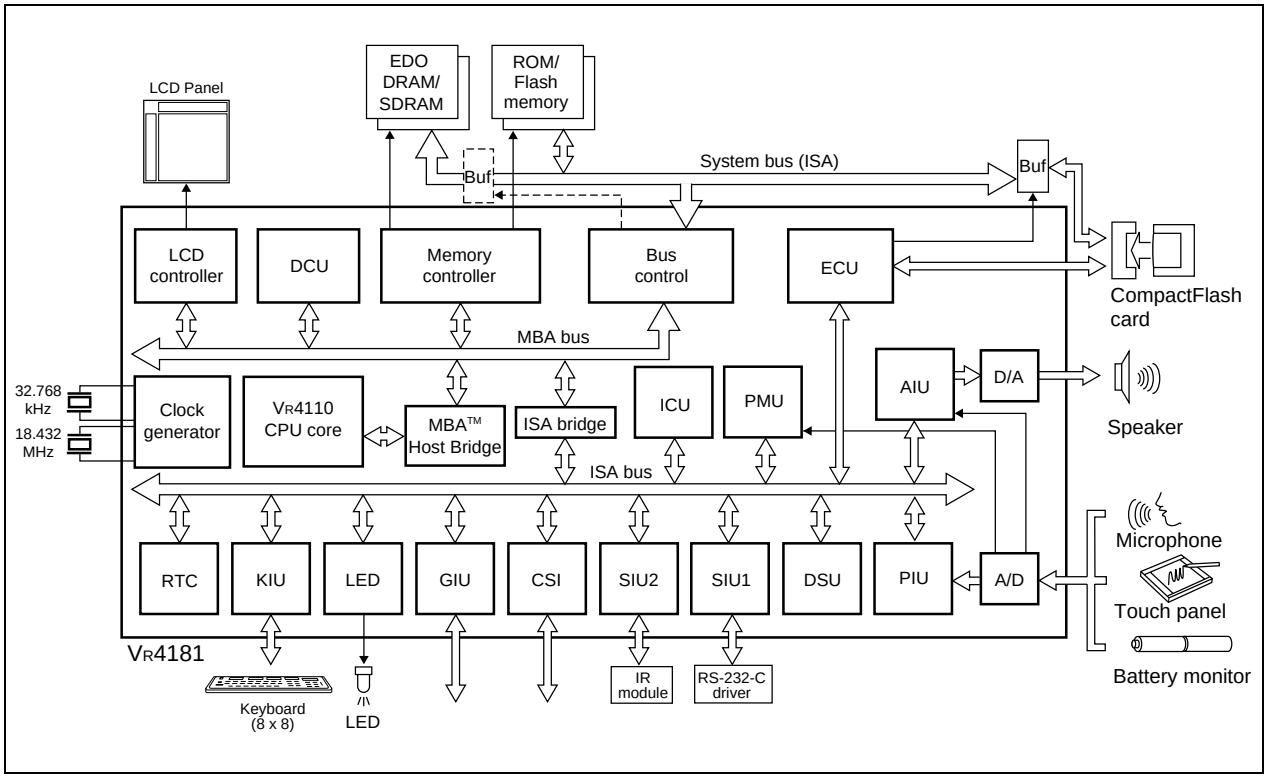
- Employs 0.25 μm process
- 64-bit RISC VR4110 CPU core with pipeline clock up to 66 MHz (operation in 32-bit mode is available)
- Optimized 5-stage pipeline
- On-chip instruction and data caches with 4 KB each in size
- Write-back cache for reducing store operation that use the system bus
- 32-bit physical address space and 40-bit virtual address space, and 32 double-entry TLB
- Instruction set: MIPS III (with the FPU, LL and SC instructions left out) and MIPS16
- Supports MADD16 and DMADD16 instructions for executing a multiply-and-accumulate operation of 16-bit data x 16-bit data + 64-bit data within one clock cycle
- Effective power management features, which include four operating modes, Fullspeed, Standby, Suspend and Hibernate mode
- On-chip PLL and clock generator
- DRAM interface supporting 16-bit width SDRAM and EDO DRAM
- Ordinary ROM/PageROM/flash memory interface
- UMA based LCD controller
- 4-channel DMA controller
- RTC unit including 3-channel timers and counters
- Two UART-compatible serial interfaces and one clocked serial interface
- IrDA (SIR) interface
- Keyboard scan interface supporting 8 x 8 key matrix
- X-Y auto-scan touch panel interface
- CompactFlash interface compatible with ExCA
- A/D and D/A converters
- Includes ISA-subset bus
- Supply voltage: 2.5 V for CPU core, 3.3 V for I/O
- Package: 160-pin LQFP

1.2 Ordering Information

Part number	Package	Maximum internal operating frequency
μPD30181GM-66-8ED	160-pin plastic LQFP (fine pitch) (24 × 24)	66 MHz

1.3 VR4181 Key Features

Figure 1-1. Internal Block Diagram



1.3.1 CPU core

The VR4181 integrates an NEC Electronics' VR4110 CPU core supporting both the MIPS III and MIPS16 instruction sets.

The VR4181 supports the following pipeline clock (PClock) and internal bus clock (TClock) frequencies. The PClock is set by attaching pull-up or pull-down resistors to the CLKSEL(2:0) pins. The frequency of the TClock, which is used in MBA bus, is set by PMUDIVREG register in Power Management Unit.

★ **Table 1-1. Supported PClock and TClock Frequencies**

PClock frequency	TClock frequency
65.4 MHz	65.4/32.7/21.8 MHz
62.0 MHz	62.0/31.0/20.7 MHz
49.1 MHz	49.1/24.6 MHz

The VR4110 core of the VR4181 includes 4 KB of instruction cache and 4 KB of data cache.

The VR4110 core also supports the following power management modes:

- Fullspeed
- Standby
- Suspend [Note]
- Hibernate

Note Suspend mode is supported only when the internal LCD controller has been disabled or the LCD panel has been powered off.

1.3.2 Bus interface

The VR4181 incorporates single bus architecture. All external memory and I/O devices are connected to the same 22-bit address bus and 16-bit data bus. These external address and data bus are together called the system bus.

When the external bus operates at a very high speed, the DRAM data bus must be isolated from other low speed devices such as ROM array. The VR4181 provides two pins, SYSEN# and SYSDIR, to control the data buffers for this isolation.

The VR4181 supports the following types of devices connected to the system bus.

Table 1-2. Devices Supported by System Bus

Device	Data width
ROM, flash memory	16 bits only
DRAM	16 bits only
CompactFlash	8 or 16 bits
External I/O	8 or 16 bits
External memory	8 or 16 bits

Six of the external bus interface signals, IORD#, IOWR#, IORDY, IOCS16#, MEMCS16# and RESET#, can be individually defined as general-purpose I/O pins or LCD interface pin if they are not needed by external system components.

1.3.3 Memory interface

The VR4181 provides control for both ROM/flash memory and DRAM. Up to four 16-bit ROM/flash memory banks may be supported utilizing either 32-Mbit or 64-Mbit single cycle or page mode devices. Bank mixing is not supported for ROM/flash memory. When a system implements less than the maximum 4 banks of ROM/flash memory, unused ROM chip select pins can be defined as general-purpose I/O pins.

The VR4181 also supports up to 2 banks of 1M x 16 or 4M x 16 EDO-type DRAM or SDRAM at bus frequencies of up to 66 MHz. When both banks are EDO-type DRAM, bank mixing is supported.

1.3.4 DMA controller (DCU)

The VR4181 provides a 4-channel DMA controller to support internal DMA transfers. The 4 channels are allocated as follows:

- Channel 1 - Audio input
- Channel 2 - Audio output
- Channel 3, 4 - Reserved

1.3.5 Interrupt controller (ICU)

The VR4181 provides an interrupt controller which combines all interrupt request sources into one of the VR4110 core interrupt inputs - NMI and Int(2:0). The interrupt controller also provides interrupt request status reporting.

1.3.6 Real-time clock

The VR4181 includes a real-time clock (RTC), which allows time keeping based on the 32.768 kHz clock as a source. The RTC operates as long as the VR4181 remains powered.

1.3.7 Audio output (D/A converter)

The VR4181 provides a 1-channel 10-bit D/A converter for generating audio output.

1.3.8 Touch panel interface and audio input (A/D converter)

The VR4181 provides an 8-channel 10-bit A/D converter for interfacing to a touch panel, an external microphone, and other types of analog input.

1.3.9 CompactFlash interface (ECU)

The VR4181 provides an ExCA-compatible bus controller supporting a single CompactFlash slot. This interface is shared with the keyboard interface logic and must be disabled when an 8 x 8 key matrix is connected to the VR4181.

1.3.10 Serial interface channel 1 (SIU1)

The VR4181 provides a 16550 UART for implementing an RS-232-C type serial interface. When the serial interface is not needed, each of the 7 serial interface pins can be individually redefined as general-purpose I/O pins.

1.3.11 Serial interface channel 2 (SIU2)

The serial interface channel 2 is also based on a 16550 UART but only reserves 2 pins for the interface. The serial interface channel 2 can be configured in one of the following modes:

- Simple 2-wire serial interface using TxD2 and RxD2
- SIR-type IrDA interface using IRDIN and IRDOUT
- Full RS-232-C compatible interface using TxD2, RxD2 and 5 GPIO pins

1.3.12 Clocked serial interface (CSI)

The VR4181 provides a clocked serial interface (CSI) which has an option to be configured as general-purpose I/O pins. This interface supports slave mode operation only. The clocked serial interface requires allocation of 4 signals; SI, SO, SCK, and FRM. The clock source for this interface is input on the pin assigned to SCK.

1.3.13 Keyboard interface (KIU)

The VR4181 provides support for an 8 x 8 key matrix. This keyboard interface can only be supported when the CompactFlash interface is disabled and reconfigured to provide the SCANIN(7:0) inputs and the SCANOUT(7:0) outputs.

1.3.14 General-purpose I/O

The VR4181 provides total 32 bits of general-purpose I/O. Sixteen of these, GPIO(31:16), are available through pins allocated to other functions as shown in the following table. The DCD1#/GPIO29 is the only one of the 16 pins that can cause the system's waking up from a low power mode if enabled by software. The other pins have no functions other than those listed below.

The remaining 16 bits of general-purpose I/O, GPIO(15:0), are allocated to pins by default. Each of these pins can be configured to support a particular interface such as CSI, secondary serial interface (RS-232-C), programmable chip selects, or color LCD control. Otherwise, each of these pins can be also defined as one of the following:

- General-purpose input
- General-purpose output
- Interrupt request input
- Wake-up input

Table 1-3. GPIO(31:0) Pin Functions

Pin designation	Alternate function	Pin designation	Alternate function
GPIO0	SI	GPIO16	IORD#
GPIO1	SO	GPIO17	IOWR#
GPIO2	SCK	GPIO18	IORDY
GPIO3	PCS0#	GPIO19	IOCS16#
GPIO4	–	GPIO20 Note	M/UBE#
GPIO5	DCD2#	GPIO21	RESET#
GPIO6	RTS2#	GPIO22	ROMCS0#
GPIO7	DTR2#	GPIO23	ROMCS1#
GPIO8	DSR2#	GPIO24	ROMCS2#
GPIO9	CTS2#	GPIO25	RxD1
GPIO10	FRM/SYSCLK	GPIO26	TxD1
GPIO11	PCS1#	GPIO27	RTS1#
GPIO12	FPD4	GPIO28	CTS1#
GPIO13	FPD5	GPIO29	DCD1#
GPIO14	FPD6/CD1#	GPIO30	DTR1#
GPIO15	FPD7/CD2#	GPIO31	DSR1#

Note This signal supports input only.

1.3.15 Programmable chip selects

The VR4181 provides support for 2 programmable chip selects (PCS) which are also available as general-purpose I/O pins. Each PCS can decode either I/O or memory accesses and can optionally be qualified to read, write, or both read and write.

1.3.16 LCD interface

The LCD controller of the VR4181 is Unified Memory Architecture (UMA) based in which the frame buffer is part of system DRAM. The LCD controller supports monochrome STN LCD panels having 4-bit data bus interfaces and color STN LCD panels having 8-bit data bus interface. When interfacing to a color LCD panel, general-purpose I/O pins must be allocated to provide the upper nibble of the 8-bit LCD data bus.

In monochrome mode, the LCD controller supports 1-bpp mode (mono), 2-bpp mode (4 gray levels) and 4-bpp mode (16 gray levels). In color mode, it supports 4-bpp mode (16 colors) and 8-bpp mode (256 colors).

The LCD controller includes a 256-entry x 18-bit color pallet. In 8-bpp color modes, the pallet is used to select 256 colors out of possible 262,144 colors.

The LCD controller supports LCD panels of up to 320 x 320 pixels. Typical LCD panel horizontal/vertical resolutions are as follows.

Table 1-4. LCD Panel Resolutions (in Pixels, TYP.)

Horizontal resolution	Vertical resolution
320	320
320	240
320	160
240	320
240	240
240	160
160	320
160	240
160	160

The LCD controller also provides power-on and power-down sequence control for the LCD panel via the VPLCD and VPBIAS pins. Power sequencing is provided to prevent latch-up damage to the panel.

The LCD controller can be disabled to allow connection of an external LCDC with integrated frame buffer RAM such as NEC Electronics' μ PD16661. When the internal LCD controller is disabled, the SHCLK, LOCLK, VPLCD, and VPBIAS pins are redefined as follows:

Table 1-5. Functions of LCD Interface Pins when LCD Controller Is Disabled

Redefined function	Default function
LCDCS#	SHCLK
MEMCS16#	LOCLK
VPGPIO1	VPLCD
VPGPIO0	VPBIAS

1.3.17 Wake-up events

The VR4181 supports 4 power management modes: Fullspeed, Standby, Suspend, and Hibernate. Of these modes, Hibernate is the lowest power mode and results in the powering off of all system components including the 2.5 V logic in the VR4181. The VR4181 3.3 V logic, which includes RTC, PMU, and non-volatile registers, remain powered during the Hibernate mode, as does the system DRAM. Software can configure the VR4181 waking up from the Hibernate mode and returning to Fullspeed mode due to any one of the following events:

- Activation of the DCD1# pin
- Activation of the POWER pin
- RTC alarm
- Activation of one of the GPIO(15:0) pins
- ★ • Activation of the CF_BUSY# pin (CompactFlash interrupt request (IREQ))

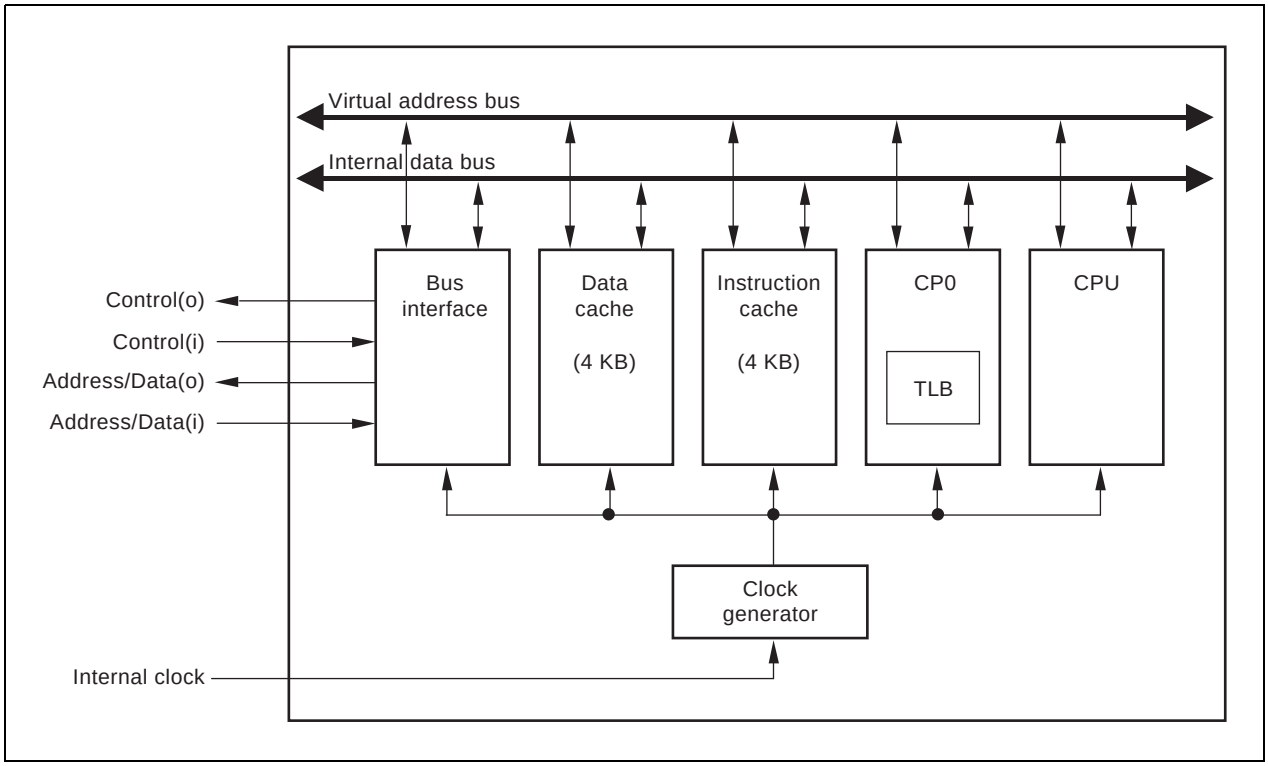
★ **Remark** Different from the VR4111™ or the VR4121™, the VR4181 will wake up after RTC reset without these wake-up events.

1.4 VR4110 CPU Core

Figure 1-2 shows the internal block diagram of the VR4110 CPU core.

In addition to the conventional high-performance integer operation units, this CPU core has the full-associative format translation lookaside buffer (TLB), which has 32 entries that provide mapping to 2-page pairs (odd and even) for one entry. Moreover, it also includes instruction cache, data cache, and bus interface.

Figure 1-2. VR4110 CPU Core Internal Block Diagram



(1) CPU

The CPU has hardware resources to process an integer instruction. They are the 64-bit register file, 64-bit integer data path, and multiply-and-accumulate operation unit.

(2) Coprocessor 0 (CP0)

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

(3) Instruction cache

The instruction cache employs direct mapping, virtual index, and physical tag. Its capacity is 4 KB.

(4) Data cache

The data cache employs direct mapping, virtual index, physical tag, and writeback. Its capacity is 4 KB.

(5) CPU bus interface

The CPU bus interface controls data transmission/reception between the VR4110 core and the MBA Host Bridge. This interface consists of two 32-bit multiplexed address/data buses (one is for input, and another is for output), clock signal, and control signals such as interrupt requests.

(6) Clock generator

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

- 32.768 kHz clock for RTC unit
  Crystal resonator input oscillated via an internal oscillator and supplied to the RTC unit.

- 18.432 MHz clock for serial interface and the VR4181's reference operating clock
  Crystal resonator input oscillated via an internal oscillator, and then multiplied by phase-locked loop (PLL) to generate a pipeline clock (PClock). The internal bus clock (TClock) is generated from PClock and supplied to peripheral units.

1.4.1 CPU registers

The VR4110 core has thirty-two 64-bit general-purpose registers (GPRs).

In addition, the processor provides the following special registers:

- 64-bit Program Counter (PC)
- 64-bit HI register, containing the integer multiply and divide upper doubleword result
- 64-bit LO register, containing the integer multiply and divide lower doubleword result

Two of the general-purpose registers have assigned functions as follows:

- r0 is hardwired to a value of zero, 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 when a zero value is needed.
- r31 is the link register used by link instructions, such as JAL (Jump and Link) instruction. 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, to process exceptions and to manage addresses.

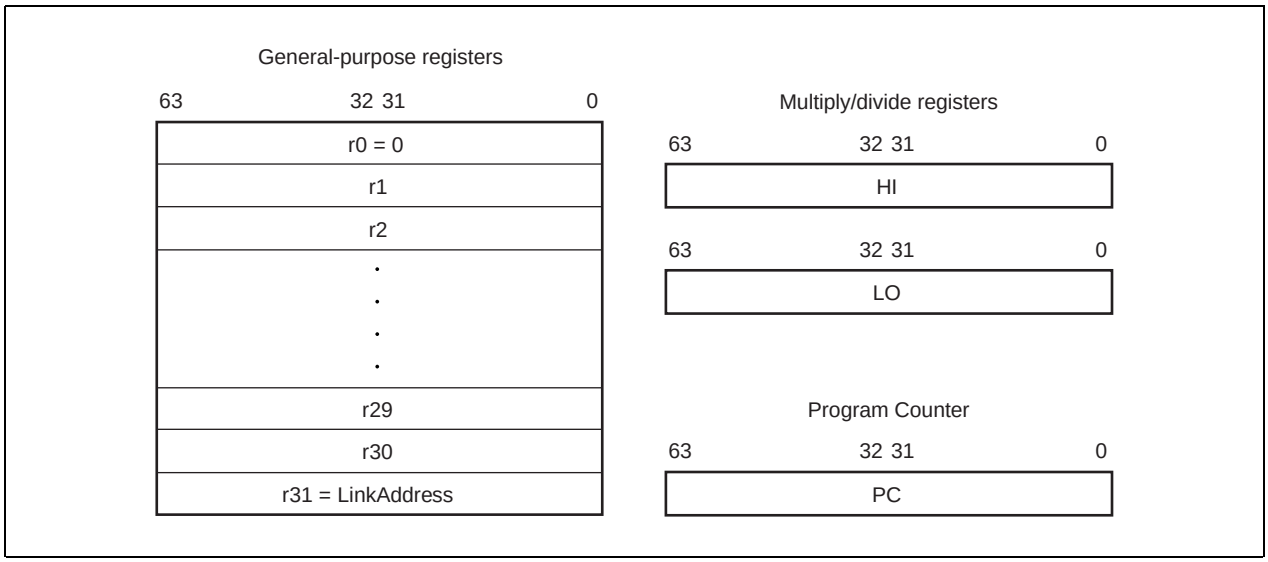
CPU registers can operate as either 32-bit or 64-bit registers, depending on the VR4181 processor mode of operation.

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

The VR4181 has no Program Status Word (PSW) register as such; this is covered by the Status and Cause registers incorporated within the CP0 (see **1.4.4 CP0 registers**).

Figure 1-3 shows the CPU registers.

Figure 1-3. CPU Registers



★ **1.4.2 CPU instruction set overview**

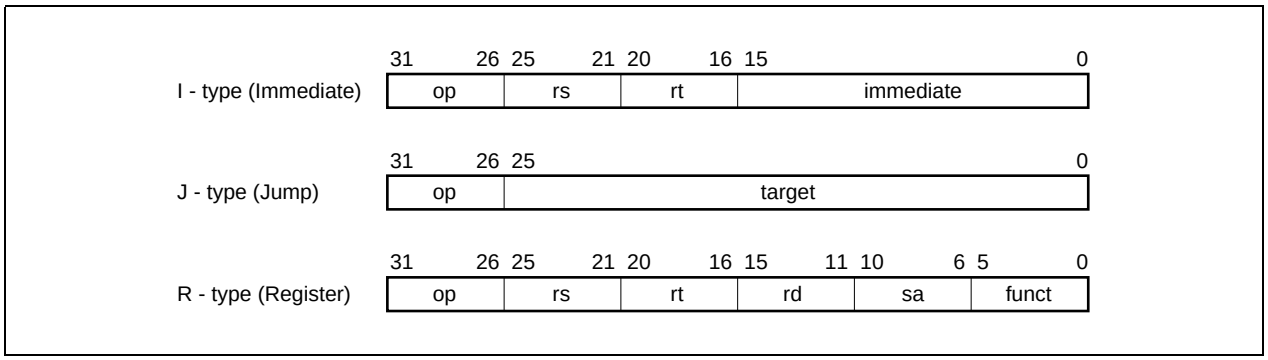
There are two types of CPU instructions: 32-bit length instructions (MIPS III) and 16-bit length instructions (MIPS16). Use of the MIPS16 instructions is enabled or disabled by setting MIPS16EN pin during a reset.

For details about instruction formats and their fields in each instruction set and operation of each instruction, refer to **VR4100 Series Architecture User's Manual**.

(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).

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.

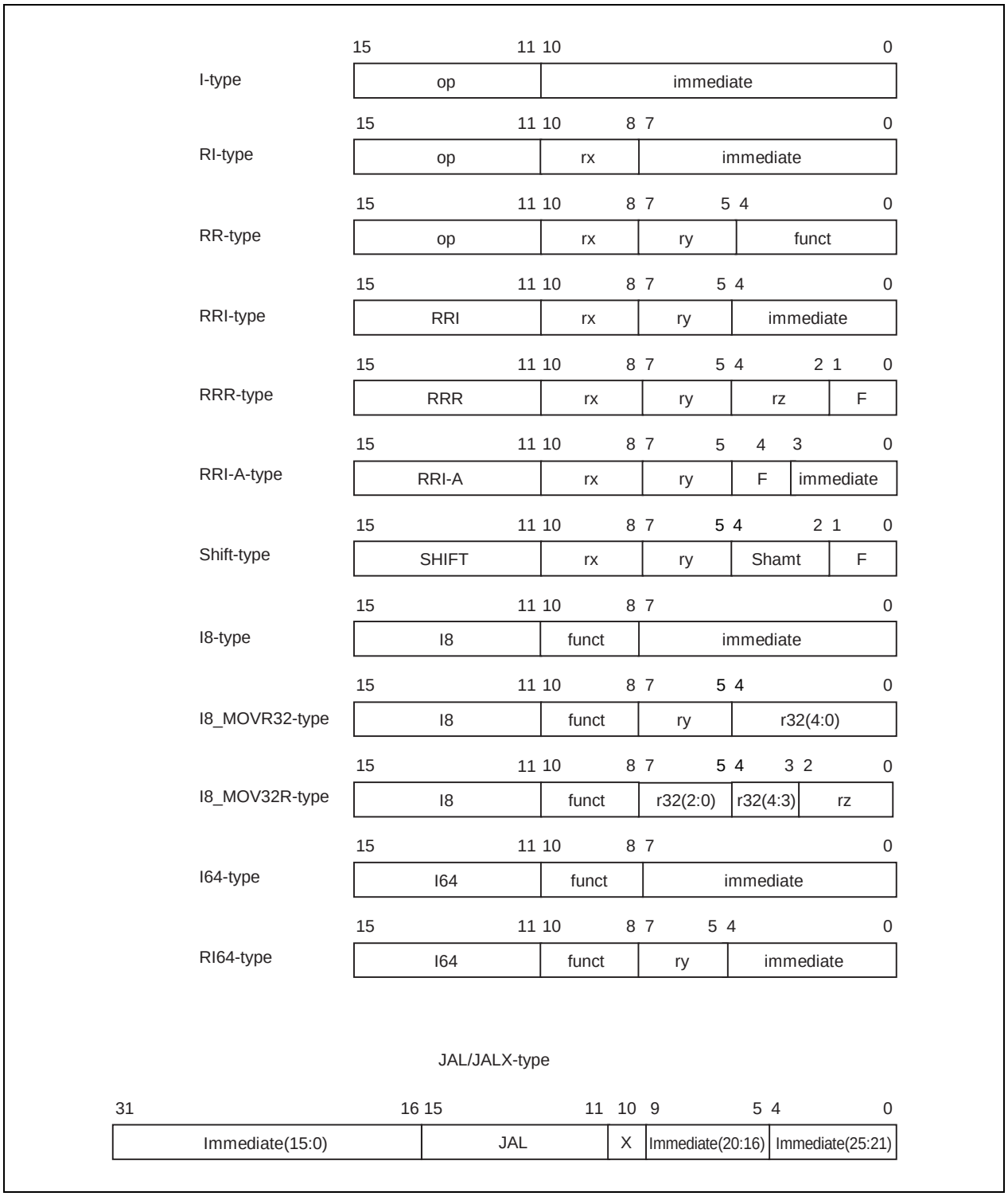
(d) System control coprocessor (CP0) instructions perform operations on CP0 registers to control the memory-management and exception-handling facilities of the processor.

(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.

(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.

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, RRIA, 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.

★ 1.4.3 Data formats and addressing

The VR4181 uses the following four data formats:

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

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. However, the VR4181 only support the little-endian order.

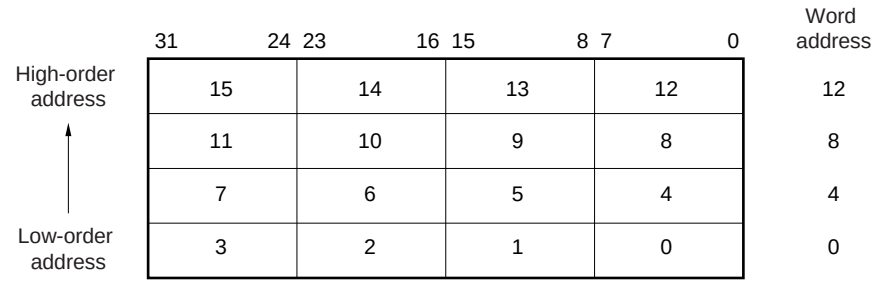
Endianness refers to the location of byte 0 within the multi-byte data structure. Figure 1-6 show the configuration.

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

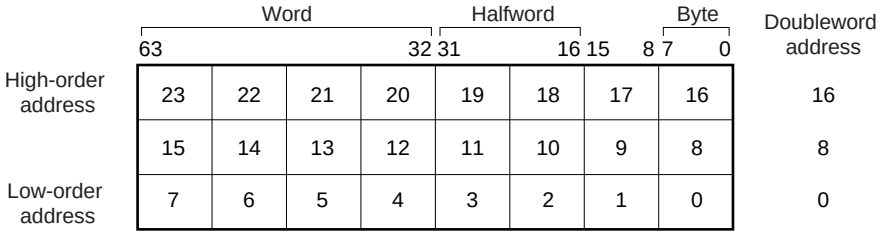
In this manual, bit designations are always little endian.

Figure 1-6. Byte Address in Little-Endian Byte Order

(a) Word data



(b) Doubleword data



Remarks 1. The lowest byte is the lowest address.

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

The 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:

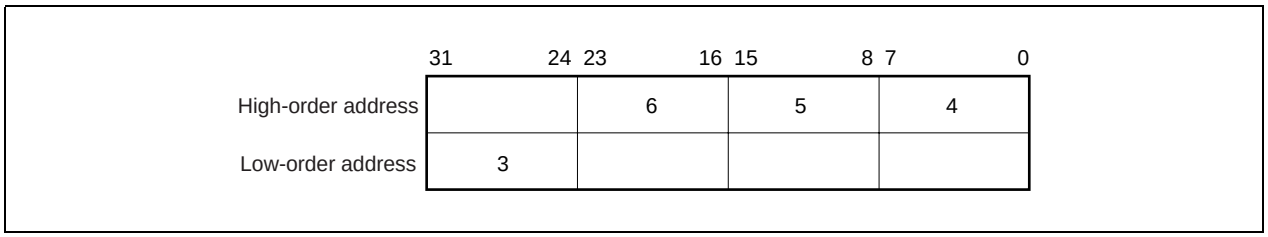
- Word access: LWL, LWR, SWL, SWR
- Doubleword access: LDL, LDR, SDL, SDR

These instructions are used in pairs of L and R.

Accessing unaligned data requires one additional instruction cycle (1 PCycle) over that required for accessing aligned data.

Figure 1-7 shows the access of an unaligned word that has byte address 3.

Figure 1-7. Unaligned Word Accessing (Little Endian)



★ 1.4.4 CP0 registers

The CP0 has thirty-two registers, each of which has its own register number.

Table 1-6 shows simple descriptions of each register. For the detailed descriptions of the registers, refer to **CHAPTER 3 CP0 REGISTERS**.

Table 1-6. System Control Coprocessor (CP0) Register Definitions

Number	Register	Usage	Description
0	Index	Memory management	Programmable pointer to TLB array
1	Random	Memory management	Pseudo-random pointer to TLB array (read only)
2	EntryLo0	Memory management	Lower half of TLB entry for even VPN
3	EntryLo1	Memory management	Lower half of TLB entry for odd VPN
4	Context	Exception processing	Pointer to kernel virtual PTE in 32-bit mode
5	PageMask	Memory management	Page size specification
6	Wired	Memory management	Number of wired TLB entries
7	–	–	Reserved for future use
8	BadVAddr	Exception processing	Virtual address where the most recent error occurred
9	Count	Exception processing	Timer count
10	EntryHi	Memory management	Higher half of TLB entry (including ASID)
11	Compare	Exception processing	Timer compare value
12	Status	Exception processing	Status indication
13	Cause	Exception processing	Cause of last exception
14	EPC	Exception processing	Exception Program Counter
15	PRId	Memory management	Processor revision identifier
16	Config	Memory management	Configuration (memory system modes) specification
17	LLAddr Note1	Memory management	Physical address for self diagnostics
18	WatchLo	Exception processing	Memory reference trap address low bits
19	WatchHi	Exception processing	Memory reference trap address high bits
20	XContext	Exception processing	Pointer to kernel virtual PTE in 64-bit mode
21 to 25	–	–	Reserved for future use
26	Parity Error Note2	Exception processing	Cache parity bits
27	Cache Error Note2	Exception processing	Index and status of cache error
28	TagLo	Memory management	Lower half of cache tag
29	TagHi	Memory management	Higher half of cache tag
30	ErrorEPC	Exception processing	Error Exception Program Counter
31	–	–	Reserved for future use

Notes1. This register is defined to maintain compatibility with the VR4000™ and VR4400™. This register is meaningless during normal operations.

2. This register is defined to maintain compatibility with the VR4100™. This register is not used in the VR4181 hardware.

1.4.5 Floating-point unit (FPU)

The VR4181 does not support the floating-point unit (FPU). Coprocessor Unusable exception will occur if any FPU instructions are executed. If necessary, FPU instructions should be emulated by software in an exception handler.

1.4.6 Memory management unit

The VR4181 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 VR4181 has three operating modes: User, Supervisor, and Kernel. The manner in which memory addresses are mapped depends on these operating modes.

In addition, the VR4181 supports the 32-bit and 64-bit addressing modes. The manner in which memory addresses are translated or mapped depends on these addressing modes.

A detailed description of the physical address space is given in **CHAPTER 4 MEMORY MANAGEMENT SYSTEM**. For details about the virtual address space, refer to **VR4100 Series Architecture User's Manual**.

(1) Translation lookaside buffer (TLB)

Virtual memory mapping is performed using the translation lookaside buffer (TLB). The TLB translates virtual addresses to physical addresses. It runs by a full-associative method and has 32 entries, each of which two successive pages are mapped.

The TLB of the VR4181 holds both instruction addresses and data addresses so that it is called as joint TLB (JTLB).

The page size can be configured, on a per-entry basis, to map a page size of 1 KB to 256 KB, in power of four. 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 an 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.

1.4.7 Cache

The VR4181 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 VR4181 has a storage capacity of 4 KB, while the data cache has a capacity of 4 KB.

For details about caches, refer to **VR4100 Series Architecture User's Manual**.

1.4.8 Instruction pipeline

The VR4181 has a 5-stage instruction pipeline. Under normal circumstances, one instruction is issued each cycle. For details, refer to **VR4100 Series Architecture User's Manual**.

★ 1.4.9 Power modes

The VR4181 supports four power modes: Fullspeed mode, Standby mode, Suspend mode, and Hibernate mode. A detailed description of these power modes is also given in **CHAPTER 10 POWER MANAGEMENT UNIT (PMU)**.

(1) Fullspeed mode

This is the normal operation mode.

The VR4181's default status sets operation under Fullspeed mode. After a reset, the VR4181 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, the pipeline clock (PClock) in the CPU core is 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.

During Standby mode, the processor returns to Fullspeed mode if any interrupt request occurs.

(3) Suspend mode

When the SUSPEND instruction has been executed, the processor can be set to Suspend mode. During Suspend mode, the pipeline clock (PClock) in the CPU core is held at high level. The VR4181 also stops supplying TClock and PCLK to peripheral units. While in this mode, the register and cache contents are retained. Contents of DRAM can also be retained by putting DRAM into self-refresh mode.

During Suspend mode, the processor returns to Fullspeed mode if any of power-on factors or some of interrupt requests occurs.

(4) Hibernate mode

When the HIBERNATE instruction has been executed, the processor can be set to Hibernate mode. During Hibernate mode, clocks other than the RTC clock (32.768 kHz) are held at high level and the PLL stops. While in this mode, contents of the registers and caches are not retained. Contents of DRAM can be retained by putting DRAM into self-refresh mode.

Power consumption during Hibernate mode is about 0 W if power to 2.5 V power supply is not applied (it does not go completely to 0 W due to the existence of a 32.768 kHz oscillator or on-chip peripheral circuits that operate at 32.768 kHz).

During Hibernate mode, the processor returns to Fullspeed mode if any of power-on factors or some of interrupt requests occurs.

★ **1.4.10 Code compatibility**

The VR4110 core is designed in consideration of the program compatibility to other VR-Series processors. However since it has some differences from other processors on their architecture, it cannot necessarily execute all programs that can be executed in other VR-Series processors, and also other VR-Series processors cannot necessarily execute all programs that can be executed in the VR4110 core.

Matters that should be paid attention to when porting programs between the VR4110 core and other VR-Series processors are listed below.

- A 16-bit length MIPS16 instruction set is added in the VR4110 core.
- Multiply-add instructions (MADD16, DMADD16) are added in the VR4110 core.
- Instructions for power modes (HIBERNATE, STANDBY, SUSPEND) are added in the VR4110 core to support power modes.
- The VR4110 core does not support floating-point instructions since it has no Floating-Point Unit (FPU).
- The VR4110 core does not have the LL bit to perform synchronization of multiprocessing. Therefore, it does not support instructions that manipulate the LL bit (LL, LLD, SC, SCD).
- The CP0 hazards of the VR4110 core are equally or less stringent than those of the VR4000.

For more information about each instruction, refer to **VR4100 Series Architecture User's Manual**, and user's manuals of each product other than the VR4100 Series.

Instructions supported by each of the VR Series processors are listed below.

Table 1-7. List of Instructions Supported by VR Series Processors

Supported instructions \ Products		VR4181 VR4111	VR4121 VR4122™	VR4300™ VR4305™ VR4310™	VR5000A™	VR5432™	VR10000™ VR12000™
MIPS I		A	A	A	A	A	A
MIPS II		A	A	A	A	A	A
MIPS III		A	A	A	A	A	A
	LL bit manipulation	N/A	N/A	A	A	A	A
MIPS IV		N/A	N/A	N/A	A	A	A
MIPS16		A	A	N/A	N/A	N/A	N/A
Multiply-add		A (16 bits)	A (32 bits)	N/A	N/A	A (32 bits)	N/A
Floating-point operation		N/A	N/A	A	A	A	A
Power mode transition		A	A	N/A	A	A	N/A

1.5 Clock Interface

The VR4181 has the following eight clocks.

- CLKX1, CLKX2 (input)
  These are oscillation inputs of 18.432 MHz, and used to generate operation clocks for the CPU core, serial interface, and other peripheral units.

- RTCX1, RTCX2 (input)
  These are oscillation inputs of 32.768 kHz, and used for PMU, RTC, and so on.

- PClock (internal)
  This clock is used to control the pipeline in the VR4110 core, and for units relating to the pipeline. This clock is generated from the clock input of CLKX1 and CLKX2 pins via the PLL. Its frequency is determined by CLKSEL(2:0) pins.

- MasterOut (internal)
★ This is a bus clock of the VR4110 core, and used for interrupt control. This clock operates in frequency of 1/4 of the TClock frequency. The contents of the CP0's Count register are incremented synchronously with this clock.

- TClock (internal)
  This is an operation clock for internal MBA bus and is supplied to the internal MBA modules (memory controller,
★ LCD controller, and DMA controller). This clock is generated from PClock and its frequency is 1/1, 1/2, or 1/3 of the PClock frequency (it is determined by internal register setting). It is set to 1/2 by default.

- PCLK (internal)
  This clock is supplied to the internal ISA peripherals. This clock is generated from TClock and its frequency is
★ determined by internal register setting. PCLK will operate only when accesses to the internal ISA bus occur.

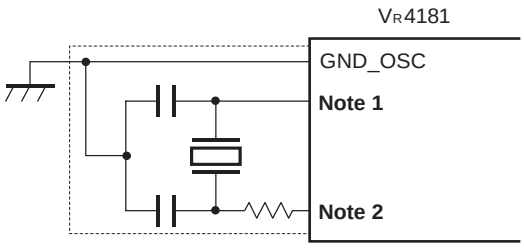
- SYSCLK (internal, output)
  This clock is used as the external ISA bus clock. It is also supplied to the internal CompactFlash controller. This
★ clock is generated from PCLK and its frequency is determined by internal register setting. SYSCLK will operate only when accesses to the external ISA bus occur.

★ - SDCLK (output)
  This clock is supplied to SDRAM. This clock operates in the same frequency as that of TClock. SDCLK will operate only when accesses to SDRAM occur.

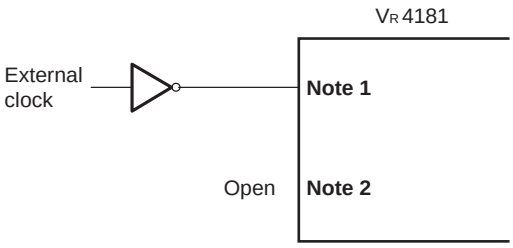
Figure 1-8 shows the external circuits of the clock oscillator.

★ **Figure 1-8. 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.

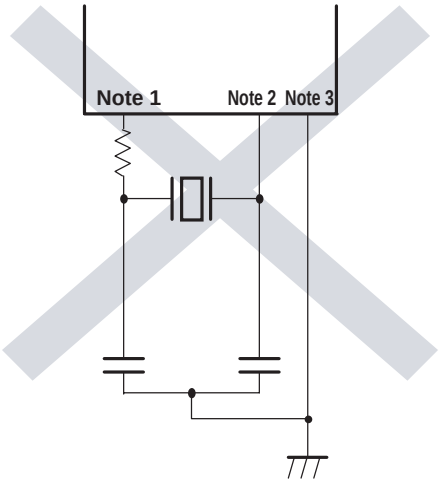
- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as GND. Do not ground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

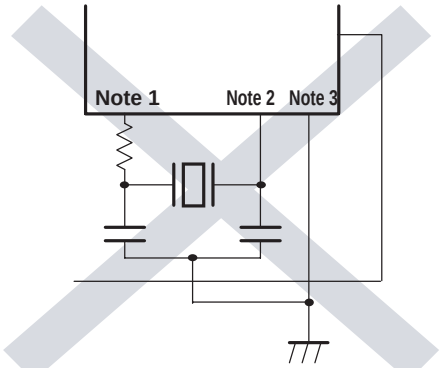
Figure 1-9 shows examples of the incorrect connection circuit of the resonator.

★ **Figure 1-9. Incorrect Connection Circuits of Resonator**

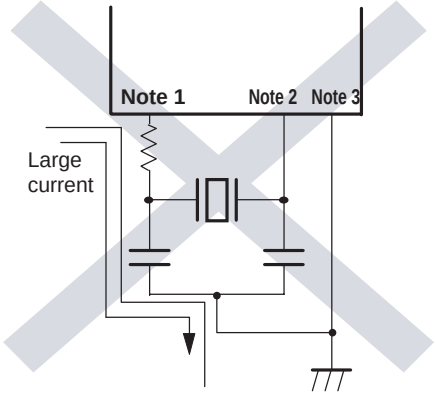
(a) Connection circuit wiring is too long.



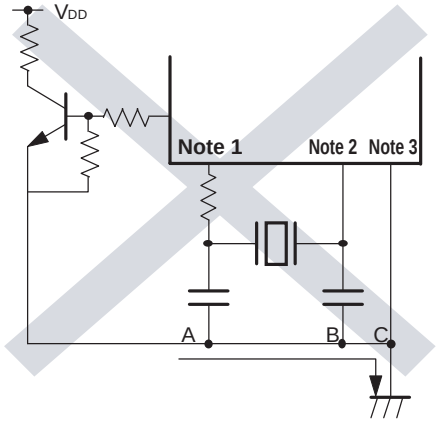
(b) There is another signal line crossing.



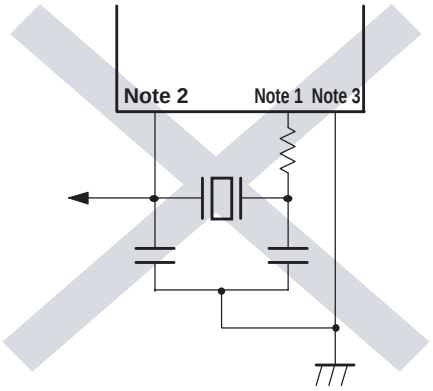
(c) A high fluctuating current flows near a signal line.



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



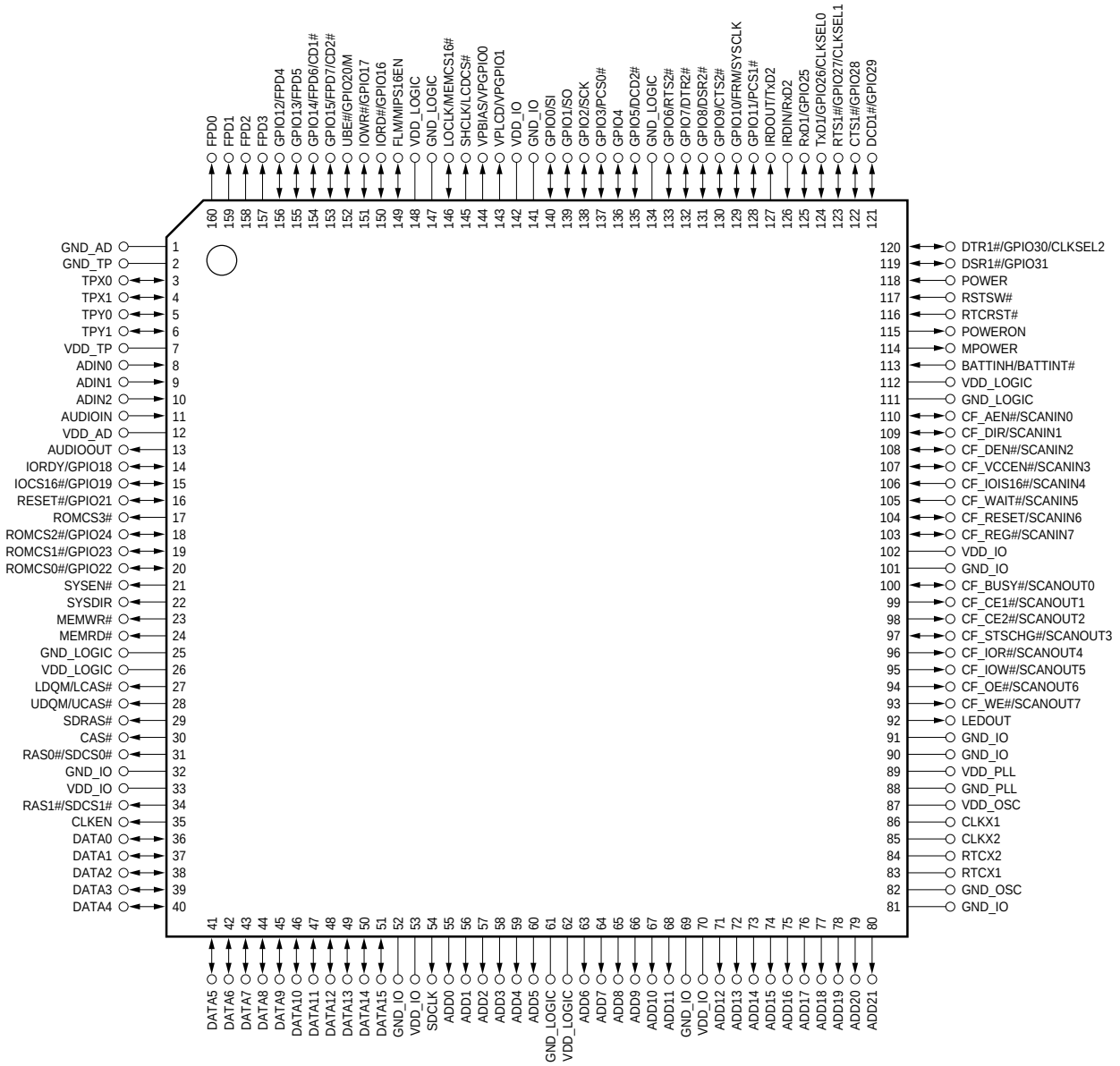
(e) A signal is fetched.



Notes1. CLKX2, RTCX2
2. CLKX1, RTCX1
3. GND_OSC

2.1 Pin Configuration

- 160-pin plastic LQFP (fine pitch) (24 × 24)



Remark # indicates active low.

Pin Identification

ADD(21:0) : Address Bus
ADIN(2:0) : Analog Data Input
AUDIOIN : Audio Input
AUDIOOUT : Audio Output
BATTINH : Battery Inhibit
BATTINT# : Battery Interrupt
CAS# : Column Address Strobe
CD1#, CD2# : Card Detect for CompactFlash
CF_AEN# : Address Enable for CompactFlash Buffer
CF_BUSY# : Ready/Busy/Interrupt Request for CompactFlash
CF_CE(2:1)# : Card Enable for CompactFlash
CF_DEN# : Data Enable for CompactFlash Buffer
CF_DIR : Data Direction for CompactFlash Buffer
CF_IOIS16# : I/O is 16 bits for CompactFlash
CF_IOR# : I/O Read Strobe for CompactFlash
CF_IOW# : I/O Write Strobe for CompactFlash
CF_OE# : Output Enable for CompactFlash
CF_REG# : Register Memory Access for CompactFlash
CF_RESET : Reset for CompactFlash
CF_STSCHG# : Status Change of CompactFlash
CF_VCCEN# : V_{CC} Enable for CompactFlash
CF_WAIT# : Wait Input for CompactFlash
CF_WE# : Write Enable for CompactFlash
CLKEN : Clock Enable for SDRAM
CLKSEL(2:0) : Clock Select
CLKX1, CLKX2 : Clock Input
CTS1#, CTS2# : Clear to Send
DATA(15:0) : Data Bus
DCD1#, DCD2# : Data Carrier Detect
DSR1#, DSR2# : Data Set Ready
DTR1#, DTR2# : Data Terminal Ready
FLM : First Line Clock for LCD
FPD(7:0) : Screen Data of LCD
FRM : Clocked Serial Frame
GND_AD : Ground for A/D and D/A Converter
GND_IO : Ground for I/O
GND_LOGIC : Ground for Logic
GND_OSC : Ground for Oscillator
GND_PLL : Ground for PLL
GND_TP : Ground for Touch Panel
GPIO(31:0) : General Purpose I/O
IOCS16# : I/O 16-bit Bus Sizing
IORD# : I/O Read
IORDY : I/O Ready
IOWR# : I/O Write
IRDIN : IrDA Data Input
IRDOUT : IrDA Data Output
LCAS# : Lower Column Address Strobe
LCDCS# : Chip Select for LCD
LDQM : Lower Byte Enable for SDRAM
LEDOUT : LED Output
LOCLK : Load Clock for LCD
M : LCD Modulation Clock
MEMCS16# : Memory 16-bit Bus Sizing
MEMRD# : Memory Read
MEMWR# : Memory Write
MIPS16EN : MIPS16 Enable
MPOWER : Main Power
PCS(1:0)# : Programmable Chip Select
POWER : Power Switch
POWERON : Power On State
RAS(1:0)# : Row Address Strobe for DRAM
RESET# : Reset Output
ROMCS(3:0)# : Chip Select for ROM
RSTSW# : Reset Switch
RTCRST# : Real-time Clock Reset
RTCX1, RTCX2 : Real-time Clock Input
RTS1#, RTS2# : Request to Send
RxD1, RxD2 : Receive Data
SCANIN(7:0) : Scan Data Input
SCANOUT(7:0) : Scan Data Output
SCK : CSI (Clocked Serial Interface) Clock
SDCLK : Operation Clock for SDRAM
SDCS(1:0)# : Chip Select for SDRAM
SDRAS# : Row Address Strobe for SDRAM
SHCLK : Shift Clock for LCD
SI : Clocked Serial Data Input
SO : Clocked Serial Data Output
SYSCLK : System Clock for System Bus
SYSDIR : System Data Direction
SYSEN# : System Data Enable
TPX(1:0) : Touch Panel Data of X
TPY(1:0) : Touch Panel Data of Y
TxD1, TxD2 : Transmit Data
UBE# : Upper Byte Enable for System Bus
UCAS# : Upper Column Address Strobe for DRAM
UDQM : Upper Byte Enable for SDRAM
VDD_AD : Power Supply for A/D and D/A Converter
VDD_IO : Power Supply for I/O
VDD_LOGIC : Power Supply for Logic
VDD_OSC : Power Supply for Oscillator
VDD_PLL : Power Supply for PLL
VDD_TP : Power Supply for Touch Panel
VPBIAS : Bias Power Control for LCD
VPGPIO(1:0) : General Purpose Output for LCD Panel Power Control
VPLCD : Logic Power Control for LCD

Remark # indicates active low.

2.2 Pin Function Description

Remark # indicates active low.

2.2.1 System bus interface signals

(1/2)

Signal name	I/O	Description of function
ADD(21:0)[Note]	Output	Address bus. Used to specify address for the DRAM, ROM, flash memory, or system bus (ISA).
DATA(15:0)	I/O	Data bus. Used to transmit and receive data between the VR4181 and DRAM, ROM, flash memory, or system bus.
IORD#/GPIO16	I/O	System bus I/O read signal output or general-purpose I/O. It is active when the VR4181 accesses the system bus to read data from an I/O port when configured as IORD#.
IOWR#/GPIO17	I/O	System bus I/O write signal output or general-purpose I/O. It is active when the VR4181 accesses the system bus to write data to an I/O port when configured as IOWR#.
IORDY/GPIO18	I/O	System bus I/O channel ready input or general-purpose I/O. Set this signal as active when system bus controller is ready to be accessed by the VR4181 when configured as IORDY.
IOCS16#/GPIO19	I/O	Bus sizing request input for system bus I/O or general-purpose I/O. Set this signal as active when system bus I/O accesses data in 16-bit width, if configured as IOCS16#.
UBE#/GPIO20/M	I/O	System bus upper byte enable output, general-purpose input, or LCD modulation output. During system bus accesses, this signal is active when the high-order byte is valid on the data bus.
RESET#/GPIO21	I/O	System bus reset output or general-purpose I/O. It is active when the VR4181 resets the system bus controller when configured as RESET#.

★ **Note** The VR4181 utilizes different addressings depending on the types of the external accesses.
During ROM accesses, bits 22 to 1 of the internal address lines are output to the ADD(21:0) pins (the minimum transfer data width is a half word (1 word = 32 bits)).
During accesses other than ROM accesses, bits 21 to 0 of the internal address lines are output to the ADD(21:0) pins (the minimum transfer data width is 1 byte).

(2/2)

Signal name	I/O	Description of function
★ SYSDIR[Note]	Output	Data bus isolation buffer direction control. This signal is valid only when ROM, ISA, or CompactFlash accesses are enabled. This becomes low level during ROM, ISA, or CompactFlash read cycle, or becomes high level during ROM, ISA, or CompactFlash write cycle.
★ SYSEN#[Note]	Output	Data bus isolation buffer enable. This signal is valid only when ROM, ISA, or CompactFlash accesses are enabled. This becomes active during ROM or ISA cycle.
SDCS(1:0)#/RAS(1:0)#	Output	SDRAM chip select for bank 0 and bank 1 or EDO DRAM row address strobes.
CAS#	Output	SDRAM column address strobe. Leave unconnected when using EDO DRAM.
SDRAS#	Output	SDRAM row address strobe. Leave unconnected when using EDO DRAM.
UDQM/UCAS#	Output	SDRAM upper byte enable or EDO DRAM upper byte column address strobe.
LDQM/LCAS#	Output	SDRAM lower byte enable or EDO DRAM lower byte column address strobe.
SDCLK	Output	SDRAM operating clock.
CLKEN	Output	SDRAM clock enable output (CKE).
ROMCS3#	Output	ROM chip select output for bank 3.
ROMCS2#/GPIO24	I/O	ROM chip select output for bank 2, or general-purpose I/O.
ROMCS1#/GPIO23	I/O	ROM chip select output for bank 1, or general-purpose I/O.
ROMCS0#/GPIO22	I/O	ROM chip select output for bank 0, or general-purpose I/O.
MEMRD#	Output	Memory read signal for ROM and system bus.
MEMWR#	Output	Memory write signal for ROM, DRAM and system bus.

Note The SYSEN# and SYSDIR signals control a buffer which is used to isolate SDRAM data bus from the bus of other low speed devices. By isolating the high-speed data bus of SDRAM, the load of the data bus between the VR4181 and SDRAM is reduced.

★ When the EXBUFFEN bit of the XISACTL register is cleared to 0, the SYSEN# and SYSDIR signals start their operation. These signals keep low level until EXBUFFEN bit is cleared to 0 after a reset.

When an isolation buffer is used, SYSEN# and SYSDIR signals function as follows;

SYSEN#	SYSDIR	Bus operation
0	0	External ISA, CompactFlash, or ROM read cycle
0	1	External ISA, CompactFlash, or flash memory mode write cycle
1	Don't care	External Buffer Disable DRAM read/write cycle or Hibernate mode

2.2.2 LCD interface signals

Signal name	I/O	Description of function
SHCLK/LCDCS#	Output	LCD shift clock output or chip select for external LCD controller.
LOCLK/MEMCS16#	I/O	LCD load clock output or bus sizing request input for system bus memory access. When using as MEMCS16#, the external agent must activate this signal at the system bus memory access in 16-bit width.
FLM/MIPS16EN	I/O	The function of this pin differs depending on the operating status. <During RTC reset (input)> This signal enables use of MIPS16 instructions. 0: Disable use of MIPS16 instructions 1: Enable use of MIPS16 instructions <During normal operation (output)> LCD first line clock output.
FPD(7:4)/GPIO(15:12)Note	Output	See **2.2.11 General-purpose I/O signals** in this section.
FPD(3:0)Note	Output	LCD screen data.
VPLCD/VPGPIO1	Output	LCD logic power control. This signal may be defined as a general-purpose output when an external LCD controller is used.
VPBIAS/VPGPIO0	Output	LCD bias power control. This signal may be defined as a general-purpose output when an external LCD controller is used.

Note Connection between FPD(7:0) of the VR4181 and LCD panel data lines differs depending on the panel data width as below.

For details, refer to **CHAPTER 21 LCD CONTROLLER**.

VR4181	LCD Panel Data (4-bit width)	LCD Panel Data (8-bit width)
FPD0	Data Line 0	Data Line 4
FPD1	Data Line 1	Data Line 5
FPD2	Data Line 2	Data Line 6
FPD3	Data Line 3	Data Line 7
FPD4	–	Data Line 0
FPD5	–	Data Line 1
FPD6	–	Data Line 2
FPD7	–	Data Line 3

2.2.3 Initialization interface signals

Signal name	I/O	Description of function
POWER	Input	VR4181 activation signal.
RSTSW#	Input	VR4181 reset signal.
RTCRST#	Input	Reset signal for internal Real-time clock and internal logic. When power is first supplied to the system, the external agent must activate this signal.
POWERON	Output	This signal indicates that the VR4181 is ready to operate. It becomes active when a power-on factor is detected and becomes inactive when the BATTINH/BATTINT# signal check has been completed.
MPOWER	Output	This signal indicates that the VR4181 is operating. This signal is inactive during Hibernate mode. During this signal being inactive, turn off the 2.5 V power supply.

2.2.4 Battery monitor interface signals

Signal name	I/O	Description of function
BATTINH/BATTINT#	Input	The function of this pin differs depending on the state of the MPOWER pin. <When MPOWER = 0> BATTINH signal Enables or disables activation on power application. 1: Enable activation 0: Disable activation <When MPOWER = 1> BATTINT# signal This is an interrupt signal that is input when remaining battery power is low during normal operations. The external agent checks the remaining battery power and activates this signal if voltage sufficient for operations cannot be supplied.

2.2.5 Clock interface signals

Signal name	I/O	Description of function
RTCX(2:1)	–	Connections to 32.768 kHz crystal resonator.
CLKX(2:1)	–	Connections to 18.432 MHz crystal resonator.

2.2.6 Touch panel interface and audio interface signals

Signal name	I/O	Description of function
TPX(1:0)	I/O	Touch panel X coordinate data. They use the voltage applied to the X coordinate and the voltage input to the Y coordinate to detect which coordinates on the touch panel are being pressed.
TPY(1:0)	I/O	Touch panel Y coordinate data. They use the voltage applied to the Y coordinate and the voltage input to the X coordinate to detect which coordinates on the touch panel are being pressed.
ADIN(2:0)	Input	General-purpose A/D data inputs.
AUDIOIN	Input	Audio input.
AUDIOOUT	Output	Audio output.

2.2.7 LED interface signals

Signal name	I/O	Description of function
LEDOUT	Output	This is an output signal for lighting LEDs.

2.2.8 CompactFlash interface and keyboard interface signals

Signal name	I/O	Description of function
CF_WE#/SCANOUT7	Output	CompactFlash write enable output or keyboard scan data output.
CF_OE#/SCANOUT6	Output	CompactFlash output enable or keyboard scan data output.
CF_IOW#/SCANOUT5	Output	CompactFlash I/O write strobe output or keyboard scan data output.
CF_IOR#/SCANOUT4	Output	CompactFlash I/O read strobe output or keyboard scan data output.
CF_STSCHG#/SCANOUT3	I/O	CompactFlash status changed input or keyboard scan data output.
CF_CE(2:1)#/ SCANOUT(2:1)	Output	CompactFlash card enable outputs or keyboard scan data outputs.
CF_BUSY#/SCANOUT0	I/O	CompactFlash ready/busy/interrupt request indication input or keyboard scan data output.
CF_REG#/SCANIN7	I/O	CompactFlash register select output or keyboard scan data input.
CF_RESET/SCANIN6	I/O	CompactFlash reset output or keyboard scan data input.
CF_WAIT#/SCANIN5	Input	CompactFlash wait input or keyboard scan data input.
CF_IOIS16#/SCANIN4	Input	CompactFlash I/O 16-bit bus input or keyboard scan data input.
CF_VCCEN#/SCANIN3	I/O	CompactFlash V_{CC} enable output or keyboard scan data input.
CF_DEN#/SCANIN2	I/O	CompactFlash data buffer enable output or keyboard scan data input.
CF_DIR/SCANIN1	I/O	CompactFlash data direction control output or keyboard scan data input.
CF_AEN#/SCANIN0	I/O	CompactFlash address buffer enable output or keyboard scan data input.

2.2.9 Serial interface channel 1 signals

Signal name	I/O	Description of function
RxD1/GPIO25	I/O	Serial channel 1 receive data input or general-purpose I/O.
TxD1/GPIO26/CLKSEL0	I/O	The function of this pin differs depending on the operating status. <During RTC reset (input)> This signal is used to set CPU core operation clock frequency[Note]. <During normal operation (input/output)> Serial channel 1 transmit data output or general-purpose I/O.
RTS1#/GPIO27/CLKSEL1	I/O	The function of this pin differs depending on the operating status. <During RTC reset (input)> This signal is used to set CPU core operation clock frequency[Note]. <During normal operation (input/output)> Serial channel 1 request to send output or general-purpose I/O.
CTS1#/GPIO28	I/O	Serial channel 1 clear to send input or general-purpose I/O.
DCD1#/GPIO29	I/O	Serial channel 1 data carrier detect input or general-purpose I/O.
DTR1#/GPIO30/CLKSEL2	I/O	The function of this pin differs depending on the operating status. <During RTC reset (input)> This signal is used to set CPU core operation clock frequency[Note]. <During normal operation (input/output)> Serial channel 1 data terminal ready output or general-purpose I/O.
DSR1#/GPIO31	I/O	Serial channel 1 Data set ready input or general-purpose I/O.

Note CLKSEL(2:0) signals are used to set the frequency of the CPU core operation clock (PClock). These signals are sampled when the RTCRST# signal goes high.

The relationship between the CLKSEL(2:0) pin settings and clock frequency is shown below.

CLKSEL(2:0)	CPU core operation frequency (PClock)
111	Reserved (98.1 MHz)
110	Reserved (90.6 MHz)
101	Reserved (84.1 MHz)
100	Reserved (78.5 MHz)
011	Reserved (69.3 MHz)
010	65.4 MHz
001	62.0 MHz
000	49.1 MHz

TClock is generated from PClock and its frequency is always 1/2 of the PClock frequency after RTC reset.

2.2.10 IrDA interface signals

Signal name	I/O	Description of function
IRDIN/RxD2	Input	IrDA receive data input or serial channel 2 receive data input. Connect this pin to GND (digital) via resistor when an IrDA receive component is connected.
IRDOUT/TxD2	Output	IrDA transmit data output or serial channel 2 transmit data output.

2.2.11 General-purpose I/O signals

Signal name	I/O	Description of function
GPIO(31:25)	I/O	See **2.2.9 Serial interface channel 1 signals** in this section
GPIO(24:16)	I/O	See **2.2.1 System bus interface signals** in this section.
GPIO15/FPD7/CD2#	I/O	General-purpose I/O, LCD screen data output, or CompactFlash card detect 2 input.
GPIO14/FPD6/CD1#	I/O	General-purpose I/O, LCD screen data output, or CompactFlash card detect 1 input.
GPIO13/FPD5	I/O	General-purpose I/O or LCD screen data output.
GPIO12/FPD4	I/O	General-purpose I/O or LCD screen data output.
GPIO11/PCS1#	I/O	General-purpose I/O or programmable chip select 1.
GPIO10/FRM/SYSCLK	I/O	General-purpose I/O, serial frame input for clocked serial interface, or external bus system clock output.
GPIO9/CTS2#	I/O	General-purpose I/O or serial channel 2 clear to send output.
GPIO8/DSR2#	I/O	General-purpose I/O or serial channel 2 data set ready input.
GPIO7/DTR2#	I/O	General-purpose I/O or serial channel 2 data terminal ready input.
GPIO6/RTS2#	I/O	General-purpose I/O or serial channel 2 request to send output.
GPIO5/DCD2#	I/O	General-purpose I/O or serial channel 2 data carrier detect input.
GPIO4	I/O	General-purpose I/O.
GPIO3/PCS0#	I/O	General-purpose I/O or programmable chip select 0.
GPIO2/SCK	I/O	General-purpose I/O or serial clock input for clocked serial interface.
GPIO1/SO	I/O	General-purpose I/O or serial data output signal for clocked serial interface.
GPIO0/SI	I/O	General-purpose I/O or serial data input signal for clocked serial interface.

2.2.12 Dedicated V_{DD}/GND signals

Signal name	Power supply	Description of function
VDD_PLL	2.5 V	Power supply dedicated for the PLL analog block.
GND_PLL	2.5 V	Ground dedicated for the PLL analog block.
VDD_TP	3.3 V	Power supply dedicated for the touch panel interface.
GND_TP	3.3 V	Ground dedicated for the touch panel interface.
VDD_AD	3.3 V	Power supply dedicated for the A/D and D/A converters. The voltage applied to this pin becomes the maximum value for the A/D and D/A interface signals.
GND_AD	3.3 V	Ground dedicated for the A/D and D/A converters. The voltage applied to this pin becomes the minimum value for the A/D and D/A interface signals.
VDD_OSC	3.3 V	Power supply dedicated for the oscillator.
GND_OSC	3.3 V	Ground dedicated for the oscillator.
VDD_LOGIC	2.5 V	Ordinary power supply of 2.5 V
GND_LOGIC	2.5 V	Ordinary ground of 2.5 V
VDD_IO	3.3 V	Ordinary power supply of 3.3 V
GND_IO	3.3 V	Ordinary ground of 3.3 V

Caution The V_R4181 has two types of power supplies. The 3.3 V power supply should be turned on at first. Turn on/off the 2.5 V power supply depending on the status of the MPOWER pin.

2.3 Pin Status in Specific Status

(1/3)

Signal Name	During RTC Reset	After RTC Reset	After Reset by Deadman's Switch or RSTSW	During Suspend Mode	During Hibernate Mode or Shutdown by HALTimer
ADD(21:0)	Hi-Z	0	0	**Note 1**	0
DATA(15:0)	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
MEMRD#	Hi-Z	1	1	1	Hi-Z
MEMWR#	Hi-Z	1	1	1	1
★ SDCS(1:0)#/RAS(1:0)#	Hi-Z	1	1	1/0 Note2	1/0 Note2
★ UDQM/UCAS#	Hi-Z	1	1	1/0 Note2	1/0 Note2
★ LDQM/LCAS#	Hi-Z	1	1	1/0 Note2	1/0 Note2
★ CAS#	Hi-Z	1	1	0	0
★ SDRAS#	Hi-Z	1	1	0	0
SDCLK	Hi-Z	Run	0	0	0
CLKEN	Hi-Z	1	1	1	0
SYSDIR	Hi-Z	0	0	0	0
★ SYSEN#	Hi-Z	0	0	0	0
IORD#/GPIO16	–	Hi-Z	Hi-Z	1/**Note 1**	Hi-Z/**Note 3**
IOWR#/GPIO17	–	Hi-Z	Hi-Z	1/**Note 1**	Hi-Z/**Note 3**
IORDY/GPIO18	–	Hi-Z	Hi-Z	**Note 1**	**Note 3**
IOCS16#/GPIO19	–	Hi-Z	Hi-Z	**Note 1**	**Note 3**
UBE#/GPIO20/M	–	Hi-Z	Hi-Z	1/**Note 1**/0	Hi-Z/**Note 3**/0
RESET#/GPIO21	–	Hi-Z	Hi-Z	**Note 1**	0/**Note 3**
ROMCS(2:0)#/GPIO(24:22)	–	Hi-Z	Hi-Z	1/**Note 1**	Hi-Z/**Note 3**
ROMCS3#	Hi-Z	Hi-Z	1	1	Hi-Z
SHCLK/LCDCS#	Hi-Z	0	0/1	0/1	0/Hi-Z
LOCLK/MEMCS16#	Hi-Z	0	0/–	0/–	0/–
FLM/MIPS16EN	**Note 4**	0	0	0	0
FPD(3:0)	Hi-Z	0	0	0	0
VPLCD/VPGPIO1	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
VPBIAS/VPGPIO0	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
POWER	–	–	–	–	–
RTCRST#	–	–	–	–	–
RSTSW#	–	–	–	–	–

Notes1. Maintains the state of the previous Fullspeed mode.

★ **2.** The state depends on the MEMCFG_REG register setting.

3. The state depends on the GPHIBSTH/GPHIBSTL register setting.

4. The input level is sampled to determine the MIPS16 instruction mode.

Remark 0: low level, 1: high level, Hi-Z: high impedance

(2/3)

Signal Name	During RTC Reset	After RTC Reset	After Reset by Deadman's Switch or RSTSW	During Suspend Mode	During Hibernate Mode or Shutdown by HALTimer
POWERON	–	–	0	0	0
MPOWER	0	0	1	1	0
BATTINH/BATTINT#	–	–	–	–	–
RTCX2, RTCX1	–	–	–	–	–
CLKX2, CLKX1	–	–	–	–	–
TPX(1:0)	–	1	1	**Note 1**	1
TPY(1:0)	–	Hi-Z	Hi-Z	**Note 1**	Hi-Z
ADIN(2:0)	–	–	–	–	–
AUDIOIN	–	–	–	–	–
AUDIOOUT	–	0	0	**Note 1**	0
CF_WE#/SCANOUT7	Hi-Z	Hi-Z	Hi-Z	**Note 1**	**Note 2**/Hi-Z
CF_OE#/SCANOUT6	Hi-Z	Hi-Z	Hi-Z	**Note 1**	**Note 2**/Hi-Z
CF_IOW#/SCANOUT5	Hi-Z	Hi-Z	Hi-Z	**Note 1**	**Note 2**/Hi-Z
CF_IOR#/SCANOUT4	Hi-Z	Hi-Z	Hi-Z	**Note 1**	**Note 2**/Hi-Z
CF_STSCHG#/SCANOUT3	Hi-Z	Hi-Z	Hi-Z	**Note 1**	**Note 1**/Hi-Z
CF_CE(2:1)#/ SCANOUT(2:1)	Hi-Z	Hi-Z	Hi-Z	**Note 1**	**Note 2**/Hi-Z
CF_BUSY#/SCANOUT0	Hi-Z	Hi-Z	Hi-Z	**Note 1**	**Note 1**/Hi-Z
CF_REG#/SCANIN7	Hi-Z	–	**Note 1**	**Note 1**	**Note 2/Note 1**
CF_RESET/SCANIN6	Hi-Z	–	**Note 1**	**Note 1**	**Note 3/Note 1**
CF_WAIT#/SCANIN5	–	–	**Note 1**	**Note 1**	–
CF_IOIS16#/SCANIN4	–	–	**Note 1**	**Note 1**	–
CF_VCCEN#/SCANIN3	Hi-Z	–	**Note 1**	**Note 1**	**Note 4/Note 1**
CF_DEN#/SCANIN2	Hi-Z	–	**Note 1**	**Note 1**	1/**Note 1**
CF_DIR/SCANIN1	Hi-Z	–	**Note 1**	**Note 1**	1/**Note 1**
CF_AEN#/SCANIN0	Hi-Z	–	**Note 1**	**Note 1**	1/**Note 1**

Notes1. Maintains the state of the previous Fullspeed mode.

2. When CF wake-up is enabled: Outputs high level.
When CF wake-up is disabled: Becomes high impedance.

3. When CF wake-up is enabled: Outputs low level.
When CF wake-up is disabled: Becomes high impedance.

4. When CF wake-up is enabled: Outputs low level.
When CF wake-up is disabled: Outputs high level.

Remark 0: low level, 1: high level, Hi-Z: high impedance

(3/3)

Signal Name	During RTC Reset	After RTC Reset	After Reset by Deadman's Switch or RSTSW	During Suspend Mode	During Hibernate Mode or Shutdown by HALTimer
RxD1/GPIO25	–	Hi-Z	Hi-Z	**Note 1**	**Note 1/Note 2**
TxD1/GPIO26/CLKSEL0	**Note 3**	Hi-Z	Hi-Z	**Note 1**	**Note 1/Note 2**
RTS1#/GPIO27/CLKSEL1	**Note 3**	Hi-Z	Hi-Z	**Note 1**	**Note 1/Note 2**
CTS1#/GPIO28	–	Hi-Z	Hi-Z	**Note 1**	**Note 1/Note 2**
DCD1#/GPIO29	–	Hi-Z	Hi-Z	**Note 1**	**Note 1/Note 2**
DTR1#/GPIO30/CLKSEL2	**Note 3**	Hi-Z	Hi-Z	**Note 1**	**Note 1/Note 2**
DSR1#/GPIO31	–	Hi-Z	Hi-Z	**Note 1**	**Note 1/Note 2**
IRDIN/RxD2	–	–	–	–	–
IRDOUT/TxD2	Hi-Z	Hi-Z	1	**Note 1**	Hi-Z
GPIO(15:14)/FPD(7:6)/ CD(2:1)#	–	Hi-Z	Hi-Z	**Note 1**/0/ **Note 1**	**Note 2/Note 1**
GPIO(13:12)/FPD(5:4)	–	Hi-Z	Hi-Z	**Note 1**/0	**Note 2/Note 1**
GPIO11/PCS1#	– /Hi-Z	Hi-Z	Hi-Z/1	**Note 1**/1	**Note 2**/Hi-Z
GPIO10/FRM/SYSCLK	– /Hi-Z	Hi-Z	Hi-Z	**Note 1**/0	**Note 2/Note 1**/ Hi-Z
GPIO9/CTS2#	–	Hi-Z	Hi-Z	**Note 1**	**Note 2/Note 1**
GPIO8/DSR2#	–	Hi-Z	Hi-Z	**Note 1**	**Note 2/Note 1**
GPIO7/DTR2#	–	Hi-Z	Hi-Z	**Note 1**	**Note 2/Note 1**
GPIO6/RTS2#	–	Hi-Z	Hi-Z	**Note 1**	**Note 2/Note 1**
GPIO5/DCD2#	–	Hi-Z	Hi-Z	**Note 1**	**Note 2/Note 1**
GPIO4	–	Hi-Z	Hi-Z	**Note 1**	**Note 2**
GPIO3/PCS0#	– /Hi-Z	Hi-Z	Hi-Z/1	**Note 1**/1	**Note 2**/Hi-Z
GPIO2/SCK	–	Hi-Z	Hi-Z	**Note 1**	**Note 2/Note 1**
GPIO1/SO	–	Hi-Z	Hi-Z	**Note 1**	**Note 2/Note 1**
GPIO0/SI	–	Hi-Z	Hi-Z	**Note 1**	**Note 2/Note 1**
LEDOUT	Hi-Z	1	**Note 1**	**Note 1**	**Note 1**

Notes1. Maintains the state of previous Fullspeed mode.

2. The state depends on the GPHIBSTH/GPHIBSTL register setting.

3. The input level is sampled to determine the CPU core operation frequency.

Remark 0: low level, 1: high level, Hi-Z: high impedance

★ 2.4 Recommended Connection of Unused Pins and I/O Circuit Types

(1/3)

Pin Name	Recommended Connection When Not Used	I/O Circuit Type
ADD(21:0)	–	A
DATA(15:0)	–	A
MEMRD#	–	A
MEMWR#	–	A
SDCS(1:0)#/RAS(1:0)#	–	A
UDQM/UCAS#	–	A
LDQM/LCAS#	–	A
CAS#	Leave open	A
SDRAS#	Leave open	A
SDCLK	Leave open	A
CLKEN	Leave open	A
SYSDIR	Leave open	A
SYSEN#	Leave open	A
IORD#/GPIO16	Connect to VDD_IO or GND_IO via resistor	A
IOWR#/GPIO17	Connect to VDD_IO or GND_IO via resistor	A
IORDY/GPIO18	Connect to VDD_IO or GND_IO via resistor	A
IOCS16#/GPIO19	Connect to VDD_IO or GND_IO via resistor	A
UBE#/GPIO20/M	Connect to VDD_IO or GND_IO via resistor	A
RESET#/GPIO21	Connect to VDD_IO or GND_IO via resistor	A
ROMCS(2:0)#/GPIO(24:22)	Connect to VDD_IO or GND_IO via resistor	A
ROMCS3#	–	A
SHCLK/LCDCS#	Leave open	A
LOCLK/MEMCS16#	Leave open	A
FLM/MIPS16EN	Connect to VDD_IO or GND_IO via resistor	A
FPD(3:0)	Leave open	A
VPLCD/VPGPIO1	Leave open	A
VPBIAS/VPGPIO0	Leave open	A
POWER	Connect to GND_IO via resistor	A
RTCRST#	–	A
RSTSW#	–	A
POWERON	Leave open	A
MPOWER	–	A
BATTINH/BATTINT#	–	A
TPX(1:0)	–	B
TPY(1:0)	–	C

Remark No specification (–) in the Recommended Connection When Not Used column indicates that the pin is always connected.

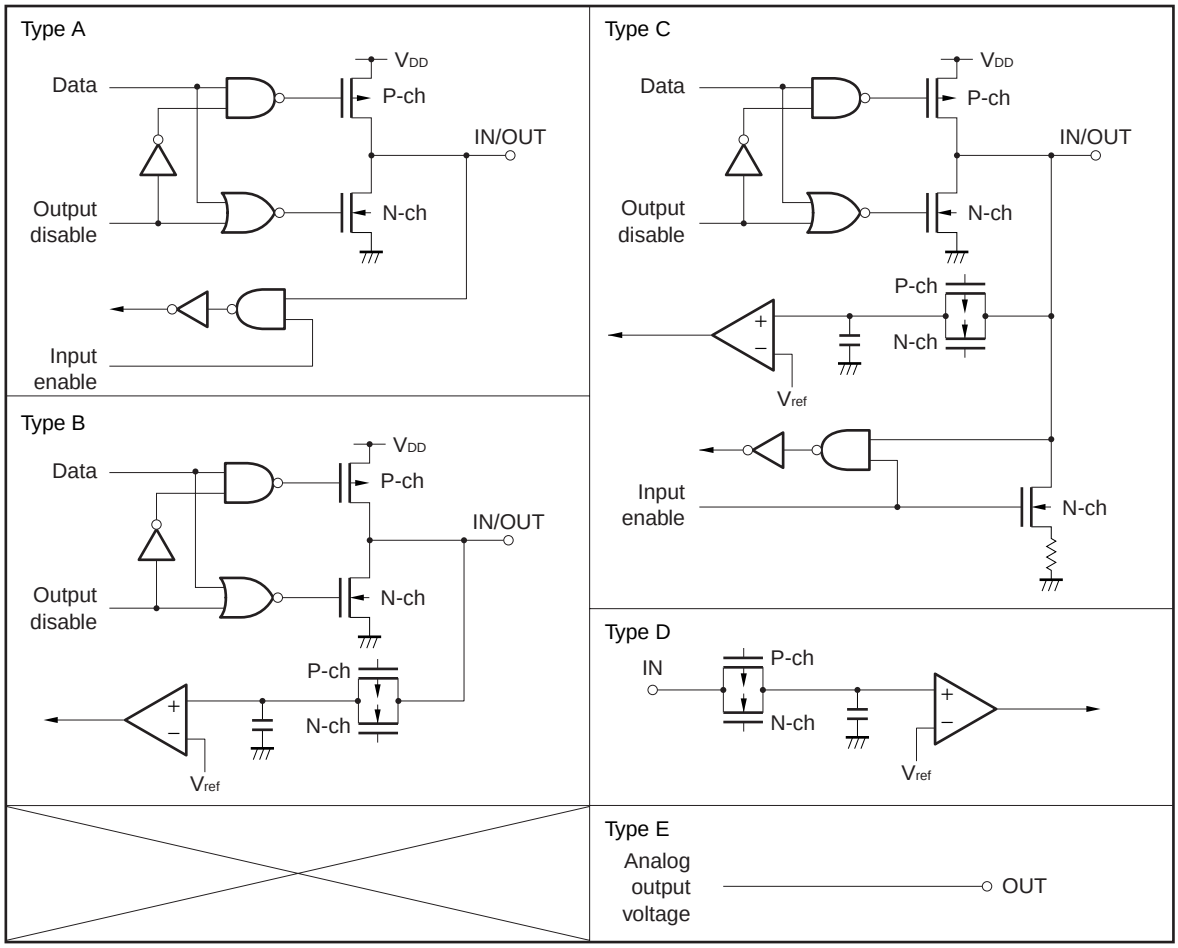
(2/3)

Pin Name	Recommended Connection When Not Used	I/O Circuit Type
ADIN(2:0)	Connect to GND_AD	D
AUDIOIN	Connect to GND_AD	D
AUDIOOUT	Leave open	E
CF_WE#/SCANOUT7	Leave open	A
CF_OE#/SCANOUT6	Leave open	A
CF_IOW#/SCANOUT5	Leave open	A
CF_IOR#/SCANOUT4	Leave open	A
CF_STSCHG#/SCANOUT3	Connect to VDD_IO via resistor	A
CF_CE(2:1)#/SCANOUT(2:1)	Leave open	A
CF_BUSY#/SCANOUT0	Connect to VDD_IO via resistor	A
CF_REG#/SCANIN7	Leave open	A
CF_RESET/SCANIN6	Leave open	A
CF_WAIT#/SCANIN5	Connect to VDD_IO via resistor	A
CF_IOIS16#/SCANIN4	Connect to VDD_IO via resistor	A
CF_VCCEN#/SCANIN3	Leave open	A
CF_DEN#/SCANIN2	Leave open	A
CF_DIR/SCANIN1	Leave open	A
CF_AEN#/SCANIN0	Leave open	A
RxD1/GPIO25	Connect to VDD_IO or GND_IO via resistor	A
TxD1/GPIO26/CLKSEL0	Connect to VDD_IO or GND_IO via resistor	A
RTS1#/GPIO27/CLKSEL1	Connect to VDD_IO or GND_IO via resistor	A
CTS1#/GPIO28	Connect to VDD_IO or GND_IO via resistor	A
DCD1#/GPIO29	Connect to VDD_IO or GND_IO via resistor	A
DTR1#/GPIO30/CLKSEL2	Connect to VDD_IO or GND_IO via resistor	A
DSR1#/GPIO31	Connect to VDD_IO or GND_IO via resistor	A
IRDIN/RxD2	Connect to VDD_IO or GND_IO via resistor	A
IRDOUT/TxD2	Leave open	A
GPIO(15:14)/FPD(7:6)/CD(2:1)#	Connect to VDD_IO or GND_IO via resistor	A
GPIO(13:12)/FPD(5:4)	Connect to VDD_IO or GND_IO via resistor	A
GPIO11/PCS1#	Connect to VDD_IO or GND_IO via resistor	A
GPIO10/FRM/SYSCLK	Connect to VDD_IO or GND_IO via resistor	A
GPIO9/CTS2#	Connect to VDD_IO or GND_IO via resistor	A
GPIO8/DSR2#	Connect to VDD_IO or GND_IO via resistor	A
GPIO7/DTR2#	Connect to VDD_IO or GND_IO via resistor	A
GPIO6/RTS2#	Connect to VDD_IO or GND_IO via resistor	A
GPIO5/DCD2#	Connect to VDD_IO or GND_IO via resistor	A
GPIO4	Connect to VDD_IO or GND_IO via resistor	A

(3/3)

Pin Name	Recommended Connection When Not Used	I/O Circuit Type
GPIO3/PCS0#	Connect to VDD_IO or GND_IO via resistor	A
GPIO2/SCK	Connect to VDD_IO or GND_IO via resistor	A
GPIO1/SO	Connect to VDD_IO or GND_IO via resistor	A
GPIO0/SI	Connect to VDD_IO or GND_IO via resistor	A
LEDOUT	Leave open	A

★ 2.5 Pin I/O Circuits



★ # CHAPTER 3 CP0 REGISTERS

3.1 Coprocessor 0 (CP0)

The Coprocessor 0 (CP0), which is also called as System Control Coprocessor, is implemented as an integral part of the CPU, and supports memory management, address translation, exception handling, and operation mode control.

Memory management, address translation, and operation mode control are provided by a block called memory management unit (MMU). The MMU contains a 32-entry TLB (translation lookaside buffer) that is used when translating virtual addresses to physical addresses.

The CP0 has registers shown in Table 3-1 that are used to set various modes for memory management and exception handling and to indicate statuses of the processor. Each CP0 register has a unique number that is used as an operand to specify a CP0 register to be accessed.

Caution When accessing the CP0 registers, some instructions require consideration of the interval time until the next instruction is executed, because there is a delay from when the contents of the CP0 register change to when this change is reflected in the CPU operation. This time lag is called a CP0 hazard. For details, refer to CHAPTER 23 COPROCESSOR 0 HAZARDS.

For details about functions of the CP0, refer to **VR4100 Series Architecture User's Manual**.

Table 3-1. CP0 Registers

Number	Register	Usage	Description
0	Index	Memory management	Programmable pointer to TLB array
1	Random	Memory management	Pseudo-random pointer to TLB array (read only)
2	EntryLo0	Memory management	Lower half of TLB entry for even VPN
3	EntryLo1	Memory management	Lower half of TLB entry for odd VPN
4	Context	Exception processing	Pointer to kernel virtual PTE in 32-bit mode
5	PageMask	Memory management	Page size specification
6	Wired	Memory management	Number of wired TLB entries
7	–	–	Reserved for future use
8	BadVAddr	Exception processing	Virtual address where the most recent error occurred
9	Count	Exception processing	Timer count
10	EntryHi	Memory management	Higher half of TLB entry (including ASID)
11	Compare	Exception processing	Timer compare value
12	Status	Exception processing	Status indication
13	Cause	Exception processing	Cause of last exception
14	EPC	Exception processing	Exception Program Counter
15	PRId	Memory management	Processor revision identifier
16	Config	Memory management	Configuration (memory system modes) specification
17	LLAddr Note1	Memory management	Physical address for self diagnostics
18	WatchLo	Exception processing	Memory reference trap address low bits
19	WatchHi	Exception processing	Memory reference trap address high bits
20	XContext	Exception processing	Pointer to kernel virtual PTE in 64-bit mode
21 to 25	–	–	Reserved for future use
26	Parity Error Note2	Exception processing	Cache parity bits
27	Cache Error Note2	Exception processing	Index and status of cache error
28	TagLo	Memory management	Lower half of cache tag
29	TagHi	Memory management	Higher half of cache tag
30	ErrorEPC	Exception processing	Error Exception Program Counter
31	–	–	Reserved for future use

Notes1. This register is defined to maintain compatibility with the VR4000 and VR4400. This register is meaningless during normal operations.

2. This register is defined to maintain compatibility with the VR4100. This register is not used in the VR4181 hardware.

3.2 Details of CP0 Registers

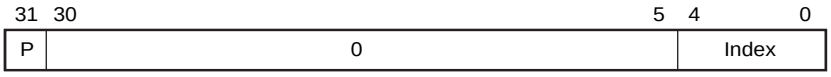
3.2.1 Index register (0)

The Index register is a 32-bit, read/write register containing five low-order 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 also specifies the TLB entry affected by TLB read (TLBR) or TLB write index (TLBWI) instructions.

★ The contents of the Index register are undefined after a reset so that it must be initialized by software.

Figure 3-1. Index Register



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: Reserved for future use. Write 0 in a write operation. When this field is read, 0 is read.

3.2.2 Random register (1)

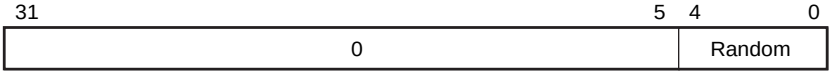
The Random register is a read-only register. The low-order 5 bits are used in referencing a TLB entry. This register is decremented each time an instruction is executed. The values that can be set in the register are as follows:

- The lower bound is the content of the Wired register.
- The upper bound is 31.

The Random register specifies the entry in the TLB that is affected by the TLBWR instruction. The register is readable to verify proper operation of the processor.

The Random register is set to the value of the upper bound upon Cold Reset. This register is also set to the upper bound when the Wired register is written. Figure 3-2 shows the format of the Random register.

Figure 3-2. Random Register



Random: TLB random index

0: Reserved for future use. Write 0 in a write operation. When this field is read, 0 is read.

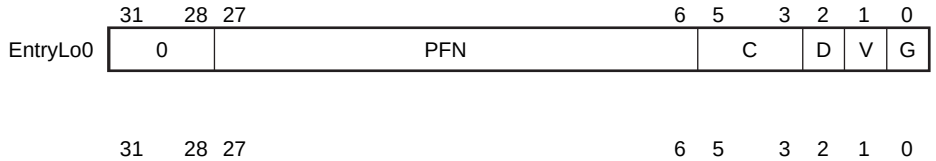
3.2.3 EntryLo0 (2) and EntryLo1 (3) registers

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 built-in TLB. When a TLB read/write operation is carried out, the EntryLo0 and EntryLo1 registers hold the contents of the low-order 32 bits of TLB entries at even and odd addresses, respectively.

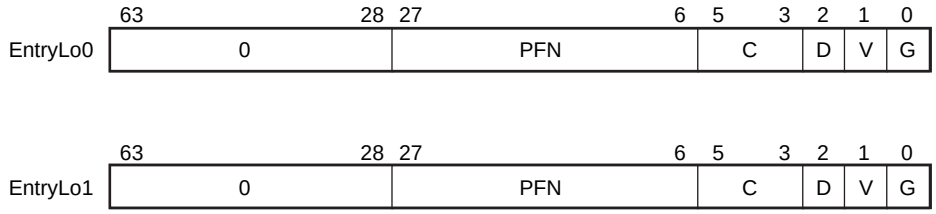
★ The contents of these registers are undefined after a reset so that they must be initialized by software.

Figure 3-3. EntryLo0 and EntryLo1 Registers

(a) 32-bit mode



(b) 64-bit mode



PFN:	Page frame number; high-order bits of the physical address.
C:	Specifies the TLB page attribute (see **Table 3-2**).
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; otherwise, 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:	Reserved for future use. Write 0 in a write operation. When this field is read, 0 is read.

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 3-2 lists the page attributes selected according to the value in the C bits.

Table 3-2. Cache Algorithm

C bit value	Cache algorithm
0	Cached
1	Cached
2	Uncached
3	Cached
4	Cached
5	Cached
6	Cached
7	Cached

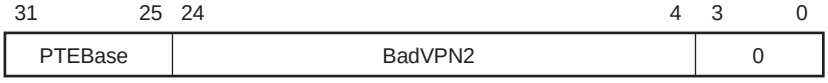
3.2.4 Context register (4)

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.

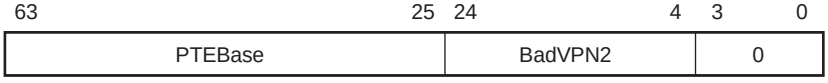
The Context register duplicates some of the information provided in the BadVAddr register, but the information is arranged in a form that is more useful for a software TLB exception handler.

Figure 3-4. Context Register

(a) 32-bit mode



(b) 64-bit mode



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: Reserved for future use. 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.

3.2.5 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 3-3. Five page sizes can be selected between 1 KB and 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 are undefined after a reset so that it must be initialized by software.

Figure 3-5. PageMask Register

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

0: Reserved for future use. Write 0 in a write operation. When this field is read, 0 is read.

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

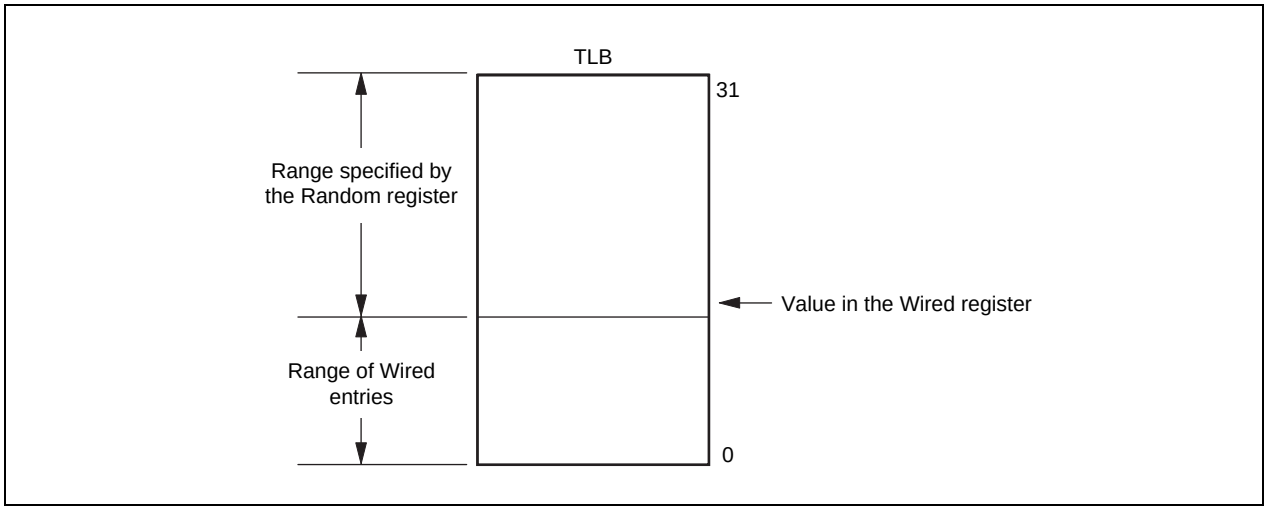
Table 3-3. Mask Values and Page Sizes

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

3.2.6 Wired register (6)

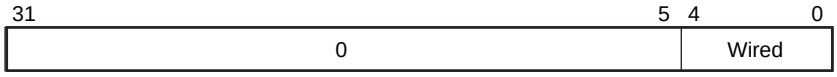
The Wired register is a read/write register that specifies the lower boundary of the random entry of the TLB as shown in Figure 3-6. Wired entries cannot be overwritten by a TLBWR instruction, but by a TLBWI instruction. Random entries can be overwritten by both instructions.

Figure 3-6. Positions Indicated by the Wired Register



The Wired register is set to 0 upon Cold Reset. Writing this register also sets the Random register to the value of its upper bound (see **3.2.2 Random register (1)**).

Figure 3-7. Wired Register



Wired: TLB wired boundary

0: Reserved for future use. Write 0 in a write operation. When this field is read, 0 is read.

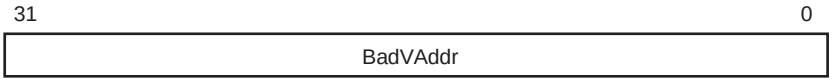
3.2.7 BadVAddr register (8)

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.

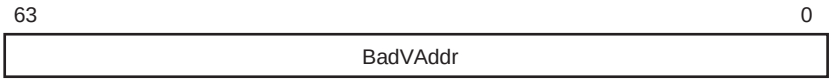
Caution This register saves no information after a bus error exception, because it is not an address error exception.

Figure 3-8. BadVAddr Register

(a) 32-bit mode



(b) 64-bit mode



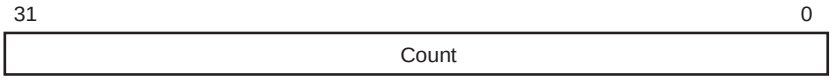
BadVAddr: Most recent virtual address for which an addressing error occurred, or for which address translation failed.

3.2.8 Count register (9)

The read/write Count register acts as a timer. It is incremented in synchronization with the MasterOut clock (1/8, 1/12, or 1/16 frequencies of the PClock), 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 self-diagnostic test, system initialization, or the establishment of inter-process synchronization.

Figure 3-9. Count Register



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

3.2.9 EntryHi register (10)

The EntryHi register is write-accessible. It is used to access the built-in TLB. The EntryHi register holds the high-order 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 high-order bit 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 **VR4100 Series Architecture User's Manual** for details of the TLB exception.

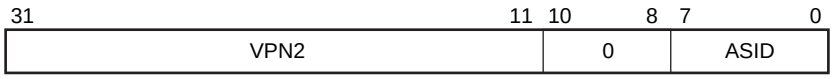
The ASID is used to read from or write to the ASID field of the TLB entry. It is also checked with the ASID of the TLB entry as the ASID of the virtual address during address translation.

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

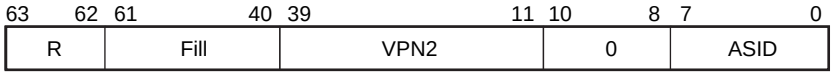
★ The contents of the EntryHi register are undefined after a reset so that it must be initialized by software.

Figure 3-10. EntryHi Register

(a) 32-bit mode



(b) 64-bit mode



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

ASID: Address space ID. An 8-bit ASID field that allows multiple processes to 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: Reserved. Ignored on write. When read, returns zero.

0: Reserved for future use. Write 0 in a write operation. When this field is read, 0 is read.

3.2.10 Compare register (11)

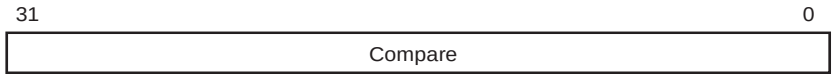
The Compare register causes a timer interrupt; it maintains a stable value that does not change on its own.

When the value of the Count register (see **3.2.8 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.

★ The contents of the Compare register are undefined after a reset.

Figure 3-11. Compare Register

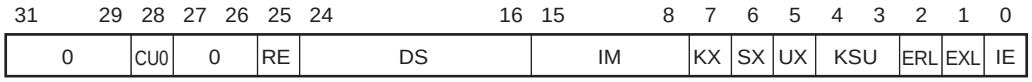


Compare: Value that is compared with the count value of the Count register.

3.2.11 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 3-12. Status Register (1/2)



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

RE: Enables/disables reversing of the endian setting in User mode (0 → Disabled, 1 → Enabled). This bit must be set to 0 since the VR4181 supports the little-endian order only.

DS: Diagnostic Status field (see **Figure 3-13**).

IM: Interrupt mask field used to enable/disable 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 VR4181.
- IM(1:0): Mask software interrupts.

Note Int(4:0) are internal signals of the VR4110 CPU core. For details about connection to the on-chip peripheral units, refer to **CHAPTER 9 INTERRUPT CONTROL UNIT (ICU)**.

Figure 3-12. Status Register (2/2)

KX:	Enables 64-bit addressing in Kernel mode (0 → 32-bit, 1 → 64-bit). 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).
UX:	Enables 64-bit addressing and operation in User mode (0 → 32-bit, 1 → 64-bit).
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:	Reserved for future use. Write 0 in a write operation. When this bit is read, 0 is read.

Figure 3-13 shows the details of the Diagnostic Status (DS) field. All DS field bits other than the TS bit are writable.

Figure 3-13. Status Register Diagnostic Status Field

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

BEV:	Specifies the base address of a TLB Refill exception vector and common exception vector (0 → Normal, 1 → Bootstrap).
TS:	Occurs the TLB to be shut down (read-only) (0 → Not shut down, 1 → Shut down). This bit is used to avoid any problems that may occur when multiple TLB entries match the same virtual address. After the TLB has been shut down, reset the processor to enable restart. Note that the TLB is shut down even if a TLB entry matching a virtual address is marked as being invalid (with the V bit cleared).
SR:	Occurs a Soft Reset or NMI exception (0 → Not occurred, 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 VR4181 hardware.
0:	Reserved for future use. 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 statuses are set.

(1) Interrupt enable

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

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

Access to the kernel address space is allowed when the processor is in Kernel mode.
Access to the supervisor address space is allowed when the processor is in Supervisor or Kernel mode.
Access to the user address space is allowed in any of the three operating modes.

(3) Addressing modes

The following Status register bit settings select 32- or 64-bit operation for each of 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. If this bit is set, an XTLB Refill exception occurs if a TLB miss occurs in the Kernel mode address space.
- 64-bit addressing and operations are enabled for Supervisor mode when SX bit = 1. If this bit is set, an XTLB Refill exception occurs if a TLB miss occurs in the Supervisor mode address space.
- 64-bit addressing and operations are enabled for User mode when UX bit = 1. If this bit is set, an XTLB Refill exception occurs if a TLB miss occurs in the User mode address space.

(4) Status after reset

The contents of the Status register are undefined after Cold Resets, except for the following bits in the Diagnostic Status field.

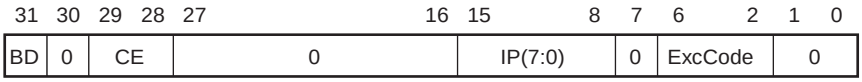
- TS and SR bits are cleared to 0.
  SR bit is 0 after Cold Reset, and is 1 after Soft Reset or NMI.
- ERL and BEV bits are set to 1.

Remark Cold Reset and Soft Reset are resets for the CPU core (see **5.3 Reset of CPU Core**). For the reset of all the VR4181 including peripheral units, refer to **CHAPTER 5 INITIALIZATION INTERFACE** and **CHAPTER 10 POWER MANAGEMENT UNIT (PMU)**.

3.2.12 Cause register (13)

The 32-bit read/write Cause register holds the cause of the most recent exception. A 5-bit exception code indicates one of the causes (see **Table 3-4**). 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 bits are used for software interrupts.

Figure 3-14. Cause Register



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 VR4181.
- IP(1:0): Software interrupts. Only these bits cause an interrupt exception, when they are set to 1 by means of software.

Note Int(4:0) are internal signals of the VR4110 CPU core. For details about connection to the on-chip peripheral units, refer to **CHAPTER 9 INTERRUPT CONTROL UNIT (ICU)**.

ExcCode: Exception code field (refer to **Table 3-4** for details).

0: Reserved for future use. Write 0 in a write operation. When this field is read, 0 is read.

Table 3-4. 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	—	Reserved for future use
23	WATCH	Watch exception
24 to 31	—	Reserved for future use

The VR4181 has eight interrupt request sources, IP7 to IP0. They are used for the purpose as follows.
For the detailed description of interrupts of the CPU core, refer to **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 the Count register and Compare register match.

(2) IP6 to IP2

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

(3) IP1 and IP0

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

3.2.13 Exception Program Counter (EPC) register (14)

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

When the MIPS16 instruction execution is disabled, either of the following addresses is contained in the EPC register:

- Virtual address of the instruction that caused the exception
- Virtual address of the immediately preceding branch or jump instruction (when the instruction associated with the exception is in a branch delay slot, and the BD bit in the Cause register is set to 1)

When the MIPS16 instruction execution is enabled, either of the following addresses is contained in the EPC register during a 32-bit instruction execution:

- Virtual address of the instruction that caused the exception and ISA mode at which an exception occurs
- Virtual address of the immediately preceding branch or jump instruction and ISA mode at which an exception occurs (when the instruction associated with the exception is in a branch delay slot of the jump instruction, and the BD bit in the Cause register is set to 1)

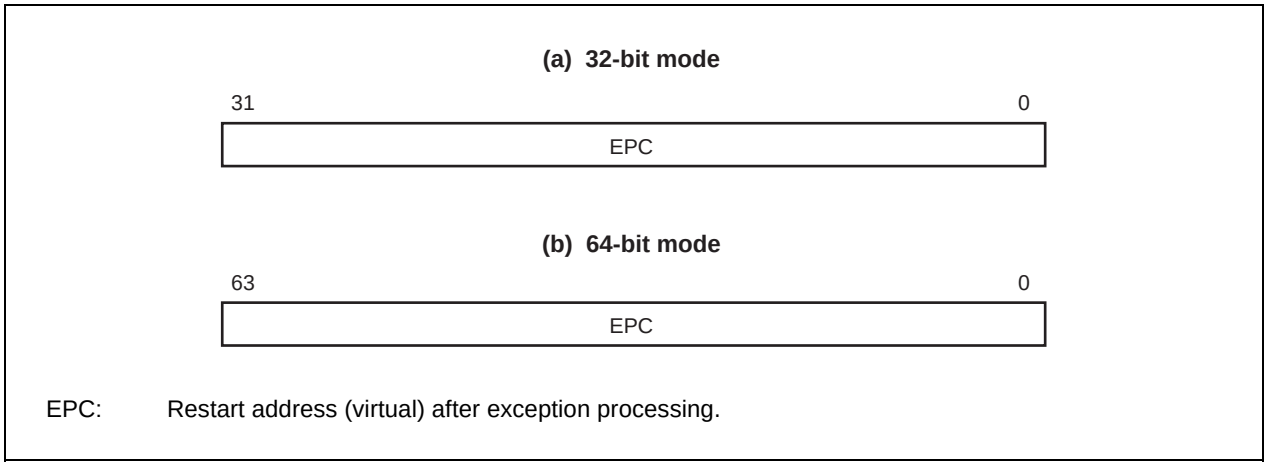
When the 16-bit instruction is executed, either of the following addresses is contained in the EPC register:

- Virtual address of the instruction that caused the exception and ISA mode at which an exception occurs
- Virtual address of the immediately preceding Extend or jump instruction and ISA mode at which an exception occurs (when the instruction associated with the exception is in a branch delay slot of the jump instruction or in the instruction following the Extend instruction, and the BD bit in the Cause register is set to 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.

★ The EPC register never indicates the address of the instruction in a branch delay slot.

Figure 3-15. EPC Register (When MIPS16 ISA Is Disabled)



EPC: Restart address (virtual) after exception processing.

Figure 3-16. EPC Register (When MIPS16 ISA Is Enabled)

(a) 32-bit mode

EPC: Bits 31 to 1 of restart address (virtual) after exception processing.

EIM: ISA mode at which an exception occurs (1 → When MIPS16 SIA instruction is executed, 0 → When MIPS III ISA instruction is executed).

(b) 64-bit mode

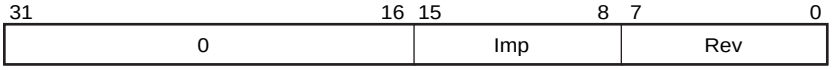
EPC: Bits 63 to 1 of restart address (virtual) after exception processing.

EIM: ISA mode at which an exception occurs (1 → When MIPS16 SIA instruction is executed, 0 → When MIPS III ISA instruction is executed).

3.2.14 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 3-17. PRId Register



Imp: CPU core processor ID number (0x0C for the VR4181)

Rev: CPU core processor revision number

0: Reserved for future use. Write 0 in a write operation. When this field is read, 0 is read.

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 CPU core revisions of the VR4181, 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 field.

3.2.15 Config register (16)

The Config register specifies various configuration options selected on the VR4181.

Some configuration options, as defined by the EC and BE fields, are set by the hardware during Cold Reset and are included in the Config register as read-only status bits for the software to access. Other configuration options (AD, EP, and K0 fields) can be read/written and controlled by software; on Cold Reset these fields are undefined. Since only a subset of the VR4000 Series™ options are available in the VR4181, some bits are set to constants (e.g., bits 14 and 13) that were variable in the VR4000 Series. The Config register should be initialized by software before caches are used.

★ The contents of the Config register are undefined after a reset so that it must be initialized by software.

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

Figure 3-18. Config Register (1/2)

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

EC: System clock ratio (read only)
- 0 → Processor clock frequency divided by 2
- 1 → Processor clock frequency divided by 3
- 2 → Processor clock frequency divided by 4
- 3 to 7 → Reserved

EP: Transfer data pattern (cache write-back pattern) setting
- 0 → DD: 1 word per 1 cycle
- Others → Reserved

AD: Accelerate data mode
- 0 → VR4000 Series compatible mode
- 1 → Reserved

M16: MIPS16 ISA mode enable/disable indication (read only)
- 0 → MIPS16 instruction cannot be executed
- 1 → MIPS16 instruction can be executed.

BE: BigEndianMem (Endian mode indication)
- 0 → Little endian
- 1 → Reserved

CS: Cache size mode indication (n = IC, DC)
- 0 → Reserved
- 1 → $2^{(n+10)}$ bytes

IC: Instruction cache size indication. $2^{(IC+10)}$ bytes in the VR4181.
- 2 → 4 KB
- Others → Reserved

DC: Data cache size indication. $2^{(DC+10)}$ bytes in the VR4181.
- 2 → 4 KB
- Others → Reserved

Figure 3-18. Config Register (2/2)

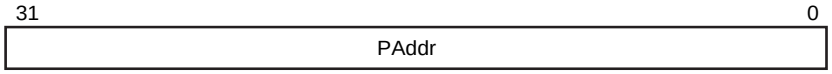
K0:	kseg0 cache coherency algorithm 2 → Uncached Others → Cached
1:	1 is returned when read.
0:	0 is returned when read.

3.2.16 Load Linked Address (LLAddr) register (17)

The read/write Load Linked Address (LLAddr) register is not used with the VR4181 processor except for diagnostic purpose, and serves no function during normal operation. The LLAddr register is implemented just for compatibility between the VR4181 and VR4000 or VR4400.

★ The contents of the LLAddr register are undefined after a reset.

Figure 3-19. LLAddr Register



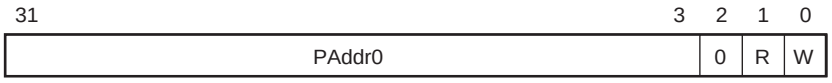
PAddr: 32-bit physical address

3.2.17 WatchLo (18) and WatchHi (19) registers

The VR4181 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.

★ The contents of these registers are undefined after a reset so that they must be initialized by software.

Figure 3-20. WatchLo Register



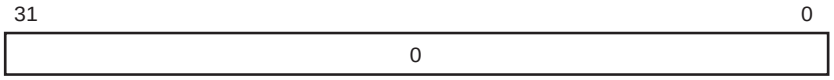
PAddr0: Specifies physical address bits 31 to 3.

★ R: Specifies detection of watch address references when load instructions are executed (1 → Detect, 0 → Not detect).

★ W: Specifies detection of watch address references when store instructions are executed (1 → Detect, 0 → Not detect).

0: Reserved for future use. Write 0 in a write operation. When this field is read, 0 is read.

Figure 3-21. WatchHi Register



0: Reserved for future use. Write 0 in a write operation. When this field is read, 0 is read.

3.2.18 XContext register (20)

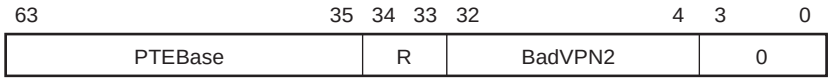
The read/write XContext register contains a pointer to an entry in the page table entry (PTE) array, an operating system data structure that stores virtual-to-physical address translations. If a TLB miss occurs, the operating system loads the untranslated data from the PTE into the TLB to handle the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

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 3-22. XContext Register



PTEBase: Base address of the PTE entry table.

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

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

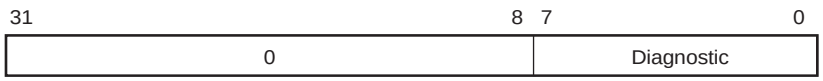
0: Reserved for future use. Write 0 in a write operation. When this field is read, 0 is read.

The 29-bit BadVPN2 field has bits 39 to 11 of the virtual address that caused the TLB 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 may be used directly to address the pair-table of 8-byte PTEs. When the page size is 4 KB or more, shifting or masking this value produces the appropriate PTE reference address.

3.2.19 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 VR4181 has no parity.

Figure 3-23. Parity Error Register



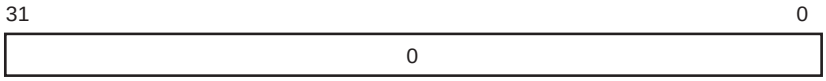
Diagnostic: 8-bit self diagnostic field.

0: Reserved for future use. Write 0 in a write operation. When this field is read, 0 is read.

3.2.20 Cache Error register (27)

The Cache Error 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 VR4181 has no parity.

Figure 3-24. Cache Error Register



0: Reserved for future use. Write 0 in a write operation. When this field is read, 0 is read.

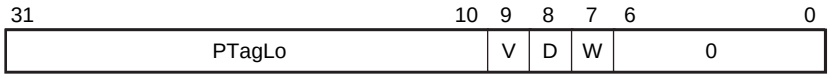
3.2.21 TagLo (28) and TagHi (29) registers

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

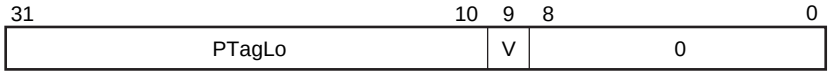
★ The contents of these registers are undefined after a reset.

Figure 3-25. TagLo Register

(a) When used with data cache



(b) When used with instruction cache



PTagLo: Specifies physical address bits 31 to 10.

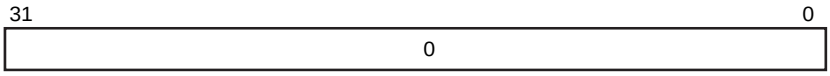
V: Valid bit

D: Dirty bit. However, this bit is defined only for the compatibility 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.

W: Writeback bit (set if cache line has been updated)

0: Reserved for future use. Write 0 in a write operation. When this field is read, 0 is read.

Figure 3-26. TagHi Register



0: Reserved for future use. Write 0 in a write operation. When this field is read, 0 is read.

3.2.22 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 after RTC reset specifies whether the execution of MIPS16 instructions is enabled or disabled.

When the MIPS16 instruction execution is disabled, either of the following addresses is contained in the ErrorEPC register:

- Virtual address of the instruction that caused the exception
- Virtual address of the immediately preceding branch or jump instruction, when the instruction associated with ★ the error exception is in a branch delay slot, and the BD bit in the Cause register is set to 1

When the MIPS16 instruction execution is enabled, either of the following addresses is contained in the ErrorEPC register during a 32-bit instruction execution:

- Virtual address of the instruction that caused the exception and ISA mode at which an exception occurs
- Virtual address of the immediately preceding branch or jump instruction and ISA mode at which an exception ★ occurs when the instruction associated with the error exception is in a branch delay slot, and the BD bit in the Cause register is set to 1

When the 16-bit instruction is executed, either of the following addresses is contained in the ErrorEPC register:

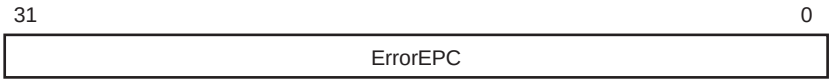
- Virtual address of the instruction that caused the exception and ISA mode at which an exception occurs
- Virtual address of the immediately preceding jump instruction or Extend instruction and ISA mode at which an exception occurs when the instruction associated with the error exception is in a branch delay slot of the jump ★ instruction or is the instruction following the Extend instruction, and the BD bit in the Cause register is set to 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 that causes an error exception.

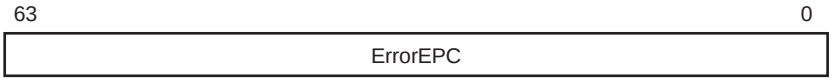
The ErrorEPC register never indicates the address of the instruction in a branch delay slot.

Figure 3-27. ErrorEPC Register (When MIPS16 ISA Is Disabled)

(a) 32-bit mode



(b) 64-bit mode



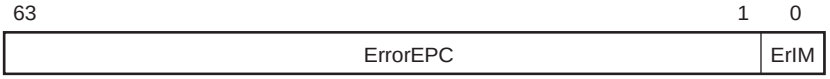
ErrorEPC: Virtual restart address after Cold reset, Soft reset, or NMI exception.

Figure 3-28. ErrorEPC Register (When MIPS16 ISA Is Enabled)

(a) 32-bit mode

ErrorEPC: Bits 31 to 1 of virtual restart address after Cold reset, Soft reset, or NMI exception.
ErIM: ISA mode at which an error exception occurs (1 → MIPS16 ISA, 0 → MIPS III ISA).

(b) 64-bit mode



ErrorEPC: Bits 63 to 1 of virtual restart address after Cold reset, Soft reset, or NMI exception.
ErIM: ISA mode at which an error exception occurs (1 → MIPS16 ISA, 0 → MIPS III ISA).

★

CHAPTER 4 MEMORY MANAGEMENT SYSTEM

4.1 Overview

The VR4181 provides a memory management unit (MMU) which uses a translation lookaside buffer (TLB) to translate virtual addresses into physical addresses.

Virtual addresses are translated into physical addresses using an on-chip TLB. The on-chip TLB is a full-associative memory that holds 32 entries, which provide mapping to 32 odd/even page pairs for one entry. The TLB is accessed through the CP0 registers. Note that the virtual address space includes areas that are translated to physical addresses without using a TLB, and areas where the use of cache memory can be selected.

The VR4181 has three operating modes: User, Supervisor, and Kernel; the manner in which memory addresses are mapped depends on these operating modes. In addition, the VR4181 supports the 32-bit and 64-bit addressing modes; the manner in which memory addresses are translated or mapped depends on these addressing modes.

For details about the memory management system and virtual address space, refer to **VR4100 Series Architecture User's Manual**.

4.2 Physical Address Space

Using a 32-bit address, the processor physical address space encompasses 4 GB. The VR4181 uses this 4 GB physical address space as shown in Figure 4-1.

Figure 4-1. VR4181 Physical Address Space

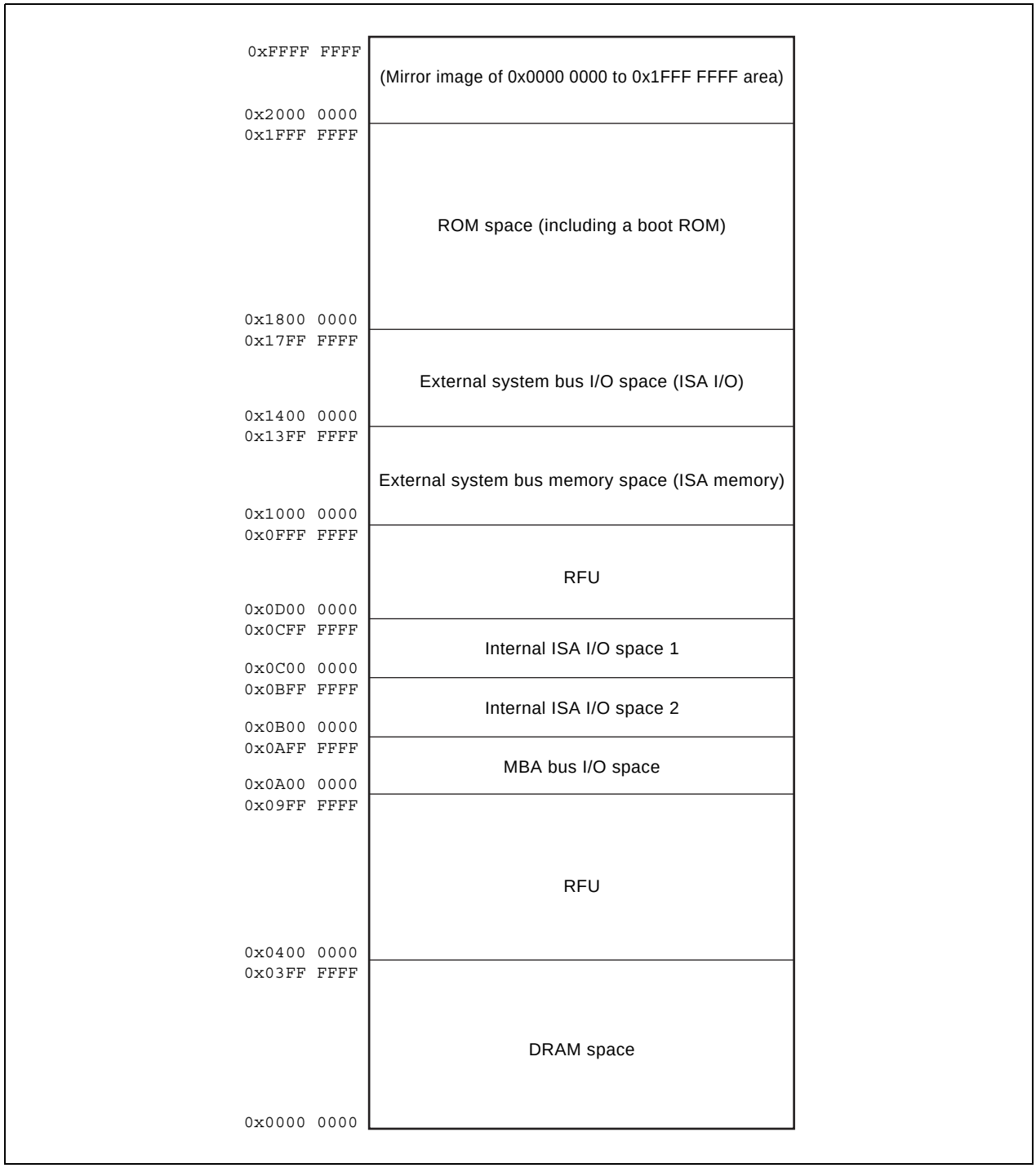


Table 4-1. VR4181 Physical Address Space

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 0x1400 0000	External system bus I/O space (ISA I/O)	64 M
0x13FF FFFF to 0x1000 0000	External system bus memory space (ISA memory)	64 M
0x0FFF FFFF to 0x0D00 0000	Space reserved for future use	48 M
0x0CFF FFFF to 0x0C00 0000	Internal ISA I/O space 1	16 M
0x0BFF FFFF to 0x0B00 0000	Internal ISA I/O space 2	16 M
0x0AFF FFFF to 0x0A00 0000	MBA bus I/O space	16 M
0x09FF FFFF to 0x0400 0000	Space reserved for future use	96 M
0x03FF FFFF to 0x0000 0000	DRAM (SDRAM) space	64 M

4.2.1 ROM space

The ROM space mapping differs depending on the capacity of the ROM being used. The ROM capacity is set via the ROMs(1:0) bits in the BCUCNTREG1 register.

The physical addresses of the ROM space are listed below.

Table 4-2. ROM Address Map

Physical address	When using 32-Mbit ROM	When using 64-Mbit ROM
0x1FFF FFFF to 0x1FC0 0000	Bank 3 (ROMCS3#)	Bank 3 (ROMCS3#)
0x1FBF FFFF to 0x1F80 0000	Bank 2 (ROMCS2#)	
0x1F7F FFFF to 0x1F40 0000	Bank 1 (ROMCS1#)	Bank 2 (ROMCS2#)
0x1F3F FFFF to 0x1F00 0000	Bank 0 (ROMCS0#)	
0x1EFF FFFF to 0x1E80 0000	Reserved for future use	Bank 1 (ROMCS1#)
0x1E7F FFFF to 0x1E00 0000		Bank 0 (ROMCS0#)

4.2.2 External system bus space

The following two types of system bus space are available.

- External system bus I/O space
  This corresponds to the ISA's I/O space.

- External system bus memory space
  This corresponds to the ISA's memory space.

4.2.3 Internal I/O space

The VR4181 has three internal I/O spaces. Each of these spaces is described below.

Table 4-3. Internal I/O Space 1

Physical address	Internal I/O
0x0C00 001F to 0x0C00 0010	SIU1
0x0C00 000F to 0x0C00 0000	SIU2

Table 4-4. Internal I/O Space 2

Physical address	Internal I/O
0x0B00 09FF to 0x0B00 0900	CSI
0x0B00 08FF to 0x0B00 0800	ECU
0x0B00 07FF to 0x0B00 0400	Reserved for future use
0x0B00 03FF to 0x0B00 0300	GIU
0x0B00 02FF to 0x0B00 02D0	Reserved for future use
0x0B00 02CF to 0x0B00 02C0	ISA Bridge
0x0B00 02BF to 0x0B00 02A0	PIU-2
0x0B00 029F to 0x0B00 0280	Reserved for future use
0x0B00 027F to 0x0B00 0260	A/D test
0x0B00 025F to 0x0B00 0240	LED
0x0B00 023F to 0x0B00 01E0	Reserved for future use
0x0B00 01DF to 0x0B00 01C0	RTC-2
0x0B00 01BF to 0x0B00 01A0	Reserved for future use
0x0B00 019F to 0x0B00 0180	KIU
0x0B00 017F to 0x0B00 0160	AIU
0x0B00 015F to 0x0B00 0140	Reserved for future use
0x0B00 013F to 0x0B00 0120	PIU-1
0x0B00 011F to 0x0B00 0100	Reserved for future use
0x0B00 00FF to 0x0B00 00E0	DSU
0x0B00 00DF to 0x0B00 00C0	RTC-1
0x0B00 00BF to 0x0B00 00A0	PMU
0x0B00 009F to 0x0B00 0080	ICU-3
0x0B00 007F to 0x0B00 0000	Reserved for future use

Table 4-5. MBA Bus I/O Space

Physical address	Internal I/O
0x0A00 06FF to 0x0A00 0600	DCU-2
0x0A00 05FF to 0x0A00 0500	Reserved for future use
0x0A00 04FF to 0x0A00 0400	LCD controller
0x0A00 03FF to 0x0A00 0300	Memory controller
0x0A00 02FF to 0x0A00 0220	Reserved for future use
0x0A00 021F to 0x0A00 0200	ICU-2
0x0A00 01FF to 0x0A00 00A0	Reserved for future use
0x0A00 009F to 0x0A00 0080	ICU-1
0x0A00 007F to 0x0A00 0050	Reserved for future use
0x0A00 004F to 0x0A00 0020	DCU-1
0x0A00 001F to 0x0A00 0000	MBA Host Bridge

4.2.4 DRAM space

The DRAM space differs depending on the capacity of the DRAM being used. The DRAM capacity is set via the B1Config(1:0) bits in the MEMCFG_REG register.

The physical addresses of the DRAM space are listed below.

★ **Table 4-6. DRAM Address Map**

Physical address	When using 16-Mbit DRAM	When using 64-Mbit DRAM
0x00FF FFFF to 0x0080 0000	Reserved for future use	Bank 1 (SDCS1#/RAS1#)
0x007F FFFF to 0x0040 0000		Bank 0 (SDCS0#/RAS0#)
0x003F FFFF to 0x0020 0000	Bank 1 (SDCS1#/RAS1#)	
0x001F FFFF to 0x0000 0000	Bank 0 (SDCS0#/RAS0#)	

CHAPTER 5 INITIALIZATION INTERFACE

This chapter describes 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.

A detailed description of the operation during and after a reset and its relationships to the power modes are also provided in **CHAPTER 10 POWER MANAGEMENT UNIT (PMU)**.

Remark # that follows signal names indicates active low.

5.1 Reset Function

There are five ways to reset the VR4181. Each is summarized below.

5.1.1 RTC reset

During power-on, set the RTCRST# pin as active. After waiting about 600 ms for the 32.768 kHz oscillator to begin oscillating when the power supply is stable at 3.0 V or above, setting the RTCRST# pin as inactive causes the RTC unit to begin counting. Then, the states of the MIPS16EN and CLKSEL(2:0) pins are read after one RTC cycle. Next,

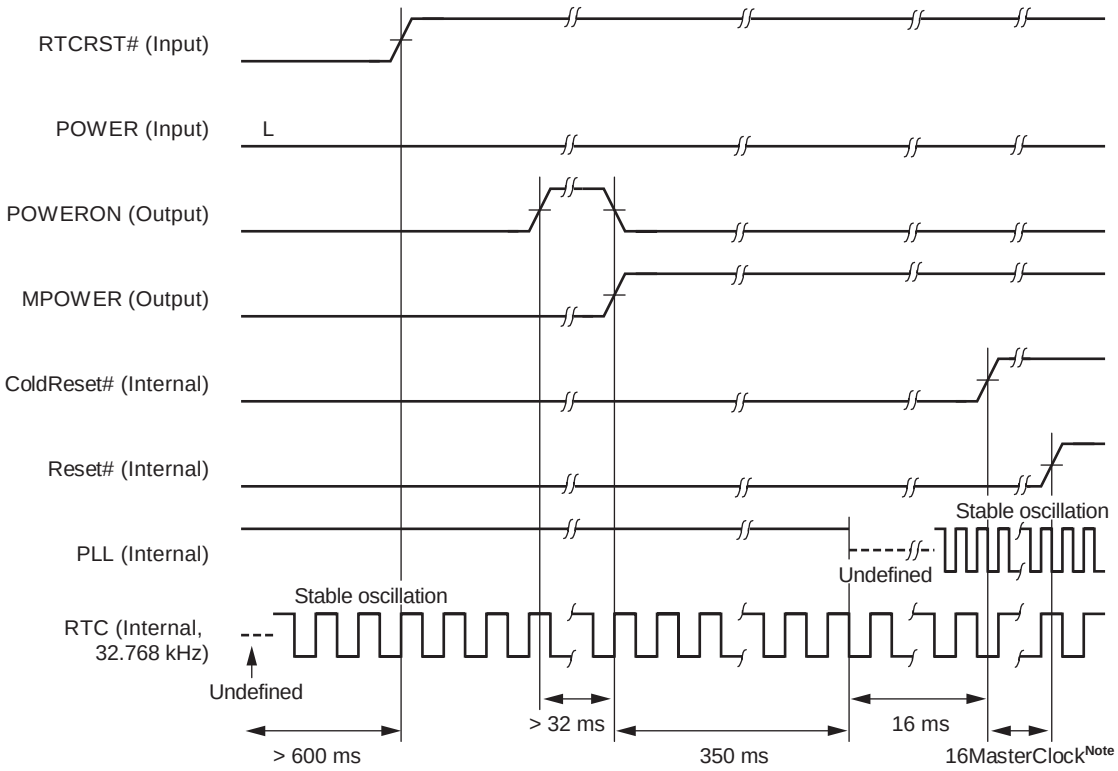
★ the VR4181 asserts the POWERON pin and checks the state of the BATTINH/BATTINT# signal. If it is at high level, the VR4181 asserts the MPOWER pin and activates the external agent's DC/DC converter. After the stabilization time period (about 350 ms) of the DC/DC converter, the VR4181 begins PLL oscillation and starts all clocks (a period of about 16 ms following the start of PLL oscillation is required for stabilization of PLL oscillation).

An RTC reset 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.

★ After a reset, the processor becomes the system bus master, which executes a Cold Reset exception sequence and begins to access the reset exception vectors in the ROM space. Since only part of the internal status is reset when a reset occurs in the VR4181, the processor should be completely initialized by software (see **5.4 Notes on Initialization**).

After power-on, the processor's pin statuses are undefined since the RTCRST# is asserted, until the 32.768 kHz clock oscillator starts oscillation. The pin statuses after oscillation starts are described in **CHAPTER 2 PIN FUNCTIONS** in this document.

Figure 5-1. RTC Reset



★ **Note** MasterClock is the basic clock used in the CPU core. Its frequency is one forth of TClock frequency.

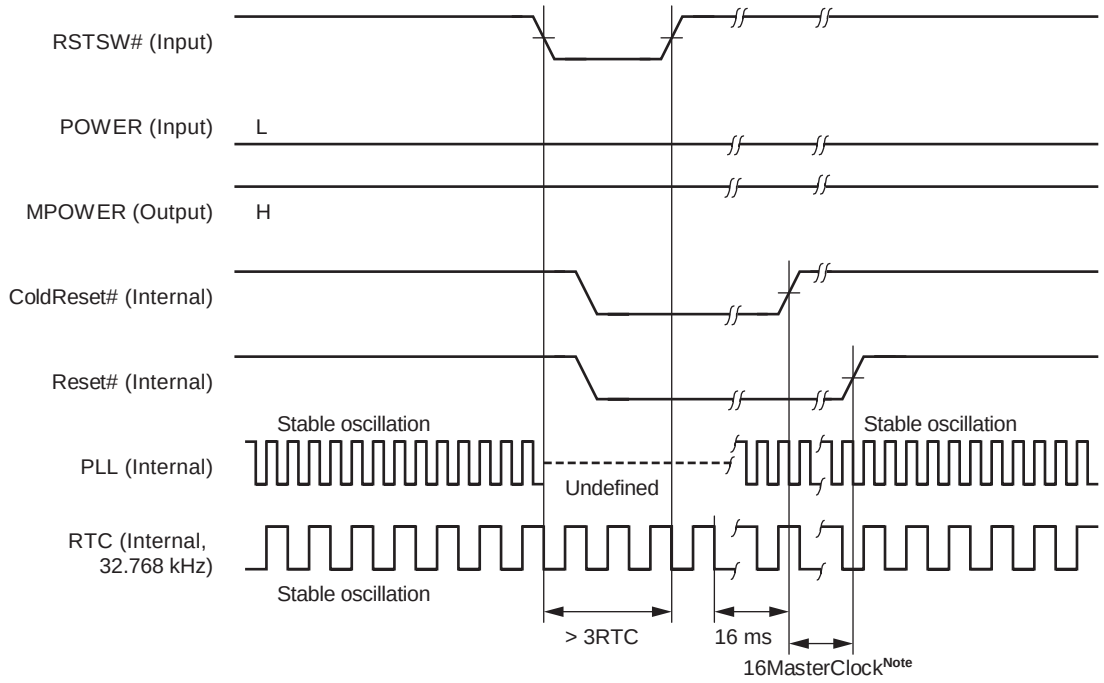
5.1.2 RSTSW reset

After the RSTSW# pin becomes active and then becomes inactive 100 μs later, the VR4181 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).

★ An RSTSW reset basically initializes the entire internal state except for the RTC timer, the GIU, and the PMU. The VR4181 has function to preserve DRAM data during RSTSW reset. For detail, refer to **CHAPTER 10 POWER MANAGEMENT UNIT (PMU)**.

★ After a reset, the processor becomes the system bus master, which executes a Cold Reset exception sequence and begins to access the reset exception vectors in the ROM space. Since only part of the internal status is reset when a reset occurs in the VR4181, the processor should be completely initialized by software (see **5.4 Notes on Initialization**).

Figure 5-2. RSTSW Reset



★ **Note** MasterClock is the basic clock used in the CPU core. Its frequency is one forth of TClock frequency.

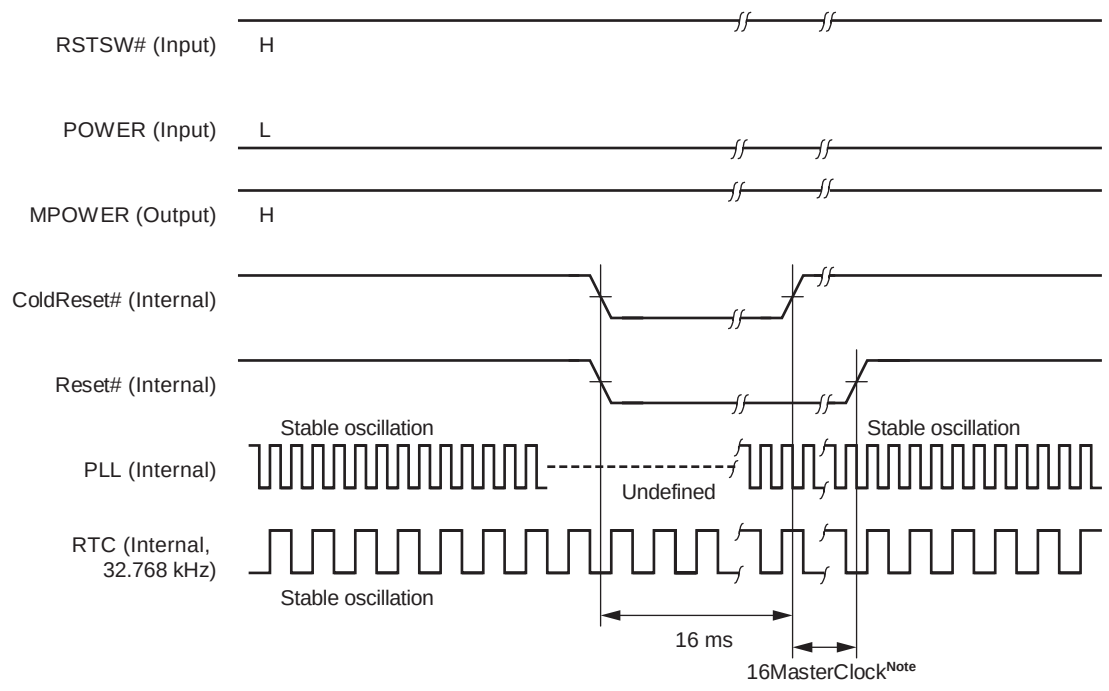
5.1.3 Deadman's Switch reset

After the Deadman's Switch unit is enabled, if the Deadman's Switch is not cleared within the specified time period, the VR4181 immediately enters to reset status. Setting and clearing of the Deadman's Switch is performed by software.

★ A Deadman's Switch reset initializes the entire internal state except for the RTC timer, the GIU, and the PMU. Since the DRAM is not switched to self-refresh mode, the contents of DRAM after a Deadman's Switch reset are not at all guaranteed.

★ After a reset, the processor becomes the system bus master, which executes a Cold Reset exception sequence and begins to access the reset vectors in the ROM space. Since only part of the internal status is reset when a reset occurs in the VR4181, the processor should be completely initialized by software (see **5.4 Notes on Initialization**).

Figure 5-3. Deadman's Switch Reset



★ **Note** MasterClock is the basic clock used in the CPU core. Its frequency is one forth of TClock frequency.

5.1.4 Software shutdown

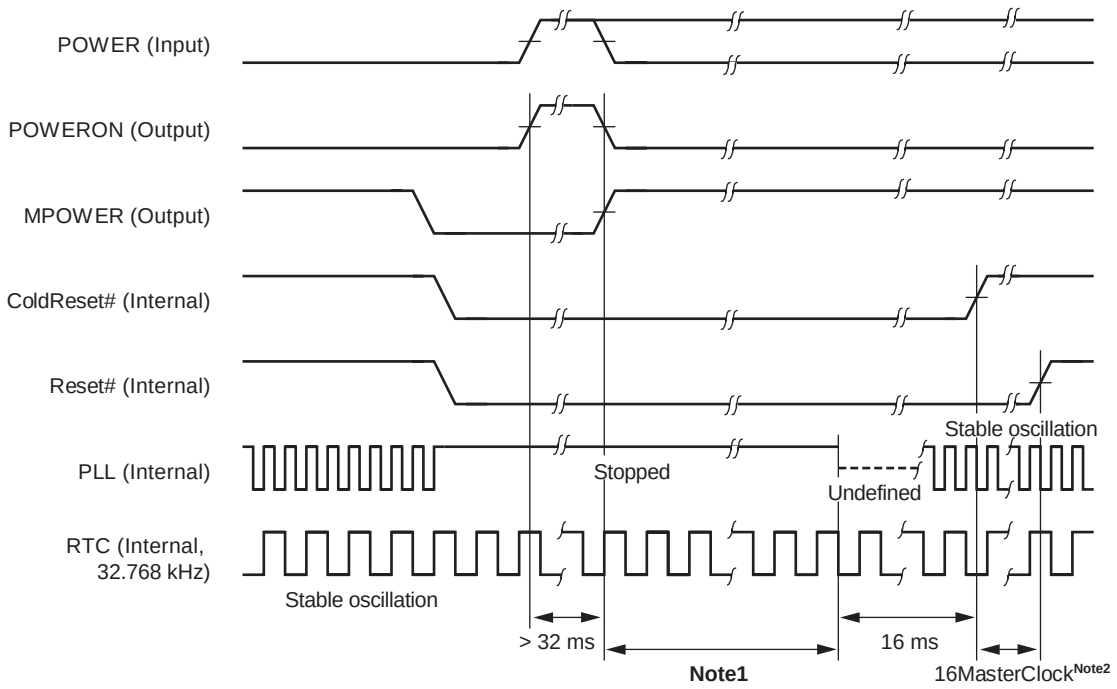
When the software executes the HIBERNATE instruction, the VR4181 sets the MPOWER pin as inactive, then enters reset status. Recovery from reset status occurs when the POWER pin or DCD# signal is asserted or when an unmasked wake-up interrupt request is occurred.

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

★ After a reset, the processor becomes the system bus master, which executes a Cold Reset exception sequence and begins to access the reset vectors in the ROM space. Since only part of the internal status is reset when a reset occurs in the VR4181, the processor should be completely initialized by software (see **5.4 Notes on Initialization**).

Cauiton The VR4181 does not set the DRAM to self-refresh mode at the transition to Hibernate mode from Fullspeed mode. To preserve DRAM data, software must set the DRAM to self-refresh mode. For details, refer to CHAPTER 10 POWER MANAGEMENT UNIT (PMU).

Figure 5-4. Software Shutdown



Notes 1. Wait time for activation. It can be changed by setting the PMUWAITREG register.

★ **2.** MasterClock is the basic clock used in the CPU core. Its frequency is one forth of TClock frequency.

5.1.5 HALTimer shutdown

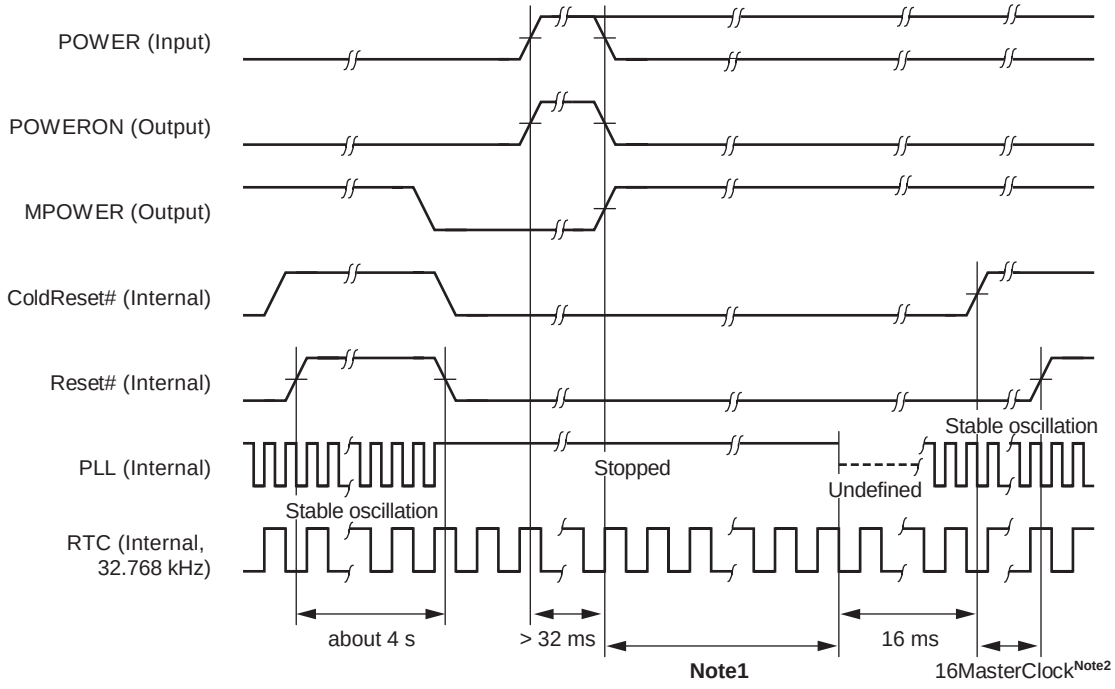
★ After an RTC reset or RSTSW reset is canceled, if the HALTimer is not canceled (the HALTIMERRST bit of the PMUCNTREG register is not set) by software within about four seconds, the VR4181 enters reset status. Recovery from reset status occurs when the POWER pin is asserted or when a ElapsedTime interrupt request occurs.

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

★ After a reset, the processor becomes the system bus master, which executes a Cold Reset exception sequence and begins to access the reset vectors in the ROM space. Since only part of the internal status is reset when a reset occurs in the VR4181, the processor should be completely initialized by software (see **5.4 Notes on Initialization**).

Caution The VR4181 does not sets the DRAM to self-refresh mode by HALTimer shutdown. Therefore, the contents of DRAM after a HALTimer shutdown are not at all guaranteed.

Figure 5-5. HALTimer Shutdown



Notes 1. Wait time for activation. It can be changed by setting the PMUWAITREG register.

★ **2.** MasterClock is the basic clock used in the CPU core. Its frequency is one forth of TClock frequency.

5.2 Power-on Sequence

The factors that cause the VR4181 to switch from Hibernate mode or shutdown mode to Fullspeed mode are ★ called activation factors. There are five activation factors: assertion of the POWER pin, the DCD1# pin or the GPIO(15:0) pins, or activation of the ElapsedTime or CompactFlash interrupt request. When an activation factor occurs, the VR4181 asserts the POWERON pin to notify to external agents that the VR4181 is ready for power-on. Three RTC clock cycles after the POWERON pin is asserted, the VR4181 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 VR4181 is not activated. If the BATTINH/BATTINT# pin's state is high, the POWERON pin is deasserted and the MPOWER pin is asserted three RTC clocks after the BATTINH/BATTINT# pin check is completed, then the VR4181 is activated.

Figure 5-6 shows a timing chart of VR4181 activation and Figure 5-7 shows a timing chart of when activation fails due to the BATTINH/BATTINT# pin's "low" state.

Remark While the MPOWER pin is inactive, 2.5 V power supply of the VR4181 (VDD_LOGIC, VDD_PLL) is not needed. In order to reduce leak current, it is recommended to turn on/off the 2.5 V power supply of the VR4181 according to MPOWER pin state.

Figure 5-6. VR4181 Activation Sequence (When Activation Is OK)

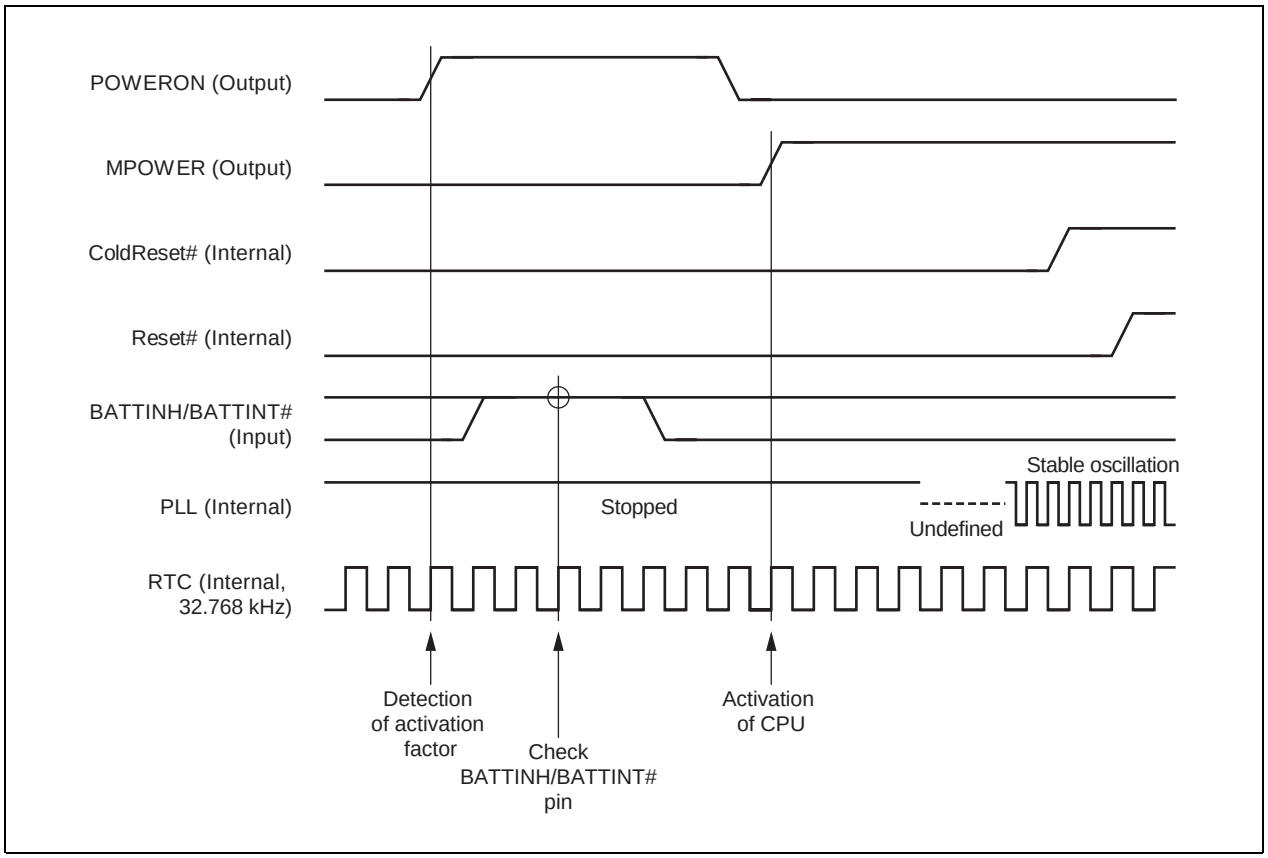
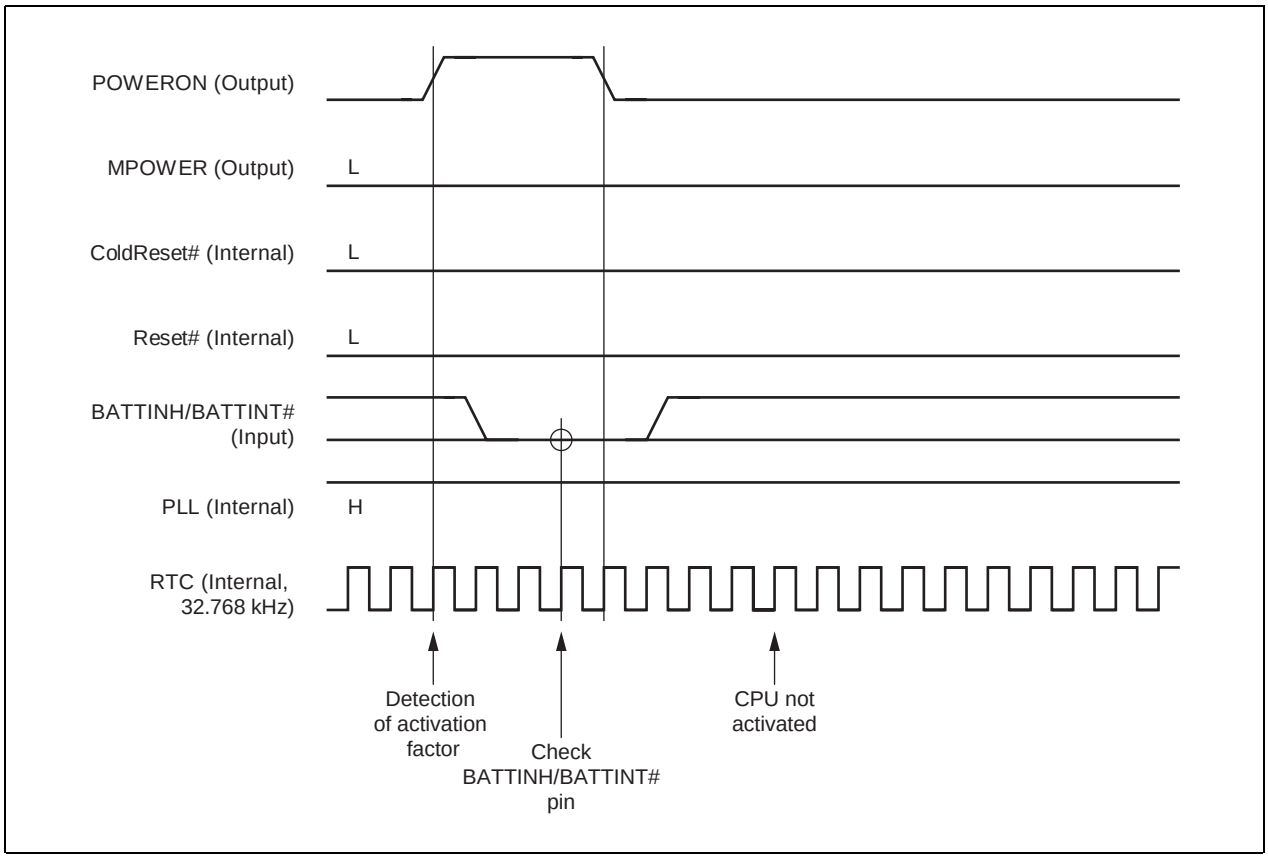


Figure 5-7. VR4181 Activation Sequence (When Activation Is NG)



★

5.3 Reset of CPU Core

This section describes the reset sequence of the VR4110 CPU core.

5.3.1 Cold Reset

In the VR4181, a Cold Reset sequence is executed in the CPU core in the following cases:

- RTC reset
- RSTSW reset
- Deadman's Switch reset
- Software shutdown
- HALTimer shutdown
- BATTINH shutdown (shutdown according to battery state)

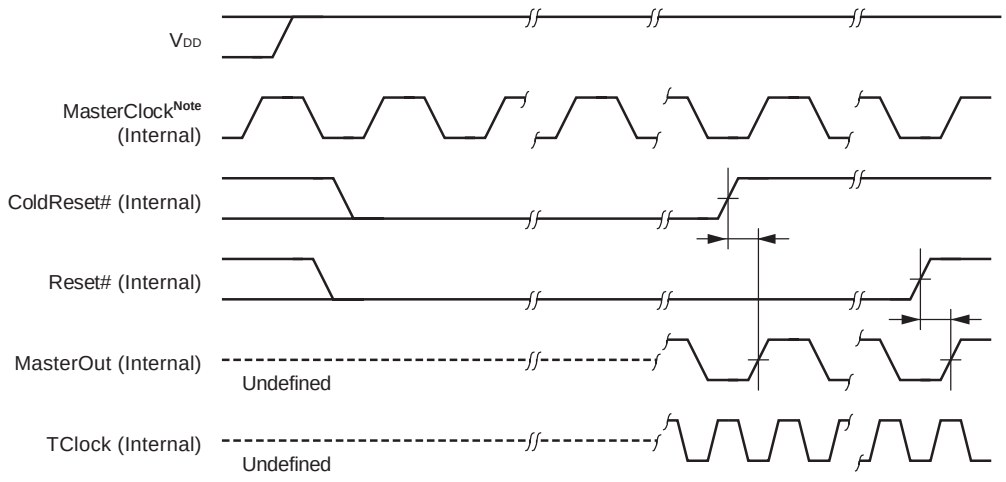
A Cold Reset completely initializes the CPU core, except for the following register bits.

- The TS and SR bits of the Status register are cleared to 0.
- The ERL and BEV bits of the Status register are set to 1.
- The upper limit value (31) is set in the Random register.
- The Wired register is initialized to 0.
- ★ The Count register is initialized to 0.
- Bits 31 to 28 of the Config register are set to 0 and bits 22 to 3 to 0x04800; 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 2 ms assertion, the ColdReset# signal is deasserted ★ synchronously with the rising edge of MasterOut (internal). Then the Reset# signal is deasserted synchronously with the rising edge of MasterOut, and the Cold Reset is completed.

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 5-8. Cold Reset



★ **Note** MasterClock is the basic clock used in the CPU core. Its frequency is one forth of TClock frequency.

5.3.2 Soft Reset

Caution Soft Reset is not supported in the current VR4181.

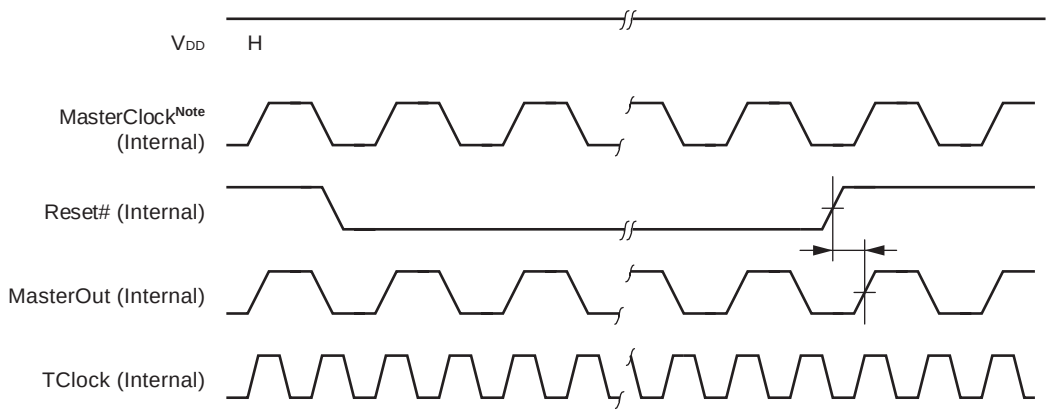
A Soft Reset initializes the CPU core without affecting the output clocks; in other words, a Soft Reset is a logical reset. In a Soft Reset, the CPU core retains as much state information as possible; all state information except for the following is retained:

- The TS bit of the Status register is cleared to 0.
- The SR, ERL and BEV bits of the Status register are set to 1.
- ★ The IP7 bit of the Cause register is cleared to 0.
- Any interrupts generated on the SysAD bus are cleared.
- NMI 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 ★ synchronized with the rising edge of MasterOut. 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 5-9. Soft Reset



★ **Note** MasterClock is the basic clock used in the CPU core. Its frequency is one forth of TClock frequency.

★ 5.4 Notes on Initialization

This section explains the case in which manipulation by software is necessary after the VR4181 has been reset.

When a Cold Reset sequence is executed, the reset exception vector is accessed. Perform manipulation described here by using the software (handler) for reset exceptions located at the reset exception vector.

5.4.1 CPU core

(1) Coprocessor 0

Be sure to initialize at least the following internal registers of the coprocessor 0 (CP0) after the RTC reset, RSTSW reset, or Deadman's Switch reset has been cleared, or the VR4181 has returned from the Hibernate mode.

- Config register
- Status register
- WatchLo register

(2) Cache tag

The contents of the tag RAM of the cache are undefined immediately after a voltage has been applied to the 2.5 V power supply when the RTC reset or Deadman's Switch reset has been cleared, or when the VR4181 has returned from the Hibernate mode. Before accessing an address at which the cache can be used, therefore, be sure to initialize the contents of the tag RAM of both the instruction cache and data cache. Use the TagLo register in CP0 to initialize the tags.

5.4.2 Internal peripheral units

(1) HALTimer

Set the HALTIMERRST bit of the PMUCNTREG register in the PMU to 1 within 4 seconds after clearing the RTC reset or RSTSW reset. This resets the HALTimer.

(2) Memory controller

Before accessing the DRAM space, be sure to initialize the registers in the memory controller. Especially when SDRAM is used, initialize SDRAM by executing the procedure described in **6.5.2 MEMCFG_REG (0x0A00 0304)**. A function to operate SDCLK only when SDRAM is accessed, for example, is not valid unless a mode setting command is issued to SDRAM by using the MEMCFG_REG register.

(3) Clock supply to peripheral units

The clock is not supplied in the default status to the peripheral units such as CSI, AIU, PIU, SIU1, and SIU2, and the A/D and D/A converters. To start using these units and converters, supply the necessary clock to them by setting the CMUCLKMSK register in the MBA Host Bridge. If these units are not used or they have finished being used, mask the clock supply by setting the CMUCLKMSK register.

(4) Alternate-function pins

The function of an alternate-function pin and the I/O direction of the GPIO pins are selected by the registers in the GIU. Be sure to set these registers in accordance with the unit or the function of the pin to be used. Exercise care in setting the registers so that signals do not conflict on the board or that a signal whose level is required does not go into a high-impedance state.

5.4.3 Returning from power mode

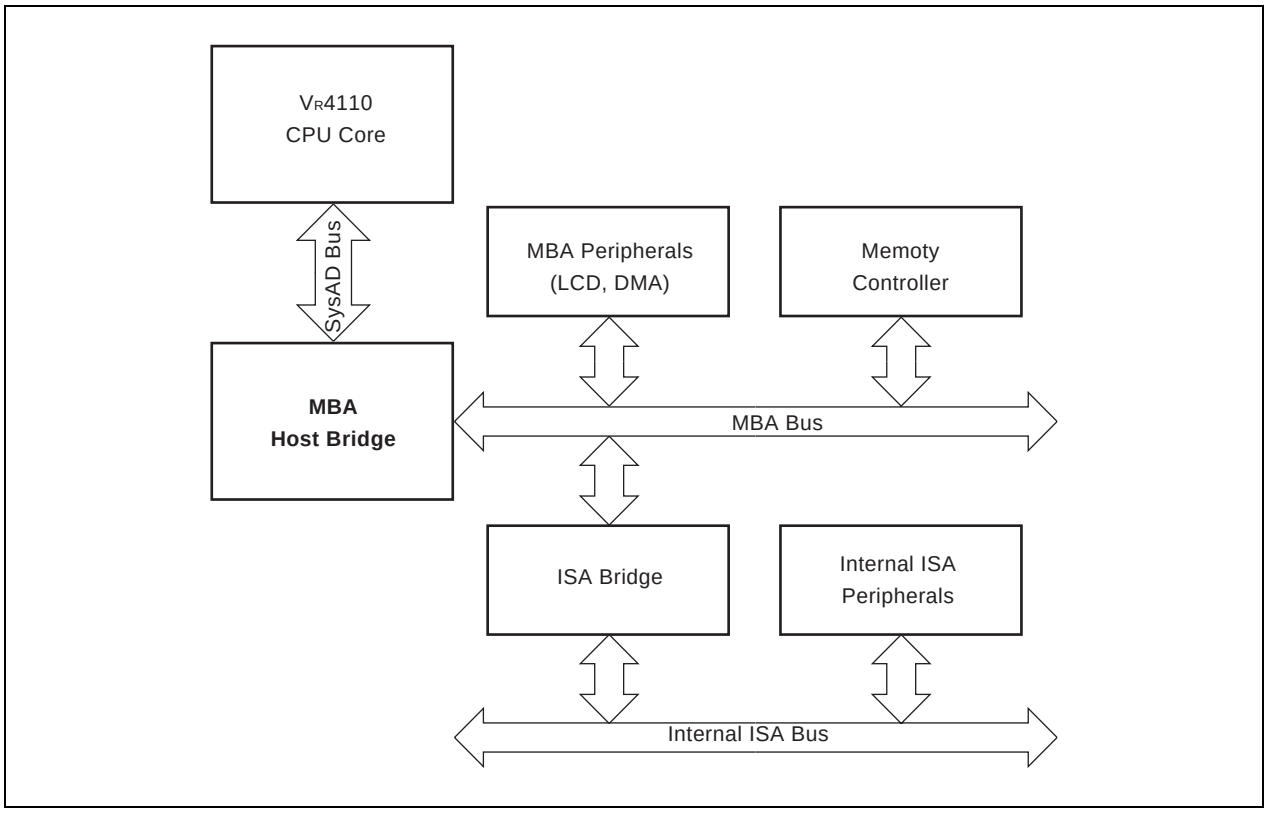
For initialization after the VR4181 has returned from the Hibernate mode or Suspend mode, refer to **10.6 DRAM Interface Control**.

CHAPTER 6 BUS CONTROL

6.1 MBA Host Bridge

The MBA (Modular Bus Architecture) Host Bridge is an interface between the CPU core and the MBA bus and operates as an external agent to the CPU core. It handles all requests from the CPU core if it is provided proper resources. The MBA Host Bridge can decode the entire physical address space to start appropriate bus accesses such as MBA requests, MBA - ISA protocols, or external ROM accesses through the peripheral bus. It also has functions as a host bridge to implement proper cycle timings and bus transaction protocols.

Figure 6-1. VR4181 Internal Bus Structure



★

6.1.1 MBA Host Bridge ROM and register address space

Physical address	Type	Device
0x1FFF FFFF to 0x1800 0000	Memory (range)	ROM
0x0A00 0014 to 0x0A00 0000	I/O (range)	Bus control registers
0x0A00 0080	I/O	Interrupt register
0x0A00 008C	I/O	Interrupt register
0x0A00 0098	I/O	Interrupt register
0x0A00 009A	I/O	Interrupt register
0x0A00 0200	I/O	Interrupt register
0x0A00 0206	I/O	Interrupt register

In addition to the decoding of above addresses, the Host Bridge generates MBA select signals if other MBA masters intend to access the above devices. The Host Bridge responds to the above addresses only upon a CPU access. For any other addresses the Host Bridge initiates an MBA cycle to access an appropriate resources.

6.1.2 MBA modules address space

(1) Memory controller

Physical address	Type	Device
0x03FF FFFF to 0x0000 0000	Memory (range)	DRAM
0x0A00 03FF to 0x0A00 0300	I/O (range)	Control registers

The MBA memory controller is selected when the above address ranges are accessed.

(2) DMA controller

Physical address	Type	Device
0x0A00 0048 to 0x0A00 0020	I/O (range)	Control registers 1
0x0A00 06FF to 0x0A00 0600	I/O (range)	Control registers 2

The MBA DMA controller is selected when the above I/O ranges are accessed.

(3) LCD module (LCD Control Unit)

Physical address	Type	Device
★ 0x0A00 05FF to 0x0A00 0400	I/O (range)	Control registers

The LCD module is selected when the above I/O range is accessed.

(4) ISA Bridge

Physical address	Type	Device
0x17FF FFFF to 0x1400 0000	I/O (64M, range)	External ISA bus (I/O)
0x13FF FFFF to 0x1000 0000	Memory (64M, range)	External ISA bus (Memory)
0x0BFF FFFF to 0x0B00 0000	I/O (16M, range)	ISA internal I/O 1
0x0CFF FFFF to 0x0C00 0000	I/O (16M, range)	ISA internal I/O 2

The ISA Bridge is selected when the above address ranges are accessed.

6.2 Bus Control Registers

External ROM accesses and supply of clocks to several internal units are controlled by the bus control registers listed below.

Table 6-1. Bus Control Registers

Physical address	R/W	Register symbol	Function
0x0A00 0000	R/W	BCUCNTREG1	BCU control register 1
0x0A00 0004	R/W	CMUCLKMSK	Clock mask register
0x0A00 000C	R/W	BCUSPEEDREG	BCU access time parameters register
0x0A00 0010	R/W	BCURFCNTREG	BCU refresh control register
0x0A00 0014	R	REVIDREG	Revision ID register
0x0A00 0018	R	CLKSPEEDREG	Clock speed register

Caution Since these registers are powered by 2.5 V power supply, the contents of these registers are cleared after Hibernate mode.

6.2.1 BCUCNTREG1 (0x0A00 0000)

Bit	15	14	13	12	11	10	9	8
Name	ROMs1	ROMs0	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R/W	R/W	R	R	R	R	R	R
At reset	1	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	ROMWEN0	Reserved	Rtype1	Rtype0	RSTOUT
R/W	R	R	R	R/W	R	R/W	R/W	R/W
At reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15, 14	ROMs(1:0)	Defines ROM size to be used (for all banks) 00 : Reserved 01 : 32 Mbit 10 : 64 Mbit 11 : Reserved
13 to 5	Reserved	0 is returned when read
★ 4	ROMWEN0	Enables flash memory write (for all banks). Write strobe can be generated when this bit is set to 1. 0 : Disabled 1 : Enabled
3	Reserved	0 is returned when read
2, 1	Rtype(1:0)	ROM type (for all banks) 00 : Ordinary ROM 01 : Flash memory 10 : Page ROM 11 : Reserved
0	RSTOUT	RESET# output control. This bit does not affect GPIO21/RESET# pin's state when this pin is not defined as RESET# output. 0 : RESET# is active (low level) 1 : RESET# is inactive (high level)

This register is used to set ROM type and capacity of ROM Bank 0, 1, 2 and 3.

★ **Caution When writing to flash memory, be sure to set Rtype(1:0) bits to 01 in addition to a setting of ROMWEN0 bit to 1.**

★ **Remark** When a ROM type other than flash memory is selected (Rtype(1:0) bits are set to other than 01), the operation of the VR4181 is undefined if a write to the ROM space is performed.

6.2.2 CMUCLKMSK (0x0A00 0004)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
At reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	MSKCSU PCLK	MSKAIU PCLK	MSKPIU PCLK	MSKADU PCLK	MSKSIU 18M	MSKADU 18M	Reserved
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
At reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 7	Reserved	0 is returned when read
6	MSKCSUPCLK	Supply/Mask Clocked Serial Interface (CSI) peripheral clock (PCLK) 0 : Mask 1 : Supply
5	MSKAIUPCLK	Supply/Mask Audio Interface (AIU) peripheral clock (PCLK) 0 : Mask 1 : Supply
4	MSKPIUPCLK	Supply/Mask Touch Panel Interface (PIU) peripheral clock (PCLK) 0 : Mask 1 : Supply
3	MSKADUPCLK	Supply/Mask A/D converter and D/A converter peripheral clock (PCLK) 0 : Mask 1 : Supply
2	MSKSIU18M	Supply/Mask Serial Interface 1 and 2 (SIU1/SIU2) 18.432 MHz clock 0 : Mask 1 : Supply
1	MSKADU18M	Supply/Mask A/D converter and D/A converter 18.432 MHz clock 0 : Mask 1 : Supply
0	Reserved	Write 0 when write. 0 is returned when read.

This register is used to mask the clocks that are supplied to CSI, AIU, PIU, SIU1, and SIU2.

6.2.3 BCUSPEEDREG (0x0A00 000C)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	WPROM2	WPROM1	WPROM0	Reserved	Reserved	Reserved	Reserved
R/W	R	R/W	R/W	R/W	R	R	R	R
At reset	0	1	1	1	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	WROMA3	WROMA2	WROMA1	WROMA0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
At reset	0	0	0	0	1	1	1	1

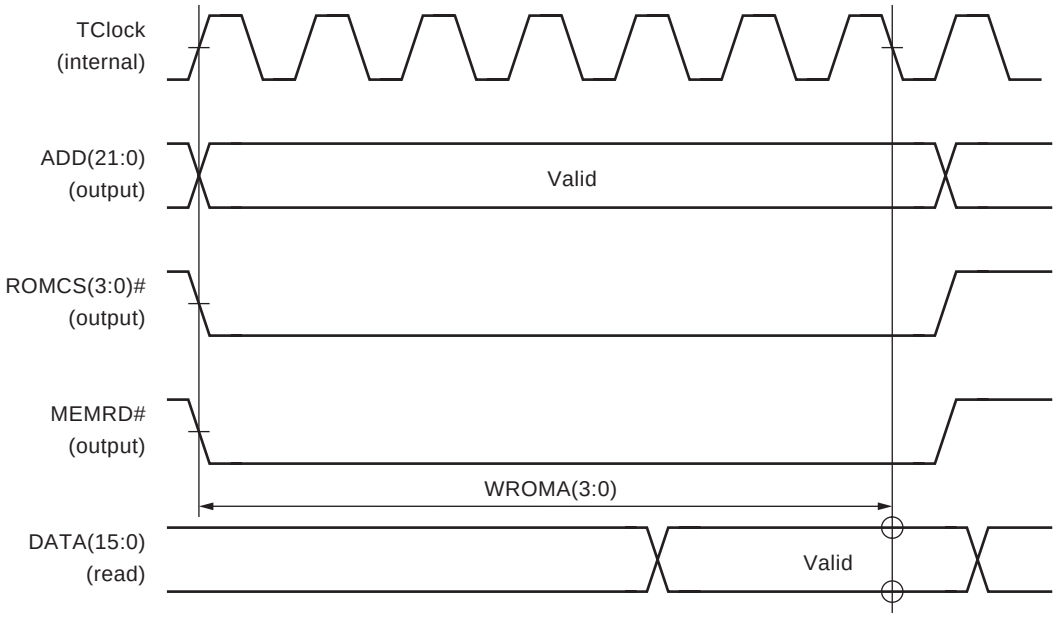
Bit	Name	Function
15	Reserved	0 is returned when read
14 to 12	WPROM(2:0)	Page ROM access speed 000 : 1.5 TClock 001 : 2.5 TClock 010 : 3.5 TClock 011 : 4.5 TClock 100 : 5.5 TClock 101 : 6.5 TClock 110 : 7.5 TClock 111 : 8.5 TClock
11 to 4	Reserved	0 is returned when read
3 to 0	WROMA(3:0)	ROM access speed 0000 : 1.5 TClock 0001 : 2.5 TClock 0010 : 3.5 TClock 0011 : 4.5 TClock 0100 : 5.5 TClock 0101 : 6.5 TClock 0110 : 7.5 TClock 0111 : 8.5 TClock 1000 : 9.5 TClock 1001 : 10.5 TClock 1010 : 11.5 TClock 1011 : 12.5 TClock 1100 : 13.5 TClock 1101 : 14.5 TClock 1110 : 15.5 TClock 1111 : 16.5 TClock

This register is used to set ROM access parameter of Bank 0, 1, 2, and 3. About the relationship between these bits and ROM cycles, refer to **Figure 6-2. ROM Read Cycle and Access Parameters**.

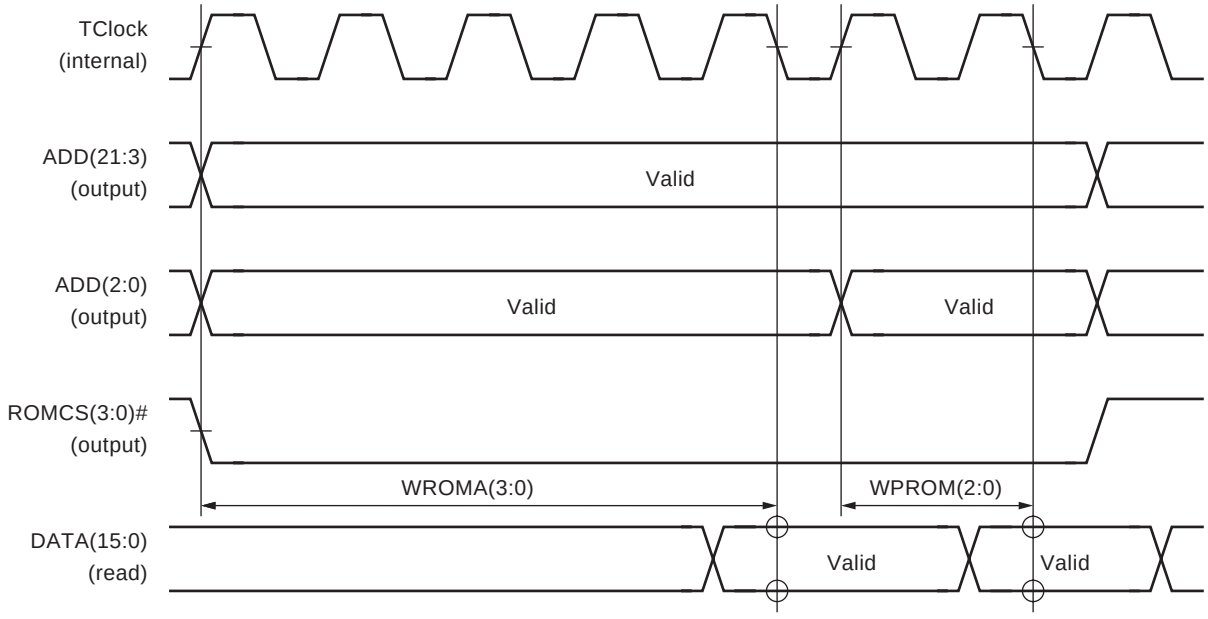
★ **Remark** The maximum burst number when using a PageROM is 8 halfwords (i.e. 128 bits; 1 word = 32 bits).

★ **Figure 6-2. ROM Read Cycle and Access Parameters**

(a) Ordinary ROM cycle



(b) PageROM cycle



Remarks 1. ROMCS(2:0)# signals are alternated with general-purpose I/O signals and are defined as general-purpose inputs after RTC reset. Set GPMD2REG and GPMD3REG registers in the GIU to use them as ROMCS(2:0)#.

2. A circle in the figure indicates the sampling timing.

6.2.4 BCURFCNTREG (0x0A00 0010)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	BRF13	BRF12	BRF11	BRF10	BRF9	BRF8
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
At reset	0	0	0	1	1	1	1	1

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
At reset	1	1	1	1	1	1	1	1

Bit	Name	Function
15, 14	Reserved	0 is returned when read
13 to 0	BRF(13:0)	These bits select the DRAM refresh rate that is based on the TClock. The refresh rate is obtained by following expression. Refresh rate = BRF(13:0) x TClock period For example, to select a 15.6 μs refresh rate with a 50 MHz TClock: BRF(13:0) = 15600 (ns) / 20 (ns) = 0x30C

Remarks 1. When the IORDY signal does not become high level though the DRAM refresh rate has elapsed during the external ISA memory or I/O cycles, a DRAM refresh cycle may be lost.

2. Refresh timing is generated from detecting match between values of the internal up counter and BCURFCNTREG register. Therefore, when the BCURFCNTREG register value is changed smaller than current value, and if the internal counter value is larger than the new BCURFCNTREG register value, the next CBR refresh timing is at next match after the counter rounds over.

6.2.5 REVIDREG (0x0A00 0014)

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

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	MNREV3	MNREV2	MNREV1	MNREV0
R/W	R	R	R	R	R	R	R	R

Bit	Name	Function
15 to 12	RID(3:0)	Processor revision ID (Read Only)
11 to 8	MJREV(3:0)	Major revision ID number (Read only)
7 to 4	Reserved	0 is returned when read
3 to 0	MNREV(3:0)	Minor revision ID (Read only)

This register is used to indicate the revision of the VR4181. The relationship between the values and the revision of the VR4181 is as follows.

VR4181 Revision	RID(3:0)	MJREV(3:0)	MINREV(3:0)
1.0	0x0	0x0	0x0
1.1	0x0	0x0	0x1
1.2	0x0	0x0	0x2
1.3	0x0	0x0	0x2

Even if the CPU core or the peripheral unit has been changed, there is no guarantee that REVIDREG register will be reflected, or that the changes to the revision number necessarily reflect real changes of the CPU core or the peripheral unit. For this reason, software should not rely on the revision number in REVIDREG register to characterize the units.

Caution Values of this register bits differ depending on the delivery date.

6.2.6 CLKSPEEDREG (0x0A00 0018)

Bit	15	14	13	12	11	10	9	8
Name	DIV2	DIV3	DIV4	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	CLKSP4	CLKSP3	CLKSP2	CLKSP1	CLKSP0
R/W	R	R	R	R	R	R	R	R

Bit	Name	Function
15 to 13	DIV(2:4)	Value used to calculate the TClock, MBA clock, and SDCLK operating frequency
12 to 5	Reserved	0 is returned when read
4 to 0	CLKSP(4:0)	Value used to calculate the CPU core operating clock (PClock) frequency

The following expression is used to calculate the PClock and TClock frequency:

(1) CPU core clock (PClock)

PClock = (18.432 MHz / CLKSP(4:0)) x 64

★ **(2) Peripheral clock (TClock)**

DIV(2:4)	Ratio	Mode
111	TClock = PClock / 1	Div1 mode
011	TClock = PClock / 2	Div2 mode
101	TClock = PClock / 3	Div3 mode
Others	Reserved	–

Remark PClock frequency is decided by CLKSEL(2:0) pin statuses during RTC reset.
TClock frequency is always a half of PClock frequency (Div2 mode) immediately after RTC reset. Software can change TClock Div mode by setting the PMUDIVREG register (0x0B00 00AC). A change becomes valid when the VR4181 restores from the Hibernate mode after setting the PMUDIVREG register. ★

6.3 ROM Interface

The VR4181 supports three ROM modes, ordinary ROM, PageROM, and flash memory. The mode setting is made via the BCUCNTREG1 register's Rtype(1:0) bits and ROMWEN0 bit. Access speed setting in ordinary ROM or PageROM mode is made via the BCUSPEEDREG register.

Remark The VR4181 supports only 16-bit access for external ROM devices.

6.3.1 External ROM devices memory mapping

Physical address	32 Mbit ROM	64 Mbit ROM
0x1FFF FFFF to 0x1FC0 0000	Bank 3 (ROMCS3#)	Bank 3 (ROMCS3#)
0x1FBF FFFF to 0x1F80 0000	Bank 2 (ROMCS2#)	
0x1F7F FFFF to 0x1F40 0000	Bank 1 (ROMCS1#)	Bank 2 (ROMCS2#)
0x1F3F FFFF to 0x1F00 0000	Bank 0 (ROMCS0#)	
0x1EFF FFFF to 0x1E80 0000	Reserved	Bank 1 (ROMCS1#)
0x1E7F FFFF to 0x1E00 0000	Reserved	Bank 0 (ROMCS0#)

Bank 3 contains boot vector and has a dedicated pin for chip select (ROMCS3#). Chip select pins for Bank 2, 1, and 0, ROMCS(2:0)#, are alternated with general-purpose I/O signals and are defined as general-purpose inputs after RTC reset. Set GPMD2REG and GPMD3REG registers in the GIU to use them as ROMCS(2:0)#.

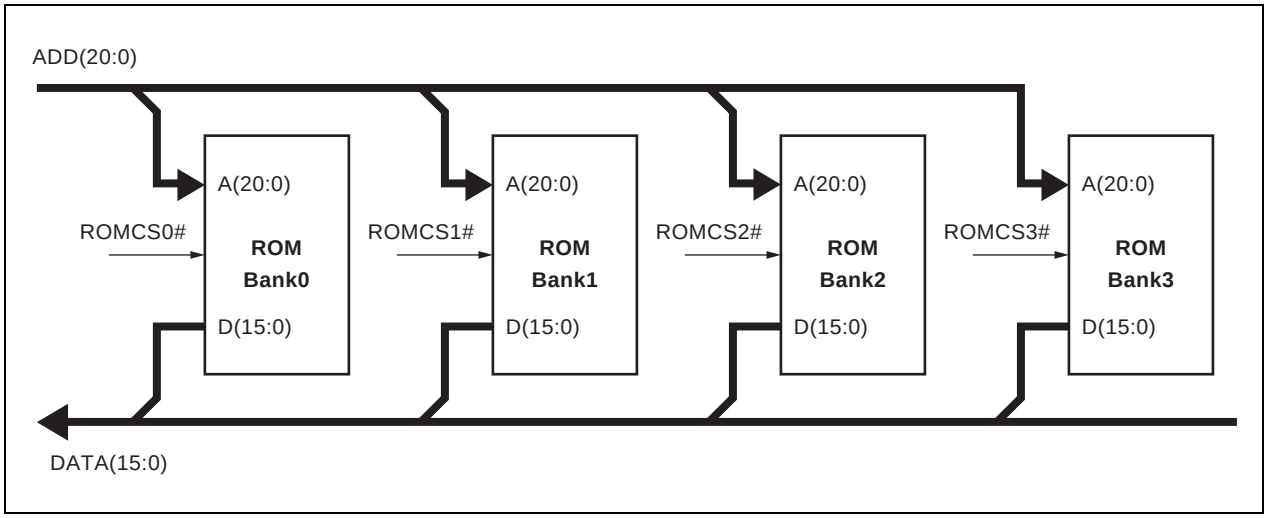
6.3.2 Connection to external ROM (x 16) devices

★ The ADD(21:0) pins are connected to the address line ADD(21:0) inside the VR4181 during DRAM accesses. However, during ROM or flash memory accesses, they are connected to the address line ADD(22:1) inside the VR4181. This allows providing a greater address space capacity for ROM or flash memory.

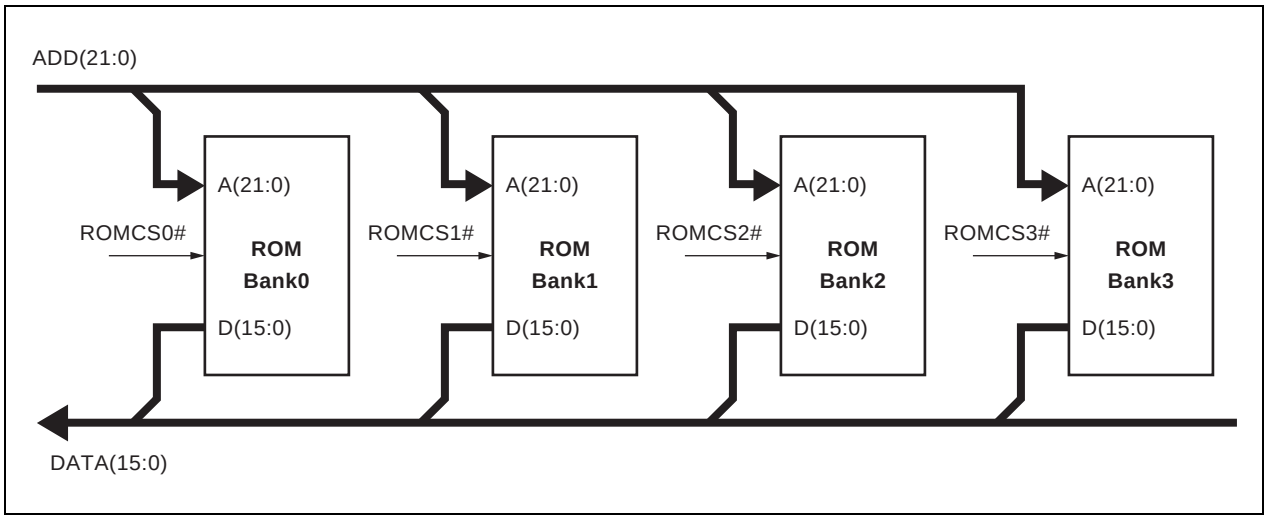
ROM address pin	32 Mbit ROM (2 Mbits x 16)		64 Mbit ROM (4 Mbits x 16)	
	VR4181 pin	CPU core physical address line	VR4181 pin	CPU core physical address
A21			ADD21	adr22
A20	ADD20	adr21	ADD20	adr21
A19	ADD19	adr20	ADD19	adr20
A18	ADD18	adr19	ADD18	adr19
A17	ADD17	adr18	ADD17	adr18
A16	ADD16	adr17	ADD16	adr17
A15	ADD15	adr16	ADD15	adr16
A14	ADD14	adr15	ADD14	adr15
A13	ADD13	adr14	ADD13	adr14
A12	ADD12	adr13	ADD12	adr13
A11	ADD11	adr12	ADD11	adr12
A10	ADD10	adr11	ADD10	adr11
A9	ADD9	adr10	ADD9	adr10
A8	ADD8	adr9	ADD8	adr9
A7	ADD7	adr8	ADD7	adr8
A6	ADD6	adr7	ADD6	adr7
A5	ADD5	adr6	ADD5	adr6
A4	ADD4	adr5	ADD4	adr5
A3	ADD3	adr4	ADD3	adr4
A2	ADD2	adr3	ADD2	adr3
A1	ADD1	adr2	ADD1	adr2
A0	ADD0	adr1	ADD0	adr1

6.3.3 Example of ROM connection

(1) 32 Mbit ordinary ROM

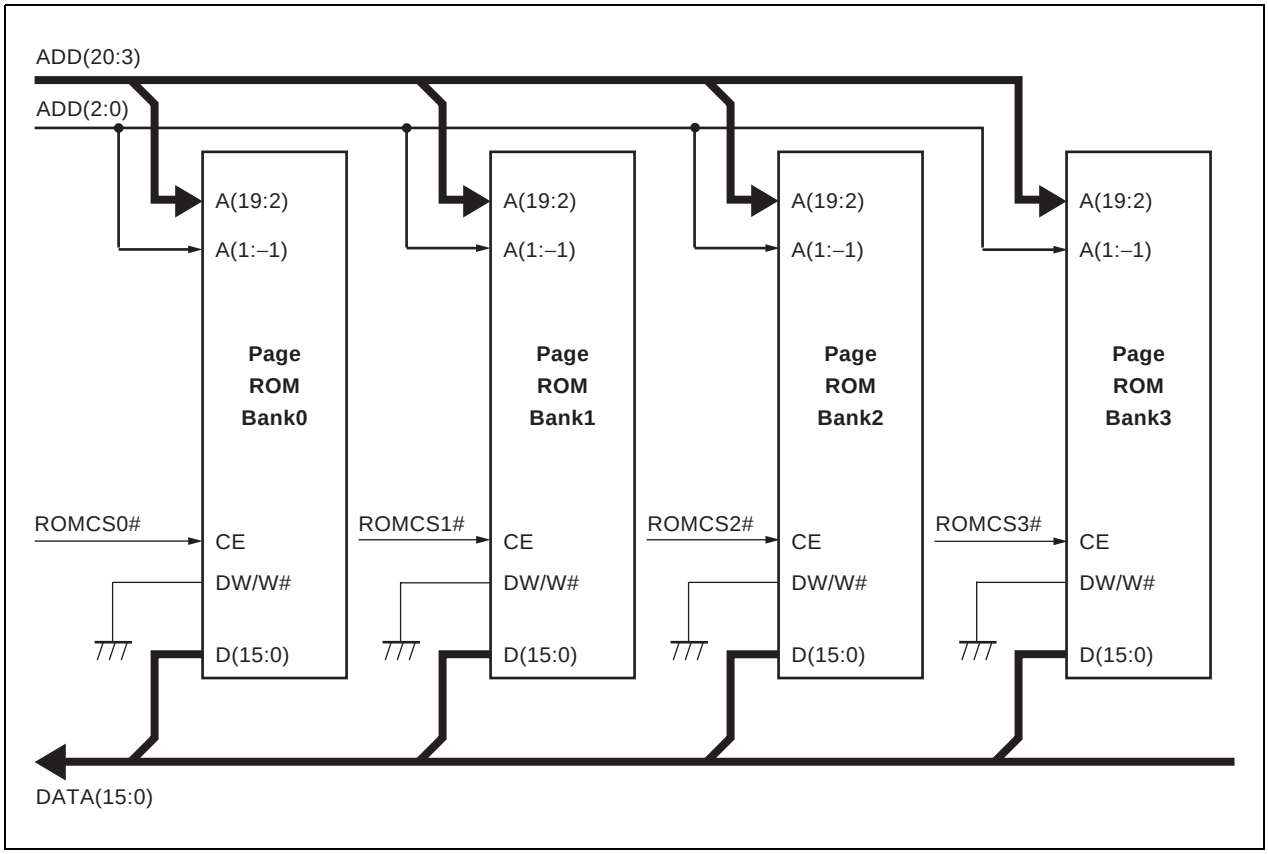


(2) 64 Mbit ordinary ROM



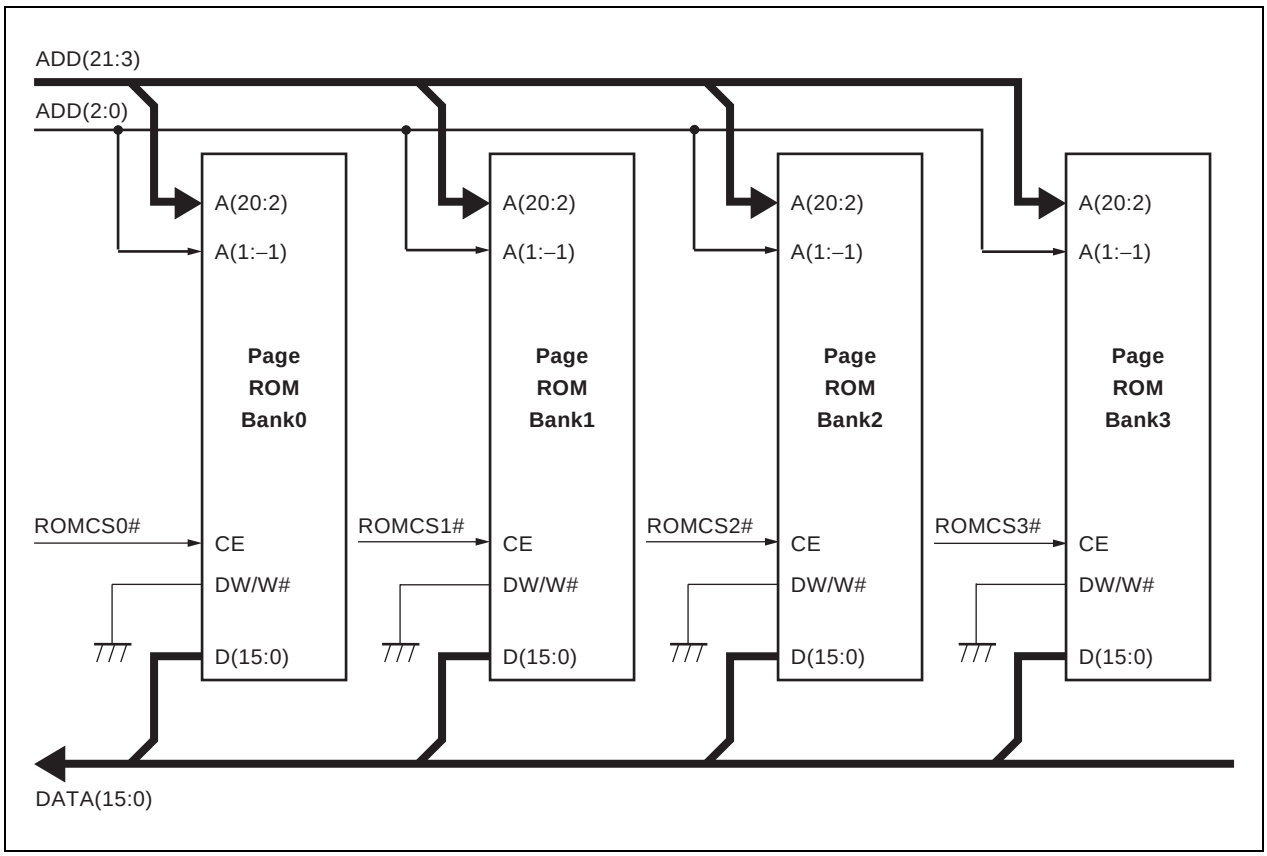
(3) 32 Mbit PageROM

★ **Remark** The maximum burst number when using a PageROM is 8 halfwords (i.e. 128 bits; 1 word = 32 bits).

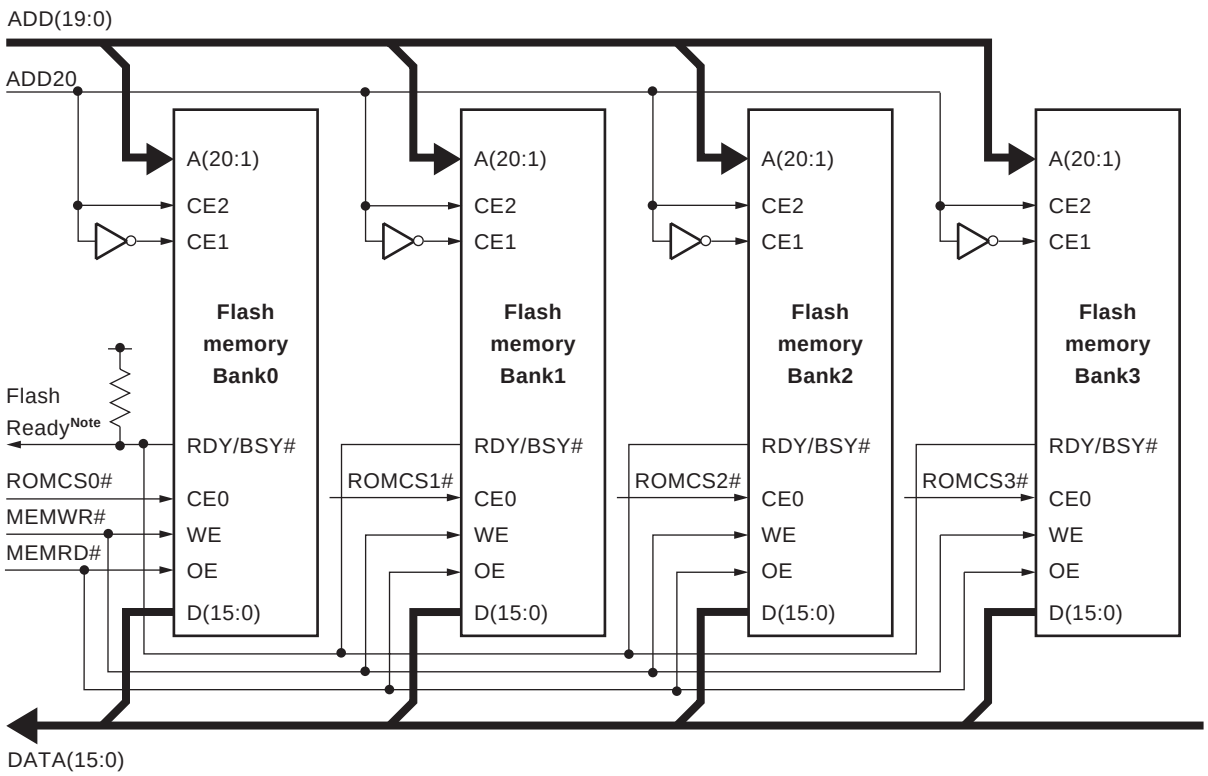


(4) 64 Mbit PageROM

★ **Remark** The maximum burst number when using a PageROM is 8 halfwords (i.e. 128 bits; 1 word = 32 bits).



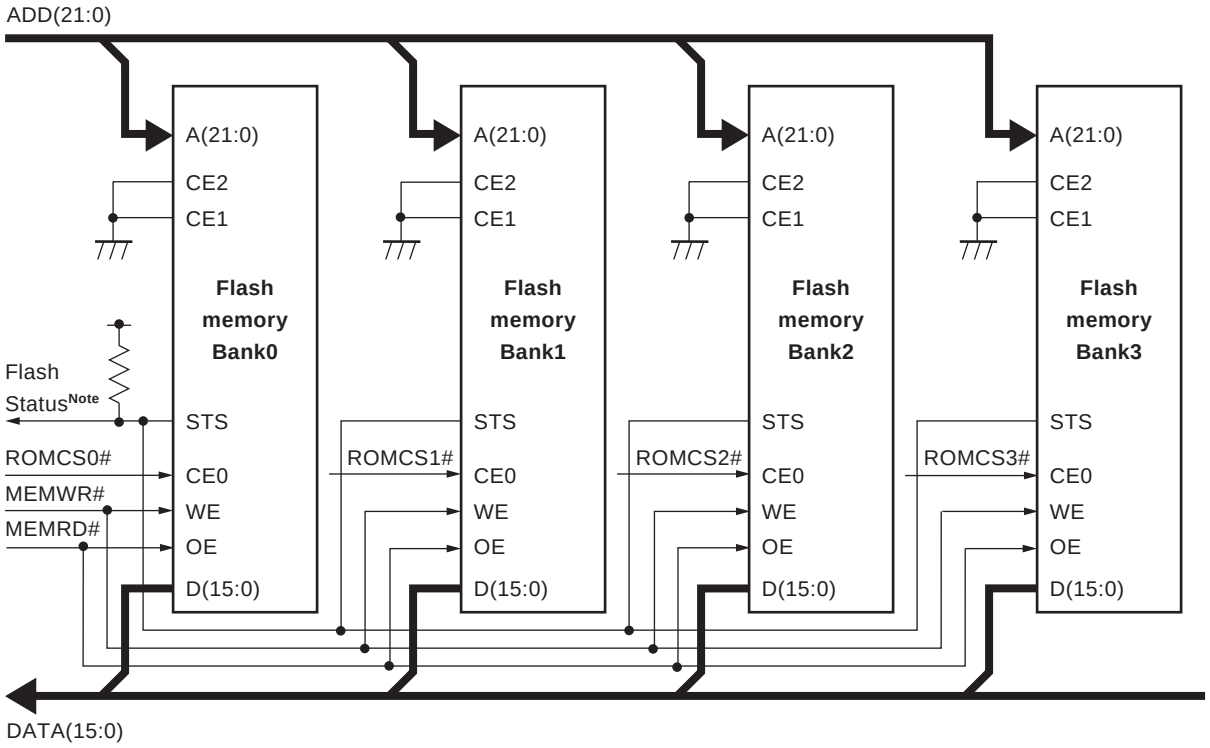
(5) 32 Mbit flash memory (when using Intel™ DD28F032)



Note There is no corresponding pin in the VR4181. Use one of the GPIO pins for this function.

Remark Use one of the GPIO pins in the VR4181 to control ON/OFF of VPP (program/erase supply voltage).

★ **(6) 64 Mbit flash memory (when using Intel StrataFlash™ 28F640J5)**



Note There is no corresponding pin in the VR4181. Using one of the GPIO pins for this function.

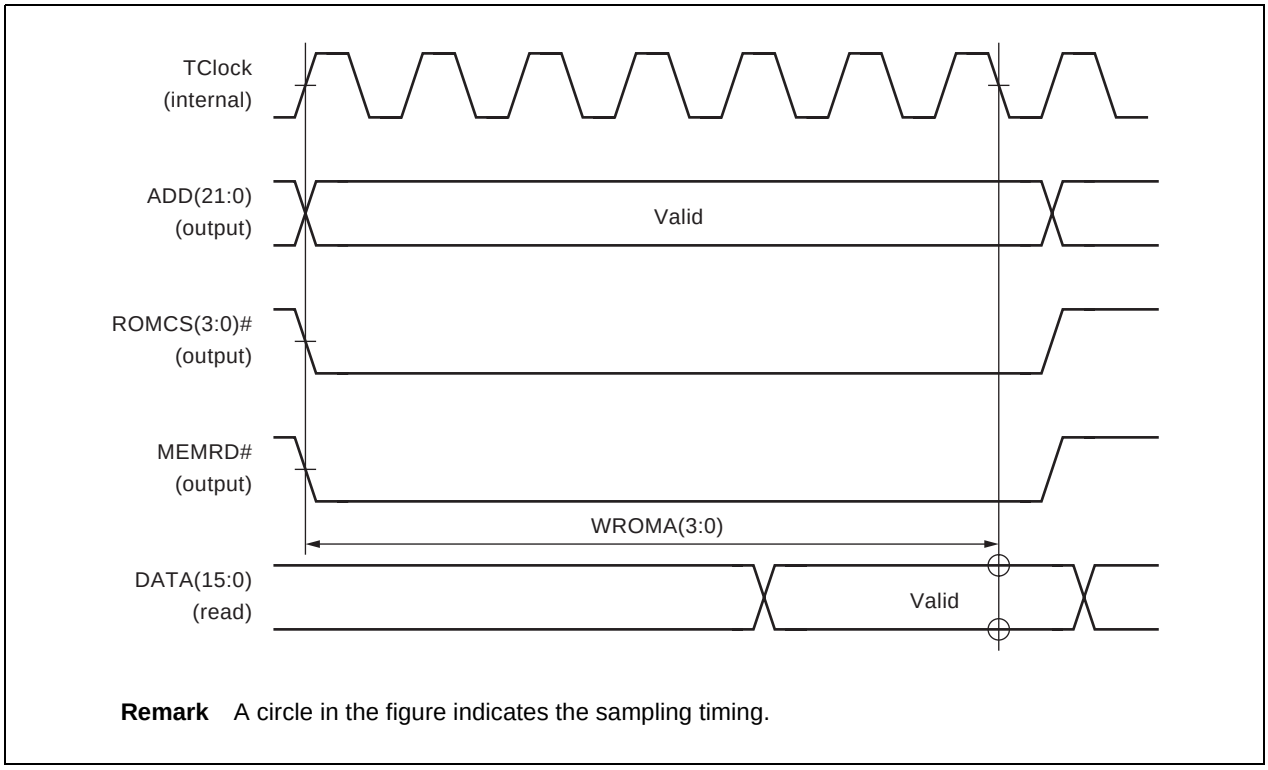
Remark Using one of the GPIO pins in the VR4181 to control ON/OFF of VPP (program/erase supply voltage).

6.3.4 External ROM cycles

The following timing diagrams illustrate the external ROM cycles depending on the settings in the bus control register and bus speed control register.

(1) Ordinary ROM read cycle

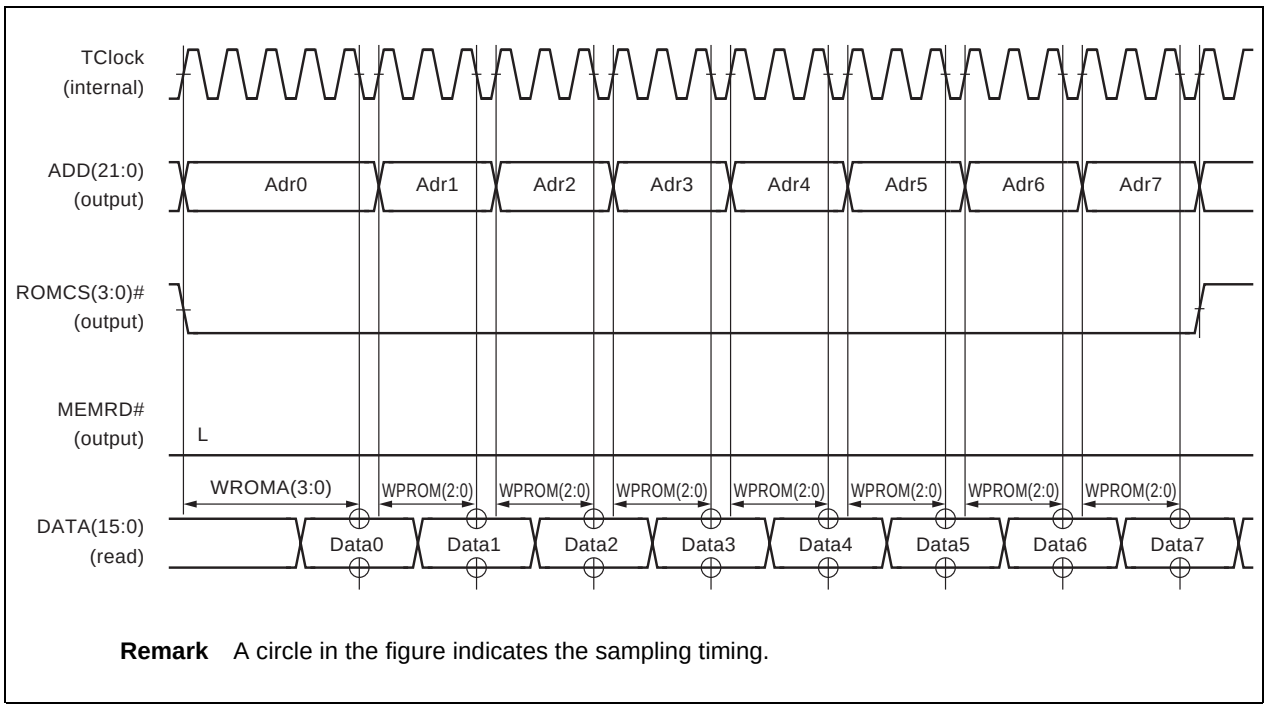
★ **Figure 6-3. Ordinary ROM Read Cycle (WROMA(3:0) = 0101)**



Remark A circle in the figure indicates the sampling timing.

(2) PageROM read cycle

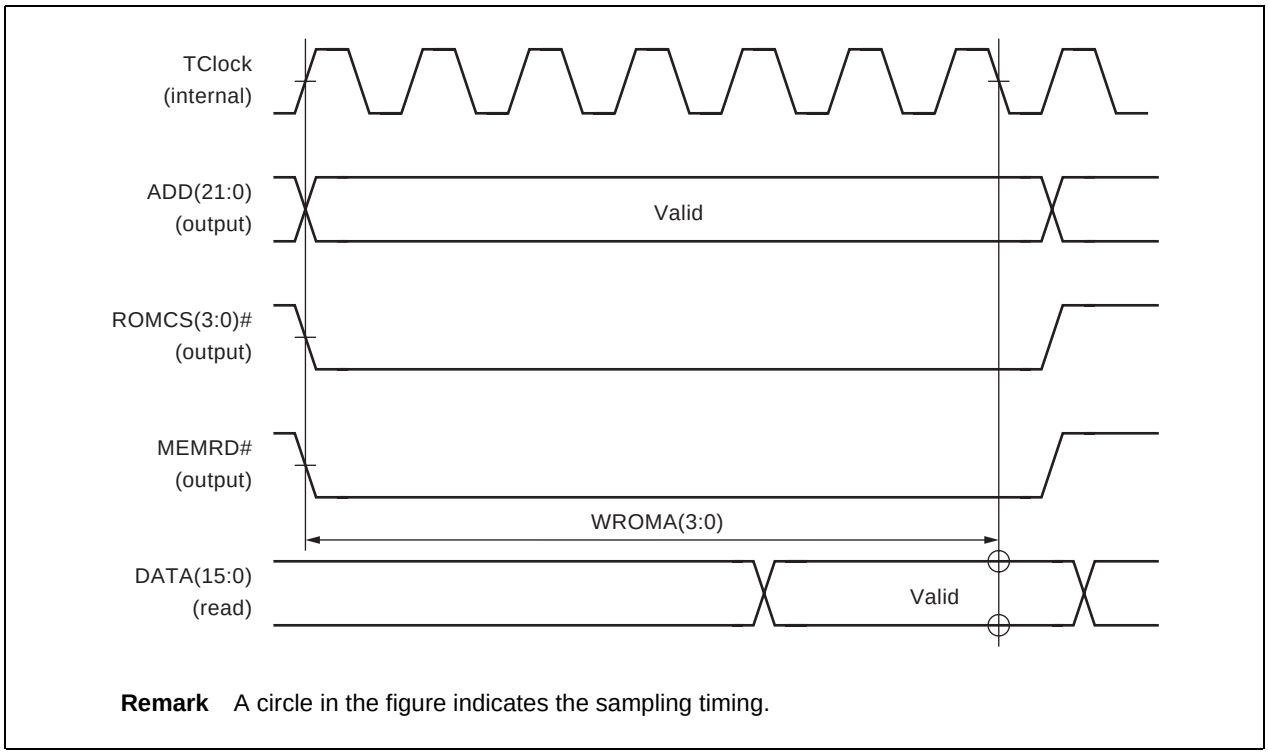
★ **Figure 6-4. PageROM Read Cycle (WROMA(3:0) = 0011, WPROM(2:0) = 001)**



Remark A circle in the figure indicates the sampling timing.

(3) Flash memory read cycle

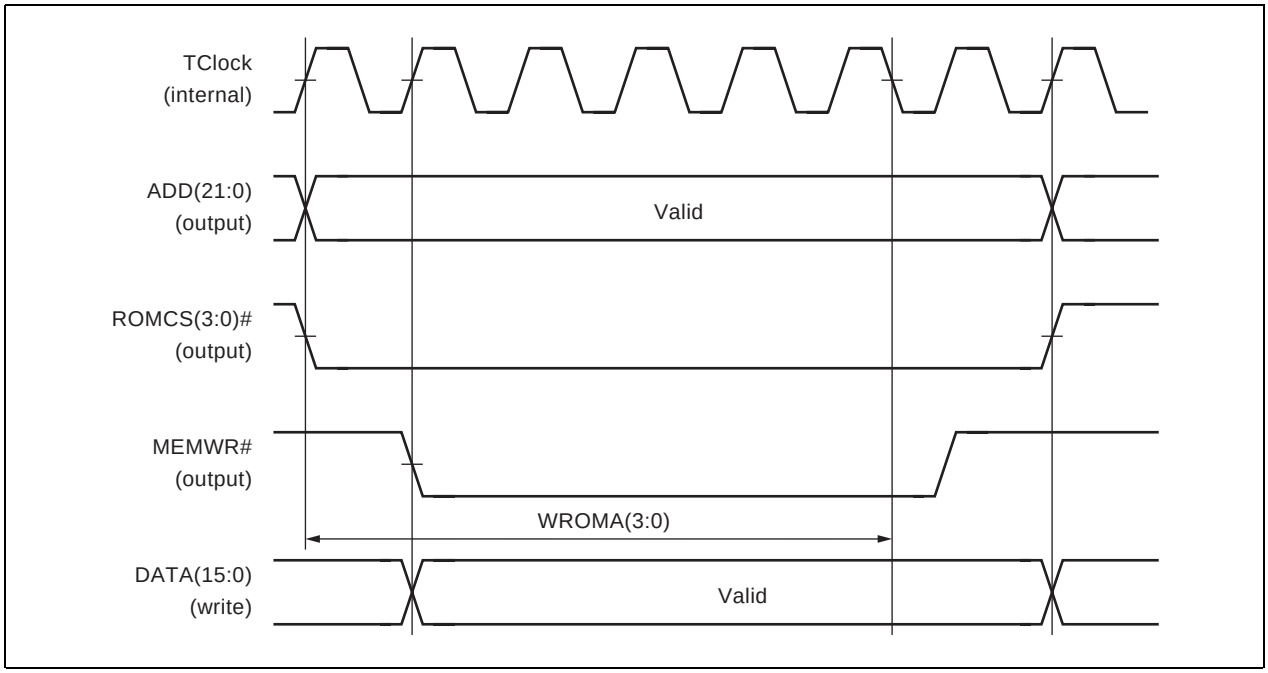
★ **Figure 6-5. Flash Memory Read Cycle (Rtype(1:0) = 01, WROMA(3:0) = 0101)**



Remark A circle in the figure indicates the sampling timing.

(4) Flash memory write cycle

★ **Figure 6-6. Flash Memory Write Cycle (Rtype(1:0) = 01, WROMA(3:0) = 0100)**

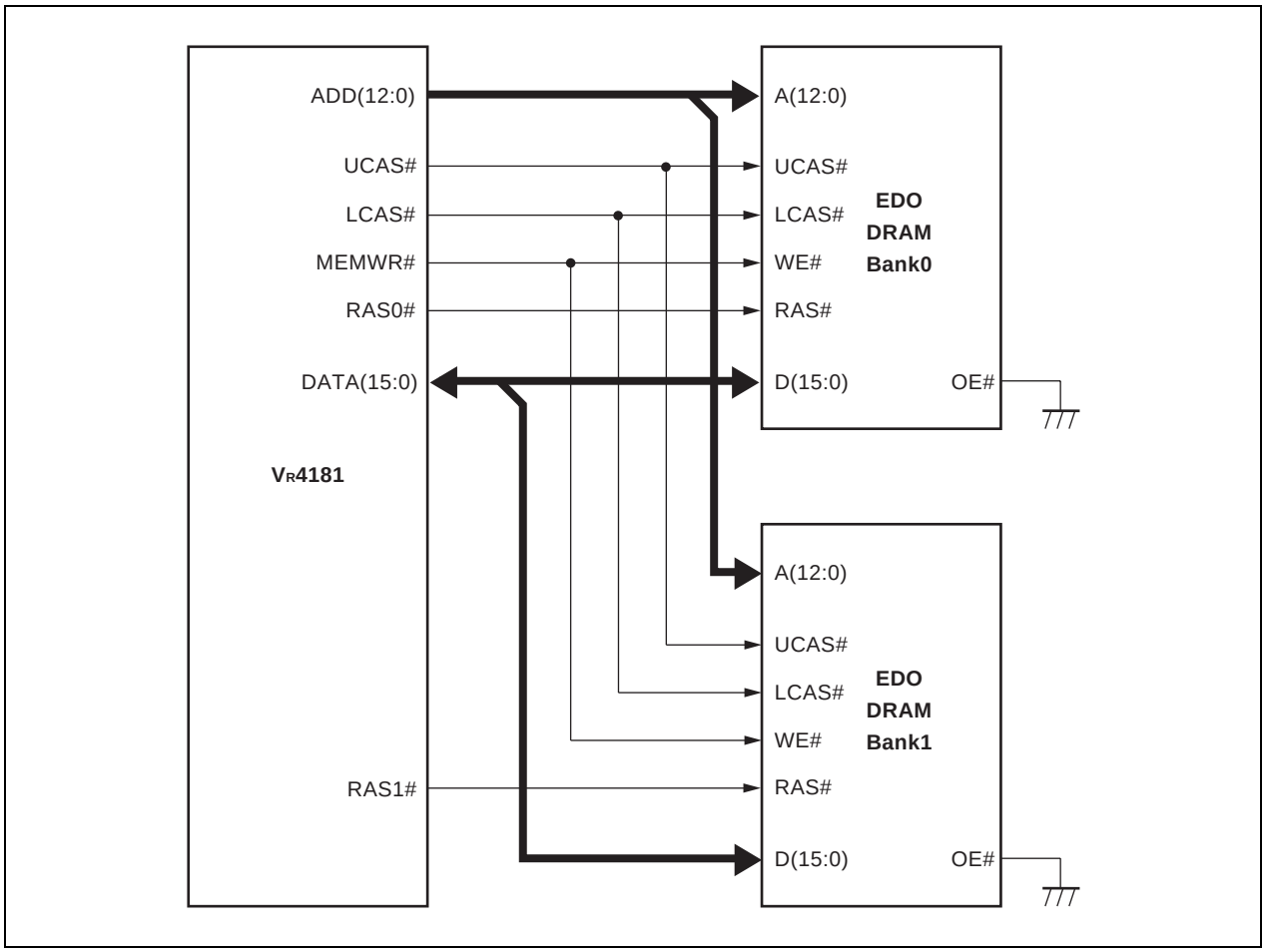


6.4 DRAM Interface

The VR4181 supports 16 Mbit or 64 Mbit DRAM (EDO DRAM or SDRAM). The DRAM size, type, and access speed is set via the memory controller's registers.

6.4.1 EDO DRAM configuration

★ **Figure 6-7. External EDO DRAM Configuration**



★ Figure 6-7 illustrates an example when connecting devices of 4 Mbits x 16.
Addresses when connecting devices of 16 Mbits or 64 Mbits are mapped as follows.

DRAM bank	Physical address (16 Mbits)	Physical address (64 Mbits)
Bank 0	0x001F FFFF to 0x0000 0000	0x007F FFFF to 0x0000 0000
Bank 1	0x003F FFFF to 0x0020 0000	0x00FF FFFF to 0x0080 0000

★ **Remark** 64 Mbit EDO DRAMs of other than 13 rows and 9 columns cannot be used with the VR4181.

6.4.2 Mixed memory mode (EDO DRAM only)

The MEMCFG_REG register provides two bits each for Bank 0 and Bank 1 to set types of DRAMs to be used. This allows the two banks to be configured with different types of DRAMs, for example, Bank 0 can be mapped on 64 Mbit devices and Bank 1 on 16 Mbit devices, to optimize the cost of the total memory required.

Table 6-2. VR4181 EDO DRAM Capacity

Bank 0	Bank 1	Total DRAM capacity
16 Mbits	0	2 MB
16 Mbits	16 Mbits	4 MB
64 Mbits	0	8 MB
16 Mbits	64 Mbits	10 MB
64 Mbits	16 Mbits	10 MB
64 Mbits	64 Mbits	16 MB

★

6.4.3 EDO DRAM timing parameters

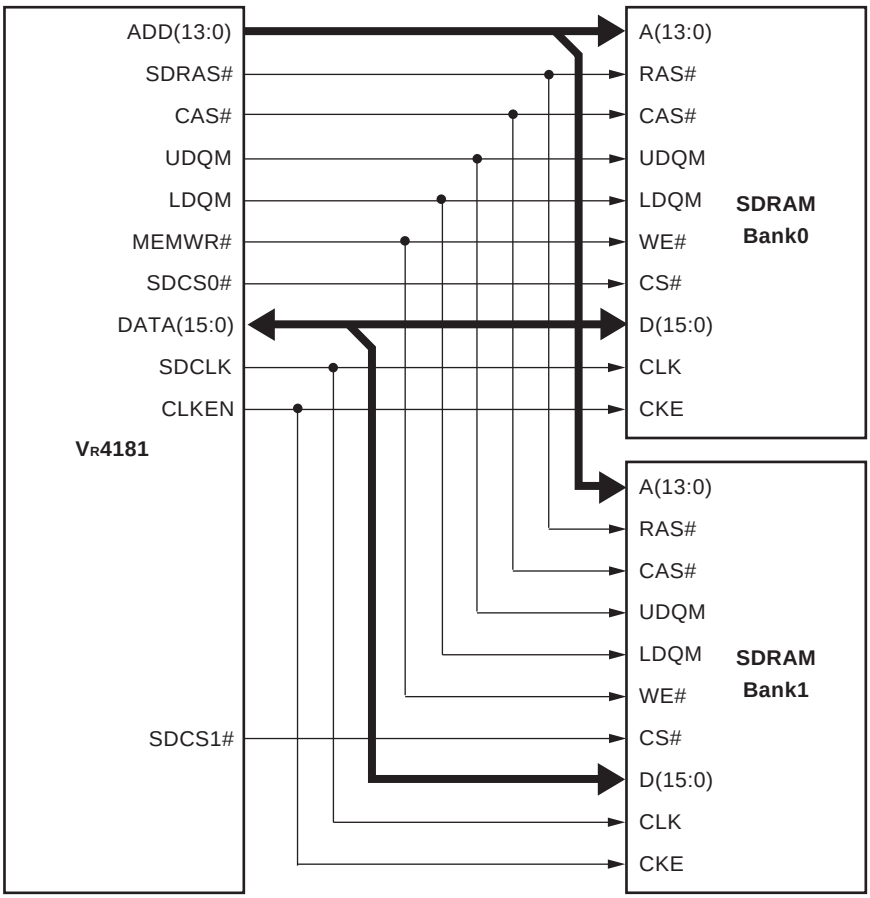
The following table shows examples of EDO DRAM timing parameters when using EDO DRAMs with access time of 60 ns. These parameters are set in EDOMCYTREG register.

TClock frequency	RAS to CAS delay	CAS pulse width	CAS precharge	RAS precharge	RAS pulse width	Self refresh RAS precharge
66 MHz	3 TClock	1 TClock	1 TClock	3 TClock	3 TClock	8 TClock
50 MHz	2 TClock	1 TClock	1 TClock	2 TClock	3 TClock	6 TClock
33 MHz	2 TClock	1/2 TClock	1/2 TClock	2 TClock	2 TClock	4 TClock
25 MHz	2 TClock	1/2 TClock	1/2 TClock	1 TClock	2 TClock	3 TClock

6.4.4 SDRAM configuration

Figure 6-8. SDRAM Configuration

★



★ Figure 6-8 illustrates an example when connecting devices of 4 Mbits x 16.

Remark The SDRAMs supported by the VR4181 are as follows.

Capacity	Configuration	Address pins	Bank address
16 Mbits	512 Kbits x 16 x 2 banks	A(10:0)	A11
64 Mbits	2 Mbits x 16 x 2 banks	A(12:0)	A13
64 Mbits	1 Mbits x 16 x 4 banks	A(11:0)	A(13:12)

6.5 Memory Controller Register Set

Table 6-3. Memory Controller Registers

Physical address	R/W	Register symbol	Function
0x0A00 0300	R/W	EDOMCYTREG	EDO DRAM timing register
0x0A00 0304	R/W	MEMCFG_REG	Memory configuration register
0x0A00 0308	R/W	MODE_REG	SDRAM mode register
0x0A00 030C	R/W	SDTIMINGREG	SDRAM timing register

Caution Since these registers are powered by 2.5 V power supply, the contents of these registers are cleared after Hibernate mode.

6.5.1 EDOMCYTREG (0x0A00 0300)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	SrefRpre2	SrefRpre1	SrefRpre0	Caspre1	Caspre0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
At reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Rcasdly1	Rcasdly0	Tcas1	Tcas0	Trp1	Trp0	Tras1	Tras0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
At reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 13	Reserved	0 is returned when read
★ 12 to 10	SrefRpre(2:0)	Self refresh RAS precharge time 000 : 3 TClock 001 : 4 TClock 010 : 6 TClock 011 : 8 TClock 100 : 11 TClock Others : Reserved
9, 8	Caspre(1:0)	CAS precharge time 00 : 1/2 TClock 01 : 1 TClock 10 : 2 TClock 11 : Reserved
7, 6	Rcasdly(1:0)	RAS to CAS delay time 00 : 2 TClock 01 : 3 TClock 10 : 5 TClock 11 : 6 TClock

(2/2)

Bit	Name	Function
5, 4	Tcas(1:0)	CAS pulse width 00 : 1/2 TClock 01 : 1 TClock 10 : 2 TClock 11 : Reserved
3, 2	Trp(1:0)	RAS precharge time 00 : 1 TClock 01 : 2 TClock 10 : 3 TClock 11 : 4 TClock
1, 0	Tras(1:0)	RAS pulse width 00 : 2 TClock 01 : 3 TClock 10 : 5 TClock 11 : 6 TClock

This register is used to set EDO DRAM timing parameters. Software must set these parameters suitable before using DRAM.

Remark Do not set Tcas = 1/2 TClock and Caspre = 1 TClock, or Tcas = 1 TClock and Caspre = 1/2 TClock at the same time.

6.5.2 MEMCFG_REG (0x0A00 0304)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	Init	Reserved	Reserved	Reserved	B1Config1	B1Config0	Reserved	Bstreftype
R/W	R/W	R	R	R	R/W	R/W	R	R/W
At reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	BstRefr	EDOAsym	Reserved	Reserved	Reserved	B0Config1	B0Config0	EDO/ SDRAM
R/W	R/W	R/W	R	R	R	R/W	R/W	R/W
At reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15	Init	This bit is for SDRAM only. When software writes 1 to this bit, the memory controller issues a SDRAM mode set command. After the SDRAM mode is set, hardware automatically resets this bit to 0. When EDO DRAM is used, this bit must not be set to 1.
14 to 12	Reserved	0 is returned when read
11, 10	B1Config(1:0)	Bank 1 capacity 00 : Bank 1 is not installed 01 : 16 Mbits 10 : 64 Mbits 11 : Reserved
9	Reserved	0 is returned when read
8	Bstreftype	Burst refresh type. This bit determines the number of CBR burst refresh cycles executed before entering and exiting self-refresh mode. 0 : 8 rows refreshed 1 : All rows refreshed
7	BstRefr	Burst refresh enable. This bit enables or disables burst CBR refresh cycles when entering or exiting self-refresh mode. 0 : Disable CBR burst refresh 1 : Enable CBR burst refresh Burst and distributive CBR refresh are mixed if this bit is set to 1. For some kind of DRAMs, mix use of burst and distributive CBR refresh may not be allowed.
6	EDOAsym	EDO DRAM configuration 0 : Asymmetrical 16 Mbit EDO DRAM : 12 rows by 8 columns 64 Mbit EDO DRAM : 13 rows by 9 columns 1 : Symmetrical 16 Mbit EDO DRAM : 10 rows by 10 columns 64 Mbit EDO DRAM : Setting prohibited

(2/2)

Bit	Name	Function
5 to 3	Reserved	0 is returned when read
2, 1	B0Config(1:0)	Bank 0 Capacity 00 : Bank 0 is not installed 01 : 16 Mbit 10 : 64 Mbit 11 : Reserved
0	EDO/SDRAM	DRAM Type 0 : EDO DRAM 1 : SDRAM

This register is used to set DRAM type (capacity, type, organization, etc.) of Bank 0 and Bank 1.

★ **Caution When using SDRAMs, set the Init bit to 1 to initialize SDRAMs before accessing them after an RTC reset or RSTSW reset is canceled or after the VR4181 restores from the Hibernate mode. An initialization of SDRAMs must be executed until the VR4181 issues the first CBR auto refresh cycle.**

Remark During the 64 Mbit SDRAM mode register write, A13 of the address bus is at high level. On the other hand, during the 16 Mbit SDRAM mode register write, A13 is at low level. In order to initialize 64-Mbit SDRAM correctly, software must execute the following sequence.

<1> Set B0Config(1:0) and B1Config(1:0) bits of MEMCFG_REG register to 01
★ <2> Set MODE_REG register to appropriate value (0x00n7, n can be any value)
<3> Initialize SDRAM by setting Init bit of MEMCFG_REG register
<4> Set B0Config(1:0) and B1Config(1:0) bits of MEMCFG_REG register to 10

6.5.3 MODE_REG (0x0A00 0308)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	0	0	BR-SW	TE-Ven1
R/W	R	R	R	R	R/W	R/W	R/W	R/W
At reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	TE-Ven2	LTMode2	LTMode1	LTMode0	WT	BL2	BL1	BL0
★ R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
At reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 12	Reserved	0 is returned when read
11, 10	0	These bits should be always written to 00.
9	BR-SW	Burst read - single write This bit should be always written to 0.
8, 7	TE-Ven(1:2)	These two bits define a JEDEC test cycle and vendor specific cycles. These bits should be always written to 00.
6 to 4 ★	LTMode(2:0)	CAS latency mode [Note] 010 : 2 clocks 011 : 3 clocks Others : Reserved
3	WT	Wrap type for the burst cycles. This bit should be always written to 0. 0 : Sequential (default)
2 to 0	BL(2:0)	Burst length. These bits should be always written to 111. 111 : Full page (When WT = 0 only. Setting prohibited when WT = 1)

Note The CAS latency mode must be set according to the operation frequency of the SDCLK (SDRAM clock).

★ This register is used to set the value output to ADD(13:0) pins during the SDRAM mode register setting cycle. This register should be written before the Init bit of MEMCFG_REG register is set to 1.

6.5.4 SDTIMINGREG (0x0A00 030C)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R/W	R/W
At reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	TRAS1	TRAS0	TRC1	TRC0	TRP1	TRP0	TRCD1	TRCD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
At reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 10	Reserved	0 is returned when read
9	Reserved	Write 0 when write. Note
8	Reserved	Write 1 when write. Note
7, 6	TRAS(1:0)	TRAS in clock cycles 00 : 3 SDCLK (for 25 MHz SDCLK) 01 : 5 SDCLK (for 66, 50, or 33 MHz SDCLK) Others : Prohibited
5, 4	TRC(1:0)	TRC in clock cycles 00 : 4 SDCLK (for 25 MHz SDCLK) 01 : 7 SDCLK (for 66, 50, or 33 MHz SDCLK) Others : Prohibited
3, 2	TRP(1:0)	TRP in clock cycles 00 : 1 SDCLK (for 25 MHz SDCLK) 01 : Prohibited 10 : 3 SDCLK (for 66, 50, or 33 MHz SDCLK) 11 : Prohibited
1, 0	TRCD(1:0)	TRCD in clock cycles 00 : 1 SDCLK (for 25 MHz SDCLK) 01 : 2 SDCLK (for 66, 50, or 33 MHz SDCLK) Others : Prohibited

★ **Note** Bits 9 and 8 must be set to 01 before using SDRAM. Especially, be sure to set 1 to bit 8 since its default value is 0. When these bits are not 01, the VR4181 may not work correctly.

This register is used to set SDRAM timing parameters. Software must set this register suitable before using SDRAM.

6.6 ISA Bridge

The VR4181 has an external bus used for ROM, flash memory, DRAM, and I/O. This bus's operation emulates an ISA bus at accesses to external memory and I/O spaces. The VR4181 also uses an ISA bus internally for the slow, embedded peripherals.

★ Among the pins used for accesses in the external ISA bus, UBE#, IOCS16#, IORDY, IOWR#, and IORD# share the pins with GPIO(20:16), as well as MEMCS16# with LOCLK. To use these pins as an external ISA bus interface, make settings in the GIU in advance.

6.7 ISA Bridge Register Set

The following registers provide configuration and control of the ISA Bridge.

Table 6-4. ISA Bridge Registers

Physical address	R/W	Register symbol	Function
0x0B00 02C0	R/W	ISABRGCTL	ISA Bridge control register
0x0B00 02C2	R/W	ISABRGSTS	ISA Bridge status register
0x0B00 02C4	R/W	XISACTL	External ISA control register

6.7.1 ISABRGCTL (0x0B00 02C0)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	PCLKDIV1	PCLKDIV0
R/W	R	R	R	R	R	R	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 2	Reserved	0 is returned when read
1, 0	PCLKDIV(1:0)	PCLK (peripheral clock) divisor rate selection. These bits select the operating frequency of PCLK. 00 : TClock / 8 01 : TClock / 4 10 : TClock / 2 11 : TClock / 1

★ This register is used to set the PCLK divisor rate. PCLK is a clock for internal ISA peripherals, and its frequency must be set to between 18.432 MHz and 33 MHz.

6.7.2 ISABRGSTS (0x0B00 02C2)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	IDLE
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 1	Reserved	0 is returned when read
0	IDLE	ISA Bridge status 0 : ISA Bridge is busy 1 : ISA Bridge is idle

This register shows the ISA Bridge operation status.

6.7.3 XISACTL (0x0B00 02C4)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	EXTRESULT	INTRESULT	EXBUFFEN
R/W	R	R	R	R	R	R/W	R/W	R/W
RTCRST	0	0	0	0	0	1	0	1
Other resets	0	0	0	0	0	1	0	1

Bit	7	6	5	4	3	2	1	0
Name	MEMWS1	MEMWS0	IOWS1	IOWS0	Reserved	Reserved	SCLKDIV1	SCLKDIV0
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 11	Reserved	0 is returned when read
10	EXTRESULT	External ISA result cycle enable 0 : Disabled. The MBA bus arbiter waits until an external ISA read is finished. 1 : Enabled. The MBA bus arbiter issues a result cycle to the ISA bridge after finishing an external ISA cycle and obtains results of the read. Normally, set 1 to this bit.
9	INTRESULT	Internal ISA result cycle enable 0 : Disabled. The MBA bus arbiter waits until an internal ISA read is finished. 1 : Enabled. The MBA bus arbiter issues a result cycle to the ISA bridge after finishing an internal ISA cycle and obtains results of the read. Normally, set 1 to this bit.
8	EXBUFFEN	External buffer enable 0 : Enable external buffer control with SYSDIR and SYSEN# pins 1 : Disable external buffer control with SYSDIR and SYSEN# pins (SYSEN# and SYSDIR pins are both forced to low level)
7, 6	MEMWS(1:0)	External ISA memory wait states (read/write strobe width) 00 : 1.5 SYSCLK cycles 01 : 2.5 SYSCLK cycles 10 : 3.5 SYSCLK cycles 11 : 4.5 SYSCLK cycles
5, 4	IOWS(1:0)	External ISA I/O wait states (read/write strobe width) 00 : 1.5 SYSCLK cycles 01 : 2.5 SYSCLK cycles 10 : 3.5 SYSCLK cycles 11 : 4.5 SYSCLK cycles

(2/2)

Bit	Name	Function
3, 2	Reserved	0 is returned when read
1, 0	SCLKDIV(1:0)	SYSCLK (external ISA bus clock) divisor rate selection 00 : PCLK / 2 01 : PCLK / 3 10 : PCLK / 6 11 : PCLK / 8

This register is used to set the external ISA configurations.

★ SYSCLK is an operation clock for the external ISA bus, and is output only when an external ISA cycle is generated.

CHAPTER 7 DMA CONTROL UNIT (DCU)

7.1 General

The DMA Control Unit (DCU) controls four channels of DMA transfer. Two of them are allocated for the AIU (microphone and speaker), though the remaining two are reserved for future use.

The Microphone channel performs the I/O-to-memory transfers from the A/D converter included in the AIU to memory. The Speaker channel performs the memory-to-I/O transfers from memory to the D/A converter included in the AIU.

Each DMA channel supports both the primary and the secondary memory buffers. The Source1/Source2 or Destination1/Destination2 Address registers for the associated channel determine the starting address of each memory buffer. The sizes of memory buffers are determined in the associated record length registers.

The DCU uses the primary and secondary DMA buffers alternately when transferring. For example, during the first DMA transfer following either hardware or software reset of the DCU, the transfer starts using the primary DMA buffer. If the total number of DMA transfers through the primary DMA buffer reaches the value set in the associated record length register, the next DMA transfer is performed using the secondary DMA buffer. Software must keep track of which buffer contains valid DMA data.

Software may configure any of the DMA channels to operate in one of two modes; auto-stop or auto-load. When a channel is configured to operate in auto-stop mode, the DCU terminates DMA transfers after the number of transfers specified by the record length register and automatically resets the DMA mask bit for that channel. Once the mask bit is automatically reset, the DCU ignores all subsequent DMA requests for this channel. To resume DMA transfers in this mode, software must again unmask DMA transfers for this channel. Once software unmasks DMA requests, the DCU resumes DMA transfers utilizing the secondary memory buffer.

When a channel is configured to operate in auto-load mode, the DCU does not terminate DMA transfers after the number of DMA transfers specified by the record length register. Instead, the DCU automatically switchs to the secondary DMA buffer and continues servicing DMA requests.

In either mode, auto-stop or auto-load, the DCU always alternates the DMA buffer to be used between the primary and secondary buffers. Software must keep track of the total number of transfers and assure the appropriate DMA buffer is loaded with new DMA data before starting another DMA transfer.

The DCU can be programmed to generate an EOP (end of process) interrupt request independent of auto-stop or auto-load mode. An EOP interrupt request is generated once the number of DMA transfers has reached to the value specified by the record length register.

Priority of each DMA channel is fixed. The channel priority is as follows.

1. AIU Microphone channel
2. AIU Speaker channel

DCU runs at the MBA bus clock (TClock) frequency.

Remark The DCU contains a 32-bit temporary storage register for each DMA channel. For memory-to-I/O transfers, the DCU performs a 32-bit memory read from DRAM and stores the read data into the temporary storage register. The DCU then transfers data from this register to the target I/O device. For a 16-bit device such as the Speaker channel, the DCU performs two I/O writes to the D/A converter for each memory read.

★ During DMA transfers, all DCU registers are write-protected if valid data is present in the temporary storage registers. Because of this, to start DMA transfers, software must read the register that is written immediately after the write to confirm that the register has been correctly set.

7.2 DCU Registers

Table 7-1. DCU Registers

Physical address	R/W	Register symbol	Function
0x0A00 0020	R/W	MICDEST1REG1	Microphone destination 1 address register 1
0x0A00 0022	R/W	MICDEST1REG2	Microphone destination 1 address register 2
0x0A00 0024	R/W	MICDEST2REG1	Microphone destination 2 address register 1
0x0A00 0026	R/W	MICDEST2REG2	Microphone destination 2 address register 2
0x0A00 0028	R/W	SPKRSRC1REG1	Speaker source 1 address register 1
0x0A00 002A	R/W	SPKRSRC1REG2	Speaker source 1 address register 2
0x0A00 002C	R/W	SPKRSRC2REG1	Speaker source 2 address register 1
0x0A00 002E	R/W	SPKRSRC2REG2	Speaker source 2 address register 2
0x0A00 0040	R/W	DMARSTREG	DMA reset register
0x0A00 0046	R/W	AIUDMAMSKREG	Audio DMA mask register
0x0A00 0600 to 0x0A00 0654	R/W	–	Reserved. Write 0 when write. 0 is returned after a read.
0x0A00 0658	R/W	MICRCLENREG	Microphone record length register
0x0A00 065A	R/W	SPKRCLENREG	Speaker record length register
0x0A00 065C	R/W	–	Reserved. Write 0 when write. 0 is returned after a read.
0x0A00 065E	R/W	MICDMACFGREG	Microphone DMA configuration register
0x0A00 0660	R/W	SPKDMACFGREG	Speaker DMA configuration register
0x0A00 0662	R/W	DMAITRQREG	DMA interrupt request register
0x0A00 0664	R/W	DMACTLREG	DMA control register
0x0A00 0666	R/W	DMAITMKREG	DMA interrupt mask register

7.2.1 Microphone destination 1 address registers

(1) MICDEST1REG1 (0x0A00 0020)

Bit	15	14	13	12	11	10	9	8
Name	MD1A15	MD1A14	MD1A13	MD1A12	MD1A11	MD1A10	MD1A9	MD1A8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
At reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	MD1A7	MD1A6	MD1A5	MD1A4	MD1A3	MD1A2	MD1A1	MD1A0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
At reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 0	MD1A(15:0)	Lower 16 bits (A(15:0)) of DMA destination 1 address for Microphone

(2) MICDEST1REG2 (0x0A00 0022)

Bit	15	14	13	12	11	10	9	8
Name	MD1A31	MD1A30	MD1A29	MD1A28	MD1A27	MD1A26	MD1A25	MD1A24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
At reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	MD1A23	MD1A22	MD1A21	MD1A20	MD1A19	MD1A18	MD1A17	MD1A16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
At reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 0	MD1A(31:16)	Upper 16 bits (A(31:16)) of DMA destination 1 address for Microphone

These two registers specify the destination memory address of the primary DMA buffer for the Microphone channel.

7.2.2 Microphone destination 2 address registers

(1) MICDEST2REG1 (0x0A00 0024)

Bit	15	14	13	12	11	10	9	8
Name	MD2A15	MD2A14	MD2A13	MD2A12	MD2A11	MD2A10	MD2A9	MD2A8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
At reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	MD2A7	MD2A6	MD2A5	MD2A4	MD2A3	MD2A2	MD2A1	MD2A0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
At reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 0	MD2A(15:0)	Lower 16 bits (A(15:0)) of DMA destination 2 address for Microphone

(2) MICDEST2REG2 (0x0A00 0026)

Bit	15	14	13	12	11	10	9	8
Name	MD2A31	MD2A30	MD2A29	MD2A28	MD2A27	MD2A26	MD2A25	MD2A24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
At reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	MD2A23	MD2A22	MD2A21	MD2A20	MD2A19	MD2A18	MD2A17	MD2A16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
At reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 0	MD2A(31:16)	Upper 16 bits (A(31:16)) of DMA destination 2 address for Microphone

These two registers specify the destination memory address of the secondary DMA buffer for the Microphone channel.

7.2.3 Speaker source 1 address registers

(1) SPKRSRC1REG1 (0x0A00 0028)

Bit	15	14	13	12	11	10	9	8
Name	SS1A15	SS1A14	SS1A13	SS1A12	SS1A11	SS1A10	SS1A9	SS1A8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
At reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	SS1A7	SS1A6	SS1A5	SS1A4	SS1A3	SS1A2	SS1A1	SS1A0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
At reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 0	SS1A(15:0)	Lower 16 bits (A(15:0)) of DMA source 1 address for Speaker

(2) SPKRSRC1REG2 (0x0A00 002A)

Bit	15	14	13	12	11	10	9	8
Name	SS1A31	SS1A30	SS1A29	SS1A28	SS1A27	SS1A26	SS1A25	SS1A24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
At reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	SS1A23	SS1A22	SS1A21	SS1A20	SS1A9	SS1A18	SS1A17	SS1A16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
At reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 0	SS1A(31:16)	Upper 16 bits (A(31:16)) of DMA source 1 address for Speaker

These two registers specify the source memory address of the primary DMA buffer for the Speaker channel.

7.2.4 Speaker source 2 address registers

(1) SPKRSRC2REG1 (0x0A00 002C)

Bit	15	14	13	12	11	10	9	8
Name	SS2A15	SS2A14	SS2A13	SS2A12	SS2A11	SS2A10	SS2A9	SS2A8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
At reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	SS2A7	SS2A6	SS2A5	SS2A4	SS2A3	SS2A2	SS2A1	SS2A0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
At reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 0	SS2A(15:0)	Lower 16 bits (A(15:0)) of DMA source 2 address for Speaker

(2) SPKRSRC2REG2 (0x0A00 002E)

Bit	15	14	13	12	11	10	9	8
Name	SS2A31	SS2A30	SS2A29	SS2A28	SS2A27	SS2A26	SS2A25	SS2A24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
At reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	SS2A23	SS2A22	SS2A21	SS2A20	SS2A9	SS2A18	SS2A17	SS2A16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
At reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 0	SS2A(31:16)	Upper 16 bits (A(31:16)) of DMA source 2 address for Speaker

These two registers specify the source memory address of the secondary DMA buffer for the Speaker channel.

7.2.5 DMARSTREG (0x0A00 0040)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
At reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	DMARST
R/W	R	R	R	R	R	R	R	R/W
At reset	0	0	0	0	0	0	0	1

Bit	Name	Function
15 to 1	Reserved	0 is returned after a read.
0	DMARST	Resets DMA functions 0 : Resets DMA channels 1 : Normal operation

When DMARST bit is written to zero, all active DMA transfers are immediately terminated and the DCU enters in the reset state. While DMARST bit is 0, all DMA requests become pending until this bit is set to 1.

7.2.6 AIUDMAMSKREG (0x0A00 0046)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
At reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	MICMSK	SPKMSK	Reserved	Reserved
R/W	R	R	R	R	R/W	R/W	R	R
At reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 4	Reserved	0 is returned after a read.
3	MICMSK	Masks DMA for Microphone (audio input) channel 0 : Microphone channel disabled 1 : Microphone channel enabled
2	SPKMSK	Masks DMA for Speaker (audio output) channel 0 : Speaker channel disabled 1 : Speaker channel enabled
1, 0	Reserved	0 is returned after a read.

7.2.7 MICRCLENREG (0x0A00 0658)

Bit	15	14	13	12	11	10	9	8
Name	MICRL15	MICRL14	MICRL13	MICRL12	MICRL11	MICRL10	MICRL9	MICRL8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
★ At reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
Name	MICRL7	MICRL6	MICRL5	MICRL4	MICRL3	MICRL2	MICRL1	MICRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
★ At reset	1	1	1	1	1	1	1	1

Bit	Name	Function
15 to 0	MICRL(15:0)	DMA Record Length for Microphone. MICRL0 bit must be written to zero.

This register defines the number of 16-bit words to be transferred during DMA operation in the Microphone channel.

7.2.8 SPKRCLENREG (0x0A00 065A)

Bit	15	14	13	12	11	10	9	8
Name	SPKRL15	SPKRL14	SPKRL13	SPKRL12	SPKRL11	SPKRL10	SPKRL9	SPKRL8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
★ At reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
Name	SPKRL7	SPKRL6	SPKRL5	SPKRL4	SPKRL3	SPKRL2	SPKRL1	SPKRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
★ At reset	1	1	1	1	1	1	1	1

Bit	Name	Function
15 to 0	SPKRL(15:0)	DMA Record Length for Speaker. SPKRL0 bit must be written to zero.

This register defines the number of 16-bit words to be transferred during DMA operation in the Speaker channel.

7.2.9 MICDMACFGREG (0x0A00 065E)

Bit	15	14	13	12	11	10	9	8
★ Name	Reserved	MicDsize1	MicDsize0	MicSrctype	MicDestype	Reserved	Reserved	MicLoad
R/W	R	R	R	R	R	R	R	R/W
At reset	0	0	1	1	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
At reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15	Reserved	0 is returned after a read.
14, 13	MicDsize(1:0)	Indicates Microphone channel data size 01 : 16 bits Values other than above do not appear.
12	MicSrctype	Indicates Microphone channel source address type 1 : I/O 0 does not appear.
11	MicDestype	Indicates Microphone channel destination address type 0 : Memory 1 does not appear.
10, 9	Reserved	0 is returned after a read.
8	MicLoad	DMA auto-stop/auto-load mode setting for Microphone channel 0 : Auto-stop 1 : Auto-load When this bit is set to 1, the DCU automatically begins transferring data to the secondary buffer when the primary buffer is full. ★ When this bit is set to 0, the DCU uses the primary buffer only.
7 to 0	Reserved	0 is returned after a read.

7.2.10 SPKDMACFGREG (0x0A00 0660)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
At reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	SpkDsize1	SpkDsize0	SpkSrctype	SpkDestype	Reserved	Reserved	SpkLoad
R/W	R	R	R	R	R	R	R	R/W
At reset	0	0	1	0	1	0	0	0

Bit	Name	Function
15 to 7	Reserved	0 is returned after a read.
6, 5	SpkDsize(1:0)	Indicates Speaker channel data size 01 : 16 bits Values other than above do not appear.
4	SpkSrctype	Indicates Speaker channel source address type 0 : Memory 1 does not appear.
3	SpkDestype	Indicates Speaker channel destination address type 1 : I/O 0 does not appear.
2, 1	Reserved	0 is returned after a read.
0	SpkLoad	DMA auto-stop/auto-load mode setting for Speaker channel 0 : Auto-stop 1 : Auto-load When this bit is set to 1, the DCU automatically begins transferring data from the secondary buffer when the primary buffer is empty. When this bit is set to 0, the DCU uses the primary buffer only.

★

7.2.11 DMAITRQREG (0x0A00 0662)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
At reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	SpkEOP	MicEOP	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R/W	R/W	R	R/W	R/W	R
At reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 6	Reserved	0 is returned after a read.
5	SpkEOP	Speaker channel end of process (EOP) interrupt status 0 : None 1 : Speaker channel EOP interrupt pending ★ The interrupt request is cleared when this bit is written to 1.
4	MicEOP	Microphone channel EOP interrupt status 0 : None 1 : Microphone channel EOP interrupt pending ★ The interrupt request is cleared when this bit is written to 1.
3	Reserved	0 is returned after a read.
2, 1	Reserved	Write 0 when write. 0 is returned after a read.
0	Reserved	0 is returned after a read.

This register indicates interrupt status of each DMA channel by end of process (EOP). Once an interrupt occurs, clear the interrupt request by writing a zero to the corresponding status bit in this register.

7.2.12 DMACTLREG (0x0A00 0664)

Bit	15	14	13	12	11	10	9	8
Name	SpkCNT1	SpkCNT0	MicCNT1	MicCNT0	Reserved	Reserved	Reserved	Reserved
R/W	R/W	R/W	R/W	R/W	R	R	R	R
At reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
At reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15, 14	SpkCNT(1:0)	Speaker channel source address count control 00 : Increment 01 : Decrement Others : Reserved
13, 12	MicCNT(1:0)	Microphone channel destination address count control 00 : Increment 01 : Decrement Others : Reserved
11 to 8	Reserved	0 is returned after a read.
7 to 0	Reserved	Write 0 when write. 0 is returned after a read.

7.2.13 DMAITMKREG (0x0A00 0666)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
At reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	SpkEOPMsk	MicEOPMsk	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R/W	R/W	R	R/W	R/W	R
At reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 6	Reserved	0 is returned after a read.
5	SpkEOPMsk	Speaker channel end of process (EOP) interrupt mask 0 : Disable 1 : Enable
4	MicEOPMsk	Microphone channel EOP interrupt mask 0 : Disable 1 : Enable
3	Reserved	0 is returned after a read.
2, 1	Reserved	Write 0 when write. 0 is returned after a read.
0	Reserved	0 is returned after a read.

CHAPTER 8 CLOCKED SERIAL INTERFACE UNIT (CSI)

8.1 Overview

The CSI manages communication via a synchronous serial bus. The CSI of the VR4181 has the following key characteristics:

- Slave-only synchronous serial interface
- Able to transmit and receive data simultaneously
- Supports fixed 8-bit character length
★ - Supports burst lengths of 1 to 65535 bits
- Continuous transfer mode for of peripherals supporting auto-scan
- Programmable clock phase and clock polarity

The CSI interface shares pins with GPIO signals as follows. When using the CSI, set these pins to use as CSI signals in the registers of the GIU in advance.

GPIO Pin	CSI Signal	Definition
GPIO10	FRM	Optional multifunction control input. In one mode, FRM determines data direction (transmit or receive). In the other mode, FRM enables (low level) or inhibits (high level) transmissions.
GPIO2	SCK	Serial clock input (Maximum frequency: 1.6 MHz)
GPIO1	SO	Serial data output
GPIO0	SI	Serial data input

★ **Caution No clock is supplied to the CSI in the initial state. When using the CSI, set the MSKCSUPCLK bit of the CMUCLKMSK register in the MBA Host Bridge to 1 in advance so that a clock is supplied.**

8.2 Operation of CSI

8.2.1 Transmit/receive operations

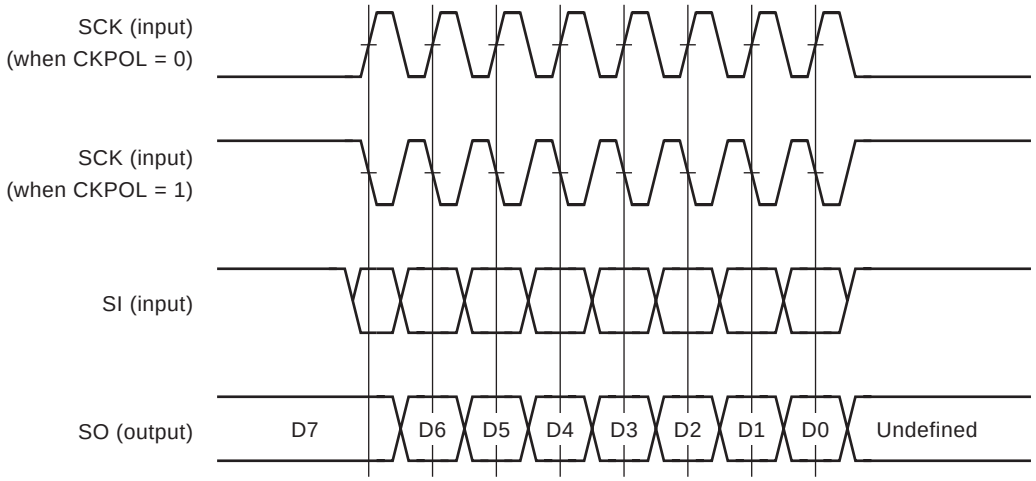
Transmit and receive operations are initiated by an external master to drive the serial clock, SCK. The characteristics of the protocol are controlled by the CSIMODE register, in particular by CKPOL, CKMD, FRMEN, and FRMMD bits. CKPOL and CKMD bits control the relationship between data driven on SO and SI, and the phase of the serial clock input to SCK. FRMEN and FRMMD bits enable and control the FRM input.

8.2.2 SCK phase and CSI transfer timing

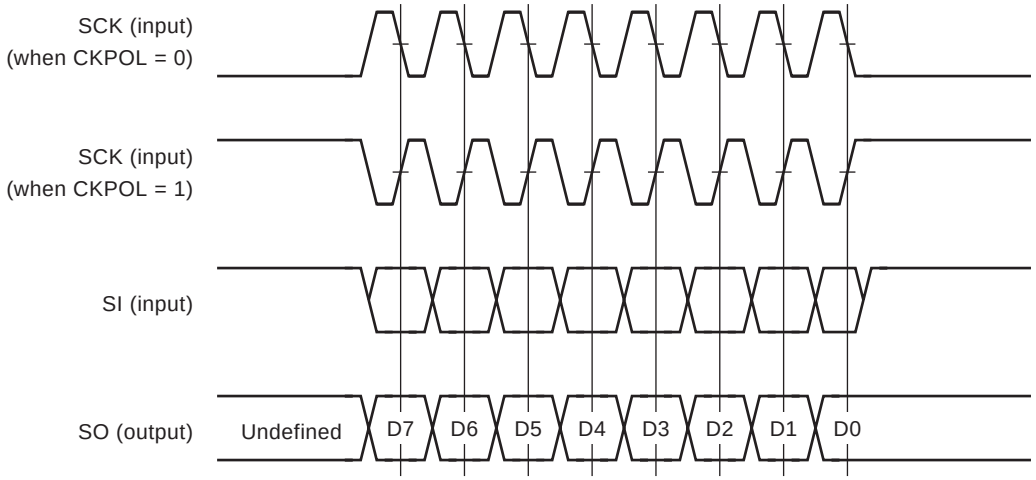
The external master drives SCK and SI and samples data driven on SO. The CSI supports 4 basic operating modes of SCK depending on the settings of CKPOL and CKMD bits. These are illustrated in the following figure.

Figure 8-1. SCK and SI/SO Relationship

(a) When CKMD bit = 0



(b) When CKMD bit = 1



★ **Caution When the CKMD bit is set to 1, the next byte data is output during the latter half of the cycle for the eighth bit of a transmit data.**

This figure illustrates CSI cycles when the FRM input is disabled (FRMEN bit = 0) or configured to provide direction control (FRMEN bit = 1 and FRMMD bit = 0). When FRMEN bit = 1 and FRMMD bit = 1, SO is driven as high impedance during a high level input to FRM.

★ In addition, this figure illustrates the CSI cycles when bit 7 of a data is transmitted or received first (i.e. when the LSBMSB bit of the CSIMODE register = 0).

The four modes of SCK are described below.

(1) When CKMD bit = 0 and CKPOL bit = 0

- Transmission

  The first transmit data bit is output before the first rising edge of SCK.

  The second transmit data and those that follow are output synchronized with the falling edge of SCK. Therefore, the external master must sample the data synchronizing with the rising edge of SCK.

★ - Reception

  The external master must output the first data bit before the first rising edge of SCK.

  The VR4181 samples receive data synchronizing with the rising edge of SCK. Therefore, the external master must output data synchronizing with the falling edge of SCK.

(2) When CKMD bit = 0 and CKPOL bit = 1

- Transmission

  The first transmit data bit is output before the first falling edge of SCK.

  The second transmit data bit and those that follow are output synchronized with the rising edge of SCK. Therefore, the external master must sample the data synchronizing with the falling edge of SCK.

★ - Reception

  The external master must output the first data bit before the first falling edge of SCK.

  The VR4181 samples receive data synchronizing with the falling edge of SCK. Therefore, the external master must output data synchronizing with the rising edge of SCK.

★ **(3) When CKMD bit = 1 and CKPOL bit = 0**

- Transmission

  The first transmit data bit is output synchronized with the first rising edge of SCK.

  The second transmit data bit and those that follow are output synchronized with the rising edge of SCK. Therefore, the external master must sample the data synchronizing with the falling edge of SCK.

- Reception

  The VR4181 samples receive data synchronizing with the falling edge of SCK. Therefore, the external master must output data synchronizing with the rising edge of SCK.

★ **(4) When CKMD bit = 1 and CKPOL bit = 1**

- Transmission

  The first transmit data bit is output synchronized with the first falling edge of SCK.

  The second transmit data bit and those that follow are output synchronized with the falling edge of SCK. Therefore, the external master must sample the data synchronizing with the rising edge of SCK.

- Reception

  The VR4181 samples receive data synchronizing with the rising edge of SCK. Therefore, the external master must output data synchronizing with the falling edge of SCK.

8.2.3 CSI transfer types

(1) Burst mode

Burst mode is supported for both transmit and receive transfers. Burst lengths for transmission and reception are independently programmable and can be set from 1 to 65535 bits. The transmit and receive shift registers are both 8-bit lengths. During burst mode, when the receive shift register goes “full”, the data is automatically transferred to the receive FIFO. When the transmit shift register goes “empty”, it is automatically reloaded from the transmit FIFO.

Once the burst length has been set and the burst transaction enabled, the CSI behaves as follows:

The CSI begins tracking the number of bits transmitted and/or received. At the end of each bit transfer, the bit count is updated and compared to the corresponding burst length value (transmit and/or receive). If the number of bits transferred is equal to the burst length, the CSI shift register is halted.

If the transfer is a reception, the contents of the shift register will be copied to the receive FIFO, a Receive Burst End interrupt request will be generated if unmasked, and additional activities on the SCK input will be ignored.

If the transfer is a transmission, a Transmit Burst End interrupt request will be generated if unmasked and
★ additional SCK cycles will cause an invalid data to be output on SO.

(2) Continuous mode

Continuous mode transfers are always defined as 8-bit fixed length transfers. In continuous mode, software must control the flow of data between the VR4181 and the external master.

When continuous mode is enabled and the receive shift register goes “full”, the data is automatically transferred to the receive FIFO. When the transmit shift register goes “empty”, it is automatically reloaded from the transmit FIFO.

8.2.4 Transmit and receive FIFOs

The CSI contains two 8-deep 16-bit FIFOs. One is for transmission and the other for reception. The transmit and receive shift registers access the FIFOs by 8 bits at a time. The CPU core accesses the FIFOs in either 8-bit or 16-bit units.

The threshold of each FIFO is independently programmable. For the transmit FIFO, an interrupt request is generated to inform the CPU that 1, 2, or 4 16-bit words are empty in the FIFO. For the receive FIFO, an interrupt request is generated to inform the CPU core that 1, 2, or 4 16-bit words can be read from the FIFO.

The FIFO control logic can also generate interrupt requests to signal an overrun condition for the receive FIFO or an underrun condition for the transmit FIFO. An overrun occurs when the receive shift register attempts to transfer data to a location in the FIFO which has not be read by the CPU core. An underrun occurs when the transmit shift register attempts to load a value from the FIFO which has not been updated by the CPU core.

(1) Overrun/underrun errors

When an overrun error occurs, the receive FIFO logic generates an overrun interrupt request if unmasked, and overwrites the next location in the FIFO with the contents of the receive shift register.

When an underrun error occurs, the transmit FIFO logic generates an underrun interrupt request if unmasked, and reloads the transmit shift register with the contents of the next location in the FIFO.

The software must recover the data loss caused by the overrun or underrun error.

8.3 CSI Registers

The CSI provides the following registers:

Table 8-1. CSI Registers

Physical address	R/W	Register symbol	Function
0x0B00 0900	R/W	CSIMODE	CSI mode register
0x0B00 0902	R	CSIRXDATA	CSI receive data register
0x0B00 0904	R/W	CSITXDATA	CSI transmit data register
0x0B00 0906	R/W	CSILSTAT	CSI line status register
0x0B00 0908	R/W	CSIINTMSK	CSI interrupt mask register
0x0B00 090A	R/W	CSIINTSTAT	CSI interrupt status register
0x0B00 090C	R/W	CSITXBLEN	CSI transmit burst length register
0x0B00 090E	R/W	CSIRXBLEN	CSI receive burst length register

8.3.1 CSIMODE (0x0B00 0900)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	FRMEN	TXEN	TXBMD	TXCLR	Reserved	RXEN	RXBMD	RXCLR
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
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	FRMMD	CKPOL	CKMD	Reserved	Reserved	Reserved	Reserved	LSBMSB
R/W	R/W	R/W	R/W	R	R	R	R	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15	FRMEN	CSI FRM enable 0 : Disabled. FRM signal input is ignored. 1 : Enabled. Mode is set by FRMMD bit.
14	TXEN	CSI transmit enable 0 : Disable 1 : Enable ★ **Remark** When using the transmit function only, communication must be performed with the RXEN bit = 0 and the RXCLR bit = 1.
13	TXBMD	CSI transmit burst mode 0 : Continuous mode 1 : Burst mode
12	TXCLR	CSI transmit buffer clear 0 : Enable transmit shift register and FIFO ★ 1 : Reset transmit shift register and FIFO
11	Reserved	0 is returned after read
10	RXEN	CSI receive enable 0 : Disable 1 : Enable ★ **Remark** When using the receive function only, communication must be performed with the TXEN bit = 0 and the TXCLR bit = 1.
9	RXBMD	CSI receive burst mode 0 : Continuous mode 1 : Burst mode
8	RXCLR	CSI receive buffer clear 0 : Enable receive shift register and FIFO ★ 1 : Reset receive shift register and FIFO

(2/2)

Bit	Name	Function
7	FRMMD	FRM mode 0 : FRM controls transfer directions (receive when FRM= 1, transmit when FRM= 0) 1 : FRM enables transfers (transmit/receive enabled when FRM = 0)
6	CKPOL	CSI clock polarity [Note] 0 : SCK is active high (1st transition is low to high) 1 : SCK is active low (1st transition is high to low)
5	CKMD	CSI clocking mode [Note] 0 : Character data is valid prior to the 1st transition of SCK 1 : Character data is valid at the 1st transition of SCK
4 to 1	Reserved	0 is returned after read
0	LSBMSB	Transmit/receive mode bit ordering 0 : Bit 7 is the first bit transmitted or received (MSB mode) 1 : Bit 0 is the first bit transmitted or received (LSB mode)

Note The TXCLR and RXCLR bits must be cleared after changing the CKPOL or CKMD bit.

★ The CKPOL bit must be set as follows according to the state of SCK when a communication is not performed:

- When SCK is at low level during no communication … CKPOL bit = 0
- When SCK is at high level during no communication … CKPOL bit = 1

8.3.2 CSIRXDATA (0x0B00 0902)

Bit	15	14	13	12	11	10	9	8
Name	RXD15	RXD14	RXD13	RXD12	RXD11	RXD10	RXD9	RXD8
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

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
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 0	RXD(15:0)	CSI receive data. CSI data received on the SI pin is read through these data bits.

8.3.3 CSITXDATA (0x0B00 0904)

Bit	15	14	13	12	11	10	9	8
Name	TXD15	TXD14	TXD13	TXD12	TXD11	TXD10	TXD9	TXD8
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
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	TXD7	TXD6	TXD5	TXD4	TXD3	TXD2	TXD1	TXD0
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
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 0	TXD(15:0)	CSI transmit data. CSI data written to these bits is transmitted on the SO pin.

8.3.4 CSILSTAT (0x0B00 0906)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	TFIFOT1	TFIFOT0	Reserved	Reserved	Reserved	TXFIFOF	TXFIFOE	TXBUSY
R/W	R/W	R/W	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	1	0
Other resets	0	0	0	0	0	0	1	0

Bit	7	6	5	4	3	2	1	0
Name	RFIFOT1	RFIFOT0	Reserved	FRMDIR	Reserved	RXFIFOF	RXFIFOE	RXBUSY
R/W	R/W	R/W	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15, 14	TFIFOT(1:0)	CSI transmit FIFO threshold. These bits select the level at which the transmit FIFO empty status is notified. 00 : 1 or more words are free in transmit FIFO 01 : 2 or more words are free in transmit FIFO 10 : 4 or more words are free in transmit FIFO 11 : Reserved
13 to 11	Reserved	0 is returned after read
10	TXFIFOF	CSI transmit FIFO full status. This bit is set to 1 when the transmit FIFO contains no free space. 0 : Transmit FIFO not full 1 : Transmit FIFO full
9	TXFIFOE	CSI transmit FIFO empty status. This bit is set to 1 when the transmit FIFO reaches to the empty level defined by TFIFOT bits. 0 : Transmit FIFO not empty 1 : Transmit FIFO empty
8	TXBUSY	CSI transmit shift register status 0 : Idle 1 : Character transmission in progress
7, 6	RFIFOT(1:0)	CSI receive FIFO threshold. These bits select the level at which the receive FIFO full status is notified. 00 : 1 or more words are valid in receive FIFO 01 : 2 or more words are valid in receive FIFO 10 : 4 or more words are valid in receive FIFO 11 : Reserved
5	Reserved	0 is returned after read

(2/2)

Bit	Name	Function
4	FRMDIR	FRM input pin status 0 : Low level (transmit direction) 1 : High level (receive direction)
3	Reserved	0 is returned after read
2	RXFIFOF	CSI receive FIFO full status. This bit is set to 1 when the receive FIFO reaches to the full level defined by RFIFOT bits. 0 : Receive FIFO not full 1 : Receive FIFO full
1	RXFIFOE	CSI receive FIFO empty status. This bit is set to 1 when the receive FIFO contains no valid data. 0 : Receive FIFO not empty 1 : Receive FIFO empty
0	RXBUSY	CSI receive shift register status 0 : Idle 1 : Character reception in progress

8.3.5 CSIINTMSK (0x0B00 0908)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	MUNDRN	MTXBEND	MTXFIFOE	MTXBUSY
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	1	1	1	1
Other resets	0	0	0	0	1	1	1	1

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	MOVRRN	MRXBEND	MRXFIFOF	MRXBUSY
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	1	1	1	1
Other resets	0	0	0	0	1	1	1	1

Bit	Name	Function
15 to 12	Reserved	0 is returned after read
11	MUNDRN	Mask of transmit FIFO underrun interrupt requests 0 : Unmasked 1 : Masked
10	MTXBEND	Mask of Transmit Burst End interrupt requests 0 : Unmasked 1 : Masked
9	MTXFIFOE	Mask of Transmit FIFO Empty interrupt requests 0 : Unmasked 1 : Masked
8	MTXBUSY	Mask of Transmit Shift Register Busy interrupt requests 0 : Unmasked 1 : Masked
7 to 4	Reserved	0 is returned after read
3	MOVRRN	Mask of Receive FIFO Overrun interrupt requests 0 : Unmasked 1 : Masked
2	MRXBEND	Mask of Receive Burst End interrupt requests 0 : Unmasked 1 : Masked
1	MRXFIFOF	Mask of Receive FIFO Full interrupt requests 0 : Unmasked 1 : Masked
0	MRXBUSY	Mask of Receive Shift Register Busy interrupt requests 0 : Unmasked 1 : Masked

8.3.6 CSIINTSTAT (0x0B00 090A)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	URNINT	TXBEINT	TXFEINT	TXBSYINT
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	ORNINT	RXBEINT	RXFFINT	RXBSYINT
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 12	Reserved	0 is returned after read
11	URNINT	Transmit FIFO Underrun interrupt request status 0 : Not pending 1 : Pending This bit is cleared by writing 1.
10	TXBEINT	Transmit Burst End interrupt request status 0 : Not pending 1 : Pending This bit is cleared by writing 1.
9	TXFEINT	Transmit FIFO Empty interrupt request status 0 : Not pending 1 : Pending This bit is cleared by writing 1.
8	TXBSYINT	Transmit Shift Register Busy interrupt request status 0 : Not pending 1 : Pending This bit is cleared by writing 1.
7 to 4	Reserved	0 is returned after read
3	ORNINT	Receive FIFO Overrun interrupt request status 0 : Not pending 1 : Pending This bit is cleared by writing 1.

(2/2)

Bit	Name	Function
2	RXBEINT	Receive Burst End interrupt request status 0 : Not pending 1 : Pending This bit is cleared by writing 1.
1	RXFFINT	Receive FIFO Full interrupt request status 0 : Not pending 1 : Pending This bit is cleared by writing 1.
0	RXBSYINT	Receive Shift Register Busy interrupt request status 0 : Not pending 1 : Pending This bit is cleared by writing 1.

8.3.7 CSITXBLEN (0x0B00 090C)

Bit	15	14	13	12	11	10	9	8
Name	TXBLN15	TXBLN14	TXBLN13	TXBLN12	TXBLN11	TXBLN10	TXBLN9	TXBLN8
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
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	TXBLN7	TXBLN6	TXBLN5	TXBLN4	TXBLN3	TXBLN2	TXBLN1	TXBLN0
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
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 0	TXBLN(15:0)	Transmit burst length. These bits determine the number of bits transmitted during one burst cycle. 0x0000 : Reserved 0x0001 : 1 bit 0x0002 : 2 bits : : 0x00FD : 253 bits 0x00FE : 254 bits 0x00FF : 255 bits : : 0xFFFD : 65533 bits 0xFFFE : 65534 bits 0xFFFF : 65535 bits

8.3.8 CSIRXBLEN (0x0B00 090E)

Bit	15	14	13	12	11	10	9	8
Name	RXBLN15	RXBLN14	RXBLN13	RXBLN12	RXBLN11	RXBLN10	RXBLN9	RXBLN8
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
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RXBLN7	RXBLN6	RXBLN5	RXBLN4	RXBLN3	RXBLN2	RXBLN1	RXBLN0
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
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 0	RXBLN(15:0)	Receive burst length. These bits determine the number of bits received during one burst cycle. 0x0000 : Reserved 0x0001 : 1 bit 0x0002 : 2 bits : : 0x00FD : 253 bits 0x00FE : 254 bits 0x00FF : 255 bits : : 0xFFFD : 65533 bits 0xFFFE : 65534 bits 0xFFFF : 65535 bits

CHAPTER 9 INTERRUPT CONTROL UNIT (ICU)

9.1 Overview

The ICU collects interrupt requests from the various on-chip peripheral units and transfers them with internal interrupt request signals (Int0, Int1, Int2, Int3, Int4, and NMI) to the CPU core.

The signals used to notice interrupt requests to the CPU are as below.

NMI: battint only. However, the signal for battint can be switched between NMI and Int0 is enabled according to NMIREG register's settings. Because NMI's interrupt masking cannot be controlled by means of software, switch to Int0 to mask battint.

★ Int4: Not used (fixed to 1 (inactive))

Int3: Not used (fixed to 1 (inactive))

Int2: rtclong2 only (RTCLong2 Timer)

Int1: rtclong1 only (RTCLong1 Timer)

Int0: All other interrupts.

For details of the interrupt sources, see **9.2 Register Set**.

How an interrupt request is notified to the CPU core is shown below.

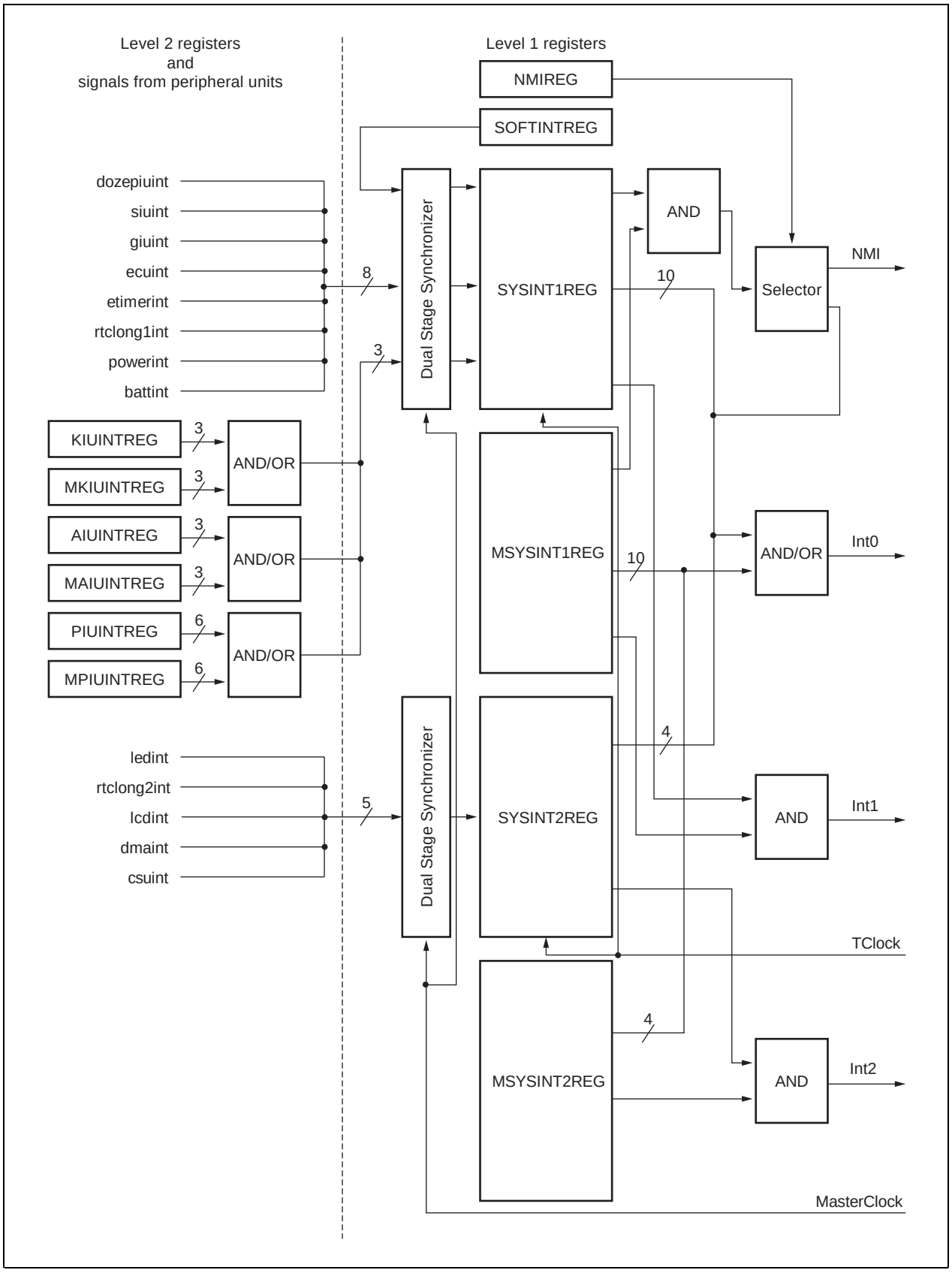
If an interrupt request occurs in the peripheral units, the corresponding bit in the interrupt indication register of Level 2 (xxxINTREG) is set to 1. The interrupt indication register is ANDed bit-wise with the corresponding interrupt mask register of Level 2 (MxxxINTREG). If the occurred interrupt request is enabled (set to 1) in the mask register, the interrupt request is notified to the interrupt indication register of Level 1 (SYSINTREG) and the corresponding bit is set to 1. At this time, the interrupt requests from the same register of Level 2 are notified to the SYSINTREG as a single interrupt request.

Interrupt requests from some units directly set their corresponding bits in the SYSINTREG.

The SYSINTREG is ANDed bit-wise with the interrupt mask register of Level 1 (MSYSINTREG). If the interrupt request is enabled (set to 1) in the MSYSINTREG, a corresponding interrupt request signal is output from the ICU to the CPU core. battintr is connected to the NMI or Int0 signal of the CPU core (selected by setting of NMIREG). rtclong2 and rtclong1 signals are connected to the Int2 or Int1 signal of the CPU core. 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 9-1. Outline of Interrupt Control



9.2 Register Set

Table 9-1. ICU Registers

	Physical address	R/W	Register symbol	Function
	0x0A00 0080	R	SYSINT1REG	Level 1 system register 1
	0x0A00 008C	R/W	MSYNT1REG	Level 1 mask system register 1
	0x0A00 0098	R/W	NMIREG	NMI register
	0x0A00 009A	R/W	SOFTINTREG	Software interrupt register
	0x0A00 0200	R	SYSINT2REG	Level 1 system register 2
	0x0A00 0206	R/W	MSYSINT2REG	Level 1 mask system register 2
	0x0B00 0082	R	PIUINTREG	Level 2 PIU register
	0x0B00 0084	R	AIUINTREG	Level 2 AIU register
★	0x0B00 0086	R/W	KIUINTREG	Level 2 KIU register
	0x0B00 008E	R/W	MPIUINTREG	Level 2 mask PIU register
★	0x0B00 0090	W	MAIUINTREG	Level 2 mask AIU register
	0x0B00 0092	R/W	MKIUINTREG	Level 2 mask KIU register

9.2.1 SYSINT1REG (0x0A00 0080)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	DOZEPIU INTR	Reserved	SOFTINTR	Reserved	SIUINTR	GIUINTR
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	KIUINTR	AIUINTR	PIUINTR	Reserved	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
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15, 14	Reserved	0 is returned when read
13	DOZEPIUINTR	PIU interrupt request during Suspend mode 0 : Not occurred 1 : Occurred
12	Reserved	0 is returned when read
11	SOFTINTR	Software interrupt request 0 : Not occurred 1 : Occurred
10	Reserved	0 is returned when read
9	SIUINTR	SIU interrupt request 0 : Not occurred 1 : Occurred
8	GIUINTR	GIU interrupt request 0 : Not occurred 1 : Occurred
7	KIUINTR	KIU interrupt request 0 : Not occurred 1 : Occurred
6	AIUINTR	AIU interrupt request 0 : Not occurred 1 : Occurred

(2/2)

Bit	Name	Function
5	PIUINTR	PIU interrupt request 0 : Not occurred 1 : Occurred
4	Reserved	0 is returned when read
3	ETIMERINTR	ElapsedTime interrupt request 0 : Not occurred 1 : Occurred
2	RTCL1INTR	RTCLong1 interrupt request 0 : Not occurred 1 : Occurred
1	POWERINTR	Power switch interrupt request 0 : Not occurred 1 : Occurred
0	BATINTR	Battery low interrupt request 0 : Not occurred 1 : Occurred

This register indicates level-1 interrupt requests' status.

9.2.2 MSYSINT1REG (0x0A00 008C)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	MDOZEPIU INTR	Reserved	MSOFT INTR	Reserved	MSIUINTR	MGIUINTR
R/W	R	R	R/W	R	R/W	R	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	MKIUINTR	MAIUINTR	MPIUINTR	Reserved	METIMER INTR	MRTCL1 INTR	MPOWER INTR	MBATINTR
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15, 14	Reserved	0 is returned when read
13	MDOZEPIUINTR	Enables PIU interrupt during Suspend mode 0 : Disable 1 : Enable
12	Reserved	0 is returned when read
11	MSOFTINTR	Enables software interrupt 0 : Disable 1 : Enable
10	Reserved	0 is returned when read
9	MSIUINTR	Enables SIU interrupt 0 : Disable 1 : Enable
8	MGIUINTR	Enables GIU interrupt 0 : Disable 1 : Enable
7	MKIUINTR	Enables KIU interrupt 0 : Disable 1 : Enable
6	MAIUINTR	Enables AIU interrupt 0 : Disable 1 : Enable

(2/2)

Bit	Name	Function
5	MPIUINTR	Enables PIU interrupt 0 : Disable 1 : Enable
4	Reserved	0 is returned when read
3	METIMERINTR	Enables ElapsedTime interrupt 0 : Disable 1 : Enable
2	MRTCL1INTR	Enables RTCLong1 interrupt 0 : Disable 1 : Enable
1	MPOWERINTR	Enables Power switch interrupt 0 : Disable 1 : Enable
0	MBATINTR	Enables battery low interrupt 0 : Disable 1 : Enable

This register is used to enable/disable level-1 interrupts.

9.2.3 NMIREG (0x0A00 0098)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	NMIORINT
R/W	R	R	R	R	R	R	R	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 1	Reserved	0 is returned when read
0	NMIORINT	Battery low interrupt request routing 0 : NMI 1 : Int0

This register is used to set the interrupt request signal used to notify the VR4110 CPU core when a battery low interrupt request has occurred.

9.2.4 SOFTINTREG (0x0A00 009A)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	SOFTINTR
R/W	R	R	R	R	R	R	R	W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 1	Reserved	0 is returned when read
0	SOFTINTR	Set/clear a software interrupt request. This bit is a write-only bit. Software interrupt request pending status is reported in the SYSINT1REG (0x0A000080). 0 : Clear 1 : Set

This register is used to set a software interrupt request.

9.2.5 SYSINT2REG (0x0A00 0200)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	LCDINTR	DMAINTR	Reserved	CSUINTR	ECUINTR	LEDINTR	RTCL2INTR
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 7	Reserved	0 is returned when read
6	LCDINTR	LCD interrupt request 0 : Not occurred 1 : Occurred
5	DMAINTR	DMA interrupt request 0 : Not occurred 1 : Occurred
4	Reserved	0 is returned when read
3	CSUINTR	CSI interrupt request 0 : Not occurred 1 : Occurred
2	ECUINTR	CompactFlash interrupt request 0 : Not occurred 1 : Occurred
1	LEDINTR	LED interrupt request 0 : Not occurred 1 : Occurred
0	RTCL2INTR	RTCLong2 interrupt request 0 : Not occurred 1 : Occurred

This register indicates level-1 interrupt requests' status.

9.2.6 MSYSINT2REG (0x0A00 0206)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	MLCDINTR	MDMAINTR	Reserved	MCSUINTR	MECUINTR	MLEDINTR	MRTCL2 INTR
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
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 7	Reserved	0 is returned when read
6	MLCDINTR	Enables LCD interrupt 0 : Disable 1 : Enable
5	MDMAINTR	Enables DMA interrupt 0 : Disable 1 : Enable
4	Reserved	Write 0 when write. 0 is returned when read
3	MCSUINTR	Enables CSI interrupt 0 : Disable 1 : Enable
2	MECUINTR	Enables CompactFlash interrupt 0 : Disable 1 : Enable
1	MLEDINTR	Enables LED interrupt 0 : Disable 1 : Enable
0	MRTCL2INTR	Enables RTCLong2 interrupt 0 : Disable 1 : Enable

This register is used to enable/disable level-1 interrupts.

9.2.7 PIUINTREG (0x0B00 0082)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	PADCMD INTR	PADADP INTR	PADPAGE1 INTR	PADPAGE0 INTR	PADDLOST INTR	Reserved	PENCHG INTR
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 7	Reserved	0 is returned when read
6	PADCMDINTR	PIU command scan interrupt request. This interrupt request occurs when a valid data is detected during a command scan. 0 : Not occurred 1 : Occurred
★ 5	PADADPINTR	PIU AD Port Scan interrupt request. This interrupt request occurs when a valid data is obtained during an A/D port scan. 0 : Not occurred 1 : Occurred
4	PADPAGE1INTR	PIU data buffer page 1 interrupt request. This interrupt request occurs when a set of valid data is stored in the page 1 of the data buffer. 0 : Not occurred 1 : Occurred
3	PADPAGE0INTR	PIU data buffer page 0 interrupt request. This interrupt request occurs when a set of valid data is stored in the page 0 of the data buffer. 0 : Not occurred 1 : Occurred
★ 2	PADDLOSTINTR	Data loss interrupt request. This interrupt request occurs when a set of data cannot be obtained within the specified time. 0 : Not occurred 1 : Occurred
1	Reserved	0 is returned when read
0	PENCHGINTR	Touch panel contact status change interrupt request. 0 : Not occurred 1 : Occurred

This register indicates when various PIU-related interrupt requests (level 2) occur.

9.2.8 AIUINTREG (0x0B00 0084)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	INTMIDLE	INTMST
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	INTSIDLE	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 10	Reserved	0 is returned when read
9	INTMIDLE	Audio input (microphone) idle interrupt request (received data is lost). This interrupt request occurs if a valid data exists in the MIDATREG register when data is received from the A/D converter. 0 : Not occurred 1 : Occurred
8	INTMST	Audio input (microphone) receive completion interrupt request. This interrupt request occurs when a 10-bit converted data from the A/D converter is received. 0 : Not occurred 1 : Occurred
7 to 2	Reserved	0 is returned when read
1	INTSIDLE	Audio output (speaker) idle interrupt request (mute). This interrupt request occurs if there is no valid data in the SODATREG register when data is transferred to the D/A converter. 0 : Not occurred 1 : Occurred
0	Reserved	0 is returned when read

This register indicates when various AIU-related interrupt requests occur.

★ 9.2.9 KIUINTREG (0x0B00 0086)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	KDATLOST	KDATRDY	KDOWNINT
R/W	R	R	R	R	R	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 3	Reserved	0 is returned when read
★ 2	KDATLOST	Keyboard Data Lost interrupt request. This interrupt request occurs if the KIUDAT0 register is updated with the next key data prior to being read by the CPU core. 0 : Not occurred 1 : Occurred This bit is cleared by writing 1.
★ 1	KDATRDY	Keyboard Data Ready interrupt request. This interrupt request occurs when a set of scanning is completed and all the KIUDAT registers are updated. 0 : Not occurred 1 : Occurred This bit is cleared by writing 1.
0	KDOWNINT	Key Down interrupt request. This interrupt request occurs when the KIU sequencer is idle and any of the SCANIN inputs has been sampled as low level. 0 : Not occurred 1 : Occurred This bit is cleared by writing 1.

★ The KDATLOST bit is also cleared when the KIUDAT0 register is read.

9.2.10 MPIUINTREG (0x0B00 008E)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	PADCMD INTR	PADADP INTR	PADPAGE1 INTR	PADPAGE0 INTR	PADDLOST INTR	Reserved	PENCHG INTR
R/W	R	R/W	R/W	R/W	R/W	R/W	R	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 7	Reserved	0 is returned when read
6	PADCMDINTR	Enables PIU command scan interrupt 0 : Disable 1 : Enable
5	PADADPINTR	Enables PIU A/D Port Scan interrupt 0 : Disable 1 : Enable
4	PADPAGE1INTR	Enables PIU data buffer page 1 interrupt 0 : Disable 1 : Enable
3	PADPAGE0INTR	Enables PIU data buffer page 0 interrupt 0 : Disable 1 : Enable
★ 2	PADDLOSTINTR	Enables data loss interrupt 0 : Disable 1 : Enable
1	Reserved	0 is returned when read
0	PENCHGINTR	Enables touch panel contact status change interrupt 0 : Disable 1 : Enable

This register is used to mask various PIU-related interrupts.

9.2.11 MAIUINTREG (0x0B00 0090)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	INTMIDLE	INTMST
★ R/W	W	W	W	W	W	W	W	W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	INTSIDLE	Reserved
★ R/W	W	W	W	W	W	W	W	W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 10	Reserved	Write 0 when write
9	INTMIDLE	Enables audio input (microphone) idle interrupt (received data is lost) 0 : Disable 1 : Enable
8	INTMST	Enables audio input (microphone) receive completion interrupt 0 : Disable 1 : Enable
7 to 2	Reserved	Write 0 when write
1	INTSIDLE	Enables audio output (speaker) idle interrupt (mute) 0 : Disable 1 : Enable
0	Reserved	Write 0 when write

This register is used to mask various AIU-related interrupts.

★ This register is a write-only register and its contents when it is read are undefined.

9.2.12 MKIUINTREG (0x0B00 0092)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	MSKKDAT LOST	MSKKDAT RDY	MSKK DOWNINT
R/W	R	R	R	R	R	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 3	Reserved	0 is returned when read
2	MSKKDATLOST	Enables Keyboard Data Lost interrupt 0 : Disable 1 : Enable This bit may be used to temporarily mask the Keyboard Data Lost interrupt request and does not affect Keyboard Data Lost event detection.
1	MSKKDATRDY	Enables Keyboard Data Ready interrupt 0 : Disable 1 : Enable This bit may be used to temporarily mask the Keyboard Data Ready interrupt request and does not affect Keyboard Data Ready event detection.
0	MSKKDOWNINT	Enables Key Down interrupt 0 : Disable 1 : Enable This bit may be used to temporarily mask the Key Down interrupt request and does not affect Key Down event detection.

CHAPTER 10 POWER MANAGEMENT UNIT (PMU)

This chapter describes the Power Management Unit (PMU) operation, register settings and power modes.

10.1 General

The PMU performs power management within the VR4181 and controls the power supply throughout the system. The PMU provides the following functions:

- Reset control
- Shutdown control
- Power-on control
- Low-power mode control

10.2 VR4181 Power Mode

This section describes the VR4181 power modes in detail. The VR4181 supports the following four power modes:

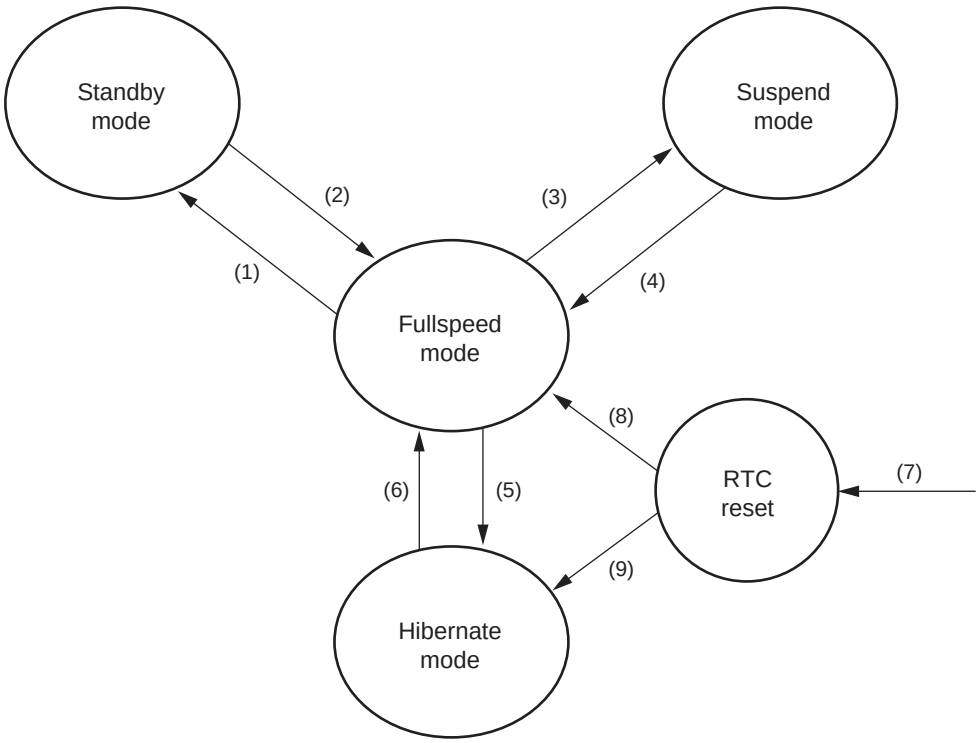
- Fullspeed mode
- Standby mode
- Suspend mode
- Hibernate mode

10.2.1 Power mode and state transition

The VR4181 transits from Fullspeed mode to Standby mode, Suspend mode, or Hibernate mode by executing a STANBY, SUSPEND, or HIBERNATE instruction respectively. An RTC reset is always valid in every mode, and initializes (resets) units in the VR4181 including the RTC.

The figure on the following page, Figure 10-1, is a conceptual diagram showing the interaction and control of the four power modes of the VR4181.

Figure 10-1. Transition of VR4181 Power Mode



Transition No.	Factors
★ (1)	STANDBY instruction
(2)	All interrupt requests
★ (3)	SUSPEND instruction <After transition> DRAM self refresh
★ (4)	Assertion of POWER Assertion and then deassertion of RSTSW# Interrupt request such as: ElapsedTime timer, Key press, DCD1# (SIU1) RTCLong1, Pen touch, CF_BUSY# RTCLong2, GPIO(15:0), BATTINTR
★ (5)	HIBERNATE instruction <After transition> DRAM self refresh Deassertion of MPOWER
(6)	Assertion of POWER Interrupt request such as: ElapsedTime timer, GPIO(15:0) DCD1#, CF_BUSY#
(7)	Assertion of RTCRST# <After transition> Deassertion of MPOWER
(8)	Assertion and then deassertion of RTCRST# BATTINH = high (normal activation)
(9)	Assertion and then deassertion of RTCRST# BATTINH = low (BATTINH shutdown)

Table 10-1 shows power mode overview and transaction:

Table 10-1. Overview of Power Modes

Mode	Internal peripheral unit					CPU core
	RTC	ICU	DMA	LCDC	Others	
Fullspeed	On	On	On	On	Selectable	On
Standby	On	On	On	On	Selectable	Off
Suspend	On	On	Off	Off	Off	Off
Hibernate	On	Off	Off	Off	Off	Off
Off	Off	Off	Off	Off	Off	Off

(1) Fullspeed mode

All internal clocks and bus clocks operate. The VR4181 can perform every function during the Fullspeed mode.

★ **(2) Standby mode**

The pipeline clock (PClock) of the CPU core is fixed to high level. PLL, timer/interrupt function of the CPU core, interrupt clock (MasterOut), internal bus clock (TClock and PCLK), and RTC clock continue their operation.

Therefore, all the on-chip peripheral units continue their operation (operation of the LCD controller and DMA also continue). The contents of caches and registers in the CPU core are retained.

To enter to Standby mode from Fullspeed mode, execute the STANDBY instruction. After the STANDBY instruction has passed the WB stage, the VR4181 waits until SysAD bus (internal) enters idle state. Then, internal clocks are shut down, and pipeline operation stops.

To restore to Fullspeed mode, generate an interrupt request of any kind. When the processor restores to Fullspeed mode from Standby mode, it starts a program execution from the General exception vector (0xBFC0 0380 when BEV = 0 or 0x8000 0180 when BEV = 1).

★ **(3) Suspend mode**

The pipeline clock (PClock) of the CPU core and the internal bus clocks (TClock and PCLK) are fixed to high level. PLL, timer/interrupt function of the CPU core, interrupt clock (MasterOut), and RTC clock continue their operation.

The contents of caches and registers in the CPU core are retained. The contents of connected DRAMs can be preserved by putting DRAMs into self-refresh mode.

To enter to Suspend mode from Fullspeed mode, execute a Suspend mode sequence (see **10.6 DRAM Interface Control**) first. After the SUSPEND instruction has passed the WB stage and DRAMs enter self-refresh mode, the VR4181 waits until SysAD bus (internal) enters idle state. Then, internal clocks are shut down, and pipeline operation stops.

To restore to Fullspeed mode from Suspend mode, one of the interrupt requests listed in Figure 10-1 (interrupt requests that can be used are limited since the internal bus clocks (TClock and PCLK) stop). When the processor restores to Fullspeed mode from Suspend mode, it starts a program execution from the General exception vector (0xBFC0 0380 when BEV = 0 or 0x8000 0180 when BEV = 1).

★ **(4) Hibernate mode**

All clocks other than the RTC clock (32.768 kHz) are fixed to high level and the PLL operation stops. An RTC and a monitor for activation factors in the PMU continue their operation.

To enter to Hibernate mode from Fullspeed mode, execute a Hibernate mode sequence (see **10.6 DRAM Interface Control**) first. After the HIBERNATE instruction has passed the WB stage and DRAMs enter self-refresh mode, the VR4181 waits until SysAD bus (internal) enters idle state. Then, MPOWER signal becomes inactive after internal clocks are shut down and pipeline operation stops.

2.5 V power supply can be stopped during MPOWER signal is inactive. If it is stopped, however, the contents of registers in the peripheral units other than PMU, GIU, LED, and RTC are not retained.

To restore to Fullspeed mode from Hibernate mode, one of the interrupt requests listed in Figure 10-1. When the processor restores to Fullspeed mode from Hibernate mode, it starts a program execution from the Cold Reset exception vector (0xBFC0 0000).

10.3 Reset Control

The operations of the RTC, peripheral units, and CPU core, and PMUINTREG register bit settings during a reset are listed below.

Table 10-2. Operations During Reset

Reset type	RTC, GIU	Peripheral units	CPU core	PMUINTREG bits
RTC reset	Reset	Reset	Cold Reset	RTCRST = 1
RSTSW reset 1	Active	Reset	Cold Reset	RSTSW = 1 SDRAM = 0
RSTSW reset 2	Active	Active	Cold Reset	RSTSW = 1 SDRAM = 1
★ Deadman's Switch reset	Active	Reset	Cold Reset	DMSRST = 1

Caution When bit 6 of the PMUINTREG register is set to 1, only the CPU core is reset during a RSTSW reset cycle, and all internal peripheral units retain their current state. Software must re-initialize or reset all peripheral units in this case.

To preserve SDRAM data during a RSTSW reset, bit 6 of the PMUINTREG register should be set to 1 when SDRAM is used.

10.3.1 RTC reset

When the RTCRST# signal becomes active, the PMU resets all internal peripheral units including the RTC unit. It also resets (Cold Reset) the CPU core.

In addition, the RTCRST bit in the PMUINTREG register is set to 1. After the CPU core is restarted, the RTCRST bit must be checked and cleared to 0 by software.

For details of the timing of RTC reset, refer to **CHAPTER 5 INITIALIZATION INTERFACE**.

10.3.2 RSTSW reset

When the RSTSW# signal becomes active, the PMU resets (Cold Reset) the CPU core. When bit 6 of the ★ PMUINTREG register is cleared to 0, the PMU also resets all internal peripheral units except for the RTC and GIU.

In addition, the RSTSW bit in the PMUINTREG register is set to 1. After the CPU core is restarted, the RSTSW bit must be checked and cleared to 0 by software.

For details of the timing of RSTSW reset, refer to **CHAPTER 5 INITIALIZATION INTERFACE**.

★ 10.3.3 Deadman's Switch reset

When the Deadman's Switch function is enabled, software must write 1 to DSWCLR bit in the DSUCLRREG register each set time, to clear the Deadman's Switch counter (for more information, refer to **CHAPTER 12 DEADMAN'S SWITCH UNIT (DSU)**).

If the Deadman's Switch counter is not cleared within the set time, the PMU resets all peripheral units except for RTC, GIU, and PMU. Then the PMU resets (Cold Reset) the CPU core.

In addition, DMSRST bit in the PMUINTREG register is set to 1. After the CPU core is restarted, DMSRST bit must be checked and cleared to 0 by software.

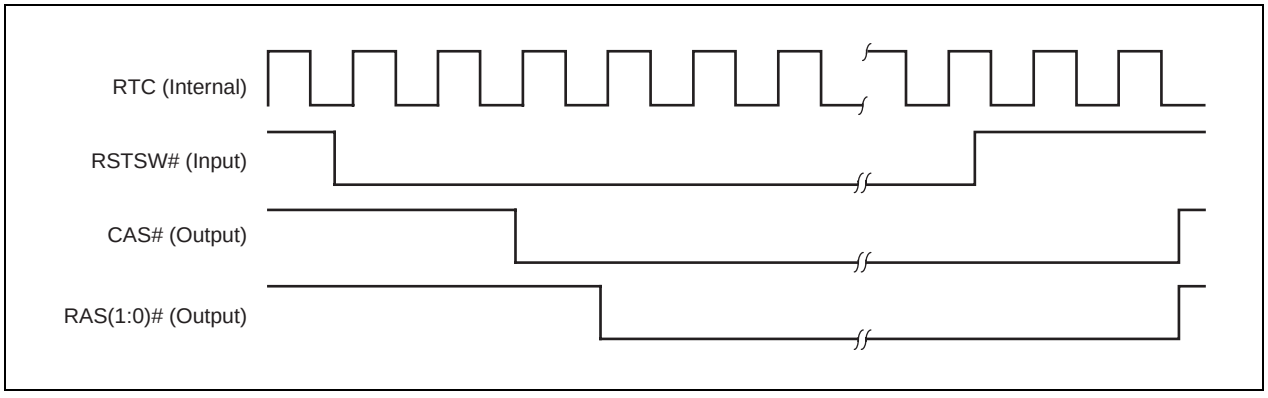
10.3.4 Preserving DRAM data on RSTSW reset

(1) Preserving EDO-DRAM data

When an RSTSW reset takes place, the PMU activates the CAS#/RAS# pins to generate a CBR self refresh request to EDO DRAM.

Remark There is no burst CBR refresh before and after CBR self refresh by RSTSW reset.

Figure 10-2. EDO DRAM Signals on RSTSW Reset (SDRAM Bit = 0)



★ (2) Preserving SDRAM data

The SDRAM bit of the PMUINTREG register can be used to preserve the contents of SDRAM connected to the VR4181 during an RSTSW reset. When the SDRAM bit is set to 1, the PMU does not reset the memory controller. Therefore, the memory controller completes current SDRAM access and performs CBR refresh cycle on an RSTSW reset. On the other hand, when the SDRAM bit is set to 0, the memory controller is reset regardless of accesses under processing and does not perform CBR refresh cycle (SDRAM data will be destroyed).

10.4 Shutdown Control

The operations of the RTC, peripheral units, and CPU core, and PMUINTREG register bit settings during a reset are listed below.

For detail of the timing of each shutdown, refer to **CHAPTER 5 INITIALIZATION INTERFACE**.

★ **Table 10-3. Operations During Shutdown**

Shutdown type	RTC, GIU	Peripheral units	CPU core	PMUINTREG bits
HALTimer shutdown	Active	Reset	Cold Reset	TIMOUTRST = 1
Software shutdown	Active	Reset	Cold Reset	–
BATTINH shutdown	Active	Reset	Cold Reset	BATTINH = 1

10.4.1 HALTimer shutdown

After the CPU core is activated (following the mode change from Shutdown or Hibernate mode to Fullspeed mode), or the CPU core is reset by RSTSW reset, software must write 1 to HALTIMERRST bit in the PMUCNTREG register within about four seconds to clear the HALTimer.

★ If the HALTimer is not reset within about four seconds after the CPU core is activated or the RSTSW reset is canceled, the PMU resets all peripheral units except for RTC and PMU. Then the PMU resets (Cold Reset) the CPU core.

In addition, TIMOUTRST bit in PMUINTREG register is set to 1. After the CPU core is restarted, TIMOUTRST bit must be checked and cleared to 0 by software.

10.4.2 Software shutdown

When the HIBERNATE instruction is executed, the PMU checks for currently pending interrupt requests. If there are no pending interrupt requests, it stops the CPU core clock. It then resets all peripheral units except for the RTC, ★ GIU, and the PMU.

The PMU register contents do not change.

10.4.3 BATTINH shutdown

If the BATTINH signal is asserted when the CPU core is going to be activated, the PMU stops CPU activation and ★ resets all peripheral units except for the RTC, GIU, and the PMU. Then it resets the CPU core.

In addition, BATTINH bit in the PMUINTREG register is set to 1. After the CPU core is restarted, BATTINH bit must be checked and cleared to 0 by software.

For details of the timing of BATTINH shutdown, see **10.5 Power-on Control** below.

10.5 Power-on Control

The causes of CPU core activation (mode change from shutdown mode or Hibernate mode to Fullspeed mode) are called activation factors. There are twenty activation factors: a power switch interrupt (POWER), sixteen types of GPIO activation interrupts (GPIO(15:0)), a DCD interrupt (DCD#), a CompactFlash interrupt, and an ElapsedTime interrupt.

Battery low detection (BATTINH/BATTINT# pin check) is a factor that prevents CPU core activation.

The period (power-on wait time) in which the POWERON pin is active at power-on can be specified by using PMUWAITREG register. After RTCRST, by which the CPU core is initialized, the period is set as 343.75 ms. Power-on wait time can be specified when activation is caused by sources other than RTCRST.

When MPOWER signal is at low level (Hibernate mode or during CPU core activation), to stop supplying voltage to the 2.5 V power-supply systems is recommended to reduce leak current. This means that this power supply can be 0 V while the MPOWER signal is inactive. The following operation will not be affected by supplying voltage of 2.3 V or more to this power supply within the period from when the MPOWER signal becomes active to when PLL starts oscillation.

Caution When the CPU core enters the Hibernate mode by executing the HIBERNATE instruction, if an activation factor occurs simultaneously, the CPU core may be activated without asserting the POWERON signal after the MPOWER signal is once de-asserted. Moreover, if RSTSW#, which is not an activation factor of the Hibernate mode, is asserted at the same time a transition to the Hibernate mode by executing the HIBERNATE instruction occurs, the CPU core may be activated without asserting the POWERON signal after the MPOWER signal is de-asserted once. ★

10.5.1 Activation via Power Switch interrupt request

When the POWER signal is asserted, the PMU asserts the POWERON signal to provide an external notification that the CPU core is being activated. After asserting the POWERON signal, the PMU checks the BATTINH signal and then de-asserts the POWERON signal.

If the BATTINH signal is at high level, the PMU cancels peripheral unit reset and starts the Cold Reset sequence to activate the CPU core.

If the BATTINH signal is at low level, the PMU sets 1 to the BATTINH bit in the PMUINTREG register and then performs another shutdown. After the CPU core is restarted, the BATTINH bit must be checked and cleared to 0 by software.

Remark Activation via Power Switch interrupt request never sets the POWERSWINTR bit in the PMUINTREG register to 1.

Figure 10-3. Activation via Power Switch Interrupt Request (BATTINH = H)

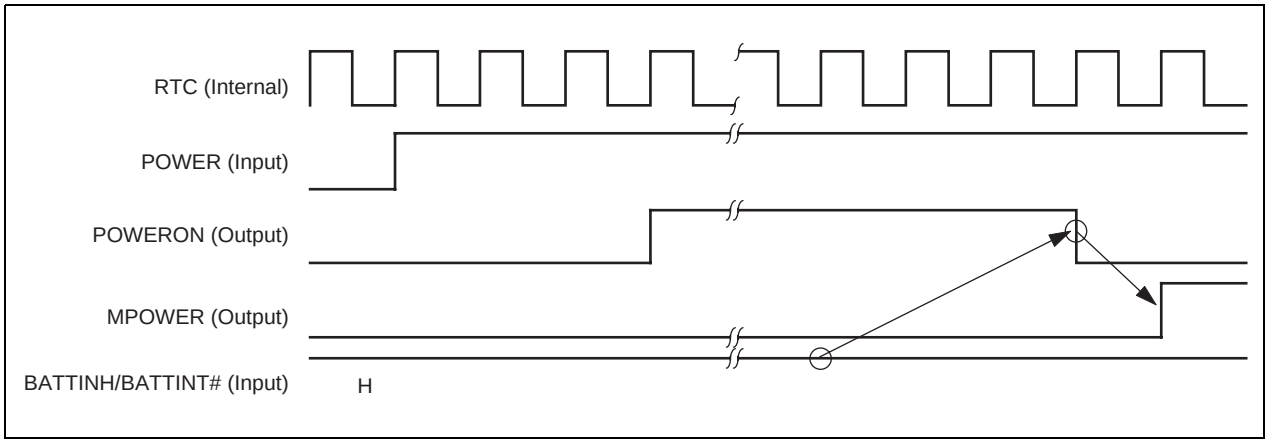
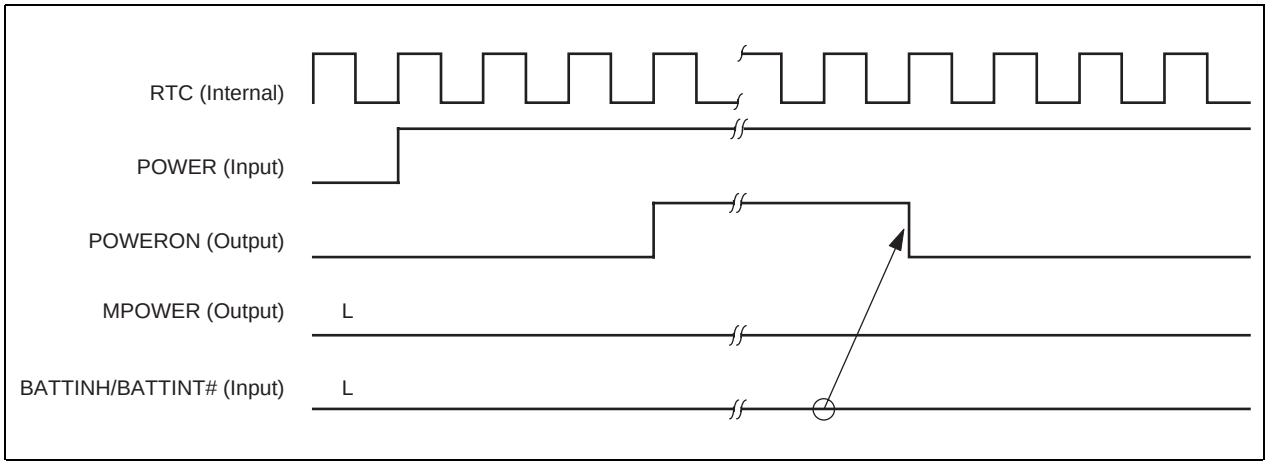


Figure 10-4. Activation via Power Switch Interrupt Request (BATTINH = L)



10.5.2 Activation via CompactFlash interrupt request

★ When the CF_BUSY# signal is asserted, the PMU asserts the POWERON signal to provide an external notification that the CPU core is being activated. After asserting the POWERON signal, the PMU checks the BATTINH signal and then de-asserts the POWERON signal.

If the BATTINH signal is at high level, the PMU cancels peripheral unit reset and starts the Cold Reset sequence to activate the CPU core.

If the BATTINH signal is at low level, the PMU sets 1 to the BATTINH bit in the PMUINTREG register and then performs another shutdown. After the CPU core is restarted, the BATTINH bit must be checked and cleared to 0 by software.

Figure 10-5. Activation via CompactFlash Interrupt Request (BATTINH = H)

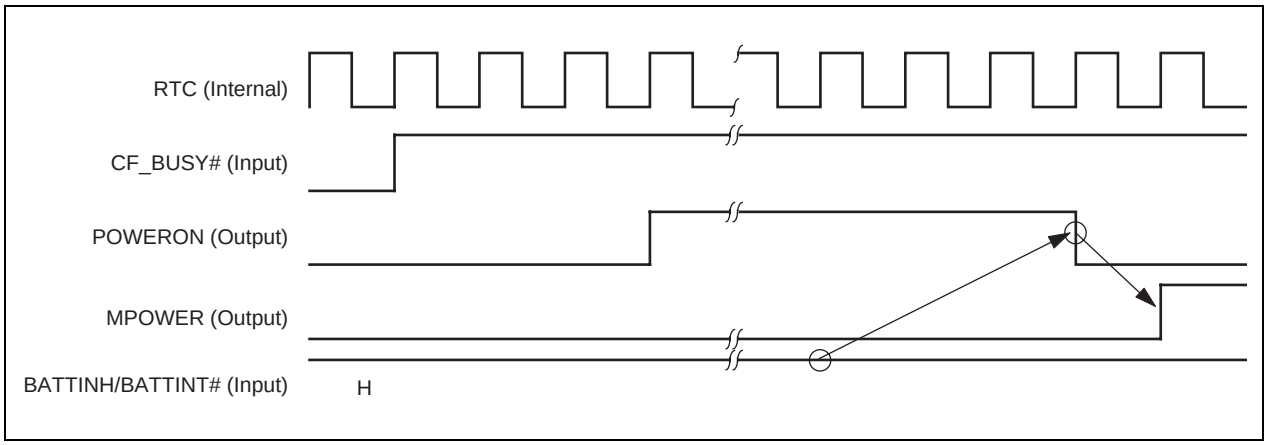
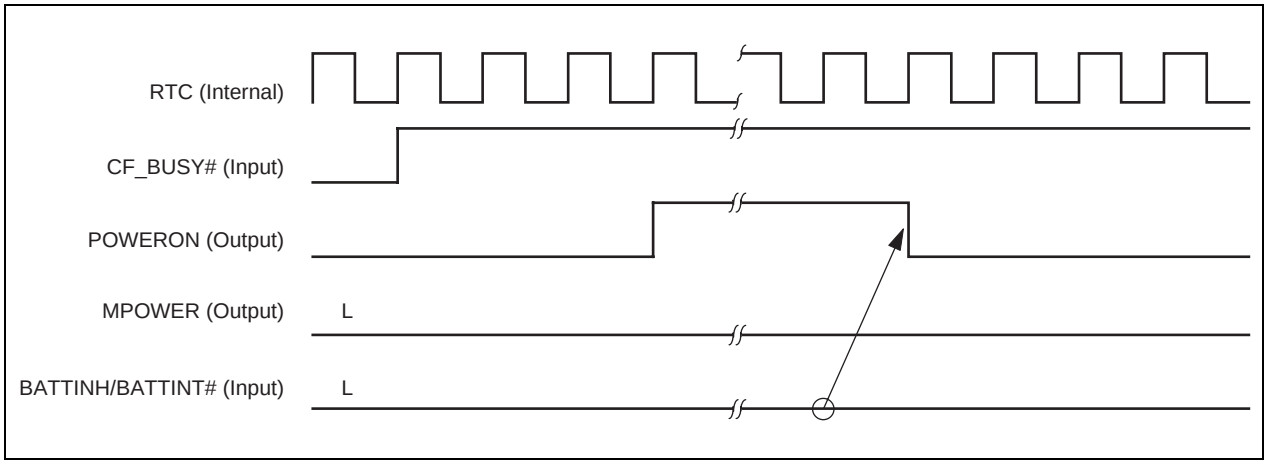


Figure 10-6. Activation via CompactFlash Interrupt Request (BATTINH = L)



10.5.3 Activation via GPIO activation interrupt request

When any of the GPIO(15:0) signals are asserted, the PMU checks the GPIO(15:0) activation interrupt enable bits in the GIU. If GPIO(15:0) activation interrupts are enabled, the PMU asserts the POWERON signal to provide an external notification that the CPU core is being activated (since the GPIO(15:0) activation enable interrupt bits are cleared after an RTC reset, the GPIO(15:0) signal cannot be used for activation immediately after an RTC reset).

After asserting the POWERON signal, the PMU checks the BATTINH signal and de-asserts the POWERON signal.

If the BATTINH signal is at high level, the PMU cancels the peripheral unit reset and starts the Cold Reset sequence to activate the CPU core.

If the BATTINH signal is at low level, the PMU sets 1 to the BATTINH bit in the PMUINTREG register and then performs another shutdown. After the CPU core is restarted, the BATTINH bit must be checked and cleared to 0 by software.

★ The CPU core sets 1 to the GPWAKEUP bit in the PMUINTREG register regardless of whether activation succeeds or fails.

Caution The changes in the GPIO signals are ignored while POWERON signal is active.

Figure 10-7. Activation via GPIO Activation Interrupt Request (BATTINH = H)

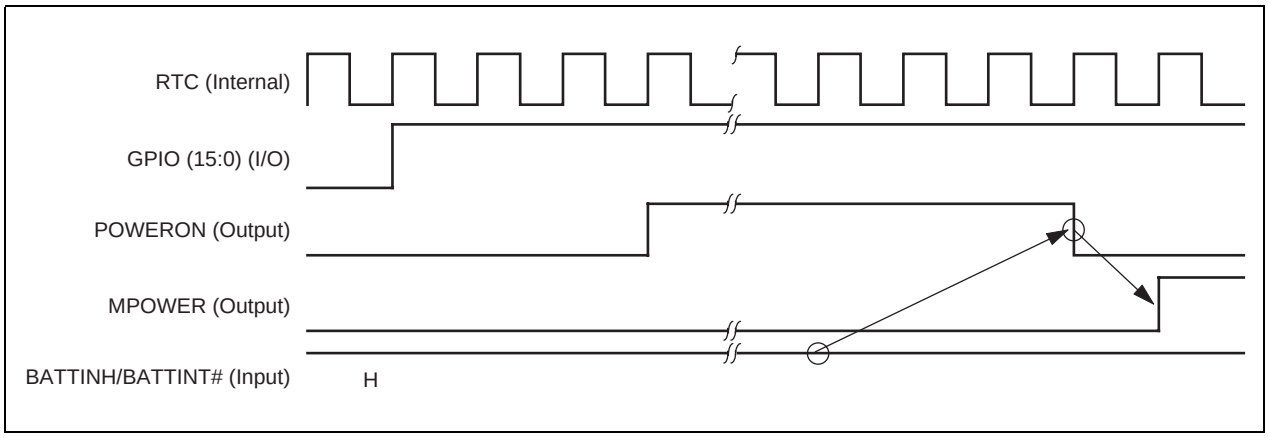
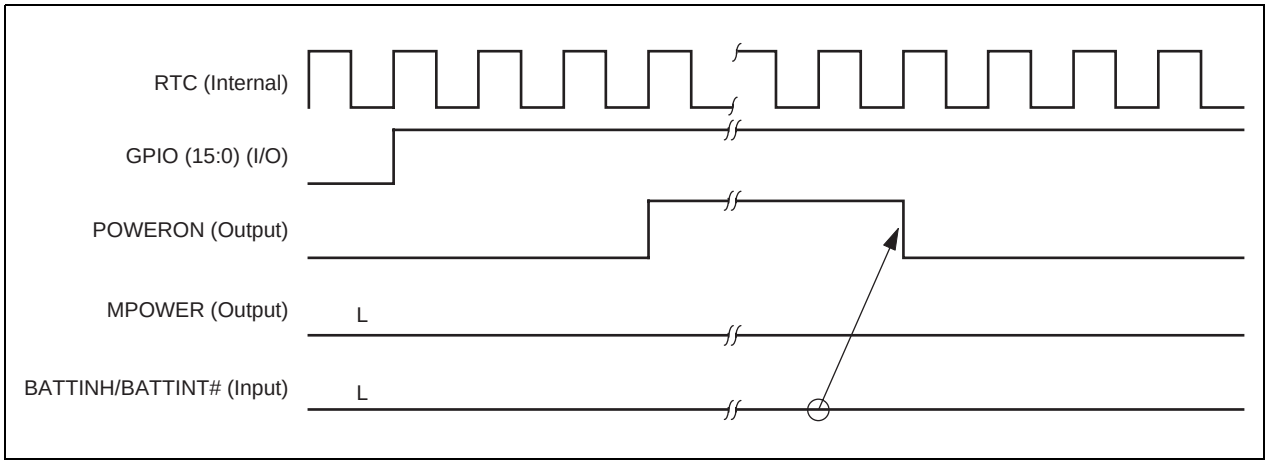


Figure 10-8. Activation via GPIO Activation Interrupt Request (BATTINH = L)



10.5.4 Activation via DCD interrupt request

When the DCD1# signal is asserted, the PMU asserts the POWERON signal to provide an external notification that the CPU core is being activated. After asserting the POWERON signal, the PMU checks the BATTINH signal and then de-asserts the POWERON signal.

If the BATTINH signal is at high level, the PMU cancels the peripheral unit reset and starts the Cold Reset sequence to activate the CPU core.

If the BATTINH signal is at low level, the PMU sets 1 to the BATTINH bit in the PMUINTREG register and then performs another shutdown. After the CPU core is restarted, the BATTINH bit must be checked and cleared to 0 by software.

The DCDST bit in the PMUINTREG register does not indicate whether a DCD interrupt has occurred but instead reflects the current status of the DCD1# pin.

★ **Cautions1. The PMU cannot recognize changes in the DCD1# signal while the POWER signal is asserted. If the DCD1# state when the POWER signal is asserted is different from that when the POWER signal is deasserted, the change in the DCD1# signal is detected only after the POWER signal is deasserted. However, if the DCD1# state when the POWER signal is asserted is the same as that when the POWER signal is deasserted, any changes in the DCD1# signal that occur while the POWER signal is asserted are not detected.**

2. The changes in the DCD1# signal are ignored while the POWERON signal is active.

3. There is no indicator which shows an activation via DCD interrupt, if DCD1# signal has already changed from active to inactive during power-on sequence. In other words, if software can not find activation factor and if the DCDST bit indicates that DCD1# signal is active, the above situation occurred.

Figure 10-9. Activation via DCD Interrupt Request (BATTINH = H)

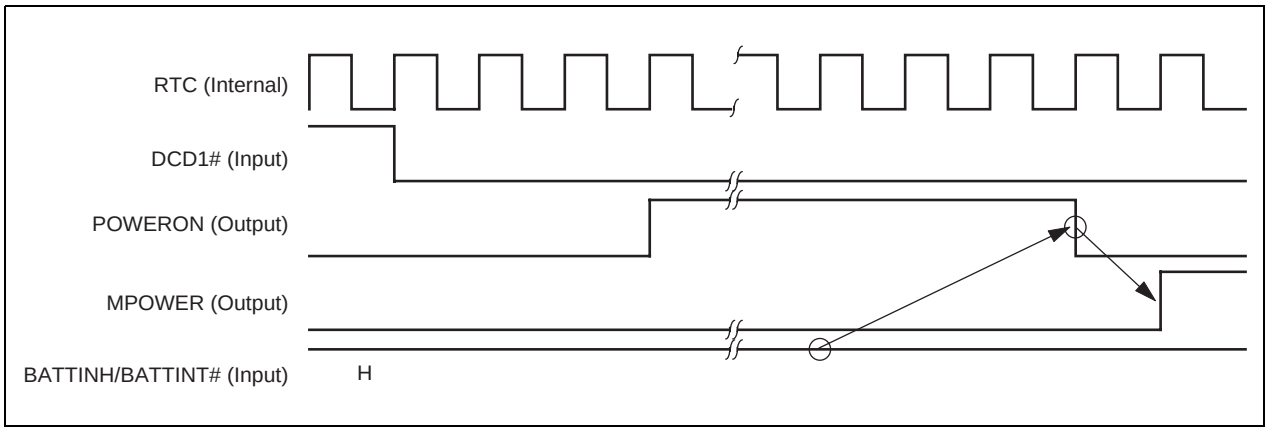
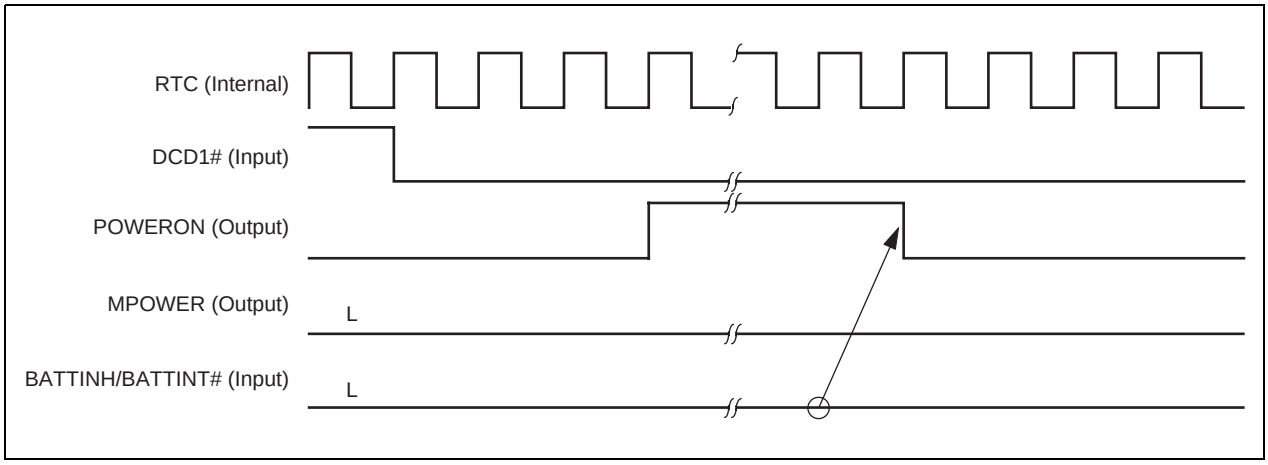


Figure 10-10. Activation via DCD Interrupt Request (BATTINH = L)



10.5.5 Activation via ElapsedTime (RTC alarm) interrupt request

When the alarm (alarm_intr signal) generated from the ElapsedTime timer is asserted, the PMU asserts the POWERON signal to provide an external notification that the CPU core is being activated. After asserting the POWERON signal, the PMU checks the BATTINH signal and then de-asserts the POWERON signal.

If the BATTINH signal is at high level, the PMU cancels the peripheral unit reset and starts the Cold Reset sequence to activate the CPU core.

If the BATTINH signal is at low level, the PMU sets 1 to the BATTINH bit in the PMUINTREG register and then performs another shutdown. After the CPU core is restarted, the BATTINH bit must be checked and cleared to 0 by software.

Caution The ElapsedTime interrupt is ignored while the POWERON signal is active. After the POWERON signal becomes inactive, the PMU is notified.

Figure 10-11. Activation via ElapsedTime Interrupt Request (BATTINH = H)

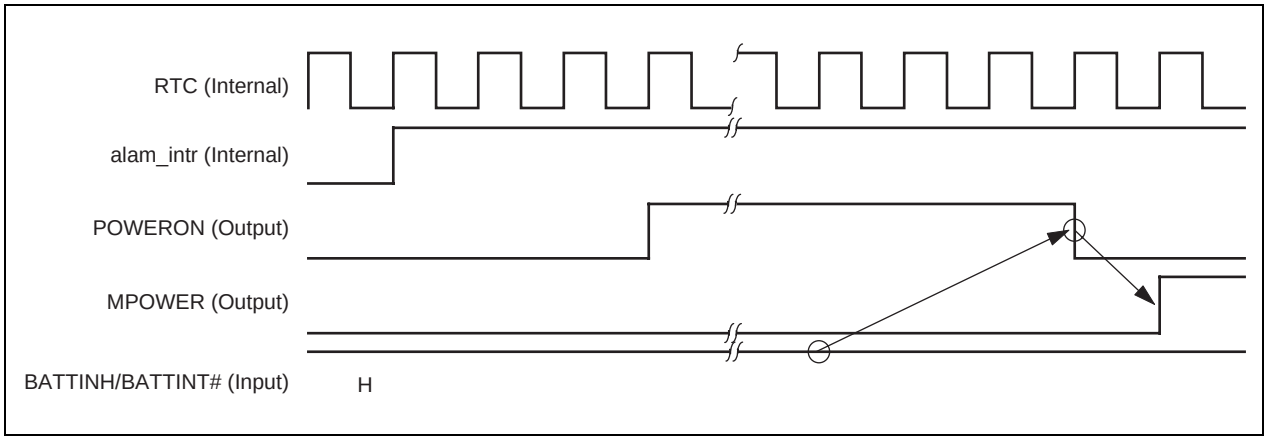
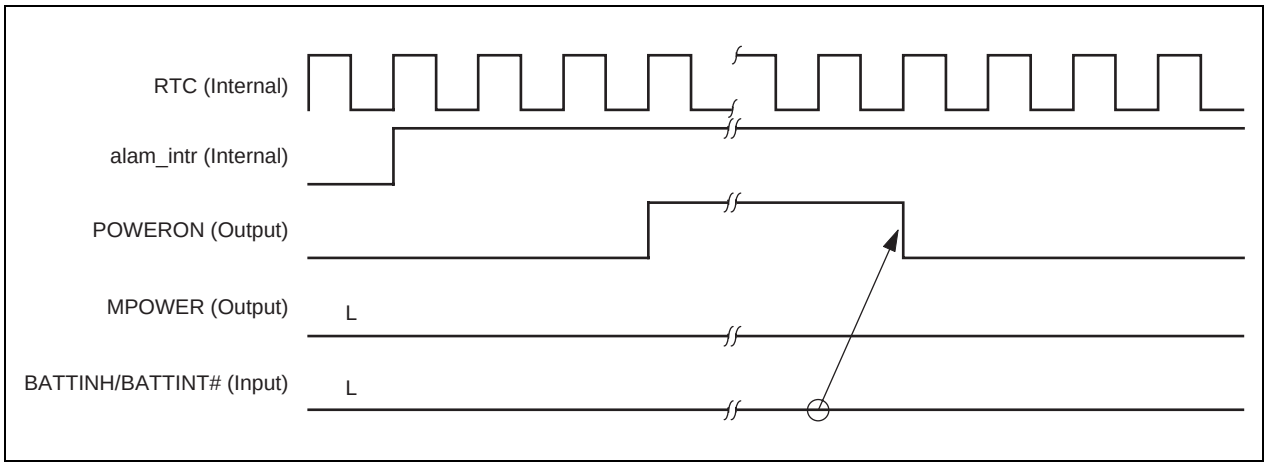


Figure 10-12. Activation via ElapsedTime Interrupt Request (BATTINH = L)



10.6 DRAM Interface Control

The PMU provides a register to control the DRAM interface during Hibernate mode or Suspend mode. The DRAMHIBCTL register permits software to directly control the state of the DRAM interface pins prior to executing a HIBERNATE or SUSPEND instruction.

The DRAMHIBCTL register also provides status indication of the memory controller.

The software flow when entering and exiting Hibernate mode or Suspend mode is shown below.

★ 10.6.1 Entering Hibernate mode (EDO DRAM)

<1> Copy contents of all 2.5 V registers (i.e. DRAM type and configuration, ROM type and configuration, etc.) that must be preserved during Hibernate mode into the general-purpose registers, MISCREG(0:15), in the GIU or into external memory.

Remark 3.3 V peripheral units: PMU, GIU, LED, and RTC
2.5 V peripheral units: all peripherals except PMU, GIU, LED, and RTC

<2> Stop operations of the DMA controller and LCD controller.

<3> Copy the codes for the Hibernate mode (<4> through <11> below) beginning at a 16-byte boundary into the cache by using a Fill operation of CACHE instruction, and jump to the cached codes.

<4> Stop all peripheral clocks by writing zero to the CMUCLKMSK register in the MBA Host Bridge.

<5> If DRAM can accept mixed use of burst and distributive CBR refresh, set a value that determines the refresh count to every 250 ns to the BCURFCNTREG register in the MBA Host Bridge. Then execute CBR refresh cycles for a specific time period (i.e. 0x3FFF × TClock period + burst refresh interval required by DRAM).

<6> Set 0x3FF to the BCURFCNTREG register in the MBA Host Bridge that determines refresh interval to maximum to prevent an interruption of a Hibernate mode sequence.

<7> Set the SUSPEND bit in the DRAMHIBCTL register to 1. If the BstRefr bit of the MEMCFG_REG register in the memory controller to 1, the memory controller performs a burst refresh cycle and then put the DRAM into self-refresh mode.

<8> Poll the OK_STOP_CLK bit in the DRAMHIBCTL register to confirm that the memory controller completes a burst refresh cycle and put the DRAM into self-refresh mode.

<9> Set the STOP_CLK bit in the DRAMHIBCTL register to 1 to stop supplying TClock to the memory controller.

<10> Set the DRAM_EN bit in the DRAMHIBCTL register to 1 so that the DRAM interface signals are latched.

<11> Execute a HIBERNATE instruction.

<12> Stop applying 2.5 V power supply when the MPOWER signal becomes low level.

Caution When entering Hibernate mode, set the BEV bit of the Status register in the CP0 of the CPU core to 1 to make sure that the vector of the exception handler points the ROM area.

★ 10.6.2 Entering Hibernate mode (SDRAM)

<1> Copy contents of all 2.5 V registers (i.e. DRAM type and configuration, ROM type and configuration, etc.) that must be preserved during Hibernate mode into the general-purpose registers, MISCREG(0:15), in the GIU or into external memory.

Remark 3.3 V peripheral units: PMU, GIU, LED, and RTC
2.5 V peripheral units: all peripherals except PMU, GIU, LED, and RTC

<2> Stop operations of the DMA controller and LCD controller.
<3> Copy the codes for the Hibernate mode (<4> through <12> below) beginning at a 16-byte boundary into the cache by using a Fill operation of CACHE instruction, and jump to the cached codes.
<4> Stop all peripheral clocks by writing zero to the CMUCLKMSK register in the MBA Host Bridge.
<5> Set the BCURFCNTREG register in the MBA Host Bridge to a value that determines refresh interval to maximum to prevent an interruption of a Hibernate mode sequence.
<6> If burst refreshes are needed, set a value that determines the refresh count to every 250 ns to the BCURFCNTREG register in the MBA Host Bridge. Then execute CBR auto refresh cycles for a specific time period (i.e. 0x3FFF × TClock period + burst refresh interval required by DRAM).
<7> Clear the BstRefr bit of the MEMCFG_REG register in the memory controller to 0 to disable a burst refresh. Then set SUSPEND bit in the DRAMHIBCTL register to 1 to put the DRAM into self-refresh mode.
<8> Poll the OK_STOP_CLK bit in the DRAMHIBCTL register to confirm that the memory controller puts the DRAM into self-refresh mode.
<9> Set the STOP_CLK bit in the DRAMHIBCTL register to 1 to stop supplying TClock to the memory controller.
<10> Set the DRAM_EN bit in the DRAMHIBCTL register to 1 so that the DRAM interface signals are latched.
<11> Clear the SUSPEND bit in the DRAMHIBCTL register to 0 after waiting for about 2 μs.
<12> Execute a HIBERNATE instruction.
<13> Stop applying 2.5 V power supply when the MPOWER signal becomes low level.

Caution When entering Hibernate mode, set the BEV bit of the Status register in the CP0 of the CPU core to 1 to make sure that the vector of the exception handler points the ROM area.

★ 10.6.3 Exiting Hibernate mode (EDO DRAM)

<1> Generate a wake-up event such as a transition on the POWER pin, a DCD interrupt, etc. which causes the PMU to start a power-on sequence.

<2> Apply 2.5 V power supply when the MPOWER signal becomes high level. The PMU waits until 3.3 V and 2.5 V power supply are stable, and then deasserts the reset signals to the VR4110 CPU core and on-chip peripheral units.

<3> Software execution resumes at the Cold Reset exception vector (0x0BFC 0000). Initialize the cache tags, and the Config, Status, and WatchLo registers in the CP0. Reset the HALTimer by setting the HALTIMERRST bit in the PMUCNTREG register to 1.

<4> Check and clear the TIMOUTRST bit in the PMUINTREG register in the case a HALTimer Shutdown had occurred.

<5> Copy the codes for the restore (<6> through <12> below) beginning at a 16-byte boundary into the cache by using a Fill operation of CACHE instruction, and jump to the cached codes. These codes can be executed on ROM.

<6> Poll the OK_STOP_CLK bit in the DRAMHIBCTL register until it is set to 1.

<7> Reinitialize all the registers and peripherals during Hibernate mode and restore those registers saved in the general-purpose registers, MISCREG(0:15) which retain values during Hibernate mode, in the GIU or in external memory.

Remark Software must wait until the OK_STOP_CLK bit in the DRAMHIBCTL register is set to 1 before reinitializing the memory controller registers. Otherwise unpredictable behavior of the memory controller could result.

<8> Clear the DRAM_EN bit in the DRAMHIBCTL register to 0 so that the DRAM interface signals are again driven directly by the memory controller.

<9> Clear SUSPEND bit in the DRAMHIBCTL register to 0 to exit self-refresh mode.

<10> Set the EDOMCYTREG and MEMCFG_REG registers in the memory controller according to the DRAM type to be used.

<11> If DRAM can accept mixed use of burst and distributive CBR refresh, set a value that determines the refresh count to every 250 ns to the BCURFCNTREG register in the MBA Host Bridge. Then execute CBR refresh cycles for a specific time period (i.e. 0x3FFF × TClock period + burst refresh interval required by DRAM).

<12> Restore to the BCURFCNTREG register in the MBA Host Bridge a value that determines refresh interval satisfying the conditions of DRAM type to be used.

★ 10.6.4 Exiting Hibernate mode (SDRAM)

<1> Generate a wake-up event such as a transition on the POWER pin, a DCD interrupt, etc. which causes the PMU to start a power-on sequence.

<2> Apply 2.5 V power supply when the MPOWER signal becomes high level. The PMU waits until 3.3 V and 2.5 V power supply are stable, and then deasserts the reset signals to the VR4110 CPU core and on-chip peripheral units.

<3> Software execution resumes at the Cold Reset exception vector (0x0BFC 0000). Initialize the cache tags, and the Config, Status, and WatchLo registers in the CP0. Reset the HALTimer by setting the HALTIMERRST bit in the PMUCNTREG register to 1.

<4> Check and clear the TIMOUTRST bit in the PMUINTREG register in the case a HALTimer Shutdown had occurred.

<5> Copy the codes for the restore (<6> through <12> below) beginning at a 16-byte boundary into the cache by using a Fill operation of CACHE instruction, and jump to the cached codes. These codes can be executed on ROM.

<6> Reinitialize all the registers and peripherals during Hibernate mode and restore those registers saved in the general-purpose registers, MISCREG(0:15) which retain values during Hibernate mode, in the GIU or in external memory.

<7> Clear the DRAM_EN bit in the DRAMHIBCTL register to 0 so that the DRAM interface signals are again driven directly by the memory controller.

<8> SDRAM exits the self-refresh mode.

<9> Set the MEMCFG_REG, MODE_REG, and SDTIMINGREG registers in the memory controller according to the SDRAM type to be used.

<11> If burst refreshes are needed, set a value that determines the refresh count to every 250 ns to the BCURFCNTREG register in the MBA Host Bridge. Then execute CBR auto refresh cycles for a specific time period (i.e. 0x3FFF × TClock period + burst refresh interval required by DRAM).

<12> Restore to the BCURFCNTREG register in the MBA Host Bridge a value that determines refresh interval satisfying the conditions of DRAM type to be used.

★ 10.6.5 Entering Suspend mode (EDO DRAM)

<1> Stop operations of the DMA controller and LCD controller.

<2> Set registers in the ICU and CP0 to allow notification of the interrupt requests used as wake-up events to Fullspeed mode to the CPU core.

<3> Copy the codes for the Suspend mode (<4> through <11> below) beginning at a 16-byte boundary into the cache by using a Fill operation of CACHE instruction, and jump to the cached codes.

<4> Stop all peripheral clocks by writing zero to the CMUCLKMSK register in the MBA Host Bridge.

<5> If DRAM can accept mixed use of burst and distributive CBR refresh, set a value that determines the refresh count to every 250 ns to the BCURFCNTREG register in the MBA Host Bridge. Then execute CBR refresh cycles for a specific time period (i.e. 0x3FFF × TClock period + burst refresh interval required by DRAM).

<6> Set 0x3FF to the BCURFCNTREG register in the MBA Host Bridge that determines refresh interval to maximum to prevent an interruption of a Suspend mode sequence.

<7> Set the SUSPEND bit in the DRAMHIBCTL register to 1. If the BstRefr bit of the MEMCFG_REG register in the memory controller to 1, the memory controller performs a burst refresh cycle and then put the DRAM into self-refresh mode.

<8> Poll the OK_STOP_CLK bit in the DRAMHIBCTL register to confirm that the memory controller completes a burst refresh cycle and put the DRAM into self-refresh mode.

<9> Set the STOP_CLK bit in the DRAMHIBCTL register to 1 to stop supplying TClock to the memory controller.

<10> Set the DRAM_EN bit in the DRAMHIBCTL register to 1 so that the DRAM interface signals are latched.

<11> Execute a SUSPEND instruction.

Caution When entering Suspend mode, set the BEV bit of the Status register in the CP0 of the CPU core to 1 to make sure that the vector of the exception handler points the ROM area.

★ 10.6.6 Entering Suspend mode (SDRAM)

<1> Stop operations of the DMA controller and LCD controller.

<2> Set registers in the ICU and CP0 to allow notification of the interrupt requests used as wake-up events to Fullspeed mode to the CPU core.

<3> Copy the codes for the Suspend mode (<4> through <12> below) beginning at a 16-byte boundary into the cache by using a Fill operation of CACHE instruction, and jump to the cached codes.

<4> Stop all peripheral clocks by writing zero to the CMUCLKMSK register in the MBA Host Bridge.

<5> Set the BCURFCNTREG register in the MBA Host Bridge to a value that determines refresh interval to maximum to prevent an interruption of a Suspend mode sequence.

<6> If burst refreshes are needed, set a value that determines the refresh count to every 250 ns to the BCURFCNTREG register in the MBA Host Bridge. Then execute CBR auto refresh cycles for a specific time period (i.e. 0x3FFF × TClock period + burst refresh interval required by DRAM).

<7> Clear the BstRefr bit of the MEMCFG_REG register in the memory controller to 0 to disable a burst refresh. Then set SUSPEND bit in the DRAMHIBCTL register to 1 to put the DRAM into self-refresh mode.

<8> Poll the OK_STOP_CLK bit in the DRAMHIBCTL register to confirm that the memory controller puts the DRAM into self-refresh mode.

<9> Set the STOP_CLK bit in the DRAMHIBCTL register to 1 to stop supplying TClock to the memory controller.

<10> Set the DRAM_EN bit in the DRAMHIBCTL register to 1 so that the DRAM interface signals are latched.

<11> Clear the SUSPEND bit in the DRAMHIBCTL register to 0 after waiting for about 2 μs.

<12> Execute a SUSPEND instruction.

Caution When entering Suspend mode, set the BEV bit of the Status register in the CP0 of the CPU core to 1 to make sure that the vector of the exception handler points the ROM area.

★ 10.6.7 Exiting Suspend mode (EDO DRAM)

<1> Generate a wake-up event from Suspend mode such as a transition on the POWER pin, a DCD interrupt, etc.
<2> Software execution resumes at the General exception vector (0x0BFC 0380 when BEV = 1).
<3> Copy the codes for the restore (<4> through <8> below) beginning at a 16-byte boundary into the cache by using a Fill operation of CACHE instruction, and jump to the cached codes. These codes can be executed on ROM.
<4> Poll the OK_STOP_CLK bit in the DRAMHIBCTL register until it is set to 1.

Remark Software must wait until the OK_STOP_CLK bit in the DRAMHIBCTL register is set to 1 before reinitializing the memory controller registers. Otherwise unpredictable behavior of the memory controller could result.

<5> Clear the DRAM_EN bit in the DRAMHIBCTL register to 0 so that the DRAM interface signals are again driven directly by the memory controller.
<6> Clear SUSPEND bit in the DRAMHIBCTL register to 0 to exit self-refresh mode.
<7> If DRAM can accept mixed use of burst and distributive CBR refresh, set a value that determines the refresh count to every 250 ns to the BCURFCNTREG register in the MBA Host Bridge. Then execute CBR refresh cycles for a specific time period (i.e. 0x3FFF × TClock period + burst refresh interval required by DRAM).
<8> Restore to the BCURFCNTREG register in the MBA Host Bridge a value that determines refresh interval satisfying the conditions of DRAM type to be used.

★ 10.6.8 Exiting Suspend mode (SDRAM)

<1> Generate a wake-up event from Suspend mode such as a transition on the POWER pin, a DCD interrupt, etc.
<2> Software execution resumes at the General exception vector (0x0BFC 0380 when BEV = 1).
<3> Copy the codes for the restore (<4> through <7> below) beginning at a 16-byte boundary into the cache by using a Fill operation of CACHE instruction, and jump to the cached codes. These codes can be executed on ROM.
<4> Clear the DRAM_EN bit in the DRAMHIBCTL register to 0 so that the DRAM interface signals are again driven directly by the memory controller.
<5> SDRAM exits the self-refresh mode.
<6> If burst refreshes are needed, set a value that determines the refresh count to every 250 ns to the BCURFCNTREG register in the MBA Host Bridge. Then execute CBR auto refresh cycles for a specific time period (i.e. 0x3FFF × TClock period + burst refresh interval required by DRAM).
<7> Restore to the BCURFCNTREG register in the MBA Host Bridge a value that determines refresh interval satisfying the conditions of DRAM type to be used.

10.7 Register Set

The PMU registers are listed below:

Table 10-4. PMU Registers

Physical address	R/W	Register symbol	Function
0x0B00 00A0	R/W	PMUINTREG	PMU interrupt status register
0x0B00 00A2	R/W	PMUCNTREG	PMU control register
0x0B00 00A8	R/W	PMUWAITREG	PMU wait counter register
0x0B00 00AC	R/W	PMUDIVREG	PMU Div mode register
0x0B00 00B2	R/W	DRAMHIBCTL	DRAM Hibernate mode control register

10.7.1 PMUINTREG (0x0B00 00A0)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	GP WAKEUP	CF_INT	DCDST	RTCINTR	BATTINH
R/W	R	R	R	R/W	R/W	R	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	SDRAM	TIMOUT RST	RTCRST	RSTSW	DMSRST	BATTINTR	POWER SWINTR
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
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 13	Reserved	0 is returned when read
12	GPWAKEUP	GPIO interrupt request detection. Cleared to 0 when 1 is written. 1 : Detected 0 : Not detected This bit must be checked and cleared to 0 after the CPU core is restarted.
11	CF_INT	CompactFlash interrupt request detection. Cleared to 0 when 1 is written. 1 : Detected 0 : Not detected This bit must be checked and cleared to 0 after the CPU core is restarted.
10	DCDST	DCD1# pin state 1 : High level (inactive) 0 : Low level (active)
9	RTCINTR	ElapsedTime (RTC alarm) interrupt request detection. Cleared to 0 when 1 is written. 1 : Detected 0 : Not detected This bit must be checked and cleared to 0 after the CPU core is restarted.
8	BATTINH	Battery low detection during activation. Cleared to 0 when 1 is written. 1 : Detected 0 : Not detected This bit must be checked and cleared to 0 after the CPU core is restarted.
7	Reserved	Write 0 when write. 0 is returned when read.
6	SDRAM	This bit determines whether the internal peripheral units are reset by RSTSW. This bit must be clear to 0 when EDO DRAM is used. 1 : Not reset (SDRAM data preserved during RSTSW) 0 : Reset (SDRAM data lost during RSTSW)

★

(2/2)

Bit	Name	Function
5	TIMOUTRST	HALTimer reset request detection. Cleared to 0 when 1 is written. 1 : Detected 0 : Not detected This bit must be checked and cleared to 0 after the CPU core is restarted.
4	RTCRST	RTC reset detection. Cleared to 0 when 1 is written. 1 : Detected 0 : Not detected This bit must be checked and cleared to 0 after the CPU core is restarted.
3	RSTSW	RSTSW interrupt request detection. Cleared to 0 when 1 is written. 1 : Detected 0 : Not detected This bit must be checked and cleared to 0 after the CPU core is restarted.
2	DMSRST	Deadman's Switch interrupt request detection. Cleared to 0 when 1 is written. 1 : Detected 0 : Not detected This bit must be checked and cleared to 0 after the CPU core is restarted.
1	BATTINTR	Battery low detection during normal operation. Cleared to 0 when 1 is written. 1 : Detected 0 : Not detected This bit must be checked and cleared to 0 after the CPU core is restarted.
0	POWERSWINTR	Power Switch interrupt request detection. Cleared to 0 when 1 is written. 1 : Detected 0 : Not detected This bit must be checked and cleared to 0 after the CPU core is restarted.

This register indicates the statuses of power-on factors and interrupt requests. It also indicates the status of the DCD1# pin.

The BATTINTR bit is set to 1 when the BATTINH/BATTINT# signal becomes low and a battery-low interrupt request occurs during modes other than the Hibernate mode (MPOWER = H).

The POWERSWINTR bit is set to 1 when the POWER signal becomes high and a Power Switch interrupt request occurs during modes other than the Hibernate mode. However, this bit is not set to 1 when the POWER signal becomes high during the Hibernate mode (MPOWER = L).

10.7.2 PMUCNTREG (0x0B00 00A2)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	STANDBY	Reserved	Selfrfresh	Suspend	Hibernate	HALTIMER RST	Reserved	Reserved
R/W	R/W	R/W	R	R	R	R/W	R	R/W
RTCRST	0	0	0	0	0	0	0	0
★ Other resets	**Note**	0	0	0	0	0	0	0

Bit	Name	Function
15 to 8	Reserved	0 is returned when read
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
6	Reserved	Write 0 when write. 0 is returned when read.
5	Selfrfresh	Self refresh status 1 : Completed 0 : Not completed
4	Suspend	Suspend mode status (always 0 during Fullspeed mode) 1 : Suspend mode 0 : Other than Suspend mode
3	Hibernate	Hibernate mode status (always 0 during Fullspeed mode) 1 : Hibernate mode 0 : Other than Hibernate mode

★ **Note** Holds the value before reset.

(2/2)

Bit	Name	Function
2	HALTIMERRST	HALTimer reset 1 : Reset 0 : Set This bit is cleared to 0 automatically after reset of the HALTimer Note1, 2.
1	Reserved	0 is returned when read
0	Reserved	Write 0 when write. 0 is returned when read.

Notes1. When the HALTIMERRST bit is cleared to 0 just after set to 1, the HALTimer may not be reset. Wait more than 6 RTC clock cycles from writing 1 to writing 0.

2. Verify that the HALTIMERRST bit is 0 before reset the HALTimer. When this bit is 1, the HALTimer is not reset even if write 1 to this bit. In this case, write 0 to this bit first, then write 1 after more than 6 RTC clock cycles.

This register is used to set CPU core shutdown and overall system operations management.

The HALTIMERRST bit must be reset within about four seconds after activation. Resetting of the HALTIMERRST bit indicates that the VR4181 itself has been activated normally. If the HALTIMERRST bit is not reset within about four seconds after activation, program execution is regarded as abnormal (possibly due to a runaway) and an automatic shutdown is performed.

10.7.3 PMUWAITREG (0x0B00 00A8)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	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
Other resets	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 4	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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15, 14	Reserved	0 is returned when read
13 to 0	WCOUNT(13:0)	Activation wait time timer count value Activation wait time = WCOUNT(13:0) x (1/32.768) ms

Note Holds the value before reset

This register is used to set the activation wait time when the CPU core is activated.

This register is set to 0x2C00 (i.e. 343.75 ms activation wait time) after RTC reset. Therefore, the 343.75 ms wait time is always inserted as an activation wait time, when the CPU core is activated immediately after RTC reset. The activation wait time can be changed by setting this register for the CPU core activation from the Hibernate mode.

When this register is set to 0x0, 0x1, 0x2, 0x3, or 0x4, the operation is not guaranteed. Software must set the value of this register to greater than 0x4 to assure reliable operation.

10.7.4 PMUDIVREG (0x0B00 00AC)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	DIV2	DIV1	DIV0
R/W	R	R	R	R	R	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 3	Reserved	0 is returned when read
2 to 0	DIV(2:0)	Divide mode 111 : RFU 110 : RFU 101 : RFU 100 : RFU 011 : DIV3 mode 010 : DIV2 mode 001 : DIV1 mode 000 : Default mode (DIV2)

Note Holds the value before reset

This register is used to set CPU core's Div mode. The Div mode setting determines the division rate of the TClock in relation to the pipeline clock (PClock) frequency.

Since the contents of this register are cleared to 0 during an RTC reset, the Div mode setting always DIV2 mode just after RTC reset.

Though the Div mode has been set via this register, the setting does not become effective immediately in the processor's operations. In order to change Div mode, software has to put the CPU core into the Hibernate mode. The Div mode will change when the CPU core wakes up from the Hibernate mode.

10.7.5 DRAMHIBCTL (0x0B00 00B2)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
★ Name	Reserved	Reserved	Reserved	Reserved	OK_STOP _CLK	STOP _CLK	SUSPEND	DRAM_EN
R/W	R	R	R	R	R	R/W	R/W	R/W
★ RTCRST	0	0	0	Undefined	0	0	0	0
★ Other resets	0	0	0	Undefined	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 5	Reserved	0 is returned when read
★ 4	Reserved	An undefined value is returned when read
3	OK_STOP_CLK	Ready to stop clocks 1 : Ready (DRAM is in self refresh mode) 0 : Not ready (MEMC is busy to do burst refresh)
2	STOP_CLK	Clock supply for MEMC 1 : Stop 0 : Supply
1	SUSPEND	Self refresh request. This bit is for software request to MEMC to perform burst refresh and enter self refresh mode 1 : Request 0 : Not request
0	DRAM_EN	DRAM interface operation enable 1 : Disabled 0 : Enabled (normal mode)

Note Holds the value before reset

CHAPTER 11 REALTIME CLOCK UNIT (RTC)

This chapter describes the RTC unit's operations and register settings.

11.1 General

The RTC unit has a total of three timers, including the following two types.

- RTCLong ..... This is a 24-bit programmable counter that counts down by 32.768 kHz clock cycle. Cycle interrupts can be occurred for up to every 512 seconds. The RTC unit of the VR4181 includes two RTCLong timers.

- ElapsedTime..... This is a 48-bit up counter that counts up by 32.768 kHz clock cycle. It counts up to 272 years before returning to zero. It includes 48-bit comparator (ECMPLREG, ECMPMREG, and ECMPHREG) and 48-bit alarm time register (ETIMELREG, ETIMEMREG, and ETIMEHREG) to enable interrupts to occur at specified times.

11.2 Register Set

The RTC registers are listed below.

Table 11-1. RTC Registers

Physical address	R/W	Register symbol	Function
0x0B00 00C0	R/W	ETIMELREG	ElapsedTime L register
0x0B00 00C2	R/W	ETIMEMREG	ElapsedTime M register
0x0B00 00C4	R/W	ETIMEHREG	ElapsedTime H register
0x0B00 00C8	R/W	ECMPLREG	ElapsedTime compare L register
0x0B00 00CA	R/W	ECMPMREG	ElapsedTime compare M register
0x0B00 00CC	R/W	ECMPHREG	ElapsedTime compare H register
0x0B00 00D0	R/W	RTCL1LREG	RTCLong1 L register
0x0B00 00D2	R/W	RTCL1HREG	RTCLong1 H register
0x0B00 00D4	R	RTCL1CNTLREG	RTCLong1 count L register
0x0B00 00D6	R	RTCL1CNTHREG	RTCLong1 count H register
0x0B00 00D8	R/W	RTCL2LREG	RTCLong2 L register
0x0B00 00DA	R/W	RTCL2HREG	RTCLong2 H register
0x0B00 00DC	R	RTCL2CNTLREG	RTCLong2 count L register
0x0B00 00DE	R	RTCL2CNTHREG	RTCLong2 count H register
0x0B00 01DE	R/W	RTCINTREG	RTC interrupt register

Each register is described in detail below.

11.2.1 ElapsedTime registers

(1) ETIMELREG (0x0B00 00C0)

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
Other resets	**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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 0	ETIME(15:0)	ElapsedTime timer bits 15 to 0

Note Continues counting.

(2) ETIMEMREG (0x0B00 00C2)

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
Other resets	**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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 0	ETIME(31:16)	ElapsedTime timer bits 31 to 16

Note Continues counting.

(3) ETIMEHREG (0x0B00 00C4)

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
Other resets	**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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 0	ETIME(47:32)	ElapsedTime timer bits 47 to 32

Note Continues counting

These registers indicate the ElapsedTime timer's value. They count up by a 32.768 kHz clock cycle and when a match occurs with the ElapsedTime compare registers, an alarm (ElapsedTime interrupt) occurs (and the counting continues). A write operation 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 timer value changes during a read operation. When using the read value as a data, be sure to read these registers twice and check that two read vales are the same.

When setting these registers again, wait until at least 100 μs (three cycles of 32.768 kHz clock) have elapsed after the first setting.

11.2.2 ElapsedTime compare registers

(1) ECMPLREG (0x0B00 00C8)

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
Other resets	**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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 0	ECMP(15:0)	Value to be compared with ElapsedTime timer bits 15 to 0

Note Holds the value before reset.

(2) ECMPMREG (0x0B00 00CA)

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
Other resets	**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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 0	ECMP(31:16)	Value to be compared with ElapsedTime timer bits 31 to 16

Note Holds the value before reset.

(3) ECMPHREG (0x0B00 00CC)

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
Other resets	**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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 0	ECMP(47:32)	Value to be compared with ElapsedTime timer bits 47 to 32

Note Holds the value before reset.

Use these registers to set the values to be compared with values in the ElapsedTime registers.

A write operation 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 (three cycles of 32.768 kHz clock) have elapsed after the first setting.

11.2.3 RTCLong1 registers

(1) RTCL1LREG (0x0B00 00D0)

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
Other resets	**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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 0	RTCL1P(15:0)	Bits 15 to 0 for RTCLong1 timer count cycle

Note Holds the value before reset.

(2) RTCL1HREG (0x0B00 00D2)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	**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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 8	Reserved	0 is returned when read
7 to 0	RTCL1P(23:16)	Bits 23 to 16 for RTCLong1 timer count cycle

Note Holds the value before reset.

Use these registers to set the RTCLong1 timer count cycle. The RTCLong1 timer begins its countdown at the value written to these registers.

A write operation is valid once values have been written to both registers (RTCL1LREG and RTCL1HREG).

When setting these registers again, wait until at least 100 μs (three cycles of 32.768 kHz clock) have elapsed after the first setting.

Cautions 1. The RTCLong1 timer is stopped when all zeros are written.

2. Any combined setting of "RTCL1HREG = 0x0000" and "RTCL1LREG = 0x0001, 0x0002, 0x0003, or 0x0004" is prohibited.

11.2.4 RTCLong1 count registers

(1) RTCL1CNTLREG (0x0B00 00D4)

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
Other resets	**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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 0	RTCL1C(15:0)	RTCLong1 timer bits 15 to 0

Note Continues counting.

(2) RTCL1CNTHREG (0x0B00 00D6)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	**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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 8	Reserved	0 is returned when read
7 to 0	RTCL1C(23:16)	RTCLong1 timer bits 23 to 16

Note Continues counting.

These registers indicate the RTCLong1 timer's values. It counts down by a 32.768 kHz clock cycle and begins counting at the value set to the RTCLong1 registers. An RTCLong1 interrupt occurs when the timer value reaches 0x00 0001 (at which point the timer returns to the start value and continues counting).

These registers have no buffers for read. Therefore, an illegal data may be read if the timer value changes during a read operation. When using the read value as a data, be sure to read the registers twice and check that two read vales are the same.

11.2.5 RTCLong2 registers

(1) RTCL2LREG (0x0B00 00D8)

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
Other resets	**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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 0	RTCL2P(15:0)	Bits 15 to 0 for RTCLong2 timer count cycle

Note Holds the value before reset.

(2) RTCL2HREG (0x0B00 00DA)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	**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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 8	Reserved	0 is returned when read
7 to 0	RTCL2P(23:16)	Bits 23 to 16 for RTCLong2 timer count cycle

Note Holds the value before reset.

Use these registers to set the RTCLong2 timer count cycle. The RTCLong2 timer begins its countdown at the value written to these registers.

A write operation is valid once values have been written to both registers (RTCL2LREG and RTCL2HREG).

When setting these registers again, wait until at least 100 μs (three cycles of 32.768 kHz clock) have elapsed after the first setting.

Cautions 1. The RTCLong2 timer is stopped when all zeros are written.

2. Any combined setting of "RTCL2HREG = 0x0000" and "RTCL2LREG = 0x0001, 0x0002, 0x0003, or 0x0004" is prohibited.

11.2.6 RTCLong2 count registers

(1) RTCL2CNTLREG (0x0B00 00DC)

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
Other resets	**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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 0	RTCL2C(15:0)	RTCLong2 timer bits 15 to 0

Note Continues counting.

(2) RTCL2CNTHREG (0x0B00 00DE)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	**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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 8	Reserved	0 is returned when read
7 to 0	RTCL2C(23:16)	RTCLong2 timer bits 23 to 16

Note Continues counting.

These registers indicate the RTCLong2 timer's values. It counts down by a 32.768 kHz clock cycle and begins counting at the value set to the RTCLong2 registers. An RTCLong2 interrupt occurs when the timer value reaches 0x00 0001 (at which point the timer returns to the start value and continues counting).

These registers have no buffers for read. Therefore, an illegal data may be read if the timer value changes during a read operation. When using the read value as a data, be sure to read the registers twice and check that two read vales are the same.

11.2.7 RTC interrupt register

(1) RTCINTREG (0x0B00 01DE)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	RTCINTR2	RTCINTR1	RTCINTR0
R/W	R	R	R	R	R	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 3	Reserved	0 is returned when read
2	RTCINTR2	RTCLong2 interrupt request. Cleared to 0 when 1 is written. 1 : Occurred 0 : Normal
1	RTCINTR1	RTCLong1 interrupt request. Cleared to 0 when 1 is written. 1 : Occurred 0 : Normal
0	RTCINTR0	ElapsedTime interrupt request. Cleared to 0 when 1 is written. 1 : Occurred 0 : Normal

Note Holds the value before reset.

This register indicates the occurrences of interrupt requests of RTC.

CHAPTER 12 DEADMAN'S SWITCH UNIT (DSU)

This chapter describes operations and register settings of the DSU (Deadman's Switch Unit).

12.1 General

The DSU detects runaway (endless loop) state of the VR4181 and resets the VR4181. Use of the DSU allows terminating runaway states that may occur due to software in earlier phase to minimize data loss.

12.2 Register Set

The DSU registers are listed below.

Table 12-1. DSU Registers

Physical address	R/W	Register symbol	Function
0x0B00 00E0	R/W	DSUCNTREG	DSU control register
0x0B00 00E2	R/W	DSUSETREG	DSU cycle setting register
0x0B00 00E4	W	DSUCLRREG	DSU clear register
0x0B00 00E6	R/W	DSUTIMREG	DSU elapsed time register

Each register is described in detail below.

12.2.1 DSUCNTREG (0x0B00 00E0)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	DSWEN
R/W	R	R	R	R	R	R	R	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 1	Reserved	0 is returned when read
0	DSWEN	Deadman's Switch function enable 1 : Enabled 0 : Disabled

This register is used to enable use of the Deadman's Switch function.

12.2.2 DSUSETREG (0x0B00 00E2)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	DEDTIME3	DEDTIME2	DEDTIME1	DEDTIME0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	1
Other resets	0	0	0	0	0	0	0	1

Bit	Name	Function
15 to 4	Reserved	0 is returned when read
3 to 0	DEDTIME(3:0)	Deadman's Switch cycle setting 1111 : 15 seconds 1110 : 14 seconds : 0010 : 2 seconds 0001 : 1 second 0000 : Setting prohibited

This register is used to set the cycle for Deadman's Switch function.

The Deadman's Switch cycle can be set in 1-second units in a range from 1 to 15 seconds. The DSWCLR bit in the DSUCLRREG register must be set by means of software within the cycle time specified in this register.

The VR4181's operation is undefined when 0x0 has been set to DEDTIME(3:0).

12.2.3 DSUCLRREG (0x0B00 00E4)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	DSWCLR
R/W	R	R	R	R	R	R	R	W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 1	Reserved	0 is returned when read
0	DSWCLR	Deadman's Switch timer clear 1 : Clear (stops timer) 0 : Timer counting

The Deadman's Switch timer is cleared by setting the DSWCLR bit in this register to 1.

★ The VR4181 automatically enters in a Cold Reset status if 1 is not written to this register within the period specified in the DSUSETREG register.

In order to restart operation of the timer, the DSWCLR bit in this register must be cleared to 0.

12.2.4 DSUTIMREG (0x0B00 00E6)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	CRTTIME3	CRTTIME2	CRTTIME1	CRTTIME0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 4	Reserved	0 is returned when read
3 to 0	CRTTIME(3:0)	Current Deadman's Switch timer value (elapsed time) 1111 : 15 seconds 1110 : 14 seconds : 0010 : 2 seconds 0001 : 1 second 0000 : Setting prohibited

This register indicates the elapsed time of the current Deadman's Switch timer.

12.3 Register Setting Flow

The DSU register setting flow is described below.

<1> Set the DSU timer count cycle (from 1 to 15 seconds).
Register: DSUSETREG, address: 0x0B00 00E2, data: 0x000x

The CPU core will be reset if the timer is not cleared (1 is not written to DSUCLRREG register) within this time period.

<2> Enable the DSU.
Register: DSUCNTREG, address: 0x0B00 00E0, data: 0x0001

<3> Clear the timer within the time period specified in step 1 above. Cancel the clearance of the timer to start another counting.
Register: DSUCLRREG, address: 0x0B00 00E4, data: 0x0001 (timer clear)
Register: DSUCLRREG, address: 0x0B00 00E4, data: 0x0000 (timer operation start)

For normal use, repeat step 3.
To obtain the current elapsed time, read the contents (4 bits) of the DSUTIMREG register (address: 0x0B00 00E6).

<4> Disable the DSU during Suspend mode or a shutdown.
Register: DSUCNTREG, address: 0x0B00 00E0, data: 0x0000

CHAPTER 13 GENERAL PURPOSE I/O UNIT (GIU)

13.1 Overview

13.1.1 GPIO pins and alternate functions

The VR4181 provides 32 general-purpose I/O divided into two groups of 16 pins each. The first group, GPIO(15:0) pins, are capable of supporting the following types of functions:

- Clocked serial interface (CSI)
- Serial interface channel 2
- Color LCD interface (upper 4-bit data) or CompactFlash Card Detect inputs
- General-purpose outputs
- Interrupt/wake-up inputs
- Programmable chip selects
- External ISA system clock output

Any of GPIO(15:0) pins can be used as interrupt/wake-up inputs.

The assignment of interface signals to particular GPIO pins is shown in the following table:

Table 13-1. Alternate Functions of GPIO(15:0) Pins

GPIO pin	Alternate signal 1	Alternate signal 2	Definition
GPIO15	FPD7	CD2#	Color LCD data bit output or Card Detect 2 input
GPIO14	FPD6	CD1#	Color LCD data bit output or Card Detect 1 input
GPIO13	FPD5	–	Color LCD data bit output
GPIO12	FPD4	–	Color LCD data bit output
GPIO11	PCS1#	–	Programmable chip select 1 output.
GPIO10	FRM	SYSCLK	CSI FRM input or SYSCLK output
GPIO9	CTS2#	–	SIU2 CTS input
GPIO8	DSR2#	–	SIU2 DSR input
GPIO7	DTR2#	–	SIU2 DTR output
GPIO6	RTS2#	–	SIU2 RTS output
GPIO5	DCD2#	–	SIU2 DCD input
GPIO4	–	–	–
GPIO3	PCS0#	–	Programmable chip select 0 output.
GPIO2	SCK	–	CSI serial clock input
GPIO1	SO	–	CSI serial data output
GPIO0	SI	–	CSI serial data input

The second group, GPIO(31:16) pins, are capable of supporting the following types of functions:

- External ISA I/O interface
- External 16-bit bus sizing signal
- ROM chip select
- Serial interface channel 1
- General-purpose input
- General-purpose output

Remark GPIO(31:16) pins can not be used as interrupt/wake-up inputs.

The assignment of interface signals to particular GPIO pins is shown in the following table:

Table 13-2. Alternate Functions of GPIO(31:16) Pins

GPIO pin	Alternate signal 1	Alternate signal 2	Definition
GPIO31	DSR1#	–	SIU1 DSR input
GPIO30	DTR1#	–	SIU1 DTR output
GPIO29	DCD1#	–	SIU1 DCD input
GPIO28	CTS1#	–	SIU1 CTS input
GPIO27	RTS1#	–	SIU1 RTS output
GPIO26	TxD1	–	SIU1 transmit data output
GPIO25	RxD1	–	SIU1 receive data input
GPIO24	ROMCS2#	–	ROM chip select for bank 2
GPIO23	ROMCS1#	–	ROM chip select for bank 1
GPIO22	ROMCS0#	–	ROM chip select for bank 0
GPIO21	RESET#	–	External ISA reset
GPIO20[Note]	UBE#	M	External ISA upper byte enable or LCD modulation output
GPIO19	IOCS16#	–	External ISA I/O 16-bit bus sizing
GPIO18	IORDY	–	External ISA I/O channel ready
GPIO17	IOWR#	–	External ISA I/O write strobe
GPIO16	IORD#	–	External ISA I/O read strobe

Note This signal supports input only.

The GPIO29/DCD1# pin can be used as an activation (wake-up) factor from Hibernate mode if enabled by software. The other pins listed above are only capable of providing general-purpose input or output, or the alternate function listed.

13.1.2 I/O direction control

For each GPIO pin, the GIU provides register fields of one buffer enable, GPENn, one output data, GPOn, and one input data, GPIn. The function of each GPIO pin is decoded by 2 register bits in one of the GPIO Mode registers. The most significant bit, GPnMD1, controls the input/output direction of the GPIO pin while the system is powered (during Fullspeed, Standby, or Suspend mode). When this bit is set to 1, the GPIO pin is normally configured as an output.

During Hibernate mode, the GPIO buffer enables are controlled by the GPHIBSTH and GPHIBSTL registers.

Remark n = 0 to 31

★ ### 13.1.3 General-purpose registers

The GIU includes sixteen 16-bit general-purpose registers. Since the contents of these registers are preserved even during Hibernate mode, these registers can be used by system software to save the state of selected registers located in the 2.5 V block prior to entering Hibernate mode. Once the VR4181 has resumed from Hibernate mode, system software can then restore the state of those 2.5 V registers from the general-purpose registers.

The general-purpose registers are located in the address range of 0x0B00 0330 to 0x0B00 034F.

13.2 Alternate Functions Overview

13.2.1 Clocked serial interface (CSI)

The clocked serial interface is enabled by writing to the GPIO Mode registers and utilizes the following GPIO pins:

Table 13-3. CSI Interface Signals

GPIO pin	CSI signal	Type
GPIO2	SCK	Input
GPIO1	SO	Output
GPIO0	SI	Input
GPIO10	FRM	Input

The GPIO10/FRM pin provides a multifunction control input option. In one mode, FRM determines data direction (transmit or receive). In the other mode, FRM prohibits transfer depending on its input level. This mode is set in bit 15, FRMEN, of the CSIMODE register (address: 0x0B00 0900) (see **CHAPTER 8 CLOCKED SERIAL INTERFACE UNIT (CSI)**).

13.2.2 Serial interface channels 1 and 2

The GIU also provides pin mapping for the serial interface (equivalent to 16550 UART) channels 1 and 2.

The serial interface channel 1 (SIU1) is enabled by writing to the GPIO Mode registers. It utilizes the following GPIO pins:

Table 13-4. Serial Interface Channel 1 (SIU1) Signals

GPIO pin	SIU1 signal	Type
GPIO26	TxD1	Output
GPIO25	RxD1	Input
GPIO31	DSR1#	Input
GPIO30	DTR1#	Output
GPIO28	CTS1#	Input
GPIO27	RTS1#	Output
GPIO29	DCD1#	Input

The GIU drives inputs to the serial interface channel 1 based on the settings in the GPIO Mode registers and bit 15, LOOPBK1, of the GPSICTL register (address: 0x0B00 031A) (for additional information, see **13.3.14 GPSICTL (0x0B00 031A)**).

When GPIO pins have been assigned to provide the serial interface channel 1 inputs, RxD1, DTR1#, RTS1#, and DCD1#, the GIU simply passes the signals driven on the GPIO pins to the corresponding serial interface channel 1 inputs. Otherwise, the GIU drives these signals based on the value programmed in the GPSICTL register as follows:

Table 13-5. Serial Interface Channel 1 (SIU1) Loopback Control

LOOPBK1 bit value	Source for driving SIU1 input
0	DSR1#: REGDSR1 (bit 9) value CTS1#: REGCTS1 (bit 10) value DCD1#: REGDCD1 (bit 8) value RxD1: REGRXD1 (bit 11) value
1	DSR1#: DTR1# output CTS1#: RTS1# output DCD1#: REGDCD1 (bit 8) value RxD1: REGRXD1 (bit 11) value

The serial interface channel 2 (SIU2) utilizes the dedicated IRDIN/RxD2 and IRDOUT/TxD2 pins. The line control signals, DTR2#, RTS2#, DCD2#, DSR2#, and CTS2#, are enabled by writing to the GPIO Mode registers and are utilized through the following GPIO pins:

Table 13-6. Serial Interface Channel 2 (SIU2) Signals

GPIO pin	SIU2 signal	Type
GPIO9	CTS2#	Input
GPIO8	DSR2#	Input
GPIO6	RTS2#	Output
GPIO5	DCD2#	Input
GPIO7	DTR2#	Output

The transmit and receive data signals, TxD2 and RxD2, are enabled by writing to the SIUIRSEL_2 register in the SIU2.

Control of the serial interface channel 2 line status inputs is identical to that of the serial interface channel 1. The GIU drives inputs to the serial interface channel 2 based on the settings in the GPIO Mode registers and bit 7, LOOPBK2, of the GPSICTL register (address: 0x0B00 031A) (for additional information, see **13.3.14 GPSICTL (0x0B00 031A)**).

When GPIO pins have been assigned to provide the serial interface channel 2 inputs, DTR2#, RTS2#, and DCD2#, the GIU simply passes the signals driven on the GPIO pins to the corresponding serial interface channel 2 inputs. Otherwise, the GIU drives these signals based on the value programmed in the GPSICTL register as follows:

Table 13-7. Serial Interface Channel 2 (SIU2) Loopback Control

LOOPBK2 bit value	Source for driving SIU2 input
0	DSR2#: REGDSR2 (bit 1) value CTS2#: REGCTS2 (bit 2) value DCD2#: REGDCD2 (bit 0) value
1	DSR2#: DTR2# output CTS2#: RTS2# output DCD2#: REGDCD2 (bit 0) value

Note that the GIU does not drive the RxD2 input. This signal is always available to the serial interface as either IRDIN or RxD2.

13.2.3 LCD interface

The GIU supports two functions for the LCD interface. The first is pin mapping for 8-bit STN color LCD panel support. The second is pin mapping for support of an external LCD controller with integrated frame buffer RAM.

For additional details about the LCD registers, see **CHAPTER 21 LCD CONTROLLER**.

(1) STN color LCD interface pin mapping

The color LCD panel interface is enabled by writing to the GPIO Mode registers and utilizes the following GPIO pins:

Table 13-8. STN Color LCD Interface Signals

GPIO pin	LCD signal	Type
GPIO(15:12)	FPD(7:4)	Output

(2) External LCD controller pin mapping

An interface to an external LCD controller can be configured by setting the LCDGPEN bit of the LCDGPMODE register to 1. In this mode the following internal LCD controller pins are redefined to support the external LCD controller interface:

Table 13-9. External LCD Controller Interface Signals

LCD pin	External LCD controller interface signal	Type
SHCLK	LCDCS#	Output
LOCLK	MEMCS16#	Input
VPLCD	General-purpose output (VPGPIO1)	Output
VPBIAS	General-purpose output (VPGPIO0)	Output

The LCDCS# output is generated by the address decode logic in the GIU. The address range can be specified by programming the LCDGPMODE register. The following address ranges are supported:

(1) 0x1338 0000 to 0x133F FFFF (512KB)
(2) 0x133C 0000 to 0x133F FFFF (256KB)
(3) 0x133E 0000 to 0x133F FFFF (128KB)
(4) 0x130A 0000 to 0x130A FFFF (64KB, the address space of the PC/AT™ is assumed)

Remark All memory cycles that access the external LCD controller address space are treated as 16-bit cycles.

The MEMCS16# input is provided to support external memory devices (besides the external LCD controller) which need accesses in 16-bit cycles. During an external memory cycle, if the MEMCS16# input is enabled and asserted, the ISA bridge will generate a 16-bit cycle.

13.2.4 Programmable chip selects

The GIU provides two programmable chip select signals, PCS(1:0)#. These chip select signals are available on the following GPIO pins:

Table 13-10. Programmable Chip Select Signals

GPIO pin	Programmable chip select	Type
GPIO11	PCS1#	Output
GPIO3	PCS0#	Output

Each programmable chip select signal can be defined individually as memory- or I/O-mapped, 8- or 16-bit data width, and 1 to 64K bytes of address ranges. The chip selects can also be qualified with I/O or memory read strobes.

13.2.5 16-bit bus cycles

The GIU generates two internal outputs (gpiocs16_l and gpmemcs16_l) to the internal ISA bus to signal the data width of the target of an external ISA cycle. The internal ISA bus uses these outputs as the IOCS16# and MEMCS16# signals that are AND'ed with the outputs from other internal ISA units.

The gpiocs16_l output is controlled by either a programmable chip select set in the PCSMODE register (0x0B00 032C) or IOCS16#/GPIO19 pin. When one of the programmable chip selects has been defined as I/O mapped and 16-bit data width, the gpiocs16_l output is asserted while the I/O cycle address is within the range specified for the programmable chip select. When the IOCS16#/GPIO19 pin has been configured as IOCS16#, the gpiocs16_l output follows the state of the IOCS16# signal.

★ The gpmemcs16_l output is controlled by a programmable chip select or the LOCLK/MEMCS16# pin. When one of the programmable chip selects has been defined as memory mapped and 16-bit data width, the gpmemcs16_l output is asserted while the memory cycle address is within the range specified for the programmable chip select. When the LOCLK/MEMCS16# pin has been configured as MEMCS16#, the gpmemcs16_l output follows the state of the MEMCS16# signal.

13.2.6 General purpose input/output

Each one of the 32 GPIO pins can be defined as a general-purpose input or a general-purpose output. When a pin is configured as a general-purpose output, a corresponding value written to the GPDATLREG register or the GPDATHREG register appears on the GPIO pin. When a pin is configured as a general-purpose input, a value driven on the GPIO pin can be read from its corresponding data bit of the GPDATLREG or GPDATHREG register.

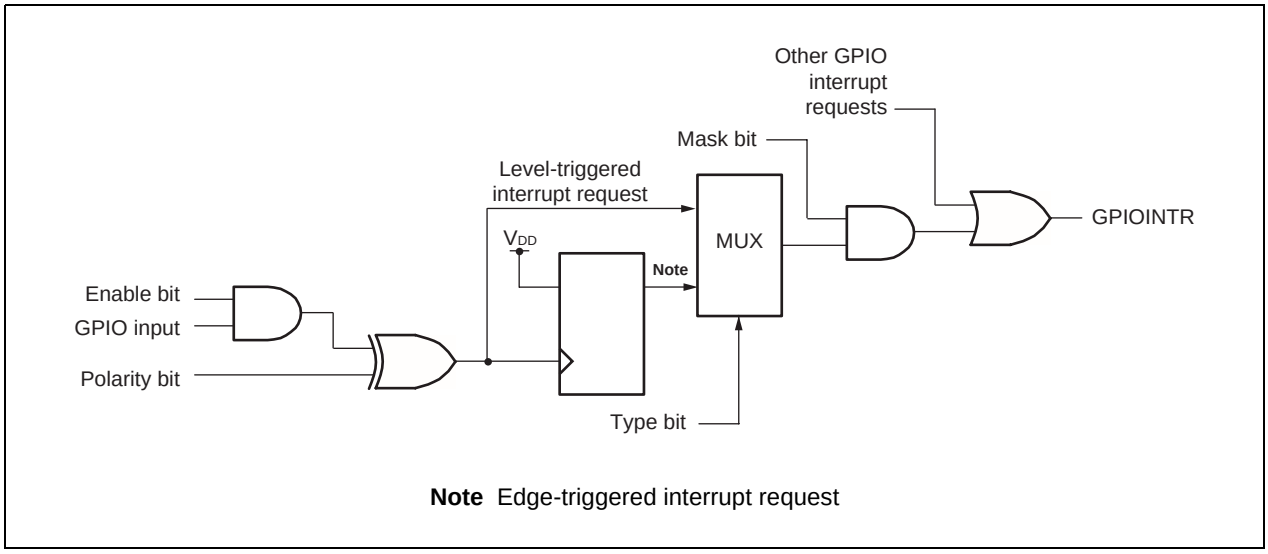
13.2.7 Interrupt requests and wake-up events

Each of the lower sixteen GPIO pins, GPIO(15:0), can be defined as an interrupt request input. The GIU provides a single asynchronous interrupt request output to the MBA Host Bridge, GPIOINTR. The MBA Host Bridge is responsible for synchronizing this interrupt request with the MasterOut clock (internal).

The GIU provides a total of five registers to support GPIO interrupt requests. The interrupt enable register, GPINTEN, is used to enable interrupt requests on a particular GPIO pin. The interrupt mask register, GPINTMSK, ★ permits temporary masking of an interrupt request for a particular GPIO pin. The interrupt type registers, GPINTTYPH and GPINTTYPL, define the interrupt trigger type (edge or level) and the level type (polarity) of the interrupt requests input to the GPIO pin. The interrupt status register, GPINTSTAT, allows software to determine the source of the GPIO interrupt request.

The functions of the enable, mask, polarity, and type bits are shown in the following figure:

Figure 13-1. GPIO(15:0) Interrupt Request Detecting Logic



Note Edge-triggered interrupt request

During Hibernate mode, any one of the GPIO(15:0) inputs can be used as a wake-up event. Wake-up event notification is asynchronous and output on the GPWAKEUP signal (internal)[Note]. To enable GPIO wake-up events, the following conditions must be met.

(1) Interrupt requests to the GPIO pin must be enabled (set in the GPINTEN register).
(2) Interrupt requests to the GPIO pin must be unmasked (set in the GPINTMSK register).
(3) The GPIO pin must be enabled during Hibernate mode (set in the GPHIBSTL register).

Note The state of this signal is displayed on the GPWAKEUP bit of the PMUINTREG register in the PMU.

13.3 Register Set

The GIU provides the following registers.

Table 13-11. GIU Registers (1/2)

Physical address	R/W	Register symbol	Function
0x0B00 0300	R/W	GPMD0REG	GPIO Mode 0 register
0x0B00 0302	R/W	GPMD1REG	GPIO Mode 1 register
0x0B00 0304	R/W	GPMD2REG	GPIO Mode 2 register
0x0B00 0306	R/W	GPMD3REG	GPIO Mode 3 register
0x0B00 0308	R/W	GPDATHREG	GPIO data high register
0x0B00 030A	R/W	GPDATLREG	GPIO data low register
0x0B00 030C	R/W	GPINTEN	GPIO interrupt enable register
0x0B00 030E	R/W	GPINTMSK	GPIO interrupt mask register
0x0B00 0310	R/W	GPINTTYPH	GPIO interrupt type high register
0x0B00 0312	R/W	GPINTTYPL	GPIO interrupt type low register
0x0B00 0314	R/W	GPINTSTAT	GPIO interrupt status register
0x0B00 0316	R/W	GPHIBSTH	GPIO Hibernate pin status high register
0x0B00 0318	R/W	GPHIBSTL	GPIO Hibernate pin status low register
0x0B00 031A	R/W	GPSICTL	GPIO serial interface control register
0x0B00 031C	R/W	KEYEN	Keyboard scan pin enable register
0x0B00 0320	R/W	PCS0STRA	Programmable chip select 0 start address register
0x0B00 0322	R/W	PCS0STPA	Programmable chip select 0 stop address register
0x0B00 0324	R/W	PCS0HIA	Programmable chip select 0 high address register
0x0B00 0326	R/W	PCS1STRA	Programmable chip select 1 start address register
0x0B00 0328	R/W	PCS1STPA	Programmable chip select 1 stop address register
0x0B00 032A	R/W	PCS1HIA	Programmable chip select 1 high address register
0x0B00 032C	R/W	PCSMODE	Programmable chip select mode register
0x0B00 032E	R/W	LCDGPMODE	LCD general-purpose mode register

Table 13-11. GIU Registers (2/2)

Physical address	R/W	Register symbol	Function
★ 0x0B00 0330	R/W	MISCREG0	General-purpose register
0x0B00 0332	R/W	MISCREG1	
0x0B00 0334	R/W	MISCREG2	
0x0B00 0336	R/W	MISCREG3	
0x0B00 0338	R/W	MISCREG4	
0x0B00 033A	R/W	MISCREG5	
0x0B00 033C	R/W	MISCREG6	
0x0B00 033E	R/W	MISCREG7	
0x0B00 0340	R/W	MISCREG8	
0x0B00 0342	R/W	MISCREG9	
0x0B00 0344	R/W	MISCREG10	
0x0B00 0346	R/W	MISCREG11	
0x0B00 0348	R/W	MISCREG12	
0x0B00 034A	R/W	MISCREG13	
0x0B00 034C	R/W	MISCREG14	
0x0B00 034E	R/W	MISCREG15	

13.3.1 GPMD0REG (0x0B00 0300)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	GP7MD1	GP7MD0	GP6MD1	GP6MD0	GP5MD1	GP5MD0	GP4MD1	GP4MD0
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	GP3MD1	GP3MD0	GP2MD1	GP2MD0	GP1MD1	GP1MD0	GP0MD1	GP0MD0
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15, 14	GP7MD(1:0)	These bits control direction and function of the GPIO7 pin as follows: 00 : General-purpose input 01 : RFU 10 : General-purpose output 11 : SIU2 DTR2# output
13, 12	GP6MD(1:0)	These bits control direction and function of the GPIO6 pin as follows: 00 : General-purpose input 01 : RFU 10 : General-purpose output 11 : SIU2 RTS2# output
11, 10	GP5MD(1:0)	These bits control direction and function of the GPIO5 pin as follows: 00 : General-purpose input 01 : SIU2 DCD2# input 10 : General-purpose output 11 : RFU
9, 8	GP4MD(1:0)	These bits control direction and function of the GPIO4 pin as follows: 00 : General-purpose input 01 : RFU 10 : General-purpose output 11 : RFU
7, 6	GP3MD(1:0)	These bits control direction and function of the GPIO3 pin as follows: 00 : General-purpose input 01 : RFU 10 : General-purpose output 11 : Programmable chip select 0 output

Note Holds the value before reset

(2/2)

Bit	Name	Function
5, 4	GP2MD(1:0)	These bits control direction and function of the GPIO2 pin as follows: 00 : General-purpose input 01 : CSI SCK input 10 : General-purpose output 11 : RFU
3, 2	GP1MD(1:0)	These bits control direction and function of the GPIO1 pin as follows: 00 : General-purpose input 01 : RFU 10 : General-purpose output 11 : CSI SO output
1, 0	GP0MD(1:0)	These bits control direction and function of the GPIO0 pin as follows: 00 : General-purpose input 01 : CSI SI input 10 : General-purpose output 11 : RFU

13.3.2 GPMD1REG (0x0B00 0302)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	GP15MD1	GP15MD0	GP14MD1	GP14MD0	GP13MD1	GP13MD0	GP12MD1	GP12MD0
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	GP11MD1	GP11MD0	GP10MD1	GP10MD0	GP9MD1	GP9MD0	GP8MD1	GP8MD0
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15, 14	GP15MD(1:0)	These bits control direction and function of the GPIO15 pin as follows: 00 : General-purpose input 01 : CD2# input 10 : General-purpose output 11 : Color LCD FPD7 output
13, 12	GP14MD(1:0)	These bits control direction and function of the GPIO14 pin as follows: 00 : General-purpose input 01 : CD1# input 10 : General-purpose output 11 : Color LCD FPD6 output
11, 10	GP13MD(1:0)	These bits control direction and function of the GPIO13 pin as follows: 00 : General-purpose input 01 : RFU 10 : General-purpose output 11 : Color LCD FPD5 output
9, 8	GP12MD(1:0)	These bits control direction and function of the GPIO12 pin as follows: 00 : General-purpose input 01 : RFU 10 : General-purpose output 11 : Color LCD FPD4 output
7, 6	GP11MD(1:0)	These bits control direction and function of the GPIO11 pin as follows: 00 : General-purpose input 01 : RFU 10 : General-purpose output 11 : Programmable chip select 1 output

Note Holds the value before reset

Remark When GPIO15 and GPIO14 pins are not defined as CD2# and CD1# signals respectively, the corresponding internal card detect signals to CompactFlash controller (ECU) are held to low level (active).

(2/2)

Bit	Name	Function
5, 4	GP10MD(1:0)	These bits control direction and function of the GPIO10 pin as follows: 00 : General-purpose input 01 : CSI FRM input 10 : General-purpose output 11 : SYSCLK output
3, 2	GP9MD(1:0)	These bits control direction and function of the GPIO9 pin as follows: 00 : General-purpose input 01 : SIU2 CTS2# input 10 : General-purpose output 11 : RFU
1, 0	GP8MD(1:0)	These bits control direction and function of the GPIO8 pin as follows: 00 : General-purpose input 01 : SIU2 DSR2# input 10 : General-purpose output 11 : RFU

13.3.3 GPMD2REG (0x0B00 0304)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	GP23MD1	GP23MD0	GP22MD1	GP22MD0	GP21MD1	GP21MD0	GP20MD1	GP20MD0
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	GP19MD1	GP19MD0	GP18MD1	GP18MD0	GP17MD1	GP17MD0	GP16MD1	GP16MD0
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15, 14	GP23MD(1:0)	These bits control direction and function of the GPIO23 pin as follows: 00 : General-purpose input 01 : RFU 10 : General-purpose output 11 : ROMCS1# output
13, 12	GP22MD(1:0)	These bits control direction and function of the GPIO22 pin as follows: 00 : General-purpose input 01 : RFU 10 : General-purpose output 11 : ROMCS0# output
11, 10	GP21MD(1:0)	These bits control direction and function of the GPIO21 pin as follows: 00 : General-purpose input 01 : RFU 10 : General-purpose output 11 : RESET# output
9, 8	GP20MD(1:0)	These bits control direction and function of the GPIO20 pin as follows: 00 : General-purpose input 01 : RFU 10 : LCD M output 11 : UBE# output
7, 6	GP19MD(1:0)	These bits control direction and function of the GPIO19 pin as follows: 00 : General-purpose input 01 : IOCS16# input 10 : General-purpose output 11 : RFU

Note Holds the value before reset

Caution LCD M output can not be used in the VR4181 of Rev.1.0.

(2/2)

Bit	Name	Function
5, 4	GP18MD(1:0)	These bits control direction and function of the GPIO18 pin as follows: 00 : General-purpose input 01 : IORDY input 10 : General-purpose output 11 : RFU
3, 2	GP17MD(1:0)	These bits control direction and function of the GPIO17 pin as follows: 00 : General-purpose input 01 : RFU 10 : General-purpose output 11 : IOWR# output
1, 0	GP16MD(1:0)	These bits control direction and function of the GPIO16 pin as follows: 00 : General-purpose input 01 : RFU 10 : General-purpose output 11 : IORD# output

13.3.4 GPMD3REG (0x0B00 0306)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	GP31MD1	GP31MD0	GP30MD1	GP30MD0	GP29MD1	GP29MD0	GP28MD1	GP28MD0
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	GP27MD1	GP27MD0	GP26MD1	GP26MD0	GP25MD1	GP25MD0	GP24MD1	GP24MD0
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15, 14	GP31MD(1:0)	These bits control direction and function of the GPIO31 pin as follows: 00 : General-purpose input 01 : SIU1 DSR1# input 10 : General-purpose output 11 : RFU
13, 12	GP30MD(1:0)	These bits control direction and function of the GPIO30 pin as follows: 00 : General-purpose input 01 : RFU 10 : General-purpose output 11 : SIU1 DTR1# output
11, 10	GP29MD(1:0)	These bits control direction and function of the GPIO29 pin as follows: 00 : General-purpose input 01 : SIU1 DCD1# input 10 : General-purpose output 11 : RFU
9, 8	GP28MD(1:0)	These bits control direction and function of the GPIO28 pin as follows: 00 : General-purpose input 01 : SIU1 CTS1# input 10 : General-purpose output 11 : RFU
7, 6	GP27MD(1:0)	These bits control direction and function of the GPIO27 pin as follows: 00 : General-purpose input 01 : RFU 10 : General-purpose output 11 : SIU1 RTS1# output

Note Holds the value before reset

(2/2)

Bit	Name	Function
5, 4	GP26MD(1:0)	These bits control direction and function of the GPIO26 pin as follows: 00 : General-purpose input 01 : RFU 10 : General-purpose output 11 : SIU1 TxD1 output
3, 2	GP25MD(1:0)	These bits control direction and function of the GPIO25 pin as follows: 00 : General-purpose input 01 : SIU1 RxD1 input 10 : General-purpose output 11 : RFU
1, 0	GP24MD(1:0)	These bits control direction and function of the GPIO24 pin as follows: 00 : General-purpose input 01 : RFU 10 : General-purpose output 11 : ROMCS2# output

13.3.5 GPDATHREG (0x0B00 0308)

Bit	15	14	13	12	11	10	9	8
Name	GPDAT31	GPDAT30	GPDAT29	GPDAT28	GPDAT27	GPDAT26	GPDAT25	GPDAT24
★ 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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	GPDAT23	GPDAT22	GPDAT21	GPDAT20	GPDAT19	GPDAT18	GPDAT17	GPDAT16
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 0	GPDAT(31:16)	General-purpose data. There is a one-to-one correspondence between these bits and GPIO pins. When a GPIO pin is configured as a general-purpose input, the value of the pin can be read from this register. When the pin is defined as a general-purpose output, the value written to this register appears on the GPIO pin. When one of the GPIO(31:16) pins is configured as other function, the corresponding bit value in this register is invalid.

Note Holds the value before reset

13.3.6 GPDATLREG (0x0B00 030A)

Bit	15	14	13	12	11	10	9	8
Name	GPDAT15	GPDAT14	GPDAT13	GPDAT12	GPDAT11	GPDAT10	GPDAT9	GPDAT8
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	GPDAT7	GPDAT6	GPDAT5	GPDAT4	GPDAT3	GPDAT2	GPDAT1	GPDAT0
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 0	GPDAT(15:0)	General-purpose data. There is a one-to-one correspondence between these bits and GPIO pins. When a GPIO pin is configured as a general-purpose input, the value of the pin can be read from this register. When the pin is defined as a general-purpose output, the value written to this register appears on the GPIO pin. When one of the GPIO(15:0) pins is configured as other function, the corresponding bit value in this register is invalid.

Note Holds the value before reset

13.3.7 GPINTEN (0x0B00 030C)

Bit	15	14	13	12	11	10	9	8
Name	GIEN15	GIEN14	GIEN13	GIEN12	GIEN11	GIEN10	GIEN9	GIEN8
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	GIEN7	GIEN6	GIEN5	GIEN4	GIEN3	GIEN2	GIEN1	GIEN0
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 0	GIEN(15:0)	GPIO interrupt enable. There is a one-to-one correspondence between these bits and GPIO pins. When one of the GPIO(15:0) pins is defined as a general-purpose input, the corresponding bit in this register enables interrupts for that pin as follows: 0 : Interrupt disabled 1 : Interrupt enabled

Note Holds the value before reset

Remark About the relationship between the GPINTEN and GPINTMSK registers, refer to **Figure 13-1. GPIO(15:0) Interrupt Request Detecting Logic**.

13.3.8 GPINTMSK (0x0B00 030E)

Bit	15	14	13	12	11	10	9	8
Name	GIMSK15	GIMSK14	GIMSK13	GIMSK12	GIMSK11	GIMSK10	GIMSK9	GIMSK8
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	GIMSK7	GIMSK6	GIMSK5	GIMSK4	GIMSK3	GIMSK2	GIMSK1	GIMSK0
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 0	GIMSK(15:0)	GPIO interrupt mask. There is a one-to-one correspondence between these bits and GPIO pins. When a GPIO pin is defined as a general-purpose input and interrupts is enabled on that pin, the interrupt can be temporarily masked by setting the corresponding bit in this register as follows: 0 : Interrupt unmasked 1 : Interrupt masked

Note Holds the value before reset

Remark About the relationship between the GPINTEN and GPINTMSK registers, refer to **Figure 13-1. GPIO(15:0) Interrupt Request Detecting Logic**.

13.3.9 GPINTTYPH (0x0B00 0310)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	I15TYP1	I15TYP0	I14TYP1	I14TYP0	I13TYP1	I13TYP0	I12TYP1	I12TYP0
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	I11TYP1	I11TYP0	I10TYP1	I10TYP0	I9TYP1	I9TYP0	I8TYP1	I8TYP0
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15, 14	I15TYP(1:0)	These bits define the type of interrupt generated when the GPIO15 pin is defined as a general-purpose input: 00 : Negative edge triggered interrupt 01 : Positive edge triggered interrupt 10 : Low level triggered interrupt 11 : High level triggered interrupt
13, 12	I14TYP(1:0)	These bits define the type of interrupt generated when the GPIO14 pin is defined as a general-purpose input: 00 : Negative edge triggered interrupt 01 : Positive edge triggered interrupt 10 : Low level triggered interrupt 11 : High level triggered interrupt
11, 10	I13TYP(1:0)	These bits define the type of interrupt generated when the GPIO13 pin is defined as a general-purpose input: 00 : Negative edge triggered interrupt 01 : Positive edge triggered interrupt 10 : Low level triggered interrupt 11 : High level triggered interrupt
9, 8	I12TYP(1:0)	These bits define the type of interrupt generated when the GPIO12 pin is defined as a general-purpose input: 00 : Negative edge triggered interrupt 01 : Positive edge triggered interrupt 10 : Low level triggered interrupt 11 : High level triggered interrupt

Note Holds the value before reset

(2/2)

Bit	Name	Function
7, 6	I11TYP(1:0)	These bits define the type of interrupt generated when the GPIO11 pin is defined as a general-purpose input: 00 : Negative edge triggered interrupt 01 : Positive edge triggered interrupt 10 : Low level triggered interrupt 11 : High level triggered interrupt
5, 4	I10TYP(1:0)	These bits define the type of interrupt generated when the GPIO10 pin is defined as a general-purpose input: 00 : Negative edge triggered interrupt 01 : Positive edge triggered interrupt 10 : Low level triggered interrupt 11 : High level triggered interrupt
3, 2	I9TYP(1:0)	These bits define the type of interrupt generated when the GPIO9 pin is defined as a general-purpose input: 00 : Negative edge triggered interrupt 01 : Positive edge triggered interrupt 10 : Low level triggered interrupt 11 : High level triggered interrupt
1, 0	I8TYP(1:0)	These bits define the type of interrupt generated when the GPIO8 pin is defined as a general-purpose input: 00 : Negative edge triggered interrupt 01 : Positive edge triggered interrupt 10 : Low level triggered interrupt 11 : High level triggered interrupt

13.3.10 GPINTTYPL (0x0B00 0312)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	I7TYP1	I7TYP0	I6TYP1	I6TYP0	I5TYP1	I5TYP0	I4TYP1	I4TYP0
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	I3TYP1	I3TYP0	I2TYP1	I2TYP0	I1TYP1	I1TYP0	I0TYP1	I0TYP0
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15, 14	I7TYP(1:0)	These bits define the type of interrupt generated when the GPIO7 pin is defined as a general-purpose input: 00 : Negative edge triggered interrupt 01 : Positive edge triggered interrupt 10 : Low level triggered interrupt 11 : High level triggered interrupt
13, 12	I6TYP(1:0)	These bits define the type of interrupt generated when the GPIO6 pin is defined as a general-purpose input: 00 : Negative edge triggered interrupt 01 : Positive edge triggered interrupt 10 : Low level triggered interrupt 11 : High level triggered interrupt
11, 10	I5TYP(1:0)	These bits define the type of interrupt generated when the GPIO5 pin is defined as a general-purpose input: 00 : Negative edge triggered interrupt 01 : Positive edge triggered interrupt 10 : Low level triggered interrupt 11 : High level triggered interrupt
9, 8	I4TYP(1:0)	These bits define the type of interrupt generated when the GPIO4 pin is defined as a general-purpose input: 00 : Negative edge triggered interrupt 01 : Positive edge triggered interrupt 10 : Low level triggered interrupt 11 : High level triggered interrupt

Note Holds the value before reset

(2/2)

Bit	Name	Function
7, 6	I3TYP(1:0)	These bits define the type of interrupt generated when the GPIO3 pin is defined as a general-purpose input: 00 : Negative edge triggered interrupt 01 : Positive edge triggered interrupt 10 : Low level triggered interrupt 11 : High level triggered interrupt
5, 4	I2TYP(1:0)	These bits define the type of interrupt generated when the GPIO2 pin is defined as a general-purpose input: 00 : Negative edge triggered interrupt 01 : Positive edge triggered interrupt 10 : Low level triggered interrupt 11 : High level triggered interrupt
3, 2	I1TYP(1:0)	These bits define the type of interrupt generated when the GPIO1 pin is defined as a general-purpose input: 00 : Negative edge triggered interrupt 01 : Positive edge triggered interrupt 10 : Low level triggered interrupt 11 : High level triggered interrupt
1, 0	I0TYP(1:0)	These bits define the type of interrupt generated when the GPIO0 pin is defined as a general-purpose input: 00 : Negative edge triggered interrupt 01 : Positive edge triggered interrupt 10 : Low level triggered interrupt 11 : High level triggered interrupt

13.3.11 GPINTSTAT (0x0B00 0314)

Bit	15	14	13	12	11	10	9	8
Name	GISTS15	GISTS14	GISTS13	GISTS12	GISTS11	GISTS10	GISTS9	GISTS8
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	GISTS7	GISTS6	GISTS5	GISTS4	GISTS3	GISTS2	GISTS1	GISTS0
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 0	GISTS(15:0)	GPIO interrupt request status. There is a one-to-one correspondence between these bits and GPIO pins. When a GPIO pin is defined as a general-purpose input, these bits reflect the interrupt request status as follows: 0 : No Interrupt request pending 1 : Interrupt request pending

Note Holds the value before reset

Interrupt request pending status is reflected regardless of the setting of the interrupt mask bits. Therefore, the status of an interrupt request can be returned as pending when this register is read even though the interrupt is masked.

When a GPIO interrupt request is defined as an edge triggered type, the interrupt request is cleared by writing 1 to the corresponding bit of this register. For example, if GPIO11 is defined as an edge triggered interrupt request input, an interrupt request generated by this pin would be cleared by writing 1 to the bit 11 of this register.

13.3.12 GPHIBSTH (0x0B00 0316)

Bit	15	14	13	12	11	10	9	8
Name	GPHST31	GPHST30	GPHST29	GPHST28	GPHST27	GPHST26	GPHST25	GPHST24
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	GPHST23	GPHST22	GPHST21	GPHST20	GPHST19	GPHST18	GPHST17	GPHST16
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 0	GPHST(31:16)	GPIO Hibernate pin state control. There is a one-to-one correspondence between these bits and GPIO pins. These bits determine the state of GPIO(31:16) pins during Hibernate mode as follows: 0 : Output pin is in high impedance Input pin is ignored during Hibernate mode 1 : Output pin remains actively driven Input pin is monitored during Hibernate mode

Note Holds the value before reset

★ **Caution GPIO29 pin (DCD1#) can be input at high level and monitored during Hibernate mode and therefore the GPHST29 bit can be set to 1. The GPHST bits for all other GPIO pins configured as inputs should be reset to 0.**

13.3.13 GPHIBSTL (0x0B00 0318)

Bit	15	14	13	12	11	10	9	8
Name	GPHST15	GPHST14	GPHST13	GPHST12	GPHST11	GPHST10	GPHST9	GPHST8
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	GPHST7	GPHST6	GPHST5	GPHST4	GPHST3	GPHST2	GPHST1	GPHST0
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 0	GPHST(15:0)	GPIO Hibernate pin state control. There is a one-to-one correspondence between these bits and GPIO pins. These bits determine the state of GPIO(15:0) pins during Hibernate mode as follows: 0 : Output pin is in high impedance Input pin is ignored during Hibernate mode 1 : Output pin remains actively driven Input pin is monitored during Hibernate mode

Note Holds the value before reset

Remark In order to support wake-up events on one of the GPIO(15:0) pins, the associated GPHST bit must be set to 1.

13.3.14 GPSICTL (0x0B00 031A)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	LOOPBK1	Reserved	Reserved	Reserved	REGRXD1	REGCTS1	REGDSR1	REGDCD1
R/W	R/W	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	1	1	1	1
Other resets	**Note**	0	0	0	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	LOOPBK2	Reserved	Reserved	Reserved	Reserved	REGCTS2	REGDSR2	REGDCD2
R/W	R/W	R	R	R	R	R/W	R/W	R/W
RTCRST	0	0	0	0	0	1	1	1
Other resets	**Note**	0	0	0	0	**Note**	**Note**	**Note**

Bit	Name	Function
15	LOOPBK1	Loopback enable for serial interface channel 1. When GPIO pins have not been allocated for the line status signals DSR1# and/or CTS1# of the serial interface channel 1, this bit can be set to 1 to allow the serial interface line status output signals to be connected to the line status input signals as follows: DTR1# output from serial interface drives the DSR1# input to serial interface RTS1# output from serial interface drives the CTS1# input to serial interface
14 to 12	Reserved	0 is returned when read
11	REGRXD1	RxD1 data. When a GPIO pin has not been enabled to provide RxD1, the RxD1 input to the serial interface channel 1 is driven with the value of this bit.
10	REGCTS1	CTS1# data. When the LOOPBK1 bit is reset to 0 and a GPIO pin has not been enabled to provide CTS1#, the CTS1# input to the serial interface channel 1 is driven with the value of this bit.
9	REGDSR1	DSR1# data. When the LOOPBK1 bit is reset to 0 and a GPIO pin has not been enabled to provide DSR1#, the DSR1# input to the serial interface channel 1 is driven with the value of this bit.
8	REGDCD1	DCD1# data. When a GPIO pin has not been enabled to provide DCD1#, the DCD1# input to the serial interface channel 1 is driven with the value of this bit.

Note Holds the value before reset

(2/2)

Bit	Name	Function
7	LOOPBK2	Loopback enable for serial interface channel 2. When GPIO pins have not be allocated for the line status signals DSR2# and/or CTS2# of the serial interface channel 2, this bit can be set to 1 to allow the serial interface line status output signals to be connected to the line status input signals as follows: DTR2# output from serial interface drives the DSR2# input to serial interface RTS2# output from serial interface drives the CTS2# input to serial interface
6 to 3	Reserved	0 is returned when read
2	REGCTS2	CTS2# data. When the LOOPBK2 bit is reset to 0 and a GPIO pin has not been enabled to provide CTS2#, the CTS2# input to the serial interface channel 2 is driven with the value of this bit.
1	REGDSR2	DSR2# data. When the LOOPBK2 bit is reset to 0 and a GPIO pin has not been enabled to provide DSR2#, the DSR2# input to the serial interface channel 2 is driven with the value of this bit.
0	REGDCD2	DCD2# data. When a GPIO pin has not been enabled to provide DCD2#, the DCD2# input to the serial interface channel 2 is driven with the value of this bit.

13.3.15 KEYEN (0x0B00 031C)

Bit	15	14	13	12	11	10	9	8
Name	KEYSEL	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R/W	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	**Note**	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	CFHIBEN	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R/W	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	**Note**	0	0	0	0	0	0	0

Bit	Name	Function
15	KEYSEL	Keyboard scan pin enable. This bit causes the pins assigned to support the CompactFlash interface to be redefined to support the keyboard scan interface. 0 : CompactFlash interface enabled 1 : Keyboard scan interface enabled
14 to 8	Reserved	0 is returned when read
7	CFHIBEN	CompactFlash interface enable during Hibernate mode 0 : Disable 1 : Enable
6 to 0	Reserved	0 is returned when read

Note Holds the value before reset

★ The GIU only provides an internal output signal (keysel) when the KEYSEL bit is set to 1. An external logic is responsible for multiplexing the pin inputs and pin outputs, and I/O buffer enable control from the ECU and the KIU.

When the CompactFlash interface is enabled during Hibernate mode, a high-to-low transition on the CompactFlash CF_BUSY# pin will cause the VR4181 to wake up and return to Fullspeed mode.

13.3.16 PCS0STRA (0x0B00 0320)

Bit	15	14	13	12	11	10	9	8
Name	PCS0STRA 15	PCS0STRA 14	PCS0STRA 13	PCS0STRA 12	PCS0STRA 11	PCS0STRA 10	PCS0STRA 9	PCS0STRA 8
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	PCS0STRA 7	PCS0STRA 6	PCS0STRA 5	PCS0STRA 4	PCS0STRA 3	PCS0STRA 2	PCS0STRA 1	PCS0STRA 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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 0	PCS0STRA(15:0)	Programmable chip select 0 start address. These bits determine the starting address for the memory or I/O chip select.

Note Holds the value before reset

13.3.17 PCS0STPA (0x0B00 0322)

Bit	15	14	13	12	11	10	9	8
Name	PCS0STPA 15	PCS0STPA 14	PCS0STPA 13	PCS0STPA 12	PCS0STPA 11	PCS0STPA 10	PCS0STPA 9	PCS0STPA 8
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	PCS0STPA 7	PCS0STPA 6	PCS0STPA 5	PCS0STPA 4	PCS0STPA 3	PCS0STPA 2	PCS0STPA 1	PCS0STPA 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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 0	PCS0STPA(15:0)	Programmable chip select 0 stop address. These bits determine the ending address for the memory or I/O chip select.

Note Holds the value before reset

13.3.18 PCS0HIA (0x0B00 0324)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	PCS0HIA 27	PCS0HIA 26	PCS0HIA 25	PCS0HIA 24
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	**Note1**	**Note1**	**Note1**	**Note1**

Bit	7	6	5	4	3	2	1	0
Name	PCS0HIA 23	PCS0HIA 22	PCS0HIA 21	PCS0HIA 20	PCS0HIA 19	PCS0HIA 18	PCS0HIA 17	PCS0HIA 16
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
Other resets	**Note1**	**Note1**	**Note1**	**Note1**	**Note1**	**Note1**	**Note1**	**Note1**

Bit	Name	Function
15 to 12	Reserved	0 is returned when read
11 to 0	PCS0HIA(27:16)	Programmable chip select 0 high address. A programmable chip select 0 will be generated when all of the following conditions have been met: • The system address bits A(15:0) are equal to or greater than PCS0STRA(15:0) and equal to or less than PCS0STPA(15:0) **Note2** • The internal address bits A(27:16) are equal to PCS0HIA(27:16) • The read/write qualifier conditions specified by the PCSMODE register have been met.

Notes 1. Holds the value before reset

2. When the PCS0 has been defined as a 16-bit chip select, bit 0 of the address is ignored.

13.3.19 PCS1STRA (0x0B00 0326)

Bit	15	14	13	12	11	10	9	8
Name	PCS1STRA 15	PCS1STRA 14	PCS1STRA 13	PCS1STRA 12	PCS1STRA 11	PCS1STRA 10	PCS1STRA 9	PCS1STRA 8
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	PCS1STRA 7	PCS1STRA 6	PCS1STRA 5	PCS1STRA 4	PCS1STRA 3	PCS1STRA 2	PCS1STRA 1	PCS1STRA 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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 0	PCS1STRA(15:0)	Programmable chip select 1 start address. These bits determine the starting address for the memory or I/O chip select.

Note Holds the value before reset

13.3.20 PCS1STPA (0x0B00 0328)

Bit	15	14	13	12	11	10	9	8
Name	PCS1STPA 15	PCS1STPA 14	PCS1STPA 13	PCS1STPA 12	PCS1STPA 11	PCS1STPA 10	PCS1STPA 9	PCS1STPA 8
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	PCS1STPA 7	PCS1STPA 6	PCS1STPA 5	PCS1STPA 4	PCS1STPA 3	PCS1STPA 2	PCS1STPA 1	PCS1STPA 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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 0	PCS1STPA(15:0)	Programmable chip select 1 stop address. These bits determine the ending address for the memory or I/O chip select.

Note Holds the value before reset

13.3.21 PCS1HIA (0x0B00 032A)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	PCS1HIA 27	PCS1HIA 26	PCS1HIA 25	PCS1HIA 24
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	**Note1**	**Note1**	**Note1**	**Note1**

Bit	7	6	5	4	3	2	1	0
Name	PCS1HIA 23	PCS1HIA 22	PCS1HIA 21	PCS1HIA 20	PCS1HIA 19	PCS1HIA 18	PCS1HIA 17	PCS1HIA 16
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
Other resets	**Note1**	**Note1**	**Note1**	**Note1**	**Note1**	**Note1**	**Note1**	**Note1**

Bit	Name	Function
15 to 12	Reserved	0 is returned when read
11 to 0	PCS1HIA(27:16)	Programmable chip select 1 high address. A programmable chip select 1 will be generated when all of the following conditions have been met: • The system address bits A(15:0) are equal to or greater than PCS1STRA(15:0) and equal to or less than PCS1STPA(15:0) **Note2** • The internal address bits A(27:16) are equal to PCS1HIA(27:16) • The read/write qualifier conditions specified by the PCSMODE register have been met.

Notes 1. Holds the value before reset

2. When the PCS1 has been defined as a 16-bit chip select, bit 0 of the address is ignored.

13.3.22 PCSMODE (0x0B00 032C)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	PCS1MIOB	PCS1DSIZE	PCS1MD1	PCS1MD0	PCS0MIOB	PCS0DSIZE	PCS0MD1	PCS0MD0
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 8	Reserved	0 is returned when read
7	PCS1MIOB	Programmable chip select 1 target cycle 0 : Enabled only during I/O cycles 1 : Enabled only during memory cycles
6	PCS1DSIZE	Programmable chip select 1 data size 0 : Defined as an 8-bit device. During accesses to the address range specified for PCS1, 8-bit cycles will be generated unless MEMCS16# or IOCS16# is asserted. 1 : Defined as a 16-bit device. During accesses to the address range specified for PCS1 16-bit cycles will be generated.
5, 4	PCS1MD(1:0)	Programmable chip select 1 mode 00 : Disabled 01 : Qualified also with I/O or memory read strobe 10 : Qualified also with I/O or memory write strobe 11 : Based on address decode only
3	PCS0MIOB	Programmable chip select 0 target cycle 0 : Enabled only during I/O cycles 1 : Enabled only during memory cycles
2	PCS0DSIZE	Programmable chip select 0 data size 0 : Defined as an 8-bit device. During accesses to the address range specified for PCS0, 8-bit cycles will be generated unless MEMCS16# or IOCS16# is asserted. 1 : Defined as a 16-bit device. During accesses to the address range specified for PCS0 16-bit cycles will be generated.
1, 0	PCS0MD(1:0)	Programmable chip select 0 mode 00 : Disabled 01 : Qualified also with I/O or Memory read strobe 10 : Qualified also with I/O or Memory write strobe 11 : Based on address decode only

Note Holds the value before reset

13.3.23 LCDGPMODE (0x0B00 032E)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	LCDGPEN	Reserved	Reserved	Reserved	LCDCS1	LCDCS0	GPVPBIAS	GPVPLCD
R/W	R/W	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	**Note**	0	0	0	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 8	Reserved	0 is returned when read
7	LCDGPEN	Control unit of LCD interface signals 0 : Controlled by internal LCD controller 1 : Controlled by external LCD controller SHCLK … LCDCS# LOCLK … MEMCS16# VPLCD … driven by the GPVPLCD bit of this register VPBIAS … driven by the GPVPBIAS bit of this register
6 to 4	Reserved	0 is returned when read
3, 2	LCDCS(1:0)	External LCD controller frame buffer address select. These bits determine the address range that will cause the LCDCS# signal to be asserted. 00 : 0x130A 0000 to 0x130A FFFF (64KB PC/AT compatible address space) 01 : 0x133E 0000 to 0x133F FFFF (128KB) 10 : 0x133C 0000 to 0x133F FFFF (256KB) 11 : 0x1338 0000 to 0x133F FFFF (512KB)
1	GPVPBIAS	Output control for VPBIAS pin. When the LCDGPEN bit is set to 1, the VPBIAS pin is driven by the value of this bit.
0	GPVPLCD	Output control for VPLCD pin. When the LCDGPEN bit is set to 1, the VPLCD pin is driven by the value of this bit.

Note Holds the value before reset

13.3.24 MISCREGn (0x0B00 0330 to 0x0B00 034E)

Remark n = 0 to 15

MISCREG0 (0x0B00 0330)
MISCREG1 (0x0B00 0332)
MISCREG2 (0x0B00 0334)
MISCREG3 (0x0B00 0336)
MISCREG4 (0x0B00 0338)
MISCREG5 (0x0B00 033A)
MISCREG6 (0x0B00 033C)
MISCREG7 (0x0B00 033E)
MISCREG8 (0x0B00 0340)
MISCREG9 (0x0B00 0342)
MISCREG10 (0x0B00 0344)
MISCREG11 (0x0B00 0346)
MISCREG12 (0x0B00 0348)
MISCREG13 (0x0B00 034A)
MISCREG14 (0x0B00 034C)
MISCREG15 (0x0B00 034E)

Bit	15	14	13	12	11	10	9	8
Name	MISCnD15	MISCnD14	MISCnD13	MISCnD12	MISCnD11	MISCnD10	MISCnD9	MISCnD8
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	7	6	5	4	3	2	1	0
Name	MISCnD7	MISCnD6	MISCnD5	MISCnD4	MISCnD3	MISCnD2	MISCnD1	MISCnD0
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
Other resets	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 0	MISCnD(15:0)	Miscellaneous data

Note Holds the value before reset

Remark n = 0 to 15

These registers are battery-backed, and its contents are retained even in Hibernate mode.

CHAPTER 14 TOUCH PANEL INTERFACE UNIT (PIU)

14.1 General

The PIU uses the on-chip A/D converter to detect the X and Y coordinates of pen contact locations on a touch panel and to scan the general-purpose A/D input port. Since the touch panel control circuit and the A/D converter (conversion precision: 10 bits) are both incorporated, the touch panel is connected directly to the VR4181.

The PIU's function, namely the detection of X and Y coordinates, is performed partly by hardware and partly by software.

Hardware tasks:
- Touch panel applied voltage control
- Reception of coordinate data

Software task:
- Processing of coordinate data based on data sampled by hardware

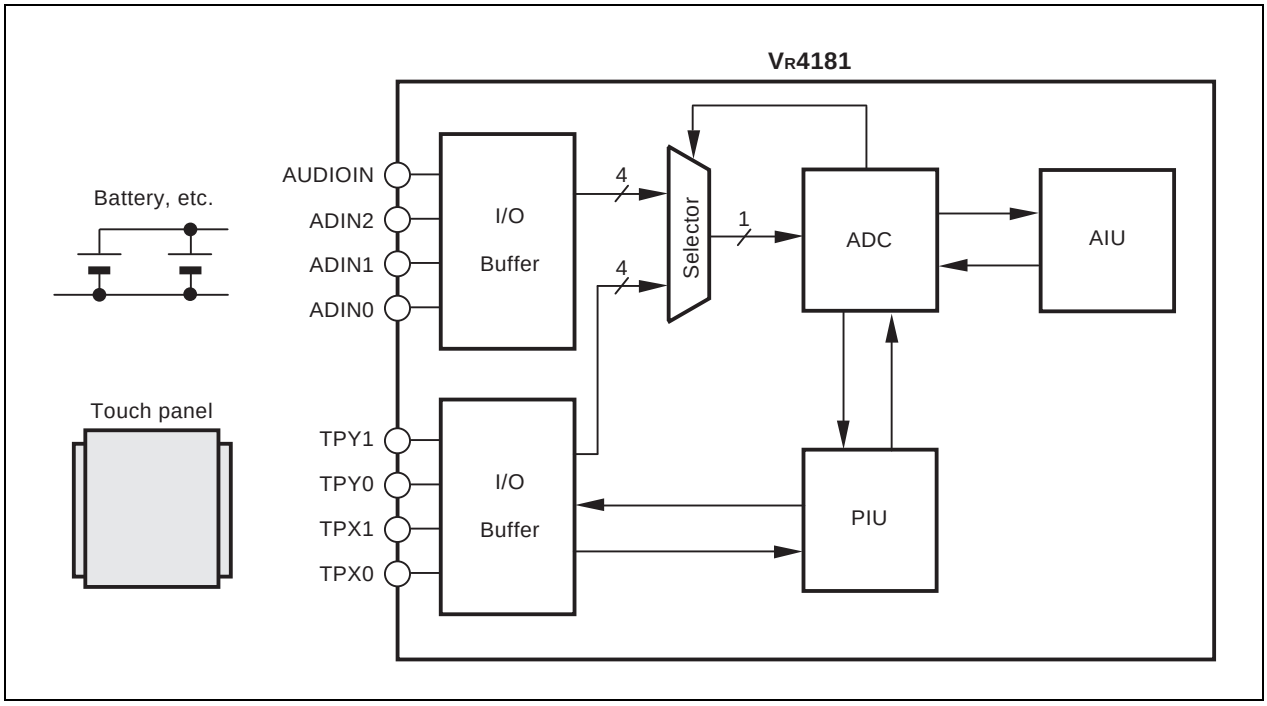
Features of the PIU's hardware tasks are described below.

- Can be directly connected to touch panel with four-pin resistance layers (on-chip touch panel driver)
- Interface for on-chip A/D converter
- Voltage detection at three general-purpose A/D ports and one audio input port
- Operation of A/D converter based on various settings and control of voltage applied to touch panel
- Sampling of X-coordinate and Y-coordinate data
- Variable coordinate data sampling interval
- Interrupt is triggered if pen touch occurs regardless of CPU operation mode (interrupts do not occur during Hibernate mode)
- Four dedicated buffers with up to two pages each for coordinate data
- Four buffers for A/D port scan
- Auto/manual options for coordinate data sampling start/stop control

★ **Caution No clocks are supplied to the PIU, A/D converter, and D/A converter in the initial state. When using the PIU, set the MSKPIUPCLK, MSKADUPCLK, and MSKADU18M bits of the CMUCLKMSK register in the MBA Host Bridge to 1 in advance so that clocks are supplied.**

14.1.1 Block diagrams

Figure 14-1. PIU Peripheral Block Diagram



- **Touch panel**

  A set of four pins are located at the edges of the X-axis and Y-axis resistance layers, and the two layers have high resistance when there is no pen contact and low resistance when there is a pen contact. The resistance between the two edges of the resistance layers is about 1 kΩ. When a voltage is applied to both edges of the Y-axis resistance layer, the voltage (V_{Y1} and V_{Y2} in the figure below) is measures at the X-axis resistance layer's pins to determine the Y coordinate. Similarly, when a voltage is applied to both edges of the X-axis resistance layer, the voltage (V_{X1} and V_{X2} in the figure below) is measures at the Y-axis resistance layer's pins ★ to determine the X coordinate. For greater precision, voltages are again measured after switching plus and minus of the voltage applied to the resistance layer's pins. The obtained data is stored into the PIUPBnmREG register (n = 0 or 1, m = 0 to 3).

Figure 14-2. Coordinate Detection Equivalent Circuits

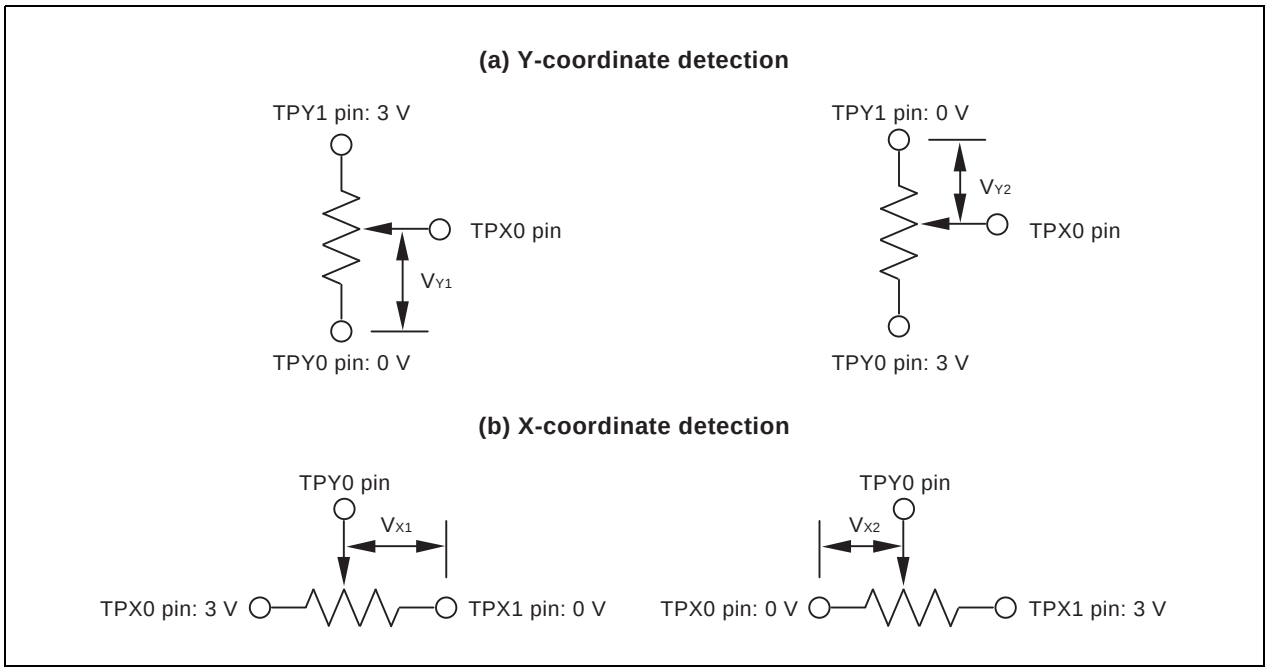
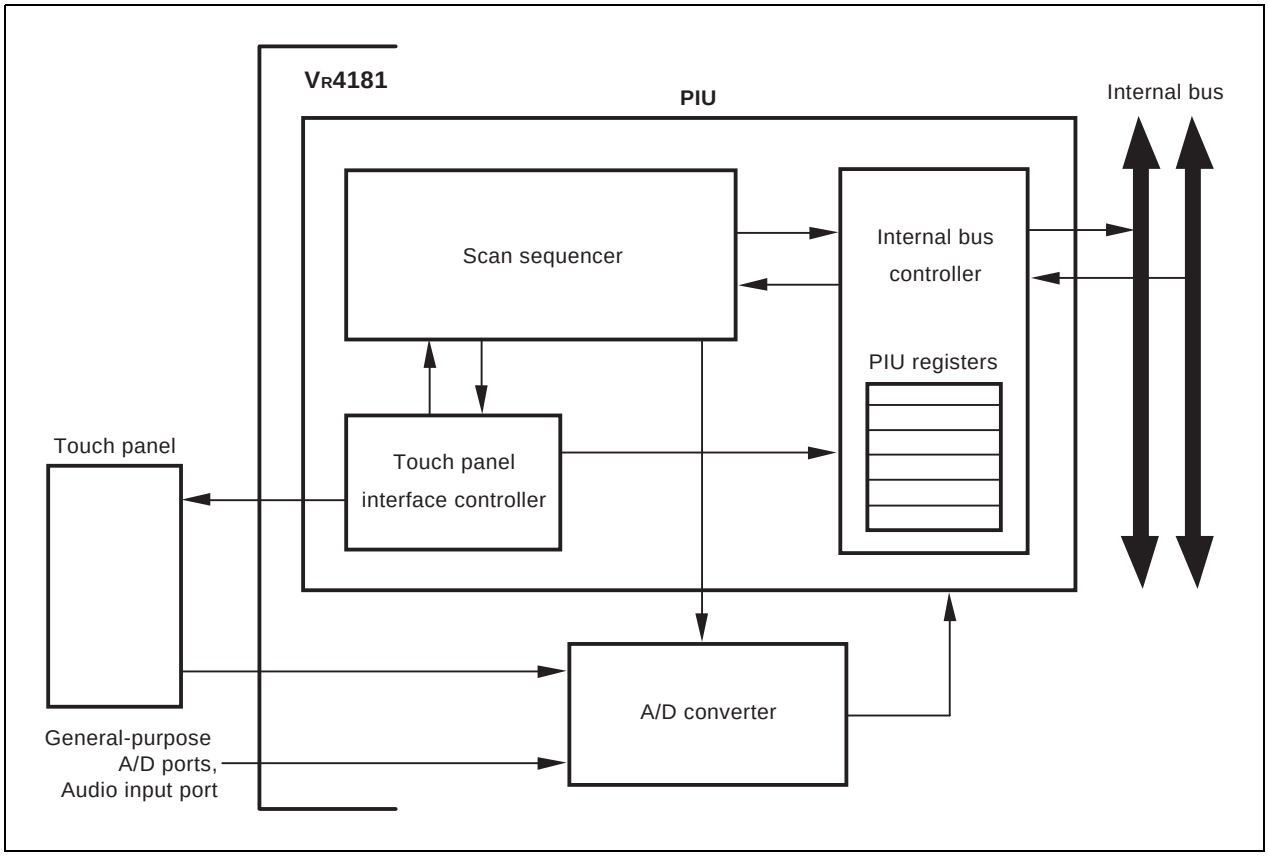


Figure 14-3. Internal Block Diagram of PIU



The PIU includes three blocks: the internal bus controller, the scan sequencer, and the touch panel interface controller.

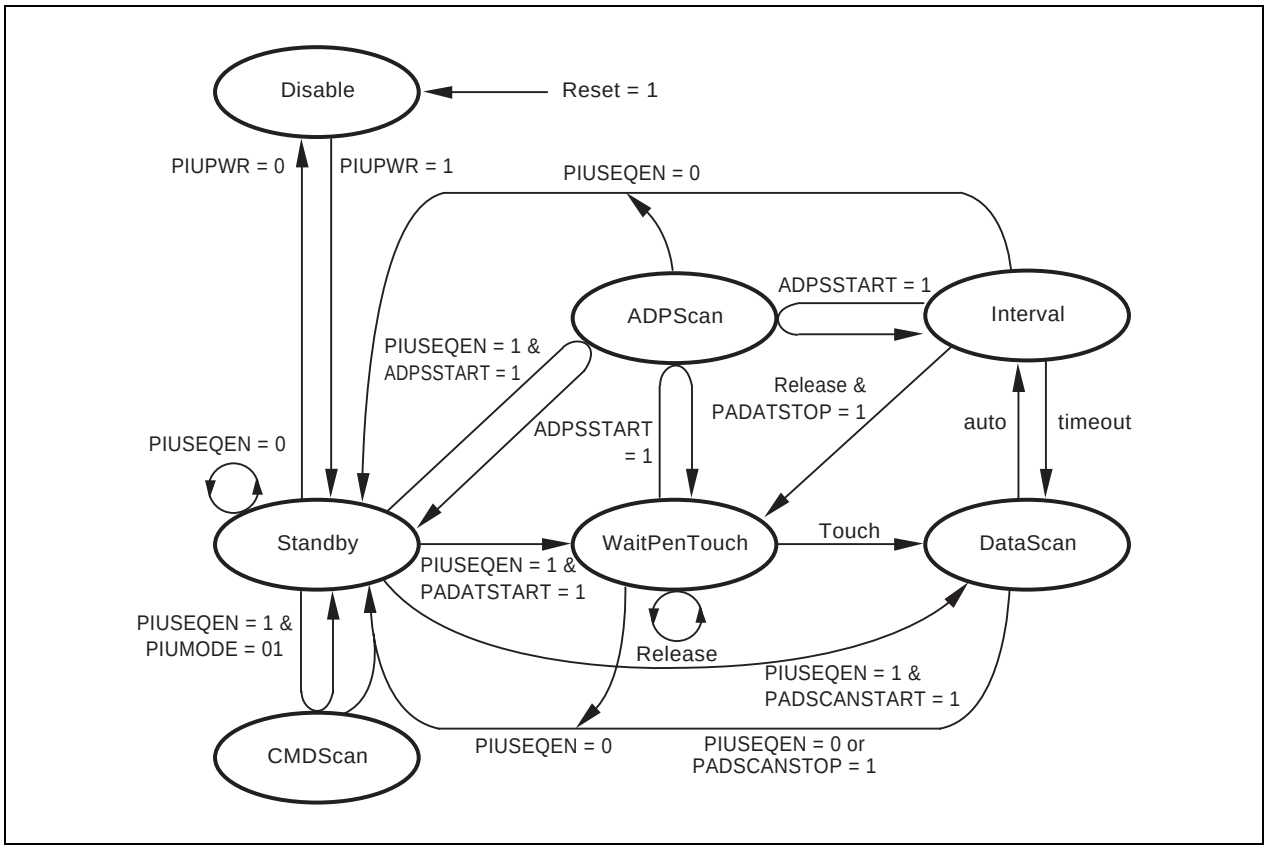
- Internal bus controller
  The internal bus controller controls the internal bus, the PIU registers, and interrupts, and communicates with the A/D converter.

- Scan sequencer
  The scan sequencer is used for PIU state management.

- Touch panel interface controller
  The touch panel interface controller is used to control the touch panel.

14.2 Scan Sequencer State Transition

Figure 14-4. Scan Sequencer State Transition Diagram



- Disable state
  In this state, the A/D converter is in standby status, the output pins are in touch detection status (no PIU interrupt), and the input pins are in mask status (to prevent misoperation when an undefined input is applied).

- Standby state
  In this state, the PIU is in scan idle status. The touch panel is in low-power status (0 V voltage is applied to the touch panel and the A/D converter is in disable status). Normally, this is the state in which various mode settings are made.

  Caution Since a state transition occurs when the PIUSEQEN bit is active, the PIUSEQEN bit must be set as active after various mode settings have been completed.

- ADPScan state
  This is the state in which voltage is measured at the A/D converter's three general-purpose ports and one audio input port. After the A/D converter is activated and voltage data is obtained, the data is stored in the PIU's internal data buffer (PIUABnREG, n = 0 to 3). After the four ports are scanned, an A/D port scan interrupt request occurs. After this interrupt occurs, the ADPSSTART bit is automatically set as inactive and the PIU enters the state in which the ADPSSTART bit has been set as active.

- CMDScan state
  In this state, the A/D converter operates according to various settings. Voltage data from one port only is fetched based on a combination of the touch panel pin setting (TPX(1:0), TPY(1:0)) and the selection of an input port (TPX(1:0), TPY(1:0), AUDIOIN, ADIN(2:0)) to the A/D converter. Use PIUCMDREG register to make the touch panel pin setting and to select the input port.

- WaitPenTouch state
  This is a standby state in which the PIU waits for a touch panel's "touch" status. When the PIU detects a touch panel's "touch" status, a touch panel contact status change interrupt request occurs inside the PIU. At this point, if the PADATSTART bit is active, the PIU enters the DataScan state. During the WaitPenTouch state, it is possible to enter Suspend mode because the panel state can be detected even while the TClock is stopped.

- DataScan state
  This is the state in which touch panel coordinates are detected. The A/D converter is activated and four data for each coordinate are sampled.

  Caution If one complete set of coordinate data is not obtained during the interval between one set of coordinate data and the next coordinate data, a data lost interrupt request occurs.

- Interval state
  This is the standby state in which the PIU waits for the next coordinate sampling period or the touch panel's "release" status. After the touch panel status is detected, the time period specified via PIUSIVLREG register elapses before the transition to the DataScan state. If the PIU detects the "release" status within the specified time period, a touch panel contact status change interrupt request occurs inside the PIU. At this point, the PIU enters the WaitPenTouch state if the PADATSTOP bit is active. If the PADATSTOP bit is inactive, it enters to the DataScan state after the specified time period has elapsed.

14.3 Register Set

The PIU registers are listed below.

Table 14-1. PIU Registers

Physical address	R/W	Register symbol	Function
0x0B00 0122	R/W	PIUCNTREG	PIU Control register
0x0B00 0124	R/W	PIUINTREG	PIU Interrupt cause register
0x0B00 0126	R/W	PIUSIVLREG	PIU Data sampling interval register
0x0B00 0128	R/W	PIUSTBLREG	PIU A/D converter start delay register
0x0B00 012A	R/W	PIUCMDREG	PIU A/D command register
0x0B00 0130	R/W	PIUASCNREG	PIU A/D port scan register
0x0B00 0132	R/W	PIUAMSKREG	PIU A/D scan mask register
★ 0x0B00 013E	R	PIUCIVLREG	PIU data sampling period count register
0x0B00 02A0	R/W	PIUPB00REG	PIU page 0 buffer 0 register
0x0B00 02A2	R/W	PIUPB01REG	PIU page 0 buffer 1 register
0x0B00 02A4	R/W	PIUPB02REG	PIU page 0 buffer 2 register
0x0B00 02A6	R/W	PIUPB03REG	PIU page 0 buffer 3 register
0x0B00 02A8	R/W	PIUPB10REG	PIU page 1 buffer 0 register
0x0B00 02AA	R/W	PIUPB11REG	PIU page 1 buffer 1 register
0x0B00 02AC	R/W	PIUPB12REG	PIU page 1 buffer 2 register
0x0B00 02AE	R/W	PIUPB13REG	PIU page 1 buffer 3 register
0x0B00 02B0	R/W	PIUAB0REG	PIU A/D scan buffer 0 register
0x0B00 02B2	R/W	PIUAB1REG	PIU A/D scan buffer 1 register
0x0B00 02B4	R/W	PIUAB2REG	PIU A/D scan buffer 2 register
0x0B00 02B6	R/W	PIUAB3REG	PIU A/D scan buffer 3 register
0x0B00 02BC	R/W	PIUPB04REG	PIU page 0 buffer 4 register
0x0B00 02BE	R/W	PIUPB14REG	PIU page 1 buffer 4 register

State of interrupt requests caused by the PIU is indicated and can be set in the following registers, which are included in the ICU (refer to **CHAPTER 9 INTERRUPT CONTROL UNIT (ICU)** for details).

Table 14-2. PIU Interrupt Registers

Physical address	R/W	Register symbol	Function
0x0B00 0082	R	PIUINTREG	PIU interrupt indication register
0x0B00 008E	R/W	MPIUINTREG	PIU interrupt mask register

14.3.1 PIUCNTREG (0x0B00 0122)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	PENSTC	PADSTATE2	PADSTATE1	PADSTATE0	PADATSTOP	PADATSTART
R/W	R	R	R	R	R	R	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	PADSCAN STOP	PADSCAN START	PADSCAN TYPE	PIUMODE1	PIUMODE0	PIUSEQEN	PIUPWR	PADRST
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
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15, 14	Reserved	0 is returned when read
13	PENSTC	Touch/release status when touch panel contact state changes 1 : Touch 0 : Release
12 to 10	PADSTATE(2:0)	Scan sequencer status 111 : CMDScan 110 : Interval 101 : DataScan 100 : WaitPenTouch 011 : RFU 010 : ADPScan 001 : Standby 000 : Disable
9	PADATSTOP	Sequencer auto stop setting during touch panel release status 1 : Auto stop after sampling data for one set of coordinates 0 : No auto stop
8	PADATSTART	Sequencer auto start setting during touch panel touch status 1 : Auto start 0 : No auto start
7	PADSCANSTOP	Forced stop setting for touch panel sequencer 1 : Forced stop after sampling data for one set of coordinates 0 : Do not stop

(2/2)

Bit	Name	Function
6	PADSCANSTART	Start setting for touch panel sequencer 1 : Forced start 0 : Do not start
5	PADSCANTYPE	Touch pressure sampling enable 1: Enable 0: Disable
4, 3	PIUMODE(1:0)	PIU mode setting 11 : RFU 10 : RFU 01 : Operates A/D converter using any command 00 : Samples coordinate data
2	PIUSEQEN	Scan sequencer operation enable 1 : Enable 0 : Disable
1	PIUPWR	PIU power mode setting 1 : Sets PIU output as active and puts into standby status 0 : Sets panel to touch detection status and set PIU operation stop enabled status
0	PADRST	PIU reset. Once the PADRST bit is set to 1, it is automatically cleared to 0 after four TClock cycles. 1 : Reset 0 : Do not reset

This register is used to make various settings for the PIU.

The PENSTC bit indicates the touch panel contact status at the time when the PENCHGINTR bit of the PIUINTREG register is set to 1. This bit's state remains as it is until the PENCHGINTR bit is cleared to 0. Also, when the PENCHGINTR bit is cleared to 0, the PENSTC bit indicates the touch panel contact status at that time. However, the PENSTC bit does not change while the PENCHGINTR bit is set to 1, even if the touch panel contact status changes between release and touch.

Some bits in this register cannot be set in a specific state of scan sequencer. The combination of the setting of this register and the sequencer state is as follows.

Table 14-3. PIUCNTREG Bit Manipulation and States

PIUCNTREG bit manipulation		Scan sequencer's state			
		Disable	Standby	WaitPenTouch	DataScan
★ PADRST Note1	0 → 1	–	Disable	Disable	Disable
PIUPWR	0 → 1	Standby	?	×	×
	1 → 0	?	Disable	×	×
PIUSEQEN	0 → 1	×	WaitPenTouch	?	?
	1 → 0	?	?	Standby	Standby
PADATSTART	0 → 1	×	–	DataScan Note2	×
	1 → 0	×	–	–	×
PADATSTOP	0 → 1	×	–	×	×
	1 → 0	×	–	×	×
PADSCANSTART	0 → 1	×	DataScan Note3	×	×
	1 → 0	×	–	×	×
PADSCANSTOP	0 → 1	×	–	×	Standby Note4
	1 → 0	×	–	×	–

PIUCNTREG bit manipulation		Scan sequencer's state		
		Interval	ADPScan	CMDScan
★ PADRST Note1	0 → 1	Disable	Disable	Disable
PIUPWR	0 → 1	?	?	?
	1 → 0	×	×	×
PIUSEQEN	0 → 1	?	?	?
	1 → 0	Standby	Standby	Standby
PADATSTART	0 → 1	×	×	×
	1 → 0	×	×	×
PADATSTOP	0 → 1	×	×	×
	1 → 0	×	×	×
PADSCANSTART	0 → 1	×	×	×
	1 → 0	×	×	×
PADSCANSTOP	0 → 1	Standby Note4	Standby Note4	Standby Note4
	1 → 0	?	–	–

Notes 1. After 1 is written, the bit is automatically cleared to 0 four TClock cycles later.

2. State transition occurs during touch status.

3. State transition occurs when the PIUSEQEN bit is set to 1.

4. State transition occurs after one set of data is sampled. The PADSCANSTOP bit is cleared to 0 after the state transition occurs.

Remark – : The bit change is retained but there is no state transition.

× : Setting prohibited (operation not guaranteed)

? : Combination of state and bit status before setting does not exist.

14.3.2 PIUINTREG (0x0B00 0124)

Bit	15	14	13	12	11	10	9	8
Name	OVP	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R/W	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	PADCMD INTR	PADADP INTR	PADPAGE1 INTR	PADPAGE0 INTR	PADDLOST INTR	Reserved	PENCHG INTR
R/W	R	R/W	R/W	R/W	R/W	R/W	R	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15	OVP	Valid page ID bit (older valid page) 1 : Page 1 retains an older valid data 0 : Page 0 retains an older valid data
14 to 7	Reserved	0 is returned when read
6	PADCMDINTR	PIU command scan interrupt request. This interrupt request occurs when a valid data is obtained during a command scan. Cleared to 0 when 1 is written. 1 : Occurred 0 : Not occurred
5	PADADPINTR	PIU A/D port scan interrupt request. This interrupt request occurs when a set of valid data is obtained during an A/D port scan. Cleared to 0 when 1 is written. 1 : Occurred 0 : Not occurred
4	PADPAGE1INTR	PIU data buffer page 1 interrupt request. This interrupt request occurs when a set of valid data is stored in the page 1 of the data buffer. Cleared to 0 when 1 is written. 1 : Occurred 0 : Not occurred
3	PADPAGE0INTR	PIU data buffer page 0 interrupt request. This interrupt request occurs when a set of valid data is stored in the page 0 of the data buffer. Cleared to 0 when 1 is written. 1 : Occurred 0 : Not occurred
★ 2	PADDLOSTINTR	Data lost interrupt request. This interrupt request occurs when a set of data cannot be obtained during a specified time period. Cleared to 0 when 1 is written. 1 : Occurred 0 : Not occurred
1	Reserved	0 is returned when read
0	PENCHGINTR	Touch panel contact status change interrupt request. Cleared to 0 when 1 is written. 1 : Occurred 0 : Not occurred

This register sets and indicates the interrupt request generation of the PIU.

When the PENCHGINTR bit is set to1, the PENSTC bit of the PIUCNTREG register indicates the touch panel contact status (touch or release) when a contact status changes. The PENSTC bit's status remains until the PENCHGINTR bit is cleared to 0. Also, when the PENCHGINTR bit is cleared to 0, the PENSTC bit indicates the touch panel contact status. However, the PENSTC bit does not change while the PENCHGINTR bit is set to 1, even if the touch panel contact status changes between release and touch.

Caution In the Hibernate mode, the VR4181 retains the touch panel status. Therefore, if the Hibernate mode has been entered while the touch panel is touched, the contact status may be mistakenly recognized as having changed, when the VR4181 returns to Fullspeed mode.

★ **If a touch panel status change interrupt request occurs immediately after the VR4181 returns from the Hibernate mode, the PENCHGINTR bit may be set to 1 due to a miss-recognition such as above. Similarly, other bits of the PIUINTREG register may be set to 1 on returning from the Hibernate mode. Therefore, set each bit of the PIUINTREG register to 1 to clear an interrupt request immediately after a restore from the Hibernate mode.**

14.3.3 PIUSIVLREG (0x0B00 0126)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	SCAN INTVAL10	SCAN INTVAL9	SCAN INTVAL8
R/W	R	R	R	R	R	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

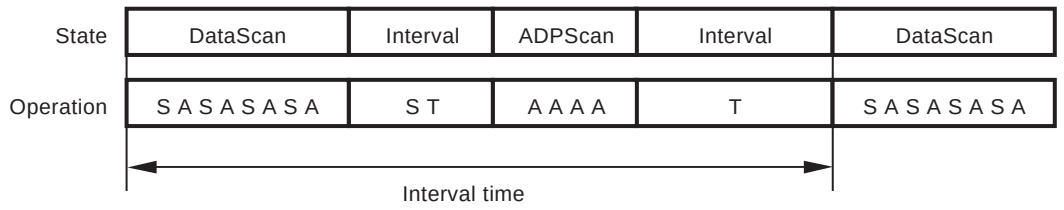
Bit	7	6	5	4	3	2	1	0
Name	SCAN INTVAL7	SCAN INTVAL6	SCAN INTVAL5	SCAN INTVAL4	SCAN INTVAL3	SCAN INTVAL2	SCAN INTVAL1	SCAN INTVAL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	1	0	1	0	0	1	1	1
Other resets	1	0	1	0	0	1	1	1

Bit	Name	Function
15 to 11	Reserved	0 is returned when read
10 to 0	SCANINTVAL(10:0)	Coordinate data scan sampling interval setting Interval = SCANINTVAL(10:0) x 30 μs

This register sets the scan interval (sampling period) for coordinate data sampling.

The sampling interval for one set of coordinate data is the value set via SCANINTVAL(10:0) multiplied by 30 μs. Accordingly, the logical range of sampling intervals that can be set in 30 μs units is from 0 μs to about 60 ms. Actually, if the sampling interval setting is shorter than the time required for obtaining a set of coordinate data or ADPScan data, a data lost interrupt request will occur. If data lost interrupt requests occur frequently, set a longer interval time.

Figure 14-5. Interval Times and States



★ **Remark** S: Voltage stabilization wait time (STABLE(5:0) in PIUSTBLREG)
A: A/D converter conversion time (about 10 μs)
T: Touch/release detection

14.3.4 PIUSTBLREG (0x0B00 0128)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	STABLE5	STABLE4	STABLE3	STABLE2	STABLE1	STABLE0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	1	1	1
Other resets	0	0	0	0	0	1	1	1

Bit	Name	Function
15 to 6	Reserved	0 is returned when read
5 to 0	STABLE(5:0)	Touch panel voltage stabilization wait time (DataScan, CMDScan state) A/D scan timeout time (ADPScan state) Touch detection start wait time (Disable, WaitPenTouch, Interval state) Wait time = STABLE(5:0) × 30 μs

★

The voltage stabilization wait time for the power applied to the touch panel can be set via the STABLE(5:0) bits in 30 μs units between 0 μs and 1,890 μs.

14.3.5 PIUCMDREG (0x0B00 012A)

(1/2)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	STABLEON	TPYEN1	TPYEN0	TPXEN1	TPXEN0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	TPYD1	TPYD0	TPXD1	TPXD0	ADCMD3	ADCMD2	ADCMD1	ADCMD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	1	1	1	1
Other resets	0	0	0	0	1	1	1	1

Bit	Name	Function
15 to 13	Reserved	0 is returned when read
12	STABLEON	Touch panel voltage stabilization wait time (STABLE(5:0) of PIUSTBLREG) enable during command scan 1 : Wait for panel voltage stabilization time 0 : Ignore panel voltage stabilization time (wait time = 0)
11, 10	TPYEN(1:0)	TPY port input/output switching during command scan 11 : TPY1 output, TPY0 output 10 : TPY1 output, TPY0 input 01 : TPY1 input, TPY0 output 00 : TPY1 input, TPY0 input
9, 8	TPXEN(1:0)	TPX port input/output switching during command scan 11 : TPX1 output, TPX0 output 10 : TPX1 output, TPX0 input 01 : TPX1 input, TPX0 output 00 : TPX1 input, TPX0 input
7, 6	TPYD(1:0)	TPY output level during command scan 11 : TPY1 = “H”, TPY0 = “H” 10 : TPY1 = “H”, TPY0 = “L” 01 : TPY1 = “L”, TPY0 = “H” 00 : TPY1 = “L”, TPY0 = “L”
5, 4	TPXD(1:0)	TPX output level during command scan 11 : TPX1 = “H”, TPX0 = “H” 10 : TPX1 = “H”, TPX0 = “L” 01 : TPX1 = “L”, TPX0 = “H” 00 : TPX1 = “L”, TPX0 = “L”

Remark L: low level, H: high level

(2/2)

Bit	Name	Function
3 to 0	ADCMD(3:0)	A/D converter input port selection for command scan 1111 : A/D converter standby mode request 1110 : RFU : 1000 : RFU 0111 : AUDIOIN port 0110 : ADIN2 port 0101 : ADIN1 port 0100 : ADIN0 port 0011 : TPY1 port 0010 : TPY0 port 0001 : TPX1 port 0000 : TPX0 port

This register switches input/output and sets output level for each port during a command scanning operation.

The setting of the TPYD bits are invalid when a port is set as input in the TPYEN bits.

The setting of the TPXD bits are invalid when a port is set as input in the TPXEN bits.

14.3.6 PIUASCNREG (0x0B00 0130)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	TPPSCAN	ADPS START
R/W	R	R	R	R	R	R	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 2	Reserved	0 is returned when read
1	TPPSCAN	Port selection for ADPScan 1 : Select TPX(1:0), TPY(1:0) (for touch panel) as A/D port 0 : Select ADIN(2:0) (general-purpose) as A/D port and AUDIOIN as audio input port
0	ADPSSTART	ADPScan start 1 : Start ADPScan 0 : Do not perform ADPScan

This register is used for ADPScan setting.

The ADPScan begins when the ADPSSTART bit is set. After the ADPScan is completed, the sequencer returns to ★ the state when ADPScan was started, and the ADPSSTART bit is cleared to 0 automatically.

If the ADPScan is not completed within the time period set via the STABLE bits of the PIUSTBLREG register, a data lost interrupt request occurs as a timeout interrupt.

Caution Manipulation of the TPPSCAN bit is valid only in the standby state. In the other states, the operation is not guaranteed.

Some bits in this register cannot be set in a specific state of scan sequencer. The combination of the setting of this register and the sequencer state is as follows.

★ **Table 14-4. PIUASCNREG Bit Manipulation and States**

PIUASCNREG bit manipulation		Scan sequencer's state			
		Disable	Standby	WaitPenTouch	DataScan
ADPSSTART[Note1]	0 → 1	×	ADPScan[Note2]	ADPScan[Note2]	×
	1 → 0	×	?	?	×
TPPSCAN	0 → 1	–	–	–	–
	1 → 0	–	–	–	–

PIUCNTREG bit manipulation		Scan sequencer's state		
		Interval	ADPScan	CMDScan
ADPSSTART[Note1]	0 → 1	ADPScan[Note2]	ADPScan[Note2]	×
	1 → 0	?	?	×
TPPSCAN	0 → 1	×	×	×
	1 → 0	?	×	×

Notes 1. Immediately after a transition to the ADPScan state, the bit is automatically cleared to 0.

2. After ADPScan is completed, the sequencer returns to the state in which the scan has started.

Remark – : The bit change is retained but there is no state transition.

× : Setting prohibited (operation not guaranteed)

? : Combination of state and bit status before setting does not exist.

14.3.7 PIUAMSKREG (0x0B00 0132)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	ADINM3	ADINM2	ADINM1	ADINM0	TPYM1	TPYM0	TPXM1	TPXM0
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
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 8	Reserved	0 is returned when read
7	ADINM3	Audio input port mask 1 : Mask 0 : Normal
6 to 4	ADINM(2:0)	General-purpose A/D port mask 1 : Mask 0 : Normal
3, 2	TPYM(1:0)	Touch panel A/D port TPY mask 1 : Mask 0 : Normal
1, 0	TPXM(1:0)	Touch panel A/D port TPX mask 1 : Mask 0 : Normal

★ This register is used to set masking each A/D port. Each bit corresponds to one port. If masked, A/D conversions are not performed for data of the corresponding port.

Settings in this register are valid only during the ADPScan state.

14.3.8 PIUCIVLREG (0x0B00 013E)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	CHECK INTVAL10	CHECK INTVAL9	CHECK INTVAL8
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	CHECK INTVAL7	CHECK INTVAL6	CHECK INTVAL5	CHECK INTVAL4	CHECK INTVAL3	CHECK INTVAL2	CHECK INTVAL1	CHECK INTVAL0
R/W	R	R	R	R	R	R	R	R
★ RTCRST	1	0	1	0	0	0	0	0
★ Other resets	1	0	1	0	0	0	0	0

Bit	Name	Function
15 to 11	Reserved	0 is returned when read
10 to 0	CHECKINTVAL(10:0)	Interval count value.

This register indicates the value of an internal register that counts down based on the PIUSIVLREG register setting.

14.3.9 PIUPBnmREG (0x0B00 02A0 to 0x0B00 02AE, 0x0B00 02BC to 0x0B00 02BE)

Remark n = 0, 1, m = 0 to 4

PIUPB00REG (0x0B00 02A0)	PIUPB10REG (0x0B00 02A8)
PIUPB01REG (0x0B00 02A2)	PIUPB11REG (0x0B00 02AA)
PIUPB02REG (0x0B00 02A4)	PIUPB12REG (0x0B00 02AC)
PIUPB03REG (0x0B00 02A6)	PIUPB13REG (0x0B00 02AE)
PIUPB04REG (0x0B00 02BC)	PIUPB14REG (0x0B00 02BE)

Bit	15	14	13	12	11	10	9	8
Name	VALID	Reserved	Reserved	Reserved	Reserved	Reserved	PADDATA9	PADDATA8
R/W	R/W	R	R	R	R	R	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	PADDATA7	PADDATA6	PADDATA5	PADDATA4	PADDATA3	PADDATA2	PADDATA1	PADDATA0
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
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15	VALID	Indicates validity of data in page buffer 1 : Valid 0 : Invalid
14 to 10	Reserved	0 is returned when read
9 to 0	PADDATA(9:0)	A/D converter's sampling data

These registers are used to store coordinate data or touch pressure data. There are four coordinate data buffers and one touch pressure data buffer, each of which holds two pages of coordinate data or pressure data, and the addresses (register addresses) where the coordinate data or the pressure data is stored are fixed. Read coordinate ★ data or pressure data from the corresponding register in a valid page.

The VALID bit, which indicates whether the data is valid, is automatically rendered invalid when the page buffer interrupt source (the PADPAGE0INTR or PADPAGE1INTR bit in the PIUINTREG register) is cleared.

Table 14-5 shows correspondences between the sampled data and the register in which the sampled data is stored.

Table 14-5. Detected Data and Page Buffers

Detected data	Page0 Buffer	Page1 Buffer
X–	PIUPB00REG	PIUPB10REG
X+	PIUPB01REG	PIUPB11REG
Y–	PIUPB02REG	PIUPB12REG
Y+	PIUPB03REG	PIUPB13REG
Z (Touch pressure)	PIUPB04REG	PIUPB14REG

14.3.10 PIUABnREG (0x0B00 02B0 to 0x0B00 02B6)

Remark n = 0 to 3
PIUAB0REG (0x0B00 02B0)
PIUAB1REG (0x0B00 02B2)
PIUAB2REG (0x0B00 02B4)
PIUAB3REG (0x0B00 02B6)

Bit	15	14	13	12	11	10	9	8
Name	VALID	Reserved	Reserved	Reserved	Reserved	Reserved	PADDATA9	PADDATA8
R/W	R/W	R	R	R	R	R	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	PADDATA7	PADDATA6	PADDATA5	PADDATA4	PADDATA3	PADDATA2	PADDATA1	PADDATA0
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
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15	VALID	Indicates validity of data in buffer 1 : Valid 0 : Invalid
14 to 10	Reserved	0 is returned when read
9 to 0	PADDATA(9:0)	A/D converter's sampling data

These registers are used to store sampling data of the general-purpose A/D port and audio input port or command scan data. There are four data buffers and the addresses (register address) where the data is stored are fixed.

The VALID bit, which indicates whether the data is valid, is automatically rendered invalid when the page buffer interrupt source (the PADADPINTR bit in the PIUINTREG register) is cleared.

Table 14-6 shows correspondences between the sampled data and the register in which the sampled data is stored.

Table 14-6. A/D Ports and Data Buffers

Register	During ADPScan		During CMDScan
	TPPSCAN = 0	TPPSCAN = 1	
PIUAB0REG	ADIN0	TPX0	CMDScan data
PIUAB1REG	ADIN1	TPX1	–
PIUAB2REG	ADIN2	TPY0	–
PIUAB3REG	AUDIOIN	TPY1	–

14.4 State Transition Flow

Be sure to initialize the PIU before scan sequencer operation. Initialization via a reset sets particular values for the sequence interval, etc., which should be re-set to appropriate values.

The following registers require initial settings.

SCANINTVAL(10:0) bit of PIUSIVLREG register

★ STABLE(5:0) bit of PIUSTBLREG register

Interrupt mask cancellation settings are required for registers other than the PIU registers.

Table 14-7. Mask Clear During Scan Sequencer Operation

Setting	Unit	Register	Bit	Value
Interrupt mask clear	ICU	MSYSINT1REG	MPIUINTR	1
	ICU	MPIUINTREG	bits 6 to 0	0x7F
★ Clock mask clear	MBA Host Bridge	CMUCLKMSK	MSKPIUPCLK	1

(1) Transition flow for voltage detection at A/D general-purpose ports and audio input port

Standby, WaitPenTouch, or Interval state

<1> PIUAMSKREG Mask setting for A/D ports and audio input port

<2> PIUASCNREG ADPSSTART = 1

↓

ADPScan state

<3> PIUASCNREG ADPSSTART = 0

↓

Standby, WaitPenTouch, or Interval state

(2) Transition flow for auto scan coordinate detection

Standby state

<1> PIUCNTREG PIUMODE(1:0) = 00

PADATSTART = 1

PADATSTOP = 1

<2> PIUCNTREG PIUSEQEN = 1

↓

WaitPenTouch state

(3) Transition flow for manual scan coordinate detection

Disable state
<1> PIUCNTREG PIUPWR = 1
↓
Standby state
<2> PIUCNTREG PIUMODE(1:0) = 00
PADSCANSTART = 1
<3> PIUCNTREG PIUSEQEN = 1
↓
DataScan state

(4) Transition flow when entering Suspend mode transition

Standby, WaitPenTouch, or Interval state
<1> PIUCNTREG PIUSEQEN = 0
↓
Standby state
<2> PIUCNTREG PIUPWR = 1
↓
Disable state

(5) Transition flow when returning from Suspend mode

Disable state
<1> PIUCNTREG PIUPWR = 1
↓
Standby state
<2> PIUCNTREG PIUMODE(1:0) = 00
PADATSTART = 1
PADATSTOP = 1
<3> PIUCNTREG PIUSEQEN = 1
↓
WaitPenTouch state
Touch detected
↓
DataScan state

(6) Transition flow for command scan

Disable state
<1> PIUCNTREG PIUPWR = 1
↓
Standby state
<2> PIUCNTREG PIUMODE(1:0) = 01
<3> PIUCNTREG Setting of touch panel pins, selection of input port
<4> PIUCNTREG PIUSEQEN = 1
↓
CMDScan state

14.5 Relationships among TPX, TPY, ADIN, and AUDIOIN Pins and States

★

State	PADSTATE(2:0)	TPX(1:0)	TPY(1:0)	AUDIOIN, ADIN(2:0)
PIU disable (pen status detection)	Disable [Note]	HH	D–	–––
Low-power standby	Standby	00	00	–––
Pen status detection	WaitPenTouch/Interval	HH	D–	–––
Voltage detection at general-purpose AD0 port	ADPScan	00	00	––I
Voltage detection at general-purpose AD1 port	ADPScan	00	00	–I–
Voltage detection at general-purpose AD2 port	ADPScan	00	00	I––
Voltage detection at audio input port	ADPScan	00	00	I––
Touch pressure detection (Z)	DataScan	HH	d–	–––
TPY1=L, TPY0=H, TPX0=samp (X–)	DataScan	–I	LH	–––
TPY1=H, TPY0=L, TPX0=samp (X+)	DataScan	–I	HL	–––
TPX1=L, TPX0=H, TPY0=samp (Y–)	DataScan	LH	–I	–––
TPX1=H, TPX0=L, TPY0=samp (Y+)	DataScan	HL	–I	–––

Note The states of pins are not guaranteed if the PADSTATE(2:0) immediately before the CPU's SUSPEND or HIBERNATE instruction execution is in a state other then the Disable state.

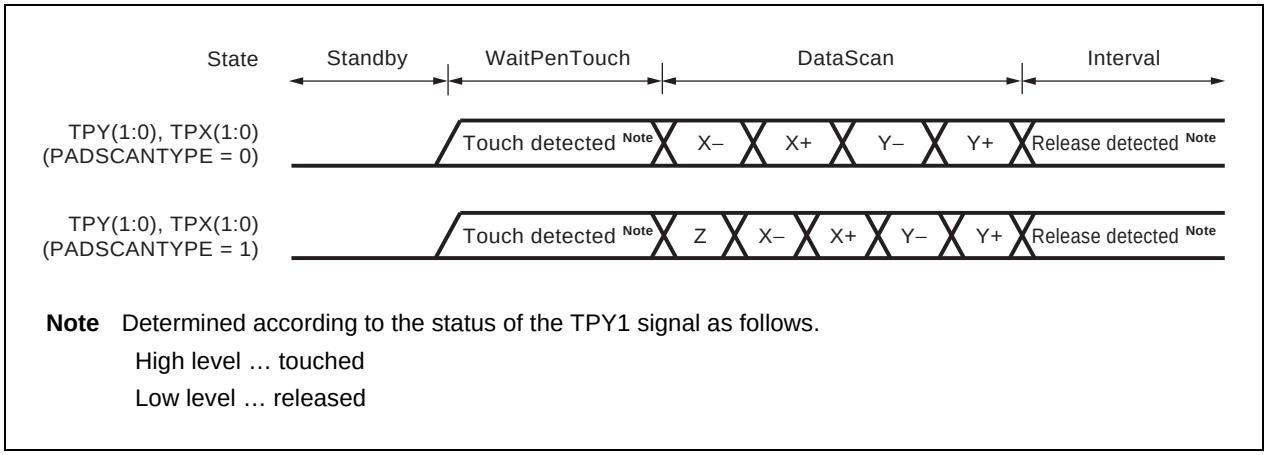
Remarks
- 0 : Low level input
- 1 : High level input
- L : Low level output
- H : High level output
- I : A/D converter input
- D : Touch interrupt request input (with a pull-down resistor)
- d : No touch interrupt request input (with a pull-down resistor)
- – : Don't care

14.6 Timing

14.6.1 Touch/release detection timing

Touch/release detection is not determined via the A/D converter but the voltage level of the TPY1 pin. The following figure shows a timing of touch/release detection and coordinate detection.

Figure 14-6. Touch/Release Detection Timing



★ **Note** Determined according to the status of the TPY1 signal as follows.

High level … touched

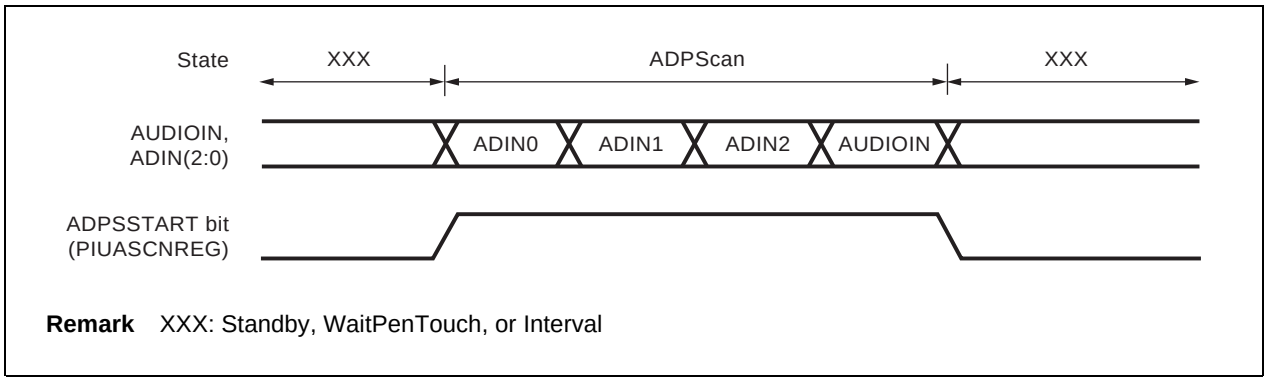
Low level … released

14.6.2 A/D port scan timing

During an A/D port scan, the four ports of A/D converter's input channel are sequentially scanned and the scanned data are stored in the data buffers dedicated to A/D port scanning.

The following figure shows an A/D port scan timing diagram.

Figure 14-7. A/D Port Scan Timing



★

Remark XXX: Standby, WaitPenTouch, or Interval

14.7 Data Loss Conditions

The PIU issues a data lost interrupt request when any of the following four conditions exist.

1. Data for one coordinate has not been obtained within the interval period
2. The A/D port scan has not been completed within the time set via PIUSTBLREG register
3. Transfer of the next coordinate data starts while valid data for both pages remains in the buffer
4. The next data transfer starts while there is valid data in the ADPScan buffer

Once a data lost interrupt request occurs, the sequencer is forcibly changed to the Standby state.
The cause and response to each condition are as follows.

(1) When data for one coordinate has not been obtained within the interval period

Cause

This condition occurs when the AIU has exclusive use of the A/D converter and the PIU is therefore unable to use the A/D converter.
If this data loss condition occurs frequently, implement a countermeasure that temporarily prohibits the AIU's use of the A/D converter.

Response

★ After clearing the data lost interrupt request by setting the PADDLOSTINTR bit to 1, set the PADATSTART bit or PADSCANSTART bit of the PIUCNTREG register to restart the coordinate detection operation. Once the data lost interrupt request is cleared, the page in which the loss occurred becomes invalid. If the valid data prior to the data loss is needed, be sure to save the data that is being stored in the page buffer before clearing the data lost interrupt request.

(2) When the A/D port scan has not been completed within the time set via PIUSTBLREG register

Cause

Same as cause of condition 1.

Response

★ After clearing the data lost interrupt request by setting the PADDLOSTINTR bit to 1, set the ADPSSTART bit of the PIUASCNREG register to restart the A/D port scan operation. Once the data lost interrupt request is cleared, the page in which the loss occurred becomes invalid. If the valid data prior to the data loss is needed, be sure to save the data that is being stored in the page buffer before clearing the data lost interrupt request.

(3) When transfer of the next coordinate data starts while valid data for both pages remains in the buffer

Cause

This condition is caused when the data buffer contains two pages of valid data (both the data buffer page 1 and data buffer page 0 interrupt requests have occurred) but the valid data has not been processed. If the A/D converter is used frequently, the time from when both pages become full until when the data loss occurs may be shorter than that of the normal operation.

Response

In this case, valid data contained in the pages when the data lost interrupt request occurs is never overwritten.

★ After two pages of valid data are processed, clear the three interrupt requests by writing 1 to the PADDLOSTINTR, PADPAGE1INTR, and PADPAGE0INTR bits in the PIUINTREG register.

After clearing these interrupt requests, set the PADATSTART or PADSCANSTART bit of the PIUCNTREG register to restart the coordinate detection operation.

(4) When the next data transfer starts while there is valid data in the ADPScan buffer

Cause

This condition is caused when valid data is not processed even while the ADPScan buffer holds valid data (A/D port scan interrupt request occurrence).

Response

In this case, valid data contained in the buffer when the data lost interrupt request occurs is never overwritten.

★ After valid data in the buffer is processed, clear the two interrupt requests by writing 1 to the PADDLOSTINTR and PADADPINTR bits in the PIUINTREG register.

After clearing these interrupt requests, set the ADPSSTART bit of the PIUASCNREG to restart the general-purpose A/D port scan operation.

CHAPTER 15 AUDIO INTERFACE UNIT (AIU)

★ 15.1 General

The AIU controls the analog output (speaker output) processing of the internal D/A converter and the analog input (microphone input) processing of the internal A/D converter. It is also used to make settings related to the A/D and D/A converters.

The main functions of the AIU are as follows:

- Holding the digital value converted by the internal A/D converter
- Holding the digital value to be converted by the internal D/A converter
- Separating data being converted by the A/D or D/A converter and transfer data by using double buffers
- Linking the update of the double buffers and the generation of DMA transfer requests with the data conversion rate

Caution No clocks are supplied to the AIU, A/D converter, and D/A converter in the initial state. When using the AIU, set the MSKAIUPCLK, MSKADUPCLK, and MSKADU18M bits of the CMUCLKMSK register in the MBA Host Bridge to 1 in advance so that clocks are supplied.

15.2 Register Set

The AIU registers are listed below.

Table 15-1. AIU Registers

Physical address	R/W	Register symbol	Function
0x0B00 0160	R/W	SDMADATREG	Speaker DMA data register
0x0B00 0162	R/W	MDMADATREG	Microphone DMA data register
0x0B00 0164	R/W	DAVREF_SETUP	D/A converter Vref setup register
0x0B00 0166	R/W	SODATREG	Speaker output data register
0x0B00 0168	R/W	SCNTREG	Speaker output control register
0x0B00 016E	R/W	SCNVC_END	Speaker sample rate control register
0x0B00 0170	R/W	MIDATREG	Microphone input data register
0x0B00 0172	R/W	MCNTREG	Microphone input control register
0x0B00 0178	R/W	DVALIDREG	Data valid indication register
0x0B00 017A	R/W	SEQREG	Sequencer enable register
0x0B00 017C	R/W	INTREG	Interrupt register
0x0B00 017E	R/W	MCNVC_END	Microphone sample rate control register

State of interrupt requests caused by AIU is indicated and can be set in the following registers, which are included in the ICU (refer to **CHAPTER 9 INTERRUPT CONTROL UNIT (ICU)** for details).

Table 15-2. AIU Interrupt Registers

Physical address	R/W	Register symbol	Function
0x0B00 0084	R	AIUINTREG	AIU interrupt indication register
0x0B00 0090	R/W	MAIUINTREG	AIU interrupt mask register

★ 15.2.1 SDMADATREG (0x0B00 0160)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	SDMA9	SDMA8
R/W	R	R	R	R	R	R	R/W	R/W
RTCRST	0	0	0	0	0	0	1	0
Other resets	0	0	0	0	0	0	1	0

Bit	7	6	5	4	3	2	1	0
Name	SDMA7	SDMA6	SDMA5	SDMA4	SDMA3	SDMA2	SDMA1	SDMA0
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
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 10	Reserved	0 is returned when read
9 to 0	SDMA(9:0)	Speaker output DMA data

This register is used to store 10-bit DMA data for speaker output. When SODATREG register is empty, the data is transferred to the SODATREG register.

Write is used for debugging and is enabled when the AIUSEN bit of the SEQREG register is set to 1. This register is initialized (0x0200) by resetting the AIUSEN bit of the SEQREG register to 0.

★ 15.2.2 MDMADATREG (0x0B00 0162)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	MDMA9	MDMA8
R/W	R	R	R	R	R	R	R/W	R/W
RTCRST	0	0	0	0	0	0	1	0
Other resets	0	0	0	0	0	0	1	0

Bit	7	6	5	4	3	2	1	0
Name	MDMA7	MDMA6	MDMA5	MDMA4	MDMA3	MDMA2	MDMA1	MDMA0
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
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 10	Reserved	0 is returned when read
9 to 0	MDMA(9:0)	Microphone input DMA data

This register is used prior to DMA transfer to store 10-bit data that has been converted by the A/D converter and stored in the MIDATREG register.

Write is used for debugging and is enabled when the AIUMEN bit of the SEQREG register is set to 1. This register is initialized (0x0200) by resetting the AIUMEN bit of the SEQREG register to 0. Therefore, if the AIUMEN bit is set to 0 during DMA transfer, invalid data may be transferred.

15.2.3 DAVREF_SETUP (0x0B00 0164)

Bit	15	14	13	12	11	10	9	8
Name	DAVREF15	DAVREF14	DAVREF13	DAVREF12	DAVREF11	DAVREF10	DAVREF9	DAVREF8
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
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	DAVREF7	DAVREF6	DAVREF5	DAVREF4	DAVREF3	DAVREF2	DAVREF1	DAVREF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RTCRST	0	1	1	1	1	1	0	1
Other resets	0	1	1	1	1	1	0	1

Bit	Name	Function
15 to 0	DAVREF(15:0)	D/A converter Vref setup time.

This register is used to select a Vref setup time for the D/A converter.

The following expression is used to calculate the value set to this register.

$$DAVREF(15:0) = 5\ \mu s \times PCLK\ frequency$$

For example, if the internal peripheral clock (PCLK) frequency is 25 MHz, the DAVREF(15:0) bits should be set to as follows;

$$DAVREF(15:0) = 5 \times 10^{-6} \times 25 \times 10^{6} = 0x007D$$

15.2.4 SODATREG (0x0B00 0166)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	SODAT9	SODAT8
R/W	R	R	R	R	R	R	R/W	R/W
RTCRST	0	0	0	0	0	0	1	0
Other resets	0	0	0	0	0	0	1	0

Bit	7	6	5	4	3	2	1	0
Name	SODAT7	SODAT6	SODAT5	SODAT4	SODAT3	SODAT2	SODAT1	SODAT0
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
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 10	Reserved	0 is returned when read
9 to 0	SODAT(9:0)	Speaker output data

This register is used to store 10-bit DMA data for speaker output. Data is received from the SDMADATREG register and is sent to the D/A converter.

Write is used for debugging and is enabled when the AIUSEN bit of the SEQREG register is set to 1. This register is initialized (0x0200) by resetting the AIUSEN bit of the SEQREG register to 0.

15.2.5 SCNTREG (0x0B00 0168)

Bit	15	14	13	12	11	10	9	8
Name	DAENAIU	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R/W	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	SSTATE	Reserved	SSTOPEN	Reserved
R/W	R	R	R	R	R	R	R/W	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15	DAENAIU	Enables D/A converter operation (Vref connection). 1 : ON 0 : OFF
14 to 4	Reserved	0 is returned when read
3	SSTATE	Indicates speaker operation state. 1 : Operating 0 : Stopped
2	Reserved	0 is returned when read
1	SSTOPEN	Speaker output DMA transfer page boundary interrupt 1 : Stop DMA request at 1-page boundary 0 : Stop DMA request at 2-page boundary
0	Reserved	0 is returned when read

This register is used to control the AIU's speaker block.

★ The DAENAIU bit controls the connection of VDD_AD and Vref input to ladder type resistors in the D/A converter. Setting this bit to 0 (OFF) allows low power consumption when not using the D/A converter. When using the D/A converter, this bit must be set following the sequence described in **15.3 Operation Sequence**.

The content of the SSTATE bit is valid only when the AIUSEN bit of the SEQREG register is set to 1.

15.2.6 SCNVC_END (0x0B00 016E)

Bit	15	14	13	12	11	10	9	8
Name	SCNVC15	SCNVC14	SCNVC13	SCNVC12	SCNVC11	SCNVC10	SCNVC9	SCNVC8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
★ RTCRST	0	0	0	0	1	0	0	0
★ Other resets	0	0	0	0	1	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	SCNVC7	SCNVC6	SCNVC5	SCNVC4	SCNVC3	SCNVC2	SCNVC1	SCNVC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
★ RTCRST	1	1	0	1	1	1	0	0
★ Other resets	1	1	0	1	1	1	0	0

Bit	Name	Function
15 to 0	SCNVC(15:0)	Speaker sample rate control

This register is used to select a conversion rate for the D/A converter.

The following expression is used to calculate the value set to this register.

SCNVC(15:0) = PCLK frequency/sample rate

For example, if the desired conversion rate is 8 ksps and internal peripheral clock (PCLK) frequency is 25 MHz, SCNVC(15:0) bits should be set to as follows;

$SCNVC(15:0) = 25 \times 10^6/8 \times 10^3 = 0x0C35$

★ **Caution Set this register to a value that determines the conversion rate as 50 ksps or less.**

15.2.7 MIDATREG (0x0B00 0170)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	MIDAT9	MIDAT8
R/W	R	R	R	R	R	R	R/W	R/W
RTCRST	0	0	0	0	0	0	1	0
Other resets	0	0	0	0	0	0	1	0

Bit	7	6	5	4	3	2	1	0
Name	MIDAT7	MIDAT6	MIDAT5	MIDAT4	MIDAT3	MIDAT2	MIDAT1	MIDAT0
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
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 10	Reserved	0 is returned when read
9 to 0	MIDAT(9:0)	Microphone input data

This register is used to store 10-bit speaker input data that has been converted by the A/D converter. Data is sent to the MDMADATREG register and is received from the A/D converter.

Write is used for debugging and is enabled when the AIUMEN bit of the SEQREG register is set to 1. This register is initialized (0x0200) by resetting the AIUMEN bit of the SEQREG register to 0.

15.2.8 MCNTREG (0x0B00 0172)

Bit	15	14	13	12	11	10	9	8
Name	ADENAIU	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R/W	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	MSTATE	Reserved	MSTOPEN	ADREQAIU
R/W	R	R	R	R	R	R	R/W	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15	ADENAIU	Enables A/D converter operation (Vref connection). 1 : ON 0 : OFF
14 to 4	Reserved	0 is returned when read
3	MSTATE	Indicates microphone operation state 1 : Operating 0 : Stopped
2	Reserved	0 is returned when read
1	MSTOPEN	Microphone input DMA transfer page boundary interrupt 1 : Stop DMA request at 1-page boundary 0 : Stop DMA request at 2-page boundary
0	ADREQAIU	Request for use of A/D converter 1 : Requesting 0 : No request

This register is used to control the AIU's microphone block.

★ The ADENAIU bit controls the connection of VDD_AD and Vref input to ladder type resistors in the A/D converter. Setting this bit to 0 (OFF) allows low power consumption when not using the A/D converter. When using the A/D converter, this bit must be set following the sequence described in **15.3 Operation Sequence**.

The content of the MSTATE bit is valid only when the AIUMEN bit of the SEQREG register is set to 1.

The AIU has priority when a conflict occurs with the PIU in relation to A/D conversion requests.

15.2.9 DVALIDREG (0x0B00 0178)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	SODATV	SDMAV	MIDATV	MDMAV
R/W	R	R	R	R	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 4	Reserved	0 is returned when read
3	SODATV	This indicates whether valid data has been stored in SODATREG. 1 : Valid data exists 0 : No valid data
2	SDMAV	This indicates whether valid data has been stored in SDMADATREG. 1 : Valid data exists 0 : No valid data
1	MIDATV	This indicates whether valid data has been stored in MIDATREG. 1 : Valid data exists 0 : No valid data
0	MDMAV	This indicates whether valid data has been stored in MDMADATREG. 1 : Valid data exists 0 : No valid data

This register indicates whether valid data has been stored in the SODATREG, SDMADATREG, MIDATREG, or MDMADATREG register.

If data has been written directly to the SODATREG, SDMADATREG, MIDATREG, or MDMADATREG register via software, the bits in this register are not set so that 1 must be written via software.

Write is used for debugging and is enabled when the AIUSEN or AIUMEN bit of the SEQREG register is set to 1.

If the AIUSEN bit = 0 or AIUMEN bit = 0 in the SEQREG register, then the SODATV bit = SDMAV bit = 0 or MIDATV bit = MDMAV bit = 0.

15.2.10 SEQREG (0x0B00 017A)

Bit	15	14	13	12	11	10	9	8
Name	AIURST	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R/W	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	AIUMEN	Reserved	Reserved	Reserved	AIUSEN
R/W	R	R	R	R/W	R	R	R	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15	AIURST	AIU reset via software 1 : Reset 0 : Normal
14 to 5	Reserved	0 is returned when read
4	AIUMEN	Microphone block operation and DMA enable 1 : Enable 0 : Disable
3 to 1	Reserved	0 is returned when read
0	AIUSEN	Speaker block operation and DMA enable 1 : Enable 0 : Disable

This register is used to enable/disable the AIU's operation.

15.2.11 INTREG (0x0B00 017C)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	MIDLEINTR	MSTINTR
R/W	R	R	R	R	R	R	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	SIDLEINTR	Reserved
R/W	R	R	R	R	R	R	R/W	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 10	Reserved	0 is returned when read
9	MIDLEINTR	Microphone idle interrupt request (receive data loss). Cleared to 0 when 1 is written. 1 : Occurred 0 : Normal
8	MSTINTR	Microphone receive completion interrupt request. Cleared to 0 when 1 is written. 1 : Occurred 0 : Normal
7 to 2	Reserved	0 is returned when read
1	SIDLEINTR	Speaker idle interrupt request (mute). Cleared to 0 when 1 is written. 1 : Occurred 0 : Normal
0	Reserved	0 is returned when read

This register indicates occurrence of various interrupt request of the AIU.

When data is received from the A/D converter, the MIDLEINTR bit is set if valid data still exists in the MIDATREG register (MIDATV bit = 1). In this case, the MIDATREG register is overwritten.

The MSTINTR bit is set when data is received in the MDMADATREG register.

When data is passed to the D/A converter, the SIDLEINTR bit is set if there is no valid data in the SODATREG register (SODATV bit = 0). However, this interrupt request is valid only after AIUSEN bit = 1 in the SODATREG register, after which SODATV bit = 1 in the DVALIDREG register.

15.2.12 MCNVC_END (0x0B00 017E)

Bit	15	14	13	12	11	10	9	8
Name	MCNVC15	MCNVC14	MCNVC13	MCNVC12	MCNVC11	MCNVC10	MCNVC9	MCNVC8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
★ RTCRST	0	0	0	0	1	0	0	0
★ Other resets	0	0	0	0	1	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	MCNVC7	MCNVC6	MCNVC5	MCNVC4	MCNVC3	MCNVC2	MCNVC1	MCNVC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
★ RTCRST	1	1	0	1	1	1	0	0
★ Other resets	1	1	0	1	1	1	0	0

Bit	Name	Function
15 to 0	MCNVC(15:0)	Microphone sample rate control.

This register is used to select a conversion rate for the A/D converter.

The following expression is used to calculate the value set to this register.

MCNVC(15:0) = PCLK frequency/sample rate

For example, if the desired conversion rate is 11.025 ksps and internal peripheral clock (PCLK) frequency is 25 MHz, the MCNVC(15:0) bits should be set to as follows;

MCNVC(15:0) = $25 \times 10^6 / 11.025 \times 10^3$ = 0x08DC

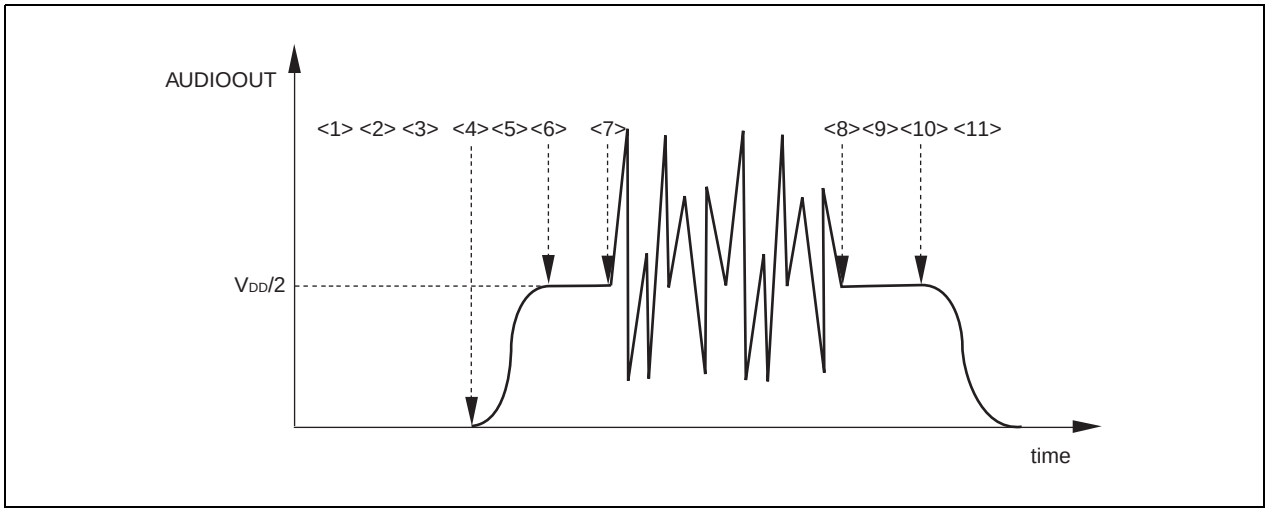
★ **Caution Set this register to a value that determines the conversion rate as 50 ksps or less.**

15.3 Operation Sequence

15.3.1 Output (speaker)

1. Set conversion rate (0x0B00 016E: SCNVC(15:0) = any value)
2. ★ Set D/A converter Vref setup time (0x0B00 0164: any value to be DVAREF(15:0)/PCLK frequency = 5 μs)
3. ★ Enable DMA after setting DMA address in DCU
4. Set D/A converter's Vref to ON (0x0B00 0168: DAENAIU = 1)
5. Wait for Vref resistor stabilization time (about 5 μs) (use the RTC counter)
   Even if speaker power is set to ON and speaker operation is enabled (AIUSEN = 1) without waiting for Vref resistor stabilization time, speaker output starts after the period calculated with the formula below.
   5 + 1/conversion rate (44.1, 22.05, 11.025, or 8) (μs)
   In this case, however, a noise may occur when speaker power is set to ON.
6. Set speaker power ON via GPIO.
7. Enable speaker operation (0x0B00 017A: AIUSEN = 1)
   DMA request
   Receive acknowledge and DMA data from DMA
   0x0B00 0178: SDMAV = SODATV = 1
   Output 10-bit data (0x0B00 0166: SODAT(9:0)) to D/A converter
   SODATV = 0, SDMAV = 1
   Send SDMADATREG data to SODATREG.
   SODATV = 1, SDMAV = 0
   Output DMA request and store the data after the next into SDMADATREG.
   SODATV = 1, SDMAV = 1
   Update data at each conversion timing interval (becomes SIDLEINTR = 1 when DMA delays and SODATV = 0 during conversion timing interval, and (mute) interrupt request occurs)
   DMA page boundary interrupt request occurs at page boundary
   Clear the page interrupt request to continue output.
8. Disable speaker operation (0x0B00 017A: AIUSEN = 0)
9. Set speaker power OFF via GPIO.
10. Set D/A converter's Vref to OFF (0x0B00 0168: DAENAIU = 0)
11. Disable DMA in DCU

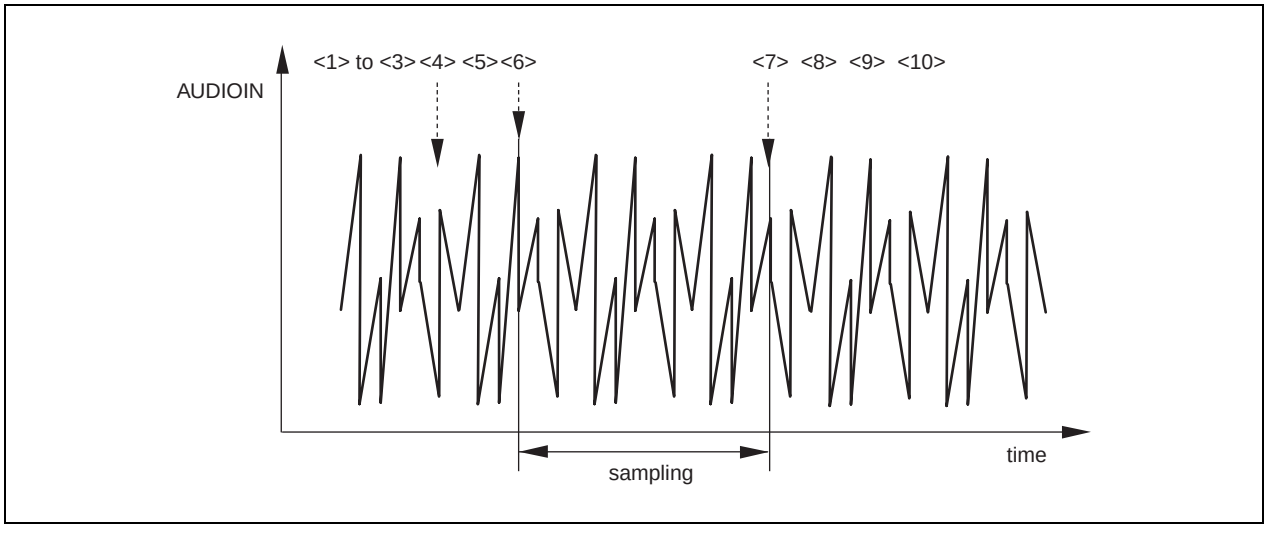
Figure 15-1. Speaker Output and AUDIOOUT Pin



15.3.2 Input (microphone)

1. Set conversion rate (0x0B00 017E: MCNVC(15:0) = any value)
★ 2. Set D/A converter Vref setup time (0x0B00 0164: any value to be DVAREF(15:0)/PCLK frequency = 5 μs)
★ 3. Enable DMA after setting DMA address in DCU
4. Set A/D converter's Vref to ON (0x0B00 0172: ADENAIU = 1)
   Microphone power can be set ON and microphone operation can be enabled (AIUMEN = 1) without waiting for Vref resistor stabilization time (about 5 μs). However, in such a case, sampling starts after the period calculated with the formula below.
   5 + 1/conversion rate (44.1, 22.05, 11.025, or 8) (μs)
5. Set microphone power ON via GPIO.
6. Enable microphone operation (0x0B00 017A: AIUMEN = 1)
   Output A/D request to A/D converter
   Acknowledge and 10-bit conversion data are returned from A/D converter.
   Store data in MIDATREG.
   0x0B00 0178: MDMAV = 0, MIDATV = 1
   Transfer data from MIDATREG to MDMADATREG.
   MDMAV = 1, MIDATV = 0
★ MSTINTR = 1 and an interrupt request (receive complete) occurs.
   Issue DMA request and store MDMADATREG data to memory.
   MDMAV = 0, MIDATV = 0
   Issue an A/D request once per conversion timing interval and receive 10-bit data (becomes MIDLEINTR = 1 when DMA delays and MIDATV = 1 during conversion timing interval, and (data loss) interrupt request occurs)
   DMA page boundary interrupt request occurs at page boundary
   Clear the page interrupt request to continue output.
7. Disable microphone operation (0x0B00 017A: AIUMEN = 0)
8. Set microphone power OFF via GPIO.
9. Set A/D converter's Vref to OFF (0x0B00 0172: ADENAIU = 0)
10. Disable DMA in DCU

Figure 15-2. AUDIOIN Pin and Microphone Operation



CHAPTER 16 KEYBOARD INTERFACE UNIT (KIU)

16.1 General

The Keyboard Interface Unit (KIU) provides the interface between the VR4181 and an external matrix type keyboard. This unit supports key matrix of 8 x 8.

The interface to the keyboard consists of SCANOUT (3-state output) and SCANIN (input) lines. The SCANOUT lines are used to search the matrix for pressed keys. The SCANIN lines are used to sense key press events and are read after each SCANOUT line being at low level to locate the pressed key.

SCANOUT and SCANIN lines are allocated by programming CompactFlash pins to support this function during the power-on. If those pins are set as for keyboard interface, CompactFlash interface cannot be used.

16.2 Functional Description

When the keyboard is idle, the SCANOUT lines are all driven to 0 volts and the SCANIN lines are pulled to V_{DD} by external 4.7 kΩ resistors. When any key in the matrix is pressed, at least one SCANIN input is driven as low and signals a key press event to the KIU.

Once the key press event has been detected, the KIU may be programmed to generate a key down interrupt request, and to begin scanning the keyboard automatically or to wait until software enables the scan operation.

Keyboard scanning is performed by sequentially driving one SCANOUT line as low while the others remain high impedance, and reading the state of the SCANIN lines and storing into keyboard data registers inside the KIU. Once the last SCANOUT line has been driven as low and the SCANIN lines read the KIU may generate a Keyboard Data Ready interrupt request to inform system software that one keyboard scan operation has been completed.

The KIU repeats this scan process until no further keys have been detected or until software disables the scan operation. At this point the KIU enters to the keyboard idle state or key press wait state.

The following table illustrates the relationship between these bits:

Table 16-1. Settings of Keyboard Scan Mode

ASTOP	ASTART	MSTART	MSTOP	Operation
0	0	0	0	Scanning disabled
X	X	X	1	Scanning stopped
0	X	1	0	Manual Scan mode. Scan operation starts as soon as a setting of the MSTART bit is detected by the scan sequencer and stops when the MSTOP bit is set to 1.
1	X	1	0	Manual Scan with Auto Stop mode. Scan operation starts as soon as a setting of the MSTART bit is detected by the scan sequencer and stops when no valid keyboard data has been read for STPREP(5:0) times of consecutive scan cycles.
0	1	0	0	Auto Scan with Manual Stop mode. Scan operation starts as soon as a key press is detected by the scan sequencer and stops when the MSTOP bit is set to 1.
1	1	0	0	Auto Scan mode. Scan operation starts as soon as a key press is detected by the scan sequencer and stops when no valid keyboard data has been read for STPREP(5:0) times of consecutive scan cycles.

16.2.1 Automatic keyboard scan mode (Auto Scan mode)

Automatic Scan mode is enabled through the ASTART and ASTOP bits of the KIUSCANREP register. When the ASTART bit is set to 1, keyboard scanning starts automatically following a key down interrupt request. When the ASTOP bit is set to 1, keyboard scanning stops automatically after no valid keyboard data (i.e. all SCANIN lines are high level) has been read for the number of scan cycles specified by the STPREP(5:0) bits of the KIUSCANREP register.

16.2.2 Manual keyboard scan mode (Manual Scan mode)

Manual Scan mode is enabled through the MSTART and MSTOP bits of the KIUSCANREP register. Software initiates a keyboard scan operation by setting the MSTART bit to 1 and terminates keyboard scanning by setting the MSTOP bit to 1. When software sets the MSTOP bit to 1, the KIU will complete the current scan operation before disabling the scan logic.

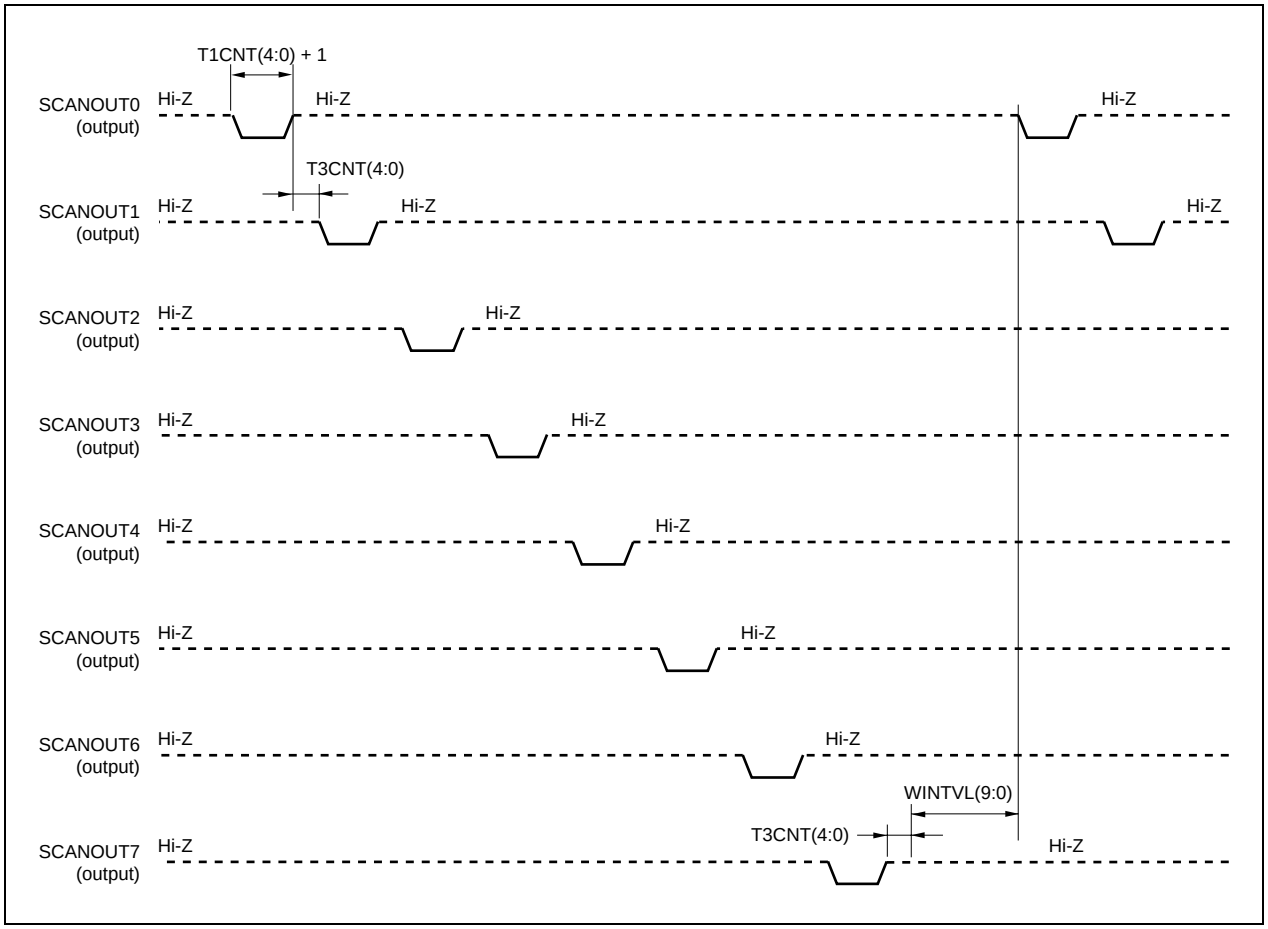
16.2.3 Key press detection

All SCANIN lines are sampled by the KIU on the rising edge of the 32.768 kHz clock. When any SCANIN line is sampled as low during a period of time from a rising edge to a falling edge of the 32.768 kHz clock, a key down interrupt request is generated. If the ASTART bit of the KIUSCANREP register is set to 1 at this time, the KIU begins scanning the keyboard.

16.2.4 Scan operation

Scan operations are controlled by the T1CNT(4:0) and T3CNT(4:0) bits of the KIUWKS register and the WINTVL(9:0) bits of the KIUWKI register. The following diagram illustrates the relationship of these register bits to the scan operation:

Figure 16-1. SCANOUT Signal Output Timing



The T1CNT(4:0) bits specify the keyboard settling time and is expressed in 32.768 kHz clock cycles. Following the low level of one of the SCANOUT(7:0) pins, the KIU will wait for the time set in the T1CNT(4:0) bits before reading returned data to the SCANIN(7:0) pins. Actually the SCANOUT pins will be driven as low for (T1CNT(4:0) + 1) 32.768 kHz clock cycles.

The T3CNT(4:0) bits specify the delay from driving one SCANOUT pin as high impedance to driving the next SCANOUT pin as low and is also expressed in 32.768 kHz clock cycles. When the SCANOUTn pin is driven as high impedance, the KIU will wait for the time set in the T3CNT(4:0) bits before driving the SCANOUTn+1 pin as low to allow the external pull-up resistors to return the SCANINn pin as high (n = 0 to 6).

The WINTVL(9:0) bits specify the interval between one scan and another in 32.768 kHz clock cycles. After the last SCANOUT pin has been driven as high impedance and a time set in the T3CNT(4:0) bits has elapsed, the KIU will wait for the time set in the WINTVL(9:0) bits before driving SCANOUT0 as low to start the next scan sequence.

16.2.5 Reading scanned data

Scanned data is read from the SCANIN(7:0) pins. When a SCANOUT pin has been driven as low and the keyboard settling time specified by the T1CNT(4:0) bits has been elapsed, the KIU latches scanned data from the SCANIN pins and stores into one of the internal key data registers.

16.2.6 Interrupts and status reporting

The KIU provides scan status indication that may be polled by the CPU core and may also generate interrupt requests to request keyboard servicing. Scan status indication is provided through the SSTAT(1:0) bits of the KIUSCANS register. These bits are decoded as follows:

SSTAT1	SSTAT0	KIU scan sequencer status
0	0	Stopped
0	1	Waiting for key press
1	0	Scanning (T1CNT or T3CNT)
1	1	During scan interval (WINTVL)

The KIU generates 3 types of maskable interrupt requests. KIU interrupt pending status is reported through the KDATLOST, KDATRDY, and KEYDOWN bits of the KIUINT register. All interrupt requests generated by the KIU should be considered asynchronous and must be externally qualified with TClock.

The key data lost interrupt request (KDATLOST bit) signals that a data from the SCANIN line written to the key data register corresponding to the SCANOUT0 pin before the previous data value is read by the CPU core. This interrupt source can be masked through the MSKKDATLOST bit of the MKIUINTREG register.

The key data ready interrupt request (KDATRDY bit) signals one complete scan operation has been completed. This interrupt request is generated during a write of a data from the SCANIN line to the key data register corresponding to the last SCANOUT pin. This interrupt request source can be masked through the MSKKDATRDY bit of the MKIUINTREG register.

The key down interrupt request (KEYDOWN bit) signals a key press event has been detected. This interrupt request is generated in synchronization with the rising edge of the 32.768 kHz clock when the keyboard interface is idle and any SCANIN pin is sampled as low during a period of time from a rising edge to a falling edge of the 32.768 kHz clock. This interrupt request source can be masked through the MSKKDOWNINT bit of the MKIUINTREG register.

The MSKKDATLOST, MSKKDATRDY, and MSKKDOWNINT bits only prevent interrupt requests from being generated on the kiuintr signal (internal). These mask bits do not disable interrupt request event detection nor do they disable interrupt status reporting in the KIUINT register.

16.3 Register Set

The KIU registers are listed below.

Table 16-2. KIU Registers

Physical address	R/W	Register symbol	Function
0x0B00 0180	R	KIUDAT0	Scan line 0 keyboard data register
0x0B00 0182	R	KIUDAT1	Scan line 1 keyboard data register
0x0B00 0184	R	KIUDAT2	Scan line 2 keyboard data register
0x0B00 0186	R	KIUDAT3	Scan line 3 keyboard data register
0x0B00 0188	R	KIUDAT4	Scan line 4 keyboard data register
0x0B00 018A	R	KIUDAT5	Scan line 5 keyboard data register
0x0B00 018C	R	KIUDAT6	Scan line 6 keyboard data register
0x0B00 018E	R	KIUDAT7	Scan line 7 keyboard data register
0x0B00 0190	R/W	KIUSCANREP	Scan control register
0x0B00 0192	R	KIUSCANS	Scan status register
0x0B00 0194	R/W	KIUWKS	Key scan stable time register
0x0B00 0196	R/W	KIUWKI	Key scan interval time register
0x0B00 0198	R/W	KIUINT	Interrupt register

State of interrupt requests caused by KIU is indicated and can be set in the following registers, which are included in the ICU (refer to **CHAPTER 9 INTERRUPT CONTROL UNIT (ICU)** for details).

Table 16-3. KIU Interrupt Registers

Physical address	R/W	Register symbol	Function
★ 0x0B00 0086	R/W	KIUINTREG	KIU interrupt indication register
0x0B00 0092	R/W	MKIUINTREG	KIU interrupt mask register

16.3.1 KIUDATn (0x0B00 0180 to 0x0B00 018E)

Remark n = 0 to 7

KIUDAT0 (0x0B00 0180) KIUDAT4 (0x0B00 0188)
KIUDAT1 (0x0B00 0182) KIUDAT5 (0x0B00 018A)
KIUDAT2 (0x0B00 0184) KIUDAT6 (0x0B00 018C)
KIUDAT3 (0x0B00 0186) KIUDAT7 (0x0B00 018E)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	RETDAT7	RETDAT6	RETDAT5	RETDAT4	RETDAT3	RETDAT2	RETDAT1	RETDAT0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 8	Reserved	0 is returned when read
7 to 0	RETDAT(7:0)	Scan data 1 : Key is released 0 : Key is pressed

These registers reflect the state of the returned signals for the selected SCANOUT pins. Each register corresponds to one SCANOUT pin as follows:

SCANOUT pin	KIUDAT register
SCANOUT7	KIUDAT7
SCANOUT6	KIUDAT6
SCANOUT5	KIUDAT5
SCANOUT4	KIUDAT4
SCANOUT3	KIUDAT3
SCANOUT2	KIUDAT2
SCANOUT1	KIUDAT1
SCANOUT0	KIUDAT0

16.3.2 KIUSCANREP (0x0B00 0190)

Bit	15	14	13	12	11	10	9	8
Name	KEYEN	Reserved	Reserved	Reserved	Reserved	Reserved	STPREP5	STPREP4
R/W	R/W	R	R	R	R	R	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	STPREP3	STPREP2	STPREP1	STPREP0	MSTOP	MSTART	ASTOP	ASTART
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
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15	KEYEN	KIU enable. This bit enables a KIU operation. When this bit is set to 0, the scan sequencer and all interrupt requests are disabled. 1 : Enable 0 : Disable
14 to 10	Reserved	0 is returned when read
9 to 4	STPREP(5:0)	Scan sequencer stop count. These bits select the number of scan operation performed after all keys have been released (0xFF is loaded to KIUDAT registers). 111111 : 63 times : 000001 : 1 time 000000 : 64 times
3	MSTOP	Scan stop (manual mode). This bit is sampled at the end of each scan operation and causes the scan sequencer to stop scanning when set to 1. 1 : Stop 0 : Operate
2	MSTART	Manual scan start (manual mode). When this bit is set to 1, the scan sequencer starts scanning the keyboard. 1 : Start 0 : Stop
1	ASTOP	Auto scan stop (auto mode). When this bit is set to 1, the scan sequencer stops scanning automatically when all keys have been released for the number of scan operation specified by the STPREP(5:0) bits. 1 : Auto stop 0 : Manual stop
0	ASTART	Auto Scan mode enable. When this bit is set to 1, the scan sequencer starts scanning automatically following a key press event. 1 : Enable 0 : Disable

16.3.3 KIUSCANS (0x0B00 0192)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	SSTAT1	SSTAT0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 2	Reserved	0 is returned when read
1, 0	SSTAT(1:0)	Scan sequencer status 11 : During scan interval (WINTVL) 10 : Scanning (T1CNT or T3CNT) 01 : Waiting for key press 00 : Stopped

★

16.3.4 KIUWKS (0x0B00 0194)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	T3CNT4	T3CNT3	T3CNT2	T3CNT1	T3CNT0	Reserved	Reserved
R/W	R	R/W	R/W	R/W	R/W	R/W	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
★ Name	Reserved	Reserved	Reserved	T1CNT4	T1CNT3	T1CNT2	T1CNT1	T1CNT0
★ R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15	Reserved	0 is returned when read
★ 14 to 10	T3CNT(4:0)	Scan idle time. These bit determine the wait time the scan sequencer waits following a deassertion of one SCANOUT pin before an assertion of the next SCANOUT pin. 11111 : 960 μs : (T3CNT(4:0) + 1) x 30 μs 00001 : 60 μs 00000 : Setting prohibited
9 to 5	Reserved	0 is returned when read
★ 4 to 0	T1CNT(4:0)	Scan data stabilization time. These bits determine the time the scan sequencer waits following an assertion of a SCANOUT pin before return data is read. 11111 : 960 μs : (T1CNT(4:0) + 1) x 30 μs 00001 : 60 μs 00000 : Setting prohibited

16.3.5 KIUWKI (0x0B00 0196)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	WINTVL9	WINTVL8
R/W	R	R	R	R	R	R	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	WINTVL7	WINTVL6	WINTVL5	WINTVL4	WINTVL3	WINTVL2	WINTVL1	WINTVL0
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
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 10	Reserved	0 is returned when read
9 to 0	WINTVL(9:0)	Scan interval time. These bits determine the time the scan sequencer waits following completion of one scan operation before starting the next scan operation. 1111111111 : 30690 μs : WINTVL(9:0) x 30 μs 0000000001 : 30 μs 0000000000 : No wait

16.3.6 KIUINT (0x0B00 0198)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	KDATLOST	KDATRDY	KEYDOWN
R/W	R	R	R	R	R	R/W	R/W	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 3	Reserved	0 is returned when read
2	KDATLOST	Key data lost interrupt request. This interrupt request occurs if the KIUDAT0 register is updated with the next key data prior to being read by the CPU core. 1 : Occurred 0 : Not occurred This bit is cleared by writing 1.
1	KDATRDY	Key data ready interrupt request. This interrupt request occurs when a set of scanning is completed and all the KIUDAT registers are updated. 1 : Occurred 0 : Not occurred This bit is cleared by writing 1.
0	KEYDOWN	Key down interrupt request. This interrupt request occurs when the KIU sequencer is idle and any of the SCANIN inputs has been sampled as low level. 1 : Occurred 0 : Not occurred This bit is cleared by writing 1.

CHAPTER 17 COMPACTFLASH CONTROLLER (ECU)

17.1 General

The VR4181 provides an ExCA-compatible controller (ECU) supporting a single CompactFlash slot. The interface for this controller is shared with that of the keyboard interface unit. To use this interface for CompactFlash control, the ★ KEYSEL bit of the KEYEN register in the GIU must be clear to 0. Also, to use CF_BUSY# signal as an activation factor, the CompactFlash interface must be enabled during Hibernate mode by writing 1 to the CFHIBEN bit of the KEYEN register.

17.2 Register Set Summary

This section provides details of the ECU registers. Two of the ECU registers are located in the I/O addressing space. These registers, as well as the Interrupt and Configuration registers, are shown in the following table.

Table 17-1. ECU Control Registers

Physical address	R/W	Register symbol	Function
0x0B00 08E0	R/W	ECUINDX	Index register (I/O space)
0x0B00 08E1	R/W	ECUDATA	Data register (I/O space)
0x0B00 08F8	R	INTSTATREG	Interrupt status register
0x0B00 08FA	R/W	INTMSKREG	Interrupt mask register
0x0B00 08FE	R/W	CFG_REG_1	Configuration register 1

The remaining ECU registers listed below are all 8-bit width and accessed through the Index register and the Data register.

Table 17-2. ECU Registers (1/2)

Index	R/W	Register symbol	Function
0x0000	R	ID_REV_REG	Identification and revision register
0x0001	R	IF_STAT_REG	Interface status register
0x0002	R/W	PWRRSETDRV	Power and RESETDRV control register
0x0003	R/W	ITGENCTREG	Interrupt and general control register
0x0004	R/W	CDSTCHGREG	Card status change register
0x0005	R/W	CRDSTATREG	Card status change interrupt configuration register
0x0006	R/W	ADWINENREG	Address window enable register
0x0007	R/W	IOCTRL_REG	I/O control register
0x0008	R/W	IOADSLB0REG	I/O start address 0 low byte register
0x0009	R/W	IOADSHB0REG	I/O start address 0 high byte register
0x000A	R/W	IOSLB0REG	I/O stop address 0 low byte register
0x000B	R/W	IOSHB0REG	I/O stop address 0 high byte register
0x000C	R/W	IOADSLB1REG	I/O start address 1 low byte register
0x000D	R/W	IOADSHB1REG	I/O start address 1 high byte register
0x000E	R/W	IOSLB1REG	I/O stop address 1 low byte register
0x000F	R/W	IOSHB1REG	I/O stop address 1 high byte register
0x0010	R/W	SYSMEMSL0REG	System memory 0 mapping start address low byte register
0x0011	R/W	MEMWID0_REG	System memory 0 mapping start address high byte register
0x0012	R/W	SYSMEMEL0REG	System memory 0 mapping stop address low byte register
0x0013	R/W	MEMSEL0_REG	System memory 0 mapping stop address high byte register
0x0014	R/W	MEMOFFL0REG	Card memory 0 offset address low byte register
0x0015	R/W	MEMOFFH0REG	Card memory 0 offset address high byte register
0x0016	R/W	DTGENCLREG	Card detect and general control register
0x0018	R/W	SYSMEMSL1REG	System memory 1 mapping start address low byte register
0x0019	R/W	MEMWID1_REG	System memory 1 mapping start address high byte register
0x001A	R/W	SYSMEMEL1REG	System memory 1 mapping stop address low byte register
0x001B	R/W	MEMSEL1_REG	System memory 1 mapping stop address high byte register
0x001C	R/W	MEMOFFL1REG	Card memory 1 offset address low byte register
0x001D	R/W	MEMOFFH1REG	Card memory 1 offset address high byte register
0x001E	R/W	GLOCTRLREG	Global control register
0x001F	R	VOLTSENREG	Card voltage sense register

Table 17-2. ECU Registers (2/2)

Index	R/W	Register symbol	Function
0x0020	R/W	SYSMEMSL2REG	System memory 2 mapping start address low byte register
0x0021	R/W	MEMWID2_REG	System memory 2 mapping start address high byte register
0x0022	R/W	SYSMEMEL2REG	System memory 2 mapping stop address low byte register
0x0023	R/W	MEMSEL2_REG	System memory 2 mapping stop address high byte register
0x0024	R/W	MEMOFFL2REG	Card memory 2 offset address low byte register
0x0025	R/W	MEMOFFH2REG	Card memory 2 offset address high byte register
0x0028	R/W	SYSMEMSL3REG	System memory 3 mapping start address low byte register
0x0029	R/W	MEMWID3_REG	System memory 3 mapping start address high byte register
0x002A	R/W	SYSMEMEL3REG	System memory 3 mapping stop address low byte register
0x002B	R/W	MEMSEL3_REG	System memory 3 mapping stop address high byte register
0x002C	R/W	MEMOFFL3REG	Card memory 3 offset address low byte register
0x002D	R/W	MEMOFFH3REG	Card memory 3 offset address high byte register
0x002F	R/W	VOLTSELREG	Card voltage select register
0x0030	R/W	SYSMEMSL4REG	System memory 4 mapping start address low byte register
0x0031	R/W	MEMWID4_REG	System memory 4 mapping start address high byte register
0x0032	R/W	SYSMEMEL4REG	System memory 4 mapping stop address low byte register
0x0033	R/W	MEMSEL4_REG	System memory 4 mapping stop address high byte register
0x0034	R/W	MEMOFFL4REG	Card memory 4 offset address low byte register
0x0035	R/W	MEMOFFH4REG	Card memory 4 offset address high byte register

17.3 ECU Control Registers

17.3.1 INTSTATREG (0x0B00 08F8)

Bit	15	14	13	12	11	10	9	8
Name	IRQ15	IRQ14	Reserved	IRQ12	IRQ11	IRQ10	IRQ9	Reserved
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	IRQ7	Reserved	IRQ5	IRQ4	IRQ3	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15, 14	IRQ(15:14)	Status of interrupt request 15 and 14 (internal) 0 : Invalid 1 : Valid
13	Reserved	0 is returned when read
12 to 9	IRQ(12:9)	Status of interrupt request 12, 11, 10 and 9 (internal) 0 : Invalid 1 : Valid
8	Reserved	0 is returned when read
7	IRQ7	Status of interrupt request 7 (internal) 0 : Invalid 1 : Valid
6	Reserved	0 is returned when read
5 to 3	IRQ(5:3)	Status of interrupt request 5, 4 and 3 (internal) 0 : Invalid 1 : Valid
2 to 0	Reserved	0 is returned when read

Remark A single bit corresponds to each interrupt request.

17.3.2 INTMSKREG (0x0B00 08FA)

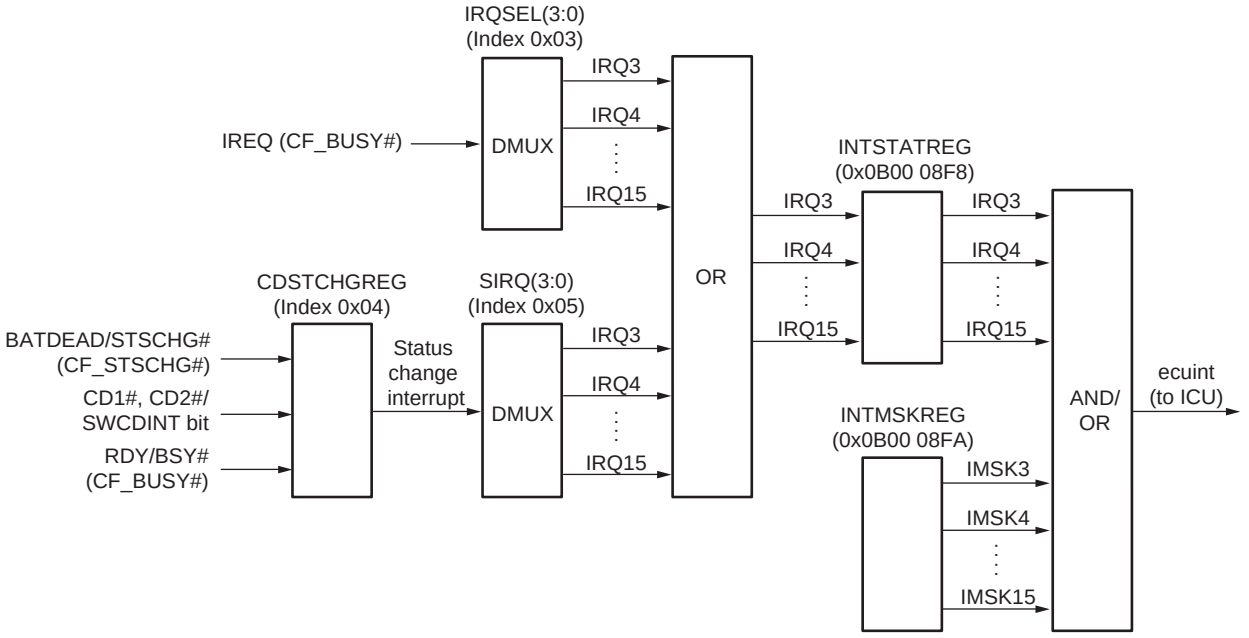
Bit	15	14	13	12	11	10	9	8
Name	IMSK015	IMSK014	Reserved	IMSK012	IMSK011	IMSK010	IMSK09	Reserved
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	IMSK07	Reserved	IMSK05	IMSK04	IMSK03	Reserved	Reserved	Reserved
R/W	R/W	R	R/W	R/W	R/W	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15, 14	IMSK0(15:14)	Mask for interrupt request 15 and 14 (internal) 0 : Unmask 1 : Mask
13	Reserved	0 is returned when read
12 to 9	IMSK0(12:9)	Mask for interrupt request 12, 11, 10, and 9 (internal) 0 : Unmask 1 : Mask
8	Reserved	0 is returned when read
7	IMSK07	Mask for interrupt request 7 (internal) 0 : Unmask 1 : Mask
6	Reserved	0 is returned when read
5 to 3	IMSK0(5:3)	Mask for interrupt request 5, 4 and 3 (internal) 0 : Unmask 1 : Mask
2 to 0	Reserved	0 is returned when read

Remark A single bit corresponds to each interrupt request.

Figure 17-1. CompactFlash Interrupt Logic



Remark All IRQ signals are ORed together to generate ecuint after ANDed with IMSK0n bits in the INTMSKREG register (n = 0 to 15).

17.3.3 CFG_REG_1 (0x0B00 08FE)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	WSE
R/W	R	R	R	R	R	R	R	R/W
Reset	0	0	0	0	0	0	0	1

Bit	Name	Function
15 to 1	Reserved	0 is returned when read
0	WSE	Internal ISA cycle 1 wait state insertion enable. This bit controls wait insertion when accessing the ECU registers. Write 1 to this bit when write. 1 : Enable

17.4 ECU Registers

17.4.1 ID_REV_REG (Index: 0x00)

Bit	7	6	5	4	3	2	1	0
Name	IFTYP1	IFTYP0	Reserved	Reserved	REV3	REV2	REV1	REV0
R/W	R	R	R	R	R	R	R	R
Reset	1	0	0	0	0	0	1	1

Bit	Name	Function
7, 6	IFTYP(1:0)	PCSC interface type These bits indicate 10 to reflect that both memory and I/O cards are supported.
5, 4	Reserved	0 is returned when read
3 to 0	REV(3:0)	Revision level. 0011 is always displayed.

17.4.2 IF_STAT_REG (Index: 0x01)

Bit	7	6	5	4	3	2	1	0
Name	Reserved	PWRON	RDY/BSY	WP	CD2	CD1	Reserved	BVD1
R/W	R	R	R	R	R	R	R	R
Reset	1	0	Undefined	Undefined	1	1	0	Undefined

Bit	Name	Function
7	Reserved	1 is returned when read
6	PWRON	CompactFlash card power status 0 : Off 1 : On
5	RDY/BSY	CompactFlash card ready/busy status. This bit indicates the current status of RDY/BSY# (CF_BUSY#) signal from a CompactFlash card. 0 : Busy 1 : Ready
4	WP	Memory write protect switch status. This bit indicates the current status of WP (CF_IOIS16#) signal from a CompactFlash card. 0 : Off 1 : On
3, 2	CD(2:1)	Complement of the values of CD1# and CD2# [Note] 11: Active (low level) 00 : Inactive (high level) Values other than above are not displayed.
1	Reserved	0 is returned when read
0	BVD1	This bit indicates the current status of STSCHG# (CF_STSCHG#) signal from a CompactFlash card.

Note The card detect pins, CD1# and CD2#, alternate with GPIO pins. When the GPIO pins are not programmed as card detect input, the CD(2:1) bits of this register always return 11 (active). In this way, the CompactFlash interface can be used without the card detect pins. When the GPIO pins are programmed as card detect, the CD(2:1) bits are reflected in actual status of the CD1# and CD2# pins.

17.4.3 PWRRSETDRV (Index: 0x02)

Bit	7	6	5	4	3	2	1	0
Name	OE	Reserved	Reserved	PWREN	Reserved	Reserved	Reserved	Reserved
R/W	R/W	R	R	R/W	R	R	R/W	R
Reset	0	0	1	0	0	0	0	0

Bit	Name	Function
7	OE	Output enable. If this bit is cleared to 0, the CompactFlash interface outputs from the VR4181 are driven to high impedance state and the CF_DEN# and CF_AEN# outputs are driven as high. **Caution This bit should not be set until this register has been written to set the CompactFlash card power enable.**
6	Reserved	0 is returned when read
5	Reserved	1 is returned when read
4	PWREN	Card power enable 0 : Disabled (Vcc is 0 V) 1 : Enabled. Voltage selected in the VOLTSELREG register (0x2F) is applied. The power to the socked is turned on when a card is inserted and off when removed. ★ **Caution The VR4181 supports cards with the card voltage of 3.3 V only. Do not set this bit to 1 unless the contents of the VOTSELREG register are 0x01.**
3, 2	Reserved	0 is returned when read
1	Reserved	Write 0 when write. 0 is returned when read.
0	Reserved	0 is returned when read

17.4.4 ITGENCTREG (Index: 0x03)

Bit	7	6	5	4	3	2	1	0
Name	RI_EN	CRDRST	CRDTYP	Reserved	IRQSEL3	IRQSEL2	IRQSEL1	IRQSEL0
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	1	1

Bit	Name	Function
7	RI_EN	Ring indicate enable. This bit is used to switch the function of the STSCHG#/RI# signal from the I/O card. The ring indicator function cannot be used in the VR4181 so that 0 must be written to this bit. 0 : Used as the STSCHG#. The current status of the signal is read from the IF_STAT_REG register if this signal is configures as a source for the card status change interrupt. 1 : Used as the RI# For memory PC Cards, this bit has no function.
6	CRDRST	Card reset. This bit is for a software reset to the PC Card to which the status of the CF_RESET signal is set. 0 : Active The CF_RESET signal will be active until this bit is set to 1. 1 : Inactive
5	CRDTYP	Card type 0 : Memory card 1 : I/O card
4	Reserved	0 is returned when read
3 to 0	IRQSEL(3:0)	Interrupt request steering for the I/O card IREQ (CF_BUSY#) signal 0000 : IRQ is not used 0001 : RFU 0010 : RFU 0011 : IRQ3 is used 0100 : IRQ4 is used 0101 : IRQ5 is used 0110 : RFU 0111 : IRQ7 is used 1000 : RFU 1001 : IRQ9 is used 1010 : IRQ10 is used 1011 : IRQ11 is used 1100 : IRQ12 is used 1101 : RFU 1110 : IRQ14 is used 1111 : IRQ15 is used

17.4.5 CDSTCHGREG (Index: 0x04)

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	CD_CHG	RDY_CHG	Reserved	BAT_DEAD
R/W	R	R	R	R	R/W	R/W	R	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7 to 4	Reserved	0 is returned when read
3	CD_CHG	Card detect (CD1# and CD2# signals) status change 0 : Not changed 1 : Changed
2	RDY_CHG	Ready (CF_BUSY# signal) change 0 : No change or I/O card installed 1 : A low-to-high change has been detected indicating that the memory card is ready to accept a new data transfer
1	Reserved	0 is returned when read
0	BAT_DEAD	Battery not usable or status change detection (CF_STSCHG# signal status) 0 : For memory cards, battery is good. For I/O cards, the RI_EN bit of the ITGENCTREG register is set to 1, or the CF_STSCHG# signal is at high level. 1 : For memory cards, a battery dead condition has been detected. For I/O cards, the RI_EN bit of the ITGENCTREG register is cleared to 0 and the CF_STSCHG# signal is at low level. When this bit is set to 1, the system software then has to read the status change register in the I/O card to determine the cause of STSCHG. **Caution CompactFlash cards do not support the BVD (battery status detection) signal so that the BVD2/SPKR signal of the ECU is internally fixed to low level.**

This register indicates the source of the card status change interrupt request. Each source can be enabled to generate this interrupt request by setting the corresponding bit in the CRDSTATREG register. The bits in this register become 0 if their corresponding enable bits are cleared to 0.

If the EXWRBK bit is set to 1 in the GLOCTRLREG register, sources for the card status change interrupt request ★ is acknowledged when 1 is set to the CD_CHG bit in the CDSTCHGREG register though it has been already set to 1. Once acknowledged, the CD_CHG bit is cleared to 0. The interrupt request signal caused by the card status change, if any of the IRQ lines is enabled, remains active until all the bits in this register become 0.

If the EXWRBK bit is not set to 1, the card status change interrupt request, when any of the IRQ lines are enabled, remains active until this register is read. In this mode, reading this register resets all status bits to 0, which has been set to 1.

17.4.6 CRDSTATREG (Index: 0x05)

Bit	7	6	5	4	3	2	1	0
Name	SIRQS3	SIRQS2	SIRQS1	SIRQS0	CD_EN	RDY_EN	Reserved	BDEAD_EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7 to 4	SIRQS(3:0)	Interrupt request steering for the I/O card STSCHG# (CF_BUSY#) signal. 0000 : IRQ is not used 0001 : RFU 0010 : RFU 0011 : IRQ3 is used 0100 : IRQ4 is used 0101 : IRQ5 is used 0110 : RFU 0111 : IRQ7 is used 1000 : RFU 1001 : IRQ9 is used 1010 : IRQ10 is used 1011 : IRQ11 is used 1100 : IRQ12 is used 1101 : RFU 1110 : IRQ14 is used 1111 : IRQ15 is used
3	CD_EN	Card detect enable. Enables a card status change interrupt request when a change has been detected on the CD1# or CD2# signals. 0 : Disable 1 : Enable
2	RDY_EN	Ready enable. Enables a card status change interrupt request when a transition has been detected on the CF_BUSY# signal. 0 : Disable 1 : Enable
1	Reserved	0 is returned when read
0	BDEAD_EN	Battery not usable or status change interrupt request enable. Enables a card status change interrupt request when a change has been detected on the CF_STSCHG# signal (battery unusable status for memory cards or status change detection for I/O cards). 0 : Disable 1 : Enable For I/O cards, the RI_EN bit of the ITGENCTREG register must be cleared to 0 in advance when using the interrupt request via the CF_STSCHG# signal.

17.4.7 ADWINENREG (Index: 0x06)

Bit	7	6	5	4	3	2	1	0
Name	IOWEN1	IOWEN0	Reserved	MWEN4	MWEN3	MWEN2	MWEN1	MWEN0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7, 6	IOWEN(1:0)	I/O window enables. Generates the card enable signals to the card when an I/O access occurs within the corresponding I/O address window. 0 : Does not generate 1 : Generates I/O addresses are output from the system bus directly to the card. **Caution The start and stop address register pairs must be set to values for the window to be used before setting these bits to 1.**
5	Reserved	0 is returned when read
4 to 0	MWEN(4:0)	Memory window enables. Generates the card enable signals to the card when a memory access occurs within the corresponding memory address window. 0 : Does not generate 1 : Generates When the system address is within the window, the computed address is output to the card. **Caution The start, stop, and offset address register pairs must be set to values for the window to be used before setting these bits to 1.**

Remark A single bit corresponds to each window.

17.4.8 IOCTRL_REG (Index: 0x07)

Bit	7	6	5	4	3	2	1	0
Name	IO1WT	W1_IOWS	IO1_CS16 MD	IO1DSZ	IO0WT	W0_IOWS	IO0_CS16 MD	IO0DSZ
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7	IO1WT	I/O window 1 wait addition in 16-bit accesses 0 : No additional wait state 1 : Adds 1 wait state
6	W1_IOWS	I/O window 1 wait addition in 8-bit accesses 0 : No additional wait state 1 : Adds 1 wait state
5	IO1_CS16MD	I/O window 1 IOCS16 source 0 : Value of the IO1DSZ bit 1 : CF_IOIS16# signal from the card
4	IO1DSZ	I/O window 1 access data size 0 : 8 bits 1 : 16 bits ★ This bit has no function when the IO1_CS16MD bit is set to 1.
3	IO0WT	I/O window 0 wait addition in 16-bit accesses ★ 0 : Without additional wait state 1 : Adds 1 wait state
2	W0_IOWS	I/O window 0 wait addition in 8-bit accesses 0 : No additional wait state 1 : Adds 1 wait state
1	IO0_CS16MD	I/O window 0 IOCS16 source 0 : Value of the IO0DSZ bit 1 : CF_IOIS16# signal from the card
0	IO0DSZ	I/O window 0 access data size 0 : 8 bits 1 : 16 bits ★ This bit has no function when the IO0_CS16MD bit is set to 1.

17.4.9 IOADSLBnREG (Index: 0x08, 0x0C)

Remark n = 0, 1
IOADSLB0REG (0x08): for Window 0
IOADSLB1REG (0x0C): for Window 1

Bit	7	6	5	4	3	2	1	0
Name	STARTA7	STARTA6	STARTA5	STARTA4	STARTA3	STARTA2	STARTA1	STARTA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7 to 0	STARTA(7:0)	I/O window start address bits 7 to 0

Low-order address bits used to determine the start address of an I/O address window. Minimum 1 byte can be specified for the I/O address window.

17.4.10 IOADSHBnREG (Index: 0x09, 0x0D)

Remark n = 0, 1
IOADSHB0REG (0x09): for Window 0
IOADSHB1REG (0x0D): for Window 1

Bit	7	6	5	4	3	2	1	0
Name	STARTA15	STARTA14	STARTA13	STARTA12	STARTA11	STARTA10	STARTA9	STARTA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7 to 0	STARTA(15:8)	I/O window start address bits 15 to 8

High-order address bits used to determine the start address of an I/O address window.

Remark Address bits 25 to 16 of an I/O window address are fixed to 0. Therefore, an I/O window is always mapped to the address space between 0x1400 0000 and 0x1400 FFFF, which is the first 64 KB of the ISA-IO space.

17.4.11 IOSLBnREG (Index: 0x0A, 0x0E)

Remark n = 0, 1
IOSLB0REG (0x0A): for Window 0
IOSLB1REG (0x0E): for Window 1

Bit	7	6	5	4	3	2	1	0
Name	STOPA7	STOPA6	STOPA5	STOPA4	STOPA3	STOPA2	STOPA1	STOPA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7 to 0	STOPA(7:0)	I/O window stop address bits 7 to 0

Low-order address bits used to determine the stop address of an I/O address window.

17.4.12 IOSHBnREG (Index: 0x0B, 0x0F)

Remark n = 0, 1
IOSHB0REG (0x0B): for Window 0
IOSHB1REG (0x0F): for Window 1

Bit	7	6	5	4	3	2	1	0
Name	STOPA15	STOPA14	STOPA13	STOPA12	STOPA11	STOPA10	STOPA9	STOPA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7 to 0	STOPA(15:8)	I/O window stop address bits 15 to 8

High-order address bits used to determine the stop address of an I/O address window.

Remark Address bits 25 to 16 of an I/O window address are fixed to 0. Therefore, an I/O window is always mapped to the address space between 0x1400 0000 and 0x1400 FFFF, which is the first 64 KB of the ISA-IO space.

17.4.13 SYSMEMSLnREG (Index: 0x10, 0x18, 0x20, 0x28, 0x30)

Remark n = 0 to 4

SYSMEMSL0REG (0x10): for Window 0
SYSMEMSL1REG (0x18): for Window 1
SYSMEMSL2REG (0x20): for Window 2
SYSMEMSL3REG (0x28): for Window 3
SYSMEMSL4REG (0x30): for Window 4

Bit	7	6	5	4	3	2	1	0
Name	MWSTART A19	MWSTART A18	MWSTART A17	MWSTART A16	MWSTART A15	MWSTART A14	MWSTART A13	MWSTART A12
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7 to 0	MWSTARTA(19:12)	Memory window start address bits 19 to 12

Low-order address bits used to determine the start address of a memory address window. Minimum 4 KB can be specified for memory address window.

17.4.14 MEMWIDn_REG (Index: 0x11, 0x19, 0x21, 0x29, 0x31)

Remark n = 0 to 4

MEMWID0_REG (0x11): for Window 0
MEMWID1_REG (0x19): for Window 1
MEMWID2_REG (0x21): for Window 2
MEMWID3_REG (0x29): for Window 3
MEMWID4_REG (0x31): for Window 4

Bit	7	6	5	4	3	2	1	0
Name	DWIDTH	ZWSEN	MWSTART A25	MWSTART A24	MWSTART A23	MWSTART A22	MWSTART A21	MWSTART A20
R/W	R/W	R/W	R/W	R\W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7	DWIDTH	Memory window data width 0 : 8 bits 1 : 16 bits
★ 6	ZWSEN	Zero wait state enable. This bit is used to set whether the zero wait state is requested to the ISA Bridge in memory accesses. 0 : Does not request 1 : Requests
5 to 0	MWSTARTA(25:20)	Memory window start address bits 25 to 20

This register is used to set the memory window data width, zero wait state enable, and high-order address bits used to determine the start address of a memory address window.

17.4.15 SYSMEMELnREG (Index: 0x12, 0x1A, 0x22, 0x2A, 0x32)

Remark n = 0 to 4

SYSMEMEL0REG (0x12): for Window 0
SYSMEMEL1REG (0x1A): for Window 1
SYSMEMEL2REG (0x22): for Window 2
SYSMEMEL3REG (0x2A): for Window 3
SYSMEMEL4REG (0x32): for Window 4

Bit	7	6	5	4	3	2	1	0
Name	MWSTOPA 19	MWSTOPA 18	MWSTOPA 17	MWSTOPA 16	MWSTOPA 15	MWSTOPA 14	MWSTOPA 13	MWSTOPA 12
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7 to 0	MWSTOPA(19:12)	Memory window stop address bits 19 to 12

Low-order address bits used to determine the stop address of a memory address window.

17.4.16 MEMSELn_REG (Index: 0x13, 0x1B, 0x23, 0x2B, 0x33)

Remark n = 0 to 4

MEMSEL0_REG (0x13): for Window 0
MEMSEL1_REG (0x1B): for Window 1
MEMSEL2_REG (0x23): for Window 2
MEMSEL3_REG (0x2B): for Window 3
MEMSEL4_REG (0x33): for Window 4

Bit	7	6	5	4	3	2	1	0
Name	M16W1	M16W0	MWSTOPA 25	MWSTOPA 24	MWSTOPA 23	MWSTOPA 22	MWSTOPA 21	MWSTOPA 20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7, 6	M16W(1:0)	Memory window wait state select for 16-bit accesses 00 : No additional wait state 01 : 2 additional wait states 10 : 3 additional wait states 11 : 4 additional wait states
5 to 0	MWSTOPA(25:20)	Memory window stop address bits 25 to 20

★ The ECU automatically inserts wait states when memory windows are accessed in 16-bit width.

17.4.17 MEMOFFLnREG (Index: 0x14, 0x1C, 0x24, 0x2C, 0x34)

Remark n = 0 to 4

MEMOFFL0REG (0x14): for Window 0
MEMOFFL1REG (0x1C): for Window 1
MEMOFFL2REG (0x24): for Window 2
MEMOFFL3REG (0x2C): for Window 3
MEMOFFL4REG (0x34): for Window 4

Bit	7	6	5	4	3	2	1	0
Name	OFFSETA 19	OFFSETA 18	OFFSETA 17	OFFSETA 16	OFFSETA 15	OFFSETA 14	OFFSETA 13	OFFSETA 12
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7 to 0	OFFSETA(19:12)	Card memory offset address bits 19 to 12

★ This register is defined to maintain compatibility with the ExCA. Settings in this register have no meaning in the VR4181.

17.4.18 MEMOFFHnREG (Index: 0x15, 0x1D, 0x25, 0x2D, 0x35)

Remark n = 0 to 4

MEMOFFH0REG (0x15): for Window 0
MEMOFFH1REG (0x1D): for Window 1
MEMOFFH2REG (0x25): for Window 2
MEMOFFH3REG (0x2D): for Window 3
MEMOFFH4REG (0x35): for Window 4

Bit	7	6	5	4	3	2	1	0
Name	WP	REG	OFFSETA 25	OFFSETA 24	OFFSETA 23	OFFSETA 22	OFFSETA 21	OFFSETA 20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7	WP	Write protect to the card through a memory window 0 : Write operation allowed 1 : Write operation prohibited
6	REG	REG# (CF_REG#) signal active of the CompactFlash. This bit is used to set which memory is to be used on accesses to the CompactFlash card. 0 : Common memory 1 : Attribute memory
5 to 0	OFFSETA(25:20)	Card memory offset address bits 25 to 20. ★ **Remark** This is defined to maintain compatibility with the ExCA. Settings in these bits have no meaning in the VR4181.

17.4.19 DTGENCLREG (Index: 0x16)

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	SWCDINT	CDRSMEN	Reserved	Reserved	CFGRSTEN	DLY16INH
R/W	R	R	W	R/W	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7, 6	Reserved	0 is returned when read
5	SWCDINT	Software card detect interrupt request 1 : Generates interrupt request This bit is valid when the CD_EN bit is set to 1 in the CRDSTATREG register. 0 is returned when read.
4	CDRSMEN	Card detect resume enable 1 : Enables notification of change on CD1# and CD2# inputs This bit is valid when the CD_EN bit is set to 1 in the CRDSTATREG register. 0 is returned when read.
3, 2	Reserved	0 is returned when read
1	CFGRSTEN	Configuration reset enable 1 : Enables initializing registers on high level of both CD1# and CD2# inputs The registers involved are all I/O registers, all memory registers, ITGENCTREG register, and ADWINENREG register.
0	DLY16INH	16-bit memory delay prohibit. This bit is used to set whether the falling edge of the WE# and OE# (CF_WE# and CF_OE#) signals of the CompactFlash is delayed in synchronization with SYSCLK when a memory window is set to be 16 bit in the DWIDTH bit of the MEMWIDn_REG register. 0 : Delayed 1 : Not delayed

★

The functionality and acknowledgment of this software interrupt request operate in the same way as those of the hardware-generated interrupt requests. The functionality and acknowledgement of the hardware card detect or card status change interrupt request are not affected by the setting of the SWCDINT bit. If card detect or card status change is signaled through the CD1# and CD2# inputs, a hardware card detect or card status change interrupt request is generated.

When the CDRSMEN bit is set to 1, the RIO# signal (internal) goes from high level to low and the CD_CHG bit in the CDSTCHGREG register is set to 1. The RIO# signal remains low until either a read or a write of 1 to the CD_CHG bit (acknowledge cycle), which causes the CD_CHG bit to be reset to 0 and the RIO# signal to go from low level to high. If the card status change is routed to any of the IRQ signals, the setting of this bit to 1 prevents the IRQ signal from going active as a result of a hardware card detect status change. Once the software detects a card detect status change interrupt request from the RIO# signal by reading the CDSTCHGREG register, it must issue a software card detect change interrupt request so that the card detect change condition generates an active interrupt request on the IRQ signal.

17.4.20 GLOCTRLREG (Index: 0x1E)

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	EXWRBK	Reserved	Reserved
R/W	R	R	R	R	R	R/W	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
7 to 3	Reserved	0 is returned when read
2	EXWRBK	Card status change interrupt request acknowledgement. 0 : Reading of the CDSTCHGREG register Each bit of the register is cleared after read. 1 : Writing 1 to the CDSTCHGREG register Each bit of the register is cleared after write of 1.
1, 0	Reserved	0 is returned when read

★

17.4.21 VOLTSENREG (Index: 0x1F)

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	VS2	VS1
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	1	0

Bit	Name	Function
7 to 2	Reserved	0 is returned when read
1, 0	VS(2:1)	Voltage sense status These bits are read-only and hardwired to 10 binary since the V_R4181 has no voltage sense pins.

17.4.22 VOLTSELREG (Index: 0x2F)

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	VCCEN1	VCCEN0
R/W	R	R	R	R	R	R	R/W	R/W
Reset	0	0	0	0	0	0	1	0

Bit	Name	Function
7 to 2	Reserved	0 is returned when read
1, 0	VCCEN(1:0)	Card connection status 01 : 3.3 V card connected 10 : No card connected **Caution Do not perform any write to this bit.**

★ If the PWREN bit of the PWRRSETDRV register is set to 1 when the VCCEN(1:0) bits are 01, the CF_VCCEN# signal becomes active.

★ **Remark** The VR4181 supports cards with the card voltage of 3.3 V only.

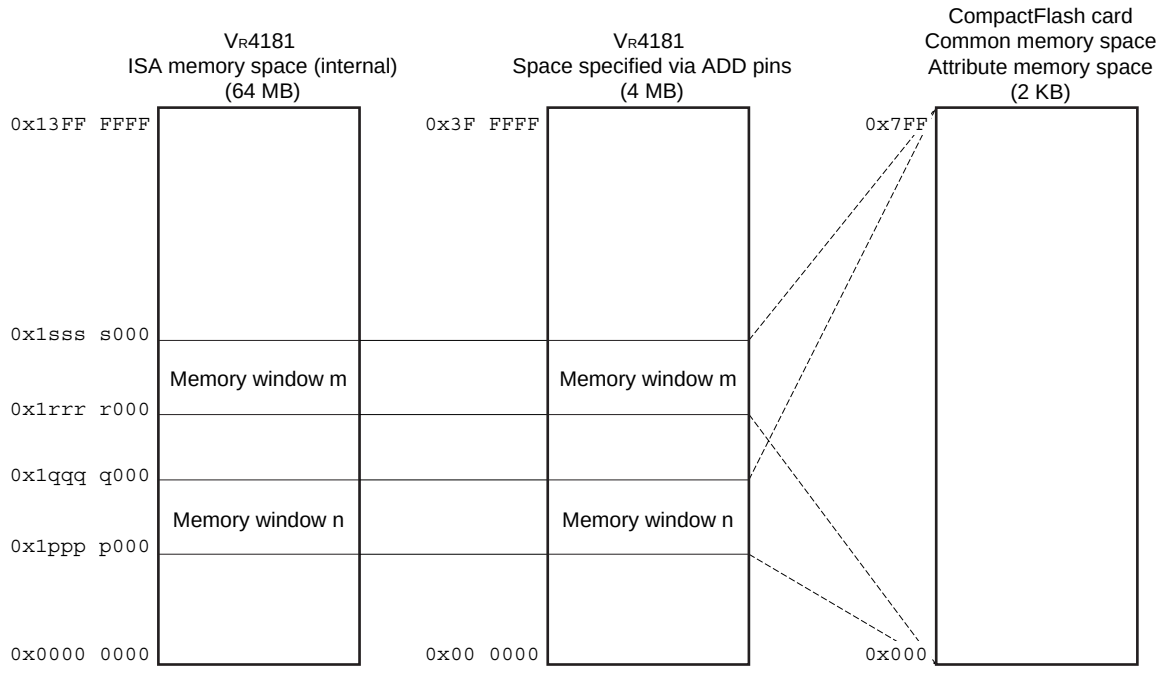
★ 17.5 Memory Mapping of CompactFlash Card

(1) Memory window

In the VR4181, memory windows can be placed at any address in the ISA memory space. The start address of a memory window is output without modification to the VR4181's ADD pins. However, spaces used for programmable chip select, LCD chip select, etc. must not be specified as a memory window.

The CompactFlash memory space is 2 KB and the minimum memory window size is 4 KB. Accordingly, when using CompactFlash with the VR4181, the card's entire memory space is mapped to a single memory window. Mapping starts from the LSB. The remaining part of the memory window becomes a mirror area occupying the lower 2 KB.

Figure 17-2. Mapping of CompactFlash Memory Space



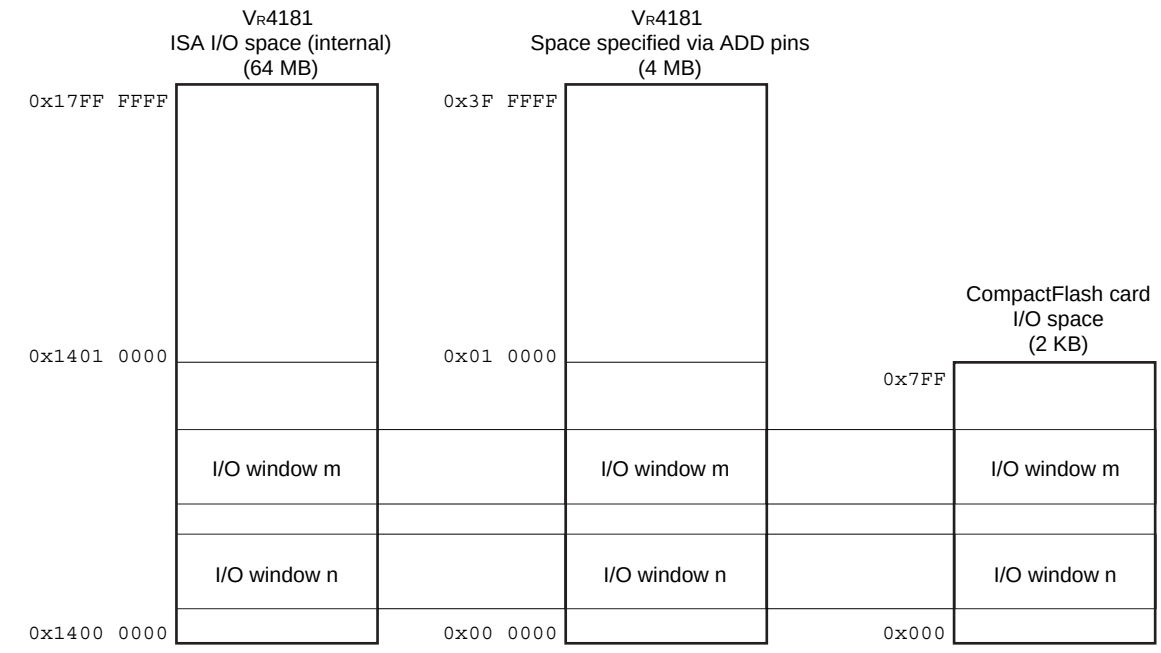
Remark m, n = 0 to 4
p, q, r, s = don't care

(2) I/O window

In the VR4181, the I/O window can be mapped to any address within the external ISA I/O space's lower 64 KB. The start address of a window is output without modification to the VR4181's ADD pins.

When using the CompactFlash card, do not map a space for programmable chip select or another external device to the lower 64 KB within external ISA I/O space where I/O windows are assigned.

Figure 17-3. Mapping of CompactFlash I/O Space



Remark m, n = 0 or 1

★ 17.6 Controlling Bus When CompactFlash Card Is Used

Access to the CompactFlash card is made via the ISA bridge. The address, data, and command signals operate based on external ISA cycles. The operations of the signals that control the bus size and wait state (MEMCS16#, IOCS16#, and IORDY) can be set in the ECU.

17.6.1 Controlling bus size

When the memory window is accessed, the data bus width is set in the DWIDTH bit of the MEMWIDn_REG register (n = 0 to 4). This setting is output from the ECU to the ISA bridge as the source of the MEMCS16# signal.

When the I/O window is accessed, the source of the data width is selected from the CF_IOIS16# signal or IOnDSZ bit (n = 0 or 1) via the IOn_CS16MD bit (n = 0 or 1) of the IOCTRL_REG register. If the CF_IOIS16# signal is selected, its status is output from the ECU to the ISA bridge as the source of the IOCS16# signal. If the IOnDSZ bit is selected, the inverted setting of the IOnDSZ bit is output.

17.6.2 Controlling wait

The number of wait states of the external ISA cycle can be selected from four types by using the MEMWS(1:0) and IOWS(1:0) bits of the XISACTL register of the ISA bridge, regardless of whether the memory or I/O is accessed.

In addition, the ECU deasserts the IORDY signal and extends the bus cycle if the CF_WAIT# signal from the CompactFlash card is asserted. Additional wait states can be controlled by ECU settings.

(1) Wait when memory window is accessed

The zero wait state can be enabled or disabled via the ZWSEN bit of the MEMWIDn_REG register (n = 0 to 4).

(a) If zero wait state is enabled

A wait state is not added regardless of the bus size. Therefore, wait states are inserted only during the period set in the MEMWS(1:0) bits of the XISACTL register.

(b) If zero wait state is disabled

The number of wait states selected in the M16W(1:0) bits of the MEMSELn_REG register (n = 0 to 4) is added in the 16-bit access mode.

In the 8-bit access mode, a 4 SYSCLK-cycle wait is added.

(2) Wait when I/O window is accessed

(a) 16-bit access

A 2 SYSCLK-cycle wait is added if the IOnWT bit of the IOCTRL_REG register (n = 0 or 1) is set to 1.

A 1 SYSCLK-cycle wait is added if the IOnWT bit of the IOCTRL_REG register (n = 0 or 1) is cleared to 0.

(b) 8-bit access

A 4 SYSCLK-cycle wait is added if the Wn_IOWS bit of the IOCTRL_REG register (n = 0 or 1) is set to 1.

A 3 SYSCLK-cycle wait is added if the Wn_IOWS bit of the IOCTRL_REG register (n = 0 or 1) is cleared to 0.

CHAPTER 18 LED CONTROL UNIT (LED)

18.1 General

This unit switches ON and OFF of LEDs at a regular interval. The interval can be set via software.

18.2 Register Set

The LED registers are listed below.

Table 18-1. LED Registers

Physical address	R/W	Register symbol	Function
★ 0x0B00 0240	R/W	LEDHTSREG	LED ON time set register
★ 0x0B00 0242	R/W	LEDLTSREG	LED OFF time set register
0x0B00 0248	R/W	LEDCNTREG	LED control register
0x0B00 024A	R/W	LEDASTCREG	LED auto stop time setting register
0x0B00 024C	R/W	LEDINTREG	LED interrupt register

These registers are described in detail below.

18.2.1 LEDHTSREG (0x0B00 0240)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	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
Other resets	0	0	0	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 5	Reserved	0 is returned when read
4 to 0	HTS(4:0)	LED ON time setting 11111 : 1.9375 seconds : 10000 : 1 second : 01000 : 0.5 seconds : 00100 : 0.25 seconds : 00010 : 0.125 seconds 00001 : 0.0625 seconds 00000 : Prohibited

Note A value before reset is retained.

This register is used to set the LED's ON time (high 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 setting 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.

18.2.2 LEDLTSREG (0x0B00 0242)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	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
Other resets	0	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**	**Note**

Bit	Name	Function
15 to 7	Reserved	0 is returned when read
6 to 0	LTS(6:0)	LED OFF time setting 1111111 : 7.9375 seconds : 1000000 : 4 seconds : 0100000 : 2 seconds : 0010000 : 1 second : 0001000 : 0.5 seconds : 0000100 : 0.25 seconds : 0000010 : 0.125 seconds 0000001 : 0.0625 seconds 0000000 : Prohibited

Note A value before reset is retained.

This register is used to set the LED's OFF time (low 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 setting is 2 seconds.

This register must not be changed once the LEDENABLE bit of LEDCNTREG register has been set to 1. The operation is not guaranteed if a change is made after that point.

18.2.3 LEDCNTREG (0x0B00 0248)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
★ Name	Reserved	Reserved	Reserved	Reserved	Reserved	LEDHLB	LEDSTOP	LEDENABLE
R/W	R	R	R	R	R	R	R/W	R/W
RTCRST	0	0	0	0	0	0	1	0
Other resets	0	0	0	0	0	0	**Note**	**Note**

Bit	Name	Function
15 to 3	Reserved	0 is returned when read
2 ★	LEDHLB	LED status indication 1 : ON 0 : OFF
1	LEDSTOP	LED blink auto stop setting 1 : Automatically stops 0 : Does not stop automatically
0	LEDENABLE	LED blink setting 1 : Blinks 0 : Does not blink

Note A value before reset is retained.

This register is used to make various LED settings.

Caution When setting LED to blink, make sure that a value other than zero has already been set to the LEDHTSREG, LEDLTSREG, and LEDASTCREG registers. The operation is not guaranteed if zero is set to these registers.

18.2.4 LEDASTCREG (0x0B00 024A)

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
Other resets	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
Other resets	1	0	1	1	0	0	0	0

Bit	Name	Function
15 to 0	ASTC(15:0)	LED auto stop time count

This register is used to set the number of ON/OFF times prior to automatic stopping of LED blink. The set value is ★ read on a read. The initial setting is 1,200 times of ON/OFF pairs (i.e. one hour in which each time includes one second of ON time and two seconds of OFF time).

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 request occurs when it reaches zero.

Caution Setting a zero to this register is prohibited. The operation is not guaranteed if zero is set to this register.

18.2.5 LEDINTREG (0x0B00 024C)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

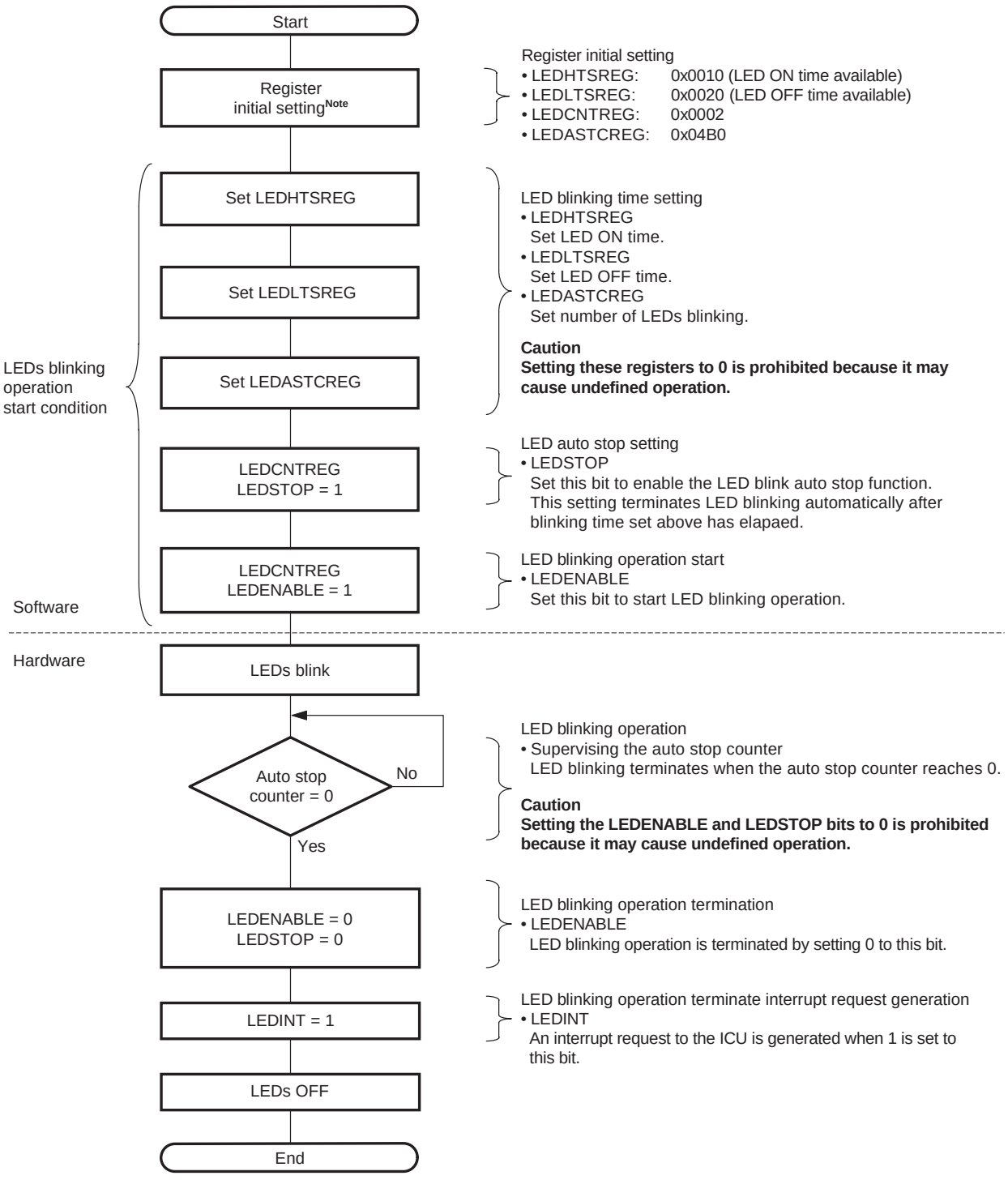
Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	LEDINT
R/W	R	R	R	R	R	R	R	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 1	Reserved	0 is returned when read
0	LEDINT	Auto stop interrupt request. Cleared to 0 when 1 is written. 1 : Occurred 0 : Not occurred

This register indicates when an auto stop interrupt request has occurred.

An auto stop interrupt request occurs when 1 has already been set to both the LEDSTOP bit and the LEDENABLE bit of the LEDCNTREG register if LEDASTCREG register is cleared to 0. When this interrupt occurs, the LEDSTOP bit and the LEDENABLE bit of the LEDCNTREG register are both cleared to 0.

18.3 Operation Flow



Note Initial setting for each register must be performed when a power is supplied to device for the first time, regardless whether LEDs blinking function is used or not.

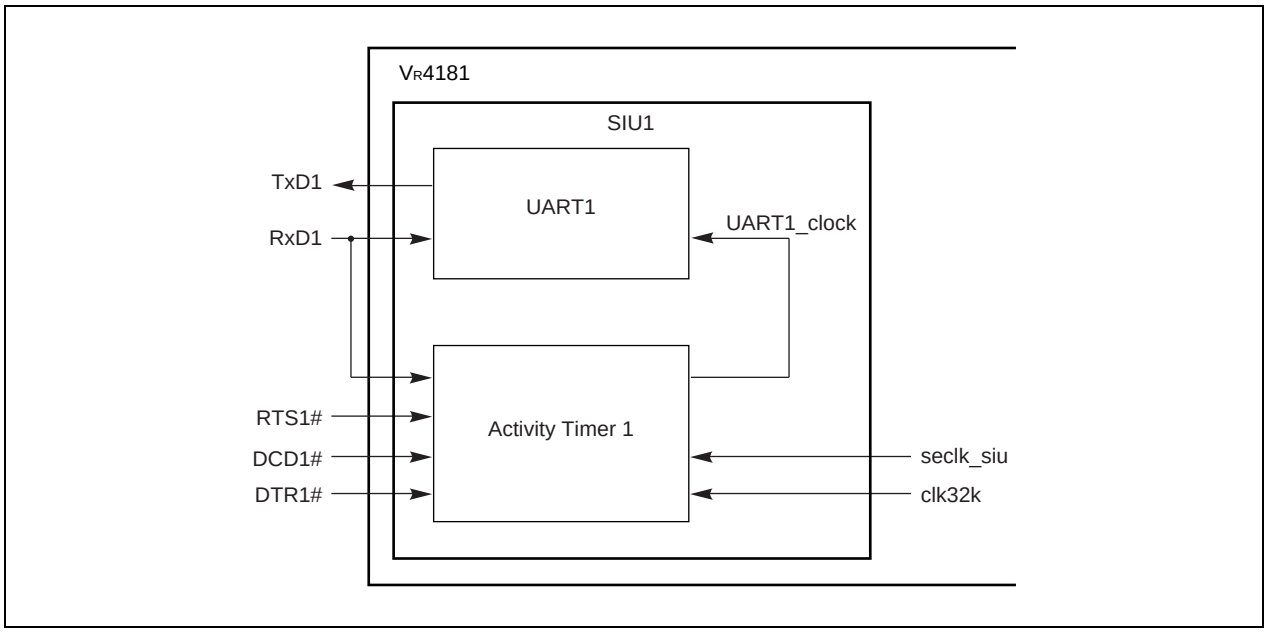
CHAPTER 19 SERIAL INTERFACE UNIT 1 (SIU1)

19.1 General

The SIU1 is a serial interface that conforms to the RS-232-C communication standard and is equipped with two one-channel interfaces, one for transmission and one for reception.

This unit is functionally compatible with the NS16550 except for the additional clock control logic to permit the 16650 core clock source to be stopped.

Figure 19-1. SIU1 Block Diagram



★ **Caution No clock is supplied to the SIU1 in the initial state. When using the SIU1, set the MSKSIU18M bit of the CMUCLKMSK register in the MBA Host Bridge to 1 in advance so that the clock is supplied.**

19.2 Clock Control Logic

The power of the 16550 core can be managed by monitoring activity on the modem status pins and writes to the transmit buffer.

The clock control logic for the 16550 core monitors activity on the four serial interface input signals; RxD1, RTS1#, DCD1#, and DTR1#. It also monitors writes to the 16550 transmit buffer. Each source has an associated mask bit which prevents a source from causing reset of the Activity Timer.

Activity on the RxD1, RTS1#, DCD1#, and DTR1# inputs is defined as any change of state (high to low or low to high). When no unmasked activity has been detected on any of the inputs, and no writes have occurred to the transmit buffer within the programmed time-out period specified in the Activity Timer block, the UART1_clock is stopped. The UART1_clock will remain stopped until any activity is detected on the monitored sources.

19.3 Register Set

The SIU1 registers are listed below.

Table 19-1. SIU1 Registers

Physical address	LCR7	R/W	Register symbol	Function
0x0C00 0010	0	R	SIURB_1	Receive buffer register (read)
		W	SIUTH_1	Transmit holding register (write)
	1	R/W	SIUDLL_1	Divisor latch (least significant byte) register
0x0C00 0011	0	R/W	SIUIE_1	Interrupt enable register
	1	R/W	SIUDLM_1	Divisor latch (most significant byte) register
0x0C00 0012	—	R	SIUIID_1	Interrupt identification register (read)
	—	W	SIUFC_1	FIFO control register (write)
0x0C00 0013	—	R/W	SIULC_1	Line control register
0x0C00 0014	—	R/W	SIUMC_1	Modem control register
★ 0x0C00 0015	—	R	SIULS_1	Line status register
0x0C00 0016	—	R/W	SIUMS_1	Modem status register
0x0C00 0017	—	R/W	SIUSC_1	Scratch register
0x0C00 0019	—	R/W	SIURESET_1	SIU reset register
0x0C00 001C	—	R/W	SIUACTMSK_1	SIU activity mask register
0x0C00 001E	—	R/W	SIUADTTMR_1	SIU Activity Timer register

Remark LCR7 is bit 7 of the SIULC_1 register.

19.3.1 SIURB_1 (0x0C00 0010: 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	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
★ Other resets	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Name	Function
7 to 0	RXD(7:0)	Serial receive data

This register stores receive data used in serial communications.

To access this register, set the LCR7 bit (bit 7 of the SIULC_1 register) to 0.

19.3.2 SIUTH_1 (0x0C00 0010: 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	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
★ Other resets	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Name	Function
7 to 0	TXD(7:0)	Serial transmit data

This register stores transmit data used in serial communications.

To access this register, set the LCR7 bit (bit 7 of the SIULC_1 register) to 0.

19.3.3 SIUDLL_1 (0x0C00 0010: 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	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
★ Other resets	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Name	Function
7 to 0	DLL(7:0)	Baud rate divisor (low-order byte)

This register is used to set the divisor (division rate) for the baud rate generator.

The data in this register and the data in the SIUDLM_1 register as upper 8 bits are together handled as 16-bit data.

To access this register, set the LCR7 bit (bit 7 of the SIULC_1 register) to 1.

19.3.4 SIUIE_1 (0x0C00 0011: LCR7 = 0)

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	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
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
7 to 4	Reserved	0 is returned when read
3	IE3	Modem status interrupt 1 : Enable 0 : Prohibit
2	IE2	Receive status interrupt 1 : Enable 0 : Prohibit
1	IE1	Transmit holding register empty interrupt 1 : Enable 0 : Prohibit
0	IE0	Receive data ready interrupt or character timeout interrupt in FIFO mode 1 : Enable 0 : Prohibit

This register is used to specify interrupt enable/prohibit settings for the five types of interrupt requests used in the SIU1.

An interrupt is enabled by setting the corresponding bit to 1.

Overall use of interrupt functions can be halted by setting bits 0 to 3 of this register to 0.

★ When interrupts are prohibited, “pending” is not displayed in the IIR0 bit in the SIUIID_1 register even when interrupt conditions have been met.

Other functions in the SIU1 are not affected even though interrupts are prohibited and the settings in the SIULS_1 register and SIUMS_1 register are valid.

To access this register, set the LCR7 bit (bit 7 of the SIULC_1 register) to 0.

19.3.5 SIUDLM_1 (0x0C00 0011: 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	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
★ Other resets	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Name	Function
7 to 0	DLM(7:0)	Baud rate divisor (high-order byte)

This register is used to set the divisor (division rate) for the baud rate generator.

The data in this register and the data in the SIUDLL_1 register as lower 8 bits are together handled as 16-bit data.

To access this register, set the LCR7 bit (bit 7 of the SIULC_1 register) to 1.

The relationship between baud rates and the settings of the SIUDLL_1 and SIUDLM_1 registers are as follows.

★ **Table 19-2. Correspondence between Baud Rates and Divisors**

Baud rate (bps)	Divisor (DLM(7:0)||DLL(7:0))	1-clock width (μs)
50	23040	20000.00
75	15360	13333.33
110	10473	9090.91
134.5	8565	7434.94
150	7680	6666.67
300	3840	3333.33
600	1920	1666.67
1200	960	833.33
1800	640	555.56
2000	576	500.00
2400	480	416.67
3600	320	277.78
4800	240	208.33
7200	160	138.89
9600	120	104.17
19200	60	52.08
38400	30	26.04
57600	20	17.36
115200	10	8.68
128000	9	7.81
144000	8	6.94
192000	6	5.21
230400	5	4.34
288000	4	3.47
384000	3	2.60
576000	2	1.74
1152000	1	0.868

19.3.6 SIUIID_1 (0x0C00 0012: Read)

Bit	7	6	5	4	3	2	1	0
Name	IIR7	IIR6	Reserved	Reserved	IIR3	IIR2	IIR1	IIR0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	1
Other resets	0	0	0	0	0	0	0	1

Bit	Name	Function
7, 6	IIR(7:6)	Becomes 11 when FCR0 bit = 1
5, 4	Reserved	0 is returned when read
3	IIR3	Pending of the character timeout interrupt request (in FIFO mode) 1 : No pending 0 : Pending
2, 1	IIR(2:1)	Indicates the priority level of interrupts. See the following table.
0	IIR0	Pending interrupt requests 1 : No pending 0 : Pending

This register indicates priority levels for interrupts and existence of pending interrupt requests.

From highest to lowest priority, the involved interrupts are the receive line status, the receive data ready, the character timeout, the transmit holding register empty, and the modem status.

The content of the IIR3 bit is valid only in the FIFO mode and it is always 0 in the 16450 mode.

The IIR2 bit becomes 1 when the IIR3 bit is set to 1.

Table 19-3. Interrupt Function

SIUIID_1 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 ready	Receive data exists or has reached the trigger level.	Read the receive buffer register or lower the data in the FIFO than trigger level.
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	Transmit holding register empty	Transmit register is empty	Read IIR (if it is the interrupt source) or write to transmit holding register
0	0	0	4th	Modem status	CTS1#, DSR1#, or DCD1#	Read modem status register

Note FIFO mode only.

19.3.7 SIUFC_1 (0x0C00 0012: Write)

Bit	7	6	5	4	3	2	1	0
Name	FCR7	FCR6	Reserved	Reserved	FCR3	FCR2	FCR1	FCR0
R/W	W	W	R	R	W	W	W	W
RTCRST	0	0	0	0	0	0	0	0
Other resets	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	Reserved	0 is returned when 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 and its counter clear. Cleared to 0 when 1 is written. 1 : FIFO and its counter clear 0 : Normal
★	1	FCR1	Receive FIFO and its counter clear. Cleared to 0 when 1 is written. 1 : FIFO and its counter clear 0 : Normal
★	0	FCR0	Receive/Transmit FIFO enable. Cleared to 0 when 1 is written. 1 : Enable 0 : Disable

This register is used to control the FIFOs.

- **FIFO interrupt modes**

  When receive FIFO is enabled and receive interrupt requests are enabled, receive interrupts can occur as described below.

  1. When the FIFO is reached to the specified trigger level, a receive data ready interrupt request is notified to the CPU.
     This interrupt is cleared when the FIFO goes below the trigger level.

  2. When the FIFO is reached to the specified trigger level, the SIUIID_1 register indicates a receive data ready interrupt request.
     Same as the interrupt above, the SUIID_1 register is cleared when the FIFO goes below the trigger level.

  3. The receive line status interrupt is assigned to a higher priority level than the receive data ready interrupt.

  4. When characters are transferred from the shift register to the receive FIFO, 1 is set to the LSR0 bit of the SIULS_1 register.
     The value of this bit returns to 0 when the FIFO becomes empty.

  When receive FIFO is enabled and receive interrupts are enabled, receive FIFO timeout interrupt requests can occur as described below.

  1. Followings are the conditions under which FIFO timeout interrupt requests occur.

     - 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 (includes the time for the second stop bit in cases where it is specified that two stop bits are required).
     - The time required for sending four characters has elapsed since the last read of the FIFO by the CPU.

     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) (which is why the elapsed time is in proportion to the baud rate).

  3. Once a timeout interrupt request has occurred, 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 occurred, the timer is reset when a new character is received or when the CPU reads the receive FIFO.

When transmit FIFO is enabled and transmit interrupts are enabled, transmit interrupt requests can occur as described below.

1. When the transmit FIFO becomes empty, a transmit holding register empty interrupt request occurs.
   This interrupt request is cleared when a character is written to the transmit holding register (from one to 16 characters can be written to the transmit FIFO during servicing of this interrupt), or when the SIUIID_1 register is read.

2. If there are not at least two bytes of character data in the transmit FIFO between one time when the LSR5 bit = 1 (transmit FIFO is empty) in the SIULS_1 register and the next time when the LSR5 bit = 1, transmit FIFO empty status is reported to the IIR bits 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 request that occurs after the FCR0 bit (FIFO enable bit) in the SIUFC_1 register 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 ready interrupt.
The priority level of the transmit FIFO empty interrupt is the same as that of the transmit holding register empty interrupt.
Whether data to be transmitted exists or not in the transmit FIFO and the transmit shift register, check the LSR6 bit of the SIULS_1 register. The LSR5 bit of the SIULS_1 register is used to check whether data to be transferred exists or not in the transmit FIFO only. Therefore, there may be data in the transmit shift register.

- **FIFO polling mode**

  When the FCR0 bit = 1 (FIFO is enabled) in the SIUFC_1 register, if the value of any or all of the SIUIE_1 register bits 3 to 0 becomes 0, SIU1 enters FIFO polling mode. Because the transmit block and receive block are controlled separately, polling mode can be set for either or both blocks.
  When in this mode, the status of the transmit block and/or receive block can be checked by reading the SIULS_1 register via a user program.
  When in the FIFO polling mode, there is no notification when the trigger level is reached or when a timeout occurs, but the receive FIFO and transmit FIFO can still store characters as they normally do.

19.3.8 SIULC_1 (0x0C00 0013)

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
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
7	LCR7	Divisor latch access register switching 1 : Divisor latch access register 0 : Receive buffer, transmit holding 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 specification 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 register.

The LCR6 bit is used to send the break status to the receive side's UART. When the LCR6 bit = 1, the serial output (TxD1) is forcibly set to the spacing (0) state.

The setting of the LCR5 bit becomes valid according to settings in the LCR4 and LCR3 bits.

19.3.9 SIUMC_1 (0x0C00 0014)

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	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
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
7 to 5	Reserved	0 is returned when read
4	MCR4	Use of 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	RTS1# output control 1 : Low level 0 : High level
0	MCR0	DTR1# output control 1 : Low level 0 : High level

This register is used to control the interface 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 (enable use of local loopback).

- **Local Loopback**

  The local loopback can be used to test the transmit/receive data path in SIU1. The following operation (local loopback) is executed inside the SIU1 when the MCR4 bit = 1.

  The transmit block's serial output (TxD1) enters the marking state (1) and the serial input (RxD1) 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 (DSR1#, CTS1#, RI (internal), and DCD1#) are cut off and the four modem control outputs (DTR1#, RTS1#, 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 sources in relation to the transmit and receive blocks. Although modem control interrupts can be used, the low-order four 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.

19.3.10 SIULS_1 (0x0C00 0015)

Bit	7	6	5	4	3	2	1	0
Name	LSR7	LSR6	LSR5	LSR4	LSR3	LSR2	LSR1	LSR0
★ R/W	R	R	R	R	R	R	R	R
RTCRST	0	1	1	0	0	0	0	0
Other resets	0	1	1	0	0	0	0	0

Bit	Name	Function
7	LSR7	Error detection (FIFO mode) 1 : Parity error, framing error, or break is detected. 0 : No error
6	LSR6	Transmit block empty 1 : No data in transmit holding register and transmit shift register No data in transmit FIFO (during FIFO mode) 0 : Data exists in transmit holding register or transmit shift register Data exists in transmit FIFO (during FIFO mode)
5	LSR5	Transmit holding register empty 1 : Character is transferred to transmit shift register (during 16450 mode) Transmit FIFO is empty (during FIFO mode) 0 : Character is stored in transmit holding register (during 16450 mode) Transmit data exists in transmit FIFO (during FIFO mode)
4	LSR4	Break interrupt 1 : Detected 0 : No break
3	LSR3	Framing error 1 : Detected 0 : No error
2	LSR2	Parity error 1 : Detected 0 : No error
1	LSR1	Overrun error 1 : Detected (receive data is overwritten) 0 : No error
0	LSR0	Receive data ready 1 : Receive data exists in FIFO 0 : No receive data in FIFO

The CPU uses this register to get information related to data transfers.

When LSR7 and LSR(4:1) bits are 1, reading this register clears these bits to 0.

Caution The LSR0 bit (receive data ready bit) is set before the serial data reception is completed. Therefore, the LSR0 bit may not be cleared if the serial receive data is read from the SIURB_1 register immediately after this bit is set.

When reading data from the SIURB_1 register, wait for the stop bit width time since the LSR0 bit is set.

LSR7 bit is valid only in FIFO mode, and it indicates always 0 in 16450 mode.

The value of LSR4 bit becomes 1 when the spacing status (0) of receive data input is held longer than the time required for transmission of one word (start bit + data bits + parity bit + stop bit). When in FIFO mode, if a break is detected for one character in the FIFO, the character is regarded as an error character and the CPU is notified of a break when that character reaches the highest position in the FIFO. When a break occurs, one “zero” character is sent to the FIFO. When the RxD1 enters marking status, and the next valid start bit is received, the next character can be transmitted.

The value of LSR3 bit becomes 1 when a zero (spacing level) stop bit is detected following the final data bit or parity bit. When 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 SIU1 prepares for synchronization again. The next start bit is assumed to be the cause of the framing error and the next data is not accepted until the next start bit has been sampled twice.

★ The value of LSR2 bit becomes 1 when a received character does not satisfy the even or odd parity specified in the LCR4 bit. When in FIFO mode, if a parity error is detected for one character within 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 LSR1 bit becomes 1 when a character is transferred to the receive buffer register before reading by the CPU and the previous character is lost. When in FIFO mode, if the data continues to be transferred to the FIFO though it exceeds the trigger level, even after the FIFO becomes full an overrun error will not occur until all characters are stored in the shift register.

The CPU is notified as soon as an overrun error occurs. The characters in the shift register are overwritten and are not transferred to the FIFO.

19.3.11 SIUMS_1 (0x0C00 0016)

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
Other resets	Undefined	Undefined	Undefined	Undefined	0	0	0	0

Bit	Name	Function
7	MSR7	DCD1# signal status 1 : Low level 0 : High level
6	MSR6	RI signal (internal) status 1 : Low level 0 : High level
5	MSR5	DSR1# input status 1 : Low level 0 : High level
4	MSR4	CTS1# input status 1 : Low level 0 : High level
3	MSR3	DCD1# signal change 1 : Changed 0 : No change
2	MSR2	RI signal (internal) change 1 : Changed 0 : No change
1	MSR1	DSR1# signal change 1 : Changed 0 : No change
0	MSR0	CTS1# signal change 1 : Changed 0 : No change

This register indicates the current status and change in status of various control signals that are input to the CPU from a modem or other peripheral device.

The MSR(3:0) bits are cleared to 0 if they are read when they are set to 1.

19.3.12 SIUSC_1 (0x0C00 0017)

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	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
★ Other resets	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Name	Function
7 to 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 SIU1.

19.3.13 SIURESET_1 (0x0C00 0019)

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	SIU RESET
R/W	R	R	R	R	R	R	R	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
7 to 1	Reserved	0 is returned when read
0	SIURESET	SIU1 reset 1 : Reset 0 : Release reset

This register is used to reset SIU1 forcibly.

19.3.14 SIUACTMSK_1 (0x0C00 001C)

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	RxDMSK	RTSMSK	DCDMSK	DTRMSK	Reserved	TxWRMSK
★ 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
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
7, 6	Reserved	0 is returned when read
5	RxDMSK	Mask for notification of change on RxD1 1 : Mask 0 : Unmask
4	RTSMSK	Mask for notification of change on RTS1# 1 : Mask 0 : Unmask
3	DCDMSK	Mask for notification of change on DCD1# 1 : Mask 0 : Unmask
2	DTRMSK	Mask for notification of change on DTR1# 1 : Mask 0 : Unmask
1	Reserved	Write 0 when write. 0 is returned when read
0	TxWRMSK	Mask for notification of transmit buffer write 1 : Mask 0 : Unmask

This register is used to set masks for notification of operation statuses to the Activity Timer of the SIU1.

When 1 is set in this register, state transition of the corresponding signals or write to transmit buffer is not notified to the Activity Timer.

19.3.15 SIUACTTMR_1 (0x0C00 001E)

Bit	7	6	5	4	3	2	1	0
Name	SIUTMO7	SIUTMO6	SIUTMO5	SIUTMO4	SIUTMO3	SIUTMO2	SIUTMO1	SIUTMO0
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
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
7 to 0	SIUTMO(7:0)	SIU activity timeout period 11111111 : 255 x 30.5 μs 11111110 : 254 x 30.5 μs : 01111111 : 127 x 30.5 μs : 00000001 : 30.5 μs 00000000 : Activity Timer disabled

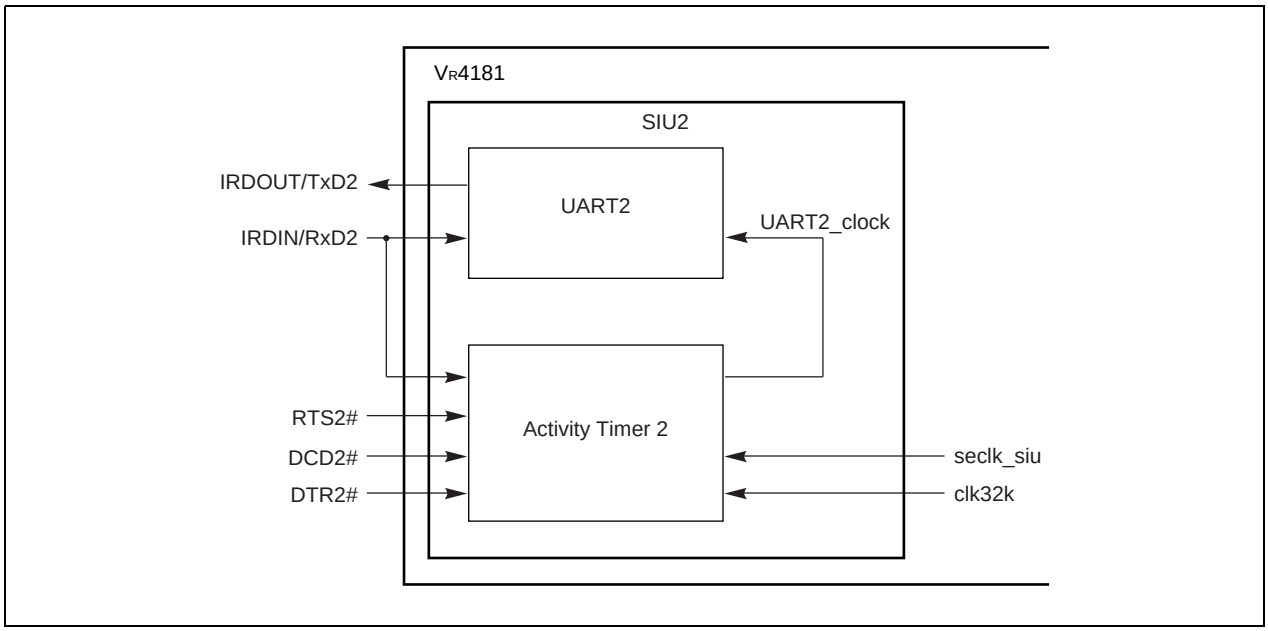
CHAPTER 20 SERIAL INTERFACE UNIT 2 (SIU2)

20.1 General

The SIU2 is a serial interface that conforms to the RS-232-C communication standard and is equipped with two ★ one-channel interfaces, one for transmission and one for reception. This unit can be also used as an interface in the IrDA format by means of register setting.

This unit is functionally compatible with the NS16550 except for the additional clock control logic to permit the 16650 core clock source to be stopped.

Figure 20-1. SIU2 Block Diagram



★ **Caution No clock is supplied to the SIU2 in the initial state. When using the SIU2, set the MSKSIU18M bit of the CMUCLKMSK register in the MBA Host Bridge to 1 in advance so that the clock is supplied.**

20.2 Clock Control Logic

The power of the 16550 core can be managed by monitoring activity on the modem status pins and writes to the transmit buffer.

The clock control logic for the 16550 core monitors activity on the four serial interface input signals; RxD2, RTS2#, DCD2#, and DTR2#. It also monitors writes to the 16550 transmit buffer. Each source has an associated mask bit which prevents a source from causing reset of the Activity Timer.

Activity on the RxD2, RTS2#, DCD2# and DTR2# inputs is defined as any change of state (high to low or low to high). When no unmasked activity has been detected on any of the inputs, and no writes have occurred to the transmit buffer within the programmed time-out period specified in the Activity Timer block, the UART2_clock is stopped. The UART2_clock will remain stopped until the activity is detected on the monitored sources.

20.3 Register Set

The SIU2 registers are listed below.

Table 20-1. SIU2 Registers

Physical address	LCR7	R/W	Register symbol	Function
0x0C00 0000	0	R	SIURB_2	Receive buffer register (read)
		W	SIUTH_2	Transmit holding register (write)
	1	R/W	SIUDLL_2	Divisor latch (least significant byte) register
0x0C00 0001	0	R/W	SIUIE_2	Interrupt enable register
	1	R/W	SIUDLM_2	Divisor latch (most significant byte) register
0x0C00 0002	—	R	SIUIID_2	Interrupt identification register (read)
	—	W	SIUFC_2	FIFO control register (write)
0x0C00 0003	—	R/W	SIULC_2	Line control register
0x0C00 0004	—	R/W	SIUMC_2	Modem control register
★ 0x0C00 0005	—	R	SIULS_2	Line status register
0x0C00 0006	—	R/W	SIUMS_2	Modem status register
0x0C00 0007	—	R/W	SIUSC_2	Scratch register
0x0C00 0008	—	R/W	SIUIRSEL_2	SIU IrDA select register
0x0C00 0009	—	R/W	SIURESET_2	SIU reset register
0x0C00 000A	—	R/W	SIUCSEL_2	SIU echo back control register
0x0C00 000C	—	R/W	SIUACTMSK_2	SIU activity mask register
0x0C00 000E	—	R/W	SIUADTTMR_2	SIU Activity Timer register

Remark LCR7 is bit 7 of the SIULC_2 register.

20.3.1 SIURB_2 (0x0C00 0000: 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	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
★ Other resets	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Name	Function
7 to 0	RXD(7:0)	Serial receive data

This register stores receive data used in serial communications.

To access this register, set the LCR7 bit (bit 7 of the SIULC_2 register) to 0.

20.3.2 SIUTH_2 (0x0C00 0000: 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	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
★ Other resets	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Name	Function
7 to 0	TXD(7:0)	Serial transmit data

This register stores transmit data used in serial communications.

To access this register, set the LCR7 bit (bit 7 of the SIULC_2 register) to 0.

20.3.3 SIUDLL_2 (0x0C00 0000: 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	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
★ Other resets	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Name	Function
7 to 0	DLL(7:0)	Baud rate divisor (low-order byte)

This register is used to set the divisor (division rate) for the baud rate generator.

The data in this register and the data in SIUDLM_2 register as upper 8 bits are together handled as 16-bit data.

To access this register, set the LCR7 bit (bit 7 of the SIULC_2 register) to 1.

20.3.4 SIUIE_2 (0x0C00 0001: LCR7 = 0)

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	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
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
7 to 4	Reserved	0 is returned when read
3	IE3	Modem status interrupt 1 : Enable 0 : Prohibit
2	IE2	Receive status interrupt 1 : Enable 0 : Prohibit
1	IE1	Transmit holding register empty interrupt 1 : Enable 0 : Prohibit
0	IE0	Receive data ready interrupt or character timeout interrupt in FIFO mode 1 : Enable 0 : Prohibit

This register is used to specify interrupt enable/prohibit settings for the five types of interrupt requests used in the SIU2.

An interrupt is enabled by setting the corresponding bit to 1.

Overall use of interrupt functions can be halted by setting bits 0 to 3 of this register to 0.

★ When interrupts are prohibited, “pending” is not displayed in the IIR0 bit in the SIUIID_2 register even when interrupt conditions have been met.

Other functions in the SIU2 are not affected even though interrupts are prohibited and the settings in the SIULS_2 register and SIUMS_2 register are valid.

To access this register, set the LCR7 bit (bit 7 of the SIULC_2 register) to 0.

20.3.5 SIUDLM_2 (0x0C00 0001: 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	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
★ Other resets	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Name	Function
7 to 0	DLM(7:0)	Baud rate divisor (high-order byte)

This register is used to set the divisor (division rate) for the baud rate generator.

The data in this register and the data in SIUDLL_2 register as lower 8 bits are together handled as 16-bit data.

To access this register, set the LCR7 bit (bit 7 of the SIULC_2 register) to 1.

The relationship between baud rates and the settings of the SIUDLL_2 and SIUDLM_2 registers are as follows.

★ **Table 20-2. Correspondence between Baud Rates and Divisors**

Baud rate (bps)	Divisor (DLM(7:0)||DLL(7:0))	1-clock width (μs)
50	23040	20000.00
75	15360	13333.33
110	10473	9090.91
134.5	8565	7434.94
150	7680	6666.67
300	3840	3333.33
600	1920	1666.67
1200	960	833.33
1800	640	555.56
2000	576	500.00
2400	480	416.67
3600	320	277.78
4800	240	208.33
7200	160	138.89
9600	120	104.17
19200	60	52.08
38400	30	26.04
57600	20	17.36
115200	10	8.68
128000	9	7.81
144000	8	6.94
192000	6	5.21
230400	5	4.34
288000	4	3.47
384000	3	2.60
576000	2	1.74
1152000	1	0.868

20.3.6 SIUIID_2 (0x0C00 0002: Read)

Bit	7	6	5	4	3	2	1	0
Name	IIR7	IIR6	Reserved	Reserved	IIR3	IIR2	IIR1	IIR0
R/W	R	R	R	R	R	R	R	R
RTCRST	0	0	0	0	0	0	0	1
Other resets	0	0	0	0	0	0	0	1

Bit	Name	Function
7, 6	IIR(7:6)	Becomes 11 when FCR0 bit = 1
5, 4	Reserved	0 is returned when read
3	IIR3	Pending of the character timeout interrupt request (in FIFO mode) 1 : No pending 0 : Pending
2, 1	IIR(2:1)	Indicates the priority level of interrupts. See the following table.
0	IIR0	Pending interrupt requests 1 : No pending 0 : Pending

This register indicates priority levels for interrupts and existence of pending interrupt requests.

From highest to lowest priority, the involved interrupts are the receive line status, the receive data ready, the character timeout, the transmit holding register empty, and the modem status.

The content of the IIR3 bit is valid only in the FIFO mode and it is always 0 in the 16450 mode.

The IIR2 bit becomes 1 when the IIR3 bit is set to 1.

Table 20-3. Interrupt Function

SIUIID_2 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	Read line status register
0	1	0	2nd	Receive data ready	Receive data exists or has reached the trigger level.	Read the receive buffer register or lower the data in the FIFO than trigger level.
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	Transmit holding register empty	Transmit register is empty	Read IIR (if it is the interrupt source) or write to transmit holding register
0	0	0	4th	Modem status	CTS2#, DSR2#, or DCD2#	Read modem status register

Note FIFO mode only.

20.3.7 SIUFC_2 (0x0C00 0002: Write)

Bit	7	6	5	4	3	2	1	0
Name	FCR7	FCR6	Reserved	Reserved	FCR3	FCR2	FCR1	FCR0
R/W	W	W	R	R	W	W	W	W
RTCRST	0	0	0	0	0	0	0	0
Other resets	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	Reserved	0 is returned when 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 and its counter clear. Cleared to 0 when 1 is written. 1 : FIFO and its counter clear 0 : Normal
1 ★	FCR1	Receive FIFO and its counter clear. Cleared to 0 when 1 is written. 1 : FIFO and its counter clear 0 : Normal
0 ★	FCR0	Receive/Transmit FIFO enable. Cleared to 0 when 1 is written. 1 : Enable 0 : Disable

This register is used to control the FIFOs.

- **FIFO interrupt modes**

  When receive FIFO is enabled and receive interrupt requests are enabled, receive interrupts can occur as described below.

  1. When the FIFO is reached to the specified trigger level, a receive data ready interrupt request is notified to the CPU.
     This interrupt is cleared when the FIFO goes below the trigger level.

  2. When the FIFO is reached to the specified trigger level, the SIUIID_2 register indicates a receive data ready interrupt request.
     Same as the interrupt above, the SUIID_2 register is cleared when the FIFO goes below the trigger level.

  3. The receive line status interrupt is assigned to a higher priority level than the receive data ready interrupt.

  4. When characters are transferred from the shift register to the receive FIFO, 1 is set to the LSR0 bit of the SIULS_2 register.
     The value of this bit returns to 0 when the FIFO becomes empty.

  When receive FIFO is enabled and receive interrupts are enabled, receive FIFO timeout interrupt requests can occur as described below.

  1. Followings are the conditions under which FIFO timeout interrupt requests occur.

     - 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 (includes the time for the second stop bit in cases where it is specified that two stop bits are required).
     - The time required for sending four characters has elapsed since the last read of the FIFO by the CPU.

     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) (which is why the elapsed time is in proportion to the baud rate).

  3. Once a timeout interrupt request has occurred, 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 occurred, the timer is reset when a new character is received or when the CPU reads the receive FIFO.

When transmit FIFO is enabled and transmit interrupts are enabled, transmit interrupt requests can occur as described below.

1. When the transmit FIFO becomes empty, a transmit holding register empty interrupt request occurs.
   This interrupt request is cleared when a character is written to the transmit holding register (from one to 16 characters can be written to the transmit FIFO during servicing of this interrupt), or when the SIUIID_2 register is read.

2. If there are not at least two bytes of character data in the transmit FIFO between one time when the LSR5 bit = 1 (transmit FIFO is empty) in the SIULS_2 register and the next time when the LSR5 bit = 1, transmit FIFO empty status is reported to the IIR bits 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 request that occurs after the FCR0 bit (FIFO enable bit) in the SIUFC_2 register 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 ready interrupt.
The priority level of the transmit FIFO empty interrupt is the same as that of the transmit holding register empty interrupt.
Whether data to be transmitted exists or not in the transmit FIFO and the transmit shift register, check the LSR6 bit of the SIULS_2 register. The LSR5 bit of the SIULS_2 register is used to check whether data to be transferred exists or not in the transmit FIFO only. Therefore, there may be data in the transmit shift register.

- **FIFO polling mode**
  When the FCR0 bit = 1 (FIFO is enabled) in the SIUFC_2 register, if the value of any or all of the SIUIE_2 register bits 3 to 0 becomes 0, SIU2 enters FIFO polling mode. Because the transmit block and receive block are controlled separately, polling mode can be set for either or both blocks.
  When in this mode, the status of the transmit block and/or receive block can be checked by reading the SIULS_2 register via a user program.
  When in the FIFO polling mode, there is no notification when the trigger level is reached or when a timeout occurs, but the receive FIFO and transmit FIFO can still store characters as they normally do.

20.3.8 SIULC_2 (0x0C00 0003)

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
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
7	LCR7	Divisor latch access register switching 1 : Divisor latch access register 0 : Receive buffer, transmit holding 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 specification 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 register.

The LCR6 bit is used to send the break status to the receive side's UART. When the LCR6 bit = 1, the serial output (TxD2) is forcibly set to the spacing (0) state.

The setting of the LCR5 bit becomes valid according to settings in the LCR4 and LCR3 bits.

20.3.9 SIUMC_2 (0x0C00 0004)

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	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
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
7 to 5	Reserved	0 is returned when read
4	MCR4	Use of 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	RTS2# output control 1 : Low level 0 : High level
0	MCR0	DTR2# output control 1 : Low level 0 : High level

This register is used to control the interface 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 (enable use of local loopback).

- **Local Loopback**

  The local loopback can be used to test the transmit/receive data path in SIU2. The following operation (local loopback) is executed inside the SIU2 when the MCR4 bit = 1.

  The transmit block's serial output (TxD2) enters the marking state (1) and the serial input (RxD2) 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 (DSR2#, CTS2#, RI (internal), and DCD2#) are cut off and the four modem control outputs (DTR2#, RTS2#, 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 sources in relation to the transmit and receive blocks. Although modem control interrupts can be used, the low-order four 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.

20.3.10 SIULS_2 (0x0C00 0005)

Bit	7	6	5	4	3	2	1	0
Name	LSR7	LSR6	LSR5	LSR4	LSR3	LSR2	LSR1	LSR0
★ R/W	R	R	R	R	R	R	R	R
RTCRST	0	1	1	0	0	0	0	0
Other resets	0	1	1	0	0	0	0	0

Bit	Name	Function
7	LSR7	Error detection (FIFO mode) 1 : Parity error, framing error, or break is detected. 0 : No error
6	LSR6	Transmit block empty 1 : No data in transmit holding register and transmit shift register No data in transmit FIFO (during FIFO mode) 0 : Data exists in transmit holding register or transmit shift register Data exists in transmit FIFO (during FIFO mode)
5	LSR5	Transmit holding register empty 1 : Character is transferred to transmit shift register (during 16450 mode) Transmit FIFO is empty (during FIFO mode) 0 : Character is stored in transmit holding register (during 16450 mode) Transmit data exists in transmit FIFO (during FIFO mode)
4	LSR4	Break interrupt 1 : Detected 0 : No break
3	LSR3	Framing error 1 : Detected 0 : No error
2	LSR2	Parity error 1 : Detected 0 : No error
1	LSR1	Overrun error 1 : Detected (receive data is overwritten) 0 : No error
0	LSR0	Receive data ready 1 : Receive data exists in FIFO 0 : No receive data in FIFO

The CPU uses this register to get information related to data transfers.

When LSR7 and LSR(4:1) bits are 1, reading this register clears these bits to 0.

Caution The LSR0 bit (receive data ready bit) is set before the serial data reception is completed. Therefore, the LSR0 bit may not be cleared if the serial receive data is read from the SIURB_2 register immediately after this bit is set.

When reading data from the SIURB_2 register, wait for the stop bit width time since the LSR0 bit is set.

LSR7 bit is valid only in FIFO mode, and it indicates always 0 in 16450 mode.

The value of LSR4 bit becomes 1 when the spacing status (0) of receive data input is held longer than the time required for transmission of one word (start bit + data bits + parity bit + stop bit). When in FIFO mode, if a break is detected for one character in the FIFO, the character is regarded as an error character and the CPU is notified of a break when that character reaches the highest position in the FIFO. When a break occurs, one “zero” character is sent to the FIFO. When the RxD2 enters marking status, and the next valid start bit is received, the next character can be transmitted.

The value of LSR3 bit becomes 1 when a zero (spacing level) stop bit is detected following the final data bit or parity bit. When 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 SIU2 prepares for synchronization again. The next start bit is assumed to be the cause of the framing error and the next data is not accepted until the next start bit has been sampled twice.

★ The value of LSR2 bit becomes 1 when a received character does not satisfy the even or odd parity specified in the LCR4 bit. When in FIFO mode, if a parity error is detected for one character within 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 LSR1 bit becomes 1 when a character is transferred to the receive buffer register before reading by the CPU and the previous character is lost. When in FIFO mode, if the data continues to be transferred to the FIFO though it exceeds the trigger level, even after the FIFO becomes full an overrun error will not occur until all characters are stored in the shift register.

The CPU is notified as soon as an overrun error occurs. The characters in the shift register are overwritten and are not transferred to the FIFO.

20.3.11 SIUMS_2 (0x0C00 0006)

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
Other resets	Undefined	Undefined	Undefined	Undefined	0	0	0	0

Bit	Name	Function
7	MSR7	DCD2# signal status 1 : Low level 0 : High level
6	MSR6	RI signal (internal) status 1 : Low level 0 : High level
5	MSR5	DSR2# input status 1 : Low level 0 : High level
4	MSR4	CTS2# input status 1 : Low level 0 : High level
3	MSR3	DCD2# signal change 1 : Changed 0 : No change
2	MSR2	RI signal (internal) change 1 : Changed 0 : No change
1	MSR1	DSR2# signal change 1 : Changed 0 : No change
0	MSR0	CTS2# signal change 1 : Changed 0 : No change

This register indicates the current status and change in status of various control signals that are input to the CPU from a modem or other peripheral device.

The MSR(3:0) bits are cleared to 0 if they are read when they are set to 1.

20.3.12 SIUSC_2 (0x0C00 0007)

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	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
★ Other resets	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Name	Function
7 to 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 SIU2.

20.3.13 SIUIRSEL_2 (0x0C00 0008)

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	SIRSEL
R/W	R	R	R/W	R	R/W	R/W	R	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
7, 6	Reserved	0 is returned when read
5	Reserved	Write 0 when write. 0 is returned when read.
4	Reserved	0 is returned when read
3, 2	Reserved	Write 0 when write. 0 is returned when read.
1	Reserved	0 is returned when read
0	SIRSEL	Selects communication format 1 : IrDA 0 : RS-232-C

This register is used to set the SIU2's communication format (IrDA or RS-232-C).

★ To use the IrDA format, an external IrDA module must be connected.

20.3.14 SIURESET_2 (0x0C00 0009)

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	SIU RESET
R/W	R	R	R	R	R	R	R	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
7 to 1	Reserved	0 is returned when read
0	SIURESET	SIU2 reset 1 : Reset 0 : Release reset

This register is used to reset SIU2 forcibly.

20.3.15 SIUCSEL_2 (0x0C00 000A)

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	SIUCSEL
R/W	R	R	R	R	R	R	R	R/W
RTCRST	0	0	0	0	0	0	0	0
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
7 to 1	Reserved	0 is returned when read
0	SIUCSEL	Mask for echo back of IrDA 1 : Mask disabled 0 : Mask enabled (echo-back mode)

This register is used to specify whether the use of echo back function on IrDA transmission and reception is enabled.

20.3.16 SIUACTMSK_2 (0x0C00 000C)

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	RxDMSK	RTSMSK	DCDMSK	DTRMSK	Reserved	TxWRMSK
★ 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
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
7, 6	Reserved	0 is returned when read
5	RxDMSK	Mask for notification of change on RxD2 1 : Mask 0 : Unmask
4	RTSMSK	Mask for notification of change on RTS2# 1 : Mask 0 : Unmask
3	DCDMSK	Mask for notification of change on DCD2# 1 : Mask 0 : Unmask
2	DTRMSK	Mask for notification of change on DTR2# 1 : Mask 0 : Unmask
1	Reserved	Write 0 when write. 0 is returned when read
0	TxWRMSK	Mask for notification of transmit buffer write 1 : Mask 0 : Unmask

This register is used to set masks for notification of operation statuses to the Activity Timer of the SIU2.

When 1 is set in this register, state transition of the corresponding signals or write to transmit buffer is not notified to the Activity Timer.

20.3.17 SIUACTTMR_2 (0x0C00 000E)

Bit	7	6	5	4	3	2	1	0
Name	SIUTMO7	SIUTMO6	SIUTMO5	SIUTMO4	SIUTMO3	SIUTMO2	SIUTMO1	SIUTMO0
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
Other resets	0	0	0	0	0	0	0	0

Bit	Name	Function
7 to 0	SIUTMO(7:0)	SIU activity timeout period 11111111 : 255 x 30.5 μs 11111110 : 254 x 30.5 μs : 01111111 : 127 x 30.5 μs : 00000001 : 30.5 μs 00000000 : Activity Timer disabled

CHAPTER 21 LCD CONTROLLER

21.1 Overview

The VR4181 includes an LCD control module that operates on the MBA bus under the Unified Memory Architecture (UMA) conventions. The frame buffer resides in the main DRAM memory. This module supports an STN LCD panel.

21.1.1 LCD interface

The VR4181 LCD controller is a UMA based controller and uses a part of DRAM memory as a frame buffer. The LCD controller supports monochrome STN LCD panels having 4-bit data bus interfaces and color STN LCD panels having 8-bit data bus interfaces. When interfacing to a color LCD panel, GPIO pins must be allocated to provide the upper nibble of the 8-bit LCD data bus.

In monochrome mode, the LCD controller supports 1-bpp (bit per pixel) mode (mono), 2-bpp mode (4 gray levels) and 4-bpp mode (16 gray levels). In color mode, the LCD controller supports 4-bpp mode (16 colors) and 8-bpp mode (256 colors). The LCD controller includes a 256-entry x 18-bit color pallet. In color 8-bpp mode, the pallet is used to ★ select 256 colors out of possible 262,144 colors.

The LCD controller can support up to 320 x 320 pixels, and typical LCD panel horizontal/vertical resolutions are as follows.

Table 21-1. LCD Panel Resolutions (in Pixels, TYP.)

Horizontal resolution	Vertical resolution
320	320
320	240
320	160
240	320
240	240
240	160
160	320
160	240
160	160

The LCD controller also provides power-on and power-down sequence control for the LCD panel via the VPLCD pin, which is for LCD logic power control, and VPBIAS pin, which is for LCD bias power control. Power sequencing is provided to prevent latch-up damage to the panel.

The LCD controller may be disabled to allow connection of an external LCDC with integrated frame buffer RAM such as NEC Electronics' μ PD16661. When the internal LCD controller is disabled by setting the LCDGPMODE register in the GIU, the SHCLK, LOCLK, VPLCD, and VPBIAS pins are redefined as follows:

Table 21-2. Redefining LCD Interface Pins When LCD Controller Is Disabled

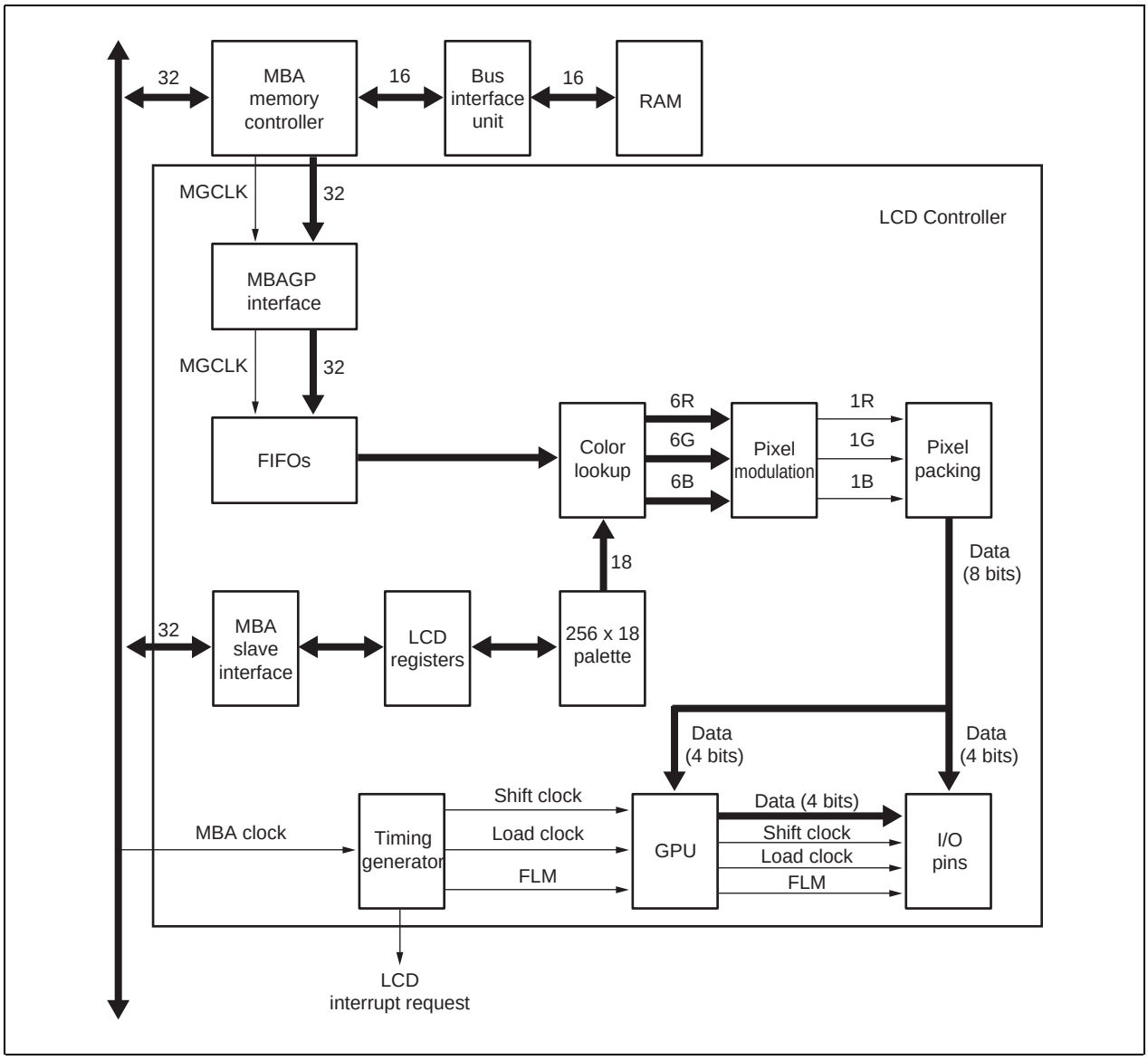
Redefined function	Default function
LCDCS#	SHCLK
MEMCS16#	LOCLK
VPGPIO1	VPLCD
VPGPIO0	VPBIAS

21.2 LCD Module Features

- Resolutions
  - Horizontal: Up to 320 pixels (the number of pixels must be multiplies of 8)
  - Vertical: Up to 320 pixels
  - Color: 4 bpp, 8 bpp (up to 256 colors)
  - Monochrome: 1 bpp, 2 bpp, 4 bpp (up to 16 gray scale)
  - Color Palette: 18 bits
- High vertical refresh rates for flicker-free LCD frame modulation

The following is a block diagram of the LCD controller.

Figure 21-1. LCD Controller Block Diagram



The LCD controller is a slave module of the MBA bus. Its registers can be accessed via the MBA slave interface. The frame data are read from main memory via the memory controller and the MBAGP (MBA Graphic port).

21.3 LCD Controller Specification

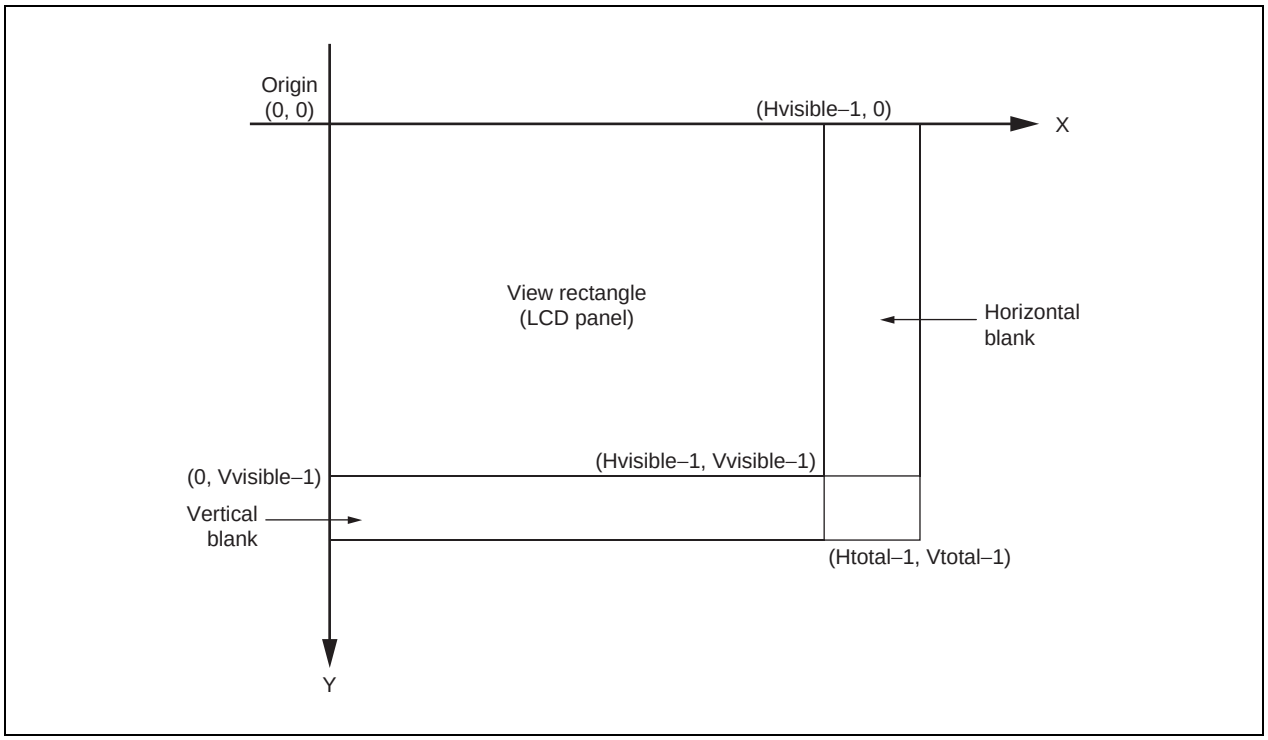
21.3.1 Panel configuration and interface

(1) View rectangle and horizontal/vertical blank

Most parameters of the LCD controller are described using a coordinate system. The x coordinate increases as a point moves to the right. The y coordinate increases as a point moves down. The origin is (0, 0).

The size of the bounding box is specified by Vtotal and Htotal. The point (Vtotal–1, Htotal–1) is the box's lower right corner and includes the horizontal and vertical blanks. Vvisible and Hvisible define the view rectangle, and outside of the view rectangle are the horizontal blank and vertical blank.

Figure 21-2. View Rectangle and Horizontal/Vertical Blank



Each parameter is defined using bit values in the LCD controller registers as follows:

- Vtotal = Vtot(8:0) VRTOTALREG (0x0A00 0408)
- Vvisible = Vact(8:0) VRVISIBREG (0x0A00 040A)
- Htotal = Htot(7:0) x 2 HRTOTALREG (0x0A00 0400)
- Hvisble = Hact(5:0) x 8 HRVISIBREG (0x0A00 0402)

Caution The following expressions must be satisfied.

1. Vtotal ≥ Vvisible

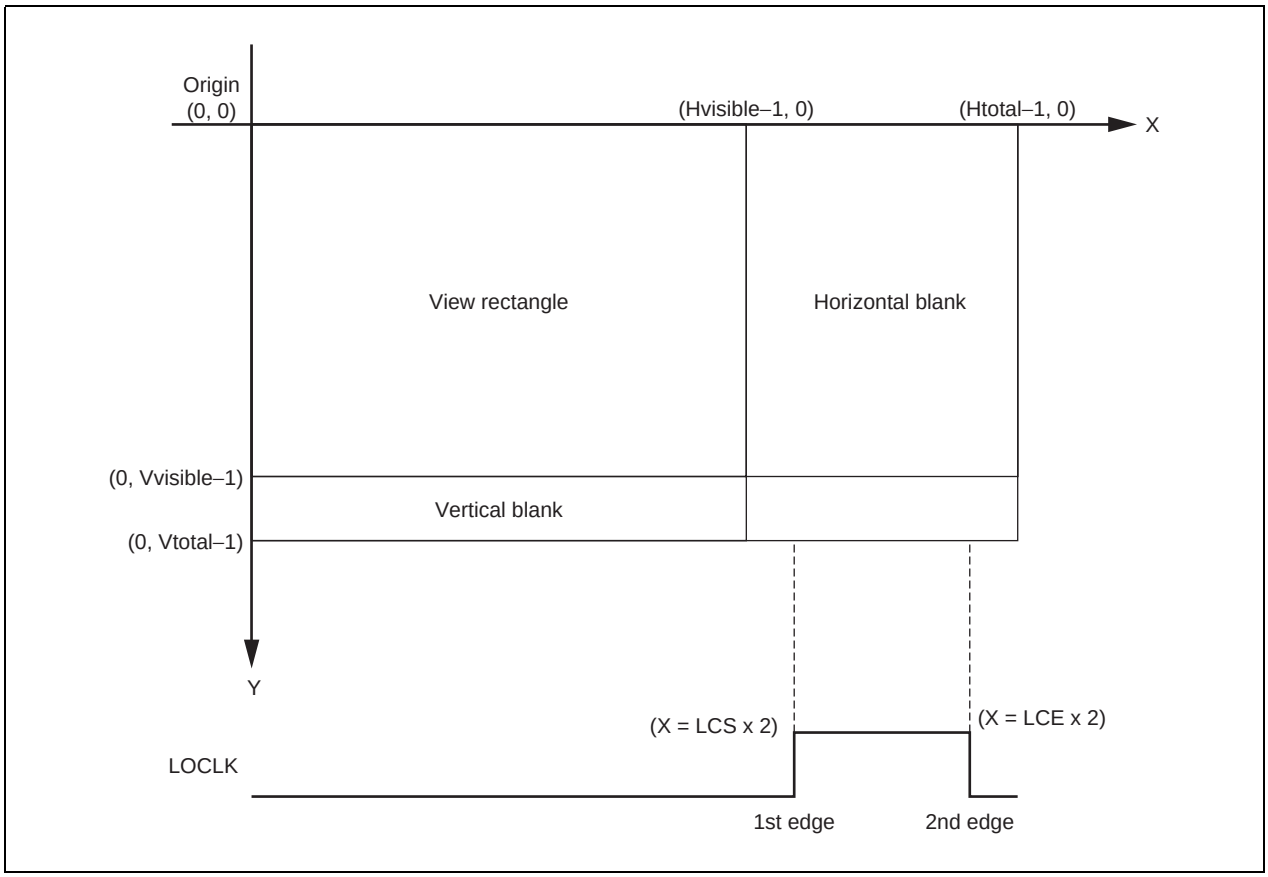
2. Htotal ≥ Hvisible + 6

(2) Load clock

The edge positions of the load clock, LOCLK, are programmable. Each row in the rectangle specified with (0, 0) and (Htotal–1, Vvisible–1) must have two LOCLK edges. The remaining rows in the frame rectangle form the vertical blank. These rows also have two LOCLK edges if the DummyL bit of the VRVISIBREG register is 1, or none if DummyL bit is 0. The first LOCLK edge is defined by the LCS(7:0) bits of the LDCLKSTREG register. The second edge is defined by the LCE(7:0) bits of the LDCLKENDREG register, and is usually inside the horizontal blank.

The LPPOL bit of the LCDCTRLREG register controls the directions of toggles. If the LPPOL bit is 0, the first LOCLK edge is positive and the second is negative. If the LPPOL bit is 1, the reverse is true.

Figure 21-3. Position of Load Clock (LOCLK)



Caution The following expression must be satisfied.

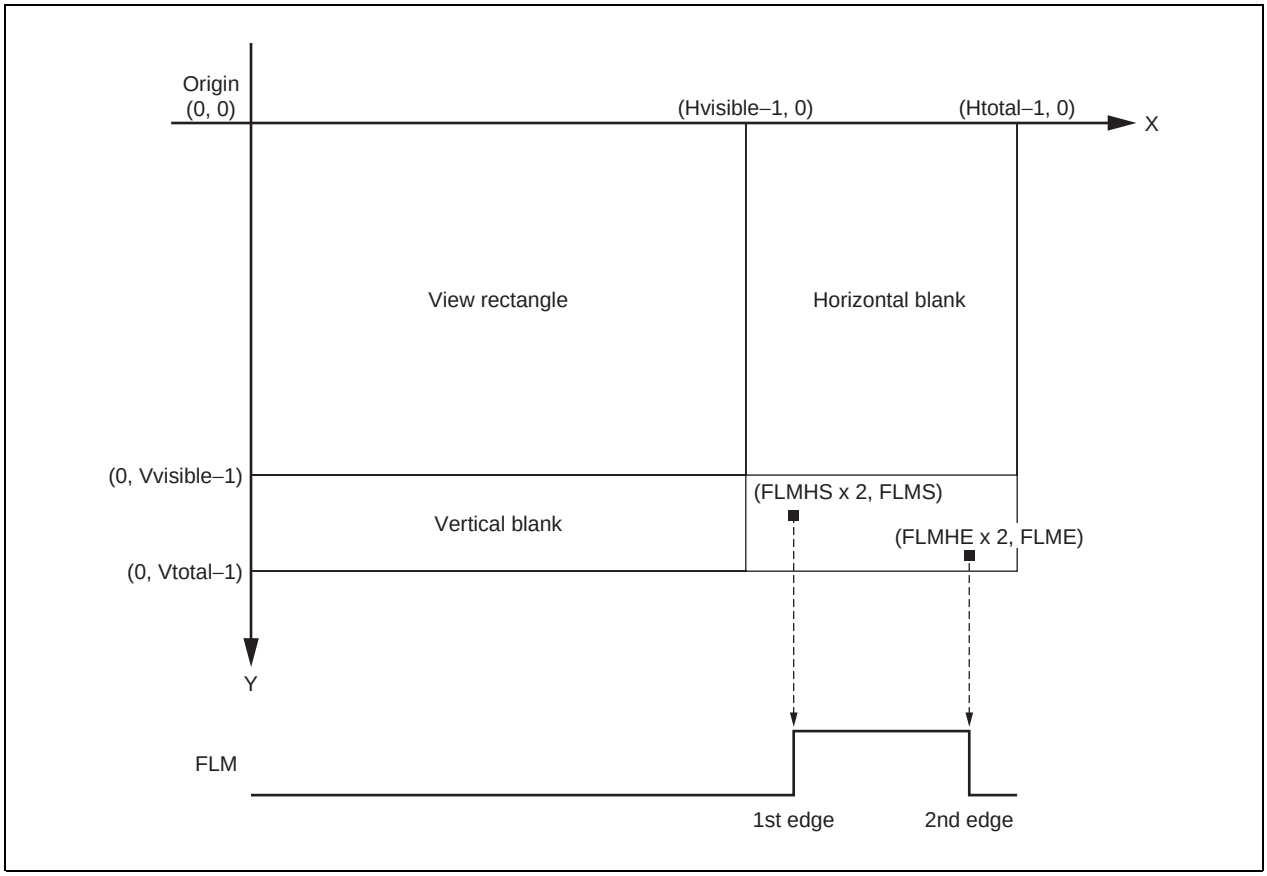
1. Htotal > LCE(7:0) x 2 > LCS(7:0) x 2

(3) Frame clock

The edge positions of the frame clock, FLM, are also programmable. There must be exactly two FLM edges inside the bounding box. The first FLM edge is defined by the FLMHS(7:0) bits of the FHSTARTREG register and the FLMS(8:0) bits of the FVSTARTREG register. The location of the first edge is at (FLMHS x 2, FLMS). The second FLM edge is defined by the FLMHE(7:0) bits of the FHENDREG register and the FLME(8:0) bits of the FVENDREG register. The location of second edge is at (FLMHE x 2, FLME).

If the FLMPOL bit of the LCDCTRLREG register is 0, the first FLM edge is positive and the second is negative. If the FLMPOL bit is 1, the reverse is true.

Figure 21-4. Position of Frame Clock (FLM)



Caution The following expressions must be satisfied.

1. **Htotal > FLMHE(7:0) x 2 > FLMHS(7:0) x 2**
2. **Vtotal > FLME(8:0), Vtotal > FLMS(8:0)**

(4) Shift clock

The shift clock (SHCLK) edges can be programmed only indirectly. The SHCLK is also output in rows of the vertical blank if the DummyL bit of the VRVISIBREG register is 1. The position of SHCLK edges are controlled by the Panelcolor and PanDbus bits of the LCDCFGREG0 register. The SCLKPOL bit of the LCDCTRLREG register determines whether data is latched into the panel on the rising or falling edges. If the SCLKPOL bit is 0, data is latched on the falling edges.

(5) M signal

Some panels also need a modulation signal, M, to operate properly. The modulation rate is controlled by MOD(7:0) bits of the LCDCFGREG0 register. If the MOD field is 0, the M signal toggles once per frame. If the MOD field is not 0, then the M signal toggles once every rows set in the MOD field. The M signal toggles at the position specified in the LCE field, the same time as the second LOCLK edge. When the MOD field is 0, the M signal toggles when the LOCLK latches the FLM.

(6) Vertical retrace interrupt

When the LCD controller goes through the vertical blank, a status signal bit VIReq of the LCDINRQREG register becomes 1. This signal can be configured to be polled or to generate an interrupt request. To enable the interrupt, set the MVIReq bit of the LCDIMSKREG register to 1. Once an interrupt request is generated, writing to the VIReq bit clears the interrupt request. However, the state of the VIReq bit changes to 0 only after the controller returns to top left corner. Note that there is some delay between the controller's entering or leaving the vertical blank and the changes in the VIReq bit.

21.3.2 Controller clocks

All LCD controller timing is based on the internal clock hpck. The hpck is derived from the gclk, which is derived from the MBA clock (TClock). The frequency of gclk can be equal to, one-half of, or one-quarter of that of the MBA clock, depending on the Pre-scal(1:0) bits of the LCDCFGREG0 register and the MBA clock frequency. The hpck frequency is programmable. In each cycle the hpck is at high level for cycles set in the HpckH(5:0) bits of the LCDCFGREG1 register, and at low level for cycles set in the HpckL(5:0) bits of the LCDCFGREG1 register. The values in HpckH and HpckL fields are not arbitrary. Their sum must be at least 5, and the following condition must be satisfied:

$$f\text{-}hpck \approx Htotal \times Vtotal \times f\text{-}refresh$$

Both the hpck and the gclk can be turned off when the panel is inactive. Setting the ContCkE bit of the LCDCTRLREG register to 1 initializes the LCD controller and turns on both clocks, or 0 turns them off.

21.3.3 Palette

The Col(1:0) bits of the LCDCFGREG0 register indicate the desired color depth. If they are set to 0, then a monochrome image can be displayed on a monochrome panel. If they are set 1, then a 4-shade gray scale image can be displayed on a monochrome panel. If they are set to 2 or 3, then a palette is enabled, and a color panel can be used.

The palette has 256 entries. Each entry has 18 bits and is 6-6-6 format for the RGB color. To access an entry first store its index in the PalIndex(7:0) bits of the CPINDCTREG register, then read from or write to the PalData(17:0) bits of the CPALDATREG register. To accelerate continuous accesses, the PalRDI bit or the PalWRI bit of the CPINDCTREG register can be set to 1. When the PalRDI bit is set to 1, the LCD controller automatically adds 1 to the PalIndex(7:0) bits of the CPINDCTREG register after reading from the PalData field; when the PalWRI bit is set to 1, the LCD controller automatically adds 1 to the PalIndex(7:0) bits after writing to the PalData field.

If the Col field is set to 2, then the pixel data provides only the lower half of the palette index. The upper half is provided by the PalPage(3:0) bits of the CPINDCTREG register. Together they specify one entry in the palette.

Finally, the hpck and the gclk must be turned on before the palette is accessed.

21.3.4 Frame buffer memory and FIFO

The frame buffer is linear and supports a packed pixel format. The length of a scan line must be a multiple of 32. The last double word of a scan line need not be completely filled. The pixels are stored in double words. The data format of each double word depends on the color depth, as shown in the following table.

Bit 31																															Bit 0
18	19	1A	1B	1C	1D	1E	1F	10	11	12	13	14	15	16	17	08	09	0A	0B	0C	0D	0E	0F	00	01	02	03	04	05	06	07
0C		0D		0E		0F		08		09		0A		0B		04		05		06		07		00		01		02		03	
06				07				04				05				02				03				00				01			
03								02								01								00							

The frame buffer memory starts from the 32-bit address specified by the FBSA(31:0) bits of the FBSTADREG1 and FBSTADREG2 registers, and ends at the address specified by the FBEA(31:0) bits of the FBENDADREG1 and FBENDADREG2 registers. The FBEA field does not necessarily show where the last pixel is stored; but it is the address of the first 32-byte page boundary that follows the memory location where the last pixel is stored, starting from the address set in the FBSA field. For example, if FBSA field is 0x0B00 0408, and the frame buffer occupies 235 bytes, then the FBEA field is 0x0B00 0508 (FBSA plus the ceiling of 235/32).

Data from the frame buffer is burst into the FIFO to conserve memory bandwidth. Each burst transfers 32 bytes. The FIFO is divided into three arrays, and each burst fills exactly one array. Bursts can not cross array borders, nor can read from and write to the same array at the same time. When the memory bandwidth is low, the FIFO bursts only when there are only the number of double words left to be read that is displayed in the FIFOC(2:0) bits of the LCDCTRLREG register. If the burst is not fast enough in relation to the refresh rate of the panel image, irreversible image degradation occurs due to a lack of data to be displayed, and an interrupt request is generated. This interrupt request can be polled from the FIFOOVERR bit of the LCDINRQREG register. It can be cleared only by stopping and then restarting controller clocks. Because image degradation is a serious problem, the value set to the FIFOC field should be carefully selected during development.

21.3.5 Panel power ON/OFF sequence

Some panels use several power supplies, and these supplies and interface logic signals must be turn on or off in sequences specified by the manufacturers. The LCD controller has signals to control these power supplies.

Each power supply is controlled via the VPBIAS or VPLCD pin. These pins are connected to a pull-up or pull-down resistor in addition to the power supply. When the power is off, these pins are placed into high impedance, so that the resistor pulls the power supply on/off input to the off state.

The power-on/off sequence is started by setting the PowerC bit of the PWRCONREG2 register. Setting this bit to 1 starts the power-on sequence. In the power-on sequence the power supply control pins are brought out of high impedance to a programmed state at a programmed time, and the panel interface signals become active at a programmed time. The following table lists the control pins and the programming register bits.

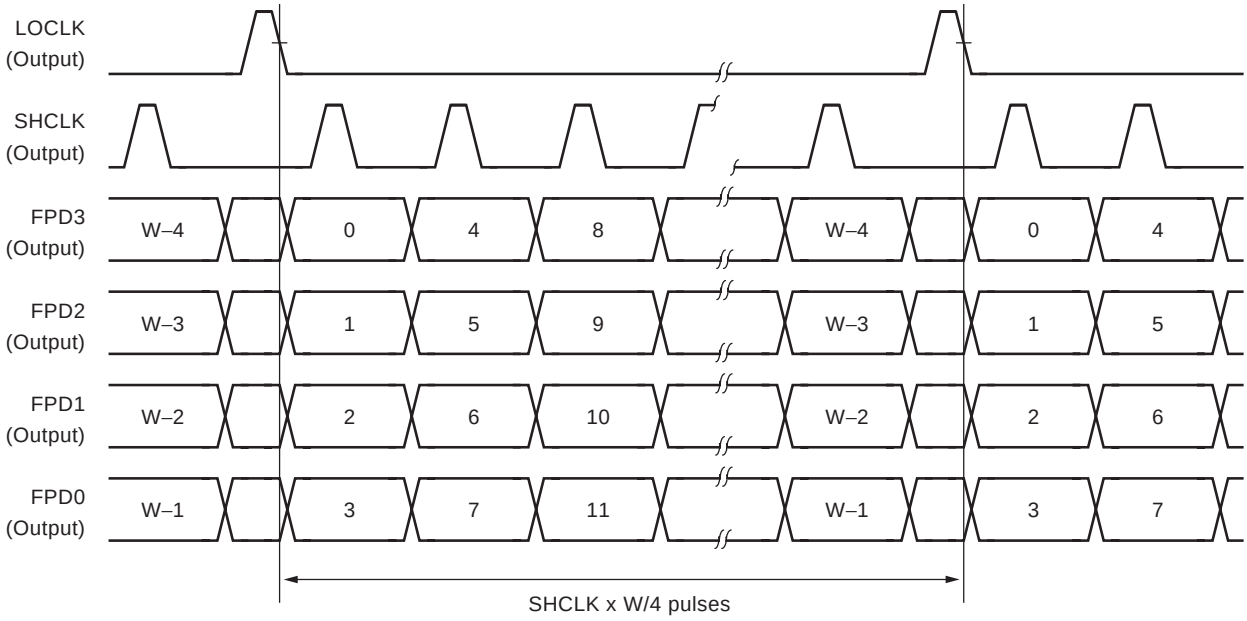
Pin	Power-on time bit	Power-on state bit
VPBIAS	Biason(4:0)	BiasCon
VPLCD	Vccon(4:0)	VccC
LCD Interface	I/Fon(4:0)	– (active)

★ For example, storing 1 in the BiasCon bit and 3 in the Biason field brings the LCD controller to make the VPBIAS signal from high impedance to high level three frames after the PowerC bit is set to 1, not counting the frame in which the PowerC bit is changed.

Setting the PowerC bit to 0 starts the power-off sequence. In the power-off sequence the control pins are put into high impedance, so that the power supply is turned off. The pins enter high impedance in the reverse order of the power-on sequence, but the time delay between the two control pins remains the same.

21.3.6 Operation of LCD controller

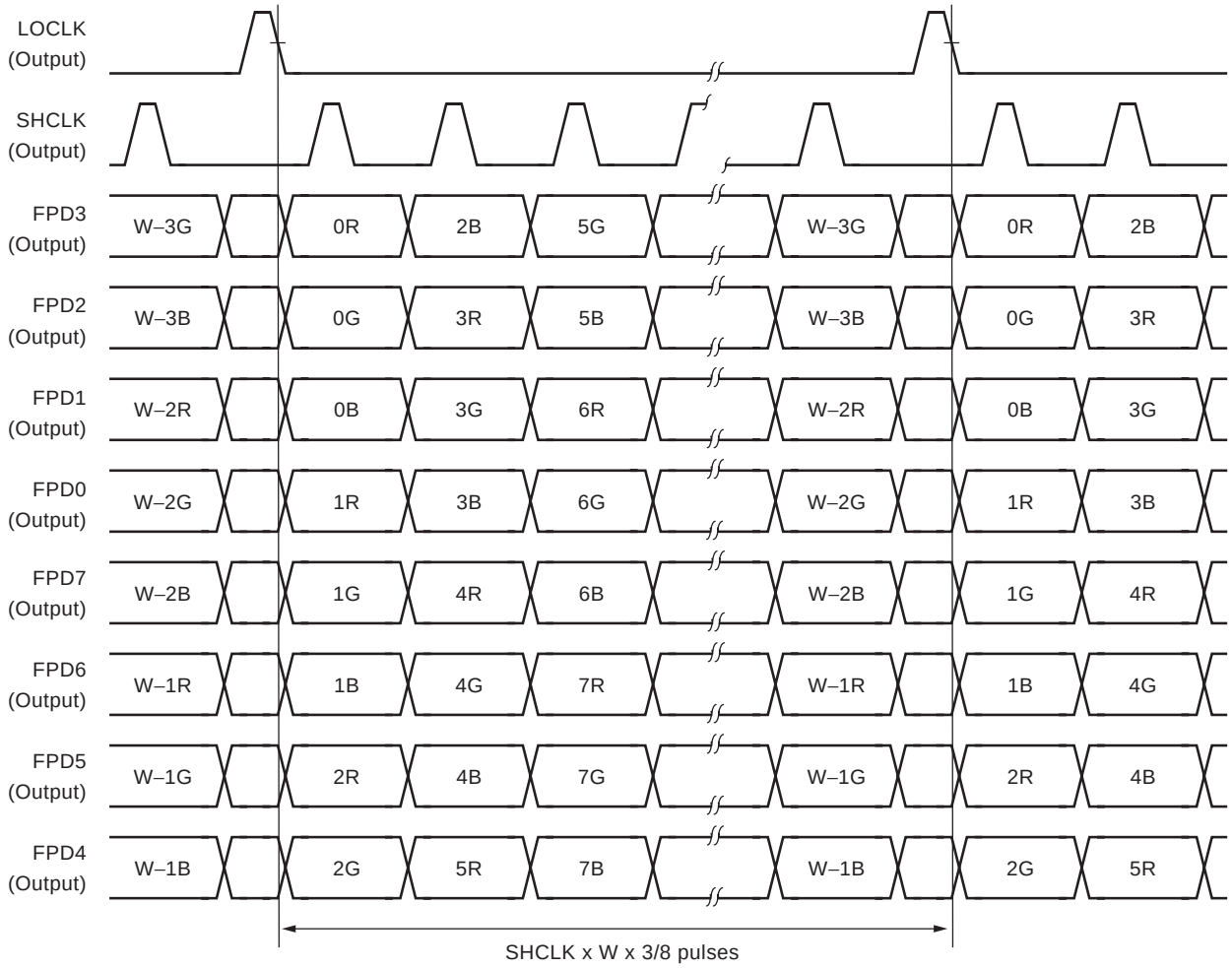
Figure 21-5. Monochrome Panel



Remark W: panel width (Hact(5:0) x 8)

The polarity (order of rising and falling edges) of the LOCLK and the SHCLK are programmable via the LPPOL and SCLKPOL bits.

Figure 21-6. Color Panel in 8-Bit Data Bus

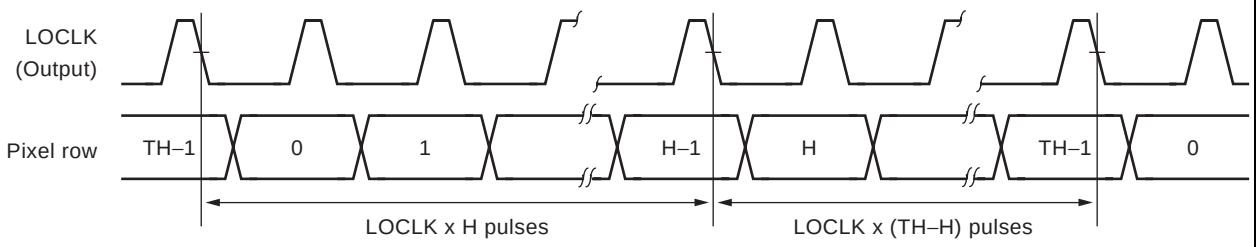


Remark W: panel width (Hact(5:0) x 8)

The polarity (order of rising and falling edges) of the LOCLK and the SHCLK are programmable via the LPPOL and SCLKPOL bits.

Remark In the color 8-bit data bus mode, FPD(3:0) are for upper 4 bits of the LCD data bus, and FPD(7:4) are for lower 4 bits of the LCD data bus.

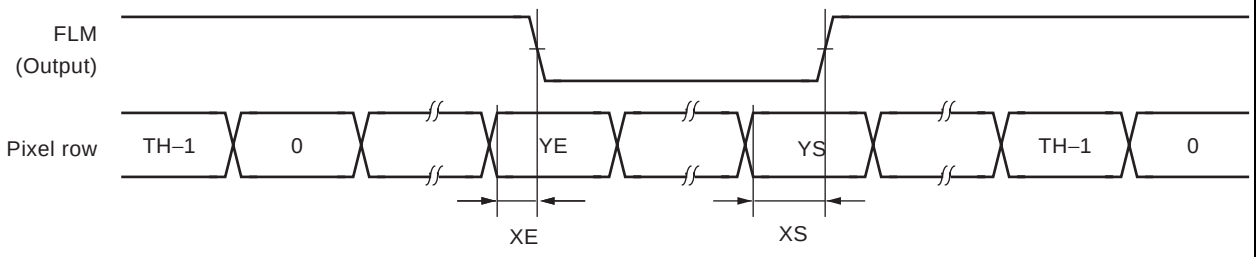
Figure 21-7. Load Clock (LOCLK)



Remark H: panel height (Vact)
TH: panel height + dummy lines (Vtotal)

Remark Dummy lines are inserted when needed. For example, some panels can display only 240 lines, but has 242 line cycles. Load clock can be deactivated during the dummy lines (see DummyL bit description in **21.4.6**).

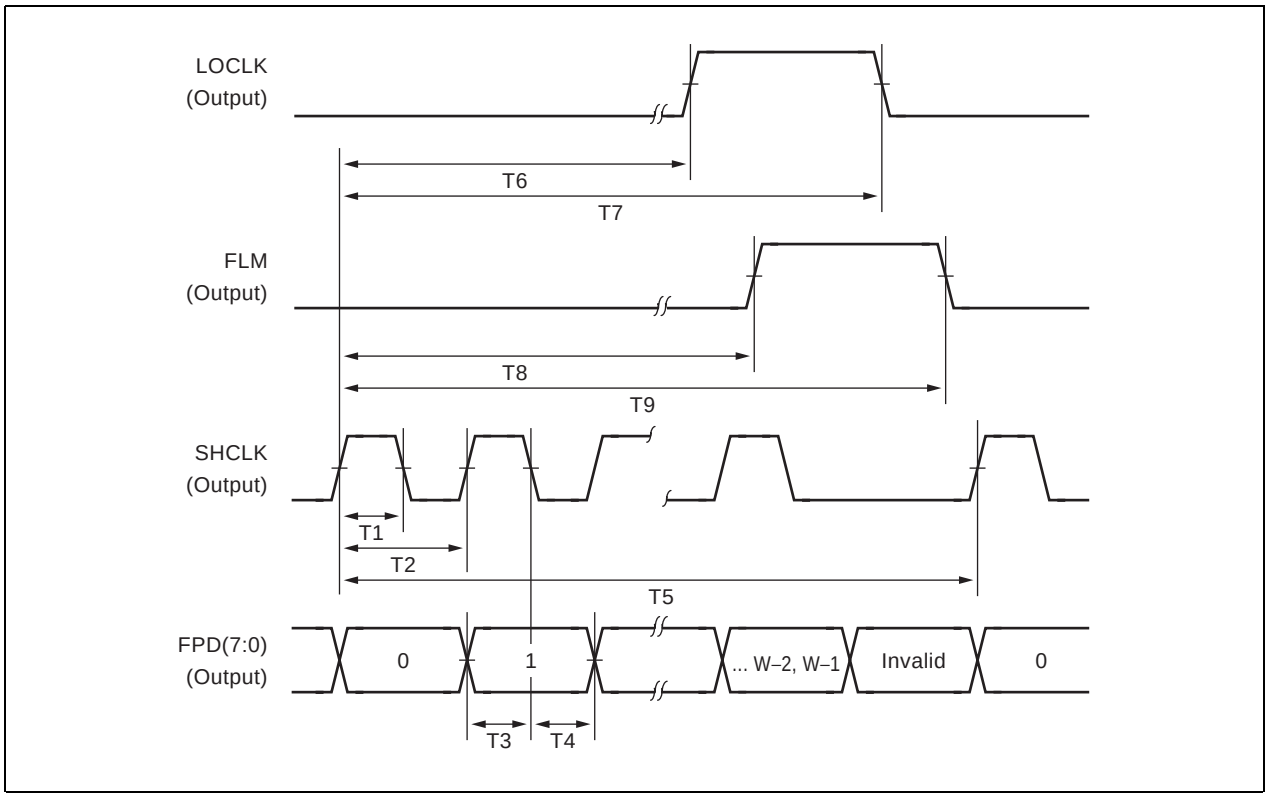
Figure 21-8. Frame Clock (FLM)



Remark YS: Y-Coordinates of the second FLM edge (FLMS)
YE: Y-Coordinates of the first FLM edge (FLME)

The polarity (order of rising and falling edges) of the FLM is programmable via the FLMPOL bit.

Figure 21-9. LCD Timing Parameters



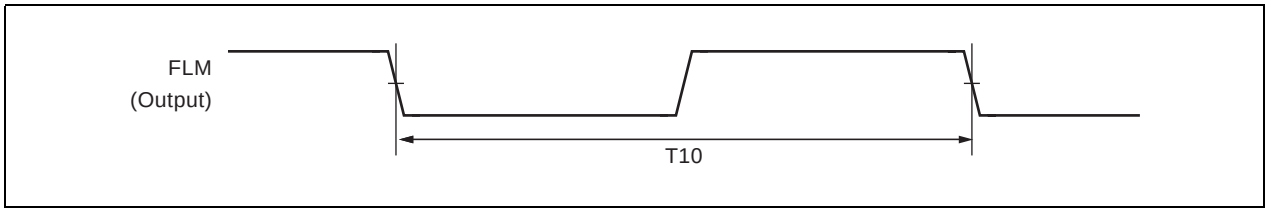
The polarity of the FLM is programmable through the FLMPOL bit. In this diagram the first edge is a rising edge.

The two FLM edges are on the same row in this diagram, but they need not be.

★ The active edge of the LOCLK is programmable through the LPPOL bit. In this diagram, the first edge is a rising edge (the falling edge is the active edge).

★ The polarity of the SHCLK is programmable through the SCLKPOL bit. In this diagram, the first edge is a rising edge (the falling edge is the active edge).

Figure 21-10. FLM Period



The definitions of parameters shown in the figures are given in the table below.

Table 21-3. LCD Controller Parameters

Symbol	Definition
Tg	gclk period This parameter is not one of the timing parameters, but all timing parameters is calculated based on this. gclk is controlled by the Pre-scal field. Tg = 1 / (frequency of gclk)
T1	Shift clock high level width Color: T1 = Tg x HpckH 4-bit bus monochrome: T1 = Tg x (HpckH + HpckL)
T2	Shift clock cycle Color: T2 = Tg x (HpckH + HpckL) 4-bit bus monochrome: T2 = Tg x (HpckH + HpckL) x 2
T3	Panel data setup time Color : T3 = Tg x HpckH 4-bit bus monochrome: T3 = Tg x (HpckH + HpckL)
T4	Panel data hold time Color: T4 = Tg x HpckL 4-bit bus monochrome: T4 = Tg x (HpckH + HpckL)
T5	Row cycle time T5 = Tg x (HpckH + HpckL) x Htot
T6	Load clock start time T6 = Tg x (HpckH + HpckL) x LCS
T7	Load clock end time T7 = Tg x (HpckH + HpckL) x LCE
T8	FLM horizontal start time T8 = Tg x (HpckH + HpckL) x FLMHS
T9	FLM horizontal end time T9 = Tg x (HpckH + HpckL) x FLMHE
T10	Panel frame period T10 = Tg x (HpckH + HpckL) x Htot x Vtot

21.4 Register Set

Table 21-4. LCD Controller Registers

Physical address	R/W	Register symbol	Function
0x0A00 0400	R/W	HRTOTALREG	Horizontal total register
0x0A00 0402	R/W	HRVISIBREG	Horizontal visible register
0x0A00 0404	R/W	LDCLKSTREG	Load clock start register
0x0A00 0406	R/W	LDCLKENDREG	Load clock end register
0x0A00 0408	R/W	VRTOTALREG	Vertical total register
0x0A00 040A	R/W	VRVISIBREG	Vertical visible register
0x0A00 040C	R/W	FVSTARTREG	FLM vertical start register
0x0A00 040E	R/W	FVENDREG	FLM vertical end register
0x0A00 0410	R/W	LCDCTRLREG	LCD control register
0x0A00 0412	R/W	LCDINRQREG	LCD interrupt request register
0x0A00 0414	R/W	LCDCFGREG0	LCD configuration register 0
0x0A00 0416	R/W	LCDCFGREG1	LCD configuration register 1
0x0A00 0418	R/W	FBSTADREG1	Frame buffer start address 1 register
0x0A00 041A	R/W	FBSTADREG2	Frame buffer start address 2 register
0x0A00 0420	R/W	FBENDADREG1	Frame buffer end address 1 register
0x0A00 0422	R/W	FBENDADREG2	Frame buffer end address 2 register
0x0A00 0424	R/W	FHSTARTREG	FLM horizontal start register
0x0A00 0426	R/W	FHENDREG	FLM horizontal end register
0x0A00 0430	R/W	PWRCONREG1	Power control register 1
0x0A00 0432	R/W	PWRCONREG2	Power control register 2
0x0A00 0434	R/W	LCDIMSKREG	LCD interrupt mask register
0x0A00 047E	R/W	CPINDCTREG	Color palette index and control register
0x0A00 0480	R/W	CPALDATREG	Color palette data register (32 bits wide)

21.4.1 HRTOTALREG (0x0A00 0400)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Htot7	Htot6	Htot5	Htot4	Htot3	Htot2	Htot1	Htot0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 8	Reserved	0 is returned when read
7 to 0	Htot(7:0)	Number of horizontal total columns. Set this register to a value one half of the horizontal total. Horizontal total = horizontal visible width + horizontal blank

21.4.2 HRVISIBREG (0x0A00 0402)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Hact5	Hact4	Hact3	Hact2	Hact1	Hact0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 6	Reserved	0 is returned when read
5 to 0	Hact(5:0)	Number of horizontal visible pixels. Set this register to a value one eighth of the horizontal visible pixels.

21.4.3 LDCLKSTREG (0x0A00 0404)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	LCS7	LCS6	LCS5	LCS4	LCS3	LCS2	LCS1	LCS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 8	Reserved	0 is returned when read
7 to 0	LCS(7:0)	X coordinate of the first edge of the LOCLK. Set this register to a value one half of the first edge of the LOCLK.

21.4.4 LDCLKENDREG (0x0A00 0406)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	LCE7	LCE6	LCE5	LCE4	LCE3	LCE2	LCE1	LCE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 8	Reserved	0 is returned when read
7 to 0	LCE(7:0)	X coordinate of the second edge of the LOCLK. Set this register to a value one half of the second edge of the LOCLK.

21.4.5 VRTOTALREG (0x0A00 0408)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Vtot8
R/W	R	R	R	R	R	R	R	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Vtot7	Vtot6	Vtot5	Vtot4	Vtot3	Vtot2	Vtot1	Vtot0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 9	Reserved	0 is returned when read
8 to 0	Vtot(8:0)	Vertical total number of lines including vertical retrace period

21.4.6 VRVISIBREG (0x0A00 040A)

Bit	15	14	13	12	11	10	9	8
Name	DummyL	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Vact8
R/W	R/W	R	R	R	R	R	R	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Vact7	Vact6	Vact5	Vact4	Vact3	Vact2	Vact1	Vact0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15	DummyL	Dummy line inserting position 0 : Immediately before vertical blank 1 : Anywhere in vertical blank
14 to 9	Reserved	0 is returned when read
8 to 0	Vact(8:0)	Vertical visible number of lines

21.4.7 FVSTARTREG (0x0A00 040C)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	FLMS8
R/W	R	R	R	R	R	R	R	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	FLMS7	FLMS6	FLMS5	FLMS4	FLMS3	FLMS2	FLMS1	FLMS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 9	Reserved	0 is returned when read
8 to 0	FLMS(8:0)	Y coordinate of the first FLM edge

21.4.8 FVENDREG (0x0A00 040E)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	FLME8
R/W	R	R	R	R	R	R	R	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	FLME7	FLME6	FLME5	FLME4	FLME3	FLME2	FLME1	FLME0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 9	Reserved	0 is returned when read
8 to 0	FLME(8:0)	Y coordinate of the second FLM edge

21.4.9 LCDCTRLREG (0x0A00 0410)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	FIFOC2	FIFOC1	FIFOC0	Reserved	ContCkE	LPPOL	FLMPOL	SCLKPOL
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 8	Reserved	0 is returned when read
7 to 5	FIFOC(2:0)	FIFO control. A FIFO transfer is performed when only the number of double words set here is left in the FIFO.
4	Reserved	0 is returned when read
3	ContCkE	LCD controller clock enable 0 : OFF 1 : ON
2	LPPOL	LOCLK clock polarity 0 : Leading edge is rising 1 : Leading edge is falling
1	FLMPOL	FLM clock polarity 0 : Leading edge is rising 1 : Leading edge is falling
0	SCLKPOL	Shift clock polarity 0 : Leading edge is rising (active edge is falling) 1 : Leading edge is falling (active edge is rising)

★

21.4.10 LCDINRQREG (0x0A00 0412)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	VIReq	FIFOOV ERR	Reserved
R/W	R	R	R	R	R	R/W	R/W	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 3	Reserved	0 is returned when read
2	VIReq	Vertical retrace interrupt request 0 : No request (outside vertical blank) 1 : Requested (vertical blank)
1	FIFOOVERR	FIFO overrun interrupt request 0 : No request 1 : Requested
0	Reserved	0 is returned when read

21.4.11 LCDCFGREG0 (0x0A00 0414)

Bit	15	14	13	12	11	10	9	8
Name	MOD7	MOD6	MOD5	MOD4	MOD3	MOD2	MOD1	MOD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Softreset	Reserved	Pre-scal1	Pre-scal0	Col1	Col0	Panelcolor	PanDbus
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 8	MOD(7:0)	LCD M signal configuration. These bits specify the number of lines between M toggles. 0 : Once per frame 1 : After every line 2 : After every 2 lines : 255 : After every 255 lines
7	Softreset	Software reset for LCD controller. The software reset is active only in test mode. 0 : Normal operation 1 : Reset
6	Reserved	0 is returned when read
5, 4	Pre-scal(1:0)	gclk (clock for LCD controller) pre-scalar mode to the MBA clock 00 : Divide by 1 01 : Divide by 2 10 : Divide by 4 11 : RFU
3, 2	Col(1:0)	Color depth selection 00 : 1 bit (black and white for monochrome panel) 01 : 2 bits (4 gray scale for monochrome panel) 10 : 4 bits (16 gray scale for monochrome or 16 colors for color panel) 11 : 8 bits (256 colors for color panel)
1	Panelcolor	Color/monochrome selection 0 : Color 1 : Monochrome
0	PanDbus	Panel data width 0 : 4 bits 1 : 8 bits (for dual scan panel or for 8-bit high scan)

★ **Remark** In the 4 bpp mode (16 gray scale) for monochrome panels, the Blue area of the color palette is used for displaying. The palette is not used in the other modes (1 bpp and 2 bpp) for monochrome panels.

21.4.12 LCDCFGREG1 (0x0A00 0416)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	HpckL5	HpckL4	HpckL3	HpckL2	HpckL1	HpckL0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	HpckH5	HpckH4	HpckH3	HpckH2	HpckH1	HpckH0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15, 14	Reserved	0 is returned when read
13 to 8	HpckL(5:0)	Number of gclk cycles for hpck low level width
7, 6	Reserved	0 is returned when read
5 to 0	HpckH(5:0)	Number of gclk cycles for hpck high level width

21.4.13 FBSTADREG1 (0x0A00 0418)

Bit	15	14	13	12	11	10	9	8
Name	FBSA15	FBSA14	FBSA13	FBSA12	FBSA11	FBSA10	FBSA9	FBSA8
R/W	R/w	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	FBSA7	FBSA6	FBSA5	FBSA4	FBSA3	FBSA2	FBSA1	FBSA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 0	FBSA(15:0)	Frame buffer start address (lower 16 bits)

Caution FBSA(2:0) bits must be cleared to 0.

21.4.14 FBSTADREG2 (0x0A00 041A)

Bit	15	14	13	12	11	10	9	8
Name	FBSA31	FBSA30	FBSA29	FBSA28	FBSA27	FBSA26	FBSA25	FBSA24
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	FBSA23	FBSA22	FBSA21	FBSA20	FBSA19	FBSA18	FBSA17	FBSA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 0	FBSA(31:16)	Frame buffer start address (upper 16 bits) FBSA(31:29) are always 0 when read.

The FBSTADREG1 and FBSTADREG2 registers are used to specify the frame buffer starting address. The frame buffer is linear and the pixels are packed. This address corresponds to the first, top left pixel of the screen.

21.4.15 FBENDADREG1 (0x0A00 0420)

Bit	15	14	13	12	11	10	9	8
Name	FBEA15	FBEA14	FBEA13	FBEA12	FBEA11	FBEA10	FBEA9	FBEA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	FBEA7	FBEA6	FBEA5	FBEA4	FBEA3	FBEA2	FBEA1	FBEA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 0	FBEA(15:0)	Frame buffer end address (lower 16 bits)

21.4.16 FBENDADREG2 (0x0A00 0422)

Bit	15	14	13	12	11	10	9	8
Name	FBEA31	FBEA30	FBEA29	FBEA28	FBEA27	FBEA26	FBEA25	FBEA24
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	FBEA23	FBEA22	FBEA21	FBEA20	FBEA19	FBEA18	FBEA17	FBEA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 0	FBEA(31:16)	Frame buffer end address (upper 16 bits) FBEA(31:29) are always 0 when read.

21.4.17 FHSTARTREG (0x0A00 0424)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	FLMHS7	FLMHS6	FLMHS5	FLMHS4	FLMHS3	FLMHS2	FLMHS1	FLMHS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 8	Reserved	0 is returned when read
7 to 0	FLMHS(7:0)	X coordinate of the first FLM edge. Set this register to a value one half of the first edge of FLM.

21.4.18 FHENDREG (0x0A00 0426)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	FLMHE7	FLMHE6	FLMHE5	FLMHE4	FLMHE3	FLMHE2	FLMHE1	FLMHE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 8	Reserved	0 is returned when read
7 to 0	FLMHE(7:0)	X coordinate of the second FLM edge. Set this register to a value one half of the second edge of FLM.

21.4.19 PWRCONREG1 (0x0A00 0430)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Biason4	Biason3	Biason2	Biason1	Biason0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15	Reserved	0 is returned when read
14 to 5	Reserved	Write 0 when write. 0 is returned when read
4 to 0	Biason(4:0)	Frame at which the bias voltage is turned on

21.4.20 PWRCONREG2 (0x0A00 0432)

Bit	15	14	13	12	11	10	9	8
Name	Testmode	VccC	Reserved	Reserved	BiasCon	PowerC	I/Fon4	I/Fon3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	I/Fon2	I/Fon1	I/Fon0	Vccon4	Vccon3	Vccon2	Vccon1	Vccon0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15	Testmode	Test mode enable 0 : Normal operation 1 : Enters test mode
14	VccC	Vcc (VPLCD) signal polarity control 0 : Active low 1 : Active high
13, 12	Reserved	Write 0 when write. 0 is returned when read
11	BiasCon	Bias (VPBIAS) signal polarity control 0 : Active low 1 : Active high
10	PowerC	Power control 0 : Off 1 : On
9 to 5	I/Fon(4:0)	Frame at which the panel logic interface signals are turned on
4 to 0	Vccon(4:0)	Frame at which the panel Vcc is turned on

21.4.21 LCDIMSKREG (0x0A00 0434)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	MVIReq	MFIFO OVERR	Reserved
R/W	R	R	R	R	R	R/W	R/W	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 3	Reserved	0 is returned when read
2	MVIReq	Vertical retrace interrupt mask 0 : Mask 1 : Unmask
1	MFIFOOVERR	FIFO overrun interrupt mask 0 : Mask 1 : Unmask
0	Reserved	0 is returned when read

21.4.22 CPINDCTREG (0x0A00 047E)

Bit	15	14	13	12	11	10	9	8
Name	PalPage3	PalPage2	PalPage1	PalPage0	Reserved	Reserved	PalRDI	PalWRI
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	PalIndex7	PalIndex6	PalIndex5	PalIndex4	PalIndex3	PalIndex2	PalIndex1	PalIndex0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15 to 12	PalPage(3:0)	Palette page select used in 4 bpp mode
11, 10	Reserved	0 is returned when read
9	PalRDI	Palette index read status 0 : No change after read 1 : Incremented by 1 after read
8	PalWRI	Palette index write status 0 : No change after write 1 : Incremented by 1 after write
7 to 0	PalIndex(7:0)	Palette index

★ **Remark** In the 4 bpp mode (16 gray scale) for monochrome panels, the Blue area of the color palette is used for displaying. The palette is not used in the other modes (1 bpp and 2 bpp) for monochrome panels.

21.4.23 CPALDATREG (0x0A0 0480)

Bit	31	30	29	28	27	26	25	24
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	PalData17	PalData16
R/W	R	R	R	R	R	R	R/W	R/W
Reset	0	0	0	0	0	0	Undefined	Undefined

Bit	15	14	13	12	11	10	9	8
Name	PalData15	PalData14	PalData13	PalData12	PalData11	PalData10	PalData9	PalData8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	7	6	5	4	3	2	1	0
Name	PalData7	PalData6	PalData5	PalData4	PalData3	PalData2	PalData1	PalData0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

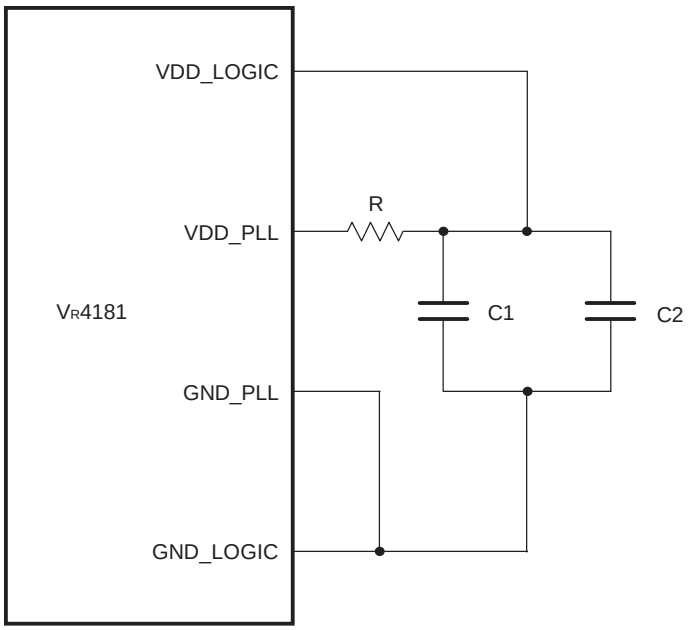
Bit	Name	Function
31 to 18	Reserved	0 is returned when read
17 to 0	PalData(17:0)	Color palette data (6-6-6 format)

★ **Caution Do not change palette data during LCD display.**

CHAPTER 22 PLL PASSIVE COMPONENTS

The VR4181 requires several external passive components for proper operation, which are connected to VDD_PLL as illustrated in Figure 22-1.

Figure 22-1. 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 analog power and ground for the PLL circuit (VDD_PLL, GND_PLL) from the regular power and ground (VDD_LOGIC, GND_LOGIC). The following values are an example for each component.

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 your specific application. In addition, the choke (inductor: *L*) can be considered for use as an alternative to the resistor (*R*) for use in filtering the power supply.

CHAPTER 23 COPROCESSOR 0 HAZARDS

The VR4110 CPU core avoids contention of its internal resources by causing a pipeline interlock in such cases as 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 on the 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 that is 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 VR4110 CPU core are as or less stringent than those of the VR4000. Table 23-1 lists the Coprocessor 0 hazards of the VR4110 CPU core. Code that complies with these hazards will run without modification on the VR4000.

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 interlocks of pipeline. Therefore, the required number of instruction must be controlled by program.

Table 23-1. Coprocessor 0 Hazards

Operation	Source		Destination	
	Source name	No. of cycles	Destination name	No. of cycles
MTC0	–		CPU general-purpose register	5
MFC0	CPU general-purpose register	3	–	
TLBR	Index, TLB	2	PageMask, EntryHi, EntryLo0, EntryLo1	5
TLBWI TLBWR	Index or Random, PageMask, EntryHi, EntryLo0, EntryLo1	2	TLB	5
TLBP	PageMask, EntryHi	2	Index	6
ERET	EPC or ErrorEPC, TLB	2	Status[EXL], Status[ERL]	4
	Status	2		
CACHE Index Load Tag	–		TagLo, TagHi, PErr	5
CACHE Index Store Tag	TagLo, TagHi, PErr	3	–	
CACHE Hit operations	cache line	3	cache line	5
Coprocessor usable test	Status[CU], [KSU], [EXL], [ERL]	2	–	
Instruction fetch	EntryHi[ASID], Status[KSU], [EXL], [ERL], [RE], Config[K0]	2	–	
	TLB	2		
Instruction fetch exception	–		EPC, Status	4
			Cause, BadVAddr, Context, XContext	5
Interrupts	Cause[IP], Status[IM], [IE], [EXL], [ERL]	2	–	
Load/Store	EntryHi[ASID], Status[KSU], [EXL], [ERL], [RE], Config[K0], TLB	3	–	
	Config[AD], [EP]	3		
	WatchHi, WatchLo	3		
Load/Store exception	–		EPC, Status, Cause, BadVAddr, Context, XContext	5
TLB shutdown	–		Status[TS]	2 (Inst.), 4 (Data)

Remark Brackets indicate a bit name or a field name of registers.

Cautions 1. If the setting of the K0 bit in the Config register is changed by executing an MTC0 instruction ★ **within the kseg0 or ckseg0 area, the change is reflected one to three instructions later from the MTC0 instruction.**

2. The instruction following an MTC0 instruction must not be an MFC0 instruction.

3. The five instructions following an MTC0 instruction for the Status register that changes the KSU bit and sets the EXL and ERL bits may be executed in the new mode, and not kernel mode. This can be avoided by setting the EXL bit first, leaving the KSU bit set to kernel, and later changing the KSU bit.

★ **4. If interrupts are disabled by setting the EXL bit in the Status register with an MTC0 instruction, an interrupt may occur immediately after the MTC0 instruction without change of the contents of the EPC register. This can be avoided by clearing the IE bit first, and later setting the EXL bit.**

5. There must be two non-load, non-CACHE instructions between a store and a CACHE instruction directed to the same cache line to be stored.

The status during execution of the following instruction 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.** After the contents of the CU0 bit of the Status register are modified, when accessing the CP0 registers in User mode or when executing an instruction such as TLB instructions, CACHE instructions, or branch instructions that use the resource of the CP0.

2. When accessing the CP0 registers in the operating mode set in 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) Interrupts

Source: The confirmation of registers judging the condition of occurrence of interrupt when an interrupt factor is detected.

(13) Loads/sores

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 Loads/Stores 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.

(15) TLB shutdown

Destination: The completion of writing to the TS bit of the Status register when a TLB shutdown occurs.

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	5 – (2 + 1)
TLBWR/TLBWI	Load or store using newly modified TLB	TLB Entry	1	5 – (3 + 1)
TLBWR/TLBWI	Instruction fetch using newly modified TLB	TLB Entry	2	5 – (2 + 1)
MTC0, Status [CU]	Coprocessor instruction that requires the setting of CU	Status [CU]	2	5 – (2 + 1)
TLBR	MFC0 EntryHi	EntryHi	1	5 – (3 + 1)
MTC0 EntryLo0	TLBWR/TLBWI	EntryLo0	2	5 – (2 + 1)
TLBP	MFC0 Index	Index	2	6 – (3 + 1)
MTC0 EntryHi	TLBP	EntryHi	2	5 – (2 + 1)
MTC0 EPC	ERET	EPC	2	5 – (2 + 1)
MTC0 Status	ERET	Status	2	5 – (2 + 1)
MTC0 Status [IE] Note	Instruction that causes an interrupt	Status [IE]	2	5 – (2 + 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.

Remark Brackets indicate a bit name or a field name of registers.

★

APPENDIX A RESTRICTIONS ON VR4181

A.1 RSTSW# During HALTimer Operation

The VR4181 ignores the RSTSW# signal even if it is asserted while the HALTimer is operating (counting). If the VR4181 is started while the RSTSW# signal is low, the RSTSW reset sequence is not executed and the VR4181 continues operating until the HALTimer is reset.

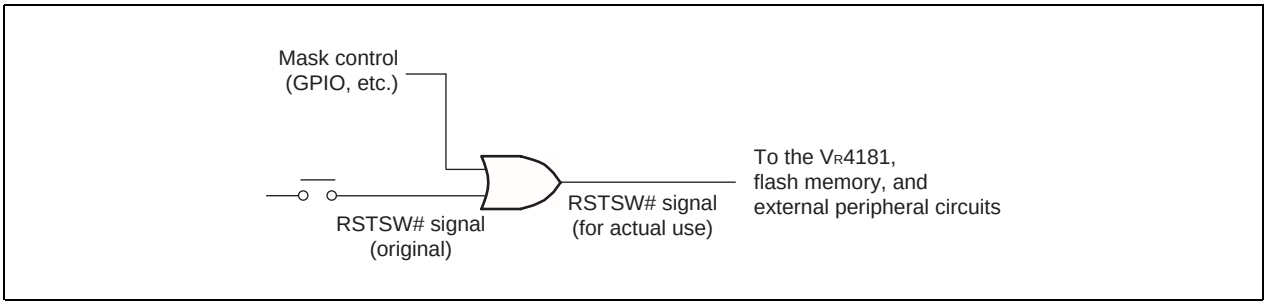
Consequently, the operation of the VR4181 may differ from the operation of the external peripheral circuits when the RSTSW# signal is used as a reset signal to the external peripheral circuits. Particularly, when the reset signal to a flash memory that includes a boot vector and the RSTSW# signal are shared, the VR4181 may not be able to read the correct program and hang up for 4 seconds between when the VR4181 is started and when the HALTimer is shut down.

[Workaround]

Do not share the reset signal to the external peripheral circuits with the RSTSW# signal.

However, if it is necessary to do so, insert a circuit like the one shown in the figure below to mask the RSTSW# signal between when the VR4181 is started and when the HALTimer is cleared, by using the GPIO pin.

Figure A-1. Mask Circuit for RSTSW# Signal



A.2 RSTSW# in Hibernate Mode

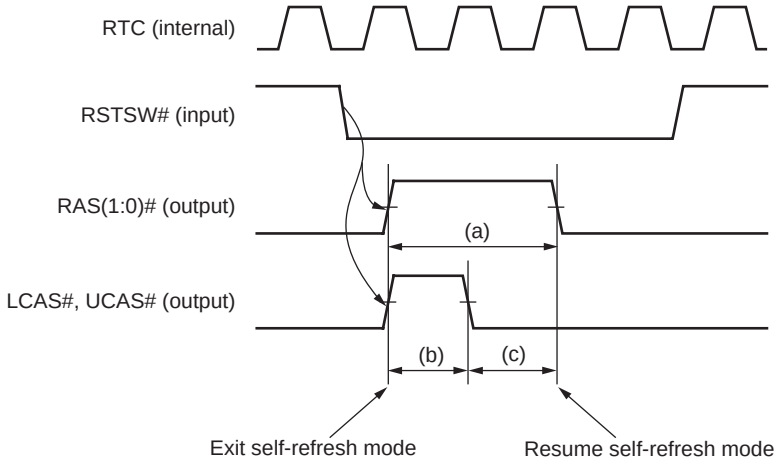
The VR4181 may release the self-refresh mode of DRAM when the RSTSW# signal is asserted in the Hibernate mode. As a result, the DRAM data may be lost.

(1) With EDO DRAM

When the RSTSW# signal goes low, the RAS# and CAS# signals go high and the self-refresh mode is released. After that DRAM returns to the self-refresh mode. At this time, the following phenomena may occur, and the DRAM data may be lost.

- DRAM is in the normal operation mode while the RAS# signal is high ((a) in Figure A-2) but a CBR refresh is not executed.
- The high-level output of the CAS# signal ((b) in Figure A-2) may be a spike.

Figure A-2. Release of Self-Refresh Mode by RSTSW# Signal (EDO DRAM)

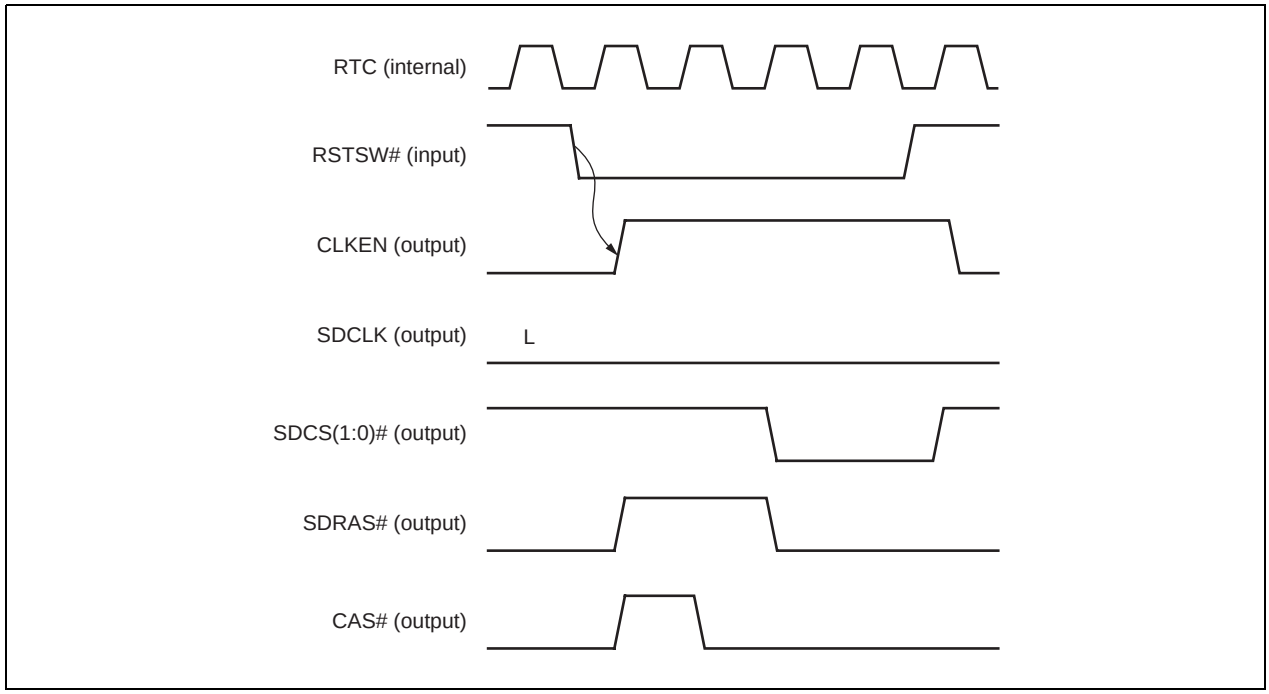


Pulse width: (a) 30 to 60 μs
(b) 0 (spike) to 30 μs
(c) 30 μs

(2) With SDRAM

When the RSTSW# signal goes low, the CLKEN (CKE) signal goes high. While the CLKEN signal is high, the self-refresh mode of SDRAM may be released. However, because the SDCLK signal is kept low, this problem does not occur in SDRAM that requires the rising edge of the SDCLK signal to release the self-refresh mode.

Figure A-3. Release of Self-Refresh Mode by RSTSW# Signal (SDRAM)



[Workaround]

Mask the RSTSW# signal via an external circuit using the MPOWER signal and GPIO pin, so that the RSTSW# signal does not go low in the Hibernate mode.

Numerics

A

B

C

H

I

K

L

R

S