NEC

User's Manual

V831™

32-bit Microprocessor

Hardware

μPD705101

Document No. U12273EJ4V0UM00 (4th edition)
Date Published January 1999 N CP(K)

[MEMO]

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note:

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

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note:

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

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note:

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

The information in this document is subject to change without notice.

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Corporation. NEC Corporation assumes no responsibility for any errors which may appear in this document.

NEC Corporation does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from use of a device described herein or any other liability arising from use of such device. No license, either express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Corporation or others.

While NEC Corporation has been making continuous effort to enhance the reliability of its semiconductor devices, the possibility of defects cannot be eliminated entirely. To minimize risks of damage or injury to persons or property arising from a defect in an NEC semiconductor device, customers must incorporate sufficient safety measures in its design, such as redundancy, fire-containment, and anti-failure features.

NEC devices are classified into the following three quality grades:

"Standard", "Special", and "Specific". The Specific quality grade applies only to devices developed based on a customer designated "quality assurance program" for a specific application. The recommended applications of a device depend on its quality grade, as indicated below. Customers must check the quality grade of each device before using it in a particular application.

- Standard: Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots
- Special: Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support)
- Specific: Aircrafts, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems or medical equipment for life support, etc.

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

Anti-radioactive design is not implemented in this product.

M7 96.5

MAJOR REVISIONS IN THIS EDITION

Page	Contents
p. 35	Change of **3.1 Internal Peripheral I/O Space**
p. 50	Addition of **4.7 Interrupt Requests by External Input Pins**
p. 51	Change of the description in **5.2 External I/O Cycle**
p. 64	Change of the description in **5.5 (4) Classification by data bus width**
p. 86	Change of the description in **6.2 Address Space and Block**
p. 92	Addition of **6.4 Wait Control by $\overline{\text{READY}}$ pin**
p. 118	Change of description in **8.7.3 Request from internal peripheral hardware**
p. 123	Change of **Figure 8-20. 16- to 32-Bit Data Bus Width (32-bit transfer bus sizing)**
p. 124	Change of **Figure 8-21. 16- to 16-Bit Data Bus Width (32-bit transfer bus sizing)**
p. 141	Addition of **Caution** to **9.2.5 (2) (b) Starting transmission/reception**
p. 176	Change of the description in **13.2.2 (3) Releasing by $\overline{\text{RESET}}$ pin input**
p. 178	Change of the description in **13.3.2 (2) Releasing by $\overline{\text{RESET}}$ pin input**
p. 181	Change of the description in **14.3 Reset**

The mark ★ shows major revised points.

Regional Information

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

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

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

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

NEC Electronics (Germany) GmbH
Duesseldorf, Germany
Tel: 0211-65 03 02
Fax: 0211-65 03 490

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

NEC Electronics Italiana s.r.l.
Milano, Italy
Tel: 02-66 75 41
Fax: 02-66 75 42 99

NEC Electronics (Germany) GmbH
Benelux Office
Eindhoven, The Netherlands
Tel: 040-2445845
Fax: 040-2444580

NEC Electronics (France) S.A.
Velizy-Villacoublay, France
Tel: 01-30-67 58 00
Fax: 01-30-67 58 99

NEC Electronics (France) S.A.
Spain Office
Madrid, Spain
Tel: 01-504-2787
Fax: 01-504-2860

NEC Electronics (Germany) GmbH
Scandinavia Office
Taeby, Sweden
Tel: 08-63 80 820
Fax: 08-63 80 388

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 Singapore Pte. Ltd.
United Square, Singapore 1130
Tel: 65-253-8311
Fax: 65-250-3583

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

NEC do Brasil S.A.
Electron Devices Division
Rodovia Presidente Dutra, Km 214
07210-902-Guarulhos-SP Brasil
Tel: 55-11-6465-6810
Fax: 55-11-6465-6829

J98. 11

PREFACE

Readers This manual is intended for user engineers who wish to understand the functions of the V831 (μPD705101) and design application systems using it.

Purpose This manual explains the hardware functions of the V831 in the following organization.

Organization The volumes of the V831 User's Manual are available: hardware (this manual) and architecture (V830 Family™ User's Manual - Architecture) manuals.

Hardware	Architecture
• General	• Register Set
• Pin Function	• Data Set
• CPU Function	• Address Space
• Internal peripheral function	• Instruction
• Appendix	• Interrupt and exception

How to Read This Manual It is assumed that the readers of this manual have a general knowledge of electricity, logic circuits, and microcomputers.

To understand the instruction functions in detail

→ Refer to the **V830 Family User's Manual - Architecture**.

To check the detailed function of a register whose name is known

→ Refer to **Appendix A Register Index**.

To understand the overall functions of the V831

→ Read this manual in the order of Table of Contents.

Legend Data significance: Left: higher digits, right: lower digits
Active low: $\overline{xxx}$ (top bar over pin or signal name)
Memory map address: Top: high-order, bottom: low-order
Note : Explanation of part of text marked Note
Caution: Item to be especially noted.
Remark: Supplement
Numeric notation: Binary ... ××××B
Decimal ... ××××
Hexadecimal ... ××××H
Prefix indicating power of 2 (address space and memory capacity):
K (kilo) : $2^{10} = 1024$
M (mega) : $2^{20} = 1024^2$
G (giga) : $2^{30} = 1024^3$

Related documents The related documents referred to in this publication may include preliminary versions. However, preliminary versions are not marked as such.

Document related to V831

Document Name	Document No.
V831 User's manual - Hardware	This manual
V830 Family User's Manual - Architecture	U12496E
V831 Data Sheet	U12979E

Document related to development tools (user's manual)

Document Name			Document No.
CA830 (C compiler)		Operation (UNIX™ based)	U11013E
		Operation (Windows™ based)	U11068E
		Assembly language	U11014E
		C language	U11010E
		Project manager	U11991E
RX830 (real-time OS)	ITRON1	Fundamental	U11730E
		Installation	U11731E
		Technical	U11713E
	μITRON Ver. 3.0	Fundamental	U13152E
		Installation	U13151E
		Technical	U13150E

CONTENTS

LIST OF FIGURES (1/3)

LIST OF FIGURES (2/3)

LIST OF FIGURES (3/3)

LIST OF TABLES

[MEMO]

CHAPTER 1 GENERAL

The V831 is a 32-bit RISC microprocessor for embedded control applications, with a high-performance 32-bit V830™ (μPD705100) processor core and many peripheral functions such as a DRAM/ROM controller, 4-channel DMA controller, real-time pulse unit, serial interface, and interrupt controller.

In addition to high interrupt response speed and optimized pipeline structure, the V831 offers sum-of-products operation instructions, concatenated shift instructions, and high-speed branch instructions to realize multimedia functions, and therefore, can provide high performance in multimedia systems such as internet/intra-net systems, car navigation systems, high-performance television, and color FAXes.

1.1 Features

- CPU function
  V830-compatible instructions
  Instruction cache : 4K bytes
  Instruction RAM : 4K bytes
  Data cache : 4K bytes
  Data RAM : 4K bytes
  Minimum number of instruction execution cycles : 1 cycle
  Memory space and I/O space : 4G bytes each
  Number of general-purpose registers : 32 bits × 32
- Interrupt/exception function
  Non-maskable : External input : 1
  Maskable : External input : 8 (of which 4 are multiplexed with internal sources)
  Internal source: 11 types
  Priority can be specified in 4 groups.
- Bus control function
  Chip select output: 8 blocks (RAS + 7 $\overline{CS}$)
  Memory and I/O space selectable for 4 $\overline{CS}$
  Linear address space of each block: 16M bytes
- Wait control function
  DRAM space : Software control of 0 or 1 wait state
  Other memory spaces : Software control of 0 to 7 wait states
  I/O space : Software control of 0 to 15 wait states
  Idle state : 0 to 3 states can be inserted.

- Memory access control function
  DRAM hiper page mode supported
  Page mode of Page-ROM supported
- DMA function
  4 channels
  Maximum number of transfers : 16,777,216 (2^{24})
  Transfer type : 2-cycle transfer
  Transfer mode : Single transfer and demand transfer
  Programmable wait function
- Serial interface function
  Asynchronous serial interface (UART) : 1 channel
  Clocked serial interface (CSI) : 1 channel
  On-chip dedicated baud rate generator : 1 channel
- Timer/counter function
  16-bit timer/event counter : 1 channel
  Timer output : 2
  16-bit capture/compare register : 4
  16-bit interval timer : 1 channel
- Port function
  3 I/O ports
- Clock generation function
  PLL clock synthesizer
- Standby function
  HALT and STOP modes
- Debug function
  Debug-dedicated synchronous serial interface : 1 channel
  Trace-dedicated interface : 1 channel
  Trace function : Branch PC trace and data trace
- Package
  160-pin plastic LQFP (24 × 24 mm)

1.2 Application Fields

- Internet/intra-net systems
- Car navigation
- High-performance television
- Color FAX

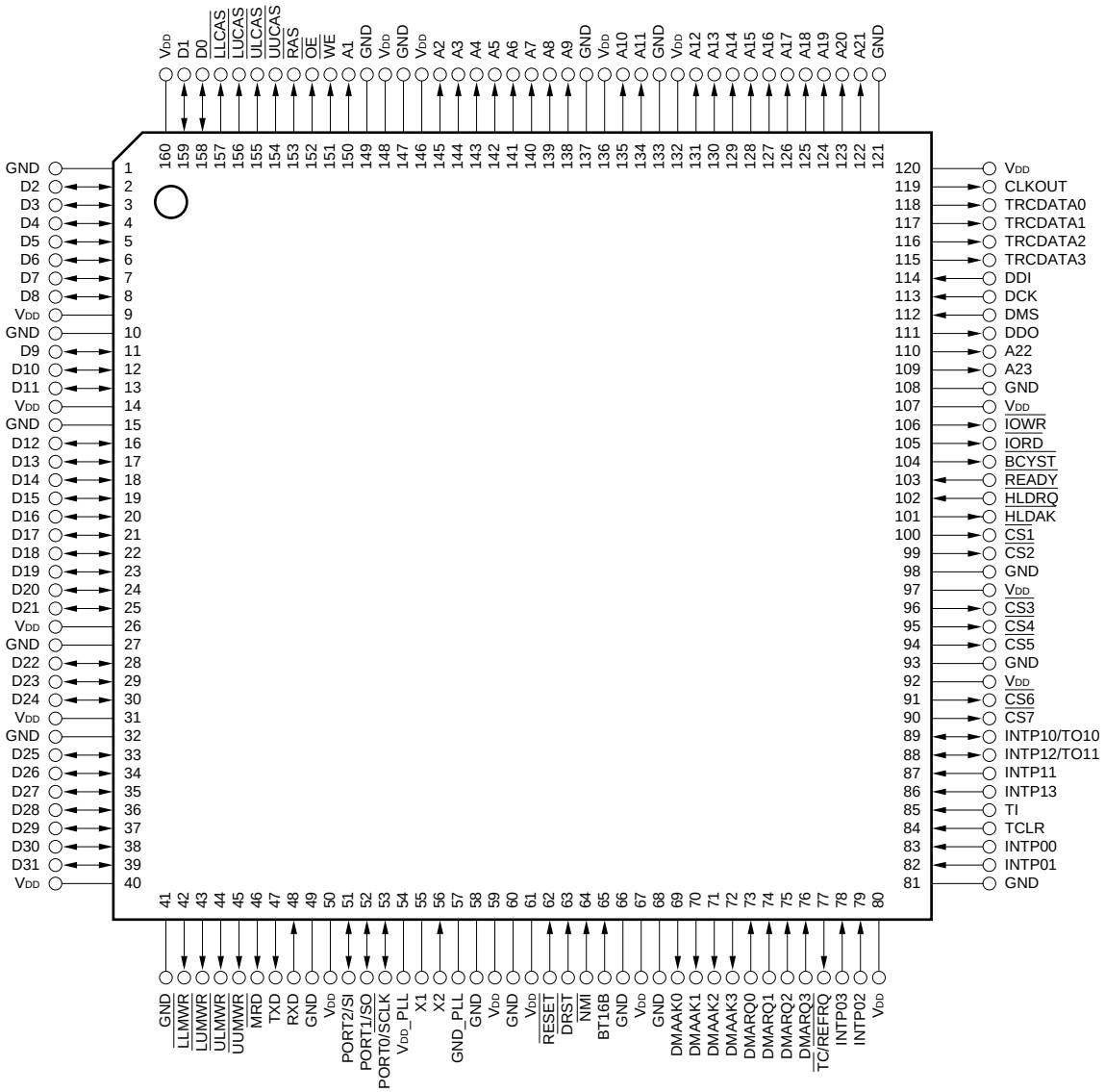
1.3 Ordering Information

Part Number	Package
μPD705101GM-100-8ED	160-pin plastic LQFP (fine pitch) (24 × 24 mm)

1.4 Pin Configuration (Top View)

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

μPD705101GM-100-8ED

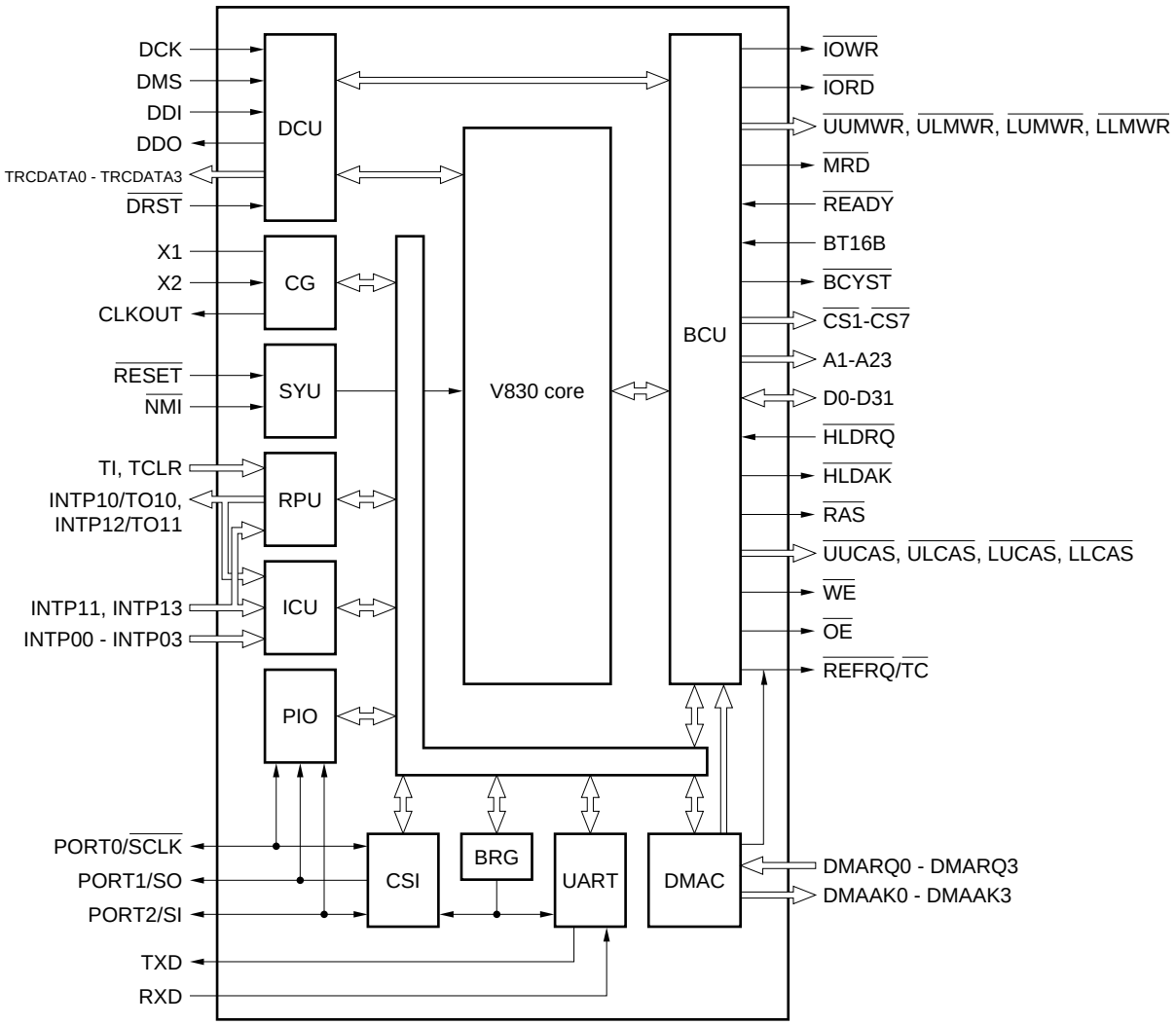


Pin names

A1 to A23 : Address Bus
$\overline{\text{BCYST}}$: Bus Cycle Start
BT16B : Boot Bus Size 16 bit
CLKOUT : Clock Out
$\overline{\text{CS1}}$ to $\overline{\text{CS7}}$: Chip Select
D0 to D31 : Data Bus
DCK : Debug Clock
DDI : Debug Data Input
DDO : Debug Data Output
DMAAK0 to DMAAK3
: DMA Acknowledge
DMARQ0 to DMARQ3
: DMA Request
DMS : Debug Mode Select
$\overline{\text{DRST}}$: Debug Reset
GND : Ground
GND_PLL : PLL Ground
$\overline{\text{HLDAK}}$: Hold Acknowledge
$\overline{\text{HLDRQ}}$: Hold Request
INTP00 to INTP03, INTP10 to INTP13
: Interrupt Request From Peripheral
$\overline{\text{IORD}}$: I/O Read
$\overline{\text{IOWR}}$: I/O Write
$\overline{\text{LLCAS}}$: Lower Lower Column Address Strobe
$\overline{\text{LLMWR}}$: Lower Lower Memory Write
$\overline{\text{LUCAS}}$: Lower Upper Column Address Strobe
$\overline{\text{LUMWR}}$: Lower Upper Memory Write
$\overline{\text{MRD}}$: Memory Read
$\overline{\text{NMI}}$: Non-Maskable Interrupt Request
$\overline{\text{OE}}$: Output Enable
PORT0 to PORT2
: Port
$\overline{\text{RAS}}$: Row Address Strobe
$\overline{\text{READY}}$: Ready
$\overline{\text{REFRQ}}$: Refresh Request
$\overline{\text{RESET}}$: Reset
RXD : Receive Data
$\overline{\text{SCLK}}$: Serial Clock
SI : Serial Input
SO : Serial Output
$\overline{\text{TC}}$: Terminal Count
TCLR : Timer Clear
TI : Timer Input
TO10, TO11: Timer Output
TRCDATA0 to TRCDATA3
: Trace Data
TXD : Transmit Data
$\overline{\text{ULCAS}}$: Upper Lower Column Address Strobe
$\overline{\text{ULMWR}}$: Upper Lower Memory Write
$\overline{\text{UUCAS}}$: Upper Upper Column Address Strobe
$\overline{\text{UUMWR}}$: Upper Upper Memory Write
V_{DD} : Power Supply
V_{DD}_PLL : PLL Power Supply
$\overline{\text{WE}}$: Write Enable
X1, X2 : Crystal Resonator

1.5 Internal Block Configuration

The internal block configuration of the V831 is as shown below.



1.6 Internal Units

The internal units of the V831 and their functions are as follows:

(1) Bus control unit (BCU)

Controls the address bus, data bus, and control bus pins. The major functions of BCU are as follows:

(a) Bus arbitration

Arbitrates the bus mastership among bus masters (CPU, DRAMC, DMAC, and external bus masters). The bus mastership can be changed after completion of the bus cycle under execution, and in an idle state.

(b) Wait control

Controls eight areas in the 16M-byte space corresponding to a $\overline{\text{RAS}}$ signal and seven chip select signals ($\overline{\text{CS1}}$ through $\overline{\text{CS7}}$). Generates chip select signals, controls wait states, and selects the type of bus cycle.

(c) DRAM controller

Generates $\overline{\text{RAS}}$ and four $\overline{\text{CAS}}$ signals, and controls access to DRAM. The hyper page mode of DRAM is supported and DRAM can be accessed in two types of cycle: normal access (off-page) and hyper page access (on-page).

(d) ROM controller

Accessing ROM with page access function is supported. The bus cycle immediately before and addresses are compared, and wait states are controlled in the normal access (off-page) and page access (on-page) modes. A page width of 8 bytes to 16 bytes can be supported.

(2) Interrupt controller (ICU)

Services maskable interrupt requests (INTP00 through INTP03, and INTP10 through INTP13) from internal peripheral hardware and external sources. The priorities of these interrupt requests can be specified in units of four groups, and edge-triggered or level-triggered interrupts can be nested.

(3) DMA controller (DMAC)

Transfers data between memory and I/O in the place of the CPU. The transfer type is 2-cycle transfer. Two transfer modes, single transfer and demand transfer, are available.

(4) Serial interface (UART/CSI/BRG)

One asynchronous serial interface (UART) channel and one clocked serial interface (CSI) channel is provided. As the serial clock source, the output of the baud rate generator (BRG) and the bus clock can be selected.

(5) Real-time pulse unit (RPU)

Provides timer/counter functions. The on-chip 16-bit timer/event counter and 16-bit interval timer can be used to calculate pulse intervals and frequencies, and to output programmable pulses.

(6) Clock generator (CG)

A frequency 3 times higher than that of a resonator connected to the X1 and X2 pins is supplied as the operating clock of the CPU. In addition, a bus clock (with the same cycle as the input clock) is also supplied as the operating clock of the peripheral units. An external clock can be also input instead of connecting a resonator.

(7) Port (PIO)

Provides port functions. Three I/O ports are available. The pins of these ports can be used as port pins or serial control pin.

(8) System control unit (SYU)

A circuit that rejects noise on the $\overline{\text{RESET}}$ signal (input)/$\overline{\text{NMI}}$ signal (input) is provided.

(9) Debug control unit (DCU)

A circuit to realize mapping and trace functions is provided to implement basic debugging functions.

[MEMO]

CHAPTER 2 PIN FUNCTION

2.1 Pin Function List

(1/2)

Pin Name	I/O	Function	Multiplexed Pin
D0 - D31	3-state I/O	Data bus	—
A1 - A23	3-state output	Address bus	—
$\overline{\text{UUCAS}}$		Column address strobe (most significant byte)	—
$\overline{\text{ULCAS}}$		Column address strobe (second byte)	—
$\overline{\text{LUCAS}}$		Column address strobe (third byte)	—
$\overline{\text{LLCAS}}$		Column address strobe (least significant byte)	—
$\overline{\text{RAS}}$		Row address strobe/chip select	—
$\overline{\text{UUMWR}}$		Memory write strobe (most significant byte)	—
$\overline{\text{ULMWR}}$		Memory write strobe (second byte)	—
$\overline{\text{LUMWR}}$		Memory write strobe (third byte)	—
$\overline{\text{LLMWR}}$		Memory write strobe (least significant byte)	—
$\overline{\text{MRD}}$		Memory read strobe	—
$\overline{\text{WE}}$		DRAM write strobe	—
$\overline{\text{OE}}$		DRAM read strobe	—
$\overline{\text{IORD}}$		I/O read strobe	—
$\overline{\text{IOWR}}$		I/O write strobe	—
$\overline{\text{REFRQ}}$		DRAM refresh request	$\overline{\text{TC}}$
$\overline{\text{CS1}}$, $\overline{\text{CS2}}$, $\overline{\text{CS7}}$		Memory chip select	—
$\overline{\text{CS3}}$ - $\overline{\text{CS6}}$		Memory chip select / I/O chip select	—
$\overline{\text{BCYST}}$		Bus cycle start	—
BT16B	Input	Specifies bus size on boot	—
$\overline{\text{READY}}$		Enables end of bus cycle	—
DMARQ0 - DMARQ3		DMA request (CH0 through CH3)	—
DMAAK0 - DMAAK3	Output	DMA enable (CH0 through CH3)	—
$\overline{\text{TC}}$		DMA transfer end	$\overline{\text{REFRQ}}$
RXD	Input	UART data input	—
TXD	Output	UART data output	—
SI	Input	CSI data input	PORT2
SO	Output	CSI data output	PORT1
$\overline{\text{SCLK}}$	I/O	CSI clock I/O	PORT0
TI	Input	Timer 1 count clock input	—
TCLR		Timer 1 clear, start	—
TO10	Output	RPU pulse output	INTP10
TO11			INTP12

(2/2)

Pin Name	I/O	Function	Multiplexed Pin
INTP10	Input	Interrupt request	TO10
INTP11			—
INTP12			TO11
INTP13			—
INTP00 - INTP03			—
$\overline{\text{HLDRQ}}$		Bus request	—
$\overline{\text{HLDAK}}$	Output	Bus enable	—
$\overline{\text{NMI}}$	Input	Non-maskable interrupt request	—
$\overline{\text{RESET}}$		System reset	—
PORT0	I/O	Port	$\overline{\text{SCLK}}$
PORT1			SO
PORT2			SI
X1	—	Connects crystal resonator. (Opened when external clock is input.)	—
X2	Input	Connects crystal resonator or inputs external clock.	—
CLKOUT	Output	Bus clock output	—
DCK	Input	Debug clock input	—
DDI		Debug data input	—
DDO	3-state output	Debug data output	—
DMS	Input	Debug mode select	—
$\overline{\text{DRST}}$		Reset input (debug module)	—
TRCDATA0 - TRCDATA3	Output	Trace data output	—
V_{DD}	—	Positive power supply	—
GND		Ground potential	—
V_{DD}_PLL		Positive power supply for PLL (internal clock generator)	—
GND_PLL		Ground potential for PLL (internal clock generator)	—

2.2 Pin Status

Table 2-1 shows the operating status of each pin.

Table 2-1. Status of Each Pin

Pin \ Operating Status	Reset	Bus Hold	HALT Mode	STOP Mode
CLKOUT	Clock output	Clock output	Clock output	0
$\overline{CS1}$ - $\overline{CS7}$	1	Hi-Z	**Note 1**	1
A1 - A23	Undefined	Hi-Z	**Note 1**	Undefined
D0 - D31	Hi-Z	Hi-Z	**Note 1**	Hi-Z
$\overline{BCYST}$	1	Hi-Z	**Note 1**	1
$\overline{MRD}$	1	Hi-Z	**Note 1**	1
$\overline{OE}$	1	Hi-Z	**Note 1**	1
$\overline{WE}$	1	Hi-Z	**Note 1**	1
$\overline{LLMWR}$, $\overline{LUMWR}$, $\overline{ULMWR}$, $\overline{UUMWR}$	1	Hi-Z	**Note 1**	1
$\overline{IORD}$	1	Hi-Z	**Note 1**	1
$\overline{IOWR}$	1	Hi-Z	**Note 1**	1
$\overline{HLDAK}$	1	0	**Note 1**	1
$\overline{RAS}$	1	Hi-Z	**Note 1**	0 Note 2
$\overline{LLCAS}$, $\overline{LUCAS}$, $\overline{ULCAS}$, $\overline{UUCAS}$	1	Hi-Z	**Note 1**	0 Note 3
$\overline{TC}$/$\overline{REFRQ}$	1	Hi-Z	**Note 1**	0 Note 4

Notes 1. DMA operation can be performed in the HALT mode.

2. The status before the STOP mode is retained if CBR self-refresh is disabled.

3. 1 if CBR self-refresh is disabled.

4. 1 if CBR self-refresh is disabled. If $\overline{TC}$ is selected, however, $\overline{TC}$ signal is output.

Remark 0 : Low-level output
1 : High-level output
Hi-Z : high impedance

2.3 Pin Function

2.3.1 Address bus

(1) A1 through A23 (Address Bus) ⋯ 3-state output

The address bus outputs address signals when the V831 accesses an external main memory or I/O unit. An address space of 2^{24} bytes can be accessed. The address signals change at the rising edge of the bus clock.

2.3.2 Data bus

(1) D0 through D31 (Data Bus) ⋯ 3-state I/O

The data bus inputs or outputs write data or read data when the V831 accesses an external main memory or I/O unit. These data signals change at the rising edge of the bus clock.

2.3.3 Bus control signals

(1) $\overline{\text{READY}}$ (Ready) ⋯ Input

This signal extends the bus cycle to match it to the access time of the memory or I/O. It is sampled at the rising edge of the bus clock immediately after the read/write signal. Be sure to observe the setup/hold time of the $\overline{\text{READY}}$ input; otherwise, the operation will not be guaranteed.

(2) $\overline{\text{HLDRQ}}$ (Hold Request) ⋯ Input

This pin requests the CPU for the bus mastership. It is sampled at the rising edge of the bus clock.

(3) $\overline{\text{HLDAK}}$ (Hold Acknowledge) ⋯ Output

This is an acknowledge signal in response to the $\overline{\text{HLDRQ}}$ input.

When the CPU receives the $\overline{\text{HLDRQ}}$ signal, it asserts the $\overline{\text{HLDAK}}$ signal active. When $\overline{\text{HLDRQ}}$ is deasserted inactive, the CPU deasserts the $\overline{\text{HLDAK}}$ signal inactive, and is granted the bus mastership again.

(4) $\overline{\text{MRD}}$ (Memory Read) ⋯ 3-state output

This is a strobe signal indicating that the bus cycle under execution is a read cycle of the external memory. It changes in synchronization with the falling of the bus clock.

If the Page-ROM cycle continues, however, this signal is always active. It is always inactive in the refresh cycle.

(5) $\overline{\text{LLMWR}}$ (Lower Lower Memory Write) ⋯ 3-state output

This is a strobe signal for a data write to the external memory. It validates the least significant byte of the data bus. This signal changes in synchronization with the falling of the bus clock.

(6) $\overline{\text{LUMWR}}$ (Lower Upper Memory Write) ⋯ 3-state output

This is a strobe signal for a data write to the external memory. It validates the third byte of the data bus. This signal changes in synchronization with the falling of the bus clock.

(7) $\overline{\text{ULMWR}}$ (Upper Lower Memory Write) ⋯ 3-state output

This is a strobe signal for a data write to the external memory. It validates the second byte of the data bus. This signal changes in synchronization with the falling of the bus clock.

(8) $\overline{\text{UUMWR}}$ (Upper Upper Memory Write) ⋯ 3-state output

This is a strobe signal for a data write to the external memory. It validates the most significant byte of the data bus. This signal changes in synchronization with the falling of the bus clock.

(9) $\overline{\text{IORD}}$ (I/O Read) ⋯ 3-state output

This is a strobe signal indicating that the bus cycle under execution is a read cycle for an external I/O. It changes in synchronization with the falling of the bus clock.

(10) $\overline{\text{IOWR}}$ (I/O Write) ⋯ 3-state output

This is a strobe signal for a data write to an external I/O. It changes in synchronization with the falling of the bus clock.

(11) BT16B (Bout Bus Size 16 bit) ⋯ Input

This signal fixes the external data bus width of an area specified by $\overline{\text{CS7}}$ on initializing the CPU to 16 bits. When this signal is asserted active, a mode supporting a 16-bit data bus system is set. This signal is sampled at the rising of the clock next to the one at which the $\overline{\text{RESET}}$ signal is made high.

BT16B can be changed only at reset. If this signal is changed at any other time, the CPU operation is not guaranteed.

(12) $\overline{\text{BCYST}}$ (Bus Cycle Start) ⋯ Output

This signal indicates the first one cycle of the bus cycle. It is generated at the timing of the $\overline{\text{CAS}}$ cycle when the DRAM is accessed. This signal changes in synchronization with the rising of the bus clock.

2.3.4 System control signals

(1) $\overline{\text{RESET}}$ (Reset) ⋯ Input

This signal initializes the V831. Be sure to hold the active period of this signal at least for the duration of 25 clocks. The low-level width input to the $\overline{\text{RESET}}$ pin on power application must be wider than the oscillation stabilization time of the resonator. Make sure that the oscillation stabilization time satisfying the specifications of the resonator used elapses.

Keep the stabilization time of the PLL to 10 ms or longer. When the $\overline{\text{RESET}}$ signal is input and deasserted inactive, the V831 initializes each signal and internal register, and starts instruction execution from address FFFFFFF0H.

(2) X1 and X2 (Crystal Resonator) ⋯ Input

Connect a crystal resonator to these pins when the internal clock generator is used. When an external clock is used, input used the clock to the X2 pin. Leave the X1 pin open.

(3) CLKOUT (Clock Out) ⋯ Output

This pin outputs an internally generated bus clock.

(4) $\overline{\text{CS1}}$, $\overline{\text{CS2}}$, and $\overline{\text{CS7}}$ (Chip Select) ⋯ 3-state output

These pins output chip select signals to the memory address space. The address block that outputs a signal is fixed for each chip select signal. These signals change in synchronization with the rising of the bus clock.

(5) $\overline{\text{CS3}}$ through $\overline{\text{CS6}}$ (Chip Select) ⋯ 3-state output

These pins output chip select signals to memory address space or I/O address space. To which of these signals are to be output is determined by register setting. The address block that outputs a signal is fixed for each chip select signal. These signals change in synchronization with the rising of the bus clock.

2.3.5 Interrupt control signals

(1) INTP10 through INTP13 (Interrupt Request From Peripheral) ⋯ Input

These are asynchronous interrupt request signals to the interrupt control unit (ICU). It can be selected whether these signals are triggered by edge or level (high level). When the RPU is used, however, the level trigger cannot be selected.

(2) INTP00 through INTP03 (Interrupt Request From Peripheral) ⋯ Input

These are asynchronous interrupt request signals for the interrupt control unit (ICU). It can be selected whether these signals are triggered by edge or level (high level).

(3) $\overline{\text{NMI}}$ (Non-Maskable Interrupt Request) ⋯ Input

This is an interrupt request signal to the CPU which cannot be masked. It is sampled at the rising edge of the clock and rejects noise of 5 clocks or less.

The $\overline{\text{NMI}}$ request is accepted at the falling edge of the $\overline{\text{NMI}}$ signal after noise has been rejected.

2.3.6 DRAM control signals

(1) $\overline{\text{REFRQ}}$ (Refresh Request) ⋯ 3-state output

This is a refresh request signal to the DRAM. It changes in synchronization with the rising of the bus clock. This signal is used to control $\overline{\text{RAS}}$ in the refresh cycle when the number of DRAMs connected is increased by decoding addresses with a external circuit.

(2) $\overline{\text{OE}}$ (Output Enable) ⋯ 3-state output

This is a read enable signal to the DRAM. It changes in synchronization with the rising of the bus clock.

(3) $\overline{\text{RAS}}$ (Row Address Strobe) ⋯ 3-state output

This is a raw address strobe signal to the DRAM. Its timing differs in the refresh cycle. This signal changes in synchronization with the rising of the bus clock.

(4) $\overline{\text{LLCAS}}$ (Lower Lower Column Address Strobe) ⋯ 3-state output

This is a column address strobe signal to the DRAM. It validates the least significant byte of the data bus. Its timing differs in the refresh cycle. This signal changes in synchronization with the falling or rising of the bus clock.

(5) $\overline{\text{LUCAS}}$ (Lower Upper Column Address Strobe) ⋯ 3-state output

This is a column address strobe signal to the DRAM. It validates the third byte of the data bus. Its timing differs in the refresh cycle. This signal changes in synchronization with the falling or rising of the bus clock.

(6) $\overline{\text{ULCAS}}$ (Upper Lower Column Address Strobe) ⋯ 3-state output

This is a column address strobe signal to the DRAM. It validates the second byte of the data bus. Its timing differs in the refresh cycle. This signal changes in synchronization with the falling or rising of the bus clock.

(7) $\overline{\text{UUCAS}}$ (Upper Upper Column Address Strobe) ··· 3-state output

This is a column address strobe signal to the DRAM. It validates the most significant byte of the data bus. Its timing differs in the refresh cycle. This signal changes in synchronization with the falling or rising of the bus clock.

(8) $\overline{\text{WE}}$ (Write Enable) ··· 3-state output

This signal indicates that the bus cycle under execution is a write cycle to the DRAM. It changes in synchronization with the rising of the bus clock.

2.3.7 DMA control signals

(1) DMARQ0 through DMARQ3 (DMA Request) ··· Input

These are DMA service request signals, and correspond to DMA channels 0 through 3. Their priorities are fixed: DMARQ0 > DMARQ1> DMARQ2 > DMARQ3. The DMARQ0 through DMARQ3 signals are sampled at the rising edge of the bus clock. Keep these signals active until the corresponding DMA requests are accepted. When the DMARQ0 through DMARQ3 signals are not used, deassert these pins inactive (the active levels of these pins can be changed by using a register of the DMAC).

(2) DMAAK0 through DMAAK3 (DMA Acknowledge) ··· Output

These signals indicate that the corresponding DMA service requests are granted. They correspond to DMA channels 0 through 3. These signals are asserted active at the rising edge of the bus clock and remain active during DMA transfer.

(3) $\overline{\text{TC}}$ (Terminal Count) ··· Output

This signal indicates that the DMA transfer by the DMA controller is completed. It is asserted active at the rising edge of the bus clock. Because this signal outputs the logical sum of $\overline{\text{TC}}$ of channels 0 through 3, generate the $\overline{\text{TC}}$ signal of each channel by ANDing the DMAAK0 through DMAAK3 signals with an external circuit.

2.3.8 Real-time pulse control signals

(1) TO10 and TO11 (Timer Output) ··· Output

These signals indicate coincidence between the value of timer 1 (TM1) and the value of the capture/compare register (CC0) of the real-time pulse unit (RPU). The TO10 and TO11 signals are set by detection of coincidence of the CC10 and CC12 registers, and are reset by detection of coincidence of the CC11 and CC13 registers. The output of these signals can be inverted by setting the mode of the RPU.

(2) TCLR (Timer Clear) ··· Input

This is the count clear start signal of TM1 of the RPU.

(3) TI (Timer Input) ··· Input

This is an external clock signal used by TM1 of the RPU. Select whether TM1 of the RPU uses an external clock signal or a clock resulting from dividing the internal bus clock, on initialization.

2.3.9 Serial control signals

(1) TXD (Transmit Data) ··· Output

This is the serial transmit data output pin of UART. The TXD signal changes in synchronization with the internal serial clock. It remains high when no data is transmitted.

(2) RXD (Receive Data) ⋯ Input

This is the serial receive data input pin of UART.

(3) $\overline{\text{SCLK}}$ (Serial Clock) ⋯ I/O

This is the serial clock I/O pin of CSI. Whether this pin is used as an input or output pin is set by a register.

(4) SO (Serial Output) ⋯ Output

This is the serial transmit data output pin of CSI. It changes in synchronization with the falling of the $\overline{\text{SCLK}}$ signal. This pin goes into a high-impedance state when no data is transmitted.

(5) SI (Serial Input) ⋯ Input

This is the serial receive data input pin of CSI. The SI signal is sampled at the rising edge of the $\overline{\text{SCLK}}$ signal.

2.3.10 Port control signals

(1) PORT0 through PORT2 (Port) ⋯ 3-state I/O

These are port signals. The input or output mode of these signals can be selected by a register.

2.3.11 Debug control signals

(1) DCK (Debug Clock) ⋯ Input

This is a debug clock input pin. The DMS and DDI signals are sampled at the rising edge of the DCK signal, and data is output from the DDO pin at the falling edge of the DCK signal. Keep this signal high when the debug function is not used.

(2) DDI (Debug Data Input) ⋯ Input

This is a debug data input pin. It is sampled at the rising edge of the DCK signal when the debug serial interface is in the Shift state. Data is input to this pin with the LSB first. Keep this pin high when the debug function is not used.

(3) DDO (Debug Data Output) ⋯ 3-state output

This is a debug data output pin. It outputs data at the falling edge of the DCK signal with the LSB first when the debug serial interface is in the Shift state.

(4) DMS (Debug Mode Select) ⋯ Input

This is a debug mode select input pin. The state machine of the debug serial interface changes depending on the level of the DMS signal. The DMS signal is sampled at the rising edge of the DCK signal. Keep this signal high when the debug function is not used.

(5) $\overline{\text{DRST}}$ (Debug Reset) ⋯ Input

This is a debug reset input pin, and inputs a negative logic signal that initializes the DCU asynchronously. When this signal is made low, the DCU is reset and invalidated. Keep this signal low when the debug function is not used.

(6) TRCDATA0 through TRCDATA3 (Trace Data) ⋯ Output

These are trace data output pins. They output packet trace data in 4-bit units, starting from the LSB, at the rising edge of the CLKOUT signal.

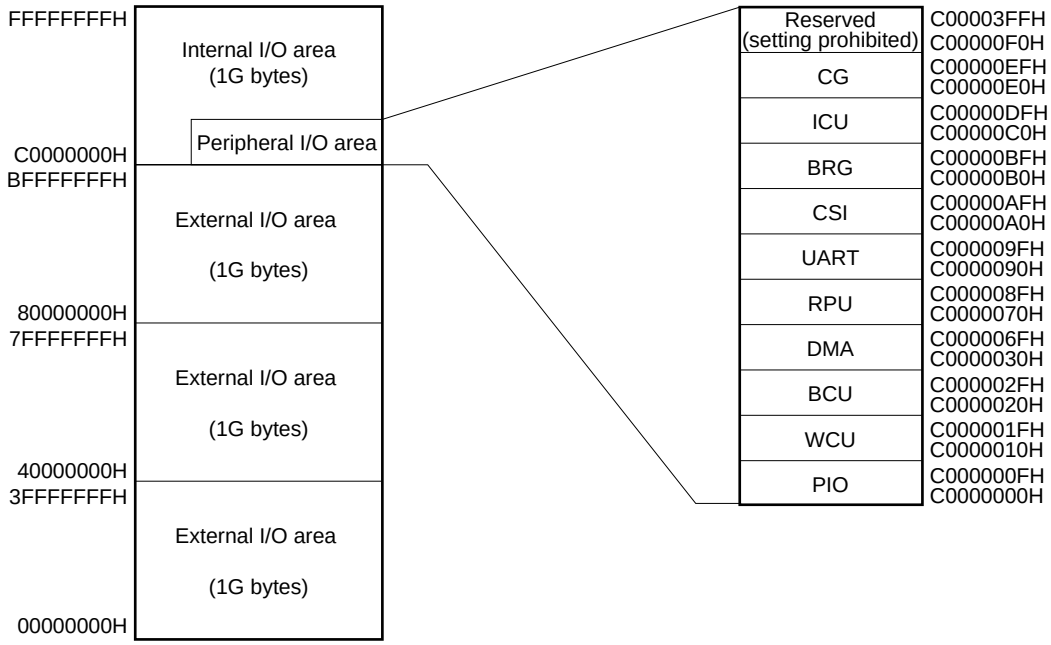
CHAPTER 3 CPU FUNCTION

This chapter explains the internal peripheral I/O space and CPU core system registers of the V831. For the instruction set and register configuration, refer to the **V830 Family User's Manual-Architecture**. For the internal peripheral I/O registers, refer to the description of each register in the function description of the peripheral units.

3.1 Internal Peripheral I/O Space

The internal peripheral I/O space is reserved at addresses C0000000H through C00003FFH (1K bytes) of the high-order 1G bytes of the internal I/O area. To access the internal I/Os, use the IN.H/OUT.H instruction (in half-word ★ units) or IN.B/OUT.B instruction (in byte units). When the internal I/O area is accessed, the bus cycle to the external device is not started and the idle state is set. For the status of each pin, refer to **5.6 Idle State**.

Figure 3-1. Internal Peripheral I/O Map



3.1.1 Notes

Note the following points when using the internal peripheral I/O space.

- Only the low-order 10 bits of a 32-bit address are used to decode a register address in terms of hardware, and an area of 1K bytes is used for register allocation.

- The low-order bit (A0) of an address is not decoded. If a register at an odd address (address 2n+1) is accessed in byte units, therefore, a register at an even address (address 2n) is accessed by the hardware.

- The V831 does not have a register that can be accessed in word units. If an internal peripheral I/O register is accessed in word units, it is forcibly accessed in half-word units.

- If a register that can be accessed in byte units is accessed in half-word units, the high-order 8 bits are undefined when the register is read. When data is written to the register, the low-order 8 bits are written.

- Addresses not allocated to registers are reserved for future expansion. If these addresses are accessed, the operation is undefined and not guaranteed.

3.2 CPU Core System Registers

With the V831, the processor ID register (PIR) is changed. PIR is a register that only identifies the CPU type number, and its contents are fixed to 00008301H (this register can be only read in 32-bit units).

For the initial value of each register after reset, refer to **Table 14-2. Initial Value of Each Register after Reset**.

CHAPTER 4 INTERRUPT/EXCEPTION PROCESSING FUNCTION

4.1 Interrupt/Exception Processing

The V831 has a dedicated interrupt controller (ICU) that can process a total of 15 interrupt requests.

An interrupt is an event that takes place independently of program execution. On the other hand, an exception is an event that takes place depending on program execution. In general, interrupts are processed prior to exceptions.

In the V831, each interrupt request from on-chip peripheral function units and external units can be processed. Exception processing can be started by an instruction (TRAP instruction) or by occurrence of an exception (such as an illegal instruction code (except address trap exception)). The cause of an exception can be identified by an exception code stored to ECR (Exception Cause Register).

Four levels of priorities can be specified in software for interrupt requests. An interrupt servicing is started at least six bus clocks after the request has been issued. Three bus clocks of these six bus clocks are used to reject noise. An internal interrupt is started at least 3 bus clocks after the request has been issued.

4.1.1 Interrupt/exception processing types

The interrupts/exceptions of the V831 can be classified into the following four types:

- Non-maskable interrupt: 1 source
- Maskable interrupt : 15 sources
- Software exception : 32 sources
- Exception trap : 4 sources

Table 4-1. Interrupt List

Type	Classification	Source of Interrupt/Exception		Exception Code (ECR)	Handler Address	Restore PC Note 2
		Name Note 1	Cause			
Reset	Interrupt	RESET	Reset input	FFF0H	FFFFFFF0H	Undefined
Non-maskable	Interrupt	NMI	NMI input	FFD0H	FFFFFFD0H	next PC Note 3
Software exception	Exception	TRAP 1nH	TRAP instruction	FFBnH	FFFFFFB0H	next PC
		TRAP 0nH	TRAP instruction	FFAnH	FFFFFFA0H	
Exception trap	Exception	NMI	Dual exception	**Note 4**	FFFFFFD0H	current PC
		FAULT	Fatal exception	Not affected	FFFFFFE0H	
		I-OPC	Illegal instruction code	FF90H	FFFFFF90H	
		DIV0	Zero division	FF80H	FFFFFF80H	

Notes 1. Handler names used in development tools or software.

2. The PC value saved to EIPC/FEPC/DPC when interrupt/exception processing is started.

3. Execution of all instructions cannot be stopped by an interrupt.

4. The exception code of an exception causing a dual exception.

Remark n = 0H to FH

Table 4-2. Interrupt List (Maskable interrupts)

Type	Classification	Group	In-Group Priority	Cause of Interrupt			Exception Code	Handler Address [Note 3]		Restore PC [Note 1]
				Name	Cause	Unit		HCCW.IHA=0	HCCW.IHA=1	
Maskable	Interrupt	GR3	3	RESERVED	Reserved	—	FEF0H	FFFFFEF0H	FE0000F0H	next PC [Note 2]
			2	INTOV1	Timer 1 overflow	RPU	FEE0H	FFFFFEE0H	FE0000E0H	
			1	INTSER	UART receive error	UART	FED0H	FFFFFED0H	FE0000D0H	
			0	INTP03	INTP03 pin input	External	FEC0H	FFFFFEC0H	FE0000C0H	
		GR2	3	INTSR	UART receive end	UART	FEB0H	FFFFFEB0H	FE0000B0H	
			2	INTST	UART transmit end	UART	FEA0H	FFFFFEA0H	FE0000A0H	
			1	INTCSI	CSI transmit/receive end	CSI	FE90H	FFFFFE90H	FE000090H	
			0	INTP02	INTP02 pin input	External	FE80H	FFFFFE80H	FE000080H	
		GR1	3	INTDMA	DMA transfer end	DMAC	FE70H	FFFFFE70H	FE000070H	
			2	INTP10/ INTCC10	INTP10 pin input/ coincidence of CC10	External/ RPU	FE60H	FFFFFE60H	FE000060H	
			1	INTP11/ INTCC11	INTP11 pin input/ coincidence of CC11	External/ RPU	FE50H	FFFFFE50H	FE000050H	
			0	INTP01	INTP01 pin input	External	FE40H	FFFFFE40H	FE000040H	
		GR0	3	INTCM4	Coincidence of CM4	RPU	FE30H	FFFFFE30H	FE000030H	
			2	INTP12/ INTCC12	INTP12 pin input/ coincidence of CC12	External/ RPU	FE20H	FFFFFE20H	FE000020H	
			1	INTP13/ INTCC13	INTP13 pin input/ coincidence of CC13	External/ RPU	FE10H	FFFFFE10H	FE000010H	
			0	INTP00	INTP00 pin input	External	FE00H	FFFFFE00H	FE000000H	

Notes 1. The PC value saved to EIPC when interrupt servicing is started.

2. Execution of all instructions cannot be stopped by an interrupt.

3. FFFFFEn0H can be selected as a handler address when HCCW.IHA = 0, and FE0000n0H can be selected when HCCW.IHA = 1 (N = 0H to FH).

Caution The exception codes and handler addresses of the maskable interrupts shown above are the values if the default priority is used. If the priority is changed, refer to Table 4-3 Relation between Priority, Exception Code, Handler Address, and Interrupt Priority.

4.2 Non-Maskable Interrupt

A non-maskable interrupt is unconditionally accepted even when interrupts are disabled. It takes precedence over all the other interrupts and is not controlled by the priority.

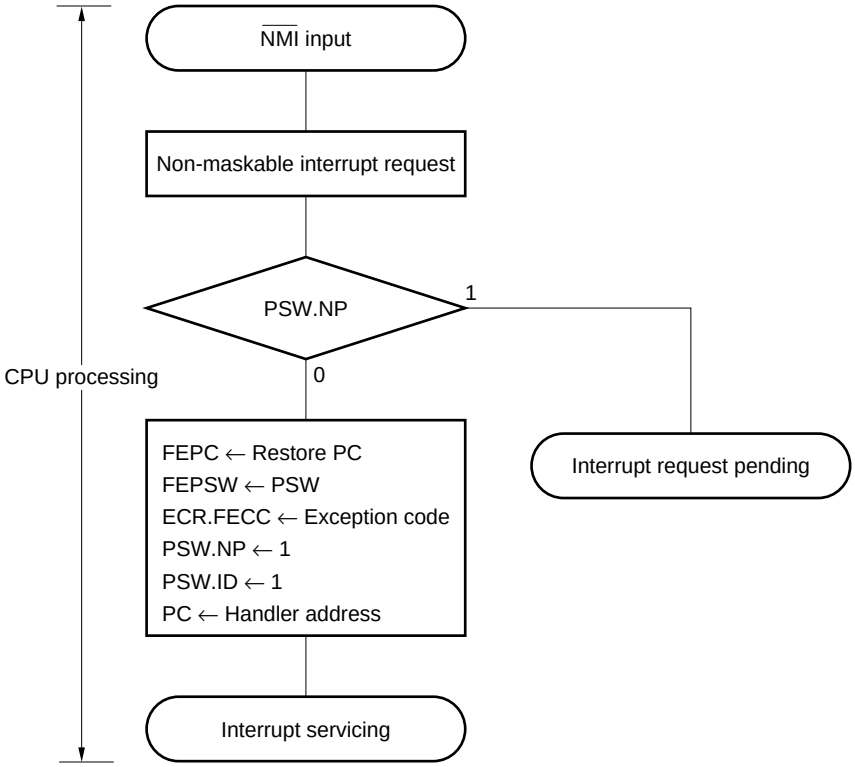
The non-maskable interrupt request is issued by the $\overline{\text{NMI}}$ pin.

4.2.1 Servicing non-maskable interrupt

If a non-maskable interrupt occurs because the $\overline{\text{NMI}}$ signal is input, the processing shown in Figure 4-1 is performed, and control is transferred to the handler routine. FEPC and FEPSW are used as status saving registers.

The $\overline{\text{NMI}}$ signal that is input while the non-maskable interrupt is processed (PSW.NP = 1) is kept pending by the CPU. In this case, if PSW.NP is cleared to 0 by using the RETI or LDSR instruction, new non-maskable interrupt servicing is started by the pending non-maskable interrupt request.

Figure 4-1. Processing Flow of Non-Maskable Interrupt



4.3 Maskable Interrupts

Maskable interrupt requests can be masked by control registers. The V831 has 15 maskable interrupt sources.

If two or more maskable interrupts occur at the same time, they are accepted according to the default priority determined in advance by the ICU. In addition to the default priority, four levels of priorities can be specified in software (priority in a group is fixed).

A maskable interrupt is masked by the logical sum of the NP, EP, and ID bits of the PSW. If interrupt level n that is passed to the CPU is lower than the interrupt enable level of the PSW (specified by bits I0 through I3 of the PSW), the interrupt is not accepted. Therefore, the highest interrupt level (n = 15) cannot be disabled by the interrupt enable level.

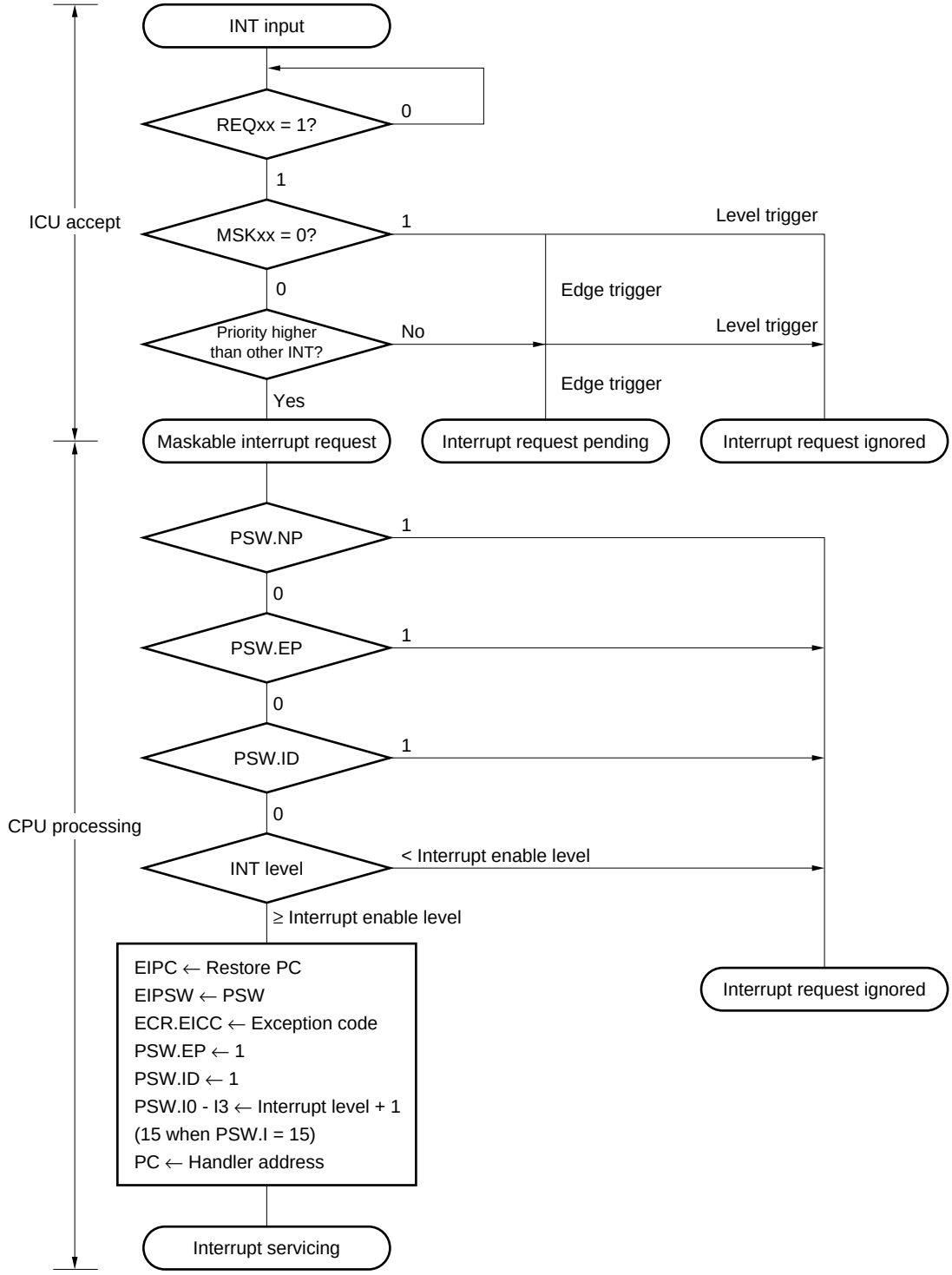
When an interrupt request is accepted, the other interrupts are disabled (PSW.ID = 1); therefore, no more maskable interrupts can be accepted. The accepted interrupt level n plus 1 (n+1) is set to the I0 through I3 bits of the PSW.

To enable nesting of interrupts, first save the EIPC and EIPSW to the memory or register, and then enable the interrupts (PSW.ID = 0, EP = 0). Execute the RETI instruction after disabling the interrupts (PSW.ID = 1), and restore EIPC and EIPSW to the original values.

4.3.1 Maskable interrupt servicing format

If a maskable interrupt occurs due to input of an interrupt request signal (INT), the processing shown in Figure 4-2 is performed, and control is transferred to the handler routine. EIPC and EIPSW are used as status saving registers. The INT input masked by the ICU and INT input that takes place while another interrupt is serviced (PSW.NP = 1 or PSW.ID = 1) is kept pending by the ICU. In this case, if the interrupt is unmasked, or if the PSW.NP and PSW.ID are cleared to 0 by using the RETI or LDSR instruction, new maskable interrupt servicing is started by the pending INT input.

Figure 4-2. Maskable Interrupt Servicing Flow



4.3.2 Priority of maskable interrupt

The V831 can nest interrupts by accepting an interrupt while it is servicing another interrupt. Nesting can be controlled by the priority.

Priority control is implemented by the default priority or by software, using the interrupt group priority register.

When two or more interrupts occur at the same time, the interrupts are serviced according to the priority (default priority) allocated in advance to each group of interrupt requests (with one group consisting of four interrupts) (refer to **Table 4-2. Interrupt List (Maskable Interrupts)**).

The priority is also controlled by software by classifying the interrupt requests into four groups, using the interrupt group priority control register. In each group, the priorities of the interrupts are fixed.

When an interrupt is accepted, the ID flag and EP flag of the PSW are automatically set. To perform nesting of interrupts, therefore, enable interrupts (PSW.ID = 0, PSW.EP = 0) in the interrupt servicing program.

(Service program of maskable interrupt or exception)

- Saves EIPC to memory or register.
- Saves EIPSW to memory or register.
- Enables accepting interrupt (PSW.ID = 0, PSW.EP = 0).

  …

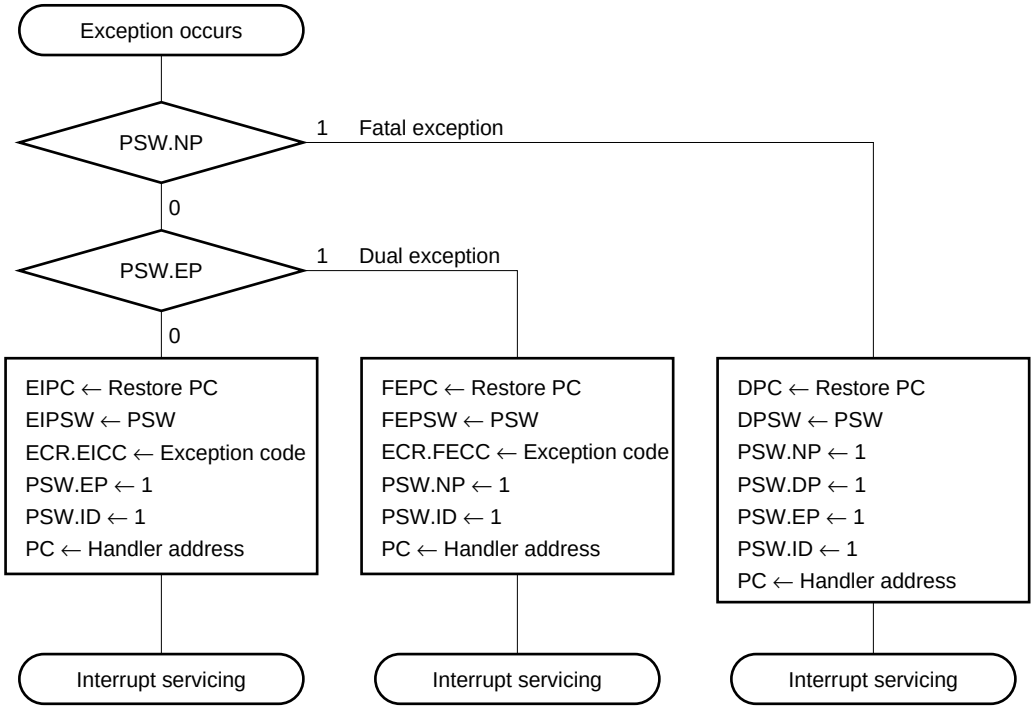
  … ← Interrupt such as INT input is accepted.

  …
- Disables accepting interrupt (PSW.ID = 1, PSW.EP = 1).
- Restores saved value to EIPSW.
- Restores saved value to EIPC.
- Executes RETI instruction.

4.4 Exception Processing

If an exception occurs, the following processing is performed and control is transferred to the handler routine.

Figure 4-3. Exception Processing Flow

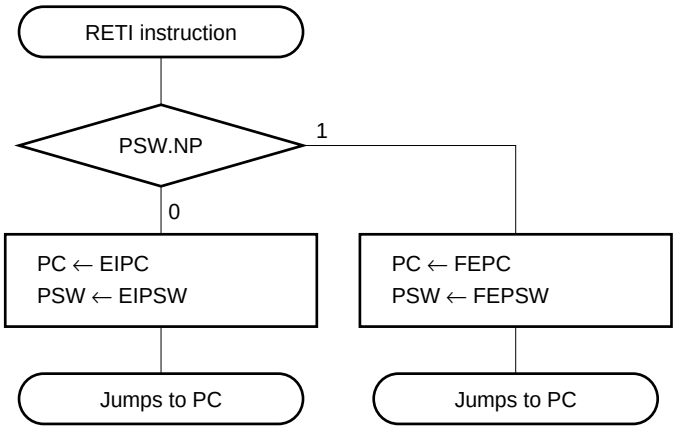


4.5 Restoring from Exception/Interrupt

4.5.1 Restoring from exception/interrupt

Program execution is restored from an exception or interrupt event other than a fatal exception by using the RETI instruction.

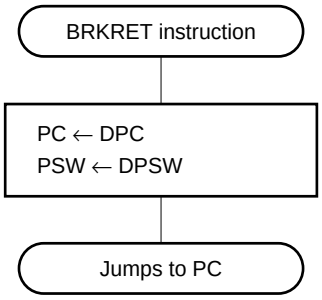
Figure 4-4. Flow of Restoration from Exception/Interrupt



4.5.2 Restoring from fatal exception routine

Program execution is restored from fatal exception processing by using the BRKRET instruction.

Figure 4-5. Flow of Restoration from Fatal Exception Routine



4.6 Interrupt Control Registers

It can be selected whether an interrupt is triggered by edge or level.

All the interrupts caused by the internal units of the V831 are triggered by edge. For an explanation of how to specify level trigger or edge trigger, refer to **4.6.5 ICU mode register (IMOD)**.

4.6.1 Interrupt group priority register (IGP)

The interrupt group priority register (IGP) specifies the priority between interrupt groups. Specify or change the priority with the interrupts masked. Table 4-3 shows the relation of the handler addresses.

Figure 4-6. Interrupt Group Priority Register (IGP)

Address: C00000C0H

	7	6	5	4	3	2	1	0
IGP	PR3		PR2		PR1		PR0	
Initial value:	1	1	1	0	0	1	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Description
7 - 0	PR3 - PR0	Group Priority Specifies priority of interrupt group n (PRn) of four interrupt groups (n = 0 to 3). 0 is the lowest priority, while 3 is the highest.

PRn		Priority
0	0	0 (Lowest)
0	1	1
1	0	2
1	1	3 (Highest)

Caution Do not use the same priority between groups; otherwise, the operation is not guaranteed.

Table 4-3. Relation between Priority, Exception Code, Handler Address, and Interrupt Priority

Priority of Each Group (setting of IGP)	Priority in Group (Fixed)	Exception Code	Handler Address		Handler Name[Note]		Interrupt Priority
			HCCW. IHA = 0	HCCW. IHA = 1	HCCW. IHA = 0	HCCW. IHA = 1	
3 (highest)	3 (highest)	FEF0H	FFFFFEF0H	FE0000F0H	INT0F	INT1F	Highest
	2	FEE0H	FFFFFEE0H	FE0000E0H	INT0E	INT1E	
	1	FED0H	FFFFFED0H	FE0000D0H	INT0D	INT1D	
	0 (lowest)	FEC0H	FFFFFEC0H	FE0000C0H	INT0C	INT1C	
2	3	FEB0H	FFFFFEB0H	FE0000B0H	INT0B	INT1B	
	2	FEA0H	FFFFFEA0H	FE0000A0H	INT0A	INT1A	
	1	FE90H	FFFFFE90H	FE000090H	INT09	INT19	
	0	FE80H	FFFFFE80H	FE000080H	INT08	INT18	
1	3	FE70H	FFFFFE70H	FE000070H	INT07	INT17	
	2	FE60H	FFFFFE60H	FE000060H	INT06	INT16	
	1	FE50H	FFFFFE50H	FE000050H	INT05	INT15	
	0	FE40H	FFFFFE40H	FE000040H	INT04	INT14	
0 (lowest)	3	FE30H	FFFFFE30H	FE000030H	INT03	INT13	
	2	FE20H	FFFFFE20H	FE000020H	INT02	INT12	
	1	FE10H	FFFFFE10H	FE000010H	INT01	INT11	
	0	FE00H	FFFFFE00H	FE000000H	INT00	INT10	Lowest

Note Handler names used in development tools or softwarwe.

4.6.2 Interrupt clear register (ICR)

This register clears interrupt requests.

Figure 4-7. Interrupt Clear Register (ICR)

Address: C00000C2H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ICR	0	CLR14	CLR13	CLR12	CLR11	CLR10	CLR9	CLR8	CLR7	CLR6	CLR5	CLR4	CLR3	CLR2	CLR1	CLR0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Description
14 - 0	CLR14 - CLR0	Clear Interrupt Request Clears the corresponding interrupt request (REQn bit of IRR register) when this register is manipulated. An interrupt request in the level mode cannot be cleared by these bits, and these bits can be only written. These bits are 0 when they are read, regardless of the ICR register. 0: Performs nothing. 1: Clears REQn bit of IRR register (n = 0 to 14).

4.6.3 Interrupt request register (IRR)

This register holds interrupt requests.

Figure 4-8. Interrupt Request Register (IRR)

Address: C00000C4H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRR	0	REQ14	REQ13	REQ12	REQ11	REQ10	REQ9	REQ8	REQ7	REQ6	REQ5	REQ4	REQ3	REQ2	REQ1	REQ0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Description
14 - 0	REQ14 - REQ0	Interrupt Request These bits are set when the corresponding interrupt requests are issued. They are not affected by the mask register. If an interrupt is triggered by edge, these bits are reset by manipulation of the ICR register. The IRR register can be read only and cannot be written. 0: No interrupt request is generated. 1: Interrupt request is generated.

4.6.4 Interrupt request mask register (IMR)

This register masks interrupt requests.

Figure 4-9. Interrupt Request Mask Register (IMR)

Address: C00000C6H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IMR	1	MSK14	MSK13	MSK12	MSK11	MSK10	MSK9	MSK8	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Description
14 - 0	MSK14 - MSK0	Mask Interrupt Request These bits mask the corresponding interrupt requests. Mask the interrupts when they are not occurring. 0: Interrupt not masked. 1: Masks interrupt.

Table 4-4. Correspondence between Each Bit of Interrupt Control Registers and Interrupt Request Signals

Bits of ICR Register	Bits of IRR Register	Bits of IMR Register	Interrupt Request Signals
CLR14	REQ14	MSK14	INTOV1
CLR13	REQ13	MSK13	INTSER
CLR12	REQ12	MSK12	INTP03
CLR11	REQ11	MSK11	INTSR
CLR10	REQ10	MSK10	INTST
CLR9	REQ9	MSK9	INTCSI
CLR8	REQ8	MSK8	INTP02
CLR7	REQ7	MSK7	INTDMA
CLR6	REQ6	MSK6	INTP10/INTCC10
CLR5	REQ5	MSK5	INTP11/INTCC11
CLR4	REQ4	MSK4	INTP01
CLR3	REQ3	MSK3	INTCM4
CLR2	REQ2	MSK2	INTP12/INTCC12
CLR1	REQ1	MSK1	INTP13/INTCC13
CLR0	REQ0	MSK0	INTP00

Caution These bits are independent of the priority of each interrupt group and correspond to fixed interrupts.

4.6.5 ICU mode register (IMOD)

This register specifies the trigger mode of the interrupt requests input from external pins (INTP00 through INTP03 and INTP10 through INTP13). Two trigger modes, level trigger and edge trigger, can be used.

(1) Level trigger

In this trigger mode, the external interrupt request is sampled at each clock. When an interrupt request is issued, hold the active level of the interrupt (high level) until the interrupt handler of the CPU recognizes the interrupt request. If the interrupt request is cleared before it is recognized, malfunctioning such as an undefined branch destination vector occurs. The interrupt request register (IRR) cannot be cleared by the interrupt clear register (ICR).

(2) Edge trigger

In this trigger mode, the external interrupt is sampled at the rising edge of clock. If the interrupt request signal changes at the edge specified by the IMOD register at the rising edge of the clock, the interrupt request is accepted. The interrupt request is counted only once even when it is input repeatedly. Because the internal interrupt is also input at an edge, clear the corresponding bit of the interrupt request register (IRR) in the interrupt servicing routine.

Caution Keep the interrupt request level for the duration of at least 3 bus clocks in the edge trigger mode because of limitations of the internal edge detection circuit.

Figure 4-10. ICU Mode Register (IMOD)

Address: C00000C8H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IMOD	ITM13		ITM12		ITM11		ITM10		ITM03		ITM02		ITM01		ITM00	
Initial value:	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
R/W:	R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	

Bit	Bit Name	Description
15 - 8	ITM13 - ITM10	Interrupt Trigger Mode1 These bits specify the trigger mode of the INTP1n pin. Change the setting of the trigger mode only when the interrupt request is not going to occur (n = 0 to 3). The INTP1n pin is multiplexed with the interrupt for capture mode of timer 1 of the RPU. Therefore, the valid trigger mode differs as follows depending on the setting of the TUM1 register of the RPU. **(1) When CMS1n = 1 and IMS1n = 1 is set to TUM1 register (n = 0 to 3)** ITM1n / Trigger mode 0 0 Level trigger (high active) 0 1 Reserved (setting prohibited) 1 0 Rising edge trigger 1 1 Reserved (setting prohibited) **(2) When CMS1n = 1 and IMS1n = 0 is set to TUM1 register (n = 0 to 3)** ITM1n / Trigger mode 0 0 Reserved (setting prohibited) 0 1 Reserved (setting prohibited) 1 0 Rising edge trigger 1 1 Reserved (setting prohibited) **(3) When CMS1n = 0 is set to TUM1 register (n = 0 to 3)** ITM1n / Trigger mode 0 0 Reserved (setting prohibited) 0 1 Reserved (setting prohibited) 1 0 Rising edge trigger 1 1 Both rising and falling edge triggers **Caution Do not change the setting of ITM13 through ITM10 while the timer 1 of RPU is operating.**
7 - 0	ITM03 - ITM00	Interrupt Trigger Mode0 These bits set the trigger mode of the INTP0n pin. Change the setting of the trigger mode only when the interrupt request is not going to occur (n = 0 to 3). ITM0n / Trigger mode 0 0 Level trigger (high active) 0 1 Reserved (setting prohibited) 1 0 Rising edge trigger 1 1 Reserved (setting prohibited)

★ 4.7 Interrupt Requests by External Input Pins

An interrupt request is input by external input pin INTP0n or INTP1n.

To input an interrupt request from an external input pin, set the following registers:

- Interrupt request mask register (IMR) (Refer to **4.6.4.**)
- ICU mode register (IMOD) (Refer to **4.6.5.**)
- Timer unit mode register (TUMI) (Refer to **10.3.1.**)

(1) When using INTP0n pin

- Set the trigger mode by using the ICU mode register (IMOD).
- Unmask “interrupt request” by using the interrupt request mask register.

(2) When using INTP1n pin as external input interrupt pin

- Set the trigger mode by using the ICU mode register (IMOD).
- Unmask the interrupt request by using the interrupt request mask register.
- Set the IMS bit of the control register and timer unit mode register (TUM1) of timer 1 to 1. (External input signals are used as interrupt request signals.)

INTP1n is multiplexed with the capture function of timer 1. To use the capture function of timer 1, the INTP1n pin corresponding to the capture register to be used inputs a capture trigger (capture interrupt). The request signal and vector address of INTP1n are multiplexed with the coincidence interrupt INTCC1n of the compare register. When using the coincidence interrupt of the compare register, therefore, the INTP1n pin function cannot be used.

For details, refer to **CHAPTER 10 TIMER/COUNTER FUNCTION**.

Remark n = 0 to 3

CHAPTER 5 BUS CONTROL FUNCTION

The BCU of the V831 can directly connect to EDO DRAM (Extended Data Output DRAM), Page-ROM, SRAM (ROM), or I/O.

To access EDO DRAM, four ××CAS signals, and the RAS, OE, and WE signals are used in addition to the address and data buses.

Addresses are output to DRAM from address pins by multiplexing row and column addresses.

To access Page-ROM or SRAM (ROM), four ××MWR signals, and the MRD and CS signals are used in addition to the address and data buses.

To access an I/O, address bus, data bus, IOWR, IORD, and CS signals are used.

When accessing Page-ROM, SRAM (ROM), or I/O, wait control can be also performed by using the READY signal.

For access arbitration with an external bus master, the HLDRQ and HLDAK signals are used.

Remarks 1. ××CAS : UUCAS, ULCAS, LUCAS, LLCAS

2. ××MWR : UUMWR, ULMWR, LUMWR, LLMWR

5.1 Features

- Directly connects to EDO DRAM, Page-ROM, SRAM (ROM), or I/O
- CAS access with 1 bus clock minimum
- DRAM byte access control with four CAS signals
- Wait control by READY signal

5.2 External I/O Cycle

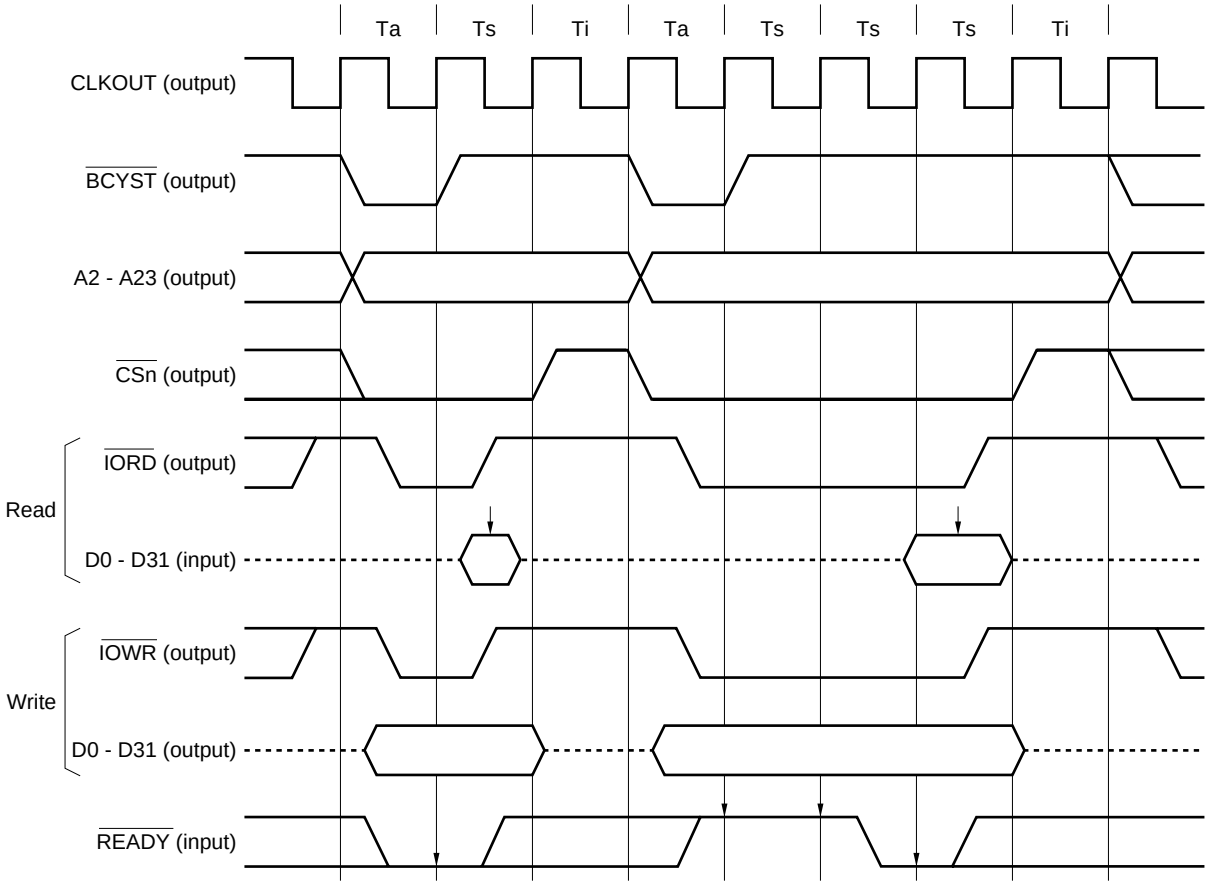
★ The I/O cycle is executed if block 3 to 6 of the I/O space is accessed by the IN/OUT instructions and if the I/O cycle is selected by the BCTC register. The I/O cycle is a single cycle only, and the basic cycle is a 2-bus clock cycle of Ta and Ts states (refer to **Figure 5-1**).

The Ta state starts output of control signals to an external device. An address is output at the rising edge of the bus clock, and the CS signal is asserted active. The BCYST signal is active in the Ta state period (1 bus clock). When the external device is read or written, the IORD or IOWR signal is asserted active at the falling edge of the bus clock. Write data is output at the rising edge of the bus clock.

The Ts state reads or writes data to an external device. During read, the data is sampled at the falling edge of the bus clock. Output of write data continues during the Ts state period. The READY signal is sampled at the rising edge of the bus clock of the Ts state. When the READY signal is active, the IORD or IOWR signal deasserted inactive, and the read/write cycle is completed. If the READY signal is inactive, the Ts state is executed once again.

After the I/O read cycle is executed by the CPU, a Ti state is forcibly inserted for the duration of one bus clock. The wait states can be also controlled by using the PWC0/PWC1 register, in addition to the READY signal. Because the number of wait states set by the PWC0/PWC1 register is ORed with the number of wait states input by the READY signal, whichever number of wait states is greater is inserted.

Figure 5-1. External I/O Cycle (32-bit bus mode)



Remarks 1. n = 3 to 6

2. The dotted lines indicate the high-impedance state.

3. The arrows indicate sampling timing.

5.2.1 Byte access control

Because only the IOWR signal is available as an I/O write strobe signal, accessing the external I/O cannot be controlled in byte units. Therefore, connect the I/O in word units when the data bus width is 32 bits, or in half word units when the data bus width is 16 bits.

The level of A1 is changed when the external I/O is accessed in half-word units via the 32-bit data bus.

5.3 SRAM (ROM) Cycle

The SRAM (ROM) cycle is executed when block 1 to 7 of the memory space is accessed and when the SRAM (ROM) cycle is selected by the BCTC register. The SRAM (ROM) cycle is classified into the following types by the length of data accessed successively and data bus width.

(1) Classification by length of data successively accessed

- Single cycle (access unit: 4/2/1 byte)
  - SRAM (ROM) access by ST instruction execution
  - SRAM (ROM) access by execution of LD instruction to uncacheable area or instruction fetch
  - SRAM (ROM) access by 2-cycle transfer of DMA

- Burst cycle (access unit: 16 bytes)
  - SRAM (ROM) access by refilling of instruction/data cache
  - SRAM (ROM) access by execution of instruction that transfers blocks with internal RAM

(2) Classification by data bus width

- The data bus width is set by the BWn bit of the DBC register (n = 1 to 6).
  - When BWn bit = 1: 16-bit bus mode
  - When BWn bit = 0: 32-bit bus mode

Figure 5-2. Example of Connection of 16M ROM (1M × 16) (in 32-bit bus mode)

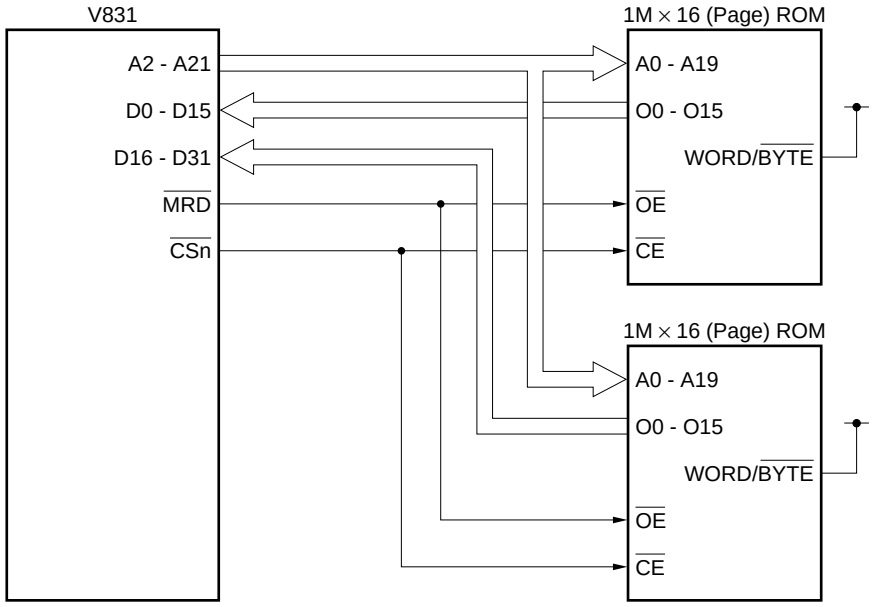
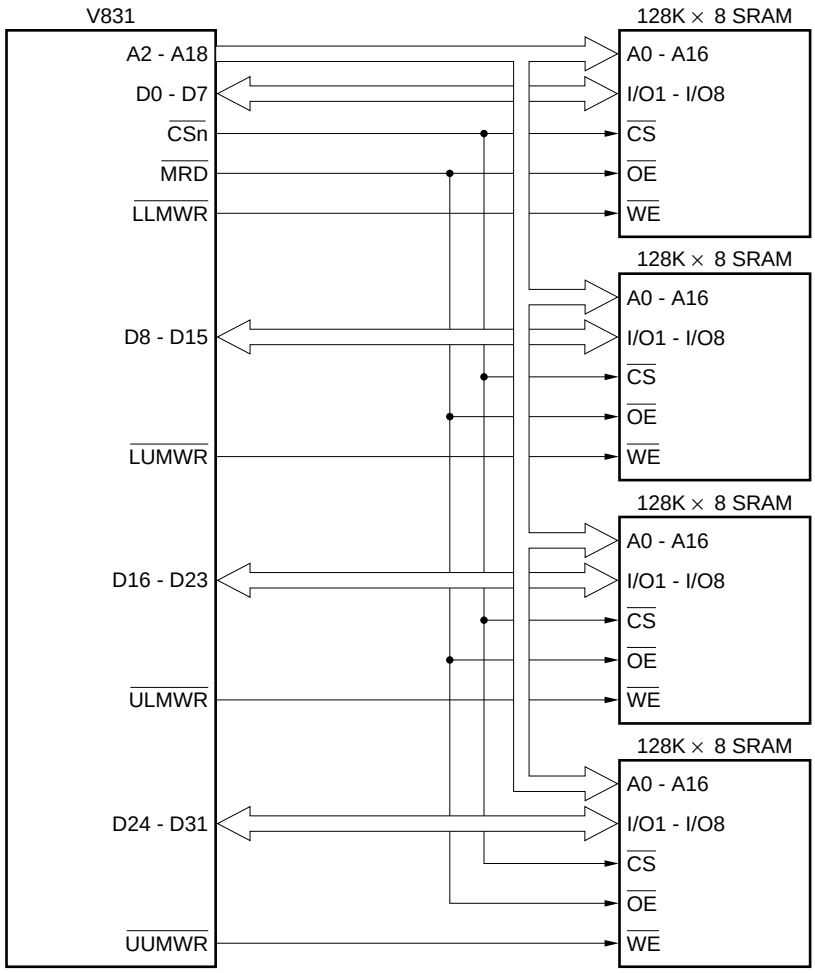


Figure 5-3. Example of Connection of 1M SRAM (128K × 8) (in 32-bit bus mode)



5.3.1 SRAM (ROM) single cycle

The SRAM (ROM) cycle is started when a block setting the SRAM (ROM) cycle is accessed by executing the ST instruction or uncacheable LD instruction, or by instruction fetch or DMA2 cycle transfer. The basic cycle is a 2-bus clock cycle of Ta and Ts states (refer to **Figure 5-4**).

The Ta state starts output of control signals to an external device. An address is output at the rising edge of the bus clock, and the $\overline{CS}$ signal is asserted active. The $\overline{BCYST}$ signal is active in the Ta state period (1 bus clock). When the external device is read or written, the $\overline{MRD}$ or $\overline{\times\times MWR}$ signal is asserted active at the falling edge of the bus clock. Write data is output at the rising edge of the bus clock.

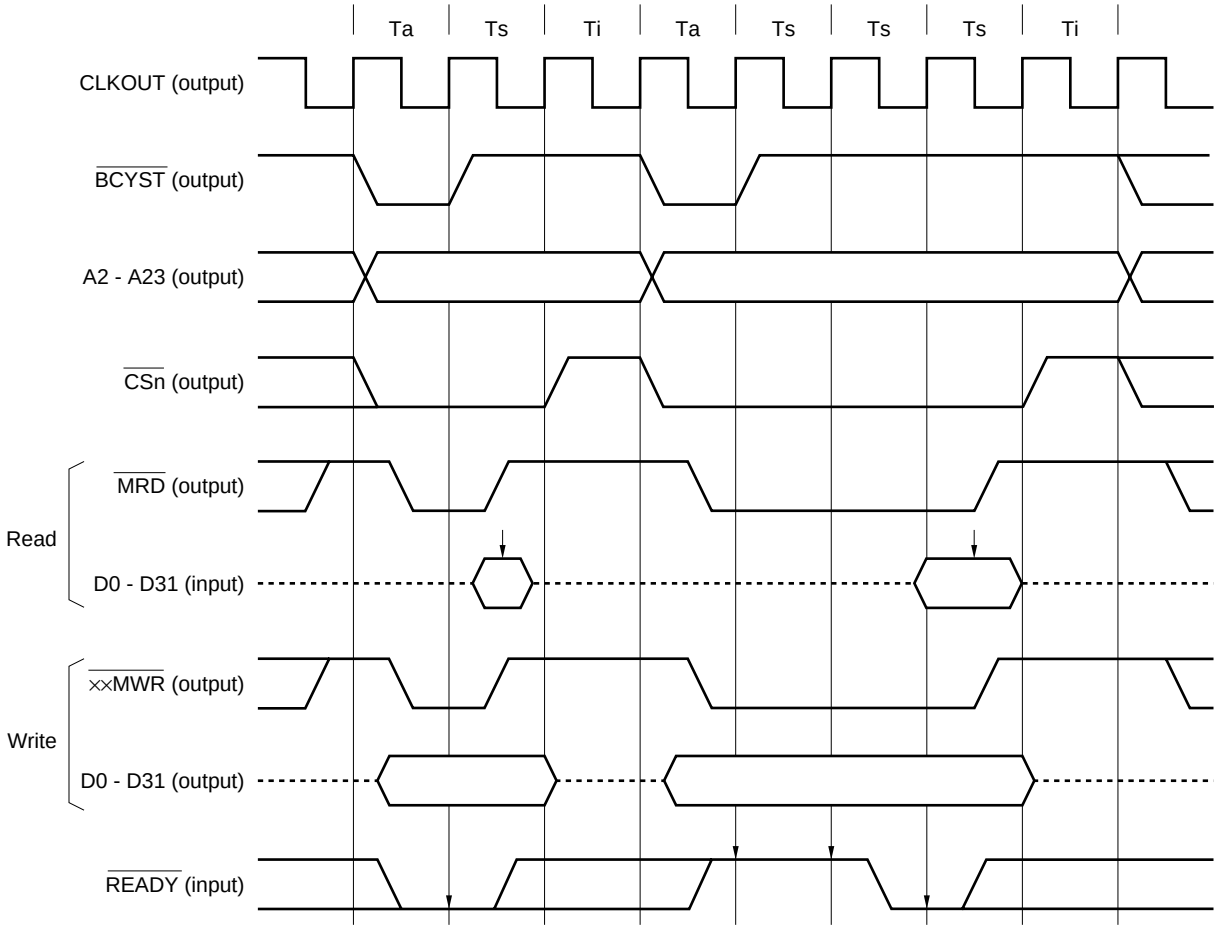
The Ts state reads or writes data of an external device. During read, the data is sampled at the falling edge of the bus clock. Output of write data continues during the Ts state period. The $\overline{READY}$ signal is sampled at the rising edge of the bus clock of the Ts state. When the $\overline{READY}$ signal is asserted active, the cycle is completed with the $\overline{MRD}$ or $\overline{\times\times MWR}$ signal deasserted inactive. If the READY signal is deasserted inactive, the Ts state is executed once again.

After the SRAM (ROM) read cycle executed by the CPU, a Ti state is forcibly inserted for the duration of one bus clock.

The wait states can be also controlled by using the PWC0/PWC1 register, in addition to the $\overline{READY}$ pin (refer to **Figure 5-4**). Because the number of wait states set by the PWC0/PWC1 register is ORed with the number of wait states input by the $\overline{READY}$ signal, whichever number of wait states is greater is inserted.

Remark $\overline{\times\times MWR}$: $\overline{UUMWR}$, $\overline{ULMWR}$, $\overline{LUMWR}$, $\overline{LLMWR}$

Figure 5-4. SRAM (ROM) Single Cycle (32-bit bus mode)



Remarks 1. n = 1 to 7

2. ××MWR: LLMWR, LUMWR, ULMWR, UUMWR

3. The dotted lines indicate the high-impedance state.

4. The arrows indicate sampling timing.

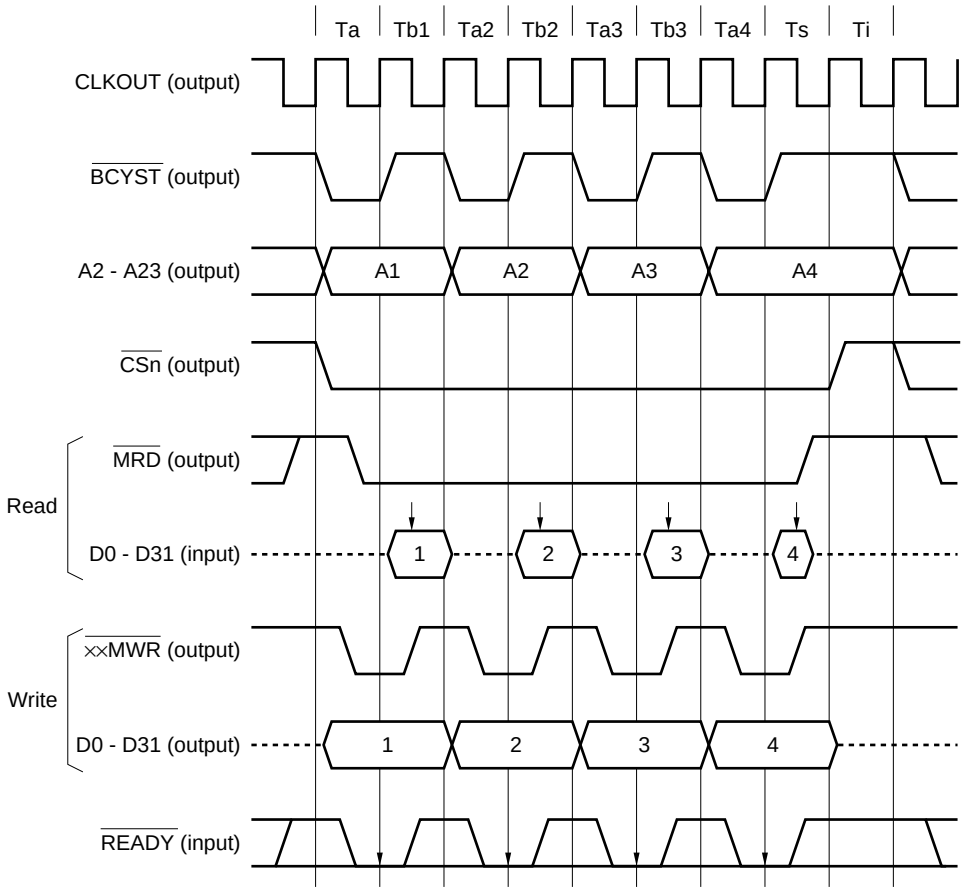
5.3.2 SRAM (ROM) burst cycle

The SRAM (ROM) burst cycle is started when block setting the SRAM (ROM) cycle is accessed by refilling the instruction/data cache or executing an instruction that transfers blocks with the internal RAM. The basic access is a 2-bus clock access. In the 32-bit bus mode, access is made four times in a row in one burst cycle (refer to **Figure 5-5**).

The timing of the SRAM (ROM) control signals for one access is the same as that in the single cycle, except the $\overline{\text{MRD}}$ signal. The $\overline{\text{MRD}}$ signal remains active during the period of the burst cycle in synchronization with the rising of the bus clock. The $\overline{\text{BCYST}}$ signal remains active during the period of Ta to Ta4 states for each access. The $\overline{\text{READY}}$ signal is sampled at the rising edge of the bus clock of the Ts state.

The wait states can be also controlled by using the PWC0/PWC1 register, in addition to the $\overline{\text{READY}}$ pin. Because the number of wait states set by the PWC0/PWC1 register is ORed with the number of wait states input by the $\overline{\text{READY}}$ signal, whichever number of wait states is greater is inserted.

Figure 5-5. SRAM (ROM) Burst Cycle



Remarks 1. n = 1 to 7

2. ××MWR: LLMWR, LUMWR, ULMWR, UUMWR

3. The dotted lines indicate the high-impedance state.

4. The arrows indicate sampling timing.

5.3.3 Byte access control

Byte access is controlled by using four $\overline{\times\times MWR}$ signals ($\overline{UUMWR}$, $\overline{ULMWR}$, $\overline{LUMWR}$, and $\overline{LLMWR}$). When the data bus width is 32 bits, the four $\overline{\times\times MWR}$ signals are used; when it is 16 bits, two $\overline{\times\times MWR}$ signals ($\overline{LUMWR}$ and $\overline{LLMWR}$) are used.

The relation between the $\overline{\times\times MWR}$ signals and access address is shown below.

Table 5-1. 32-Bit Data Bus ($\overline{\times\times MWR}$)

Data Size	Address		$\overline{\times\times MWR}$			
	A1	A0	$\overline{UUMWR}$	$\overline{ULMWR}$	$\overline{LUMWR}$	$\overline{LLMWR}$
Byte	0	0	1	1	1	0
	0	1	1	1	0	1
	1	0	1	0	1	1
	1	1	0	1	1	1
Half word (16 bits)	0	0	1	1	0	0
	1	0	0	0	1	1
Word (32 bits)	0	0	0	0	0	0

Remarks 1: High-level output
0: Low-level output

Table 5-2. 16-Bit Data Bus ($\overline{\times\times MWR}$)

Data Size		Address	$\overline{\times\times MWR}$			
		A0	$\overline{UUMWR}$	$\overline{ULMWR}$	$\overline{LUMWR}$	$\overline{LLMWR}$
Byte		0	1	1	1	0
		1	1	1	0	1
Half word (16 bits)		0	1	1	0	0
Word (32 bits)	First	0	1	1	0	0
	Second	0	1	1	0	0

Remarks 1: High-level output
0: Low-level output

5.4 Page-ROM Cycle

When the block 7 area of the memory space is accessed with the Page-ROM cycle selected by the CT7 bit of the BCTC register, the Page-ROM cycle is executed. The Page-ROM cycle is classified into the following types by the length of data accessed successively and the data bus width.

(1) Classification by length of data accessed successively

- **Single cycle (access unit: 4/2/1 byte)**
  - Page-ROM access by execution of LD instruction to uncacheable area or instruction fetch
  - Page-ROM access by 2-cycle transfer of DMA

- **Burst cycle (access unit: 16 bytes)**
  - Page-ROM access by refilling instruction cache
  - Page-ROM access by execution of instruction that transfers blocks with internal RAM

(2) Classification by data bus width

- The data bus width is set by using the BT16B pin. Access is made four times in a row in the burst cycle and in the 32-bit bus mode. In the 16-bit bus mode, access is made eight times in a row.
  - When BT16B pin = 1: 16-bit bus mode
  - When BT16B pin = 0: 32-bit bus mode

5.4.1 Page-ROM single cycle

This cycle is started to access block 7 for which the Page-ROM cycle is selected by execution of the uncacheable LD instruction, instruction fetch, or DMA2 cycle transfer. Wait states can be controlled by using the $\overline{\text{READY}}$ pin or an internal register. To control wait states, the WS7 bit of he PWC1 register is used. The bus timing is the same as the SRAM (ROM) single cycle.

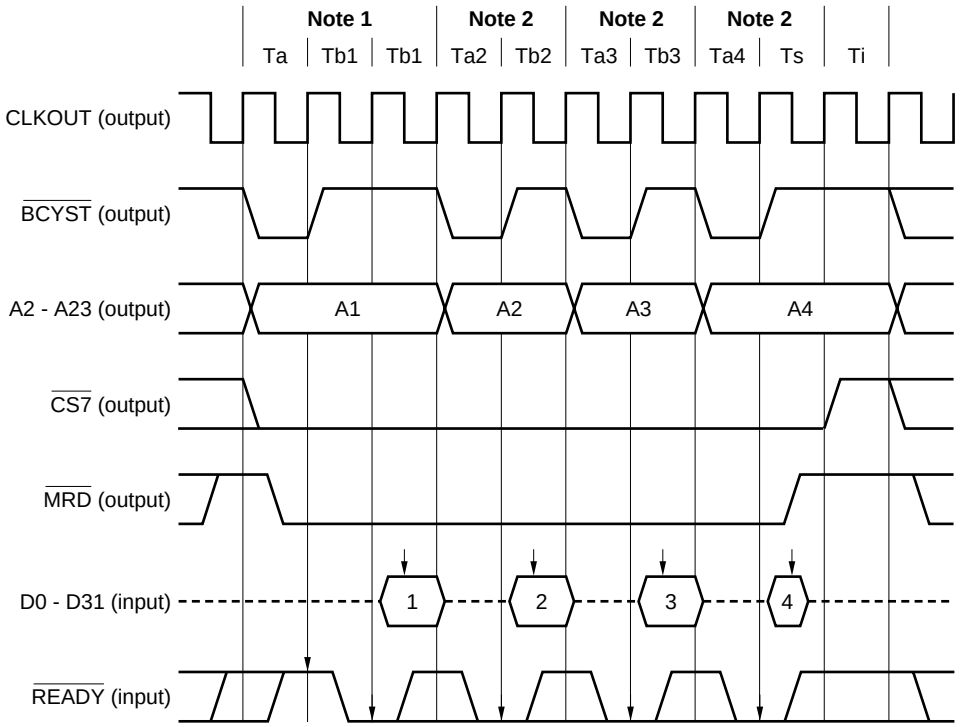
5.4.2 Page-ROM burst cycle

This cycle is started to access block 7 for which the Page-ROM cycle is selected by refilling the instruction cache or executing an instruction that transfers blocks with the internal RAM. The bus timing is the same as the SRAM (ROM) burst cycle, but the number of wait states is set differently.

If the PS bit of the PRC register is 0 (page size is 16 bytes) in the 32-bit bus mode or 16-bit bus mode, the normal access (off-page) is executed as the first access, and on-page access is executed from the second access and onward. The basic cycle of one access is 2-bus clock cycles. (Refer to **Figure 5-6**.) At the first off-page access, the wait states are controlled by the WS7 bit of the PWC1 register. At the second on-page access and onward, the wait states are controlled by the PWS bit of the PR register. The wait states can be also controlled by the $\overline{\text{READY}}$ pin.

If the PS bit of the PRC register is 1 (page size is 8 bytes) in the 16-bit bus mode, the normal access (off-page) is executed at the first and fifth accesses, and on-page access is executed for the second through fourth, and sixth through eighth accesses. The number of wait states for each access is the same as that in the 32-bit bus mode.

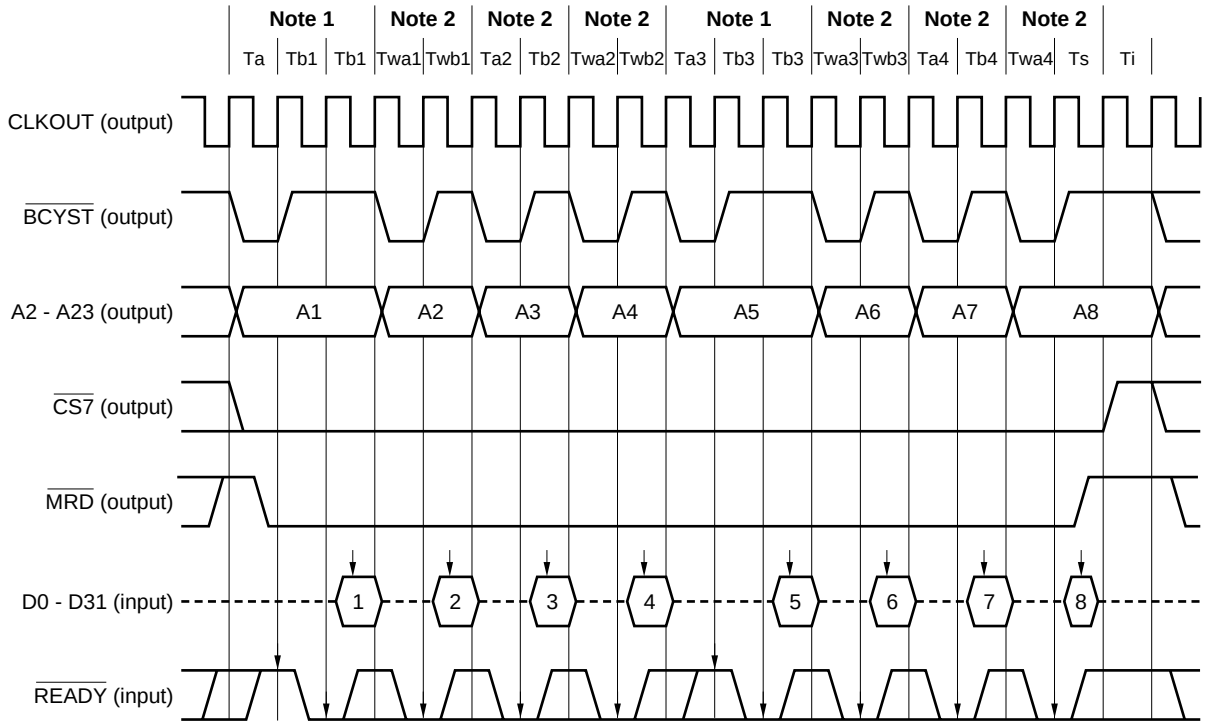
Figure 5-6. Page-ROM Burst Cycle (32-bit bus mode)



Notes 1. off-page access
2. on-page access

Remarks 1. The dotted lines indicate the high-impedance state.
2. The arrows indicate the sampling timing.

Figure 5-7. Page-ROM Burst Cycle (16-bit bus mode, 8-byte page size)



Notes 1. off-page access

2. on-page access

Remarks 1. The dotted lines indicate the high-impedance state.

2. The arrows indicate the sampling timing.

5.5 DRAM Cycle

The EDO DRAM cycle is executed when the block 0 area of the memory space is accessed. The EDO DRAM cycle is classified into the following types by the length of data successively accessed, on-page/off-page, $\overline{CAS}$ cycle period, and data bus width.

(1) Classification by length of data successively accessed

- **Single cycle (access unit: 4/2/1 byte)**
  - DRAM access by execution of ST Instruction
  - DRAM access by execution of LD instruction to uncacheable area or instruction fetch
  - DRAM access by 2-cycle transfer of DMA

- **Burst cycle (access unit: 16 bytes)**
  - DRAM access by refilling instruction/data cache
  - DRAM access by execution of instruction that transfers blocks with internal RAM

(2) Classification by on-page/off-page

- **on-page cycle**
  - $\overline{RAS}$ active and same row address as preceding DRAM cycle

- **off-page cycle**
  - $\overline{RAS}$ inactive
  - $\overline{RAS}$ active and different row address from preceding DRAM cycle

(3) Classification by $\overline{CAS}$ cycle period

The $\overline{CAS}$ cycle period is set by the CRWT bit of the DRC register during read/write.

- **1-clock $\overline{CAS}$ cycle**
  - CRWT bit = 00
  - CRWT bit = 10 (write cycle only)

- **2-clock $\overline{CAS}$ cycle**
  - CRWT bit = 11
  - CRWT bit = 10 (read cycle only)

★ **(4) Classification by data bus width**

The data bus width is set by the BW0 bit of the DBC register.

- When BW0 bit = 0: 32-bit bus mode
- When BW0 bit = 1: 16-bit bus mode

Figure 5-8. Example of Connection with 16M EDO-DRAM (1M × 16) (in 16-bit bus mode)

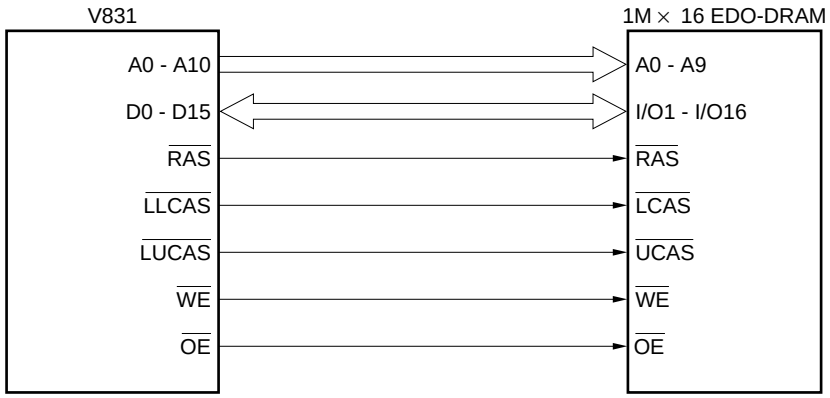
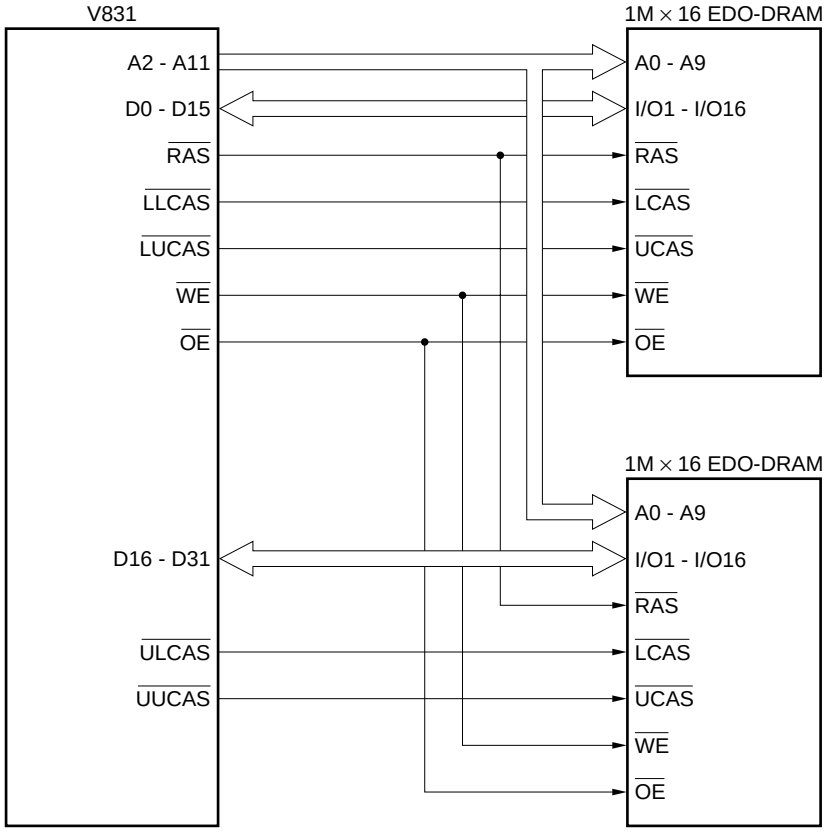


Figure 5-9. Example of Connection with 16M EDO-DRAM (1M × 16) (in 32-bit bus mode)



5.5.1 DRAM single cycle

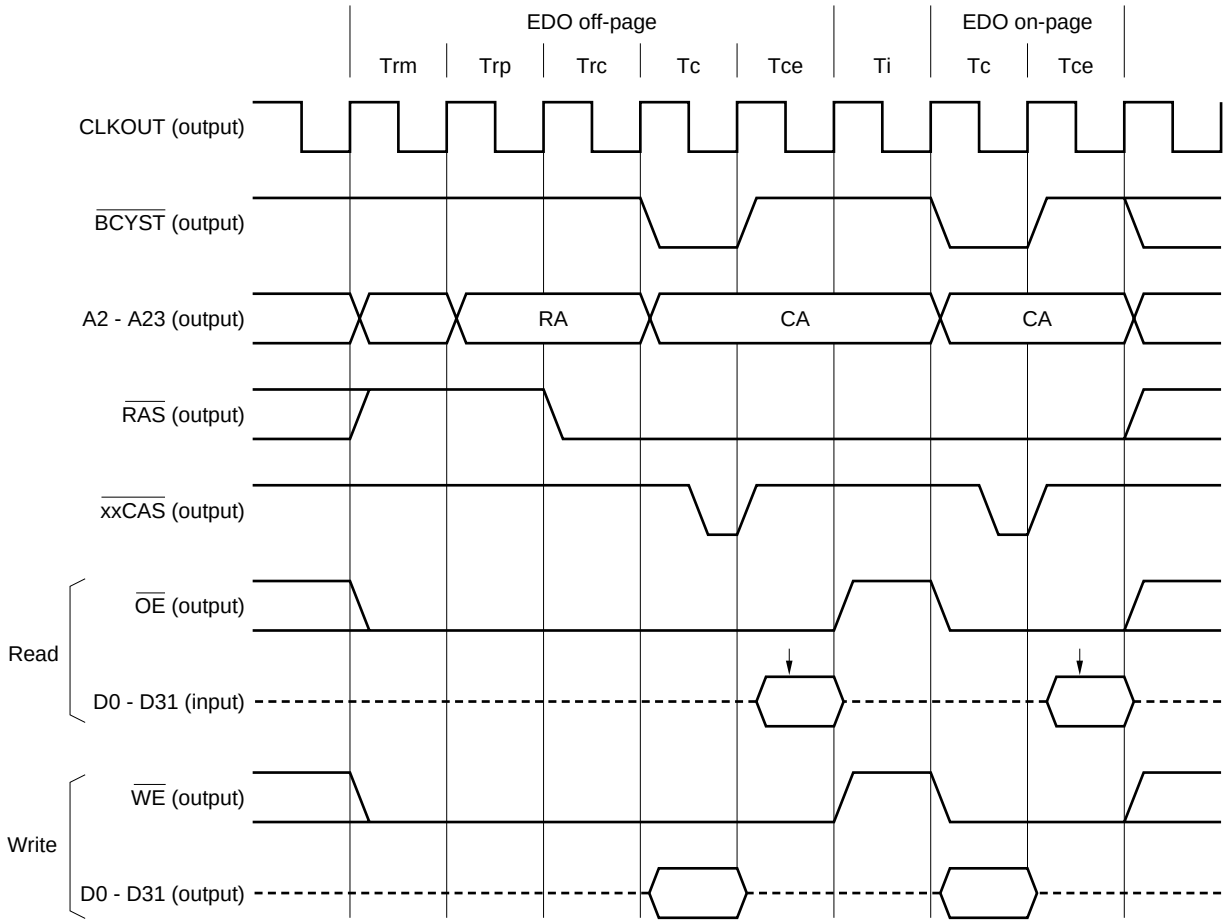
This cycle is started to access block 0 by executing the ST instruction or uncacheable LD instruction, or instruction fetch or DMA2 cycle transfer. Figure 5-10 shows the timing of the single 1-clock $\overline{CAS}$ off-page/on-page cycle, and Figure 5-11 shows the timing of the single 2-clock CAS off-page/on-page cycle.

In the case of the 1-clock $\overline{CAS}$ cycle, the cycle for on-page access is a 2-bus clock cycle of Tc and Tce. Tc is a $\overline{CAS}$ assert state and Tce is a $\overline{CAS}$ cycle end state. Because the $\overline{RAS}$ signal is precharged during off-page access, at least three states, Trm (ROW miss state), Trp ($\overline{RAS}$ precharge state), and Trc ($\overline{RAS}$-$\overline{CAS}$ state), are inserted before the on-page access.

In the case of the 2-clock $\overline{CAS}$ cycle, the cycle for the on-page access is a 3-bus clock cycle of Tca, Tcn, and Tce. Tca is a $\overline{CAS}$ assert state, Tcn is a $\overline{CAS}$ negate state, and Tce is a $\overline{CAS}$ cycle end state. Because $\overline{RAS}$ is precharged during the off-page access, at least three states, Trm (ROW miss state), Trp ($\overline{RAS}$ precharge state), and Trc ($\overline{RAS}$-$\overline{CAS}$ state), are inserted before the on-page access.

One state of Ti cycle is forcibly inserted after the read cycle started by the CPU.

Figure 5-10. DRAM Single 1-Clock $\overline{\text{CAS}}$ on-page/off-page Cycle (32-bit bus mode)

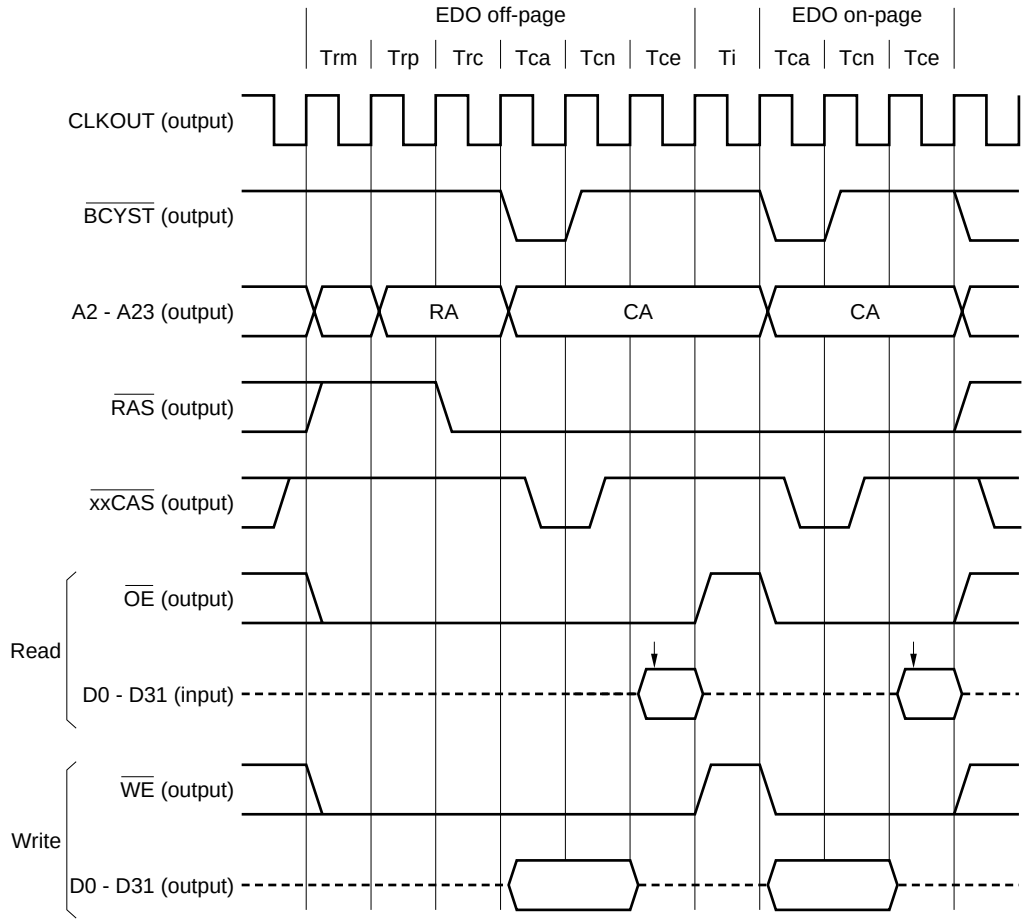


Remarks 1. $\overline{\text{xxCAS}}$: $\overline{\text{LLCAS}}$, $\overline{\text{LUCAS}}$, $\overline{\text{ULCAS}}$, $\overline{\text{UUCAS}}$

2. The dotted lines indicate the high-impedance state.

3. The arrows indicate the sampling timing.

Figure 5-11. DRAM Single 2-Clock $\overline{\text{CAS}}$ on-page/off-page Cycle (32-bit bus mode)



Remarks 1. $\overline{\text{xxCAS}}$: $\overline{\text{LLCAS}}$, $\overline{\text{LUCAS}}$, $\overline{\text{ULCAS}}$, $\overline{\text{UUCAS}}$

2. The dotted lines indicate the high-impedance state.

3. The arrows indicate the sampling timing.

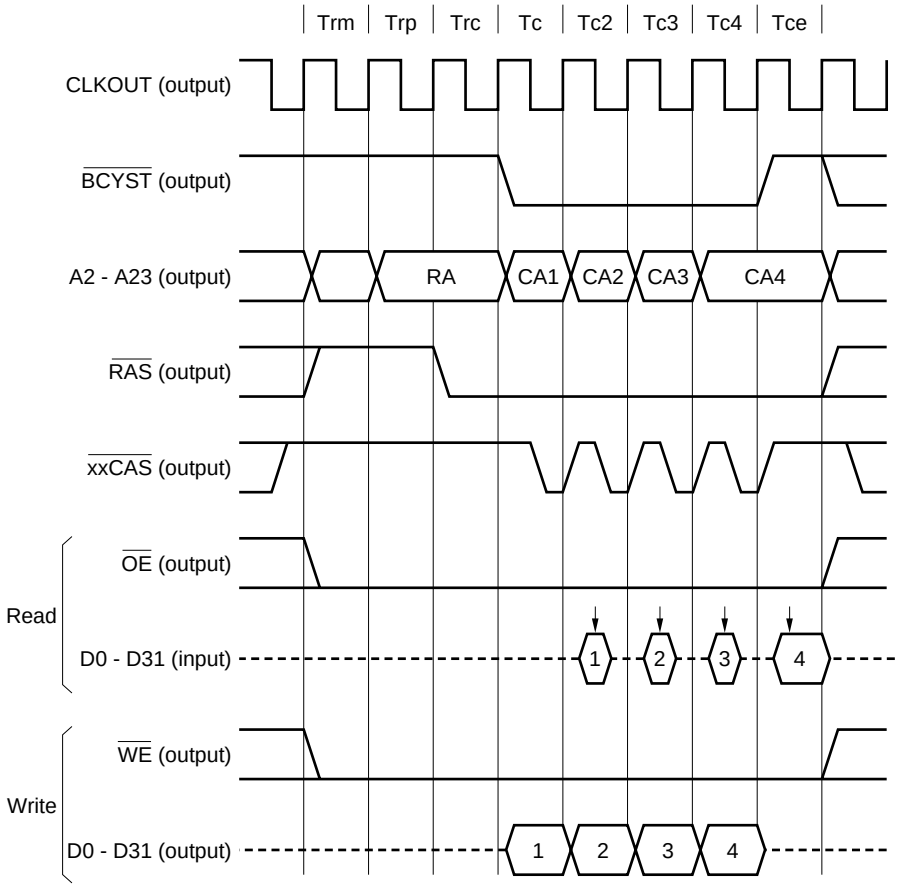
5.5.2 DRAM burst cycle

The DRAM burst cycle is started when block 0 is accessed by refilling the instruction/data cache, or executing an instruction that transfers blocks with the internal RAM. This is a 16-byte successive bus cycle. Page access is executed from the second access and onward. Figure 5-12 shows the burst 1-clock $\overline{CAS}$ off-page cycle, and Figure 5-13 shows the burst 1-clock $\overline{CAS}$ on-page cycle. Figure 5-14 shows the timing of the burst 2-clock $\overline{CAS}$ off-page cycle, and Figure 5-15 shows the timing of the burst 2-clock $\overline{CAS}$ on-page cycle.

In the case of the 1-clock $\overline{CAS}$ cycle, the cycle for on-page access is a 5-bus clock cycle (Tc through Tc4, and Tce) . Tc through Tc4 are the $\overline{CAS}$ assert states for four accesses, and Tce is a $\overline{CAS}$ cycle end state. Because the $\overline{RAS}$ signal is precharged during off-page access, at least three states (Trm, Trp, and Trc) are inserted before the on-page access.

In the case of the 2-clock $\overline{CAS}$ cycle, the cycle for the on-page access is a 9-bus clock cycle (Tca through Tcn4 and Tce). Tca through Tca4 are $\overline{CAS}$ assert state for four accesses, Tcn1 through Tcn4 are $\overline{CAS}$ negate states, and Tce is a $\overline{CAS}$ cycle end state. Because $\overline{RAS}$ is precharged during the off-page access, at least three states (Trm, Trp, and Trc) are inserted before the on-page access.

Figure 5-12. DRAM Burst 1-Clock $\overline{\text{CAS}}$ off-page Cycle (32-bit bus mode)

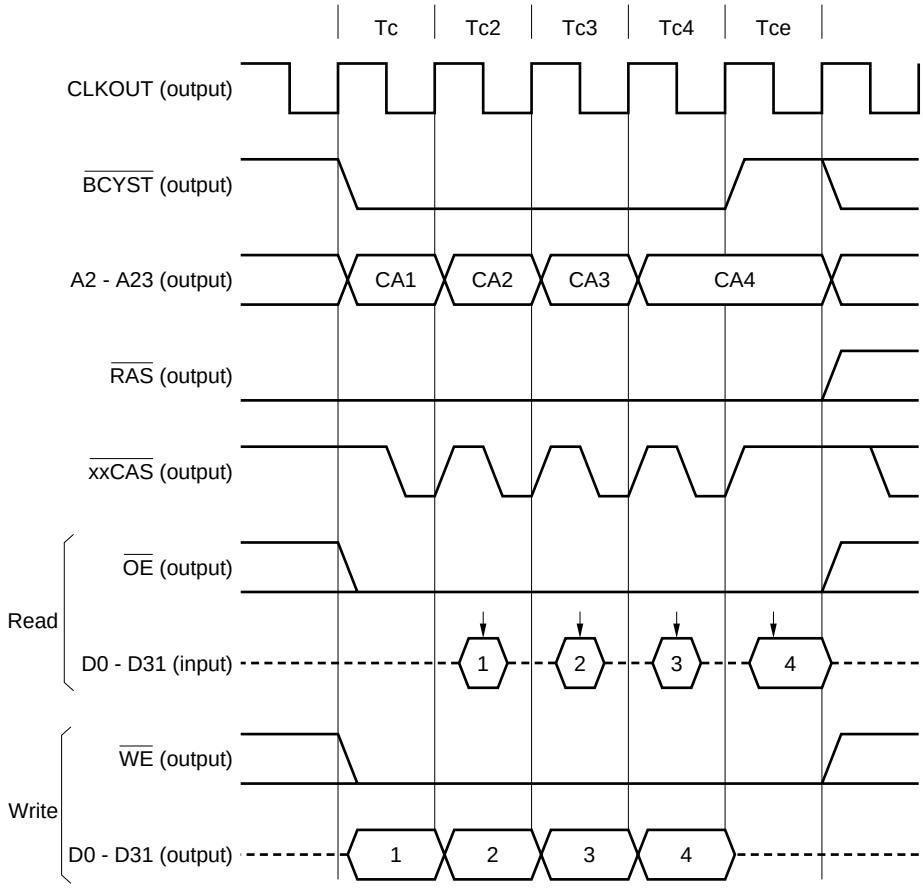


Remarks 1. $\overline{\text{xxCAS}}$: $\overline{\text{LLCAS}}$, $\overline{\text{LUCAS}}$, $\overline{\text{ULCAS}}$, $\overline{\text{UUCAS}}$

2. The dotted lines indicate the high-impedance state.

3. The arrows indicate the sampling timing.

Figure 5-13. DRAM Burst 1-Clock $\overline{\text{CAS}}$ on-page Cycle (32-bit bus mode)

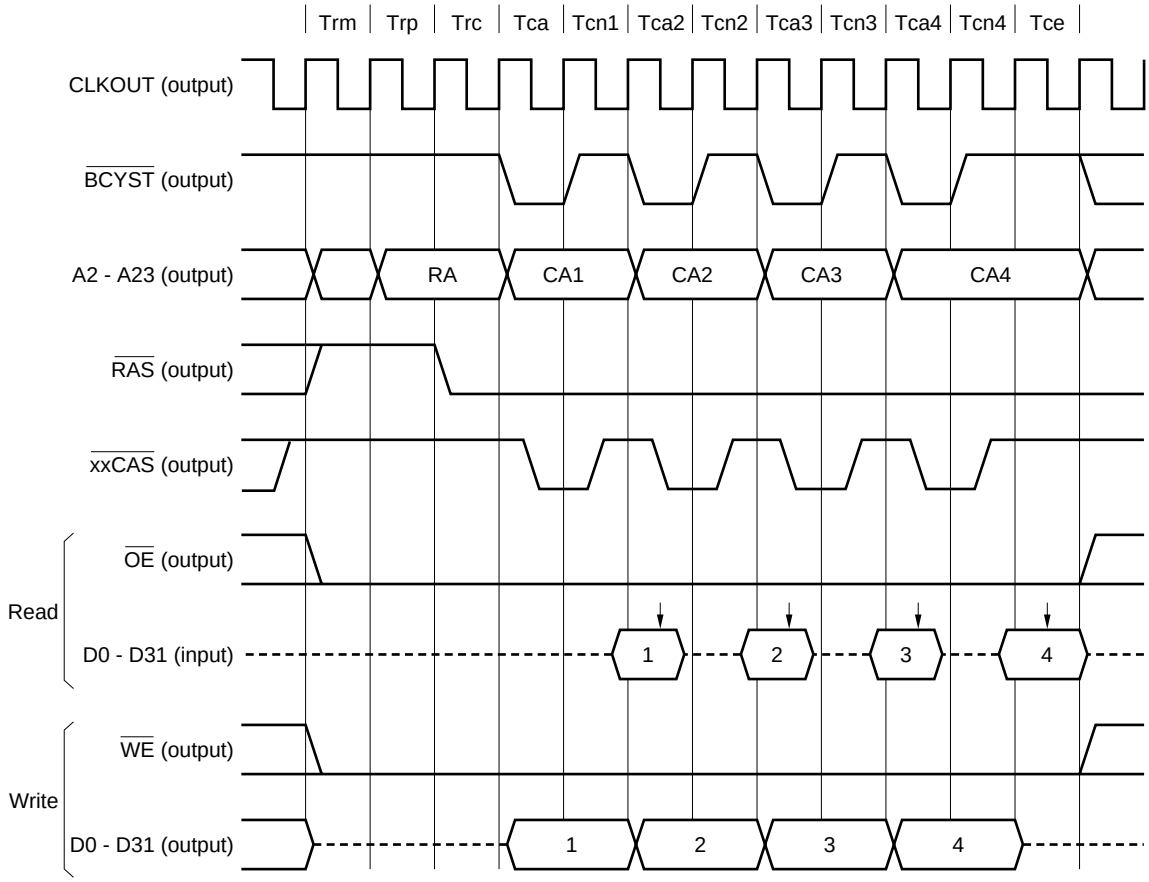


Remarks 1. $\overline{\text{xxCAS}}$: $\overline{\text{LLCAS}}$, $\overline{\text{LUCAS}}$, $\overline{\text{ULCAS}}$, $\overline{\text{UUCAS}}$

2. The dotted lines indicate the high-impedance state.

3. The arrows indicate the sampling timing.

Figure 5-14. DRAM Burst 2-Clock $\overline{\text{CAS}}$ off-page Cycle (32-bit bus mode)

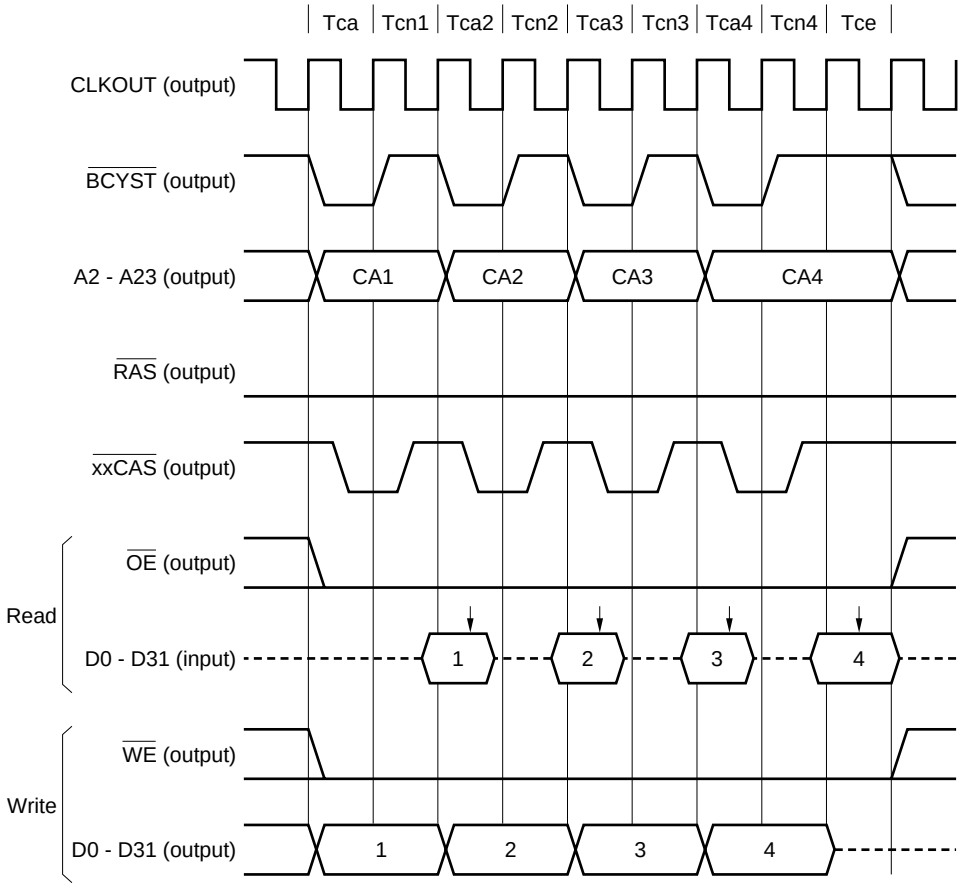


Remarks 1. xxCAS: LLCAS, LUCAS, ULCAS, UUCAS

2. The dotted lines indicate the high-impedance state.

3. The arrows indicate the sampling timing.

Figure 5-15. DRAM Burst 2-Clock $\overline{\text{CAS}}$ on-page Cycle (32-bit bus mode)



Remarks 1. xxCAS: LLCAS, LUCAS, ULCAS, UUCAS

2. The dotted lines indicate the high-impedance state.

3. The arrows indicate the sampling timing.

5.5.3 Timing control

For DRAM access timing control of the V831, a $\overline{\text{RAS}}$ precharge period and $\overline{\text{RAS}}$-$\overline{\text{CAS}}$ delay period can be set in addition to the $\overline{\text{CAS}}$ cycle period. Note, however, that the wait states cannot be controlled by the $\overline{\text{READY}}$ pin during DRAM access.

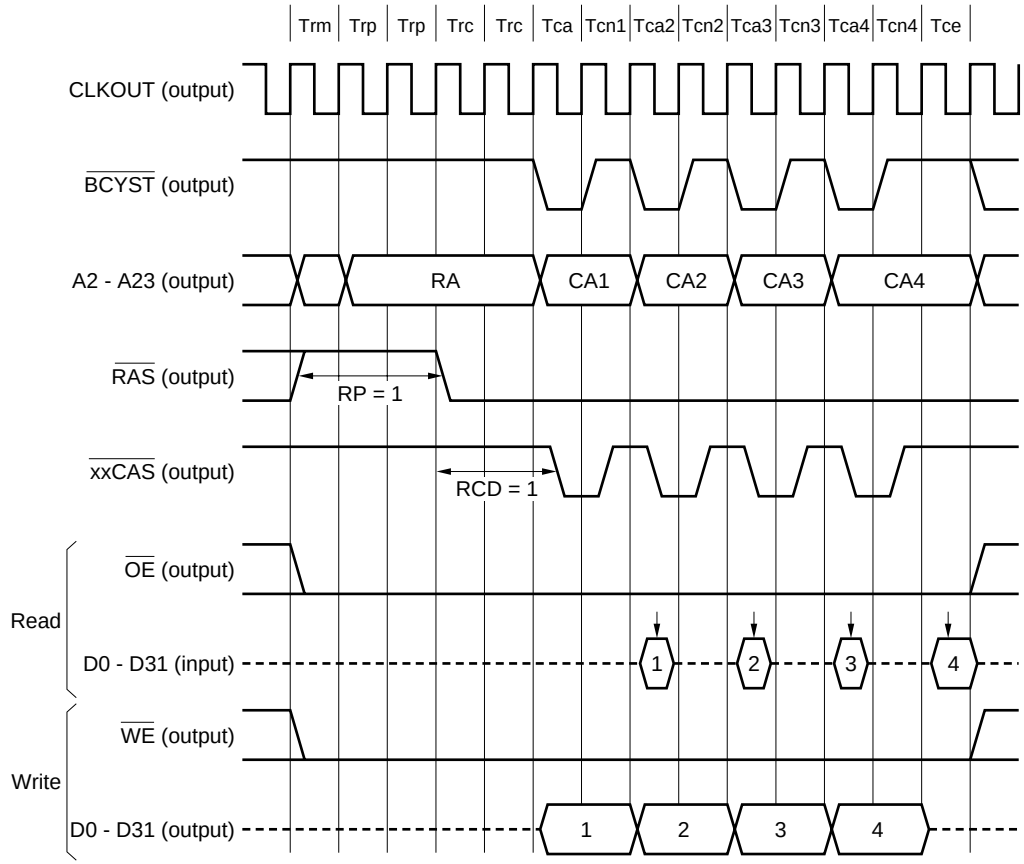
(1) Control during $\overline{\text{RAS}}$ precharge period

The $\overline{\text{RAS}}$ precharge period is set by using the RP bit of the DRC register. When the RP bit is cleared to 0, the $\overline{\text{RAS}}$ precharge period is 2 bus clocks; when the RP bit is set to 1, the $\overline{\text{RAS}}$ precharge period is 3 bus clocks. Figure 5-16 shows the RAS precharge period when the RP bit is set to 1.

(2) $\overline{\text{RAS}}$-$\overline{\text{CAS}}$ delay time

The $\overline{\text{RAS}}$-$\overline{\text{CAS}}$ delay period during read is set by the RCD bit of the DRC register. When the RCD bit is cleared to 0, the $\overline{\text{RAS}}$-$\overline{\text{CAS}}$ delay time is 1.5 bus clocks; when the RCD bit is set to 1, it is 2.5 bus clocks. The timing shown in Figure 5-16 is when the RCD bit is 1.

Figure 5-16. DRAM Access Timing (burst off-page cycle)



Remarks 1. $\overline{\text{xxCAS}}$: $\overline{\text{LLCAS}}$, $\overline{\text{LUCAS}}$, $\overline{\text{ULCAS}}$, $\overline{\text{UUCAS}}$

2. The dotted lines indicate the high-impedance state.

3. The arrows indicate the sampling timing.

5.5.4 Byte access control

The four ××CAS signals (UUCAS, ULCAS, LUCAS, and LLCAS) are used to control byte access. When the data bus is 32 bits wide, all the four ××CAS signals are used. When the data bus is 16 bits wide, however, only two ××CAS signals (LUCAS and LLCAS) are used. Tables 5-3 and 5-4 show the relations between the ××CAS signals and access addresses.

Table 5-3. 32-Bit Data Bus (××CAS)

Data Size	Address		××CAS			
	A1	A0	UUCAS	ULCAS	LUCAS	LLCAS
Byte	0	0	1	1	1	0
	0	1	1	1	0	1
	1	0	1	0	1	1
	1	1	0	1	1	1
Half word (16 bits)	0	0	1	1	0	0
	1	0	0	0	1	1
Word (32 bits)	0	0	0	0	0	0

Table 5-4. 16-Bit Data Bus (××CAS)

Data Size		Address	××CAS			
		A0	UUCAS	ULCAS	LUCAS	LLCAS
Byte		0	1	1	1	0
		1	1	1	0	1
Half word (16 bits)		0	1	1	0	0
Word (32 bits)	First	0	1	1	0	0
	Second	0	1	1	0	0

Remarks 1: High-level output
0: Low-level output

5.5.5 Refresh control

The CBR refresh cycle and CBR self-refresh cycle can be automatically generated.

(1) CBR refresh cycle

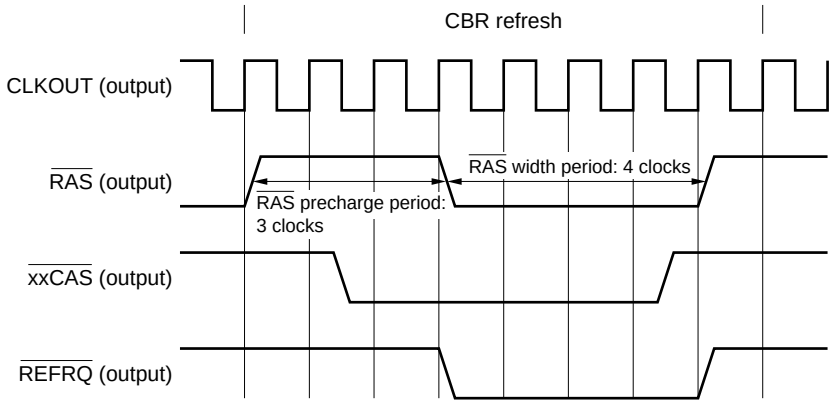
Figure 5-17 shows the timing of the CBR refresh cycle. The $\overline{RAS}$ active period of the CBR refresh cycle can be set by the RFW bit of the RFC register.

The $\overline{RAS}$ precharge period is set by the RP bit of the DRC register. The wait states cannot be controlled by the $\overline{READY}$ pin.

Table 5-5. $\overline{RAS}$ Active Period

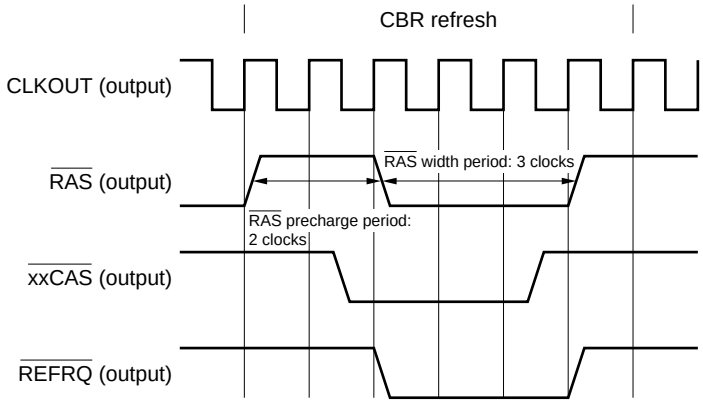
RFC Register		$\overline{RAS}$ Active Period
RFW1	RFW0	
0	0	3 bus clocks
0	1	4 bus clocks
1	0	5 bus clocks

Figure 5-17. CBR Refresh Cycle (when RFW1, 0 = 01, RP = 1)



Remark $\overline{xxCAS}$: $\overline{LLCAS}$, $\overline{LUCAS}$, $\overline{ULCAS}$, $\overline{UUCAS}$

Figure 5-18. CBR Refresh Cycle (when RFW1, 0 = 00, RP = 0)



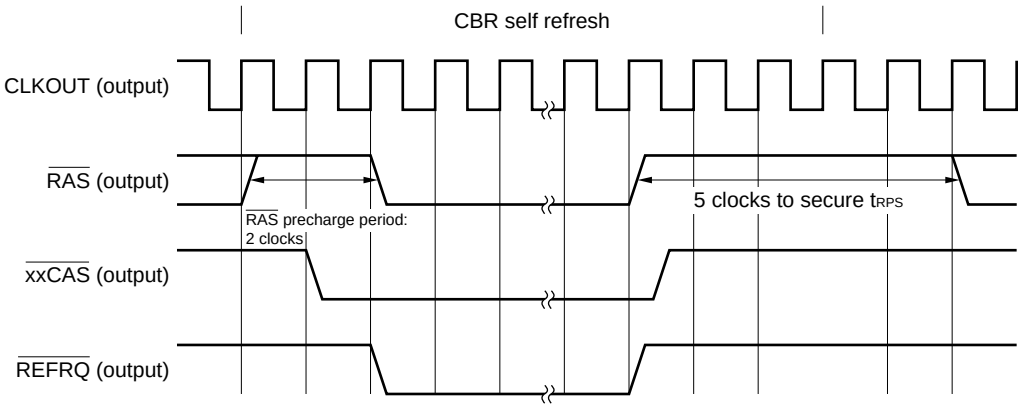
Remark xxCAS: LLCAS, LUCAS, ULCAS, UUCAS

(2) CBR self-refresh cycle

The CBR self-refresh cycle is generated in the STOP mode. The self-refresh cycle is started when the STBY instruction is executed with the REN bit of the RFC register set to 1. The self-refresh cycle is cleared by using the RESET or NMI pin.

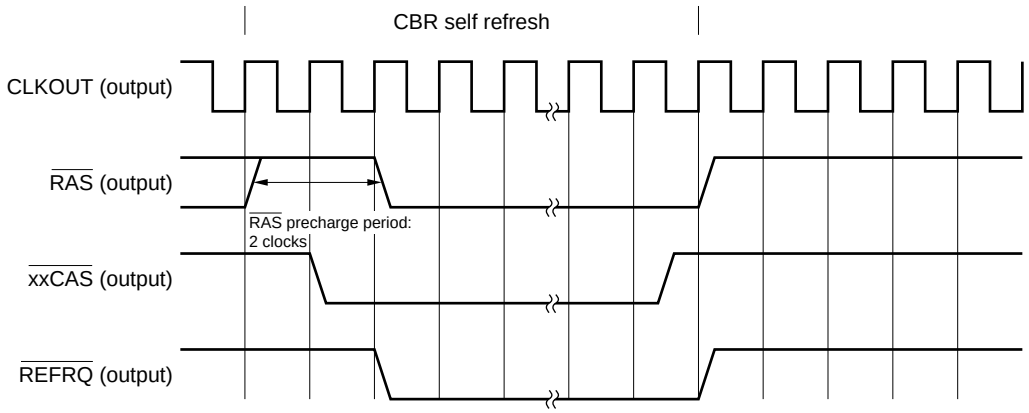
Figure 5-19 shows the timing of the CBR self-refresh cycle. The RAS precharge period (t_{RPS}) after the CBR self refresh is at least 5 bus clocks.

Figure 5-19. CBR Self-Refresh Cycle (when cleared by NMI, RP = 0)



Remark xxCAS: LLCAS, LUCAS, ULCAS, UUCAS

Figure 5-20. CBR Self-Refresh Cycle (when cleared by $\overline{\text{RESET}}$, RP = 0)



Remark $\overline{\text{xxCAS}}$: $\overline{\text{LLCAS}}$, $\overline{\text{LUCAS}}$, $\overline{\text{ULCAS}}$, $\overline{\text{UUCAS}}$

5.6 Idle State

The number of idle states for block n after the read cycle is set by using the ISn bit (n = 0 to 7) of the PIC register. The values of the bus control signals during the idle period are shown below.

Table 5-6. Values of Bus Control Signals during Idle Period

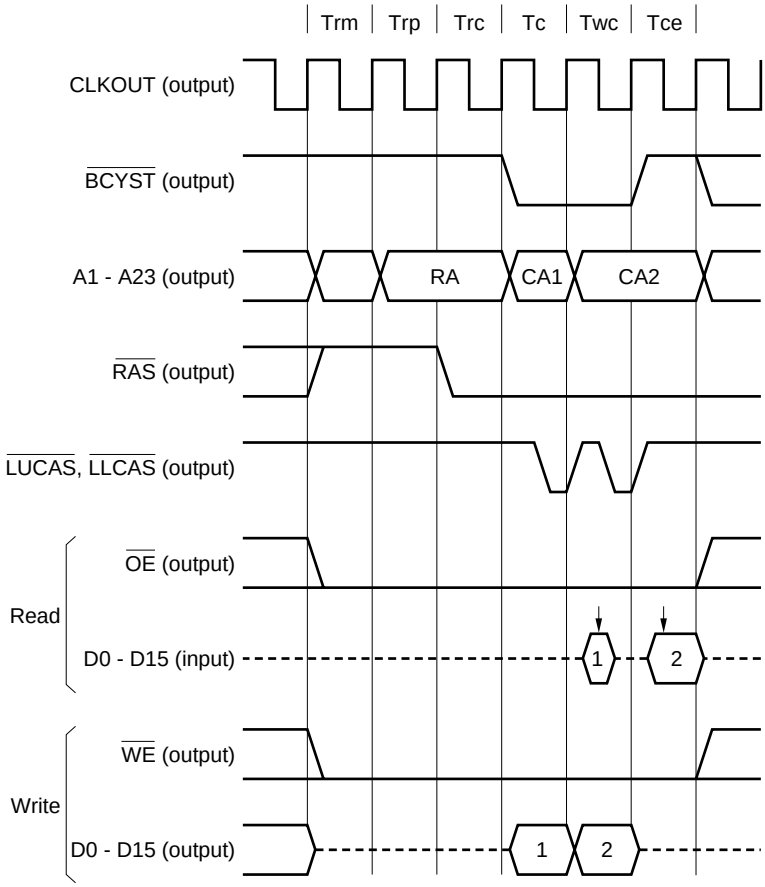
Signal Name	Signal Value
A1 - A23	Retains value in preceding cycle
D0 - D31	High impedance
$\overline{\text{RAS}}$	Retains value in preceding cycle
$\overline{\text{UUMWR}}$, $\overline{\text{ULMWR}}$, $\overline{\text{LUMWR}}$, $\overline{\text{LLMWR}}$, $\overline{\text{MRD}}$, $\overline{\text{CS1}}$ - $\overline{\text{CS7}}$, $\overline{\text{IORD}}$, $\overline{\text{IOWR}}$, $\overline{\text{BCYST}}$, $\overline{\text{UUCAS}}$, $\overline{\text{ULCAS}}$, $\overline{\text{LUCAS}}$, $\overline{\text{LLCAS}}$, $\overline{\text{WE}}$, $\overline{\text{OE}}$	Inactive

5.7 Bus Sizing

The V831 has a bus sizing function that changes the bus width between 32 bits and 16 bits. When 32-bit data is accessed via the 16-bit data bus in the single cycle, access is made two times. If 32-bit data is accessed via the 16-bit data bus in the burst cycle, access takes place eight times in a row.

The timing when the DRAM is accessed in word units in the single cycle with a data bus width of 16 bits (refer to **Figure 5-21**), the timing when the DRAM is accessed in the burst cycle (refer to **Figure 5-22**), the timing when the I/O is accessed in word units (refer to **Figure 5-23**), and the timing when the SRAM is accessed in word units in the single cycle (refer to **Figure 5-24**) are shown below.

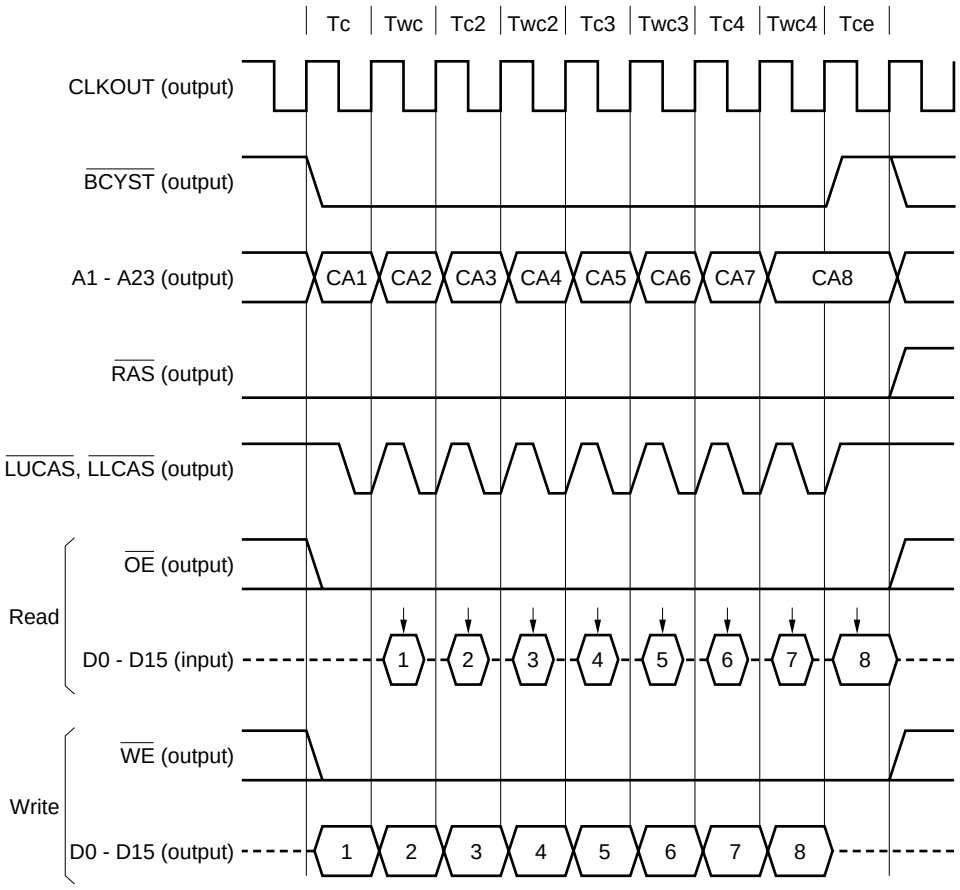
Figure 5-21. Additional Access in DRAM Single Cycle due to Bus Sizing



Remarks 1. The dotted lines indicate the high-impedance state.

2. The arrows indicate the sampling timing.

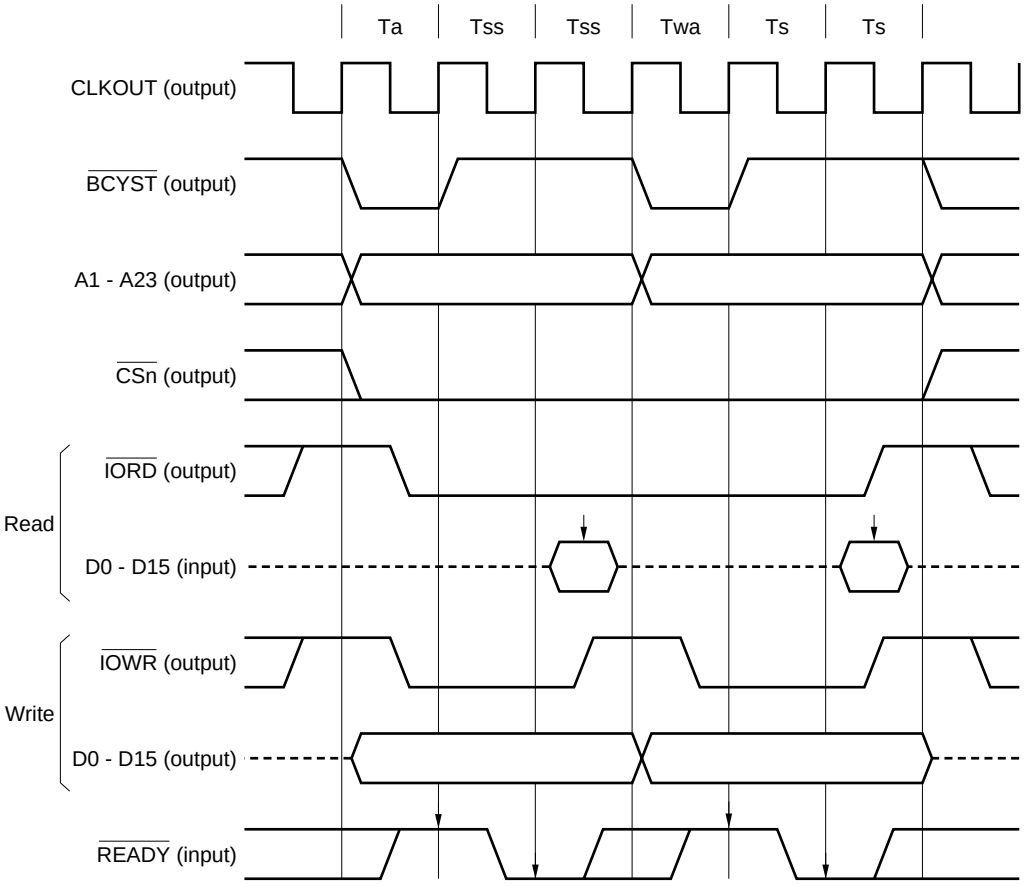
Figure 5-22. Additional Access in DRAM Burst Cycle due to Bus Sizing



Remarks 1. The dotted lines indicate the high-impedance state.

2. The arrows indicate the sampling timing.

Figure 5-23. Additional Access in I/O Cycle due to Bus Sizing

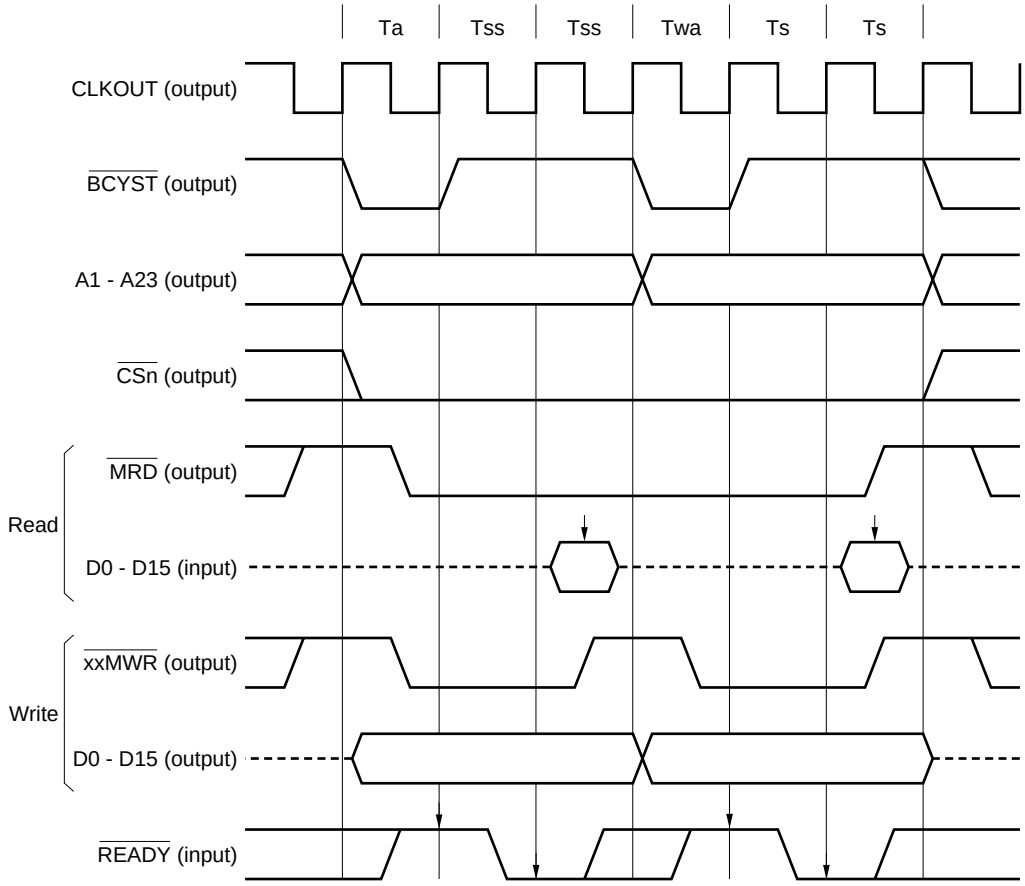


Remarks 1. n = 1 to 7

2. The dotted lines indicate the high-impedance state.

3. The arrows indicate the sampling timing.

Figure 5-24. Additional Access in SRAM Single Cycle due to Bus Sizing



Remarks **1.** n = 1 to 7

2. xxMWR: LLMWR, LUMWR, ULMWR, UUMWR

3. The dotted lines indicate the high-impedance state.

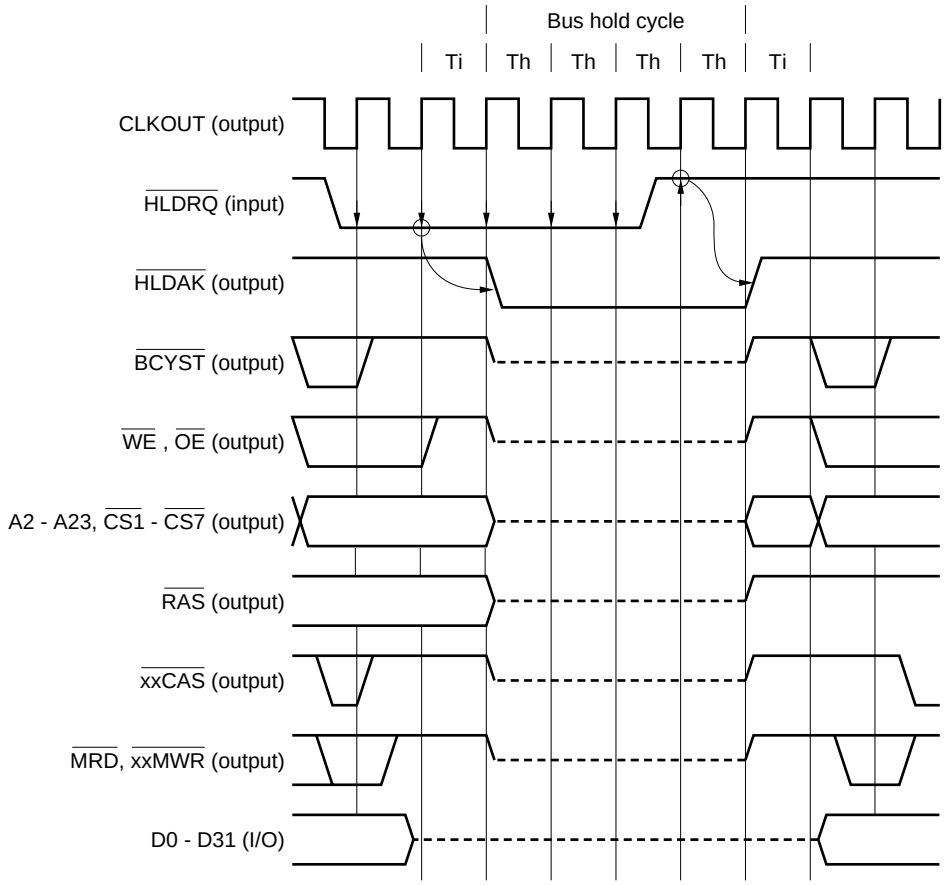
4. The arrows indicate the sampling timing.

5.8 Bus Hold Cycle

An external bus master can request the V831 for the bus mastership by asserting the $\overline{\text{HLDRQ}}$ signal active. The V831 arbitrates the bus, asserts the $\overline{\text{HLDAK}}$ signal active, and releases the bus. When the $\overline{\text{HLDRQ}}$ signal is asserted active, a bus hold request is accepted after one bus clock cycle, and the idle cycle (Ti) is started. After the idle cycle (Ti) has been executed for one bus clock cycle, the bus hold cycle (Th) is started, and the bus goes into a high-impedance state (refer to **Figure 5-25**).

After the bus hold cycle, the $\overline{\text{RAS}}$ signal is deasserted inactive. Bus hold is not released by the $\overline{\text{NMI}}$ signal.

Figure 5-25. Bus Hold Cycle



Remarks 1. $\overline{\text{xxCAS}}$: $\overline{\text{LLCAS}}$, $\overline{\text{LUCAS}}$, $\overline{\text{ULCAS}}$, $\overline{\text{UUCAS}}$

2. $\overline{\text{xxMWR}}$: $\overline{\text{LLMWR}}$, $\overline{\text{LUMWR}}$, $\overline{\text{ULMWR}}$, $\overline{\text{UUMWR}}$

3. The dotted lines indicate the high-impedance state.

4. The arrows indicate the sampling timing.

5.9 Bus Arbitration

The V831 performs bus arbitration between two internal bus masters (CPU and DMA), DRAM refresh, and an external bus master. The priority of this bus arbitration is as follows:

Bus lock > DRAM refresh = external bus master > DMA > CPU

(1) Bus lock > external bus master

The external bus master cannot acquire the bus mastership between the bus lock read cycle started by the CAXI instruction and bus lock write cycle.

DRAM refresh is executed even during bus lock.

(2) External bus master > DMA

The external bus master can acquire the bus mastership during DMA transfer. DMA cannot acquire the bus mastership while the external bus master is using the bus.

However, the external bus master cannot acquire the bus mastership between the read cycle of one DMA transfer and write cycle.

(3) DRAM refresh > DMA

If a DRAM refresh request is generated while DMA is executing demand transfer, the refresh request of DRAMC takes precedence, and refresh is executed.

(4) DRAM refresh, external bus master

If a DRAM refresh request is generated while the external bus master is using the bus, the refresh request is kept pending. The pending refresh request can be stored in the refresh request queue of the BCU up to seven times. When the external bus master has released the bus, refresh is executed by the number of times the refresh request has been stored in the refresh request queue.

5.10 Write Buffer Operation

The V831 has four stages of internal write buffers to speed up the write operation. Therefore, the CPU can execute the next instruction without having to wait for the completion of the bus cycle that has been started by a store instruction. If all the write buffers are used, the store instruction waits until one of the write buffers becomes empty. A store instruction executed to write to the internal RAM does not wait because it does not use a write buffer. To ensure the sequential relation at memory access, all the contents of the write buffers are output to the external memory before processing is executed in the following cases:

If a data cache miss occurs when a load instruction is executed.
If the non-cache area is accessed by a load instruction.

Similarly, all the contents of the write buffers are output to the external memory and then the instruction is executed in the following cases:

I/O access instruction
Block transfer instruction (BILD, BDLD, BIST, BDST)
HALT, STBY, CAXI instructions

The bus hold operation is performed regardless of the operations of the write buffers. If data written to the external memory is transferred by means of DMA, therefore, perform I/O write, etc., after execution of the last store instruction, output all the contents of the write buffers, and then execute the DMA transfer.

5.11 Memory Mapped I/O

With the V831, there may be a time lag between the execution of the store instruction and the corresponding bus write operation. Therefore, care must be exercised if memory mapped I/O is used in critical timing.

Note that the INT instruction extends zero while the LD instruction extends the sign.

CHAPTER 6 WAIT CONTROL FUNCTION

The bus control unit (BCU) controls eight blocks respectively corresponding to seven $\overline{CS}$ signals to select a type of bus cycle, generate the $\overline{CS}$ signals, select data bus width, control wait states, and insert idle states.

6.1 Features

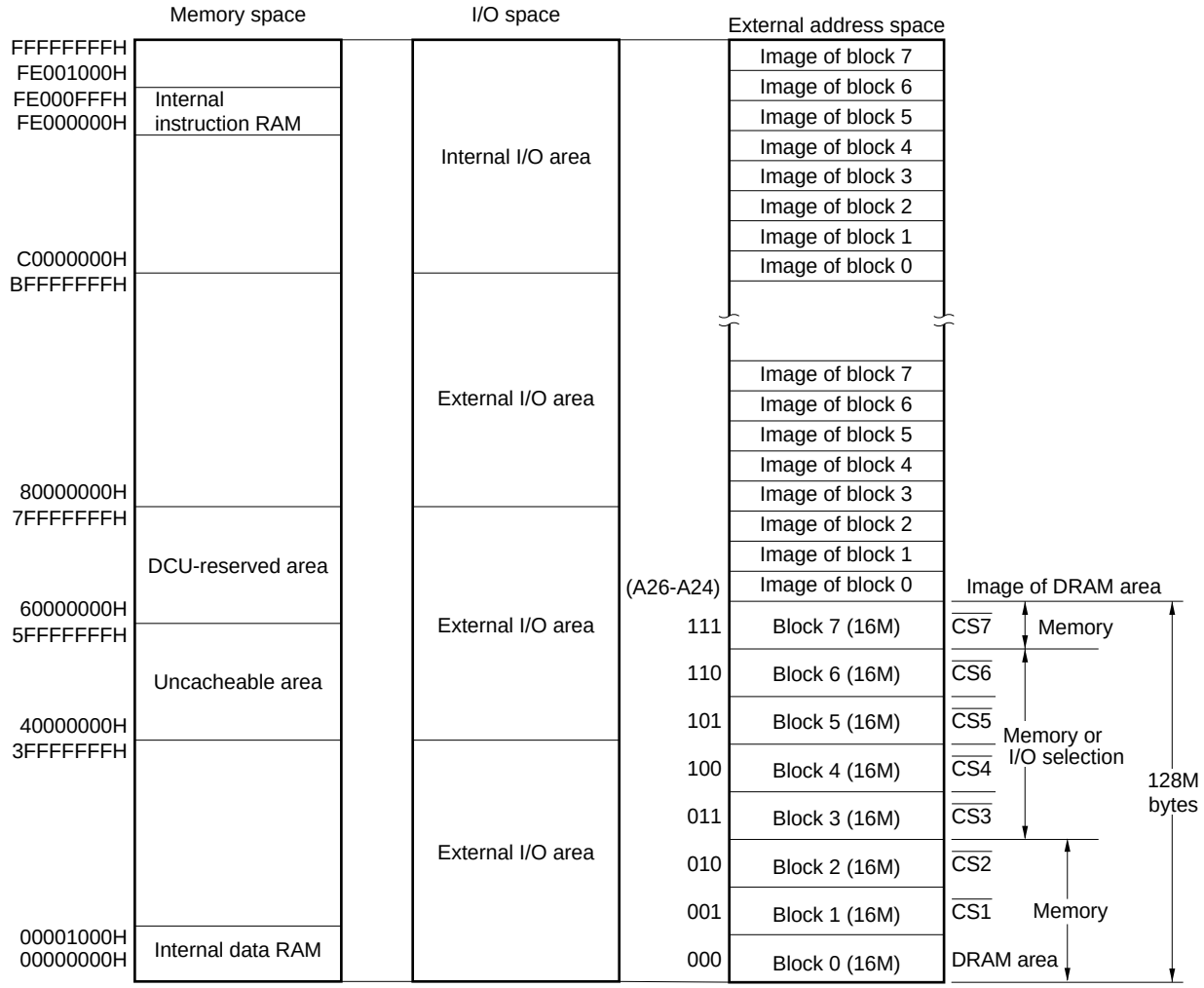
- Controls 8 blocks in accordance with I/O and memory spaces
- Linear address space of each block: 16M bytes
- Bus cycle select function
  - Block 0 : EDO DRAM
  - Block 1, 2 : SRAM (ROM)
  - Blocks 3 through 6 : I/O or SRAM (ROM) selectable
  - Block 7 : Page-ROM or SRAM (ROM) selectable
- Data bus width select function
  - Data bus width selectable between 32 bits and 16 bits for each block
- Wait control function
  - Block 0 : EDO DRAM access timing controlable
  - Blocks 1 through 4 and 7 : 0 to 7 wait states
  - Blocks 5 and 6 : 0 to 15 wait states
- Idle state insertion function
  0 to 3 states for each block (bus clock)

6.2 Address Space and Block

The internal 4G-byte memory and I/O spaces are divided into blocks with each block consisting of 16M bytes. The linear address space of each block is 16M bytes. Because addresses 60000000H through 7FFFFFFFH of the uncacheable area are used as the debug monitor space of the DCU, do not connect an external memory to this area.

★ The image of each block overlaps the DCU-reserved area but there is no problem in terms of operation. However, do not map external devices to the addresses (60000000 through 7FFFFFFF) of the DCU-reserved area.

Figure 6-1. Address Space



6.3 Wait Control Registers

BCU has a bus cycle type control register (BCTC), data bus width control register (DBC), programmable wait control registers (PWC0 and PWC1) that perform wait control.

6.3.1 Bus cycle type control register (BCTC)

This register sets the type of the bus cycle for blocks 3 through 7. However, the bus cycle of block 0 is fixed to the DRAM cycle, and the bus cycles of blocks 1 and 2 are fixed to SRAM/ROM cycles. This register can be read/written in 8-bit units.

Figure 6-2. Bus Cycle Type Control Register (BCTC)

Address: C0000010H

	7	6	5	4	3	2	1	0
BCTC	CT7	CT6	CT5	CT4	CT3	0	0	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R	R	R

Bit	Bit Name	Description
7	CT7	Cycle Type7 When memory block 7 is accessed, the $\overline{CS7}$ signal is output. This bit specifies the cycle to be started at that time. 0: SRAM (ROM) cycle 1: Page-ROM cycle
6 - 3	CT6 - CT3	Cycle Type6 - 3 These bits specify whether the $\overline{CS6}$ through $\overline{CS3}$ signals are output to the memory or I/O, or specify the cycle to be started. 0: SRAM (ROM) cycle 1: I/O cycle

6.3.2 Data bus width control register (DBC)

This register specifies the data bus width for blocks 0 through 6. It can be read/written in 8-bit units. However, the bus width of block 7 ($\overline{CS7}$), where the normal boot ROM is placed, is specified by the value of the BT16B pin.

Figure 6-3. Data Bus Width Control Register (DBC)

Address: C0000012H

	7	6	5	4	3	2	1	0
DBC	BW7	BW6	BW5	BW4	BW3	BW2	BW1	BW0
Initial value:	BT16B	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Description
7	BW7	Bus Width This bit reads the value of the BT16B pin. The data bus width of block 7 ($\overline{CS7}$) specified by the value of the BT16B pin is as follows: 0: 32-bit bus width 1: 16-bit bus width
6 - 0	BW6 – BW0	Bus Width These bits specify the data bus width of blocks 6 through 0: 0: 32-bit bus width 1: 16-bit bus width

6.3.3 Programmable wait control register 0 (PWC0)

This register sets the number of wait states used when blocks 1 through 3 are accessed. It can be read/written in 16-bit units. Up to seven wait states can be inserted.

Block 0 is fixed to the DRAM area. The number of wait states when accessing the DRAM is specified by the DRAM configuration register (DRC).

Figure 6-4. Programmable Wait Control Register 0 (PWC0)

Address: C0000014H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWC0	0	WS3			0	WS2			0	WS1			0	0	0	0
Initial value:	0	1	1	1	0	1	1	1	0	1	1	1	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Description
14 - 12	WS3	Wait States3 This bit specifies the number of wait states when block 3 ($\overline{CS3}$) is accessed. The number of wait states is 0 to 7.
10 - 8	WS2	Wait States2 This bit specifies the number of wait states when block 2 ($\overline{CS2}$) is accessed. The number of wait states is 0 to 7. The method of setting the number of wait states is the same as that of WS3.
6 - 4	WS1	Wait States1 This bit specifies the number of wait states when block 1 ($\overline{CS1}$) is accessed. The number of wait states is 0 to 7. The method of setting the number of wait states is the same as that of WS3.

WS3			Number of wait states	WS3			Number of wait states
0	0	0	0	1	0	0	4
0	0	1	1	1	0	1	5
0	1	0	2	1	1	0	6
0	1	1	3	1	1	1	7

6.3.4 Programmable wait control register 1 (PWC1)

This register specifies the number of wait states when blocks 4 through 7 are accessed. It can be read/written in 16-bit units. Up to seven wait states can be inserted when accessing blocks 4 and 7. Up to 15 wait states can be inserted when accessing blocks 5 and 6.

Figure 6-5. Programmable Wait Control Register 1 (PWC1)

Address: C0000016H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWC1	0	WS7			WS6				WS5				0	WS4		
Initial value:	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Description
14 - 12	WS7	Wait States7 This bit specifies the number of wait states when block 7 ($\overline{CS7}$) is accessed. The number of wait states is 0 to 7. If the Page-ROM cycle is selected by the CT7 bit of the BCTC, the number of wait states is that in the single cycle or that in the burst cycle during off-page access. The number of wait states during on-page access can be specified by the PWS bit of PRC register.
11 - 8	WS6	Wait States6 This bit specifies the number of wait states when block 6 ($\overline{CS6}$) is accessed. The number of wait states is 0 to 15.
7 - 4	WS5	Wait States5 This bit specifies the number of wait states when block 5 ($\overline{CS5}$) is accessed. The number of wait states is 0 to 15. The method of setting the number of wait states is the same as that of WS6
2 - 0	WS4	Wait States4 This bit specifies the number of wait states when block 4 ($\overline{CS4}$) is accessed. The number of wait states is 0 to 7. The method of setting the number of wait states is the same as that of WS7.

WS7			Number of wait states	WS7			Number of wait states
0	0	0	0	1	0	0	4
0	0	1	1	1	0	1	5
0	1	0	2	1	1	0	6
0	1	1	3	1	1	1	7

WS6				Number of wait states	WS6				Number of wait states
0	0	0	0	0	1	0	0	0	8
0	0	0	1	1	1	0	0	1	9
0	0	1	0	2	1	0	1	0	10
0	0	1	1	3	1	0	1	1	11
0	1	0	0	4	1	1	0	0	12
0	1	0	1	5	1	1	0	1	13
0	1	1	0	6	1	1	1	0	14
0	1	1	1	7	1	1	1	1	15

6.3.5 Programmable idle control register (PIC)

This register specifies the number of idle states inserted after blocks 0 through 7 have been accessed for read. It can be read/written in 16-bit units. Up to three idle states can be inserted.

When blocks 0 through 7 are accessed by the V830 CPU core, one idle state is always inserted after the read cycle. This idle state is counted as the set value of ISn of the PIC register. When ISn (n = 0 to 7) is 0 or 1, one idle state is inserted.

However, the setting of the PIC register is valid even during 2-cycle transfer of DMA.

Figure 6-6. Programmable Idle Control Register (PIC)

Address: C0000018H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIC	IS7		IS6		IS5		IS4		IS3		IS2		IS1		IS0	
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Description
15 - 0	IS7 - IS0	Idle State7 - 0 These bits specify the number of idle states. IS7 through IS0 correspond to blocks 7 through 0, respectively. When the block corresponding to the $\overline{CSn}$ signal is accessed for read, the number of idle states specified by ISn (n = 0 to 7) is automatically inserted after read access.

ISn		Number of inserted idle states
0	0	0
0	1	1
1	0	2
1	1	3

★ 6.4 Wait Control by $\overline{\text{READY}}$ Pin

The V831 can control wait states to be inserted in cycles accessing the I/O, SRAM, and ROM area, by using the PWC0 and PWC1 registers, and $\overline{\text{READY}}$ pin.

6.4.1 Sampling timing of $\overline{\text{READY}}$ pin

- I/O area: The $\overline{\text{READY}}$ pin is sampled at the rising edge of the Ts state. If the programmed number of wait states has not been completed, or if the $\overline{\text{READY}}$ pin is not active (if a wait state is inserted), the Ts cycle is repeated.
- Single cycle of SRAM or ROM: The $\overline{\text{READY}}$ pin is sampled at the rising edge of the Ts state. If the programmed number of times of wait has not been completed, or if the $\overline{\text{READY}}$ pin is not active, the Ts cycle is repeated.
- Burst cycle of SRAM or ROM: The $\overline{\text{READY}}$ pin is sampled at the rising edge of each Tb state and the rising edge of the Ts state. When a wait state is inserted, the Tb cycle or Ts cycle is repeated.
- Page-ROM cycle: The $\overline{\text{READY}}$ pin is sampled at the rising edge of the Tb state and the rising edge of the Ts state. When a wait state is inserted, the Tb cycle or Ts cycle is repeated. A different number of wait states can be programmed in the on-page mode and off-page mode.

6.4.2 When using both $\overline{\text{READY}}$ pin and programmable wait

The number of wait states set by the PWC0 and PWC1 registers and the number of wait states set by using the $\overline{\text{READY}}$ pin is logically ORed. Therefore, the larger number of wait states is inserted.

When performing wait control by using the $\overline{\text{READY}}$ pin, the wait cycle is not cleared even if the $\overline{\text{READY}}$ pin is deasserted inactive at the above sampling timing, until the programmed number of wait states has been inserted. The wait cycle is not cleared unless the $\overline{\text{READY}}$ pin is inactive in the Ts state after the programmed wait cycle, even if the $\overline{\text{READY}}$ pin is deasserted inactive before the programmed wait cycle is cleared. When performing wait control by using the $\overline{\text{READY}}$ pin, therefore, it is recommended to set the number of waits states of the PWC0 and PWC1 registers for the corresponding block to 0 to avoid confusion.

- DRAM cycle: The wait state cannot be controlled by the $\overline{\text{RESET}}$ pin input. The precharge period, RAS - CAS delay time, and CAS cycle period are programmable. For details, refer to **5.5 DRAM Cycle** and **CHAPTER 7 MEMORY ACCESS CONTROL FUNCTION**.

CHAPTER 7 MEMORY ACCESS CONTROL FUNCTION

This chapter explains the DRAM control function and Page-ROM control function.

The BCU of the V831 can be directly connected to EDO DRAM, Page-ROM, and SRAM (ROM).

The EDO DRAM is accessed by using the address bus, data bus, and four ××CAS signals, RAS, OE, and WE signals. An address of the DRAM is output from the address pins with the row address and column address multiplexed.

Page access to the Page-ROM is enabled in the burst access mode. The page size can be selected from 8 or 16 bytes.

The SRAM (ROM) is accessed by using the address bus and data bus, four ××MWR signals, MRD, and CS signals.

Remarks 1. ××CAS: UUCAS, ULCAS, LUCAS, LLCAS

2. ××MWR: UUMWR, ULMWR, LUMWR, LLMWR

7.1 Features

- DRAM control function
  - Generation of RAS, LLCAS, LUCAS, ULCAS, UUCAS, REFRQ, OE, WE signals
  - 8, 9, and 10 multiplexed address bits
  - DRAM access timing control
    CAS access period : 1 or 2 bus clocks selectable
    RAS-CAS delay period : 1.5 or 2.5 bus clocks selectable
    RAS precharge period : 2 or 3 bus clocks selectable
  - CBR refresh and CBR self-refresh functions
- Page-ROM control function
  - Page size : 8 or 16 bytes
  - Wait control during page access : 0 or 1 wait state

7.2 DRAM Control Function

The BCU generates RAS, LLCAS, LUCAS, ULCAS, UUCAS, REFRQ, OE, and WE signals and controls access to the DRAM. Addresses are output to the DRAM from the address pins by multiplexing row and column addresses.

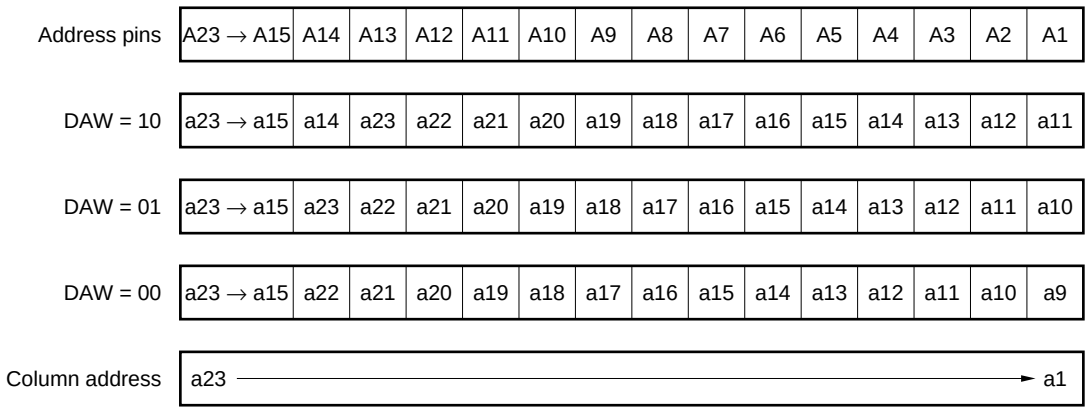
The connected DRAM must be of x8 bits or more and have a hyper page mode (EDO).

The refresh mode is a CAS-before-RAS (CBR) mode, and the refresh cycle can be arbitrarily set. CBR self refresh is performed in the STOP mode.

7.2.1 Address multiplex function

Row addresses and column addresses are multiplexed, as shown in Figure 7-1, in the DRAM cycle, depending on the values of the DAW bit of the DRAM configuration register (DRC). a1 through a23 in this figure indicate the addresses output by the CPU, and A1 through A23 indicate the address pins of the V831.

Figure 7-1. Output of Row Address and Column Address



7.2.2 Judgment of on-page/off-page

If the $\overline{RAS}$ signal is active when page access is enabled because the HPAE bit of the DRAM configuration register (DRC) is 1, whether the DRAM access to be started is in the same page as the previous DRAM access. Table 7-1 shows the relation between an address to be compared and address shift.

Table 7-1. Address Compared by on-page/off-page Judgment

Address Shift	Data Bus Width	
	16 bits	32 bits
8	a23 – a9	a23 – a10
9	a23 – a10	a23 – a11
10	a23 – a11	a23 – a12

7.2.3 DRAM configuration register (DRC)

This register sets an address multiplex width during DRAM access, and the output timing of the $\overline{RAS}$ and $\overline{CAS}$ signals. It can be read/written in 8-bit units.

Figure 7-2. DRAM Configuration Register (DRC)

Address: C0000020H

	7	6	5	4	3	2	1	0
DRC	HPAE	RP	RCD	CRWT		0	DAW	
Initial value:	0	1	1	1	1	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Description
7	HPAE	Hyper Page Mode Enable This bit controls start of page access supporting the hyper page mode of the DRAM. However, page access cannot be disabled when the DRAM is accessed the second time and onward in the burst cycle. 0: Disables start (off-page access is always performed). 1: Enables start (start of on-page access is enabled).
6	RP	$\overline{RAS}$ Precharge Sets the precharge period of $\overline{RAS}$. 0: 2 bus clocks 1: 3 bus clocks
5	RCD	$\overline{RAS}$-$\overline{CAS}$ Delay Sets $\overline{RAS}$-$\overline{CAS}$ delay time. 0: 1.5 bus clocks 1: 2.5 bus clocks
4, 3	CRWT	$\overline{CAS}$ Read Write Timing Sets the $\overline{CAS}$ cycle period during DRAM read/write.
1, 0	DAW	DRAM Address Width Sets an address width for a column address in the DRAM cycle.

CRWT		$\overline{CAS}$ cycle time	
		Read cycle	Write cycle
0	0	1 bus clock	
0	1	Setting prohibited	
1	0	2 bus clocks	1 bus clock
1	1	2 bus clocks	

DAW		Address width
0	0	8 bits
0	1	9 bits
1	0	10 bits
1	1	Setting prohibited

7.2.4 Refresh function

The BCU can automatically generate the distributed CBR refresh cycle necessary for refreshing the external DRAM. Whether refreshing is enabled or disabled and the refresh interval are set by the refresh control register (RFC).

The BCU has a refresh request queue that can store refresh requests up to seven times.

(1) Refresh request queue

The BCU has a refresh request queue that can store refresh requests up to seven times. When the bus is released, the refresh cycle is successively generated until the contents of the refresh request queue reach “0”.

If a refresh request is generated when the contents of the refresh request queue are “7”, the contents of the queue are not changed and remain “7”.

(2) Refresh control register (RFC)

The refresh control register (RFC) enables or disables refreshing, and sets the length of the refresh cycle and refresh interval. The RFC register can be read/written in 16-bit units. The refresh interval can be calculated by the following expression:

Refresh interval (μs) = Refresh count clock (t_{RCY}) $\times$ Interval factor (RI)

The refresh count clock and interval factor are specified by the RCC bit and RI bit of the RFC register.

Because the BCU always asserts the $\overline{RAS}$ signal active after the DRAM has been accessed, if the refresh cycle is longer than the maximum value of the $\overline{RAS}$ pulse width (t_{RAS}), keep the refresh cycle shorter than the maximum value of t_{RAS}.

Caution To change the setting of the RI bit of the RFC register, be sure to disable refreshing by using the REN bit, and then change the content of the RI bit.

Figure 7-3. Refresh Control Register (RFC)

Address: C0000022H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RFC	REN	0	0	RFTC	RFW		RCC		0	RI						
Initial value:	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Description
15	REN	Refresh Enable Enables or disables CBR refresh and CBR self refresh. 0: Disables CBR refresh and CBR self refresh 1: Enables CBR refresh and CBR self refresh
12	RFTC	Refresh Request Terminal Count Selects output of $\overline{\text{REFRQ}}$/$\overline{\text{TC}}$. 0: $\overline{\text{TC}}$ output 1: $\overline{\text{REFRQ}}$ output
11, 10	RFW	Refresh Wait Sets the $\overline{\text{RAS}}$ active period of the CBR refresh cycle.

RFW		$\overline{\text{RAS}}$ active period
0	0	3 bus clocks
0	1	4 bus clocks
1	0	5 bus clocks
1	1	Setting prohibited

9, 8	RCC	Refresh Count Clock Specifies a refresh count clock (t_{RCY}). ϕ indicates the frequency of the internal bus clock.

RCC		Refresh count clock (t_{RCY})
0	0	$32/\phi$
0	1	$128/\phi$
1	-	Setting prohibited

6 - 0	RI	Refresh Interval Sets the interval factor of the interval timer for refresh timing generation.

RI						Interval factor
0	0	0	0	0	0	1
0	0	0	0	0	1	2
:	:	:	:	:	:	:
1	1	1	1	1	1	128

7.3 Page-ROM Control Function

The BCU controls page access to the Page-ROM. The page size and the number of wait states during page access can be set by using the Page-ROM configuration register (PRC).

7.3.1 Page-ROM configuration register (PRC)

This register sets the number of wait states during on-page access of the Page-ROM and the page size. This register is valid only when the Page-ROM cycle is started for block 7.

Figure 7-4. Page-ROM Configuration Register (PRC)

Address: C0000024H

	7	6	5	4	3	2	1	0
PRC	0	0	0	0	0	0	PWS	PS
Initial value:	0	0	0	0	0	0	1	1
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Description
1	PWS	Page-ROM Wait States Sets the number of wait states during on-page access of the Page-ROM. 0 : 0 wait 1 : 1 wait This bit is valid only when the Page-ROM cycle is selected by the CT7 bit of the BCTC register. The number of wait states during off-page access is in accordance with the setting of the WS7 bit of the PWC1 register.
0	PS	Page Size Sets the page size of the Page-ROM.

PS	Page size
0	16 bytes This setting is valid in the 16-/32-bit bus mode. In both the modes, the first access in the burst cycle is off-page access.
1	8 bytes This setting is valid only in the 16-bit bus mode. In the burst cycle in which access is made eight times in a row, the first and fifth accesses are off-page accesses. This setting is ignored in the 32-bit bus mode. In the burst cycle in which access is made four times in a row, only the first access is off-page access.

CHAPTER 8 DMA FUNCTION

The V831 has a DMA (Direct Memory Access) controller that executes and control DMA transfer.

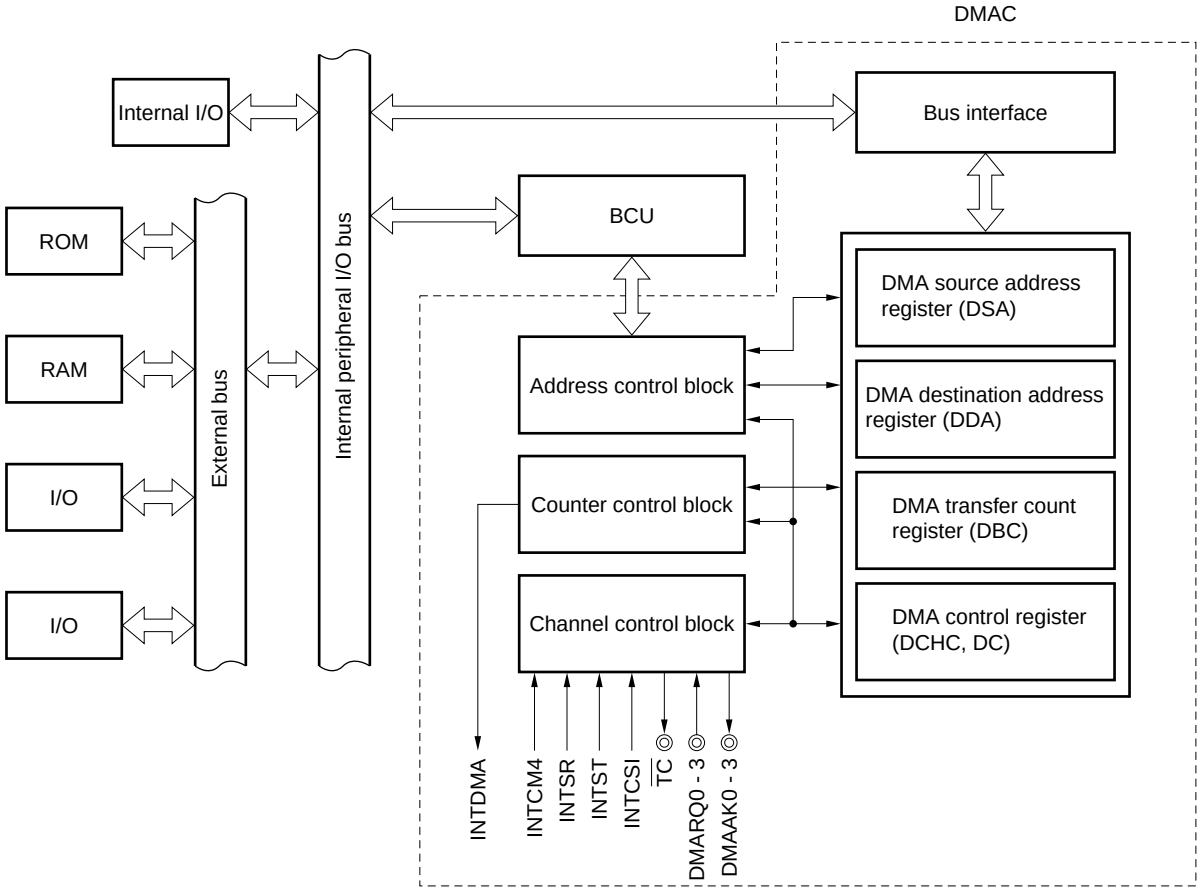
The DMAC (DMA controller) transfers data between memory and I/O or between memory areas according to a DMA request issued by the internal peripheral hardware (serial interface and timer) or external DMARQ pin, or by means of software trigger.

8.1 Features

- Four independent DMA channels
- Transfer unit: Bytes, half words (2 bytes), words (4 bytes)
- Maximum number of transfers: 16,777,216 (2^{24}) times
- Transfer type: 2-cycle transfer
- Two transfer modes
  - Single transfer mode
  - Demand transfer mode
- Transfer request
  - External DMARQ pin ($\times$ 4)
  - Request from internal peripheral hardware (serial interface ($\times$ 3 channels) and timer)
  - Request from software
- Transfer source and destination
  - Between memory and I/O
  - Between memory and memory
- Programmable wait function
- DMA transfer end output signal ($\overline{TC}$)

8.2 Configuration

Figure 8-1. DMAC Block Diagram



8.3 DMA Control Registers

8.3.1 DMA source address registers 0 through 3 (DSA0 through DSA3)

These registers specify the DMA transfer source addresses (32 bits) of DMA channels 0 through 3. Each of these registers consist of two 16-bit registers: DSAnH and DSAnL (n = 0 to 3). Set an address value in accordance with the DMA transfer data size (set by the DCHC register) (half-word transfer: multiple of 2, word transfer: multiple of 4).

The next DMA transfer source address is held during DMA transfer.

(1) DMA source address registers 0H through 3H (DSA0H through DSA3H)

Figure 8-2. DMA Source Address Registers 0H through 3H (DSA0H through DSA3H)

Address of DSA0H: C0000030H
Address of DSA1H: C0000040H
Address of DSA2H: C0000050H
Address of DSA3H: C0000060H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSA0H - 3H	SA31	SA30	SA29	SA28	SA27	SA26	SA25	SA24	SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16
Initial value:	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Description
15 - 8	SA31 - SA24	Source Address These bits specify the address (A31 through A24) of the DMA transfer source. They hold the next DMA transfer source address during DMA transfer. **Caution Although a 32-bit address is specified, only 24 bits of the address, A23 through A1, are output to an external device. Therefore, the width of the counter is 24 bits and bits SA31 through SA24 are fixed. This means that data cannot be transferred over blocks corresponding to the chip select signals.**
7 - 0	SA23 - SA16	Source Address These bits specify the address (A23 through A16) of the DMA transfer source. They hold the next DMA transfer source address during DMA transfer.

(2) DMA source address registers 0L through 3L (DSA0L through DSA3L)

Figure 8-3. DMA Source Address Registers 0L through 3L (DSA0L through DSA3L)

Address of DSA0L: C0000032H
Address of DSA1L: C0000042H
Address of DSA2L: C0000052H
Address of DSA3L: C0000062H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSA0L - 3L	SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0
Initial value:	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Description
15 - 0	SA15 - SA0	Source Address These bits specify the address (A15 through A0) at the DMA transfer source. They hold the next DMA transfer source address during DMA transfer.

8.3.2 DMA destination address registers 0 through 3 (DDA0 through DDA3)

These registers specify the DMA transfer destination addresses (32 bits) of DMA channels 0 through 3. Each of these registers consist of two 16-bit registers: DDAnH and DDAnL (n = 0 to 3). Set an address value in accordance with the DMA transfer data size (set by the DCHC register) (half-word transfer: multiple of 2, word transfer: multiple of 4). They hold the next DMA transfer destination address during DMA transfer.

(1) DMA destination address registers 0H through 3H (DDA0H through DDA3H)

Figure 8-4. DMA Destination Address Registers 0H through 3H (DDA0H through DDA3H)

Address of DDA0H: C0000034H
Address of DDA1H: C0000044H
Address of DDA2H: C0000054H
Address of DDA3H: C0000064H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DDA0H - 3H	DA31	DA30	DA29	DA28	DA27	DA26	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16
Initial value:	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Description
15 - 8	DA31 - DA24	Destination Address These bits specify the address (A31 through A24) of the DMA transfer destination. They hold the next DMA transfer destination address during DMA transfer. **Caution Although a 32-bit address is specified, only 24 bits of the address, A23 through A1, are output to an external device. Therefore, the width of the counter is 24 bits and bits DA31 through DA24 (8 bits) are fixed. This means that data cannot be transferred over blocks corresponding to the chip select signals.**
7 - 0	DA23 - DA16	Destination Address These bits specify the address (A23 through A16) of the DMA transfer destination. They hold the next DMA transfer destination address during DMA transfer.

(2) DMA destination address registers 0L through 3L (DDA0L through DDA3L)

Figure 8-5. DMA Destination Address Registers 0L through 3L (DDA0L through DDA3L)

Address of DDA0L: C0000036H
Address of DDA1L: C0000046H
Address of DDA2L: C0000056H
Address of DDA3L: C0000066H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DDA0L - 3L	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
Initial value:	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Description
15 - 0	DA15 - DA0	Destination Address These bits specify the address (A15 through A0) at the DMA transfer destination. They hold the next DMA transfer destination address during DMA transfer.

8.3.3 DMA byte count registers 0 through 3 (DBC0 through DBC3)

These registers specify the number of times of byte transfer (24 bits) by DMA channels 0 through 3. Each of these registers consists of two 16-bit registers: DBCnH and DBCnL (n = 0 to 3).

These registers hold the remaining number of times byte transfer is to be executed during DMA transfer.

The values by which the values of these registers are decremented during byte, half word, and word transfer are shown below. Transfer ends when a borrow occurs.

Byte transfer : Decremented by one [Setting] Number of transfers – 1
Half word transfer: Decremented by two [Setting] (Number of transfers – 1) × 2
Word transfer : Decremented by four [Setting] (Number of transfers – 1) × 4

Figure 8-6. DMA Byte Count Registers 0 through 3 (DBC0 through DBC3)

(1) DMA byte count registers 0H through 3H (DBC0H through DBC3H)

Address of DBC0H: C0000038H
Address of DBC1H: C0000048H
Address of DBC2H: C0000058H
Address of DBC3H: C0000068H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBC0H - 3H	0	0	0	0	0	0	0	0	BC							
Initial value:	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

(2) DMA byte count registers 0L through 3L (DBC0L through DBC3L)

Address of DBC0L: C000003AH
Address of DBC1L: C000004AH
Address of DBC2L: C000005AH
Address of DBC3L: C000006AH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBC0L - 3L	BC															
Initial value:	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Description
7 - 0: DBCnH 15 - 0: DBCnL	BC	Byte Count These bits specify the number of times of byte transfer. They hold the remaining number of times byte transfer is to be executed during DMA transfer. Bit 7 of the DBCnH register is the MSB, and bit 0 of the DBCnL register is the LSB (n = 0 to 3).

BC	Number of idle states inserted
000000H	First byte transfers or remaining number of byte transfers
000001H	Second byte transfers or remaining number of byte transfers
:	:
FFFFFFH	Byte transfer of 2^{24} times or remaining number of byte transfers

8.3.4 DMA channel control registers 0 through 3 (DCHC0 through DCHC3)

These 16-bit registers control the DMA transfer operation modes of DMA channels 0 through 3.

Figure 8-7. DMA Channel Control Registers 0 through 3 (DCHC0 through DCHC3) (1/3)

Address of DCHC0: C000003CH
Address of DCHC1: C000004CH
Address of DCHC2: C000005CH
Address of DCHC3: C000006CH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCHC0 - 3	0	TTYP			TBT		SAD		DAD		DAL	DRL	TM	DS		EN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Description
14 - 12	TTYP	Transfer Type These bits specify the cause that starts DMA transfer.
11, 10	TBT	Transfer Block Type These bits specify whether the address blocks at the transfer source and destination are in memory or I/O.
9, 8	SAD	Source Address Count Direction These bits specify the count direction of the transfer source address of DMA channel n (n = 0 to 3).

TTYP			Start cause
0	0	0	Started by DMARQn signal: 2-cycle transfer
0	0	1	Started by software: 2-cycle transfer
0	1	0	Setting prohibited
0	1	1	Setting prohibited
1	0	0	Started by INTST signal: 2-cycle transfer
1	0	1	Started by INTSR signal: 2-cycle transfer
1	1	0	Started by INTCSI signal: 2-cycle transfer
1	1	1	Started by INTCM4 signal: 2-cycle transfer

TBT		Transfer block
0	0	Memory to memory
0	1	Memory to I/O
1	0	I/O to memory
1	1	Setting prohibited

SAD		Count direction
0	0	Increment
0	1	Decrement
1	0	Fixed
1	1	Setting prohibited

Figure 8-7. DMA Channel Control Registers 0 through 3 (DCHC0 through DCHC3) (2/3)

Address of DCHC0: C000003CH
Address of DCHC1: C000004CH
Address of DCHC2: C000005CH
Address of DCHC3: C000006CH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCHC0 - 3	0	TTYP			TBT		SAD		DAD		DAL	DRL	TM	DS		EN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Description
7, 6	DAD	Destination Address Count Direction These bits specify the count direction of the transfer destination address of DMA channel n (n = 0 to 3). DAD (0 0) Increment DAD (0 1) Decrement DAD (1 0) Fixed DAD (1 1) Setting prohibited
5	DAL	DMAAK Level This bit specifies the active level of the DMAAKn signal (n = 0 to 3). 0: Active low 1: Active high **Caution The DMAAKn signal goes high from when the device has been reset until this register is set.**
4	DRL	DMARQ Level This bit specifies the level at which the DMARQn signal is detected (n = 0 to 3). 0: Detects low level 1: Detects high level
3	TM	Transfer Mode This bits specifies a transfer mode during DMA transfer (n = 0 to 3). 0: Single transfer mode 1: Demand transfer mode **Caution Select the signal transfer mode when DMA transfer is started by a request from the internal peripheral hardware.**
2, 1	DS	Data Size These bits specify the transfer size for DMA transfer. If the transfer destination is the I/O space, set the transfer data size greater than the data bus width (transfer destination) set by the DBC register (except when the address count is fixed (DAD bit = 10)). DS (0 0) Byte unit DS (0 1) Half word (2 bytes) unit DS (1 0) Word (4 bytes) unit DS (1 1) Setting prohibited

Figure 8-7. DMA Channel Control Registers 0 through 3 (DCHC0 through DCHC3) (3/3)

Address of DCHC0: C000003CH
Address of DCHC1: C000004CH
Address of DCHC2: C000005CH
Address of DCHC3: C000006CH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCHC0 - 3	0	TTYP			TBT		SAD		DAD		DAL	DRL	TM	DS		EN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Description
0	EN	Enable This bit enables or disables DMA transfer of DMA channel n (n = 0 to 3). It is reset when DMA transfer is completed by terminal count. To start DMA transfer, set the EN bit to 1, TCn bit of the DC register to 0, and MEN bit to 1 (n = 0 to 3). 0: Disables DMA transfer (reset) 1: Enables DMA transfer

8.3.5 DMA control register (DC)

This 16-bit register controls the DMA transfer operation mode.

Figure 8-8. DMA Control Register (DC)

Address: C000006EH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DC	0	0	0	0	0	0	0	0	TC3	TC2	TC1	TC0	0	0	0	MEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Description
7 - 4	TC3 - TC0	Terminal Count Channel3 - 0 These bits are status bits that indicate whether DMA transfer by DMA channel n has been completed (n = 0 to 3). They can be only read. These bits are set when DMA transfer has been completed by terminal count, and are reset when they are read. 0: DMA transfer not completed (reset) 1: DMA transfer completed
0	MEN	Master Enable This bit enables or disables DMA transfer by all the DMA channels (n = 0 to 3). It is also reset by the $\overline{\text{NMI}}$ signal. To start DMA transfer, set the MEN bit to 1, TCn bit to 0, and EN bit of the DCHCn register to 1 (n = 0 to 3). To enable or disable transfer by each channel, use the EN bit of the DCHC0 through DCHC3 registers. **Example** To set the MEN bit to 1 only when DMA transfer is enabled, and to set the MEN bit to 1 during NMI processing only when the MEN bit is cleared by the $\overline{\text{NMI}}$ signal 0: Disables DMA transfer (reset) 1: Enables DMA transfer

8.4 Transfer Mode

8.4.1 Single transfer mode

In the single transfer mode, the DMAC releases the bus each time it has executed a transfer. If a DMA transfer request is issued after that, the DMAC executes a transfer once again. This is repeated until the terminal count is generated.

If another DMA transfer request with a higher priority is issued while the DMAC has released the bus, the DMA request with the higher priority always takes precedence.

Figures 8-9 and 8-10 show examples of single transfer. Figure 8-10 is an example where a DMA request with the higher priority is issued. In this example, DMA channel 0 is in the demand transfer mode and channel 1 is in the single transfer mode.

Figure 8-9. Example of Single Transfer 1

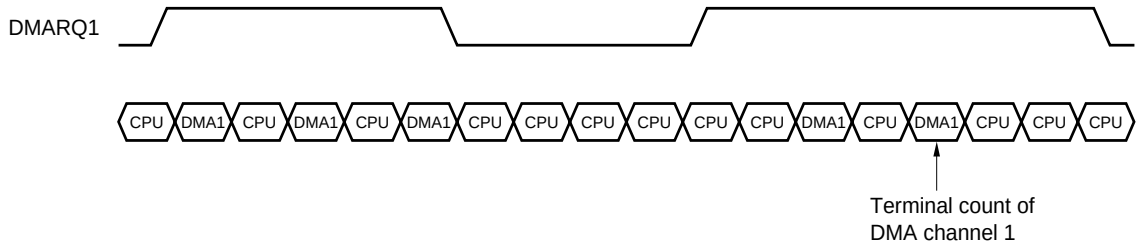
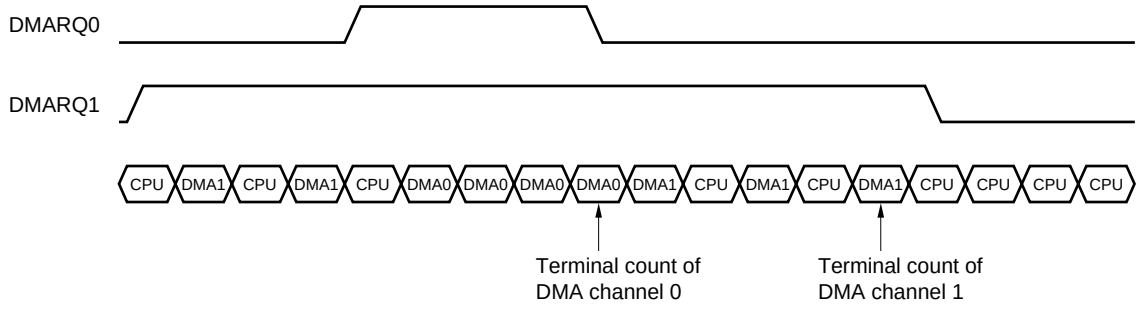


Figure 8-10. Example of Single Transfer 2



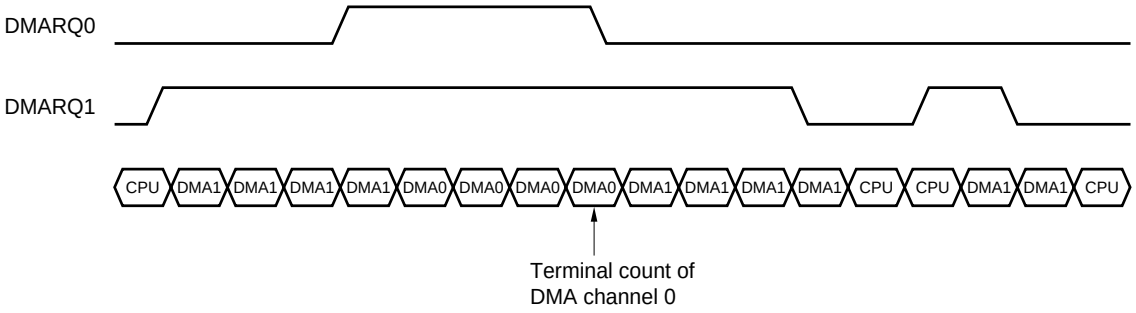
8.4.2 Demand transfer mode

In the demand transfer mode, the DMAC does not release the bus while DMA transfer requests are being issued. As long as DMA transfer requests are being issued, transfer continues until the terminal count is generated.

If DMA transfer requests are stopped and then a request is issued again, transfer can be resumed.

Figure 8-11 shows an example of demand transfer. This is an example where a DMA request with the higher priority is issued, and DMA channels 0 and 1 are in the demand transfer mode.

Figure 8-11. Example of Demand Transfer



8.5 DMA Transfer Type and Subject to Transfer

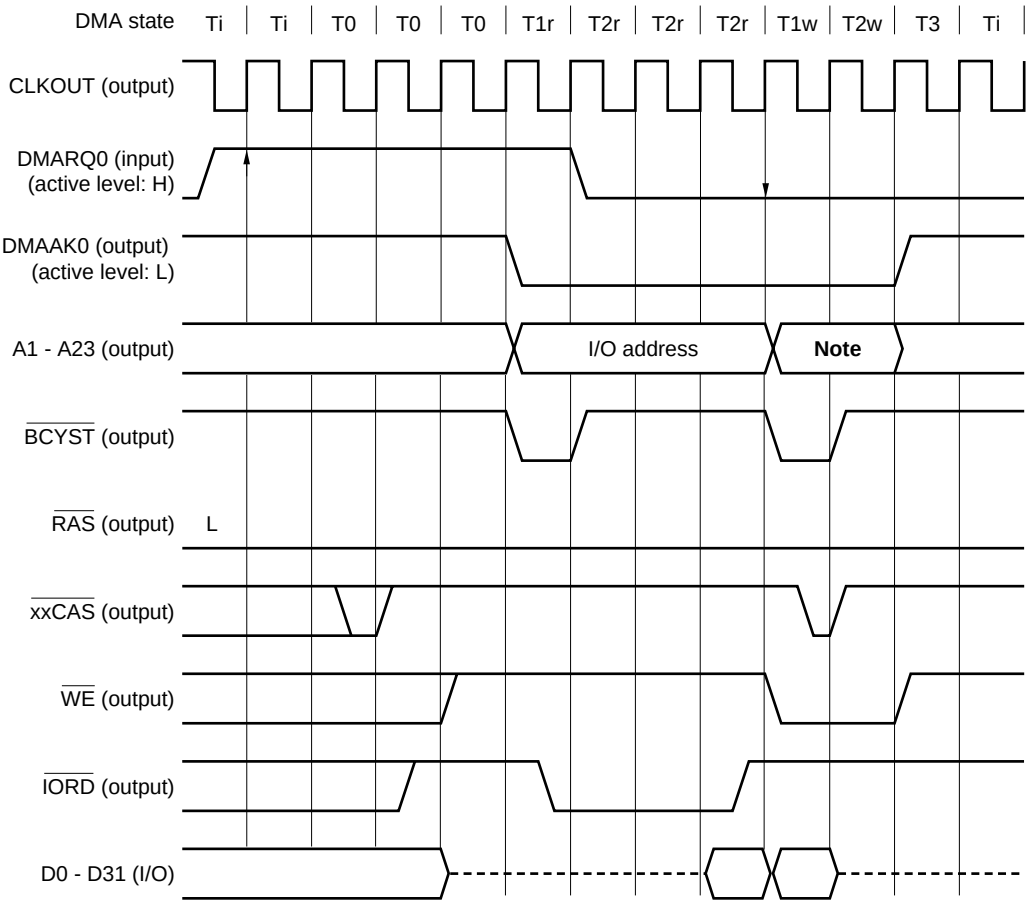
8.5.1 Two-cycle transfer

The two-cycle transfer is to transfer data in two cycles, from the transfer source to the DMAC and from the DMAC to transfer destination.

First cycle : Outputs transfer source address and reads data from transfer source to DMAC
Second cycle: Outputs transfer destination address and writes data from DMAC to transfer destination.

Figure 8-12 shows an example of 2-cycle transfer.

Figure 8-12. Two-Cycle Demand Transfer (external I/O to DRAM (on-page))



Note DRAM column address

Remarks 1. ××CAS: LLCAS, LUCAS, ULCAS, UUCAS

2. The dotted lines indicate the high-impedance state.

3. The arrows indicate the sampling timing.

8.5.2 Subject to transfer

The relation between the transfer type and subject to transfer is shown below.

Subject to DMA Transfer	Transfer Type
Between I/O (external or internal) and memory	2-cycle
Between memory and memory	2-cycle

Cautions 1. The DMAC cannot access the internal RAM.

2. The DMAC cannot access the internal I/O in the CPU core. Use the IN.W/OUT.W instruction to access the internal I/O.

3. Do not write anything to the DMAC register during DMA transfer.

4. If DMA transfer is executed (to write data) to the cacheable area, invalidate the cache as necessary because the values of the memory and cache differ.

8.6 Priorities of DMA Channels

The priorities of the DMA channels are fixed as follows:

DMA channel 0 > DMA channel 1 > DMA channel 2 > DMA channel 3

The DMA request is sampled only at the rising edge of the clock in the Ti state of the DMA state, and at the rising edge of the clock at which the $\overline{\text{BCYST}}$ signal is asserted active in write cycle (sampling in the write cycle is performed only during demand transfer). At this time, the priorities are valid, and DMA transfer with the higher priority is executed starting from the next transfer.

8.7 DMA Transfer Request

The DMA transfer request is issued from three sources: external DMARQ pin, software, or internal peripheral hardware. These sources are specified by the DMA channel control register (DCHC). The DMAAK signal is output regardless of which of the three sources a DMA request has been issued.

8.7.1 Request from DMARQ pin

The request from the DMARQ pin is sampled at the falling edge of the clock in the Ti state of the DMA state. This request must be continuously issued until the corresponding DMAAK signal is asserted active.

If the DMARQ pin becomes active when the DMAC is in the Ti state, the DMAC enters the T0 state and starts DMA transfer.

The request from the DMARQ pin is sampled in the demand transfer mode at the rising edge of the clock at which the $\overline{\text{BCYST}}$ signal is asserted active in the write cycle.

If the next transfer is not executed in the demand transfer mode, check the $\overline{\text{BCYST}}$ signal in the T1r state, and then deassert the DMARQ signal inactive (because the request is sampled at the rising edge of the clock at which the $\overline{\text{BCYST}}$ signal is asserted active in the write cycle, process the request in the read cycle).

When the DMARQ pin is asserted active, DMA transfer is started. However, if the DMARQ pin is left active, the next DMA transfer is started. When designing a system, allow enough time to assert the DMARQ pin active. The following describes the sampling timing of the DMARQ pin for the next DMA transfer in the transfer mode.

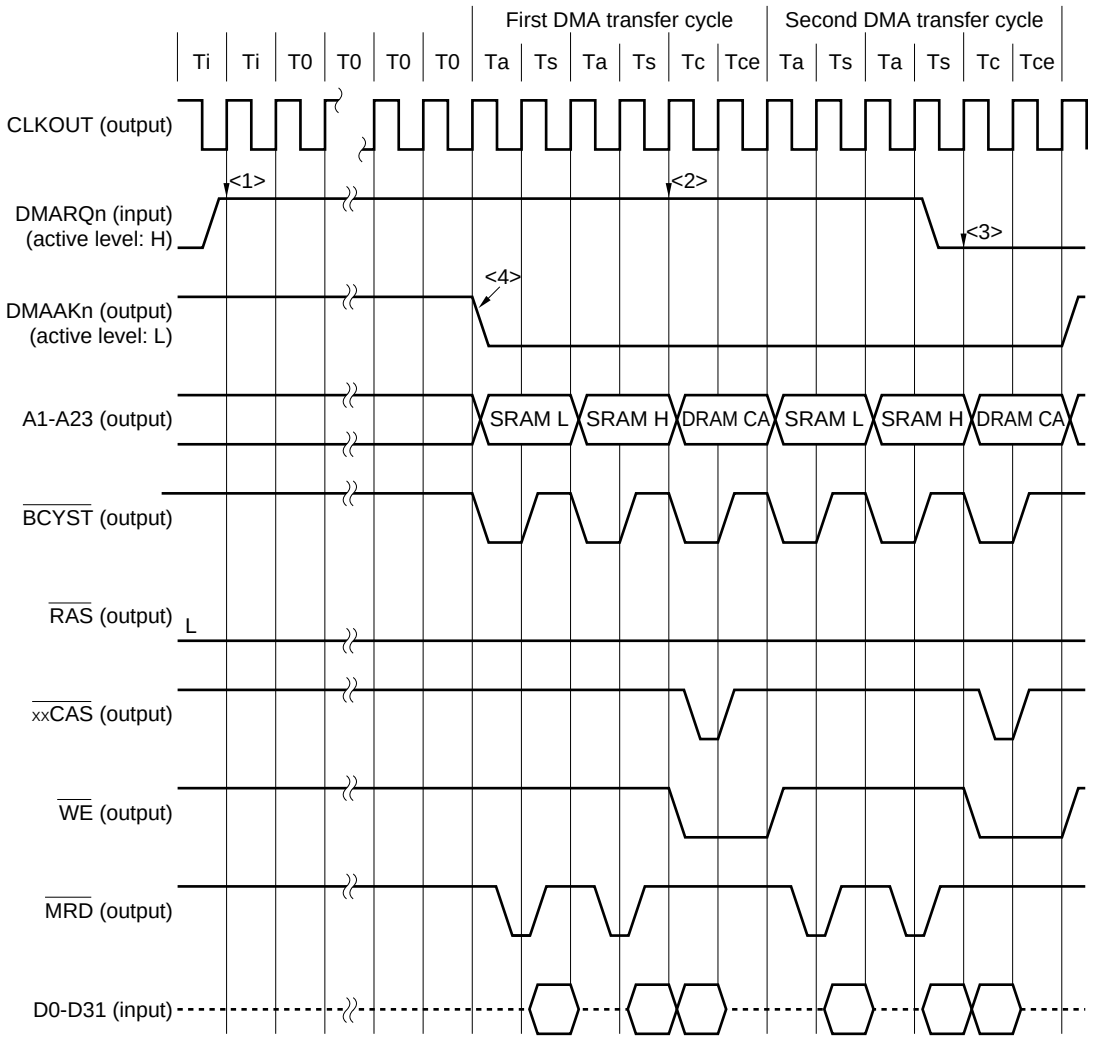
(1) Two-cycle demand transfer

The DMARQ pin is sampled with the timing <1> shown in Figure 8-13 in demand transfer. The DMARQ pin in the second DMA transfer cycle is sampled with the timing <2>.

If the second DMA transfer cycle must not be generated, deassert the DMARQ pin inactive in the timing from <4> to <2>.

The DMA transfer idle state is not set in the timing in Figure 8-13. When idle state is set, an idle state is inserted between the SRAM read and DRAM write (between Ts and Tc).

Figure 8-13. Two-Cycle Demand Transfer (16-bit SRAM to 32-bit DRAM, 32-bit transfer)

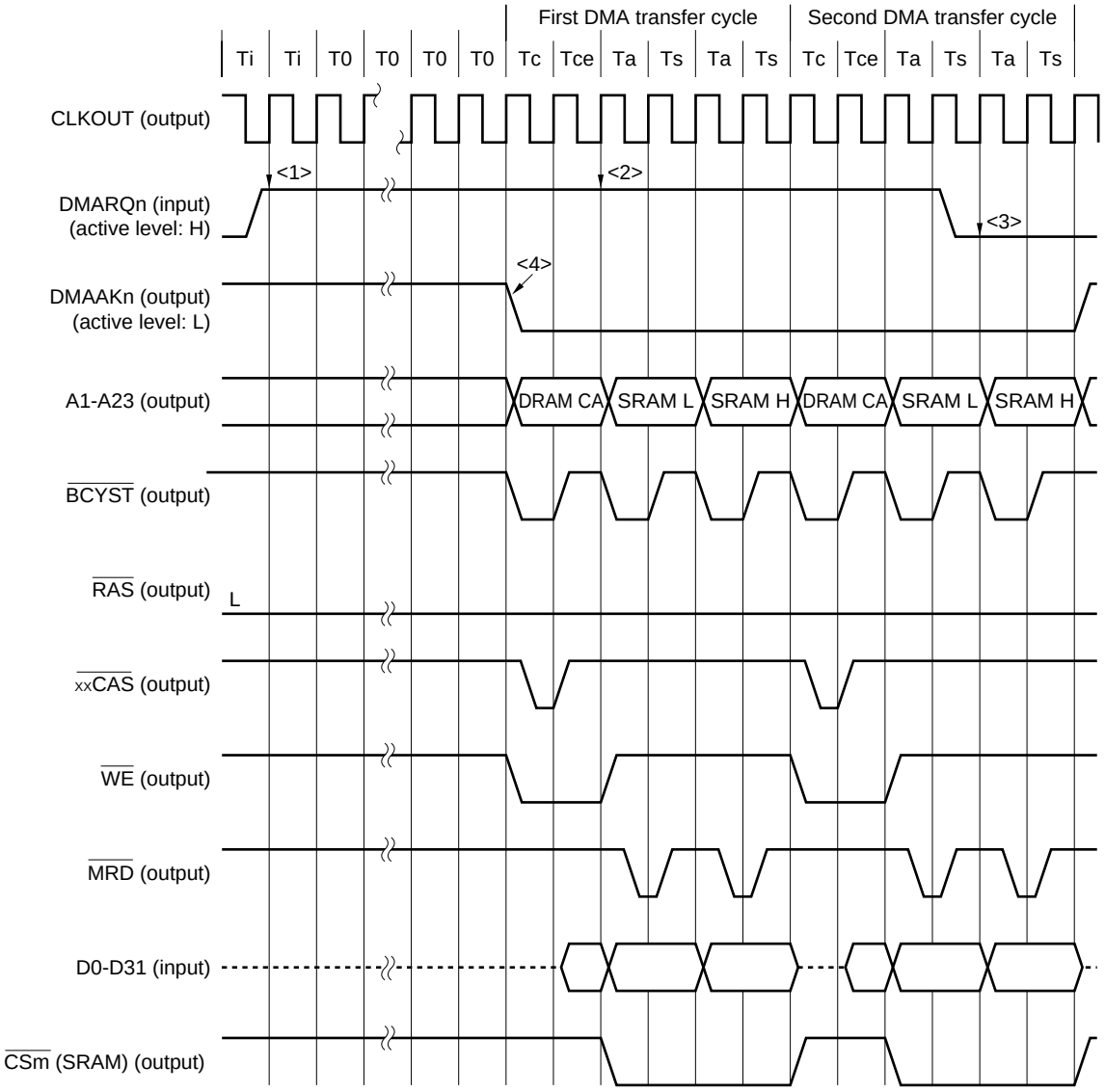


Remarks 1. n = 0 to 3

2. xxCAS: LLCAS, LUCAS, ULCAS, UUCAS

3. The dotted lines indicate the high-impedance state.

Figure 8-14. Two-Cycle Demand Transfer (32-bit DRAM to 16-bit SRAM, 32-bit transfer)



Remarks 1. n = 0 to 3

2. xxCAS: LLCAS, LUCAS, ULCAS, UUCAS

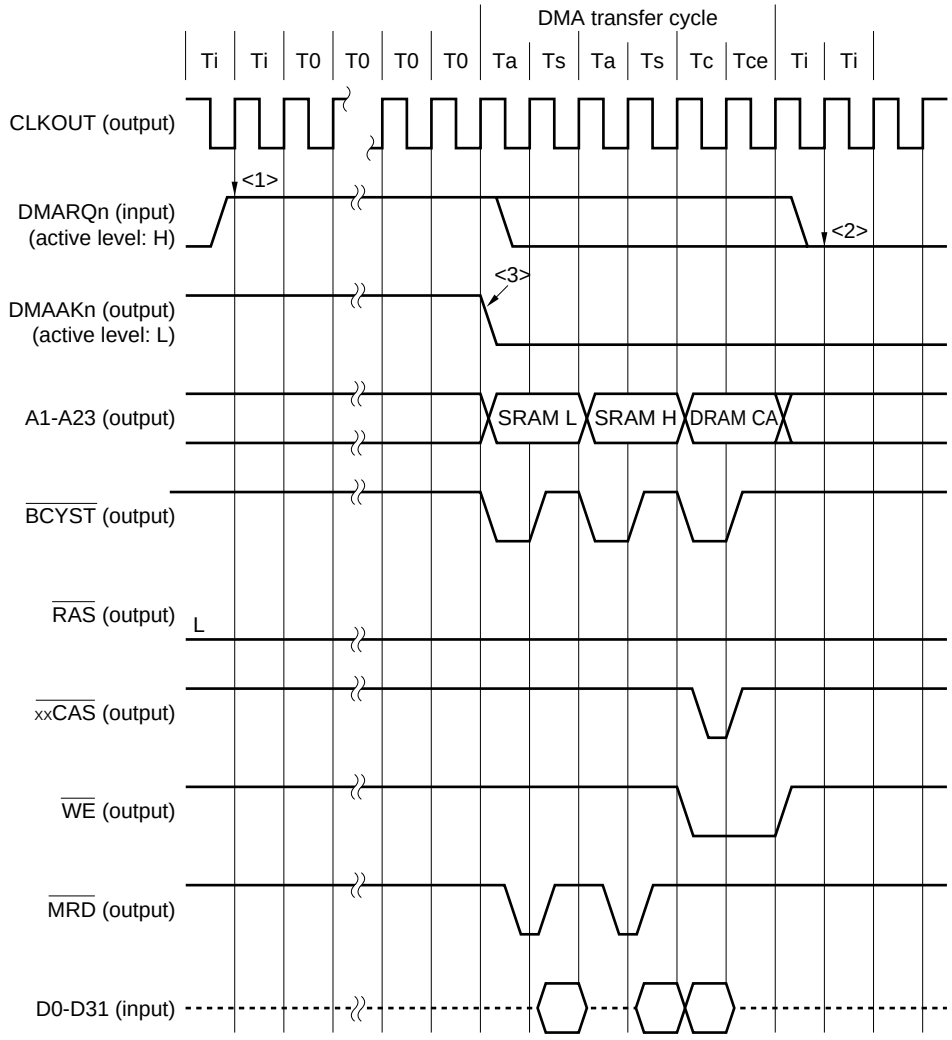
3. m = 1 to 7

4. The dotted lines indicate the high-impedance state.

(2) Single transfer

The DMARQ pin is sampled with the timing <1> in the single transfer shown in Figure 8-15. The state of the DMARQ pin is retained up to the timing <3>. The DMARQ pin is sampled in the second DMA transfer cycle with the timing <2>. If a second DMA transfer cycle must not be generated in single transfer, deassert the DMARQ pin inactive in the timing from <3> to <2>.

Figure 8-15. Single Transfer (16-bit SRAM to 32-bit DRAM, 32-bit transfer)

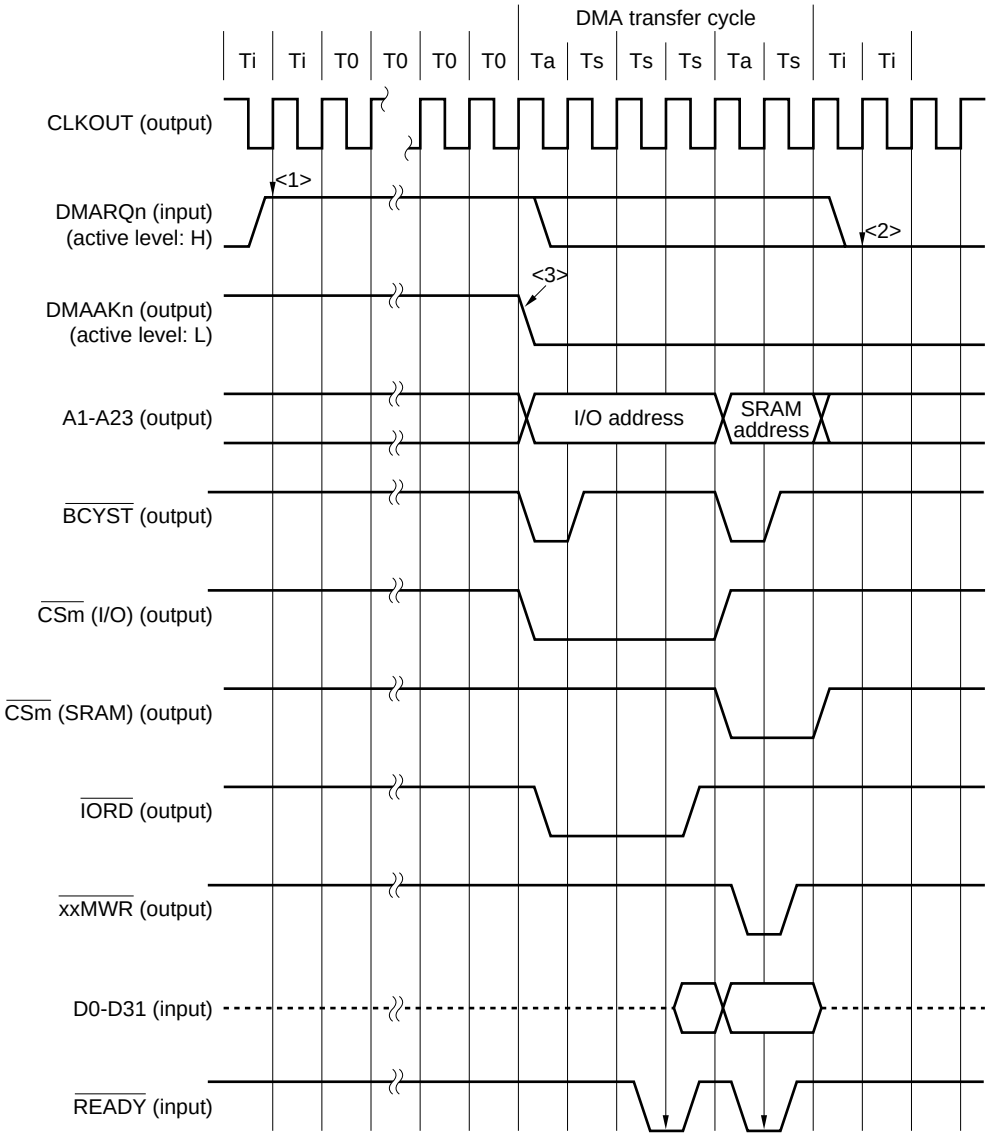


Remarks 1. n = 0 to 3

2. xxCAS: LLCAS, LUCAS, ULCAS, UUCAS

3. The dotted lines indicate the high-impedance state.

Figure 8-16. Single Transfer (8-bit I/O to 32-bit SRAM, 8-bit transfer)



Remarks 1. n = 0 to 3

2. m = 1 to 7

3. ××MWR: LLMWR, LUMWR, ULMWR, UUMWR

4. The arrows indicate the sampling timing.

5. The dotted lines indicate the high-impedance state.

8.7.2 Request from software

When a DMA request is issued from software and when the EN bit of the DCHC register is set to 1, DMA transfer is started.

8.7.3 Request from internal peripheral hardware

The following four types of transfer request signals (interrupt request signals) are issued from the internal peripheral hardware.

- Transmit end interrupt from UART (INTST)
- Receive end interrupt from UART (INTSR)
- Transmit/receive end interrupt from CSI (INTCSI)
- Compare register 4 (CM4) coincidence interrupt from RPU (INTCM4)

Transfer is executed once each time a transfer request is issued from the internal peripheral hardware. Even if the next transfer request is issued before ongoing transfer is completed, that request is ignored. The transfer request from one internal peripheral hardware unit cannot be used with multiple channels.

Use the single transfer mode.

★ The transfer request from the internal peripheral hardware is generated even if it is masked by the interrupt request mask register (IMR). When an internal peripheral hardware interrupt request is used as a DMA transfer request signal, an interrupt request is generated. If an interrupt should not occur from the same internal peripheral when a transfer request from the internal peripheral hardware is used, mask the corresponding interrupt request by using the interrupt mask register.

Caution Issue a transfer request from the internal peripheral hardware after setting of DMA transfer (including enabling transfer) has been completed. If a transfer request from the internal peripheral hardware is generated before DMA transfer is set, transfer is executed immediately after the setting of DMA transfer. Here are two examples of this.

Example 1. To process data of CSI with two tasks (tasks 1 and 2)

Task 1: Software transfer from CSI by interrupt servicing of CPU

Task 2: DMA transfer from CSI

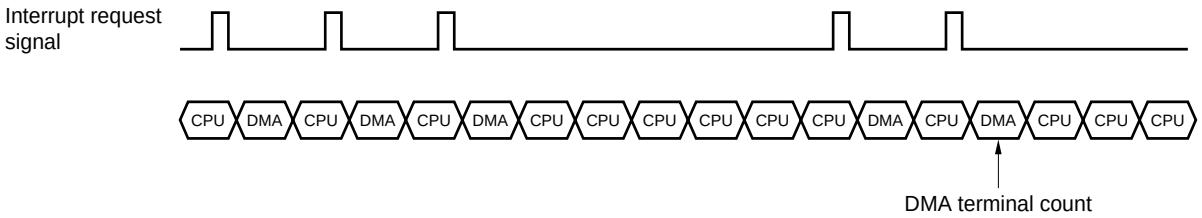
If DMA transfer of task 2 is set after data has been received to CSI and the data of CSI has been received by task 1 (software transfer), the DMA transfer request by task 1 is retained. Consequently, DMA transfer is executed, even if data has not been received to CSI, as soon as DMA transfer of task 2 has been set.

If starting DMA transfer from the DMARQ pin is set first and then the setting is changed to starting DMA transfer from the internal peripheral hardware, and if an interrupt occurs even once from the internal peripheral hardware, DMA transfer is started as soon as the setting has been changed to starting it from the internal peripheral hardware.

Example 2. To execute DMA transfer by INTCM4 for a certain period while a timer (CM4) operates

Because the transfer request by the timer interrupt is retained by the interrupt request register (IRR), DMA transfer is started as soon as DMA transfer has been enabled.

Figure 8-17. Example of Transfer on Request from Internal Peripheral Hardware



8.8 DMA Transfer End Interrupt

When DMA transfer is completed and the TCn bit of the DC register is set to 1, a DMA transfer end interrupt request is issued to the Interrupt controller (refer to **Table 4-2. Interrupt List (Maskable Interrupts)**).

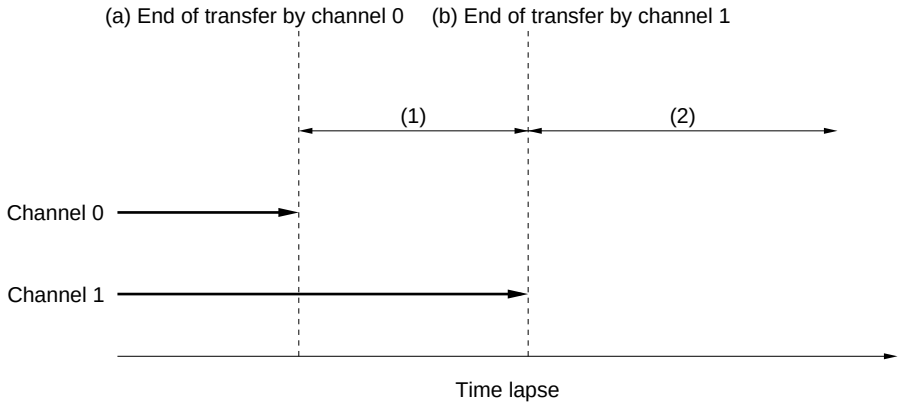
8.8.1 TCn bit reference and DMA transfer end interrupt

The DMAC of the V831 generates a DMA transfer end interrupt (INTDMA) and sets the bit of the channel corresponding to the TCn bit of the DMA control register (DC) to 1 when DMA transfer is completed. In a system where two or more DMA channels are used, the timing to read the TCn bit in the INTDMA handler and the timing to clear the interrupt latch of INTDMA must be carefully determined (n = 0 to 3).

For example, when DMA transfer is executed by using two channels (channels 0 and 1), process the interrupt handler of INTDMA in the following sequence:

<1> Clear the interrupt latch of the DMA transfer end interrupt (INTDMA).
<2> Read the TCn bit of the DMA control register (DC).
<3> Process transfer completion of all the DMA channels corresponding to the bit for which TCn is set.

Figure 8-18. Transfer End Processing of Channels 0 and 1



To process transfer completion by INTDMA during period (1)

Because the TCn bit of only channel 0 is set, the transfer completion processing of only channel 0 is performed.

INTDMA occurs again in the timing when DMA transfer of channel 1 has been completed (b), and transfer completion processing of channel 1 is performed.

To process DMA transfer completion for the first time during period (2)

If the TCn bit is read during the period of <2>, transfer completion of both channels 0 and 1 is processed by the handler of INTDMA because both channels 0 and 1 are set.

To process DMA transfer completion during period of (2), interrupts (a) and (b) occur. Because the interrupt request of interrupt (b) is already active in the timing of (a), the interrupt request is overwritten in the timing of (b), and the CPU cannot identify interrupt (b).

Caution Read the TCn bit of the DC register after clearing the interrupt latch of INTDMA. If this sequence is reversed, and if transfer by channel 1 has been completed immediately after the DC register has been read, the interrupt of channel 1 does not occur and completion processing of channel 1 cannot be performed because the interrupt latch of INTDMA is cleared.

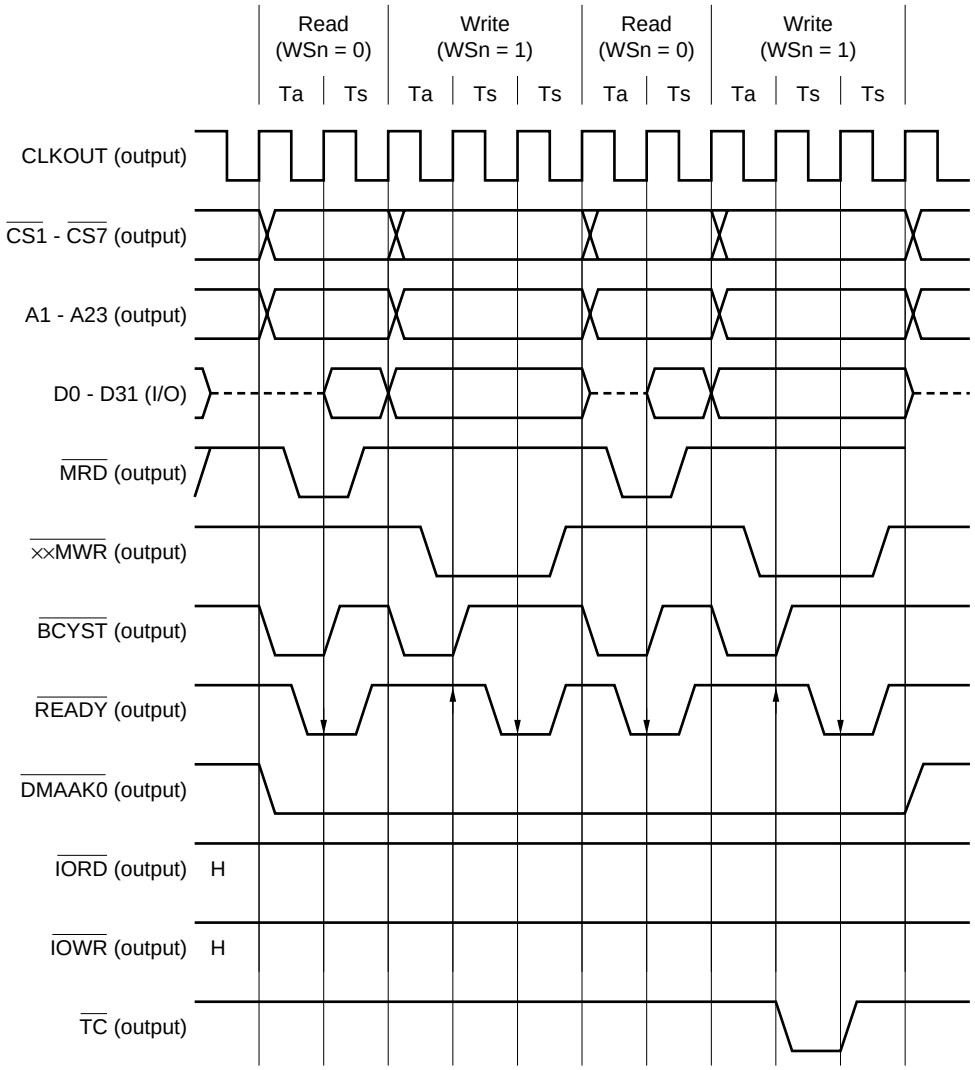
8.9 DMA Transfer End Output

The $\overline{TC}$ signals become active for the duration of one clock at the clock next to the one at which the $\overline{BCYST}$ signal is asserted active in the write cycle in which DMA transfer is completed (the $\overline{TC}$ signal is output at the second clock of the write cycle when the internal peripheral I/O is written). Figure 8-19 shows the output timing of the $\overline{TC}$ signal when transfer is executed from an SRAM area to another SRAM area.

The $\overline{TC}$ signal is output by ORing the DMA transfer end outputs of channels 0 through 3. The DMA transfer end outputs of channels 0 through 3 can be created by ANDing the DMAAK0 through DMAAK3 signals with an external circuit.

The function of the pin multiplexed with the $\overline{TC}$/$\overline{REFRQ}$ pin is selected by using the RFTC bit of the RFC register. Because the $\overline{REFRQ}$ pin is selected (RFTC bit = 1) as default assumption, the $\overline{REFRQ}$ signal is output.

Figure 8-19. DMA Transfer End Output Timing



Remarks 1. ××MWR: LLMWR, LUMWR, ULMWR, UUMWR

2. The dotted lines indicate the high-impedance state

3. The arrows indicate the sampling timing.

8.10 Abort

8.10.1 Aborting by $\overline{\text{NMI}}$ signal

DMA transfer under execution can be forcibly stopped by inputting the $\overline{\text{NMI}}$ signal (in this case, the transfer is aborted after the write cycle has been completed). At this time, the DMAC clears the MEN bit of the DC register to disable DMA transfer. If the MEN bit is set to 1 again, the DMA transfer can be resumed from where it has been aborted.

If the $\overline{\text{NMI}}$ signal is input while the non-maskable interrupt is serviced, the CPU core keeps this interrupt pending. However, a request to abort DMA transfer is not kept pending. Therefore, even if the $\overline{\text{NMI}}$ signal is input while the MEN bit is 0, the request to abort DMA transfer is ignored.

The MEN bit is not cleared to 0 even if the $\overline{\text{NMI}}$ signal is input in the STOP mode (while the bus clock is stopped). Even if the bus clock output is released, the MEN bit is cleared to 0 if the $\overline{\text{NMI}}$ signal is active.

8.10.2 Temporary stop by $\overline{\text{HLDRQ}}$ signal or refresh

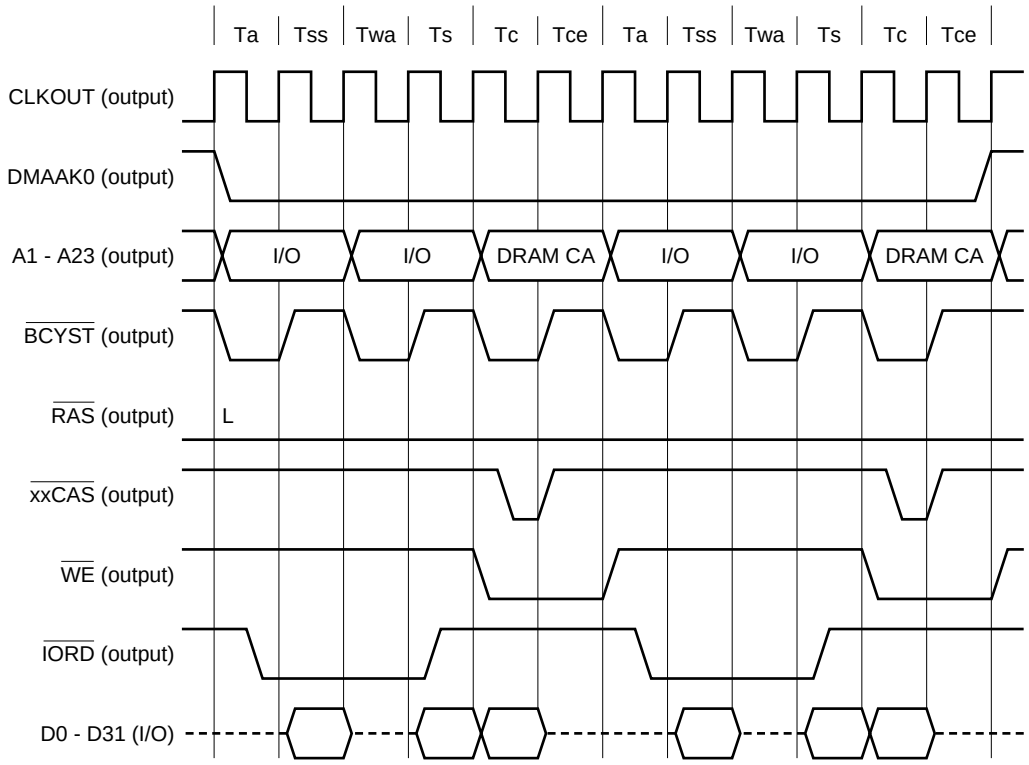
DMA transfer can be temporarily stopped (after the write cycle has been completed) by inputting the $\overline{\text{HLDRQ}}$ signal or issuing a DRAM refresh request (refer to **5.9 Bus Arbitration**). If the DMA request is active when a bus master having a priority higher than the DMAC has released the bus, the DMA transfer is resumed. The DMAAK signal is deasserted inactive while the DMA transfer is temporarily stopped.

8.11 Bus Sizing during DMA Transfer

The V831 has a bus sizing function that selects a bus width from 32 bits or 16 bits. To access 32-bit data via the 16-bit data bus, the data is accessed two times. These two accesses are made in the burst cycle.

Figure 8-20 shows an example of 32-bit (word) demand transfer where the data at the transfer source is 16 bits long (I/O) and the transfer destination data bus width is 32 bits (DRAM).

★ **Figure 8-20. 16- to 32-Bit Data Bus Width (32-bit transfer bus sizing)**



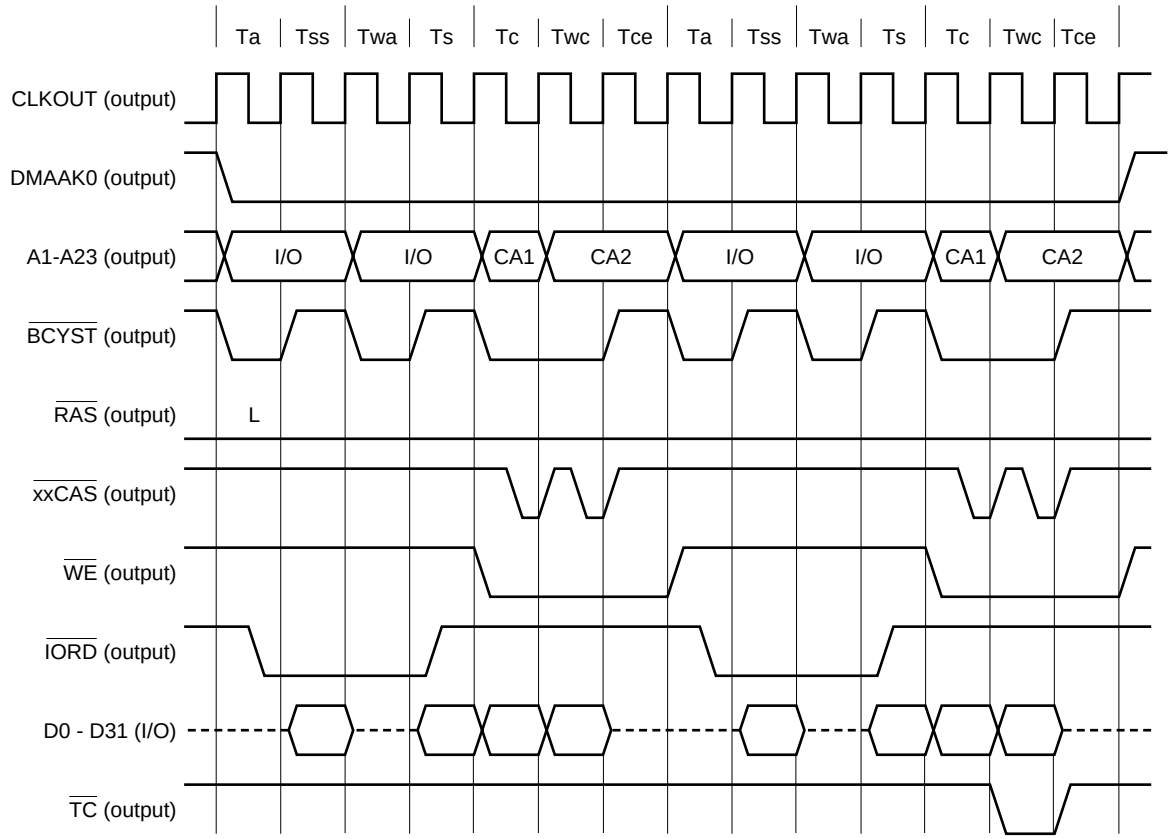
Remarks 1. xxCAS: LLCAS, LUCAS, ULCAS, UUCAS

2. The dotted lines indicate the high-impedance state.

Figure 8-21 shows an example of 32-bit (word) demand transfer where the data at the transfer source is 16 bits long (I/O) and the data bus width at the transfer destination is 16 bits (DRAM).

If the DRAM write cycle is used as burst cycle as a result of bus sizing, the output timing of the TC signal is the Twc state. In a cycle other than the DRAM write cycle, the TC signal is output in the first write cycle.

★ **Figure 8-21. 16- to 16-Bit Data Bus Width (32-bit transfer bus sizing)**



Remarks 1. xxCAS: LLCAS, LUCAS, ULCAS, UUCAS

2. The dotted lines indicate the high-impedance state.

CHAPTER 9 SERIAL INTERFACE FUNCTION

The V831 has two transmit/receive channels to provide serial interface functions. As the interfacing modes, the following two types are available with each type provided by one channel.

- Asynchronous serial interface: UART (Universal Asynchronous Receiver/Transmitter)
- Clocked serial interface : CSI

One channel of BRG (baud rate generator) is provided and can be used exclusively with UART and CSI.

9.1 Asynchronous Serial Interface (UART)

9.1.1 General

The UART of the V831 has the following features:

- Transmit buffer register is not provided.
- A dedicated baud rate generator is provided so that any baud rate can be set.

(1) Deletion of transmit buffer

The conventional UART has a transmit buffer and a receive buffer at both the transimit and receive sides. The V831 omits the transmit buffer for the purpose of mitigating the hardware, and starts transmission processing by transferring data to the transmit shift register. Therefore, the transmit enable control function for the transmit buffer and transmission processing control function by the CTS (serial transmission control) pin are also omitted, and these controls should be implemented by software controlling interrupts.

A receive buffer is provided as in the conventional processor.

(2) On-chip dedicated baud rate generator

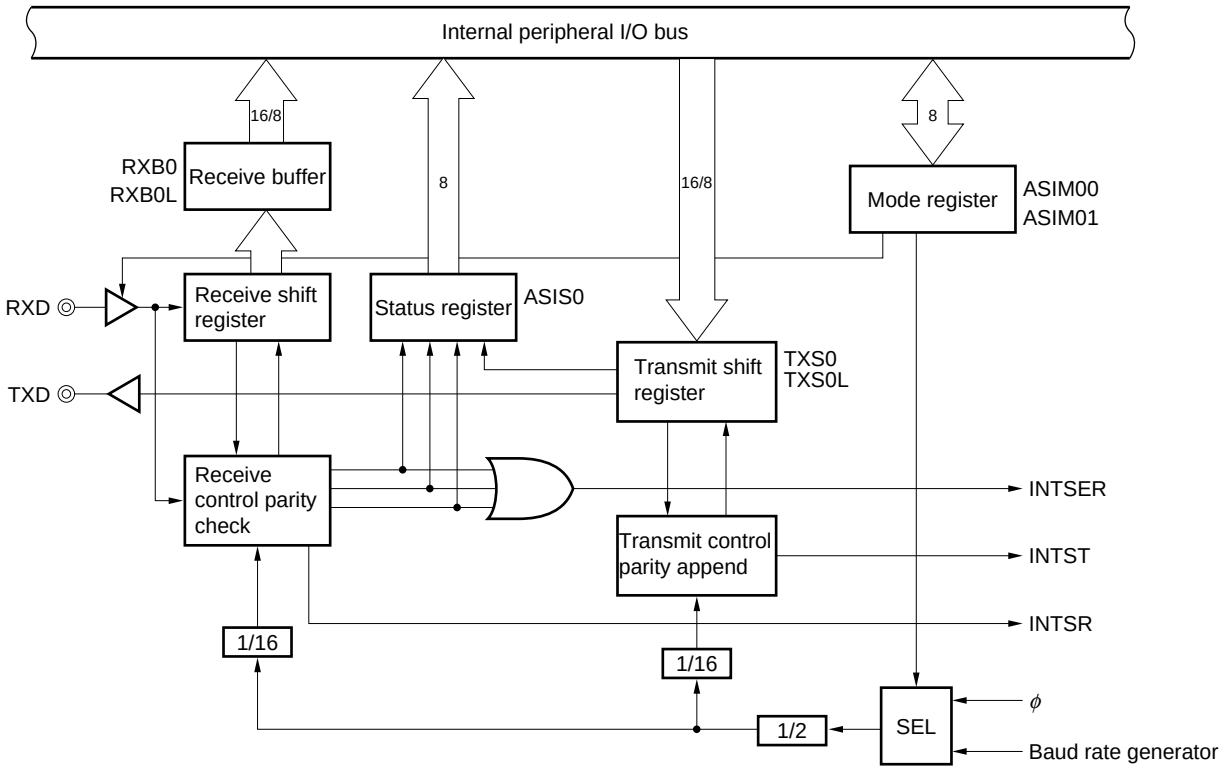
The V831 has one channel of dedicated baud rate generator that generates serial clocks and can set an accurate serial transfer rate.

9.1.2 Features

- Full duplex communication. Receive buffer (RXB) is provided (transmit buffer (TXB) is not provided).
- Two-pin configuration (The UART of the V831 does not have the $\overline{\text{SCLK}}$ and CTS pins.)
  - TXD: Transmit data output pin
  - RXD: Receive data input pin
- Transfer rate: 150 bps to 76800 bps (bus clock: 33 MHz, with BRG)
- Baud rate generator
  Serial clock source can be selected from band rate generator output or bus clock (ϕ)
- Receive error detection function
  - Parity error
  - Framing error
  - Overrun error
- Three interrupt sources
  - Receive error interrupt (INTSER)
    The interrupt is generated by ORing three types of receive errors (for the details of the errors, refer to **9.1.4 (3) Asynchronous serial interface status register (ASIS0)**).
  - Receive end interrupt (INTSR)
    The receive end interrupt request is generated after completion of receive data transfer from the shift register to the receive buffer in the reception enabled status.
  - Transmit end interrupt (INTST)
    The transmit end interrupt is generated after completion of serial transfer of transmit data (9, 8, or 7 bits) from the shift register. The character length of the transmit/receive data is specified by the ASIM00 and ASIM01 registers.
- Character length : 7 or 8 bits
  : 9 bits (with extension bit appended)
- Parity function : Odd, even, 0, or none
- Transmit stop bit : 1 or 2 bits

9.1.3 Configuration

Figure 9-1. Block Diagram of UART



Remark ϕ = bus clock (33 MHz to 16.7 MHz)

9.1.4 Mode registers and control registers

(1) Asynchronous serial interface mode register 00 (ASIM00)

This register specifies the transfer mode of the UART. It can be read or written in 8-bit units.

Caution If the value of ASIM00 is changed during transmission or reception by the UART, the operation is not guaranteed.

Figure 9-2. Asynchronous Serial Interface Mode Register 00 (ASIM00) (1/2)

Address: C0000090H

	7	6	5	4	3	2	1	0
ASIM00	1	RXE0	PS01	PS00	CL0	SL0	0	SCLS0
Initial value:	1	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R	R/W

Bit	Bit Name	Description
6	RXE0	Receive Enable Enables or disables reception. 0: Disables reception 1: Enables reception The receive shift register does not detect the start bit when reception is disabled. The shift in processing and transfer to the receive buffer are not performed, and the contents of the receive buffer are retained. While reception is enabled, the receive shift operation is started in synchronization with detection of the start bit. When reception of one frame has been completed, the contents of the receive shift register are transferred to the receive buffer. In addition, the receive end interrupt (INTSR0) is generated in synchronization with transfer to the receive buffer.
5, 4	PS01, PS00	Parity Select Specifies a parity bit. PS01 / PS00 / Operation 0 / 0 / No parity 0 / 1 / Specifies 0 parity Transmission side → Transmits data with 0 parity bit Reception side → Does not generate parity error on reception 1 / 0 / Specifies odd parity 1 / 1 / Specifies even parity
3	CL0	Character Length Specifies the character length of one frame. 0: 7 bits 1: 8 bits
2	SL0	Stop Bit Length Specifies the stop bit. 0: 1 bit 1: 2 bits

Figure 9-2. Asynchronous Serial Interface Mode Register 00 (ASIM00) (2/2)

Address: C0000090H

	7	6	5	4	3	2	1	0
ASIM00	1	RXE0	PS01	PS00	CL0	SL0	0	SCLS0
Initial value:	1	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R	R/W

Bit	Bit Name	Description
0	SCLS0	Serial Clock Source Specifies a serial clock. 0: Baud rate generator output 1: ϕ (bus clock) The serial clock is generated by dividing the serial clock source specified by the SCLS0 bit of the ASIM00 register by two. The clock resulting from dividing the serial clock by 16 is used as the baud rate clock of the UART. • When SCLS0 = 1 ϕ (bus clock) is selected as the serial clock source. The baud rate can be calculated by the following expression because a sampling rate of $\times$ 16 is used. Baud rate = ϕ/2/16 (bps) The value of the baud rate when a representative clock is used based on the above expression is shown below. ϕ: 33 MHz / 25 MHz / 20 MHz / 16 MHz Baud rate (bps): 1031 K / 781 K / 625 K / 500 K • When SCLS0 = 0 The baud rate generator output is selected as the serial clock source. For details of the baud rate generator, refer to **9.3 Baud Rate Generator**.

(2) Asynchronous serial interface mode register 01 (ASIM01)

This register specifies the transfer mode of the UART. It can be read or written in 8-bit units.

Figure 9-3. Asynchronous Serial Interface Mode Register 01 (ASIM01)

Address: C0000092H

	7	6	5	4	3	2	1	0
ASIM01	0	0	0	0	0	0	0	ESB0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Description
0	ESB0	Extended Bit Select Specifies operation without parity (PS01, PS00 = 00). 0: Disables extended bit operation 1: Enables extended bit operation When the extended bit is specified, 1 data bit is appended to the high-order nibble of the 8-bit transmit/receive data, extending the number of bits to 9 bits. This function is valid only when the operation without parity is specified by the ASIM00 register. When 0 parity, or odd or even parity is specified, the specification by the ESB0 bit is invalid, and the extended bit cannot be appended.

(3) Asynchronous serial interface status register (ASIS0)

This register consists of 3-bit error flags indicating an error status on completion of reception of the UART, and transmit status flags. It can be only read in 8-bit units.

If a receive error occurs, read this register, and then read the receive buffer RXB0 or RXB0L, and clear the error flag to 0.

The status flag indicating a receive error always indicates the latest error. If the same error has occurred several times before the receive data is read, only the status of the error that occurred last is retained.

Figure 9-4. Asynchronous Serial Interface Status Register (ASIS0)

Address: C0000094H

	7	6	5	4	3	2	1	0
ASIS0	SOT0	0	0	0	0	PE0	FE0	OVE0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Description
7	SOT0	Status Of Transmission Status flag indicating the transmission operation status. 1: Transmit start timing 0: Transmit end timing
2	PE0	Parity Error Status flag indicating parity error. 1: Transmit parity and receive parity do not coincide. 0: Data is read from receive buffer.
1	FE0	Framing Error Status flag indicating framing error. 1: Stop bit is not detected. 0: Data is read from receive buffer.
0	OVE0	Overrun Error Status flag indicating overrun error. 1: UART completes next reception before it receives data from receive buffer. 0: Receive data is read from receive buffer. Because the contents of the receive shift register are transferred to the receive buffer each time one frame has been received, the next data is written over the receive buffer and the previous receive data is not retained if an overrun error occurs.

(4) Receive buffer (RXB0, RXB0L)

RXB0 is a 9-bit buffer register and its high-order bits contain 0 when a 7- or 8-bit character is received.

To access this register in half word units (16 bits), specify RXB0. To access it in byte units, specify RXB0L. The receive buffer register can be only read.

While reception is enabled, the receive data is transferred from the receive shift register to the receive buffer in synchronization with the end of shift in processing of one frame. When the data has been transferred to the receive buffer, the reception end interrupt request (INTSR) is generated.

While reception is disabled, the data is not transferred to the receive buffer even after the shift in processing of one frame has been completed, and the contents of the receive buffer are retained. Nor is the reception end interrupt request (INTSR) generated.

RXEB0 is an extended bit. The extended bit is stored to the RXEB0 bit when the extended bit operation is enabled by the ASIM01 register. When the extended bit operation is disabled, 0 is stored to this bit.

Figure 9-5. Receive Buffer (RXB0, RXB0L)

Address: C0000098H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXB0	0	0	0	0	0	0	0	RXEB0	RXB07	RXB06	RXB05	RXB04	RXB03	RXB02	RXB01	RXB00
Initial value:	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: C000009AH

	7	6	5	4	3	2	1	0
RXB0L	RXB07	RXB06	RXB05	RXB04	RXB03	RXB02	RXB01	RXB00
Initial value:	–	–	–	–	–	–	–	–
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Description
8	RXEB0	Receive Extended Buffer Extended bit used when 9-bit character is received. This bit is 0 when 7- or 8-bit character is received.
7 - 0	RXB0n (n = 7 - 0)	Receive Buffer Stores receive data. RXB07 is 0 when 7-bit character is received.

(5) Transmit shift register (TXS0, TXS0L)

TXS0 is a 9-bit shift register for transmit processing. By writing data to this register, the transmit operation is started.

Because the UART of the V831 does not have a transmit buffer, an interrupt request is not generated on completion of transmission (completion of transfer to the buffer), but a transmit end interrupt request (INTST) occurs in synchronization with completion of one frame, including the data of TXS0.

To access this register in half word units (16 bits), specify TXS0. To access it in byte units, specify TXS0L.

TXED0 is an extended bit. The extended bit is stored to the TXED0 bit when the extended bit operation is enabled by the ASIM01 register. When the extended bit operation is disabled, 0 is stored to this bit.

Figure 9-6. Transmit Shift Register (TXS0, TXS0L)

Address: C000009CH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXS0	0	0	0	0	0	0	0	TXED0	TXS07	TXS06	TXS05	TXS04	TXS03	TXS02	TXS01	TXS00
Initial value:	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Address: C000009EH

	7	6	5	4	3	2	1	0
TXS0L	TXS07	TXS06	TXS05	TXS04	TXS03	TXS02	TXS01	TXS00
Initial value:	–	–	–	–	–	–	–	–
R/W:	W	W	W	W	W	W	W	W

Bit	Bit Name	Description
8	TXED0	Transmit Extended Data Extended bit when 9-bit character is transmitted
7 - 0	TXS0n (n = 7 - 0)	Transmit Shifter Writes transmit data

9.1.5 Interrupt requests

The UART generates the following three types of interrupt requests.

(1) Receive error interrupt (INTSER)

A receive error interrupt request is generated by ORing the three types of receive errors while reception is enabled (refer to **Figure 9-4 Asynchronous Serial Interface Status Register (ASIS0)**).

The receive error interrupt request is not generated while reception is disabled.

(2) Receive end interrupt (INTSR)

The receive end interrupt is generated when data is shifted in the receive shift register and transferred to the receive buffer while reception is enabled. The receive end interrupt request also is generated if a receive error occurs. The receive end interrupt request is not generated while reception is disabled.

(3) Transmit end interrupt (INTST)

The UART of the V831 generates the transmit end interrupt request if transmit data of one frame including a 7-, 8-, or 9-bit character is shifted out from the transmit shift register because it does not have a transmit buffer.

The transmit end interrupt request is output when transmission of the last bit of the transmit data is started.

DMA transfer can be executed by using the transmit end interrupt and receive end interrupt (refer to **CHAPTER 8 DMA FUNCTION**).

9.1.6 Basic operation

(1) Transmission

(a) Transmission enabled status

The UART is always in the transmission enabled status. Because the UART of the V831 does not have a CTS (serial transmission control) input pin, use a general-purpose input port to check whether the other party is in the receive enabled status.

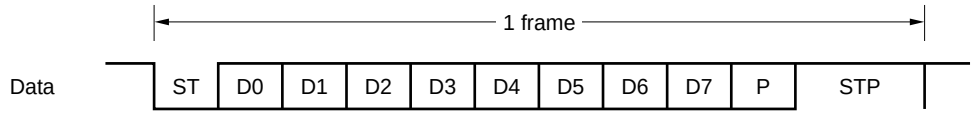
(b) Starting transmit operation

The transmit operation is started when data is written to the transmit shift register (TXS0, TXS0L).

(c) Format of transmit data

Figure 9-7 shows the format of the transmit data. One frame of the data consists of a start bit, character bits, a parity bit, and stop bits.

Figure 9-7. Transmit Data Format of UART



ST : Start bit (1 bit)
D0 - D7: Character bit (7 or 8 bits)
P : Parity, extended bit (odd parity, even parity, 0 parity, no parity, or extended bit)
STP : Stop bit (1 or 2 bits)

(d) Transmit interrupt request

The transmit end interrupt request is generated when one frame of data has been transmitted.

Caution The empty status of TXS0 does not cause the transmit end interrupt. The transmit end interrupt is caused to be generated by completion of transmission of one frame. Therefore, the transmit end interrupt does not occur even if TXS0 is empty after reset.

(2) Reception

(a) Reception enabled status

The receive operation is enabled when the RXE0 bit of the ASIM00 register is set to 1.

- RXE0 = 1 (reception enabled status)
- RXE0 = 0 (reception disabled status)

In the reception disabled status, the reception hardware stands by in the initial status. At this time, the receive end interrupt or receive error interrupt does not occur, and the contents of the receive buffer are retained.

(b) Starting receive operation

The receive operation is started when the start bit is detected.

The RXD pin is sampled by the bus clock from the baud rate generator or serial clock. The RXD pin is sampled again eight serial clocks after the falling edge of the RXD pin has been detected. If the RXD pin is low, it is recognized as the start bit. Then the receive processing operation is started and the RXD pin input is sampled in units of 16 serial clocks.

If a high level of the RXD pin is detected eight serial clocks after the falling edge of the RXD pin has been detected, this falling edge is not recognized as the start bit. The serial clock counter for sample timing generation is initialized and stopped, and waits for input of the next falling edge.

(c) Receive end interrupt request

When one frame of data has been received while reception is enabled (RXE0 = 1), the receive data in the shift register is transferred to RXB0, and the receive end interrupt request is generated.

The receive end interrupt request is not generated while reception is disabled (RXE0 = 0).

(d) Receive error flag

The three error flags (0 through 2 bits of ASIS0 register), parity error, framing error, and overrun error flags, are affected in synchronization with the receive operation. The receive error interrupt is generated as a result of ORing these three error flags.

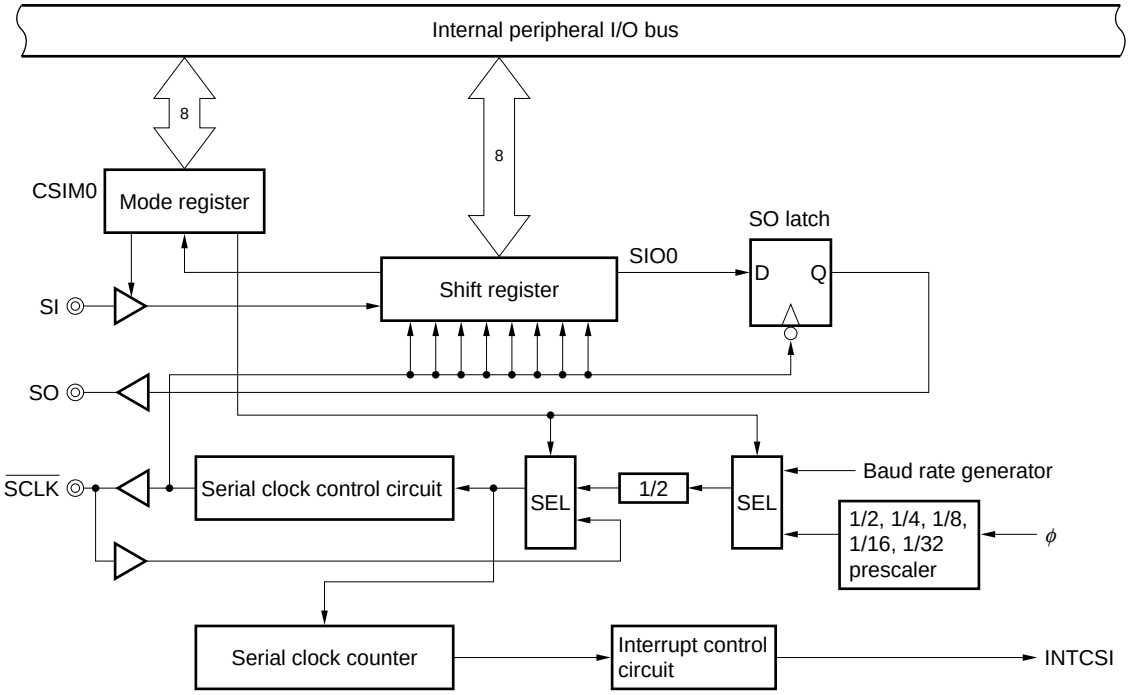
9.2 Clocked Serial Interface (CSI)

9.2.1 Features

- High-speed transfer: 8.25 Mbps MAX. (bus clock: 33 MHz)
- Half duplex communication for transmission/reception (buffer is not provided)
- Character length: 8 bits
- External or internal clock selectable

9.2.2 Configuration

Figure 9-8. Block Diagram of CSI



Remark ϕ = bus clock (33 MHz to 16.7 MHz)

9.2.3 Mode registers and control registers

(1) Clocked serial interface mode register 0 (CSIM0)

This register specifies the basic operation mode of the CSI. It can be read or written in 8-bit units (however, CSOT0 (bit 5) can be only read).

Figure 9-9. Clocked Serial Interface Mode Register 0 (CSIM0) (1/2)

Address: C00000A0H

	7	6	5	4	3	2	1	0
CSIM0	CTXE0	CRXE0	CSOT0	MOD0	0	CLS02	CLS01	CLS00
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Description
7	CTXE0	CSI Transmit Enable Enables or disables transmission. 0: Disables transmission. 1: Enables transmission. The output buffer of the SO pin goes into a high-impedance state when CTXE0 = 0.
6	CRXE0	CSI Receive Enable Enables or disables reception. 0: Disables reception. 1: Enables reception. If the serial clock is input when transmission is enabled (CTXE0 = 1) and reception is disabled, 0 is input to the shift register.
5	CSOT0	CSI Status Of Transmission Indicates that transmission is in progress. 0: End of transmission (write to SIO0 register) 1: Transmission in progress (INTCSI occurs) While transmission is in progress (CSOT0 = 1), the next transmit or receive operation is not started.
4	MOD0	Mode Specifies the first bit. 0: MSB first 1: LSB first

Figure 9-9. Clocked Serial Interface Mode Register 0 (CSIM0) (2/2)

Address: C00000A0H

	7	6	5	4	3	2	1	0
CSIM0	CTXE0	CRXE0	CSOT0	MOD0	0	CLS02	CLS01	CLS00
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Description
2 - 0	CLS02 - CLS00	Clock Source Specifies a serial clock.

CLS02	CLS01	CLS00	Specifies serial clock		$\overline{\text{SCLK}}$ pin
0	0	0	External clock		Input
0	0	1	Internal clock	Specified by BPRM0 register[Note 1]	Output
0	1	0		Reserved (setting prohibited)	—
0	1	1		ϕ/4[Note 2]	Output
1	0	0		ϕ/8[Note 2]	Output
1	0	1		ϕ/16[Note 2]	Output
1	1	0		ϕ/32[Note 2]	Output
1	1	1		ϕ/64[Note 2]	Output

Notes1. For the setting of the BPRM0 register, refer to **9.3 Baud Rate Generator.**

2. ϕ/4, ϕ/8, ϕ/16, ϕ/32, and ϕ/64 are divided signals.

ϕ = bus clock (16 MHz to 33 MHz)

(2) Serial I/O shift register 0 (SIO0)

This register converts parallel data into serial data, or vice versa. It performs shift operation when CTXE0 = 1 or CRXE0 = 1. This register can be read or written in 8-bit units.

Figure 9-10. Serial I/O Shift Register 0 (SIO0)

Address: C00000A2H

	7	6	5	4	3	2	1	0
SIO0	SIO07	SIO06	SIO05	SIO04	SIO03	SIO02	SIO01	SIO00
Initial value:	–	–	–	–	–	–	–	–
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Description
7 - 0	SIO07 - SIO00	Serial I/O Data is shifted in (received) or shifted out (transmitted) from the MSB or LSB.

9.2.4 Pin function

The clocked serial interface (CSI) uses the following pins. These pins are multiplexed with an I/O port. Because the I/O port is selected in the initial status, set the PC2 through PC0 bits of the port control mode register (PC) to use the CSI (refer to **11.2.3 Port control mode register (PC)**).

- SO : Serial data output pin
- SI : Serial data input pin
- $\overline{\text{SCLK}}$: Serial clock I/O pin (Select the mode of this pin by using the CSIM0 register.)

9.2.5 Basic operation

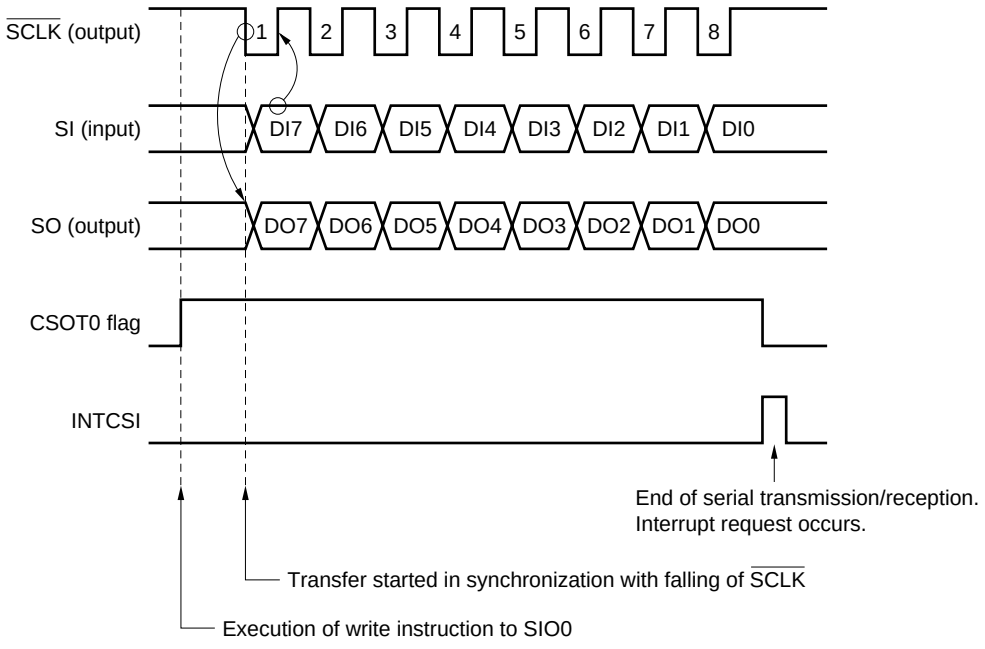
(1) Transfer format

The V831 performs interfacing by using one clock line and two data lines. Serial transfer is started when an instruction that writes transfer data to the SIO0 register is executed.

During transmission, the data is output from the SO pin in synchronization with the falling of the $\overline{\text{SCLK}}$ signal. During reception, the input data of the SI pin is latched in synchronization with the rising of the $\overline{\text{SCLK}}$ pin.

The $\overline{\text{SCLK}}$ signal is stopped when the serial clock counter overflows (at the rising of the eighth count), and an interrupt request signal (INTCSI) is generated.

Figure 9-11. CSI Transfer Timing



(2) Enabling transmission/reception

Because the CSI of the V831 has only one 8-bit shift register and does not have a buffer, transmission and reception are performed simultaneously.

(a) Transmission/reception enabling condition

Transmission is enabled when CTXE0 = 1.

Reception is enabled when CRXE0 = 1.

Transmission and reception are enabled when CTXE0 = CRXE0 = 1.

(i) Disabling SIO output by CTXE0

The serial output goes into a high-impedance state when CTXE0 = 1.

The data of the shift register is output when CTXE0 = 1.

(ii) Disabling SIO input by CRXE0

The shift register input is 0 when CRXE0 = 0.

Serial data is input to the shift register when CRXE0 = 1.

(iii) To check transmit data

When the interface is used as a two-wire interface using the SI and SO signals, the interface receives transmit data by itself, and sets CTXE0 and CRXE0 to 1 to check that a bus conflict does not occur.

(b) Starting transmission/reception

Transmission or reception is started by reading/writing the shift register. When the transmit enable bit (CTXE0) and receive enable bit (CRXE0) are set as follows, starting transmission or reception is controlled.

Table 9-1. Start Condition

CTXE0	CRXE0	Start Condition
0	0	Does not start
0	1	Reads shift register
1	0	Writes shift register
1	1	Writes shift register
0	0 →1	Rewrites CRXE0 bit

Transfer is not started even if CTXE0 is set to 1 after the shift register has been written when CTXE0 is 0.

When the CRXE0 bit is set from 0 to 1 when CTXE0 is 0, the serial clock is generated and reception is started.

The transmit or receive operation is not started when the CSOT0 bit of the CSIM0 register is 1 (transfer is in progress).

★ **Caution When an external clock is used as the serial clock, and if the serial clock input from an external source is stopped before transfer or reception of data has been completed, the CSI of the V831 assumes the next serial clock to be the continuation of the aborted data (serial clock counter is not cleared).**

To stop and redo transfer or reception of such data, initialize the CSI in the following sequence:

- **Clear the CTXE and CRXE bits (temporarily stop transfer/reception).**
- **Set the CTXE and CRXE bits (enable transfer/reception again).**
- **To only transfer or receive data, clear and set the corresponding disable/enable bit.**

9.3 Baud Rate Generator

9.3.1 Configuration and function

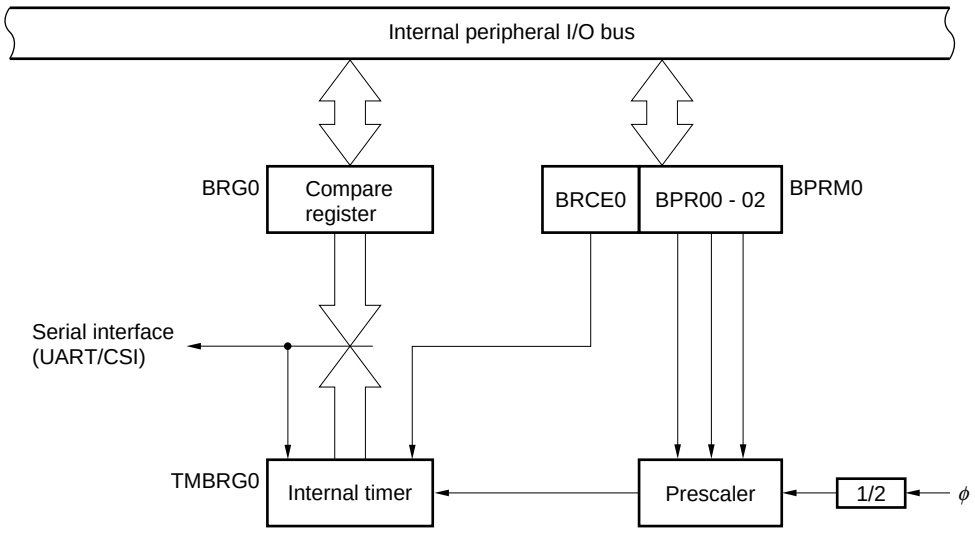
The serial interface can use the serial clock output by the baud rate generator or the divided value of ϕ (bus clock) as a baud rate.

The serial clock source is specified by the following registers.

- In the case of UART: Specified by the SCLS0 bit of the ASIM00 register (refer to **Figure 9-2**).
- In the case of CSI : Specified by the CLS02 through CLS00 bits of the CSIM0 register (refer to **Figure 9-9**).

When the baud rate generator output is specified, the baud rate generator (BRG) is selected as the clock source. The baud rate generator is shared by the UART and CSI.

Figure 9-12. Block Configuration of Baud Rate Generator (BRG)



Remark ϕ = bus clock (33 MHz to 16.7 MHz)

The baud rate generator consists of an 8-bit timer (TMBRG0) that generates a shift clock for transmission/reception, a compare register (BRG0), mode register (BPRM0), and prescaler.

(1) Input clock

The bus clock (ϕ) is input to the BRG.

(2) Setting of BRG

(a) UART

When the baud rate generator is used with the UART, the actual baud rate is calculated by the following expression because a sampling rate of $\times$ 16 is used.

$$\text{Baud rate} = \frac{\phi}{2 \times m \times 2^n \times 16 \times 2} \text{ [bps]}$$

where,

ϕ = Bus clock frequency [Hz]

m = Set value of BRG0 register ($1 \leq m \leq 256$[Note])

n = Set value of prescaler (BPR00 through BPR02 of BPRM0 register) (n = 0, 1, 2, 3, or 4)

Note The value of m = 256 is set by writing 0 to the BRG0 register.

(b) CSI

When the baud rate generator is used with the CSI, the actual baud rate is calculated by the following expression.

$$\text{Baud rate} = \frac{\phi}{2 \times m \times 2^n \times 2} \text{ [bps]}$$

where,

ϕ = Bus clock frequency [Hz]

m = Set value of BRG0 register ($1 \leq m \leq 256$[Note])

n = Set value of prescaler (BPR00 through BPR02 of BPRM0 register) (n = 0, 1, 2, 3, or 4)

Note The value of m = 256 is set by writing 0 to the BRG0 register.

Table 9-2 shows the set values of the baud rate generator when the representative clocks are used.

Table 9-2. BRG Setting Data

Baud Rate [bps]		ϕ = 33 MHz			ϕ = 25 MHz			ϕ = 20 MHz			ϕ = 16 MHz		
UART	CSI	BPR	BRG0	Error	BPR	BRG0	Error	BPR	BRG0	Error	BPR	BRG0	Error
110	1760	–	–	–	4	222	0.02%	4	178	0.25%	4	142	0.03%
150	2400	4	215	0.07%	4	163	0.15%	4	130	0.16%	3	208	0.16%
300	4800	3	215	0.07%	3	163	0.15%	3	130	0.16%	2	208	0.16%
600	9600	2	215	0.07%	2	163	0.15%	2	130	0.16%	1	208	0.16%
1200	19200	1	215	0.07%	1	163	0.15%	1	130	0.16%	0	208	0.16%
2400	38400	0	215	0.07%	0	163	0.15%	0	130	0.16%	0	104	0.16%
4800	76800	0	107	0.39%	0	81	0.47%	0	65	0.16%	0	52	0.16%
9600	153600	0	54	0.54%	0	41	0.76%	0	33	1.36%	0	26	0.16%
10400	166400	0	50	0.84%	0	38	1.16%	0	30	0.16%	0	24	0.16%
19200	307200	0	27	0.54%	0	20	1.73%	0	16	1.73%	0	13	0.16%
38400	614400	0	13	3.29%	0	10	1.73%	0	8	1.73%	0	7	6.99%[Note]
76800	1228800	0	7	4.09%	0	5	1.73%	0	4	1.73%	–	–	–
153600	2457600	0	3	11.9%[Note]	0	2	27.2%	0	2	1.73%	–	–	–

Note This setting cannot be made because the error rate is too high.

Remark BPR = BPR00 through BPR02 bits of BPRM0 register

(3) Error rate of baud rate generator

The error rate of the baud rate generator can be calculated by the following expression.

$$\text{Error [\%]} = \left(\frac{\text{Actual baud rate (baud rate including error)}}{\text{Desired baud rate (normal baud rate)}} - 1 \right) \times 100$$

Example $\left(\frac{9520}{9600} - 1 \right) \times 100 = -0.833$ [%]

$\left(\frac{5000}{4800} - 1 \right) \times 100 = +4.167$ [%]

(4) Permissible error range of baud rate generator

The permissible range of the baud rate generator depends on the number of bits of one frame.

Basically, the permissible limit is a baud rate error of ± 5 % and a sampling timing of ± 4.5 % at 16 bits. The actual permissible limit, however, is a baud rate error of ± 2.3 %, considering that both the transmission and reception sides include an error.

9.3.2 Baud rate generator compare register (BRG0)

This is an 8-bit compare register that sets the timer/count value of the baud rate generator and can be read/written in 8-bit units.

The internal timer (TMBRG0) is cleared by writing to the BRG0 register. Therefore, this register cannot be rewritten by software during transmission/reception.

Figure 9-13. Baud Rate Generator Compare Register (BRG0)

Address: C00000B0H

	7	6	5	4	3	2	1	0
BRG0	BRG07	BRG06	BRG05	BRG04	BRG03	BRG02	BRG01	BRG00
Initial value:	–	–	–	–	–	–	–	–
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

9.3.3 Baud rate generator prescaler mode register (BPRM0)

This register controls the timer/count operation of the dedicated baud rate generator and selects a count clock. It can be read or written in 8-bit units.

Figure 9-14. Baud Rate Generator Prescaler Mode Register (BPRM0)

Address: C00000B2H

	7	6	5	4	3	2	1	0
BPRM0	BRCE0	0	0	0	0	BPR02	BPR01	BPR00
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Description
7	BRCE0	Baud Rate Generator Count Enable Controls the count operation of BRG. 0: Clears and stops count operation. 1: Enables count operation
2 - 0	BPR02 - BPR00	Baud Rate Generator Prescaler Specifies a count clock to be input to TMBRG0.

BPR02	BPR01	BPR00	Operation
0	0	0	ϕ/2 (n = 0)
0	0	1	ϕ/4 (n = 1)
0	1	0	ϕ/8 (n = 2)
0	1	1	ϕ/16 (n = 3)
1	×	×	ϕ/32 (n = 4)

Remark n: Prescaler set value, ϕ: Bus clock

Caution The count clock cannot be changed during transmission/reception.

CHAPTER 10 TIMER/COUNTER FUNCTION

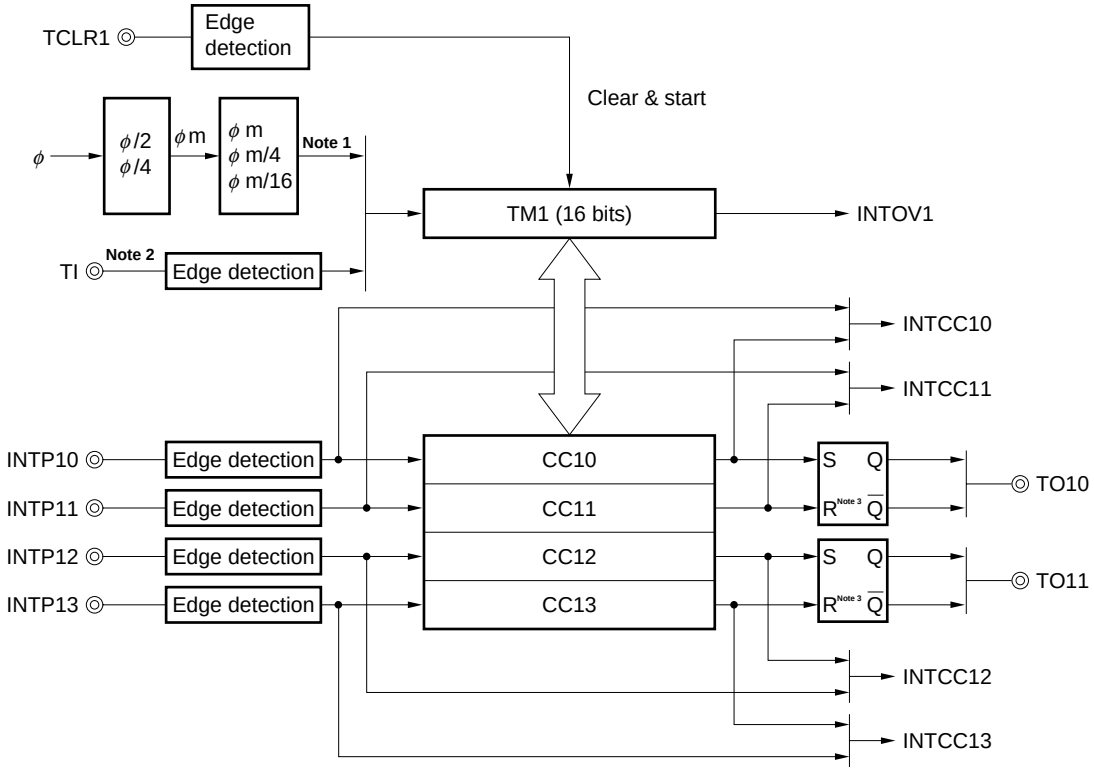
10.1 Features

- Measures pulse interval and frequency and outputs programmable pulse
  - 16-bit measurement
  - Can generate pulses of various shapes (interval pulse, one-shot pulse)
- Timer 1
  - 16-bit timer/event counter
  - Source of count clock : 2 types (selected by dividing system clock, external pulse input)
  - Capture/compare register: × 4
  - Count clear pin : TCLR
  - Interrupt source : 5 types
  - External pulse output : 2 pins
- Timer 4
  - 16-bit interval timer
  - Count clock selected by dividing system clock
  - Compare register: × 1
  - Interrupt source : 1 type

10.2 Configuration

(1) Timer 1 (16-bit timer/event counter)

Figure 10-1. Block Configuration of Timer 1



Notes 1. Internal count clock

2. External count clock (TI: 4.125 MHz max.)

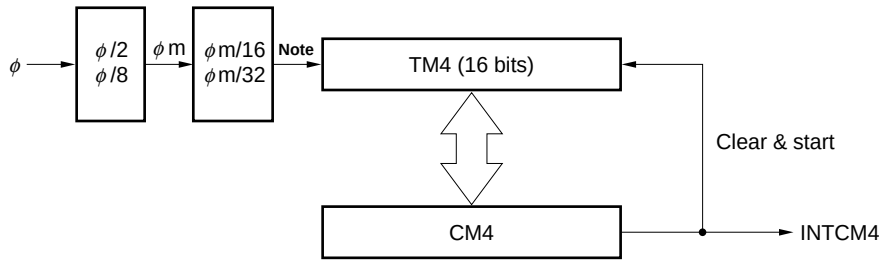
3. Reset priority

Remarks 1. ϕ = bus clock (33 MHz to 16.7 MHz)

2. ϕm = intermediate clock

(2) Timer 4 (16-bit interval timer)

Figure 10-2. Block Configuration of Timer 4



Note Internal count clock

Remarks 1. φ = bus clock (33 MHz to 16.7 MHz)

2. φm = intermediate clock

10.2.1 Timer 1

(1) Timer 1 (TM1)

Timer 1 functions as a 16-bit timer/event counter. It is mainly used to measure cycles and frequencies. It can be also used to output pulses. The TM1 register can be only read in 16-bit units.

Figure 10-3. Timer 1 (TM1)

Address: C0000078H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TM1																
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

The timer can be started or stopped by the timer control 1 (TMC1) (refer to **Figure 10-8. Timer Control Register 1 (TMC1)**). The internal/external count clock can be selected by using the TMC1 register.

(a) External count clock

The TM1 operates as an event counter when an external count clock is selected. The valid edge of the count clock is specified by the timer unit mode register (TUM1) and the value of TM1 is counted up by the TI pin input.

(b) Internal count clock

When the internal count clock is selected, TM1 operates as a free running timer. TM1 is counted up by the internal clock (ϕ/2 to ϕ/64) specified by the TMC1 register.

When the internal clock is selected, ϕm (intermediate clock) is selected from ϕ/2 or ϕ/4 by the prescaler at the preceding stage, and then the final count clock is selected from ϕm, ϕm/4, or ϕm/16 by the prescaler at the following stage. This means that a total of six different count clocks can be selected.

When the timer overflows, an overflow interrupt (INTOV1) can be generated. The timer can be stopped after it has overflown, by using the TUM1 register.

The TUM1 register can also be used to clear and start the timer by an external input signal (TCLR). At this time, the prescaler is also cleared. Therefore, the time from input of the external signal (TCLR) to the first timer count up is fixed depending on the division ratio of the prescaler.

Caution The count clock cannot be changed during timer operation.

(c) Setting of capture/compare register

- As capture register

  An interrupt signal (INTCC10 through INTCC13) is generated by an external input signal (INTP10 through INTP13). The valid edges of INTP10 through INTP13 can be selected from the rising edge, falling edge, or both the rising and falling edges by using the external interrupt mode register (IMOD) (refer to **Figure 4-10. ICU Mode Register (IMOD)**).

- As compare register

  An interrupt signal (INTCC10 through INTCC13) can be generated in response to the coincidence signal from the compare register if so set by the TUM1 register (refer to **Figure10-7. Timer Unit Mode Register (TUM1)**).

Caution Do not change the mode during timer operation.

10.2.2 Timer 4

(1) Timer 4 (TM4)

Timer 4 functions as a 16-bit timer. It is mainly used as an interval timer for software. The TM4 register can be only read in 16-bit units.

Figure 10-4. Timer 4 (TM4)

Address: C000008AH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TM4																
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

The timer can be started or stopped by the timer control register (TMC4). The following count clock can be selected by using the TMC4 register.

First, ϕm (intermediate clock) is selected from ϕ/2 or ϕ/8 by the prescaler at the preceding stage, and then the count clock is selected from ϕm/16 or ϕm/32 by the prescaler at the following stage. This means a total of four types of count clocks can be selected.

The TM4 is cleared to 0 by the coincidence signal issued from the compare register, and counting up is resumed.

Cautions 1. When the TM4 is used as an interval timer, because the timer is cleared by the next count clock if the value of the compare register coincides with the timer register value, the value of the timer may not be 0 even if it has been read immediately after the coincidence interrupt has occurred if the division ratio is too high.

2. The count clock cannot be changed during timer operation.

(2) Compare register (CM4)

CM4 is a 16-bit register and is connected to TM4. It can be read or written in 16-bit units.

Figure 10-5. Compare Register (CM4)

Address: C000008CH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CM4																
Initial value:	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

10.2.3 Capture/compare registers (CC10 through CC13)

The capture/compare registers (CC10 through CC13) can be used as capture registers or compare registers depending on the specification by the TUM1 register. These registers can be read or written in 16-bit units.

Figure 10-6. Capture/Compare Registers (CC10 through CC13)

Address of CC10: C0000080H
Address of CC11: C0000082H
Address of CC12: C0000084H
Address of CC13: C0000086H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC10-CC13																
Initial value:	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

A compare register is used to compare its value with the count value of the timer at each count clock of the timer connected. When the value of the compare register coincides with the count value of the timer, a coincidence signal is issued. The coincidence signal of the compare register is used to generate an interrupt request.

Each compare register has a set/reset output function. The corresponding timer output is set or reset by this function in synchronization with generation of the coincidence signal.

A capture register latches the value of the timer connected (asynchronous with the count clock) when it detects the valid edge of the corresponding interrupt request signal input pin (INTP10 through INTP13), using this signal as a capture trigger.

(1) As capture register

When a capture/compare register is used as a capture register, an interrupt is generated when the valid edge of the input pin (INTP1n) is detected. At this time, an interrupt cannot be generated by the coincidence signal INTCC1n of the compare register (n = 0 to 3).

If the capture (latch) timing of the capture register and a write operation to the register by an instruction contend, the latter takes precedence, and the capture operation is ignored.

(2) As compare register

When a capture/compare register is used as a compare register, coincidence signal INTCC1n or the valid edge of the input pin (INTP1n) can be selected as an interrupt signal by using the TUM1 register. When INTP1n is selected, an external interrupt can be accepted in parallel with the specification of the timer output (n = 0 to 3).

The compare register compares its value with the count value of TM1. A value must be assigned to the CC1n register specified as a compare register. Even when all the four registers CC10 through CC13 are specified as compare registers, for example, and the compare results of only two of them are used, assign an arbitrary value to the remaining two registers. Otherwise, the operation is not guaranteed.

The following table lists the functions of the capture/compare registers and compare registers.

Table 10-1. Capture/Compare Registers

Timer	Register	Timer Restart	Generated Interrupt Signal	Capture Trigger	Timer Output (Set/Reset)	Other Function
TM1	CC10	–	INTCC10	INTP10	TO10 (Set)	–
	CC11	–	INTCC11	INTP11	TO10 (Reset)	–
	CC12	–	INTCC12	INTP12	TO11 (Set)	–
	CC13	–	INTCC13	INTP13	TO11 (Reset)	–
	TM1	–	INTOV1	–	–	External clear
TM4	CM4	○	INTCM4	–	–	–

10.3 Timer/Counter Control Registers

10.3.1 Timer unit mode register (TUM1)

The TUM1 register specifies the operation modes of the capture/compare registers. This register can be read or written in 16-bit units.

Figure 10-7. Timer Unit Mode Register (TUM1) (1/2)

Address: C0000072H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TUM1	0	0	OST	ECLR1	TES11	TES10	CES11	CES10	CMS13	CMS12	CMS11	CMS10	IMS13	IMS12	IMS11	IMS10
Initial value:	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Description
13	OST	Overflow Stop Specifies an operation after the timer has overflowed. This flag is valid only for TM1. 0: Timer continues counting up after overflow. 1: Timer holds 0000H and stops after overflow. At this time, the CE1 bit of the TMC1 register remains 1. The timer resumes counting up by the following operation. • Writing 1 to the CE1 bit (When ECLR1 = 0) • Trigger input to the timer clear pin (TCLR1) (When ECLR1 = 1)
12	ECLR1	External Input Timer Clear Enables clearing the timer by inputting an external clear signal (TCLR) to TM1. 0: Does not clear TM1 by external input. 1: Clears TM1 by external input. After cleared, TM1 starts counting up.
11, 10	TES11, TES10	TI1 Edge Select Specifies the valid edge of the external clock input (TI1).

TES11	TES10	Valid edge
0	0	RFU (reserved)
0	1	RFU (reserved)
1	0	Rising edge (initial value)
1	1	Both rising and falling edges

Bit	Bit Name	Description
9, 8	CES11, CES10	TCLR1 Edge Select Specifies the valid edge of the external clear input (TCLR).

CES11	CES10	Valid edge
0	0	RFU (reserved)
0	1	RFU (reserved)
1	0	Rising edge (initial value)
1	1	Both rising and falling edges

Figure 10-7. Timer Unit Mode Register (TUM1) (2/2)

Address: C0000072H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TUM1	0	0	OST	ECLR1	TES11	TES10	CES11	CES10	CMS13	CMS12	CMS11	CMS10	IMS13	IMS12	IMS11	IMS10
Initial value:	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Description
7 - 4	CMS13 - CMS10	Capture/Compare Mode Select Select the operation modes of the capture/compare registers (CC13 through CC10). 0: Operate as capture registers. However, the capture operation is performed only when the CE bit of the TMC1 register is 1. 1: Operate as compare registers.
3 - 0	IMS13 - IMS10	Interrupt Mode Select Selects INTP1n or INTCC1n as an interrupt source (n = 3 to 0). 0: Uses coincidence signal (INTCC1n) of compare register as interrupt request signal. 1: Uses an external input signal (INTP1n) as an interrupt request signal.

Caution If the CMS1n and IMS1n are changed during the timer operation, these operations are not guaranteed (n = 3 to 0).

10.3.2 Timer control register 1 (TMC1)

TMC1 controls the operation of timer 1 (TM1) and can be read or written in 8-bit units.

Figure 10-8. Timer Control Register 1 (TMC1)

Address: C0000074H

	7	6	5	4	3	2	1	0
TMC1	CE1	0	0	ETI	PRS11	PRS10	PRM11	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Description
7	CE1	Count Enable Controls the timer operation. 0: Timer is cleared to 0000H and does not operate. 1: Timer performs count operation. If ECLR1 bit of TUM1 = 1, however, the timer does not count up until the timer clear signal (TCLR) is input. When the ECLR1 bit of TUM1 is 1, setting the CE1 bit to 1 starts counting of the timer. Therefore, even if the ECLR1 bit is cleared to 0 after the CE1 bit has been set with ECLR1 bit = 1, the timer does not start.
4	ETI	External TI1 Input Specifies an external or internal count clock. 0: Specifies ϕ (internal) 1: Specifies TI1 (external)
3, 2	PRS11, PRS10	Prescaler Clock Select Selects an internal count clock (ϕ m: intermediate clock).
1	PRM11	Prescaler Clock Mode Specifies intermediate clock ϕ m of the count clock (ϕ: bus clock). 0: ϕ/2 1: ϕ/4

PRS11	PRS10	Count clock
0	0	ϕ m
0	1	ϕ m/4
1	0	RFU (reserved)
1	1	ϕ m/16

Caution If the clock is changed during the count operation, the operation is not guaranteed. To change the clock, stop the count operation.

10.3.3 Timer control register 4 (TMC4)

TMC4 controls the operation of timer 4 (TM4) and can be read or written in 8-bit units.

Figure 10-9. Timer Control Register 4 (TMC4)

Address: C0000088H

	7	6	5	4	3	2	1	0
TMC4	CE4	0	0	0	0	PRS40	PRM41	PRM40
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Description
7	CE4	Count Enable Controls the operation of the timer. 0: Timer is cleared to 0000H and stops. 1: Timer performs count operation.
2	PRS40	Prescaler Clock Select Selects an internal count clock (ϕ m: intermediate clock). 0: ϕ m/16 1: ϕ m/32
1, 0	PRM41, PRM40	Prescaler Clock Mode Specifies intermediate clock ϕ m of the count clock (ϕ: bus clock).

PRM41	PRM40	ϕ m
0	0	ϕ/2
0	1	RFU (reserved)
1	0	ϕ/8
1	1	RFU (reserved)

Caution If the clock is changed during the count operation, the operation is not guaranteed. To change the clock, stop the count operation.

10.3.4 Timer output control register (TOC1)

TOC1 specifies timer output modes and can be read or written in 8-bit units.

Figure 10-10. Timer Output Control Register (TOC1)

Address: C0000076H

	7	6	5	4	3	2	1	0
TOC1	ETNO11	ALV11	ETNO10	ALV10	0	0	TOPC11	TOPC10
Initial value:	0	0	0	0	0	0	1	1
R/W:	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Description
7, 5	ENTO11, ENTO10	Enable TO×× pin Enables the corresponding timer output (TO11 or TO10). 0: The timer output is disabled. The corresponding TO11 or TO10 pin outputs a level in reverse phase to the ALV bit (inactive level). Even if a coincidence signal is generated from the corresponding compare register, the level of the TO11 or TO10 pin does not change. 1: The timer output is enabled. The timer output changes if a coincidence signal is generated from the corresponding compare register. A level in reverse phase to the ALV bit (inactive level) is output until the coincidence signal is generated first since the timer output has been enabled.
6, 4	ALV11, ALV10	Active Level TO×× pin Specifies the active level of the timer output. 0: Active level is low (0). 1: Active level is high (1).
1	TOPC11	Prescaler Clock Mode Selects the function of the multiplexed pin TO11/INTP12. 0: TO11 output 1: INTP12 input (initial value)
0	TOPC10	Prescaler Clock Mode Selects the function of the multiplexed pin TO10/INTP10. 0: TO10 output 1: INTP10 input (initial value)

Caution The TO10 and TO11 output does not change with the external interrupt signal (INTP1n). When using TO10 and TO11, specify the capture/compare registers as compare registers (CMS1n = 1) (n = 0 to 3).

Remark Flip-flop of TO10 and TO11 outputs gives priority to reset.

10.3.5 ICU mode register (IMOD)

This control register specifies the valid edge of an external interrupt signal.

When CC1n of TM1 is used as a capture register, this register detects the valid edge of an external interrupt (INTP1n) as a capture trigger (n = 0 to 3). This valid edge can be specified by the ICU mode register (IMOD) (for details, refer to **4.6.5 ICU mode register (IMOD)**).

10.3.6 Timer overflow status register (TOVS)

This register stores overflow flags from timers 1 and 4 (TM1 and TM4). It can be read or written in 8-bit units. Occurrence of an overflow can be polled by testing and resetting this TOVS register by software.

Figure 10-11. Timer Overflow Status Register (TOVS)

Address: C0000070H

	7	6	5	4	3	2	1	0
TOVS	0	0	0	OVF4	0	0	OVF1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R/W	R

Bit	Bit Name	Description
4, 1	OVF4, OVF1	Overflow Flag TMn (n = 4 or 1) overflow flag 0: TMn does not overflow. 1: TMn overflows. Because the TOVS register is of master/slave configuration, data cannot be transferred to the slave (to the TOVS register) during an access period by the CPU. Therefore, even if an overflow occurs while the TOVS register is read, the value of the flag is not affected, but is reflected on the second reading.

Caution TM1 generates an interrupt request signal (INTOV1) to the interrupt controller in synchronization with the overflow. However, the interrupt operation and TOVS are completely independent, and INTOV1 does not occur even if 1 is written to OVF1. The flag can be cleared by writing 0 to OVF1.

10.4 Operation

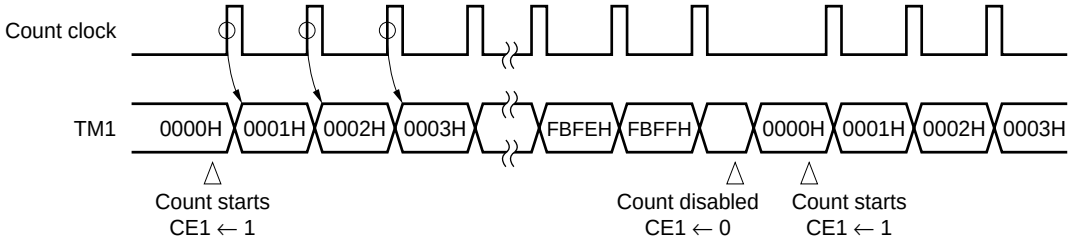
10.4.1 Timer 1

Timer 1 functions as a 16-bit free running timer or an event counter of external signals. The operation of this timer 1 specified by the timer control register 1 (TMC1).

Timer 1 counts up the internal clock (ϕ/2 to ϕ/64) specified by the PRS11, PRS10, and PRM11 bits of the TMC1 register, or external clock input (TI). If the external clock is specified as the count clock at this time, TM1 operates as an event counter. If the timer overflows as a result of counting, an overflow interrupt (INTOV1) is generated.

If the count value of TM1 coincides with the value of the CC10 through CC13 registers when TM1 operates as a free running timer, interrupt signals (INTCC10 through INTCC13) are generated and the timer output signals (TO10 and TO11) can be set or reset. The count value of TM1 can be captured to the CC10 through CC13 registers (capture operation) in synchronization with the valid edge detected from external interrupt request input pins (INTP10 through INTP13) as external triggers. The captured value is retained until the next capture trigger is generated.

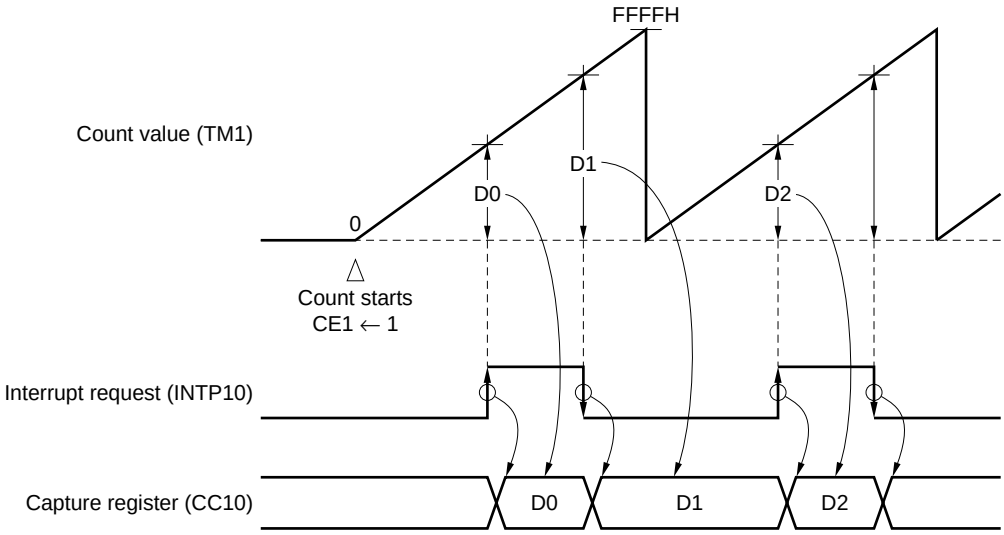
Figure 10-12. Basic Operation of Timer 1



If both the rising and falling edges are specified as the capture triggers, the width of a pulse input from an external source can be measured.

If the rising edge is specified as the capture trigger, the cycle of the input pulse can be measured.

Figure 10-13. Example of Capture Operation



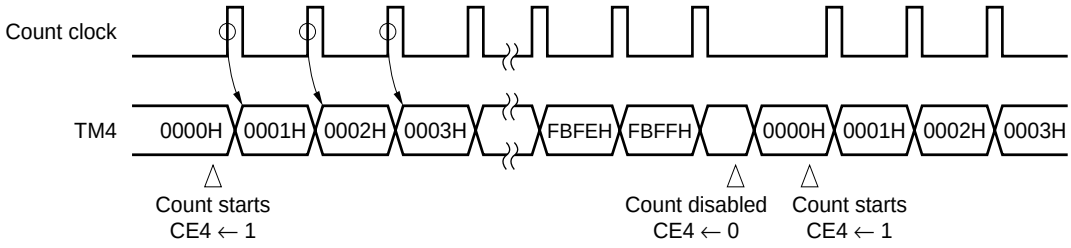
10.4.2 Timer 4

Timer 4 functions as a 16-bit interval timer. The operation of this timer is specified by the timer control register 4 (TMC4).

Timer 4 counts up the internal clock (ϕ/32 to ϕ/256) specified by the PRS40, PRM41, and PRM40 bits of the TMC4 register. If the value of TM4 coincides with the value of CM4 as a result of counting, TM4 is cleared, and at the same time, a coincidence interrupt (INTCM4) occurs.

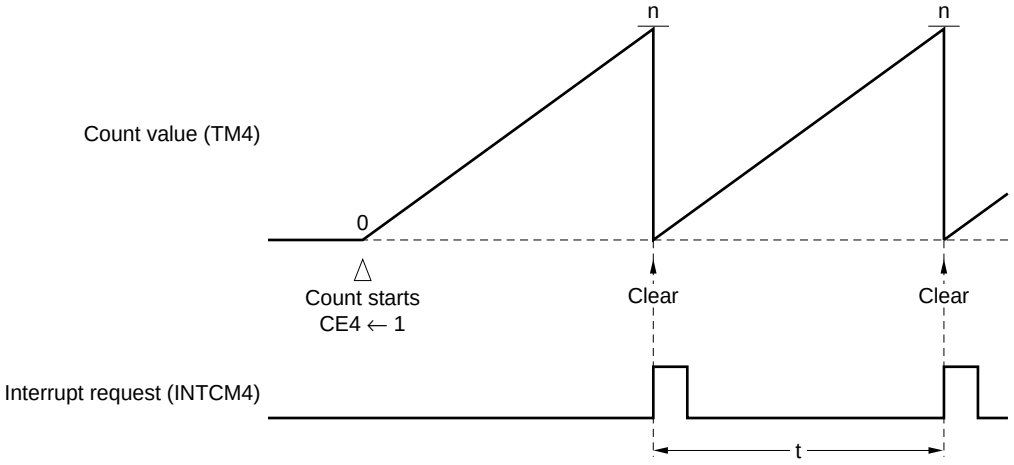
This coincidence interrupt (INTCM4) can be used to execute DMA transfer (refer to **CHAPTER 8 DMA FUNCTION**).

Figure 10-14. Basic Operation of Timer 4



If coincidence is detected as a result of a compare operation, TM4 is cleared to 0 by the next count clock input. This function allows TM4 to operate as an interval timer at a count clock cycle of the value set to CM4 plus 1.

Figure 10-15. Example of Compare Operation

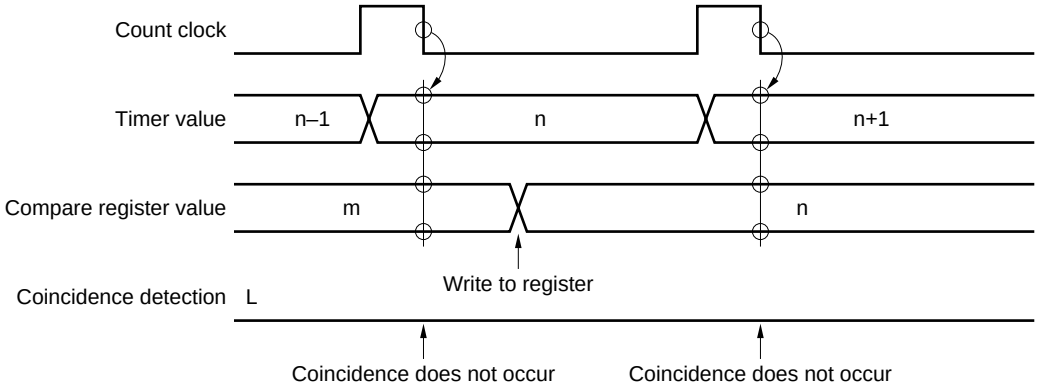


Remark n : Value of CM4 register
t : Interval cycle = (n+1) × Count clock cycle

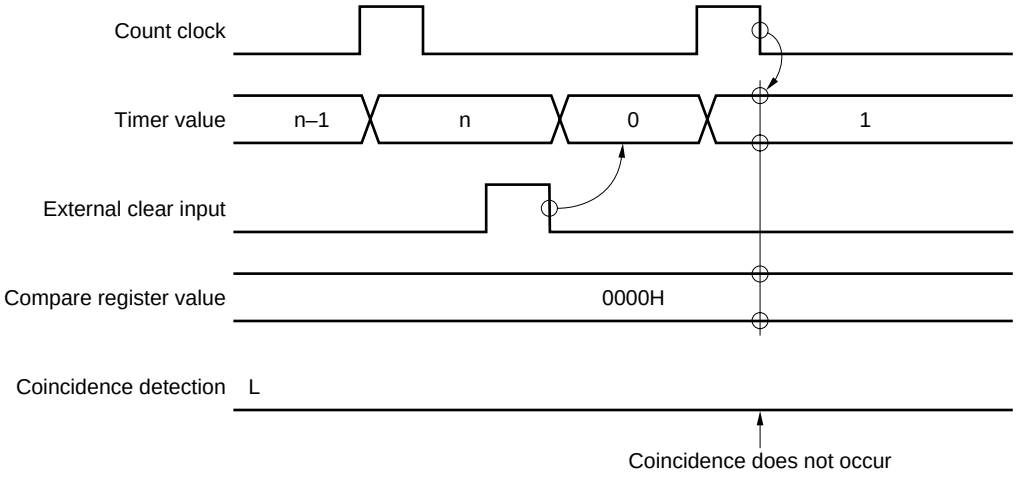
10.5 Notes

Coincidence is always detected by the compare register immediately after the timer has counted up. Coincidence does not take place in the following cases.

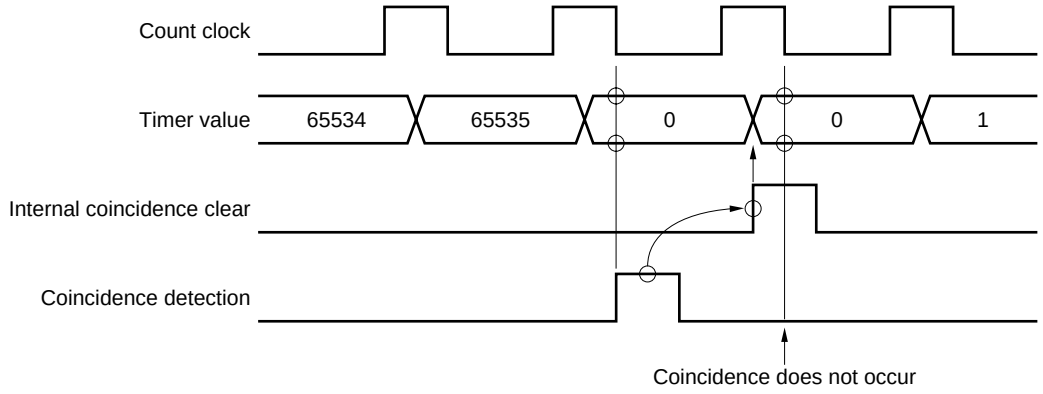
(1) When compare register is rewritten (TM1, TM4)



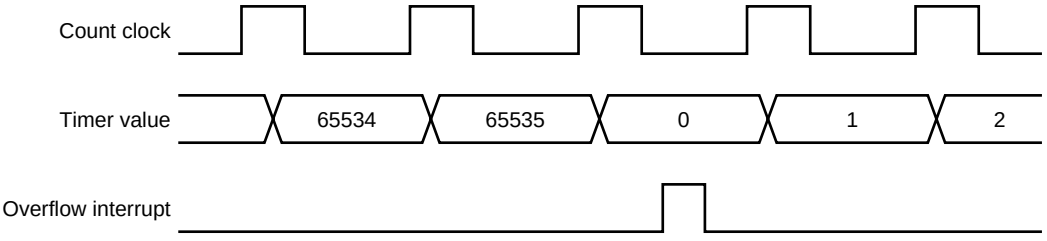
(2) When TM1 is cleared by external signal



(3) When TM4 is cleared (when compare register value = 0000H)



Caution The value of timer is cleared to 0 when the timer overflows if the timer operates as a free running timer.



[MEMO]

CHAPTER 11 PORT FUNCTION

The ports of the V831 are 3-bit I/O ports that can be set in the input or output mode in 1-bit units.

In addition to the port function, these ports are also used as the I/O lines of the clocked serial interface (CSI) in the control mode.

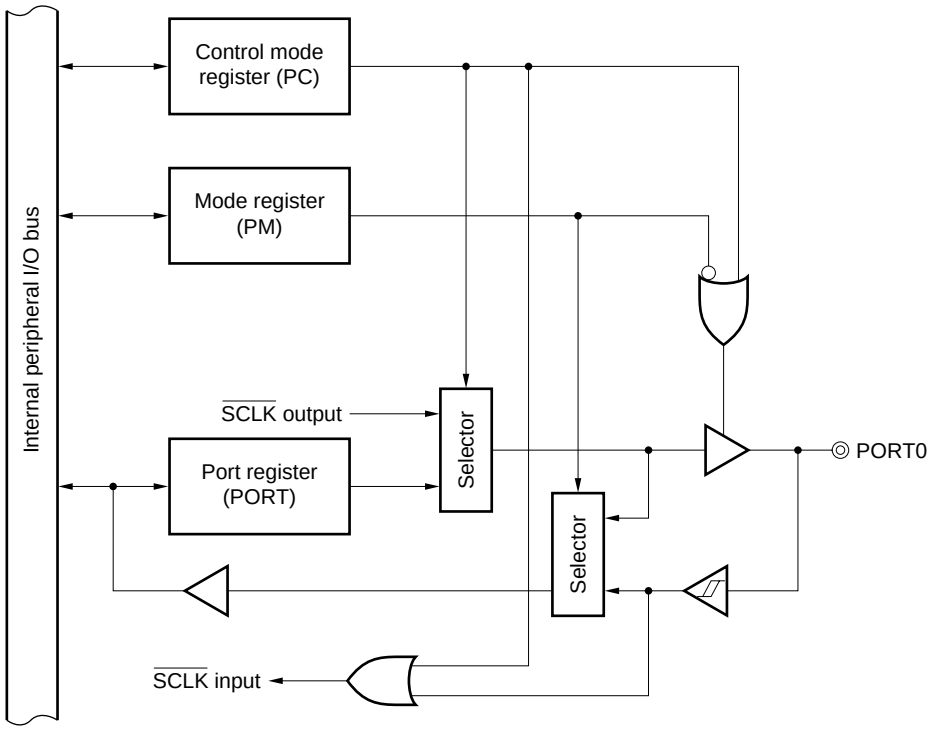
Table 11-1. Operation in Control Mode

Port	Control Mode	Remark
Port 0	$\overline{\text{SCLK}}$	Multiplexed with I/O line of clocked serial interface (CSI)
Port 1	SO	
Port 2	SI	

11.1 Configuration

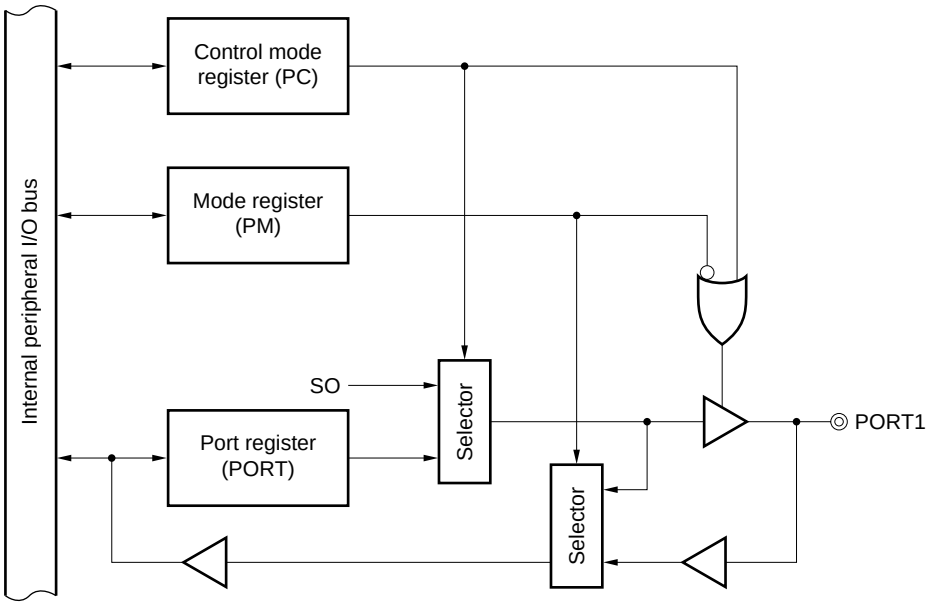
(1) Configuration of port 0

Figure 11-1. Block Diagram of Port 0



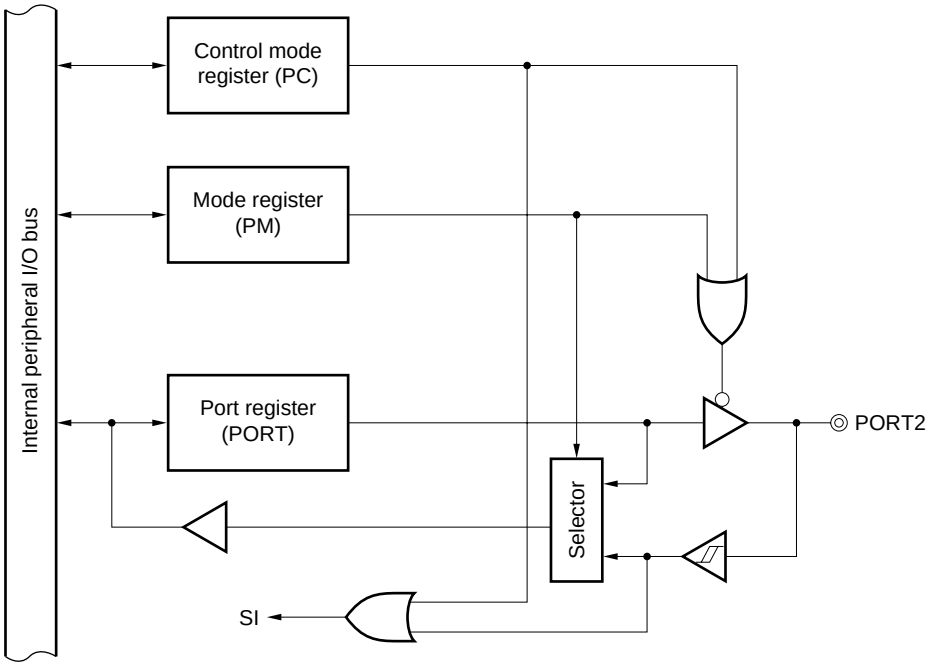
(2) Configuration of port 1

Figure 11-2. Block Diagram of Port 1



(3) Configuration of port 2

Figure 11-3. Block Diagram of Port 2



11.2 Port Control Register

11.2.1 I/O port register (PORT)

This register is for the 3-bit I/O port, PORT, and can be read or written in 8-bit units.

In the input mode, the levels of the port pins can be directly read from this register. However, the register does not retain the values.

In the output mode, the values written to this register are output to the port pins.

Figure 11-4. I/O Port Register (PORT)

Address: C0000000H

	7	6	5	4	3	2	1	0
PORT	0	0	0	0	0	PORT2	PORT1	PORT0
Initial value:	–	–	–	–	–	–	–	–
R/W:	R	R	R	R	R	R/W	R/W	R/W

11.2.2 I/O mode register (PM)

This register sets PORT in the input mode or output mode in 1-bit units, and can be read or written in 8-bit units.

Figure 11-5. I/O Mode Register (PM)

Address: C0000002H

	7	6	5	4	3	2	1	0
PM	1	1	1	1	1	PM2	PM1	PM0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Description
2 - 0	PM2 - PM0	Port Mode Specifies the input/output mode of PORT0 through 2 pins. 0: Output mode (output buffer ON) 1: Input mode (output buffer OFF)

11.2.3 Port control mode register (PC)

This register sets the control modes to select the pot or CSI operation and can be read or written in 8-bit units.

Figure 11-6. Port Control Mode Register (PC)

Address: C0000004H

	7	6	5	4	3	2	1	0
PC	0	0	0	0	0	PC2	PC1	PC0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Description
2 - 0	PC2 - PC0	Port Control Specifies the operation mode of PORT2 through PORT0 pins

PC2	PC1	PC0	Operation mode
0	0	0	I/O port mode
1	1	1	CSI interface
Others			Reserved (setting prohibited)

Caution Set the CLS02 through CSL00 bits of the clocked serial interface mode register (CSIM0) of CSI (refer to 9.2) before setting the port control mode register (PC).

CHAPTER 12 CLOCK GENERATION FUNCTION

The clock generator generates and controls the CPU clock and bus clock (ϕ) supplied to the internal hardware units, including the CPU.

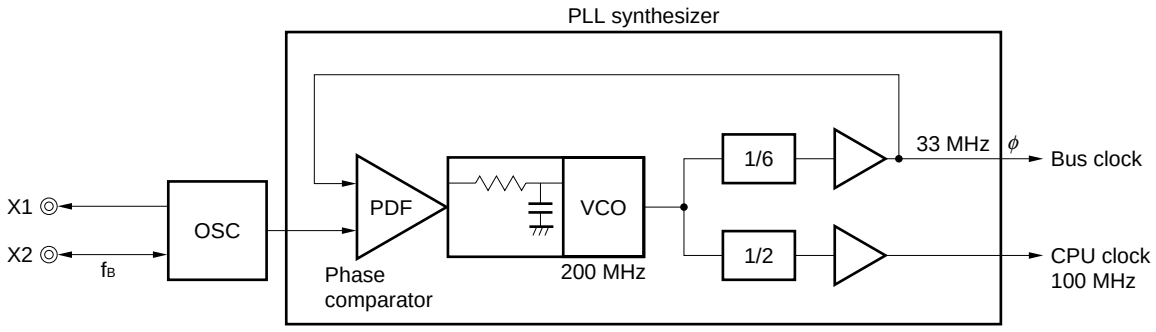
Table 12-1. Multiplication Function by PLL Synthesizer

	Specification
Bus clock (ϕ)	16.7 - 33 MHz (f_B)
CPU clock	50 - 100 MHz ($3 \times f_B$)

Remark PLL: Phase Locked Loop

12.1 Configuration

Figure 12-1. Block Diagram of Clock Generation Function



f_B : Oscillation frequency or external clock frequency

ϕ : Bus clock

OSC: Oscillator

PFD: Phase Frequency Detector

VCO: Voltage Controlled Oscillator

12.2 Selecting Input Clock

The clock generator consists of a clock oscillator and a PLL synthesizer. For example, a bus clock of 33 MHz and a CPU clock of 100 MHz can be generated by connecting a 33 MHz crystal resonator or ceramic resonator between the X1 and X2 pins.

An external clock can be directly connected to the oscillator. In this case, input the clock signal to the X2 pin, and open the X1 pin.

12.2.1 Lockup time

Immediately after the power has been applied or the STOP mode has been released, the PLL is in the phase lock status at a specific frequency, and it takes a certain time (lockup time) until the frequency of the PLL is stabilized. The status until the stabilization is called unlock status, and the stabilization status is called lock status.

Immediately after power application or releasing the STOP mode by the $\overline{\text{RESET}}$ signal, make sure that this stabilization time elapse, by using the $\overline{\text{RESET}}$ signal (refer to **13.4 Ensuring Oscillation Stabilization Time**).

When the STOP mode is released by the $\overline{\text{NMI}}$ signal, the oscillation stabilization time is automatically ensured.

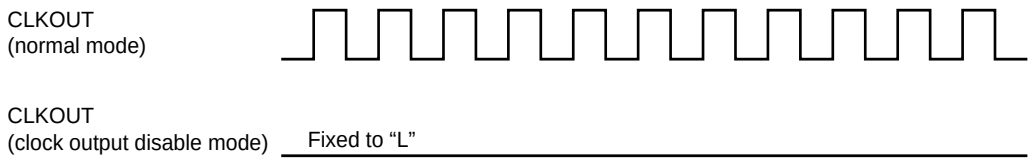
12.3 Clock Output Control

The operations of the CLKOUT pin can be selected by the COE bit of the clock control register (CGC). The power consumption can be effectively reduced by using the standby mode (HALT or STOP) (power management mode).

12.3.1 Clock output disable mode

In this mode, output of the clock from the CLKOUT pin is disabled. Because the operation of CLKOUT is completely stopped, the power consumption can be substantially reduced and the noise radiation from the CLKOUT pin can be suppressed.

Figure 12-2. Clock Output Disable Mode



12.4 Clock Control Registers

12.4.1 Clock control register (CGC)

This register controls output of CLKOUT. It can be read or written in 8-bit units.

Figure 12-3. Clock Control Register (CGC)

Address: C00000E0H

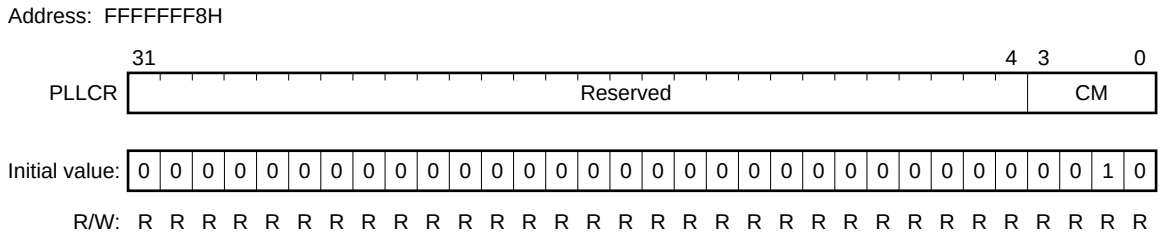
	7	6	5	4	3	2	1	0
CGC	KEY				CESEL	0	0	COE
Initial value:	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R	R	R/W

Bit	Bit Name	Description
7 - 4	KEY	KEY Data Identifies data. Be sure to set these bits to “0111”. No data can be written to the CESEL and COE bits if the value of the KEY data is not “0111”. These bits are always “0000” when read.
3	CESEL	Crystal External Select Specifies the functions of the X1 and X2 pins. 0: Connect a resonator to the X1 and X2 pins. 1: Connect an external clock to the X2 pin and leave the X1 pin unconnected.
0	COE	Clock Out Enable Enables or disables output of CLKOUT. 0: Disables output (CLKOUT pin is fixed to low level). 1: Enables output.

12.4.2 PLL control register (PLLCR)

This register indicates the status of the PLL. This register is mapped to the I/O space and can be only read in 32-bit units.

Figure 12-4. PLL Control Register (PLLCR)



Bit	Bit Name	Description
31 - 4	Reserved	Reserved field
3 - 0	CM	Clock Mode These bits are always “0010” when read.

CM				CPU clock: bus clock
0	0	0	0	Reserved
0	0	0	1	Reserved
0	0	1	0	3:1
0	0	1	1	Reserved
0	1	×	×	Reserved
1	×	×	×	Reserved

CHAPTER 13 STANDBY FUNCTION

The V831 has a standby function to control the operating clock and reduce the power consumption. This function can be used in the following two modes:

- HALT mode (stops only the CPU clock)
- STOP mode (stops the entire system, including the clock generator)

Each of these modes can be set by executing the HALT or STBY instruction.

13.1 Standby Mode

The following two standby modes can be used.

(1) HALT mode

In this mode, the clock generator (oscillator and PLL synthesizer) operates, but the operating clock of the CPU is stopped. The other internal peripheral functions are supplied with the clock and continue operation. By using this mode in combination with the normal mode, the power consumption of the entire system can be reduced.

(2) STOP mode

In this mode, the clock generator (PLL synthesizer) is stopped and the entire system is stopped. Because the PLL synthesizer and internal peripheral functions are stopped, the power consumption can be reduced more than in the HALT mode.

Because the clock output of the PLL synthesizer is stopped, make sure that sufficient time elapses after the STOP mode is released until the oscillator, CPU clock, and bus clock are stabilized. The PLL circuit may require lock up time depending on the program.

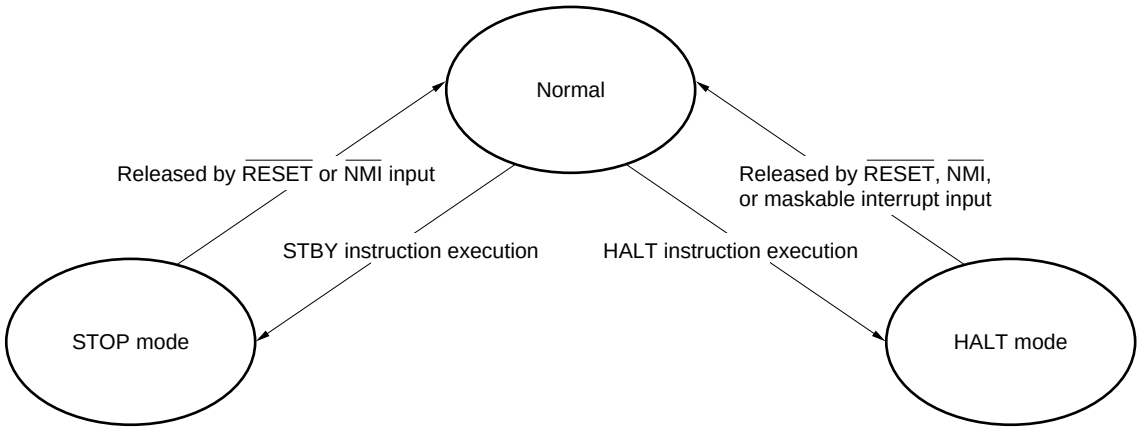
Table 13-1 shows the operations of the clock generator in the HALT and STOP modes. By selecting each mode as the application requires, the power consumption of the system can be efficiently reduced.

Table 13-1. Operation of Clock Generator in Standby Mode

Standby Mode	Oscillator (OSC)	PLL Synthesizer	Clock Supply to Peripheral I/O	Clock Supply to CPU
Normal mode	○	○	○	○
HALT mode	○	○	○	×
STOP mode	○	×	×	×

Remark ○ : Operates
× : Stopped

Figure 13-1. Status Transition



13.2 HALT mode

13.2.1 Setting and operating status of HALT mode

In the HALT mode, the clock generator (oscillator and PLL synthesizer) operates, but the operating clock of the CPU is stopped. The other internal peripheral functions are supplied with the clock and continue operation. When the HALT instruction is executed, the HALT mode is set. If the HALT mode is set while the CPU is not operating, the overall power consumption of the system can be reduced. In the HALT mode, program execution is stopped, but the previous contents of all the registers are retained. In addition, the on-chip peripheral functions independent of the instruction processing of the CPU continue operation. The $\overline{\text{HLDRQ}}$ signal in the HALT mode is accepted.

To set the HALT mode in the non-maskable interrupt service routine, enable acknowledgment of a new non-maskable interrupt request by clearing PSW.NP to 0 (status in which NMI is not serviced) before executing the HALT instruction. If PSW.NP is not 0, the normal operation mode cannot be restored from the HALT mode by $\overline{\text{NMI}}$.

Table 13-2 shows the hardware status in the HALT mode.

Table 13-2. Operating Status in HALT Mode

Function	Operating Status[Note 1]	
Oscillator	Operates	
PLL synthesizer	Operates	
Bus clock	Operates	
CPU	Stops	
Port output	Retained	
Peripheral function	Operates	
Internal data	Internal data such as registers of CPU retain status before HALT mode is set.	
A1 - A23	Undefined	High impedance when $\overline{\text{HLDAK}}$ = 0
D0 - D31	High impedance	
$\overline{\text{BCYST}}$	1	High impedance when $\overline{\text{HLDAK}}$ = 0
$\overline{\text{CS1}}$ - $\overline{\text{CS7}}$		
$\overline{\text{IORD}}$, $\overline{\text{IOWR}}$		
$\overline{\text{MRD}}$, $\overline{\text{WE}}$, $\overline{\text{OE}}$, $\overline{\text{LLMWR}}$, $\overline{\text{LUMWR}}$, $\overline{\text{ULMWR}}$, $\overline{\text{UUMWR}}$		
$\overline{\text{REFRQ}}$, $\overline{\text{LLCAS}}$, $\overline{\text{LUCAS}}$, $\overline{\text{ULCAS}}$, $\overline{\text{UUCAS}}$	1[Note 2]	
RAS	**Note 3**	
$\overline{\text{HLDRQ}}$	Operates	
CLKOUT	Clock output (when clock output is not disabled)	

Notes 1. Each pin is in the operating status during DMA transfer.

2. Other than CBR refresh

3. The previous status is retained before CBR refresh is executed. This pin is set to “1” after CBR refresh.

13.2.2 Releasing HALT mode

The HALT mode can be released by the non-maskable interrupt request, an unmasked maskable interrupt request, and $\overline{\text{RESET}}$ pin input.

(1) Releasing by non-maskable interrupt request

When the $\overline{\text{NMI}}$ signal is detected, supply of the CPU clock is resumed.

(2) Releasing by maskable interrupt request

Supply of the CPU clock is resumed by an unmasked maskable interrupt request.

To set the HALT mode in an interrupt routine, enable the interrupt that releases the HALT mode before executing the HALT instruction. Enable the interrupt (PSW.ID = 0, EP = 0) and set the interrupt enable level.

Table 13-3. Releasing HALT Mode by Interrupt Request

Releasing Source	EI Status (PWS.ID = 0)	DI Status (PSW.ID = 1)
Non-maskable interrupt request	Handler address branch	
Maskable interrupt request	Handler address branch	Not released

(3) Releasing by $\overline{\text{RESET}}$ pin input

★ Same as the normal reset operation. Therefore, the status of the register before the standby mode is set is not retained.

13.3 STOP Mode

13.3.1 Setting and operating status of STOP mode

In the STOP mode, the PLL synthesizer is stopped (the oscillator is not stopped).

The power consumption can be reduced by stopping the PLL synthesizer and internal peripheral circuit. When the STBY instruction is executed, the CBR self refresh cycle is started, and the STOP mode is set.

Make sure that the oscillation stabilization time elapses after the STOP mode has been released.

In the STOP mode, program execution is stopped, but the previous contents of all the registers are retained. In addition, the internal peripheral functions are also stopped.

Because the internal peripheral functions and clock supply by CLKOUT are stopped in the STOP mode, confirm that the internal peripheral functions and external peripheral functions have stopped before setting the STOP mode, and then execute the STBY instruction.

Before executing the STBY instruction, disable DMA transfer by clearing the EN bit of DCHC0 through DCHC3 of DMAC to 0.

To set the STOP mode in the non-maskable interrupt service routine, enable acknowledgment of a new non-maskable interrupt request by clearing PSW.NP to 0 (status in which NMI is not serviced) before executing the STBY instruction. If PSW.NP is not 0, the normal operation mode cannot be restored from the STOP mode by $\overline{\text{NMI}}$.

Table 13-4 shows the hardware status in the STOP mode.

Table 13-4. Operating Status in STOP Mode

Function	Operating Status
Oscillator	Operates
PLL synthesizer	Stops
Bus clock	Stops
CPU	Stops
Port output	Retained
Peripheral function	Stops
Internal data	Internal data such as registers of CPU retain status before STOP mode is set.
A1 - A23	Undefined
D0 - D31	High impedance
$\overline{\text{BCYST}}$	1
$\overline{\text{CS1}}$ - $\overline{\text{CS7}}$	
$\overline{\text{IORD}}$, $\overline{\text{IOWR}}$	
$\overline{\text{MRD}}$, $\overline{\text{WE}}$, $\overline{\text{OE}}$, $\overline{\text{LLMWR}}$, $\overline{\text{LUMWR}}$, $\overline{\text{ULMWR}}$, $\overline{\text{UUMWR}}$	
$\overline{\text{REFRQ}}$, $\overline{\text{RAS}}$, $\overline{\text{LLCAS}}$, $\overline{\text{LUCAS}}$, $\overline{\text{ULCAS}}$, $\overline{\text{UUCAS}}$	CBR self refresh[Note]
$\overline{\text{HLDRQ}}$	Not accepted
CLKOUT	0

Note CBR self refresh is not executed when it is disabled. In this case, the status of this pin before the STOP mode is set is retained.

13.3.2 Releasing STOP mode

The STOP mode is released by the non-maskable interrupt request or $\overline{\text{RESET}}$ pin input. Make sure that the oscillation stabilization time of the oscillator elapses after the STOP mode has been released.

(1) Releasing by non-maskable interrupt request (NMI)

When the $\overline{\text{NMI}}$ signal is detected, the PLL synthesizer resumes operation. After that, it starts supplying the CPU clock and bus clock after the oscillation stabilization time has elapsed.

The interrupt processing started by the $\overline{\text{NMI}}$ signal when the STOP mode is released is treated as equivalent to the normal non-maskable interrupt processing. If the two must be distinguished in a program, prepare a software status in advance, and set the status before executing the STBY instruction. When this status is checked by the non-maskable interrupt processing, the $\overline{\text{NMI}}$ signal releasing the STOP mode can be distinguished from the normal $\overline{\text{NMI}}$ signal.

(2) Releasing by $\overline{\text{RESET}}$ input

★ Same as the normal reset operation. Therefore, the values of the registers before the standby mode is set are not retained.

13.4 Ensuring Oscillation Stabilization Time

After the STOP mode has been released, enough time must elapse until the PLL circuit operation stabilizes.

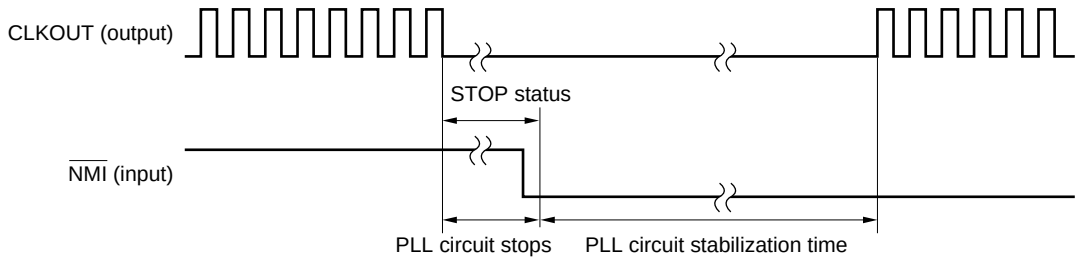
(1) To ensure lapse of time by using oscillation stabilization time ensuring timer ($\overline{\text{NMI}}$ signal input)

The STOP mode is released if a valid edge is input to the $\overline{\text{NMI}}$ pin.

In this case, the oscillation stabilization time required for clock output to be stabilized is ensured by the oscillation stabilization time ensuring timer.

After a specific time, clock output is started, and execution branches to the handler address of the NMI processing.

Figure 13-2. STOP Mode Releasing Timing (with $\overline{\text{NMI}}$ signal input)



PLL circuit stabilization time: 31 ms (resonator or external clock: 33 MHz)
62.8 ms (resonator or external clock: 16.7 MHz)

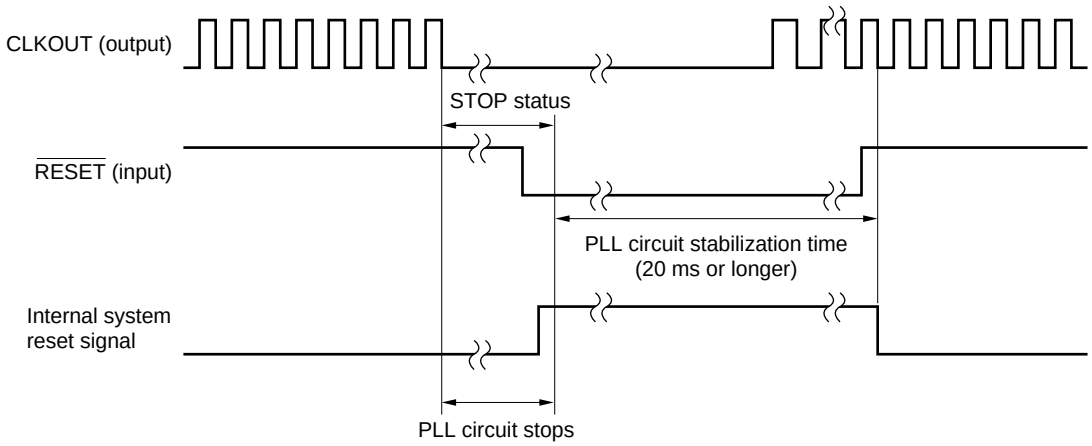
(2) To ensure time lapse by signal level width ($\overline{\text{RESET}}$ signal input)

When a falling edge is input to the $\overline{\text{RESET}}$ pin, the STOP mode is released. Ensure that the time during which clock output from the oscillator is stabilized elapses by using the low-level width input to the $\overline{\text{RESET}}$ pin.

Input the low-level width for 20 ms or longer to stabilize the PLL.

After a rising edge is input to the $\overline{\text{RESET}}$ pin, supply of the clock is started, and execution branches to the handler address that is used for system reset.

Figure 13-3. STOP Mode Releasing Timing (with $\overline{\text{RESET}}$ signal input)



[MEMO]

CHAPTER 14 RESET/NMI CONTROL FUNCTION

The reset/NMI control functions are implemented by the system control unit (SYU). The system control unit is a circuit that controls the $\overline{\text{RESET}}$ and $\overline{\text{NMI}}$ signals.

14.1 Features

- $\overline{\text{RESET}}$ and $\overline{\text{NMI}}$ pins have noise rejection circuit using the external input clock sampling.
- Performs forced reset, reset mask, and NMI mask processing from debug control unit

14.2 Non-Maskable Interrupt (NMI)

The $\overline{\text{NMI}}$ signal is sampled at the rising edge of the external input clock (this clock is not stopped even in the STOP mode and its cycle is the same as that of the bus clock).

Noise of less than 5 external input clocks is rejected and then the valid edge of the $\overline{\text{NMI}}$ signal is detected. Therefore, the $\overline{\text{NMI}}$ signal must be kept low for the duration of 5 external input clocks or longer.

When the $\overline{\text{NMI}}$ signal goes low, the interrupt is detected. Because the $\overline{\text{NMI}}$ signal is detected at the falling edge, it can be deasserted inactive once the non-maskable interrupt request has been detected. The detected interrupt request is retained in the CPU until the CPU starts the interrupt processing.

14.3 Reset

The system is reset and the on-chip hardware units are initialized when a low level is input to the $\overline{\text{RESET}}$ pin.

When the $\overline{\text{RESET}}$ pin goes high, the reset status is cleared, and the CPU starts program execution. Initialize the contents of each register in software as necessary.

The valid edge of the $\overline{\text{RESET}}$ signal is detected after noise width of less than 5 external input clocks has been rejected. Therefore, the $\overline{\text{RESET}}$ signal must be kept low for the duration of 5 external input clocks or longer. To satisfy the minimum width (20 clocks) of the reset enable for the CPU core, the width of the external reset signal must be 25 external input clocks or wider.

★ The minimum number of clocks required for the first $\overline{\text{BCYST}}$ signal to be asserted active after the reset signal has been cleared is 10 bus clocks.

14.3.1 Pin function

Table 14-1 shows the status of the output pins during the system reset period and immediately after reset. This status is retained during the reset period.

If the $\overline{\text{HLDRQ}}$ signal is inactive after the $\overline{\text{RESET}}$ pin is kept low for the duration of 25 external input clocks or longer and then it is deasserted inactive, a memory read cycle is started to fetch instructions.

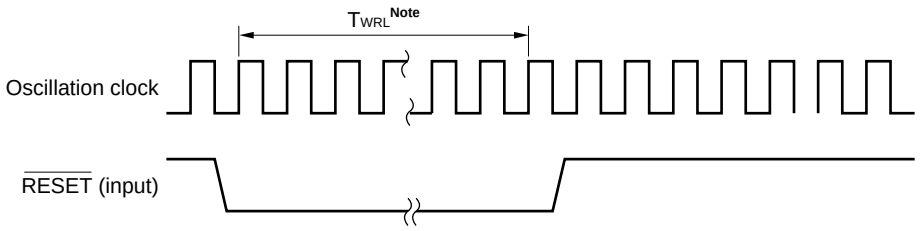
Keep the $\overline{\text{HLDRQ}}$ signal inactive during power-ON reset. Connect a pull-up or pull-down resistor to the pins that go into a high-impedance state at reset. Otherwise, the memory contents may be lost when these pins go into a high-impedance state.

The CLKOUT pin outputs the clock even during the reset period.

Table 14-1. Status of Output Pin Immediately after Reset

Function	Operating Status
A1 - A23	Undefined
D0 - D31	High impedance
$\overline{CS1}$ - $\overline{CS7}$	1
$\overline{BCYST}$	1
$\overline{IORD}$, $\overline{IOWR}$	1
$\overline{WE}$, $\overline{OE}$	1
$\overline{LLMWR}$, $\overline{LUMWR}$, $\overline{ULMWR}$, $\overline{UUMWR}$	1
$\overline{LLCAS}$, $\overline{LUCAS}$, $\overline{ULCAS}$, $\overline{UUCAS}$	1
$\overline{RAS}$	1
CLKOUT	Clock output
$\overline{HLDAK}$	1
DMAAK0 - DMAAK3	1
PORT2/SI	High impedance
PORT1/SO	High impedance
PORT0/$\overline{SCLK}$	High impedance
TXD	1
DDO	Undefined
TRCDATA0 - TRCDATA3	Undefined
$\overline{TC}$/$\overline{REFRQ}$	1
TO10/INTP10, TO11/INTP12	High impedance

Figure 14-1. Accepting Reset Signal



Note The minimum value of T_{WRL} (low-level period of the $\overline{RESET}$ signal) differs depending on the following status.

- On power application or releasing STOP status (STBY instruction): 20 ms (stabilization time of oscillator + PLL oscillation stabilization time)
- Normal (other than above): 25 clocks

14.3.2 Initialize

Table 14-2 shows the value of each register after reset.

Initialize the contents of each register in software as necessary. Care must be exercised in handling the clock control register (CGC) because this register is related to system setting (such as X1 and X2 pin functions and CLKOUT pin operation).

Table 14-2. Initial Value of Each Register after Reset

Register			Initial Value after Reset
System registers	Program counter	PC	FFFFFFF0H
	Exception/interrupt status saving registers	EIPC	Undefined
		EIPSW	Undefined
	NMI/dual exception status saving registers	FEPC	Undefined
		FEPS	Undefined
	Exception cause register	ECR	0000FFF0H
	Program status word	PSW	00008000H
	Processor ID register[Note]	PIR	00008301H
	Task control word	TKCW	000000E0H
	Debug exception status saving registers	DPC	Undefined
		DPSW	Undefined
	Hardware configuration control register	HCCW	00000000H
Internal registers	PLL control register[Note]	PLLCR	00000002H
	Cache memory control register	CMCR	00000000H
	Instruction cache tag register	ICTR	xxxxx000H
	Data cache tag register	DCTR	xxxxx000H
	Instruction RAM register	IRAMR	Undefined

Note These registers are fixed to the initial value with the V831.

[MEMO]

CHAPTER 15 DEBUG/TRACE FUNCTION

The V831 has a debug control unit (DCU) that implements a debug/trace function.

15.1 Features

- Signals for debug: 10 (dedicated: $\overline{\text{DRST}}$, DCK, DMS, DDI, DDO, TRCDATA0 through TRCDATA3)
  (multiplexed: CLKOUT)
  - On-chip debugging can be executed if wiring and connectors for debugging are mounted on the user board (however, the fatal exception handler (FFFFFFE0H through FFFFFFEFH) cannot be used).
  - Debug interface for interfacing the host machine via a debug unit is provided.
  - Trace interface that monitors execution status of user program is provided.
- Debug function
  - Forced reset function (can forcibly reset CPU core and peripheral functions)
  - Forced break function (can forcibly stop user program execution)
  - Can stop user program execution at any address.
  - Can start user program execution from any address.
  - User resources (such as memory and I/O) can be read or written while user program is stopped.
  - User program can be downloaded.
  - Mask function (can mask external input signals ($\overline{\text{RESET}}$, $\overline{\text{HLDRQ}}$, $\overline{\text{NMI}}$, INTP00 through INTP03, and INTP10 through INTP13))
- Trace function
  - PC trace (branch trace)
    Can trace all branches (transition of processing) that take place during user program execution.
    Can select trace sources from the 9 classified branches by function.
  - Data trace
    Can trace all data accesses made via external data bus and internal peripheral I/O bus (internal data RAM, data cache, and system I/O of CPU core cannot be traced)
    Write data can be traced by write access.
  - Real-time trace
  - Forced start/stop of trace. Starts trace from any execution PC.
  - Trace buffer provided (PC trace and data trace multiplexed)
    Can store trace data of 8 to 40 sources

[MEMO]

APPENDIX A REGISTER INDEX

[R]

[S]

[T]

APPENDIX B GENERAL INDEX

[T]

[U]

[V]

[W]

[X]

[MEMO]

NEC

Facsimile Message

Although NEC has taken all possible steps to ensure that the documentation supplied to our customers is complete, bug free and up-to-date, we readily accept that errors may occur. Despite all the care and precautions we've taken, you may encounter problems in the documentation. Please complete this form whenever you'd like to report errors or suggest improvements to us.

From:

Name

Company

Tel. FAX

Address

Thank you for your kind support.

North America
NEC Electronics Inc.
Corporate Communications Dept.
Fax: 1-800-729-9288
1-408-588-6130

Europe
NEC Electronics (Europe) GmbH
Technical Documentation Dept.
Fax: +49-211-6503-274

South America
NEC do Brasil S.A.
Fax: +55-11-6465-6829

Hong Kong, Philippines, Oceania
NEC Electronics Hong Kong Ltd.
Fax: +852-2886-9022/9044

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

Taiwan
NEC Electronics Taiwan Ltd.
Fax: 02-2719-5951

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-250-3583

Japan
NEC Semiconductor Technical Hotline
Fax: 044-548-7900

I would like to report the following error/make the following suggestion:

Document title: _____

Document number: _____ Page number: _____

If possible, please fax the referenced page or drawing.

Document Rating	Excellent	Good	Acceptable	Poor
Clarity	❑	❑	❑	❑
Technical Accuracy	❑	❑	❑	❑
Organization	❑	❑	❑	❑