NEC

Preliminary User's Manual

Instruction Cache, Data Cache

NB85E, NB85ET

NB85E212
NB85E213
NB85E252
NB85E263

Document No. A14247EJ4V1UM00 (4th edition)
Date Published January 2002 NS CP(N)

[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 export of this product from Japan is regulated by the Japanese government. To export this product may be prohibited without governmental license, the need for which must be judged by the customer. The export or re-export of this product from a country other than Japan may also be prohibited without a license from that country. Please call an NEC sales representative.

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

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

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

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

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

M5D 98. 12

Regional Information

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

J01.12

Major Revisions in This Edition

Pages	Description
p.27	Addition of **Caution 2** in **1.4.2 Tag clear function**
p.28	Addition of **Caution 2** in **1.4.2 Tag clear function**
pp.28, 29	Addition of description to **Caution** and addition of **Remark 3** in **1.4.3 Autofill function (way 0 only)**
p.29	Addition of **Caution 2** in **1.5 Instruction Cache Setting Procedure**
p.30	Addition of description to **Caution** in **1.6.2 Operation on instruction cache miss**
p.37	Addition of **1.9 (11) Simultaneous operation of refill read cycle and cache access by specific instruction that performs branch**
p.40	Addition of **Caution** in **Figure 2-1 NB85E and Data Cache Connection Example**
p.54	Addition of description in **2.5 Data Cache Setting Procedure**
p.97	Modification of **Example 1** in **2.10 (6) Other**
p.97	Addition of **2.10 (7) Operation during debugging**

The mark ⋆ shows major revised points.

PREFACE

Target Readers

This manual is intended to give users an understanding the functions of the instruction caches (NB85E212 and NB85E213) and data caches (NB85E252 and NB85E263) for the NB85E and NB85ET CPU cores for CBICs in order to design application systems using these CPU cores.

Purpose

This manual's purpose is to help the user understand the functions of the instruction and data caches.

Organization

This manual is organized as follows.

CHAPTER 1 INSTRUCTION CACHE

This chapter explains the NB85E212 and NB85E213, which are instruction caches.

CHAPTER 2 DATA CACHE

This chapter explains the NB85E252 and NB85E263, which are data caches.

How to Use This Manual

This manual assumes that the reader has general knowledge of electrical engineering, logic circuits, and microcontrollers.

To gain a general understanding of the functions of the instruction and data caches, refer to:

→ This manual in the order of **CONTENTS**

For information about the functions of the NB85E and NB85ET, refer to:

→ **NB85E Hardware User's Manual (A13971E)** and **NB85ET Hardware User's Manual (A14342E)**

In this manual, unless specifically noted, the NB85E is described as the typical CPU core product. When using the NB85ET, read "NB85E" as "NB85ET".

Conventions

Data significance:	Higher digits on the left and lower digits on the right
Active low representation:	xxxZ (Z is appended to the pin or signal name)
Note:	Footnote for item marked with **Note** in the text
Caution:	Information requiring particular attention
Remark:	Supplementary information
Numerical representation:	Binary … xxxx or xxxxB
	Decimal … xxxx
	Hexadecimal … xxxxH
Prefix indicating the power of 2 (address space, memory capacity):	
	K (kilo) … 2^{10} = 1,024
	M (mega) … 2^{20} = $1,024^2$
	G (giga) … 2^{30} = $1,024^3$
Data types:	Word … 32 bits
	Halfword … 16 bits
	Byte … 8 bits

Related Documents

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

- V850E1 Architecture User's Manual (U14559E)
- NB85E Hardware User's Manual (A13971E)
- NB85ET Hardware User's Manual (A14342E)
- CB-9 Family VX/VM Type NB85E, NB85ET Design Manual (A14335E)

The related documents listed above are subject to change without notice. Be sure to use the latest version of each document for designing.

CONTENTS

LIST OF FIGURES (1/2)

LIST OF FIGURES (2/2)

LIST OF TABLES

CHAPTER 1 INSTRUCTION CACHE

1.1 Outline

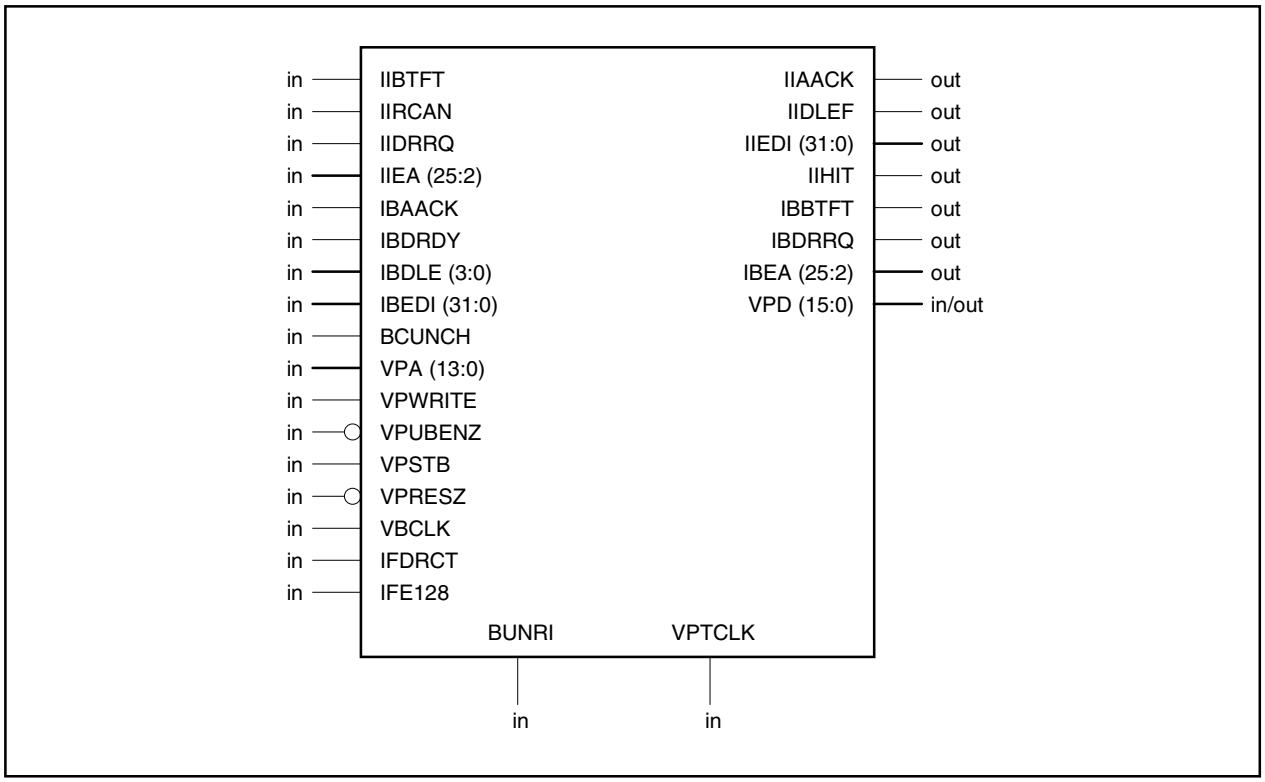
The NB85E212 and NB85E213 are instruction cache memories for the NB85E.
They can be directly connected to the instruction cache interface incorporated in the NB85E.
The following two types of instruction caches are available.

- NB85E212 ... 4 KB 2-way set-associative instruction cache (4 words × 128 entries × 2 ways = 4 KB)
- NB85E213 ... 8 KB 2-way set-associative instruction cache (4 words × 256 entries × 2 ways = 8 KB)

1.1.1 Features

- Use of Least Recently Used (LRU) algorithm
  This algorithm, which makes the block that has remained unaccessed the longest subject to replacement when a miss occurs, is used in the 2-way set-associative type cache. The probability of hitting is high compared to the directly mapped type.
- Using the tag clear function, the contents of all tags can be cleared (invalidated).
- Using the autofill function, instructions for one way can be filled automatically (way 0 only).
  A filled way is locked automatically, and replacing data in the way or writing to tags is disabled. Thus, it can also be used as a ROM that can operate in one cycle.

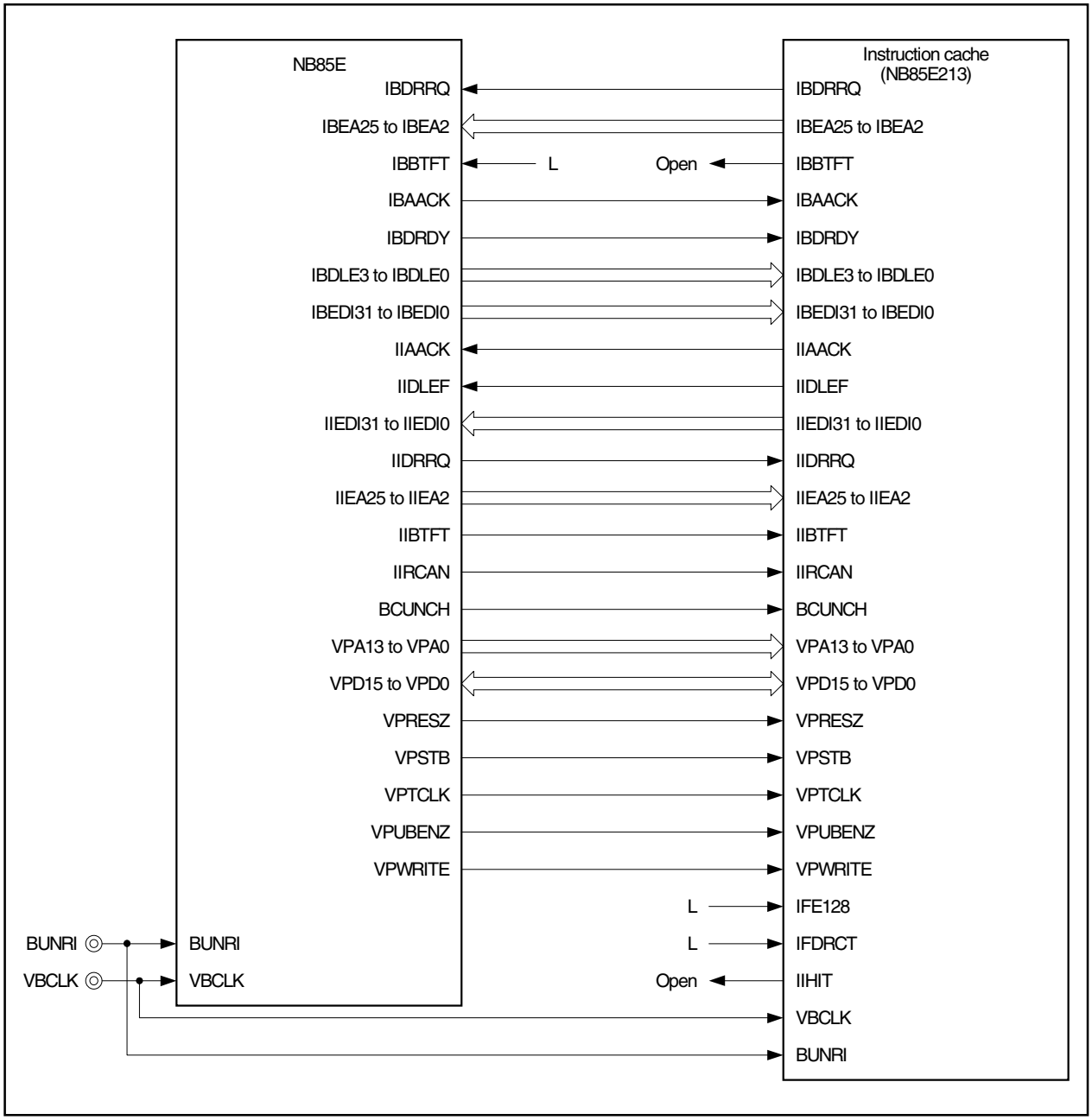
1.1.2 Symbol diagram



1.1.3 NB85E connection example

The following figure shows an example of the connection of an instruction cache to the NB85E.

Figure 1-1. NB85E and Instruction Cache Connection Example



1.2 Pin Functions

1.2.1 List of pin functions

Pin Name		I/O	Function
NB85E connection pins	IIBTFT	Input	Input branch target fetch status from NB85E
	IIRCAN	Input	Input code cancel status from NB85E
	IIDRRQ	Input	Input fetch request from NB85E
	IIEA25 to IIEA2	Input	Input fetch address from NB85E
	IIAACK	Output	Output address acknowledge to NB85E
	IIDLEF	Output	Output data latch enable to NB85E
	IIEDI31 to IIEDI0	Output	Output data to NB85E
	IBAACK	Input	Input address acknowledge from NB85E
	IBDRDY	Input	Input data ready from NB85E
	IBDLE3 to IBDLE0	Input	Input data latch enable from NB85E
	IBEDI31 to IBEDI0	Input	Input data from NB85E
	BCUNCH	Input	Input uncache status from NB85E
	IBEA25 to IBEA2	Output	Output fetch address to NB85E
	IBBTFT	Output	NEC reserved pin (leave open)
	IBDRRQ	Output	Output fetch request to NB85E
	VPA13 to VPA0	Input	Input address (for NPB)
	VPWRITE	Input	Input write access strobe (for NPB)
	VPUBENZ	Input	Input higher byte enable (for NPB)
	VPSTB	Input	Input data strobe (for NPB)
	VPD15 to VPD0	I/O	Input and output data (for NPB)
	VPRESZ	Input	Input reset
	VBCLK	Input	Input internal system clock
Cache type selection pins	IFDRCT	Input	NEC reserved pin (input a low level)
	IFE128	Input	Input entry selection
Status pin	IIHIT	Output	Output tag hit status
Test mode pins	BUNRI	Input	Input normal/test mode selection
	VPTCLK	Input	Input clock for test

1.2.2 Explanation of pin functions

(1) NB85E connection pins

(a) IIBTFT (input)

IIBTFT inputs the branch target fetch status from the NB85E.

A high level is input when fetching the target address on a branch instruction.

(b) IIRCAN (input)

IIRCAN inputs the code cancel status from the NB85E.

This is the signal for canceling the preceding request when data becomes unnecessary due to a branch or interrupt after the NB85E outputs a fetch request to the instruction cache.

(c) IIDRRQ (input)

IIDRRQ inputs a fetch request from the NB85E.

(d) IIEA25 to IIEA2 (input)

IIEA25 to IIEA2 constitute a bus that inputs a fetch address from the NB85E.

The address to be fetched is input from external memory at the same time as the fetch request (IIDRRQ).

(e) IIAACK (output)

IIAACK outputs an address acknowledge to the NB85E.

This signal is output to the NB85E when a fetch address from the NB85E (IIEA25 to IIEA2) is recognized.

(f) IIDLEF (output)

IIDLEF outputs a data latch enable to the NB85E.

(g) IIEDI31 to IIEDI0 (output)

IIEDI31 to IIEDI0 constitute a bus that outputs data to the NB85E.

This bus outputs the data that the NB85E is to read.

(h) IBAACK (input)

IBAACK inputs an address acknowledge from the NB85E.

This signal is input when the NB85E recognizes the IBEA25 to IBEA2 signals output from the instruction cache.

(i) IBDRDY (input)

IBDRDY inputs a data ready from the NB85E.

This is input when the NB85E is finished getting the data it was to read from external memory at the time of a miss, and indicates to the instruction cache that preparations for refill have been made.

(j) IBDLE3 to IBDLE0 (input)

IBDLE3 to IBDLE0 input a data latch enable from the NB85E.

(k) IBEDI31 to IBEDI0 (input)

IBEDI31 to IBEDI0 constitute a bus that inputs data from the NB85E.

Refill data is input from the NB85E when a miss occurs.

(l) BCUNCH (input)

BCUNCH inputs the uncache status from the NB85E.

A low level is input when an area for which the instruction cache setting was set to cacheable by the cache configuration register (BHC) of the NB85E is accessed.

(m) IBEA25 to IBEA2 (output)

IBEA25 to IBEA2 constitute a bus that outputs a fetch address to the NB85E.

This bus outputs the address that the NB85E should read when a miss occurs.

(n) IBBTFT (output)

IBBTFT is reserved for NEC. Leave it open.

Fix the IBBTFT pin of the NB85E to low level.

(o) IBDRRQ (output)

IBDRRQ outputs a fetch request to the NB85E.

This pin outputs a signal requesting that the NB85E perform a fetch from external memory.

(p) VPA13 to VPA0, VPWRITE, VPUBENZ, VPSTB, VPD15 to VPD0 (NPB pins)

Refer to the **NB85E Hardware User's Manual (A13971E)**.

(q) VPRESZ (input)

VPRESZ inputs a reset.

(r) VBCLK (input)

VBCLK inputs the internal system clock.

(2) Cache type selection pins

(a) IFDRCT (input)

IFDRCT is reserved for NEC. Always input a low level.

(b) IFE128 (input)

IFE128 inputs the entry selection.

Entries are as follows depending on the level input to this pin.

- Low level: 256 entries (Fix NB85E213 to low level)
- High level: 128 entries (Fix NB85E212 to high level)

(3) Status pin

(a) IIHIT (output)

IIHIT indicates that the cache was hit.

This pin outputs a high level on a hit. If not using this pin, leave it open.

(4) Test mode pins

(a) BUNRI (input)

BUNRI is an input pin that selects normal or test mode.

(b) VPTCLK (input)

VPTCLK inputs the clock for testing.

1.2.3 Pin status

The following table shows the status in each operating mode of the pins that have output functions.

Table 1-1. Pin Status in Each Operating Mode

Pin Name		Pin Status			
		Reset	STOP Mode	HALT Mode	Test Mode
NB85E connection pins	IIAACK	L	Maintained	Operating	Operating
	IIDLEF	L	Maintained	Operating	Operating
	IIEDI31 to IIEDI0	Undefined	Maintained	Operating	Operating
	IBEA25 to IBEA2	Undefined	Maintained	Operating	Operating
	IBBTFT	L	Maintained	Operating	Operating
	IBDRRQ	L	Maintained	Operating	Operating
	VPD15 to VPD0	Hi-Z	Maintained	Operating	Operating
Status pin	IIHIT	L	Maintained	Operating	Operating

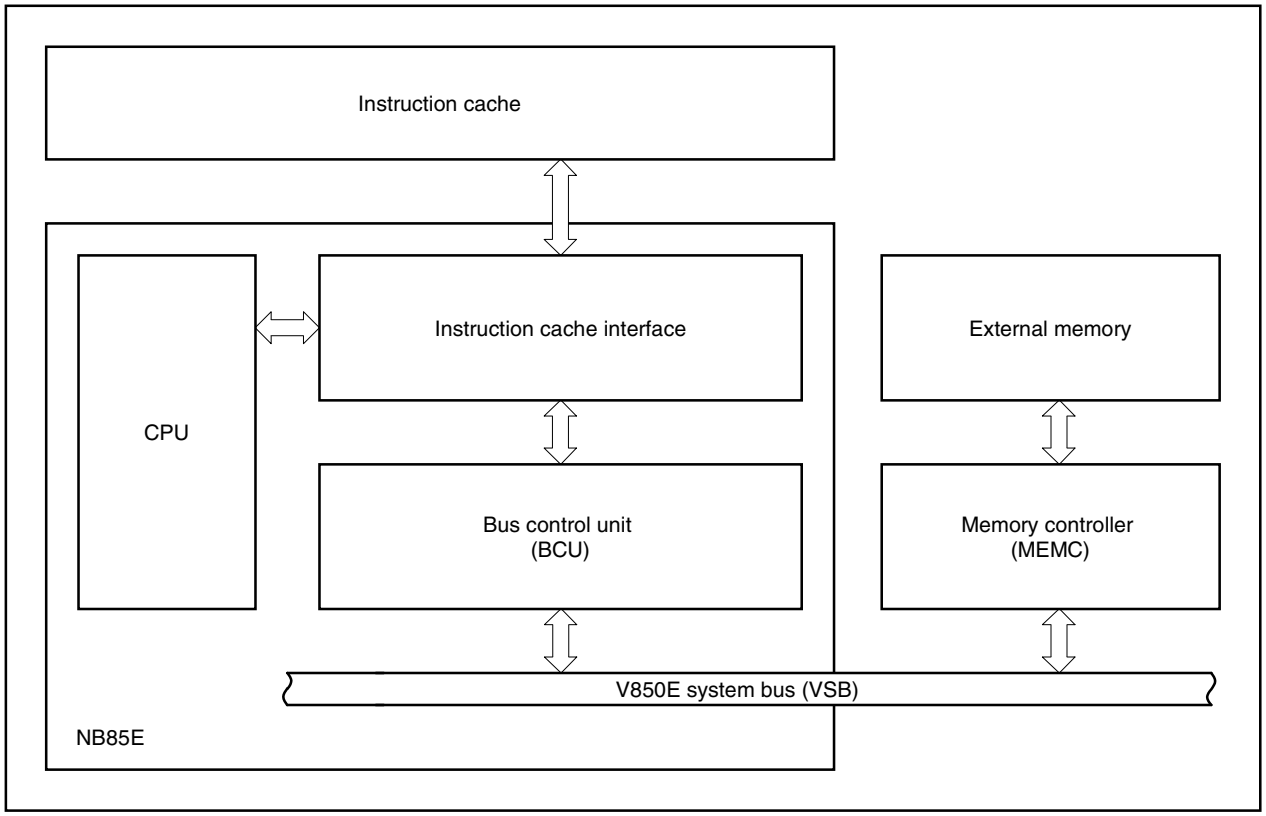
Remark L: Low-level output
Hi-Z: High impedance
Maintained: The previous status is maintained

1.3 Configuration of Instruction Cache

The following two types of instruction caches are available. The NB85E can be accessed by one of these instruction caches in one cycle.

- 4 KB 2-way set-associative instruction cache (NB85E212)
- 8 KB 2-way set-associative instruction cache (NB85E213)

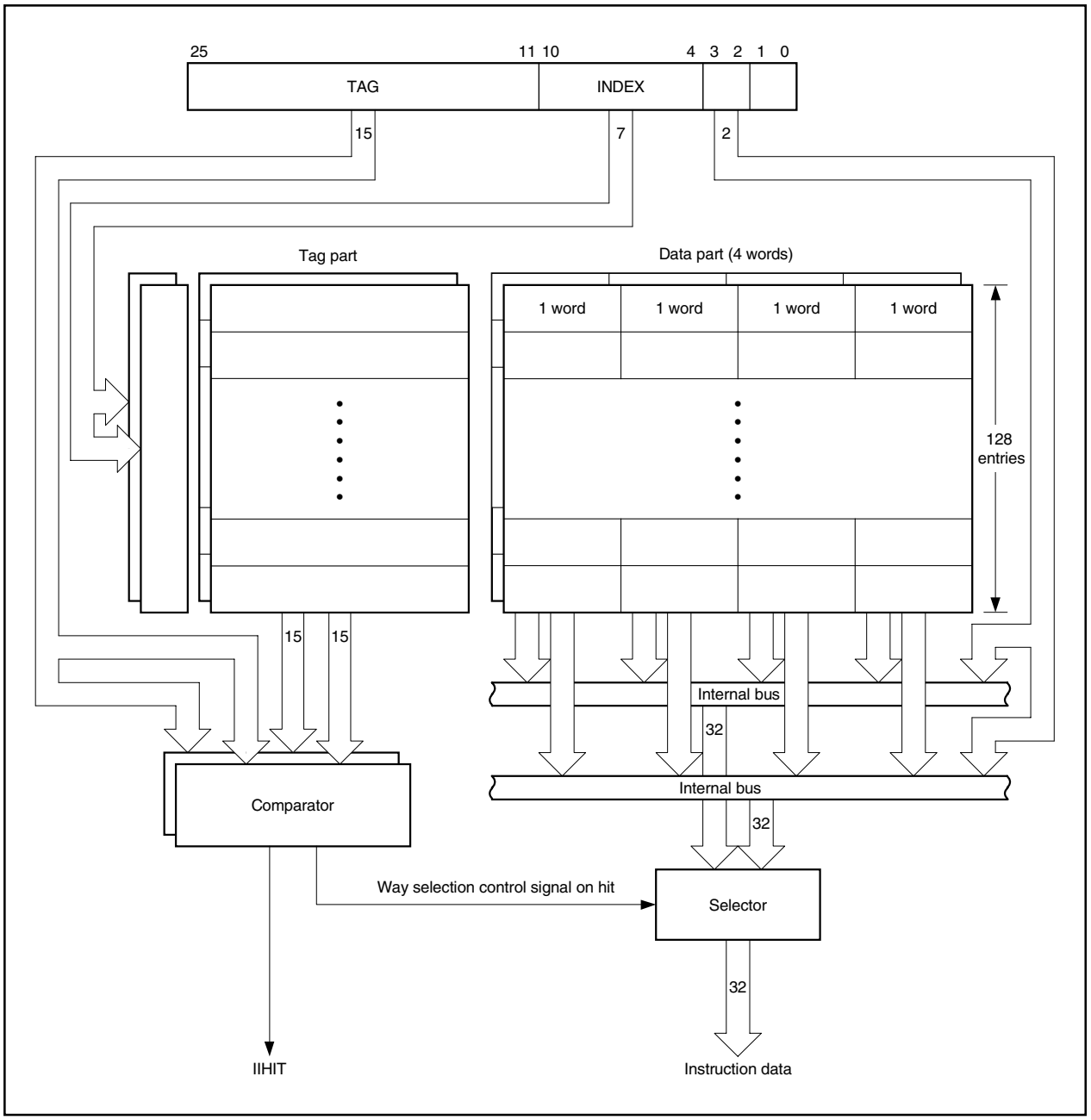
Figure 1-2. Instruction Cache Configuration Example



1.3.1 4 KB 2-way set-associative instruction cache

The data memory of a 4 KB 2-way set-associative instruction cache has two ways, each consisting of a block of 128 entries of 4 words per line, for a total capacity of 4 KB.

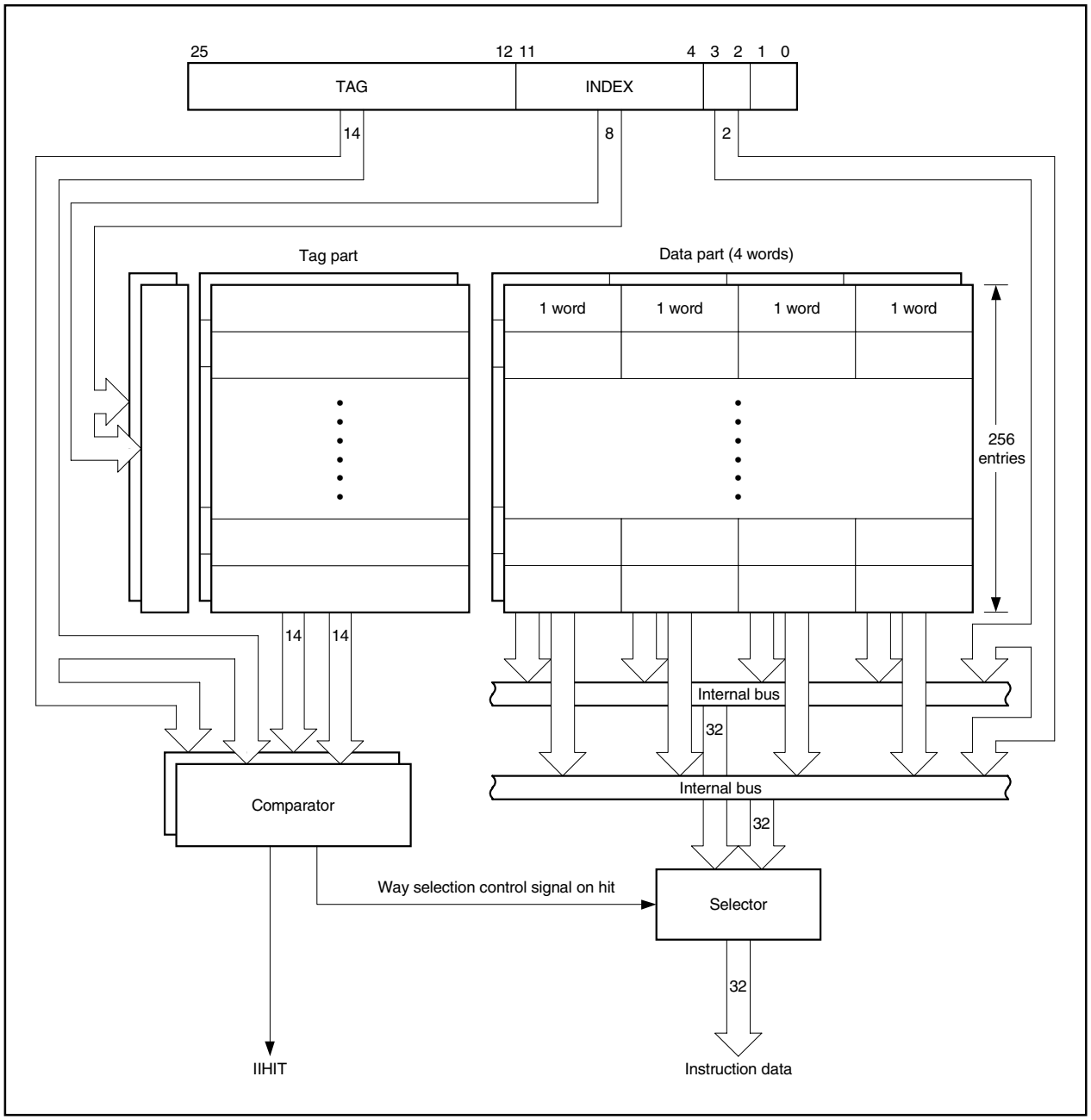
Figure 1-3. Configuration of 4 KB 2-Way Set-Associative Instruction Cache



1.3.2 8 KB 2-way set-associative instruction cache

The data memory of an 8 KB 2-way set-associative instruction cache has two ways, each consisting of a block of 256 entries of 4 words per line, for a total capacity of 8 KB.

Figure 1-4. Configuration of 8 KB 2-Way Set-Associative Instruction Cache



1.4 Instruction Cache Control Functions

1.4.1 Control registers

The following are the instruction cache control functions.

- Tag clear function
- Autofill function (way 0 only)

These functions are controlled by the following registers.

Address	Register Name	Symbol	R/W	Manipulatable Bits			Initial Value
				1 Bit	8 Bits	16 Bits	
FFFFF070H	Instruction cache control register	ICC	R/W	–	–	√	0003H[Note 1]
FFFFF070H	Instruction cache control register L	ICCL	R/W	√	√	–	03H[Note 2]
FFFFF071H	Instruction cache control register H	ICCH	R/W	√	√	–	00H
FFFFF074H	Instruction cache data configuration register	ICD	R/W	–	–	√	Undefined

Notes 1. While reset is active, the value of this register becomes 0003H, and tag initialization begins automatically. Upon completion of tag initialization, the value changes to 0000H.

2. While reset is active, the value of this register becomes 03H, and tag initialization begins automatically. Upon completion of tag initialization, the value changes to 00H.

Remark The ICC register and ICD register are allocated in the peripheral I/O area of the NB85E.

(1) Instruction cache control register (ICC)

The ICC register sets two types of functions: tag clear and autofill.

The ICC register can be read or written in 16-bit units.

This register can be read or written in 8- or 1-bit units when the higher 8 bits of the ICC register are used as the ICCH register and the lower 8 bits are used as the ICCL register.

Cautions 1. If any of bits 0, 1, or 4 is set (1), do not forcibly clear (0) that bit.

2. Do not set (1) bit 4 at the same time as the other bits.

3. Do not set (1) bit 12. Bit 12 can only be cleared (0).

4. Make ICC register settings using an uncacheable area (except for setting bit 4).

Figure 1-5. Instruction Cache Control Register (ICC)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
ICC	0	0	0	LOCK0	0	0	0	0	0	0	0	FILL0	0	0	TCLR1	TCLR0	FFFFF070H	0003H[Note]

Bit position	Bit name	Description
12	LOCK0	This bit shows the cache lock status of way 0. When way 0 is filled, the cache is locked and this bit is set (1) automatically. Clearing (0) this bit releases the cache lock of way 0. 0: Way 0 is not locked 1: Way 0 is locked
4	FILL0	This bit sets way 0 autofill. Setting (1) this bit autofills way 0. When autofill is complete, this bit is cleared (0) automatically. 0: Way 0 fill complete 1: Way 0 fill operating
1	TCLR1	This bit sets way 1 tag clear. Setting (1) this bit clears (invalidates) way 1 tags. When tag clear is complete, this bit is cleared (0) automatically. 0: Way 1 tag clear complete 1: Way 1 tag clear operating
0	TCLR0	This bit sets way 0 tag clear. Setting (1) this bit clears (invalidates) way 0 tags. When tag clear is complete, this bit is cleared (0) automatically. 0: Way 0 tag clear complete 1: Way 0 tag clear operating

Note While reset is active, the value of this register becomes 0003H, and tag initialization begins automatically. Upon completion of tag initialization, the value changes to 0000H.

(2) Instruction cache data configuration register (ICD)

The ICD register sets the address of the memory area to be autofilled when using the autofill function.

The ICD register can be read or written in 16-bit units.

Cautions 1. Do not overwrite the ICD register while autofill is operating.

2. Since the initial value of the ICD register is undefined, when using the autofill function, be sure to set a value in the ICD register prior to setting (1) the FILL0 bit of the ICC register. If the FILL0 bit of the ICC register is set (1) without setting a value in the ICD register, the operation cannot be guaranteed.

Figure 1-6. Instruction Cache Data Configuration Register (ICD)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
ICD	0	DATA14	DATA13	DATA12	DATA11	DATA10	DATA9	DATA8	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0	Address FFFFF074H	Initial value Undefined

Bit position	Bit name	Description
14 to 1	DATA14 to DATA1	These bits set the higher 14 bits of the tag information (bits 25 to 12 of the start address of the memory area to be autofilled).
0	DATA0	NB85E212: This bit sets the LSB of the tag information (bit 11 of the start address of the memory area to be autofilled). NB85E213: Be sure to set this bit to 0 (operation when 1 is set is not guaranteed).

1.4.2 Tag clear function

The tag clear function clears (invalidates) the tags of one way.

In addition, it automatically clears (invalidates) the tags of all ways on a reset.

Use the following procedure to perform instruction cache tag clear.

<1> Read the instruction cache control register (ICC) and confirm that bits 0 and 1 (TCLR0, TCLR1) are all cleared (0).

<2> Read the ICC register and confirm that bit 12 (LOCK0) is cleared (0). Bit 13 of the ICC register is always cleared (0).

<3> Set the TCLR0 and TCLR1 bits of the ICC register as follows.

Cautions 1. To clear tags, the TCLR0 bit or TCLR1 bit of the ICC register must be set (1) twice.

★ **2. Perform all of <1> to <3> above (tag clear) using an uncacheable area (tags are not cleared if the above processing is performed in a cacheable area).**

- **When clearing way 0 and way 1 at the same time:**
  - **(a) Set (1) the TCLR0 and TCLR1 bits.**
  - **(b) Read the TCLR0 and TCLR1 bits to confirm that these bits are cleared (0).**
  - **(c) Perform (a) and (b) above again.**
- **When clearing way 0 and way 1 individually[Note]:**
  - **(a) Set (1) the TCLR0 bit.**
  - **(b) Read the TCLR0 bit to confirm that this bit is cleared (0).**
  - **(c) Perform (a) and (b) above again.**
  - **(d) Set (1) the TCLR1 bit.**
  - **(e) Read the TCLR1 bit to confirm that this bit is cleared (0).**
  - **(f) Perform (d) and (e) above again.**

Note The setting can also be made in order of (d)-(e)-(f)-(a)-(b)-(c).

Cautions 1. Way 0 shares the counter to clear tags with way 1.

Thus, clear tags (set (1) the TCLR0 bit or TCLR1 bit of the ICC register) when the counter for tag clearing is stopped (TCLR0 = TCLR1 = 0). When clearing the tags of way 0 and way 1 individually, if tag clearing for either way is executed during tag clear execution for the other way (TCLR0 or TCLR1 = 1), the counter stops in the middle of tag clearing. Consequently, normal tag clearing cannot be performed because the counter switches to perform the other tag clear operation still indicating the value it had when stopped halfway. Be sure to confirm that tag clearing for one way is completed (TCLR0 or TCLR1 = 0) before performing tag clearing for the other way.

When setting both bits at the same time as shown below, there is no problem.

```
                mov     0x3, r2
LOP0:
                ld.h    ICC[r0], r1
                cmp     r0, r1
                bnz     LOP0
                st.h    r2, ICC[r0]
LOP1:                                   – First TAG clear
                ld.h    ICC[r0], r1
                cmp     r0, r1
                bnz     LOP1
                st.h    r2, ICC[r0]
LOP2:                                   – Second TAG clear
                ld.h    ICC[r0], r1
                cmp     r0, r1
                bnz     LOP2
```

★ **Cautions 2. Be sure not to perform other processing simultaneously with tag clearing before reading the TCLR0 and TCLR1 bits of the ICC register and confirming that these bits are cleared (0).**

Remark The clock count required for a tag clear operation is shown below (the parenthesized values are the clock count required for one tag clear operation. To actually clear tags, the required clock count is doubled because a tag clear operation is performed twice sequentially).

- NB85E212: 256 clocks (128 clocks)
- NB85E213: 512 clocks (256 clocks)

1.4.3 Autofill function (way 0 only)

The autofill function automatically fills instructions for one way.

Once autofilled, a way is automatically locked and write disabled and it operates the same as ROM that is accessible in one cycle. When the lock is released, it again operates as an instruction cache.

Use the following procedure to perform instruction cache autofill.

<1> Clear (invalidate) the tags of way 0 (see **1.4.2 Tag clear function**).
<2> Set the tag information corresponding to the memory area to be autofilled in the instruction cache data configuration register (ICD).
<3> Branch to the cacheable area corresponding to the tag information set in the ICD register.
<4> Set (1) bit 4 (FILL0) of the instruction cache control register (ICC).
<5> When autofill is complete, bit 12 (LOCK0) of the ICC register is automatically set (1) and the way 0 is locked. At that same time, read bit FILL0 of the ICC register and confirm that that bit is cleared (0).

Caution Perform the above operations in the areas shown below.

<1>, <2>, <3> ..... Uncacheable area
<4> ..... Cacheable area
★ **If bit 4 (FILL0) of the ICC register is set (1) using an uncacheable area, autofill cannot be performed (invalid operation).**
<5> ..... Either a cacheable area or an uncacheable area is fine

Remarks 1. A lock is released by clearing (0) bit LOCK0 of the ICC register.

2. The number of clocks required for a 1-way autofill is as follows (when the VSB wait count is 0, VSB data bus size is 32-bit, and 4-byte (1 word) data is filled in 1 clock).

- NB85E212: 512 clocks
- NB85E213: 1024 clocks

★ **3.** Since autofill is performed from the external memory to the instruction cache via the VSB, other processing can be performed at the same time, but only if the processing involves operations within the CPU (processing without any VSB and NPB accesses).

1.5 Instruction Cache Setting Procedure

The instruction cache settings are performed using the following procedure with the initial settings of the user program immediately following system reset.

<1> Wait until the ICC register value becomes “0000H” (that is, tag initialization is completed).
<2> Execute st.h r0, 0xfffff072[r0].
<3> Set the ICC and ICD registers.
<4> Make the instruction cache setting of the BHC register of the NB85E “cacheable”.

Cautions 1. Always input a low level to the IFIUNCH0 pin of the NB85E (instruction cache enable). If a low level is not input to the IFIUNCH0 pin, the “cacheable” setting in the BHC register is invalid even if made.

★ **2. Be sure to make the BHC register settings using an uncacheable area (an instruction is not correctly fetched if settings are made using a cacheable area).**

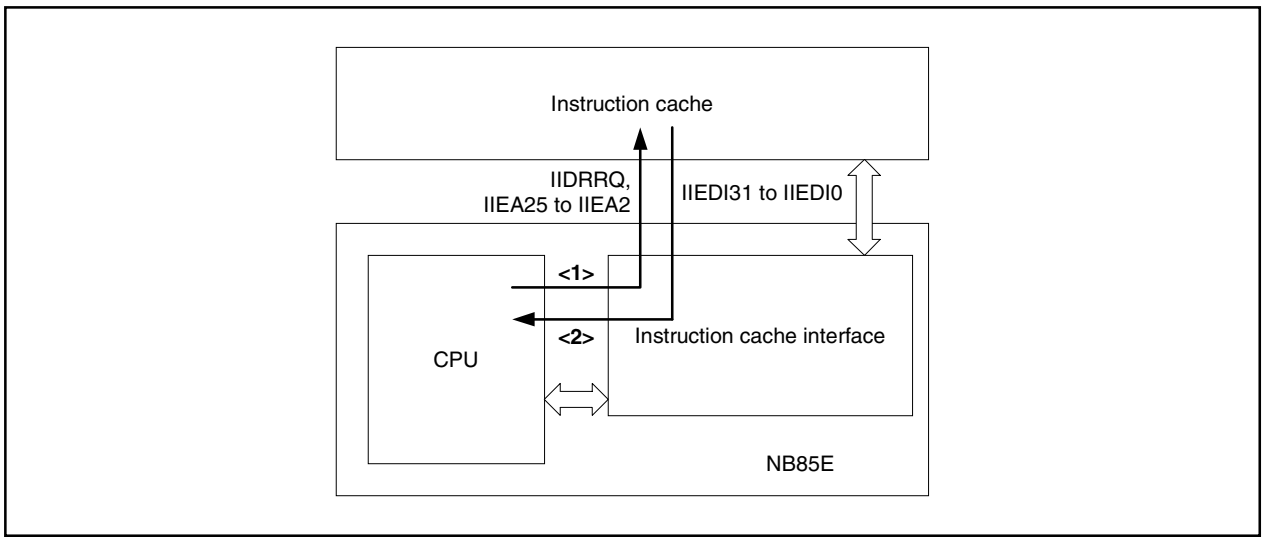
1.6 Operation

The instruction cache automatically performs a caching operation whenever there is a fetch access to a cacheable area set using the cache configuration register (BHC) of the NB85E.

1.6.1 Operation on instruction cache hit

<1> On a fetch access from external memory, output the fetch request (IIDRRQ) and address (IIEA25 to IIEA2) to the instruction cache.
<2> If a hit occurs due to the address existing in the instruction cache, read the data by passing through IIEDI31 to IIEDI0 from the instruction cache.

Figure 1-7. Operation on Instruction Cache Hit



1.6.2 Operation on instruction cache miss

<1> On a fetch access from external memory, output the fetch request (IIDRRQ) and address (IIEA25 to IIEA2) to the instruction cache.

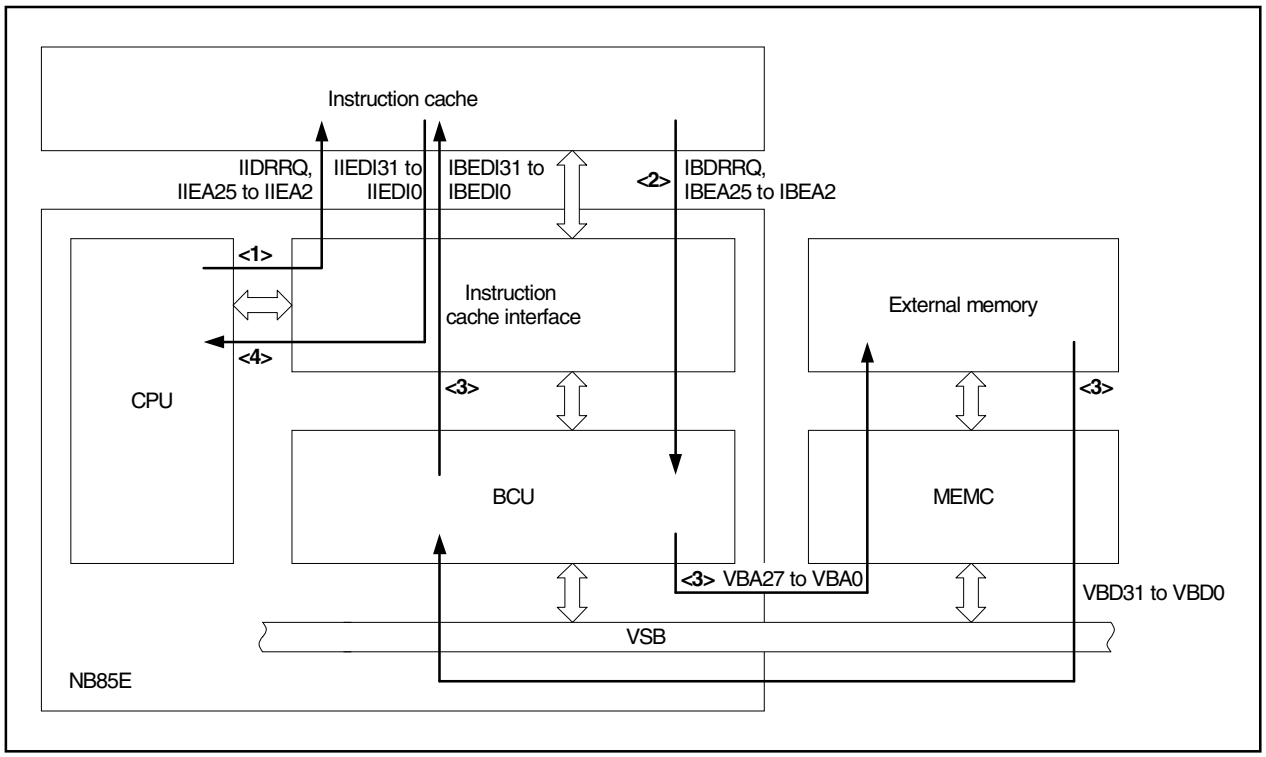
<2> If a miss occurs due to the address not existing in the instruction cache, output a fetch request (IBDRRQ) and the address to be read (IBEA25 to IBEA2) from the instruction cache to the BCU.

<3> The BCU of the internal NB85E outputs the address (VBA27 to VBA0) to external memory via the VSB and refills the instruction cache with one line (4 words) at the address to be read.

<4> The instruction cache then transfers the required data among the 4 words of refill data to the CPU by passing through IIEDI31 to IIEDI0.

★ **Caution The miss penalty time when a miss occurs is the time required to refill 4-word data, and it varies depending on such things as memory controller (MEMC) specifications for external memory, memory type, bus width, and VSB bus cycle wait insertion time.**

Figure 1-8. Operation on Instruction Cache Miss



1.7 Bus Cycle Issued by Instruction Cache

The instruction cache issues a 4-word burst read (4R) sequential refill read cycle.

Figure 1-9 shows timing examples in the case of a 32-bit data bus and a 16-bit data bus.

The bus cycle indicated in Figure 1-9 (a) 32-bit data bus is 4 times greater when an 8-bit data bus is used as a result of bus sizing.

Remarks 1. The timing example is when no waits are used.

2. All signals in the timing example are NB85E signals.

3. The broken-line levels of the VBTTYP1, VBTTYP0, VBD31 to VBD0, VBWAIT, VBAHLD, and VBLAST signals indicate the undefined state (weak unknown) driven by the bus holder in the NB85E.

4. The circles indicate the sampling timing.

5. For details of the VSB signals (VBxxx, VDxxx), refer to the **NB85E Hardware User's Manual (A13971E)**.

Figure 1-9. Sequential Refill Read Cycle (4R) (1/2)

(a) 32-bit data bus

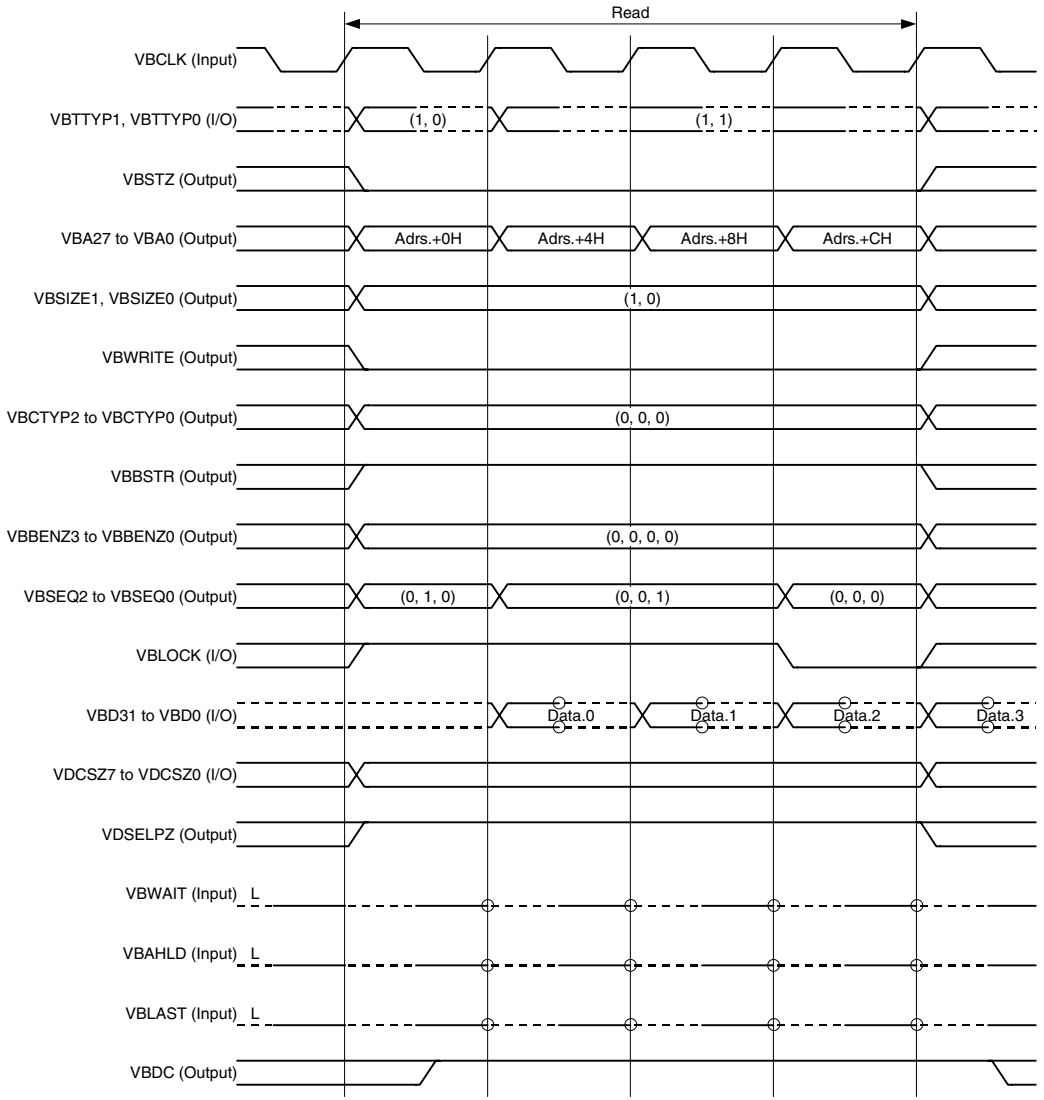
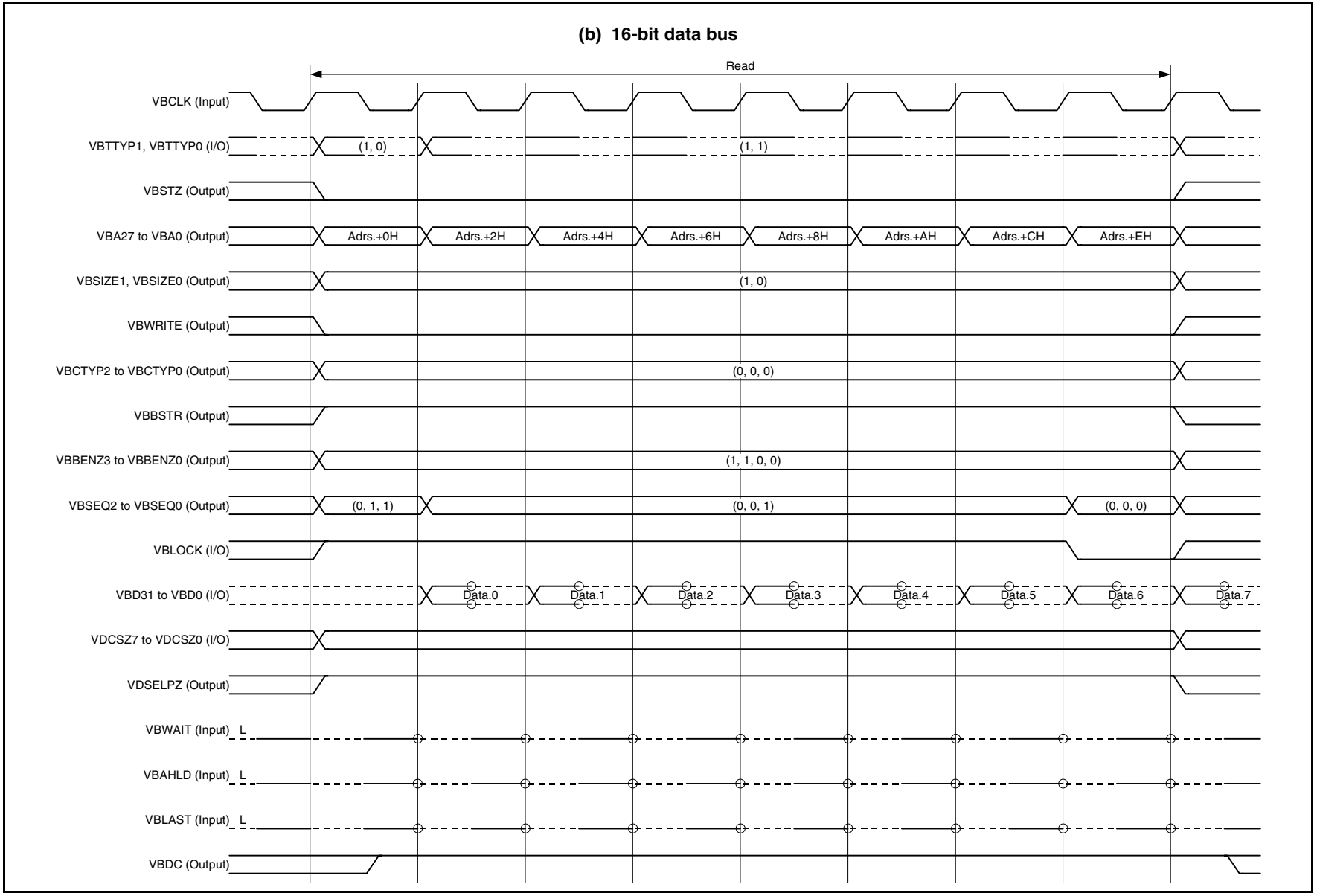


Figure 1-9. Sequential Refill Read Cycle (4R) (2/2)

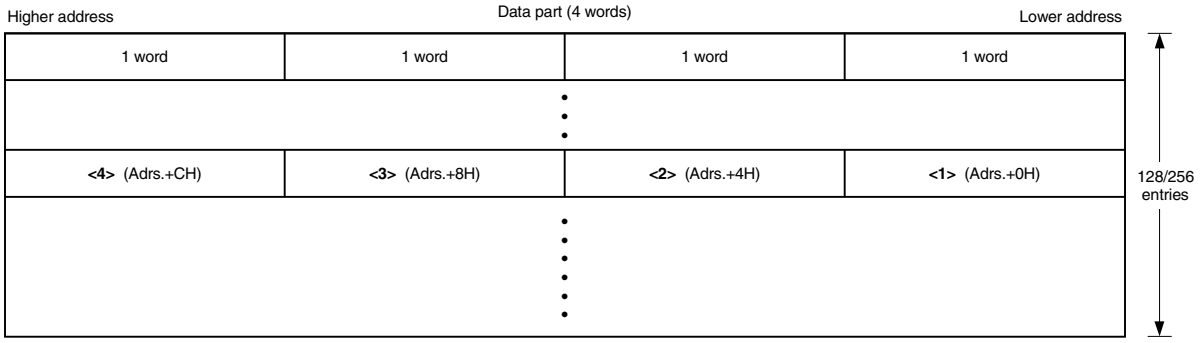


1.8 Refill Sequence to Instruction Cache

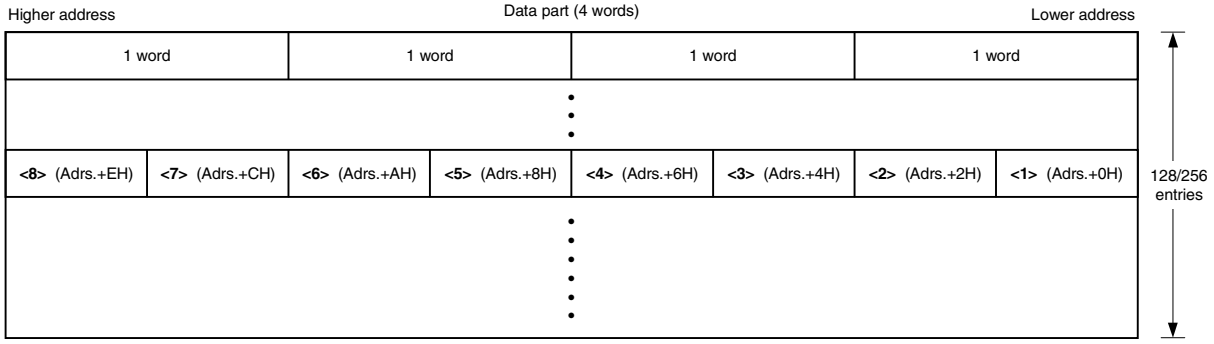
Figure 1-10 shows the refill sequence to the data part of an instruction cache when a miss occurs.

Figure 1-10. Refill Sequence to Instruction Cache

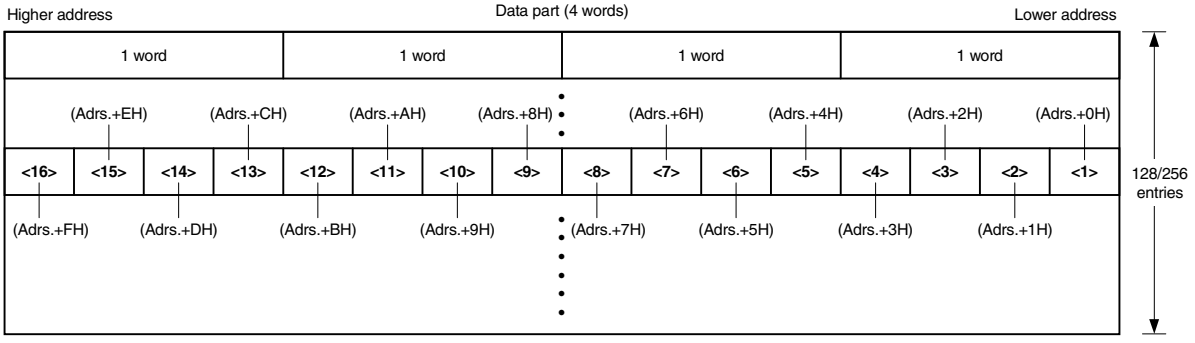
(a) 32-bit data bus



(b) 16-bit data bus



(c) 8-bit data bus



Remarks 1. The numbers within pointed brackets (< >) indicate the refill sequence.

2. (Adrs.+n): Data of address in ( ) (n = 0H to FH)

1.9 Cautions

(1) Connection to NB85E

Connect pins that have the same pin names. However, leave the IBBTFT pin of the instruction cache open and fix the IBBTFT pin of the NB85E to low level.

(2) Setting cache type selection pins

Input the levels shown below to cache type selection pins beginning with IF.

Pin Name	Input Level	
	NB85E212	NB85E213
IFE128	High level	Low level
IFDRCT	Low level	Low level

(3) Bus cycle status

For an area for which the instruction cache setting was set to cacheable by the cache configuration register (BHC) of the NB85E, the VBCTYP2 to VBCTYP0 signals of the NB85E always indicate a normal opcode fetch and do not indicate an opcode fetch of the destination address for a branch instruction.

(4) Operation on reset

At the time of a reset, tags are automatically cleared (invalidated), which puts the next data replacement in a state of being performed from way 0. Therefore, if there is an access to the instruction cache within a period of as many clock cycles as the number of lines after a reset, the CPU stops until the tags are cleared (become invalid).

(5) Setting registers

Be sure to set the NB85E registers shown below using an uncacheable area. However, set bit 4 of the instruction cache control register (ICC) using a cacheable area.

- Chip area select control registers (CSC0, CSC1)
- Peripheral I/O area select control register (BPC)
- Bus size configuration register (BSC)
- Endian configuration register (BEC)
- Cache configuration register (BHC)
- Instruction cache control register (ICC)[Note]
- Instruction cache data configuration register (ICD)

Note Excluding bit 4

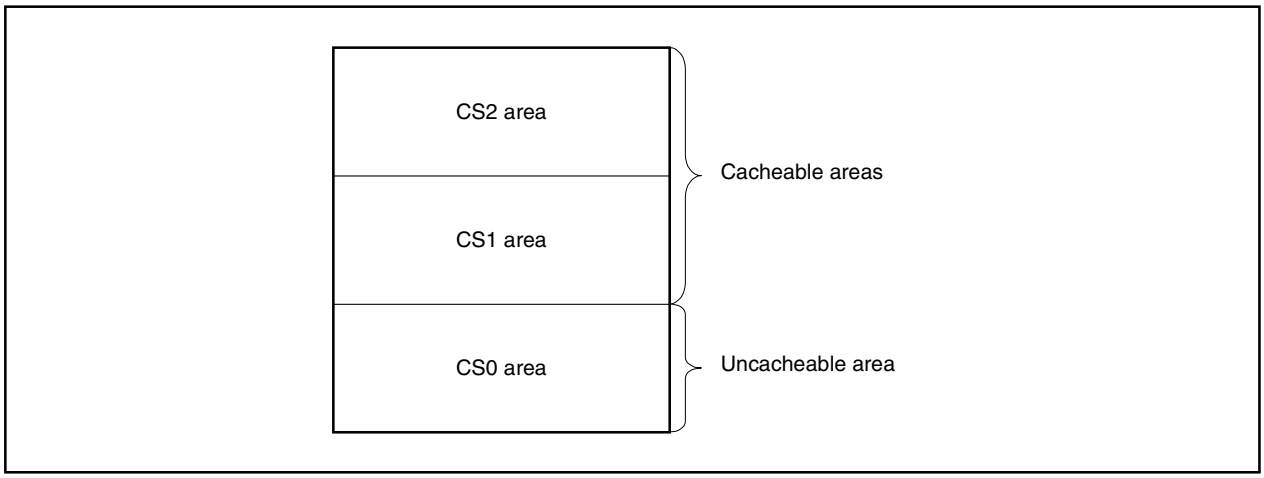
(6) Access to memory boundary

If adjacent chip select (CSn) areas are a cacheable area and an uncacheable area, continuous access across the memory boundary is possible only by using a branch instruction (n = 7 to 0). Operation is not guaranteed if the memory boundary is continuously accessed by an instruction other than a branch instruction. An example is shown below.

Example Suppose that cache area settings are as shown in Figure 1-11. In this case, access to the memory areas is as follows.

- From CS0 area to CS1 area, access is possible only by using a branch instruction.
- From CS1 area to CS2 area, continuous access is possible.

Figure 1-11. Cache Area Setting Example



(7) Initial program settings

Always execute the following instruction before setting the BHC register of the NB85E with the initial settings of the user program immediately following system reset.

```
st.h   r0, 0xfffff072[r0]
```

Following execution of this instruction, the cache is enabled by setting “cache enable” (BHn0 bit = 1) as the instruction cache setting with the BHC register (n = 7 to 0).

(8) Setting BHC register of NB85E

In the case of CSn areas for which an instruction to set the BHC register exists, cache enable/disable settings for the instruction cache using this instruction cannot be performed (n = 7 to 0). Instruction cache enable/disable settings are possible only for CSn areas for which no instruction for setting the BHC register exists.

For example, if a BHC register setting instruction exists in the CS0 area, the instruction cache of the CS0 area cannot be set (cache enable/disable settings). In this case, only the instruction cache settings for areas CS1 to CS7 are possible.

However, instruction cache settings for all CSn areas from instructions that exist in memory areas connected to VFB or VDB are possible.

Remark VFB: Dedicated bus used to directly connect ROM (V850E fetch bus)
VDB: Dedicated bus used to directly connect RAM (V850E data bus)

(9) Test bus auto wiring tool support

This instruction cache does not support test bus auto wiring tools because although it has a BUNRI pin, it does not have test buses (TBOx, TBIx).

(10) Tag clear procedure

Way 0 shares the counter to clear tags with way 1.

Thus, clear tags (set (1) the TCLR0 bit or TCLR1 bit of the ICC register) when the counter for tag clearing is stopped (TCLR0 = TCLR1 = 0). When clearing the tags of way 0 and way 1 individually, if tag clearing for either way is executed during tag clear execution for the other way (TCLR0 or TCLR1 = 1), the counter stops in the middle of tag clearing. Consequently, normal tag clearing cannot be performed because the counter switches to perform the other tag clear operation still indicating the value it had when stopped halfway. Be sure to confirm that tag clearing for one way is completed (TCLR0 or TCLR1 = 0) before performing tag clearing for the other way.

★ **(11) Simultaneous operation of refill read cycle and cache access by specific instruction that performs branch**

In the NB85E instruction cache, an instruction that is read in the bus cycle in which the refill read cycle is started may be discarded without being registered in the instruction cache (even if this operation occurs, program execution itself is performed normally and the execution result is correct).

This operation may occur when all the following conditions (a) to (c) are satisfied (even if all the following conditions are satisfied, however, this operation may not always occur since this operation occurs only when multiple conditions such as the internal status of the instruction cache or instruction execution timing are satisfied at the same time).

(a) When the instruction of a cache line (16 bytes) in the instruction cache is being executed by the CPU

(b) When the cache lines of the address following the above cache line do not exist in the instruction cache, and a miss occurs due to accessing the following cache lines by an instruction prefetch of the CPU

(c) When a specific instruction[Note] in the cache line that performs branch operation is executed by the CPU, and access to the branch destination is requested for the instruction cache at the same time as the above miss occurrence, and then the branch destination generates a hit in the instruction cache

Note The target instructions are as follows.

Bcond, CALLT, CTRET, DBRET, DBTRAP, JARL, JMP, JR, RETI, SWITCH, TRAP, DISPOSE imm5, list12[reg1] (instruction with branch to [reg1])

When this operation occurs, the refill read cycle is started due to a miss occurrence in the following cache lines. The instructions that are read in that cycle are discarded without being registered in the instruction cache.

This may lower the performance of program execution. For example, if this operation occurs due to the conditional branch instruction of the program loop block, the performance is lowered because the invalid refill read cycle of the following cache line occurs on every loop. Especially, if the loop is small, the performance deterioration by the invalid bus cycle is larger.

To avoid occurrence of this operation, allocate the branch instruction to the 6-byte area at the start of the cache line (16-byte boundary). This prevents the occurrence conditions ((a) to (c)) being satisfied, and so this operation does not occur.

CHAPTER 2 DATA CACHE

2.1 Outline

The NB85E252 and NB85E263 are data cache memories for the NB85E.
They can be directly connected to the data cache interface incorporated in the NB85E.
The following two types of data cache are available.

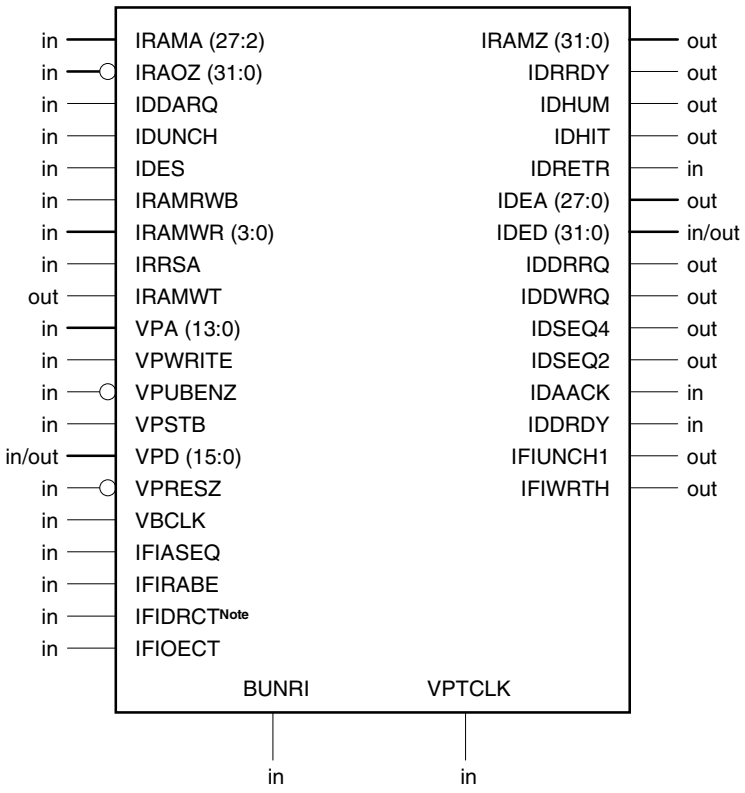
- NB85E252 ... 4 KB directly mapped data cache (4 words × 256 entries = 4 KB)
- NB85E263 ... 8 KB 2-way set-associative data cache (4 words × 256 entries × 2 ways = 8 KB)

2.1.1 Features

- Use of Least Recently Used (LRU) algorithm
  This algorithm, which makes the block that has remained unaccessed the longest subject to replacement when a miss occurs, is used in the 2-way set-associative type cache. The probability of hitting is high compared to the directly mapped type.
- Using the tag clear function, the contents of all tags can be cleared (invalidated).
- Using the tag fill function, the contents of all tags can be filled with addresses of memory to be filled. By locking a filled way, it also can be used as data RAM. However, the DMA operation of the NB85E cannot be performed by using a locked data RAM.
- Using the lock function, any way can be locked.
  Writing to a tag of a locked way is disabled.
- Using the data flush function, dirty data lines can be flushed in writeback mode.

Remark Dirty data is data in the cache memory that must be written back to the main memory if data of the same address in the cache memory and main memory is different. In contrast, if data of the same address in the cache memory and main memory is the same, the data in the cache memory is called clean data.

2.1.2 Symbol diagram

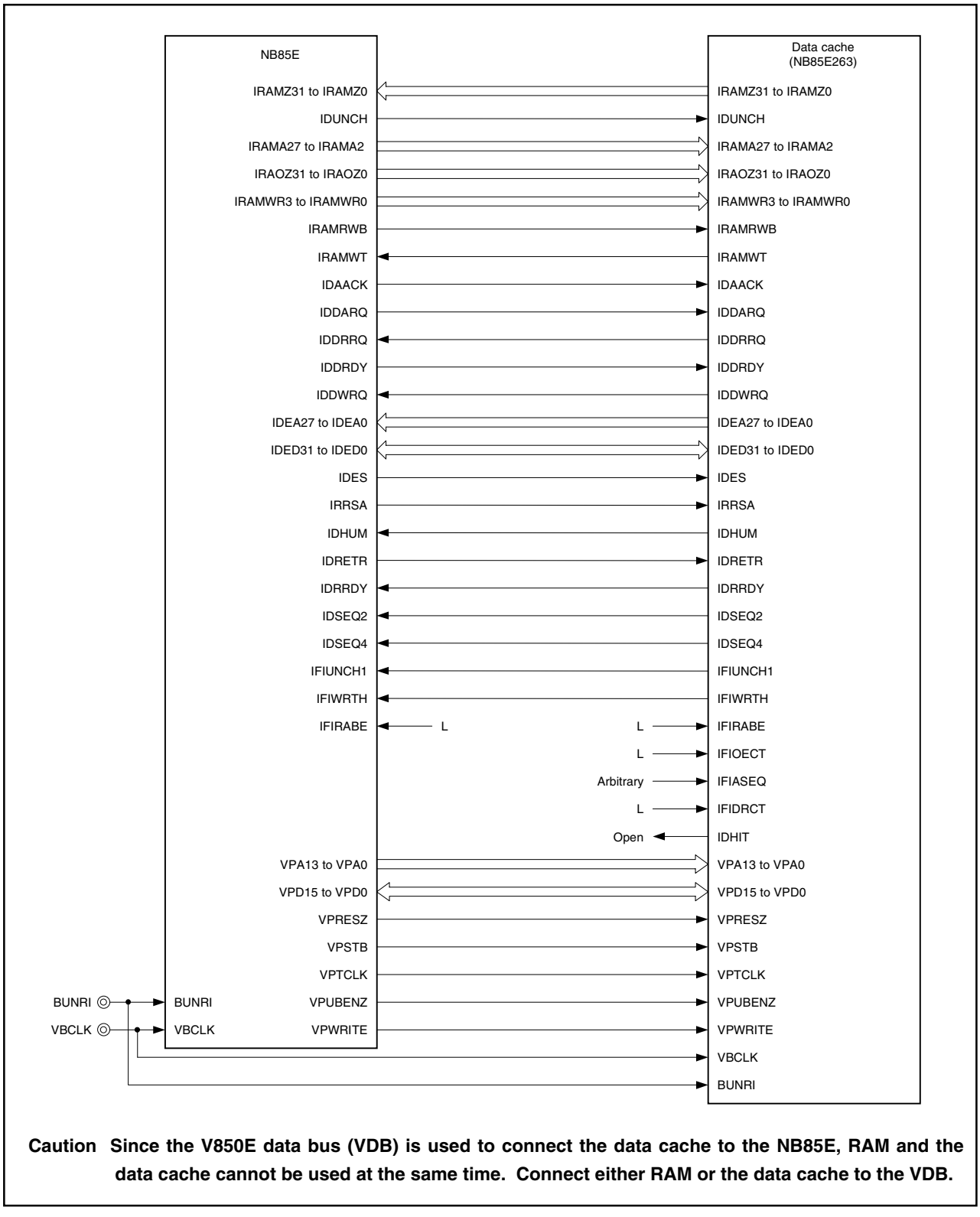


Note This pin is only in the NB85E263. It is not in the NB85E252.

2.1.3 NB85E connection example

The following figure shows an example of the connection of a data cache to the NB85E.

Figure 2-1. NB85E and Data Cache Connection Example



★ **Caution Since the V850E data bus (VDB) is used to connect the data cache to the NB85E, RAM and the data cache cannot be used at the same time. Connect either RAM or the data cache to the VDB.**

2.2 Pin Functions

2.2.1 List of pin functions

(1/2)

Pin Name		I/O	Function
NB85E connection pins	IRAMA27 to IRAMA2	Input	Input address from NB85E
	IRAOZ31 to IRAOZ0	Input	Input data from NB85E
	IDDARQ	Input	Input read/write access request from NB85E
	IDUNCH	Input	Input uncache status
	IDES	Input	NEC reserved pin (connect to IDES pin of NB85E)
	IRAMRWB	Input	Input read/write status from NB85E
	IRAMWR3 to IRAMWR0	Input	Input write enable from NB85E
	IRRSA	Input	Input VDB hold status
	IRAMWT	Output	Output wait to NB85E
	IRAMZ31 to IRAMZ0	Output	Output data to NB85E
	IDRRDY	Output	Output read data ready to NB85E
	IDHUM	Output	Output hit under miss read
	IDRETR	Input	Input read retry request
	IDEA27 to IDEA0	Output	Output address
	IDED31 to IDED0	I/O	Input and output data
	IDDRRQ	Output	Output VSB read operation request to NB85E
	IDDWRQ	Output	Output VSB write operation request to NB85E
	IDSEQ4	Output	Output read or write operation type setting
	IDSEQ2	Output	Output read or write operation type setting
	IDAACK	Input	Input acknowledge
	IDDRDY	Input	Input read data ready from NB85E
	IFIUNCH1	Output	Output data cache setting to NB85E
	IFIWRTH	Output	Output writeback or write through mode selection
	VPA13 to VPA0	Input	Input address (for NPB)
	VPWRITE	Input	Input write access strobe (for NPB)
	VPUBENZ	Input	Input higher byte enable (for NPB)
	VPSTB	Input	Input data strobe (for NPB)
	VPD15 to VPD0	I/O	Input and output data (for NPB)
	VPRESZ	Input	Input reset
	VBCLK	Input	Input internal system clock

Remark VDB: V850E data bus
VSB: V850E system bus
NPB: NEC peripheral I/O bus

(2/2)

Pin Name		I/O	Function
Cache type selection pins	IFIASEQ	Input	Input refill mode selection
	IFIRABE	Input	NEC reserved pin (input a low level)
	IFIDRCT[Note]	Input	NEC reserved pin (input a low level)
	IFIOECT	Input	NEC reserved pin (input a low level)
Status pin	IDHIT	Output	Output tag hit status
Test mode pins	BUNRI	Input	Input normal/test mode selection
	VPTCLK	Input	Input clock for test

Note This pin is only in the NB85E263. It is not in the NB85E252.

2.2.2 Explanation of pin functions

(1) NB85E connection pins

(a) IRAMA27 to IRAMA2 (input)

IRAMA27 to IRAMA2 constitute a bus that inputs an address from the NB85E.

(b) IRAOZ31 to IRAOZ0 (input)

IRAOZ31 to IRAOZ0 constitute a bus that inputs data from the NB85E.

(c) IDDARQ (input)

IDDARQ inputs a read or write access request from the NB85E.

(d) IDUNCH (input)

IDUNCH inputs the uncache status.

A low level is input when an area for which the data cache setting was set to cacheable by the cache configuration register (BHC) of the NB85E is accessed.

(e) IDES (input)

IDES is reserved for NEC. Be sure to connect it to the IDES pin of the NB85E.

(f) IRAMRWB (input)

IRAMRWB inputs read or write status from the NB85E.

The read or write status of the data cache is as follows depending on the input level at this pin.

- Low level: Write
- High level: Read

(g) IRAMWR3 to IRAMWR0 (input)

IRAMWR3 to IRAMWR0 input a write enable from the NB85E.

These pins indicate the valid byte data in the data bus (IRAOZ31 to IRAOZ0). A high level is input when byte data is valid.

High Level Signal	Valid Byte Data
IRAMWR0	IRAOZ7 to IRAOZ0
IRAMWR1	IRAOZ15 to IRAOZ8
IRAMWR2	IRAOZ23 to IRAOZ16
IRAMWR3	IRAOZ31 to IRAOZ24

(h) IRRSA (input)

IRRSA inputs the V850E data bus (VDB) hold status.

A high level is input if the VDB is in a RAM access or hold status.

(i) IRAMWT (output)

IRAMWT outputs a wait to the NB85E.

This pin outputs high level during the wait interval.

(j) IRAMZ31 to IRAMZ0 (output)

IRAMZ31 to IRAMZ0 constitute a bus that outputs data to the NB85E.

(k) IDRRDY (output)

IDRRDY outputs a read data ready to the NB85E.

(l) IDHUM (output)

IDHUM outputs a hit under miss read.

The next access to the data cache is performed during access to external memory upon occurrence of a miss during read, and if the data hit on that access is input to the NB85E prior to data from external memory (hit under miss), a high level is output.

(m) IDRETR (input)

IDRETR inputs a read retry request.

(n) IDEA27 to IDEA0 (output)

IDEA27 to IDEA0 constitute a bus that outputs an address to the NB85E.

This bus outputs the address the NB85E should access when a miss occurs.

(o) IDED31 to IDED0 (I/O)

IDED31 to IDED0 constitute a bus that inputs and outputs data from and to the NB85E.

This bus passes data refilled in the data cache and data to write to external memory when in writeback mode.

(p) IDDRRQ, IDDWRQ, IDSEQ4, and IDSEQ2 (output)

IDDRRQ, IDDWRQ, IDSEQ4, and IDSEQ2 output operation type settings to the NB85E.

IDDRRQ	IDDWRQ	IDSEQ4	IDSEQ2	Operation Type
H	L	H	L	4-word sequential read
H	L	L	H	2-word sequential read
H	L	L	L	1-word read
L	H	H	L	4-word sequential write
L	H	L	H	2-word sequential write
L	H	L	L	1-word write
H	H	H	H	1-word write
H	H	H	L	1-halfword write
H	H	L	L	1-byte write
Other than above				Setting prohibited

Remark L: Low-level output
H: High-level output

(i) IDDRRQ (output)

IDDRRQ outputs a VSB read operation request to the NB85E.

(ii) IDDWRQ (output)

IDDWRQ outputs a VSB write operation request to the NB85E.

(iii) IDSEQ4, IDSEQ2 (output)

IDSEQ4 and IDSEQ2 output read and write operation type settings to the NB85E.

(q) IDAACK (input)

IDAACK inputs an acknowledge.

This signal is input when the NB85E recognizes IDEA27 to IDEA0 signals output from the data cache.

(r) IDDRDY (input)

IDDRDY inputs a read data ready from the NB85E.

This is input when the NB85E is finished getting the data it was to read from external memory at the time of a miss, and indicates to the data cache that preparations for refill have been made.

(s) IFIUNCH1 (output)

IFIUNCH1 outputs the data cache setting to the NB85E.

This pin outputs a low level when the data cache is enabled and a high level when the data cache is disabled (both the DC11 and DC10 bits of the DCC register are cleared (0)).

IFIUNCH1 must be connected to IFIUNCH1 in the NB85E.

(t) IFIWRTH (output)

IFIWRTH outputs the writeback/write through mode selection.

This pin outputs a low level for writeback mode and a high level for write through mode.

(u) VPA13 to VPA0, VPWRITE, VPUBENZ, VPSTB, VPD15 to VPD0 (NPB pins)

Refer to the **NB85E Hardware User's Manual (A13971E)**.

(v) VPRESZ (input)

VPRESZ inputs a reset.

(w) VBCLK (input)

VBCLK inputs the internal system clock.

(2) Cache type selection pins

(a) IFIASEQ (input)

IFIASEQ inputs the refill mode selection.

The refill modes are as follows depending on the level input to this pin.

- Low level: Critical first mode
- High level: Sequential mode

Remark Critical first mode is a technique of taking the data needed first when taking a line of data from external memory. Because the data is passed quickly to the CPU, overall system performance generally improves when operating in critical first mode. However, when memory that can be continuously accessed is connected, performance may be better if sequential mode is set.

(b) IFIRABE, IFIDRCT, and IFIOECT (input)

IFIRABE, IFIDRCT, and IFIOECT are reserved for NEC. Always input a low level.

The IFIDRCT pin is only in the NB85E263. It is not in the NB85E252.

(3) Status pin

(a) IDHIT (output)

IDHIT indicates that the cache was hit.

When data is hit during read from data cache, a high level is output. During write, the status is not indicated. Leave this pin open when not used.

(4) Test mode pins

(a) BUNRI (input)

BUNRI is an input pin that selects normal or test mode.

(b) VPTCLK (input)

VPTCLK inputs the clock for testing.

2.2.3 Pin status

The following table shows the status in each operating mode of the pins that have output functions.

Table 2-1. Pin Status in Each Operating Mode

Pin Name		Pin Status			
		Reset	STOP Mode	HALT Mode	Test Mode
NB85E connection pins	IRAMWT	Undefined	Maintained	Operating	Operating
	IRAMZ31 to IRAMZ0	Undefined	Maintained	Operating	Operating
	IDRRDY	Undefined	Maintained	Operating	Operating
	IDHUM	Undefined	Maintained	Operating	Operating
	IDEA27 to IDEA0	Undefined	Maintained	Operating	Operating
	IDED31 to IDED0	Undefined	Maintained	Operating	Operating
	IDDRRQ	Undefined	Maintained	Operating	Operating
	IDDWRQ	Undefined	Maintained	Operating	Operating
	IDSEQ4	Undefined	Maintained	Operating	Operating
	IDSEQ2	Undefined	Maintained	Operating	Operating
	IFIUNCH1	H	Maintained	Operating	Operating
	IFIWRTH	H	Maintained	Operating	Operating
	VPD15 to VPD0	Hi-Z	Maintained	Operating	Operating
Status pin	IDHIT	Undefined	Maintained	Operating	Operating

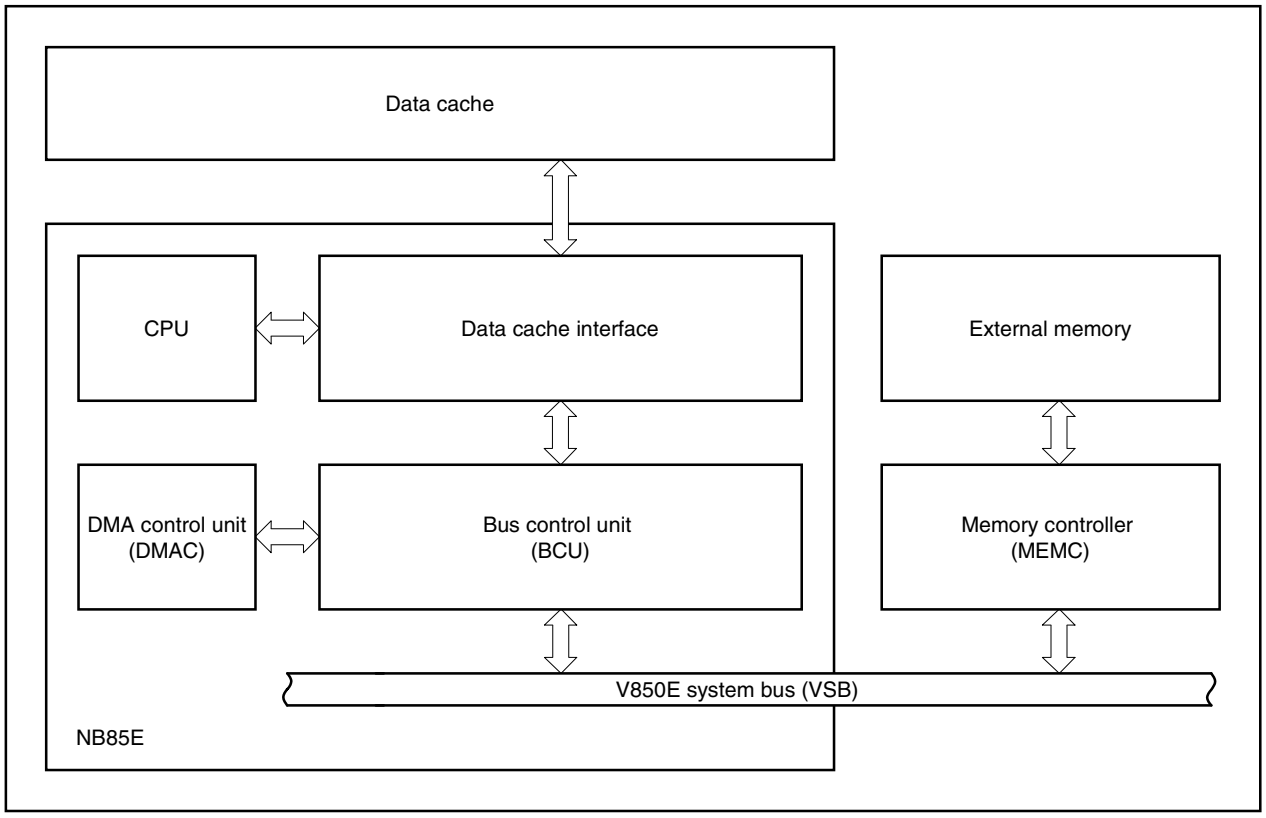
Remark H: High-level output
Hi-Z: High impedance
Maintained: The previous status is maintained

2.3 Configuration of Data Cache

The following two types of data caches are available. The NB85E can be accessed by one of these data caches in one cycle.

- 4 KB directly mapped data cache (NB85E252)
- 8 KB 2-way set-associative data cache (NB85E263)

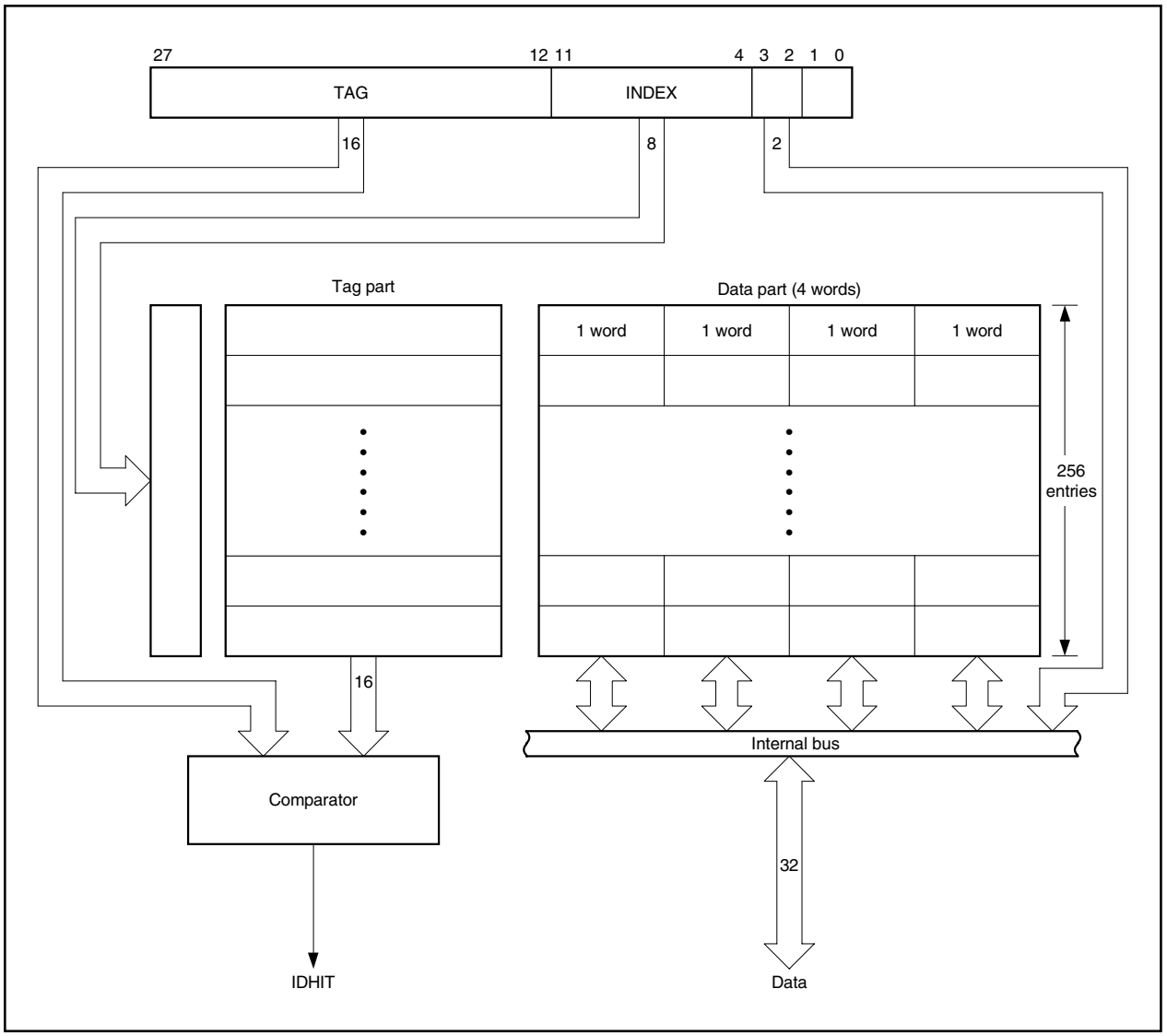
Figure 2-2. Data Cache Configuration Example



2.3.1 4 KB directly mapped data cache

The data memory of a 4 KB directly mapped data cache, which consists of a block of 256 entries of 4 words per line, has a total capacity of 4 KB.

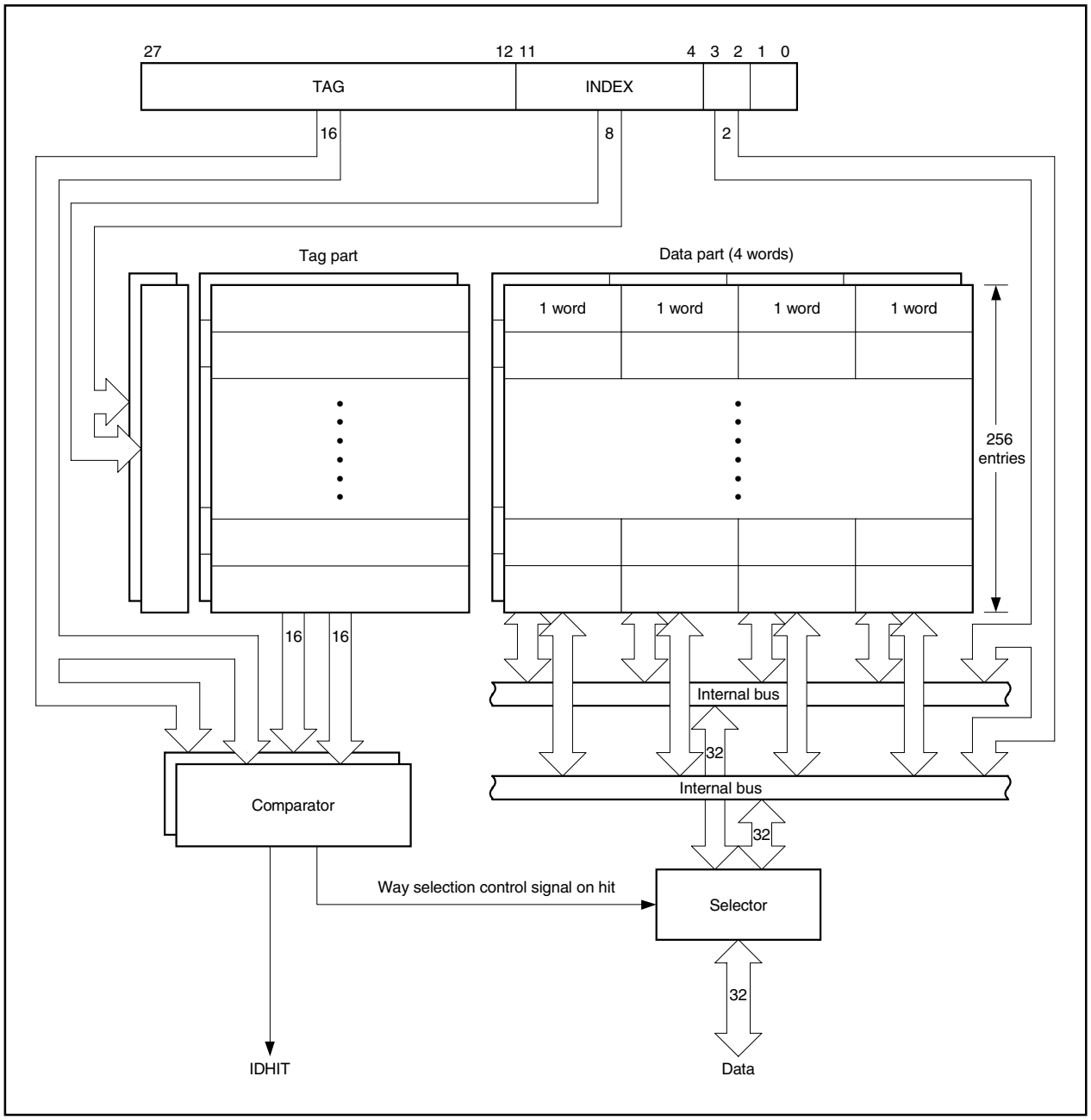
Figure 2-3. Configuration of 4 KB Directly Mapped Data Cache



2.3.2 8 KB 2-way set-associative data cache

The data memory of an 8 KB 2-way set-associative data cache has 2 ways, each consisting of a block of 256 entries of 4 words per line, for a total capacity of 8 KB.

Figure 2-4. Configuration of 8 KB 2-Way Set-Associative Data Cache



2.4 Data Cache Control Functions

2.4.1 Control registers

The following are the data cache control functions.

- Tag clear function
- Tag fill function
- Lock function
- Data flush function

These functions are controlled by the following registers.

Address	Register Name	Symbol	R/W	Manipulatable Bits			Initial Value
				1 Bit	8 Bits	16 Bits	
FFFFF078H	Data cache control register	DCC	R/W			√	0003H[Note]
FFFFF07CH	Data cache data configuration register	DCD	R/W			√	Undefined

Note While reset is active, the value of this register becomes 0003H, and tag initialization begins automatically. Upon completion of tag initialization, the value changes to 0000H.

Remark The DCC register and DCD register are allocated in the peripheral I/O area of the NB85E.

(1) Data cache control register (DCC)

The DCC register sets four types of functions: tag clear, tag fill, lock, and data flush. In addition, three operating modes can be selected using DCC register settings.

The DCC register can be read or written in 16-bit units.

Cautions 1. If any of bits 0, 1, 4, or 5 is set (1), do not forcibly clear (0) that bit.

2. Settings in bits 1, 5, and 13 are valid only for the NB85E263. Be sure to set these bits to 0 in the NB85E252.

3. After data cache initialization on a reset, settings in bits 10 and 11 can be changed one time only before the first access.

4. Be sure to set the data cache enable area (using the NB85E's cache configuration register (BHC)) after setting the operation mode with the DCC register. Failure to do this will result in the inability to set the data cache enable area on the NB85E side.

Figure 2-5. Data Cache Control Register (DCC) (1/2)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
DCC	0	0	DC13	DC12	DC11	DC10	0	0	0	0	DC05	DC04	0	0	DC01	DC00	FFFFF078H	0003H

Bit position	Bit name	Description
13	DC13	This bit selects the cache lock setting of way 1 and clear or fill for the tag clear or tag fill function. Setting (1) this bit locks the cache of way 1 and disables writing.

DC13	Cache lock	Tag clear/fill
0	Way 1 is not locked	Way 1 tag clear function is valid
1	Way 1 is locked	Way 1 tag fill function is valid

Bit position	Bit name	Description
12	DC12	This bit selects the cache lock setting of way 0 and clear or fill for the tag clear or tag fill function. Setting (1) this bit locks the cache of way 0 and disables writing.

DC12	Cache lock	Tag clear/fill
0	Way 0 is not locked	Way 0 tag clear function is valid
1	Way 0 is locked	Way 0 tag fill function is valid

Bit position	Bit name	Description
11, 10	DC11, DC10	These bits set the operating mode.

DC11	DC10	Operating mode
0	0	Cache access disabled
1	0	Write through mode
0	1	Writeback mode (write allocate disabled)
1	1	Writeback mode (write allocate enabled)

Remark Write allocate refers to refilling the data cache with data of the address to be written from external memory when a miss occurs in writing to the data cache.

Bit position	Bit name	Description
5	DC05	This bit sets a way 1 data flush. Setting (1) this bit flushes a way 1 dirty data line. When a data flush is complete, this bit is cleared (0) automatically. 0: Way 1 data flush complete 1: Way 1 data flush operating

Figure 2-5. Data Cache Control Register (DCC) (2/2)

Bit position	Bit name	Description
4	DC04	This bit sets a way 0 data flush. Setting (1) this bit flushes a way 0 dirty data line. When a data flush is complete, this bit is cleared (0) automatically. 0: Way 0 data flush complete 1: Way 0 data flush operating
1	DC01	This bit sets a way 1 tag clear or tag fill. Setting (1) this bit clears or fills way 1 tags. When a tag clear or tag fill is complete, this bit is cleared (0) automatically. 0: Way 1 tag clear or tag fill complete 1: Way 1 tag clear or tag fill operating **Remark** Select clear or fill using bit 13 (DC13).
0	DC00	This bit sets a way 0 tag clear or tag fill. Setting (1) this bit clears or fills way 0 tags. When a tag clear or tag fill is complete, this bit is cleared (0) automatically. 0: Way 0 tag clear or tag fill complete 1: Way 0 tag clear or tag fill operating **Remark** Select clear or fill using bit 12 (DC12).

(2) Data cache data configuration register (DCD)

The DCD register sets the address of the memory area to be tag filled when using the tag fill function.

The DCD register can be read or written in 16-bit units.

Cautions 1. Do not overwrite the DCD register while tag fill is operating.

2. Since the initial value of the DCD register is undefined, when using the tag fill function, be sure to set a value in the DCD register prior to setting (1) the DC0n bit of the DCC register (n = 0, 1). If the DC0n bit of the DCC register is set (1) without setting a value in the DCD register, the operation cannot be guaranteed.

Figure 2-6. Data Cache Data Configuration Register (DCD)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
DCD	DD15	DD14	DD13	DD12	DD11	DD10	DD9	DD8	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0	FFFFF07CH	Undefined

Bit position	Bit name	Description
15 to 0	DD15 to DD0	These bits set tag information (bits 27 to 12 of the start address of the memory area to be tag filled).

2.4.2 Tag clear function

The tag clear function clears (invalidates) the tags of one or two ways.

In addition, it automatically clears (invalidates) the tags of all ways on a reset.

Use the following procedure to perform a data cache tag clear.

<1> Read the data cache control register (DCC) and confirm that bits 0, 1, 4, and 5 (DC00, DC01, DC04, DC05) are all cleared (0).

<2> Clear (0) DCC register bit 12 (DC12), bit 13 (DC13), or both depending on the way for which tags are to be cleared.

<3> Set (1) DCC register bit 0 (DC00), bit 1 (DC01) or both depending on the way for which tags are to be cleared.

<4> Read DCC register bit DC00, DC01, or both depending on the way for which tags were cleared and confirm that that bit is cleared (0).

Caution The tag clear function does not flush dirty data lines even in writeback mode. If data flush is necessary, use the data flush function.

2.4.3 Tag fill function

The tag fill function fills the tags of one or two ways.

By locking a filled way, the data cache can be used as data RAM. When the lock is released, it again operates as a data cache.

In write through mode, a bus cycle is issued even on a write access to a filled or locked address area.

Use the following procedure to perform a data cache tag fill.

<1> Set the tag information corresponding to the memory area to be tag filled in the data cache data configuration register (DCD).

<2> Read the data cache control register (DCC) and confirm that bits 0, 1, 4, and 5 (DC00, DC01, DC04, DC05) are all cleared (0).

<3> Set (1) DCC register bit 12 (DC12), bit 13 (DC13), or both depending on the way to be tag filled.

<4> Set (1) DCC register bit 0 (DC00), bit 1 (DC01), or both depending on the way to be tag filled.

<5> Read DCC register bit DC00, DC01, or both depending on the way that was tag filled and confirm that that bit is cleared (0).

2.4.4 Lock function

The lock function locks any way.

Once locked, a way is write disabled and operates the same as data RAM that is accessible in one cycle. When the lock is released, it again operates as a data cache.

In write through mode, a bus cycle is issued even on an access to a locked line.

Set or release a data cache lock as follows.

- Setting a lock

  Set (1) bit 12 (DC12) or bit 13 (DC13) of the data cache control register (DCC) depending on the way for which the lock is to be set.

- Releasing a lock

  Clear (0) bit DC12 or bit DC13 of the DCC register depending on the way for which the lock is to be released.

2.4.5 Data flush function

The data flush function flushes a dirty data line in writeback mode.

Use the following procedure to perform a data cache data flush.

<1> Read the data cache control register (DCC) and confirm that bits 0, 1, 4, and 5 (DC00, DC01, DC04, DC05) are all cleared (0).
<2> Clear (0) DCC register bit 12 (DC12), bit 13 (DC13), or both depending on the way to be data flushed.
<3> Set (1) DCC register bit 4 (DC04), bit 5 (DC05), or both depending on the way to be data flushed.
<4> Read DCC register bit DC04, DC05, or both depending on the way that was data flushed and confirm that that bit is cleared (0).

Cautions 1. If bits DC04 and DC00 or bits DC05 and DC01 are set (1) at the same time, data flush and tag clear are performed at the same time.

2. When manipulating multiple ways simultaneously, make the settings the same. For example, a way 0 data flush and a way 1 data flush and tag clear cannot be performed simultaneously.

★ 2.5 Data Cache Setting Procedure

2.5.1 Setting to validate data cache

To validate the data cache, follow the procedure below using the initial settings of the user program immediately following system reset.

<1> Set the data cache control register (DCC) and data cache data configuration register (DCD) (always set the operating mode with bits DC11 and DC10 of the DCC register).
<2> Make the data cache setting of the cache configuration register (BHC) of the NB85E “cacheable”.

2.5.2 Setting to validate, invalidate, and revalidate data cache

To invalidate (OFF) the data cache in the middle of a user program and validate (ON) it again, use the data cache control register (DCC) or the cache configuration register (BHC) of the NB85E, and follow the procedure below.

(1) Using DCC register

<1> Read the DCC register and confirm that a tag clear is not in progress.
<2> Write 0x0 to the DCC register.
<3> Read the DCC register and confirm that the register is cleared (0).
:
<Operation with data cache OFF>
:
<4> Read the DCC register and confirm that the register is cleared (0).
<5> Set the operating mode with bits 11 and 10 (DC11, DC10) of the DCC register.
<6> NOP

(2) Using BHC register of NB85E

<1> Make the data cache setting of the BHC register of the NB85E “uncacheable”.
<2> Read the DCC register and confirm that a tag clear is not in progress.
<3> Set the tag clear bit of the DCC register.
<4> Read the DCC register and confirm that the tag clear is complete.
:
<Operation with data cache OFF>
:
<5> Make the data cache setting of the BHC register of the NB85E “cacheable”.

Caution Do not perform other LD/ST operations after BHC register write <1> and before tag clear completion using the DCC register <4>.

2.6 Operation

The data cache automatically performs a caching operation whenever there is an access to a cacheable area set using the cache configuration register (BHC) of the NB85E.

The data cache has three operating modes that are selected according to data cache control register (DCC) settings. Table 2-2 shows a list of the operating modes.

Table 2-2. List of Operating Modes

Operating Mode	Access Type			Refill Mode
Write through mode	Read	Hit		–
		Miss		Sequential
				Critical first
	Write	Hit		–
		Miss		–
Writeback mode (write allocate disabled)	Read	Hit		–
		Miss	Clean data	Sequential
				Critical first
			Dirty data	Sequential
				Critical first
	Write	Hit		–
		Miss		–
Writeback mode (write allocate enabled)	Read	Hit		–
		Miss	Clean data	Sequential
				Critical first
			Dirty data	Sequential
				Critical first
	Write	Hit		–
		Miss	Clean data	Sequential
				Critical first
			Dirty data	Sequential
				Critical first

Data cache operation is described below for each operating mode.

2.6.1 Write through mode

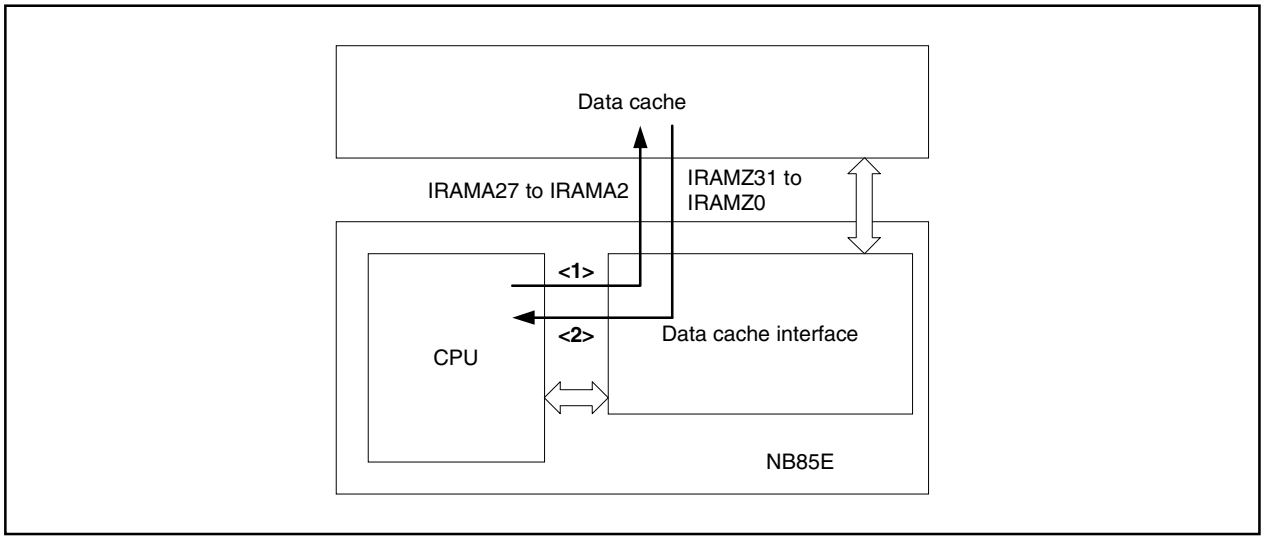
(1) On a read

(a) Data cache hit

<1> When reading data from external memory, output the address (IRAMA27 to IRAMA2) to the data cache.

<2> If a hit occurs due to the address existing in the data cache, read the data by passing through IRAMZ31 to IRAMZ0 from the data cache.

Figure 2-7. Operation on Data Cache Hit (Write Through Mode, Read)



(b) Data cache miss

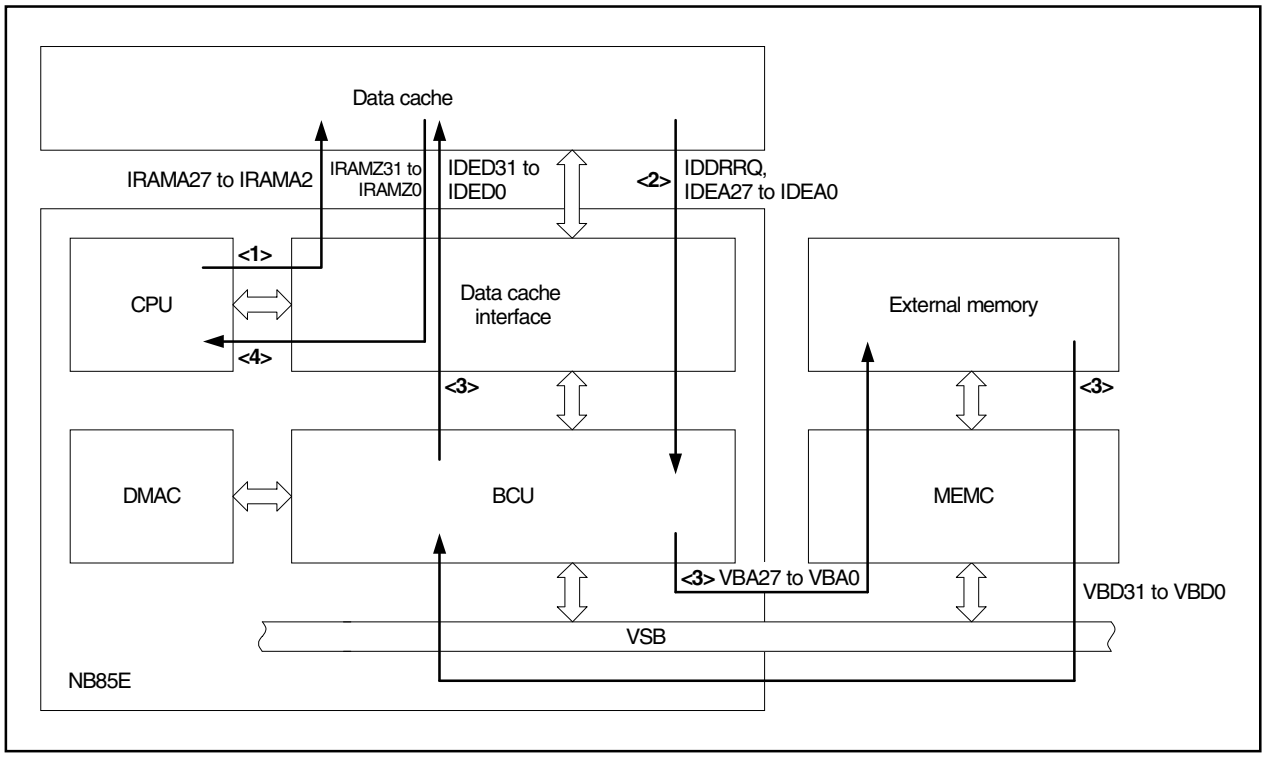
<1> When reading data from external memory, output the address (IRAMA27 to IRAMA2) to the data cache.

<2> If a miss occurs due to the address not existing in the data cache, output a fetch request (IDDRRQ) and the address to be read (IDEA27 to IDEA0) from the data cache to the BCU.

<3> The BCU of the internal NB85E outputs the address (VBA27 to VBA0) to external memory via the VSB and refills the data cache with one line (4 words) of data of the address to be read.

<4> The data cache then transfers the required data among the 4 words of refill data to the CPU by passing through IRAMZ31 to IRAMZ0.

Figure 2-8. Operation on Data Cache Miss (Write Through Mode, Read)



(2) On a write

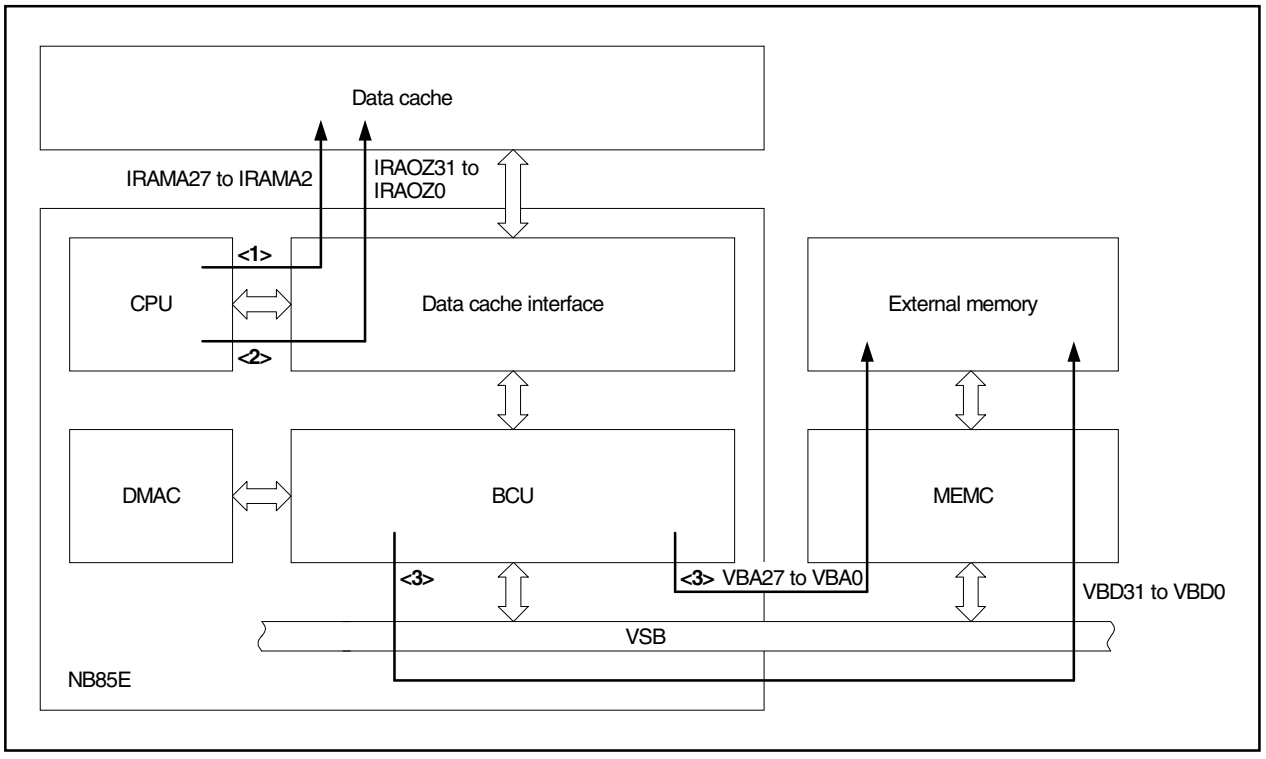
(a) Data cache hit

<1> When writing data to external memory, output the address (IRAMA27 to IRAMA2) to the data cache.

<2> If a hit occurs due to the external memory address to be written existing in the data cache, write the data to the data cache by passing through IRAOZ31 to IRAOZ0.

<3> The BCU of the internal NB85E outputs the address (VBA27 to VBA0) to external memory via the VSB and writes the same data as IRAOZ31 to IRAOZ0 to external memory by passing through VBD31 to VBD0.

Figure 2-9. Operation on Data Cache Hit (Write Through Mode, Write)

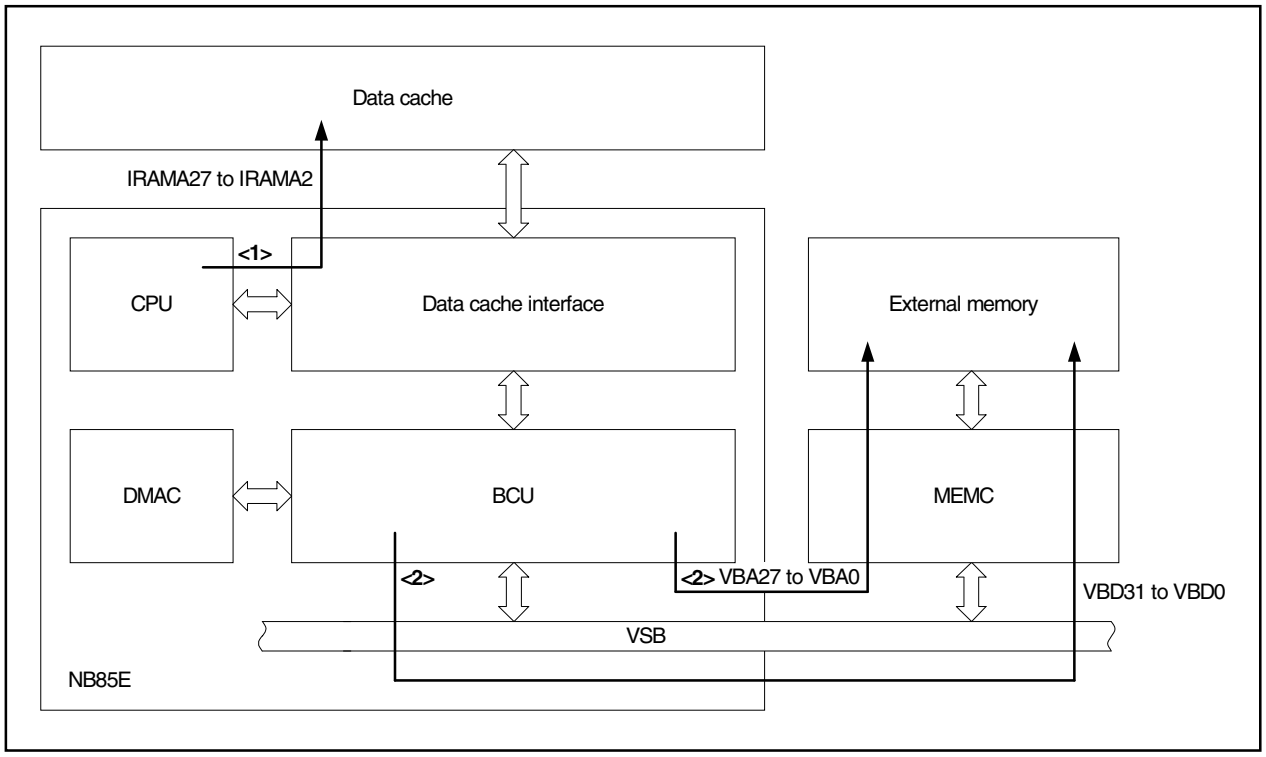


(b) Data cache miss

<1> When writing data to external memory, output the address (IRAMA27 to IRAMA2) to the data cache. If a miss occurs due to the address not existing in the data cache, data is not written to the data cache.

<2> The BCU of the internal NB85E outputs the address (VBA27 to VBA0) to external memory via the VSB and writes the data to be written to external memory by passing through VBD31 to VBD0.

Figure 2-10. Operation on Data Cache Miss (Write Through Mode, Write)



2.6.2 Writeback mode (write allocate disabled)

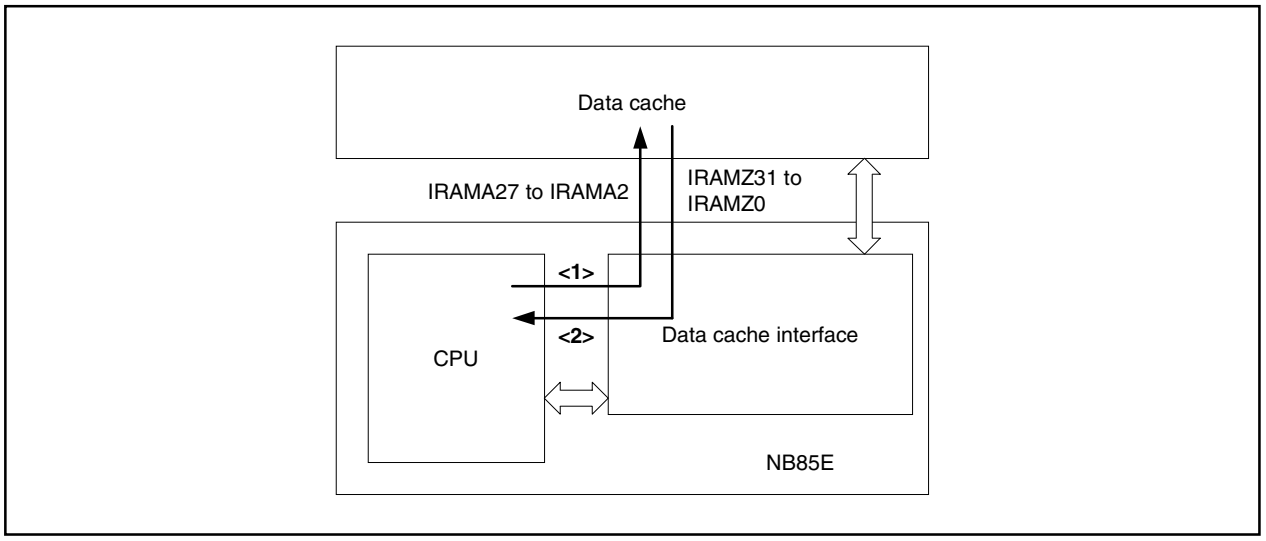
(1) On a read

(a) Data cache hit

<1> When reading data from external memory, output the address (IRAMA27 to IRAMA2) to the data cache.

<2> If a hit occurs due to the address existing in the data cache, read the data by passing through IRAMZ31 to IRAMZ0 from the data cache.

Figure 2-11. Operation on Data Cache Hit (Writeback Mode, Write Allocate Disabled, Read)



(b) Data cache miss

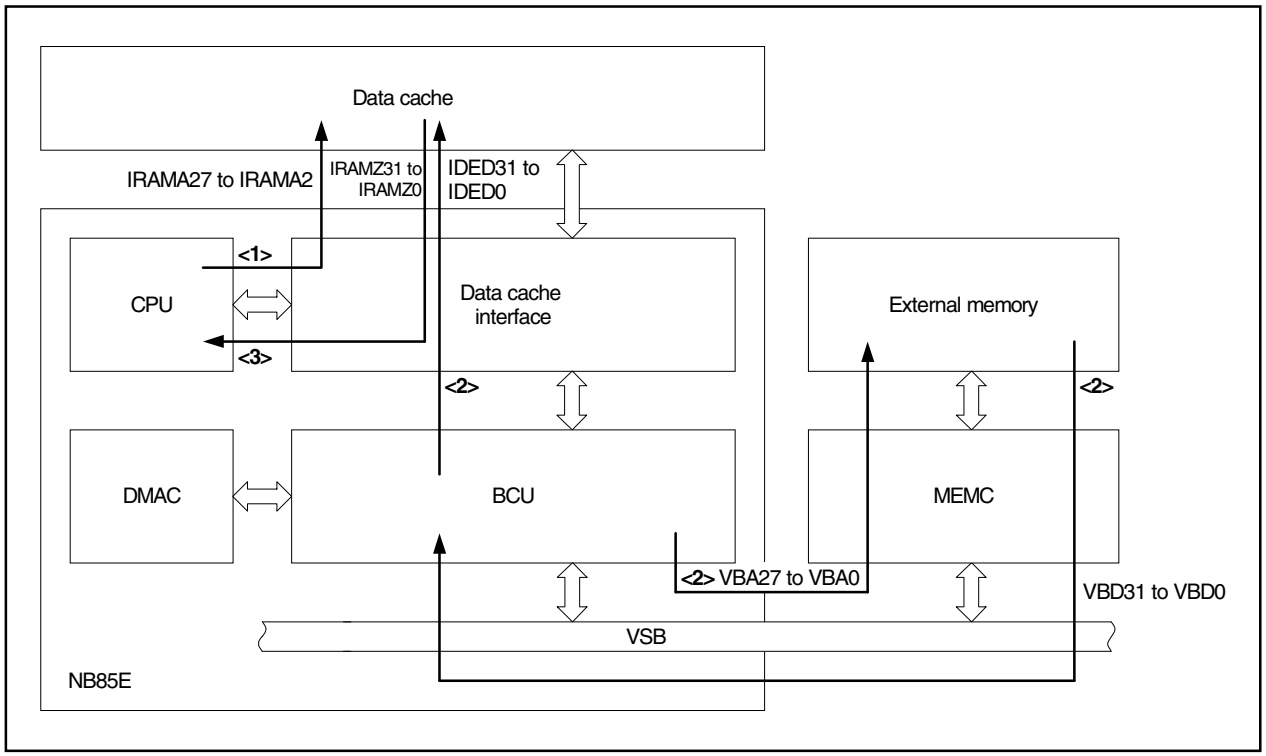
(i) When data being replaced is clean data

<1> When reading data from external memory, output the address (IRAMA27 to IRAMA2) to the data cache. If a miss occurs due to the address not existing in the data cache, perform tag and data replacement. If the data being replaced is clean data, a write operation to external memory is not performed at this time.

<2> The BCU of the internal NB85E outputs the address to be read (VBA27 to VBA0) to external memory via the VSB and refills the data cache with one line (4 words) of data of that address.

<3> The data cache then transfers the required data among the 4 words of refill data to the CPU by passing through IRAMZ31 to IRAMZ0.

Figure 2-12. Operation on Data Cache Miss (Writeback Mode, Write Allocate Disabled, Read, Clean Data)



(ii) When data being replaced is dirty data

<1> When reading data from external memory, output the address (IRAMA27 to IRAMA2) to the data cache. If a miss occurs due to the address not existing in the data cache, perform tag and data replacement.

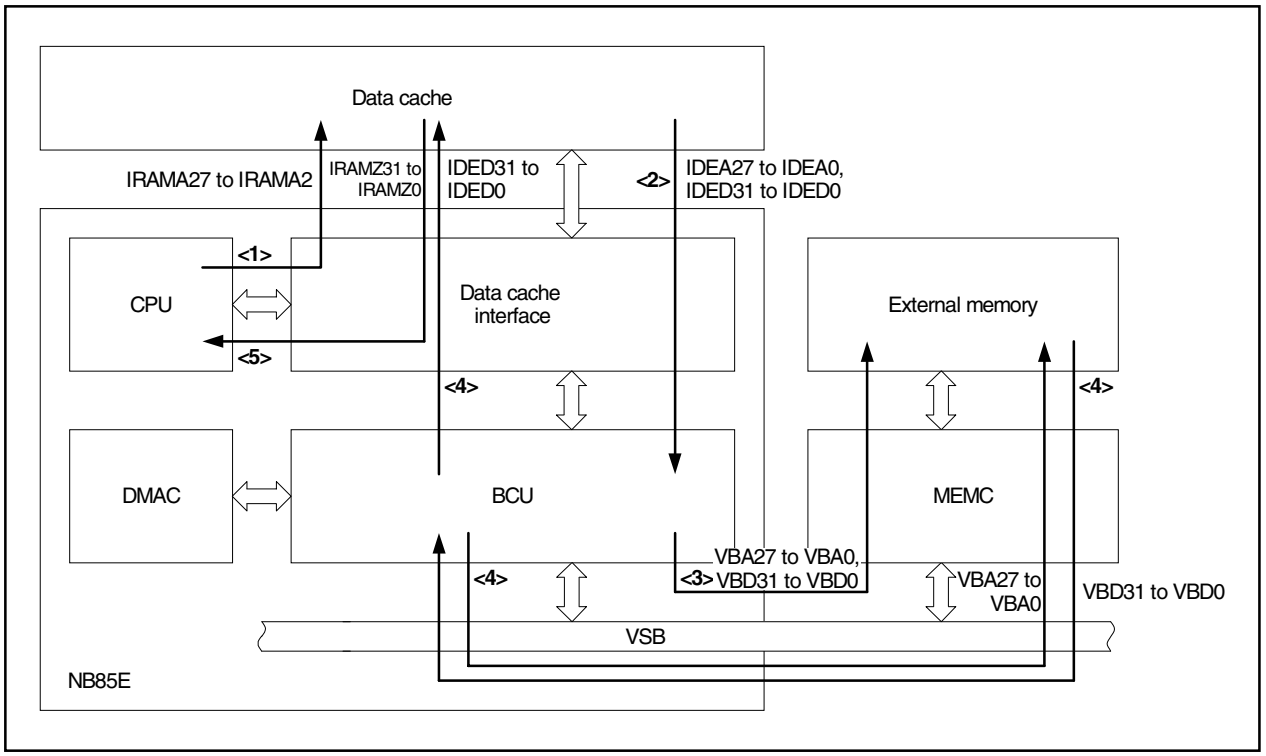
<2> If the data being replaced is dirty data, read the address corresponding to the line that has dirty data from the tag and output it in IDEA27 to IDEA0. At the same time, read the dirty data and output it in IDED31 to IDED0.

<3> The BCU of the internal NB85E outputs the address (VBA27 to VBA0) of the dirty data to external memory via the VSB and writes one line (4 words) of the dirty data to be replaced to external memory.

<4> It then outputs the address to be read (VBA27 to VBA0) to external memory and refills the data cache with 4 words of data of that address.

<5> The data cache then transfers the required data among the 4 words of refill data to the CPU by passing through IRAMZ31 to IRAMZ0.

Figure 2-13. Operation on Data Cache Miss (Writeback Mode, Write Allocate Disabled, Read, Dirty Data)



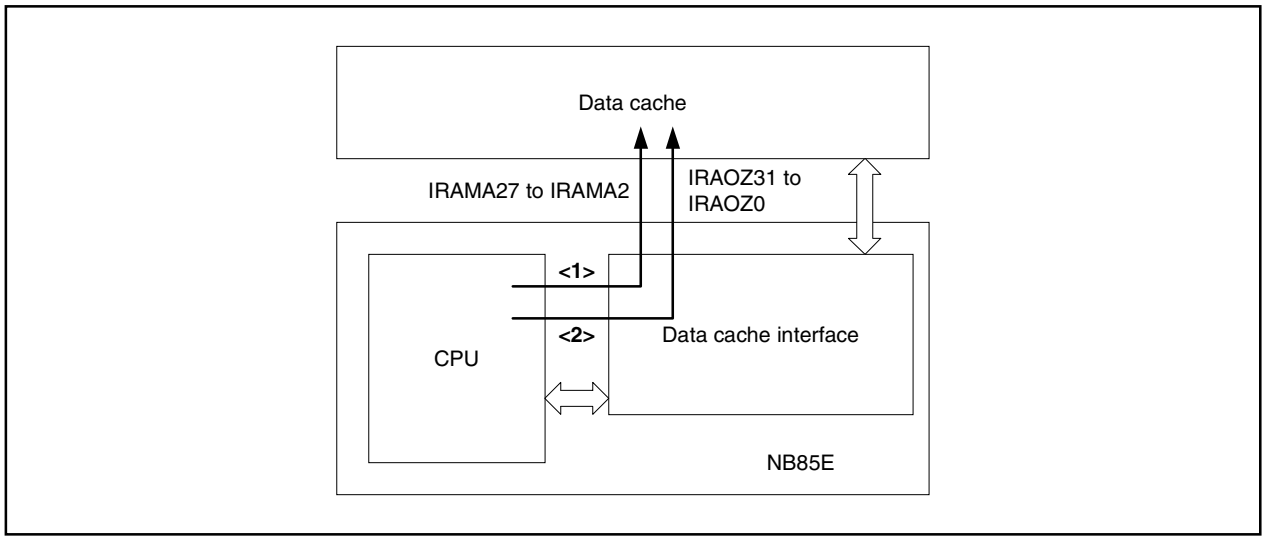
(2) On a write

(a) Data cache hit

<1> When writing data to external memory, output the address (IRAMA27 to IRAMA2) to the data cache.

<2> If a hit occurs due to the external memory address to be written existing in the data cache, write the data to the data cache by passing through IRAOZ31 to IRAOZ0.

Figure 2-14. Operation on Data Cache Hit (Writeback Mode, Write Allocate Disabled, Write)



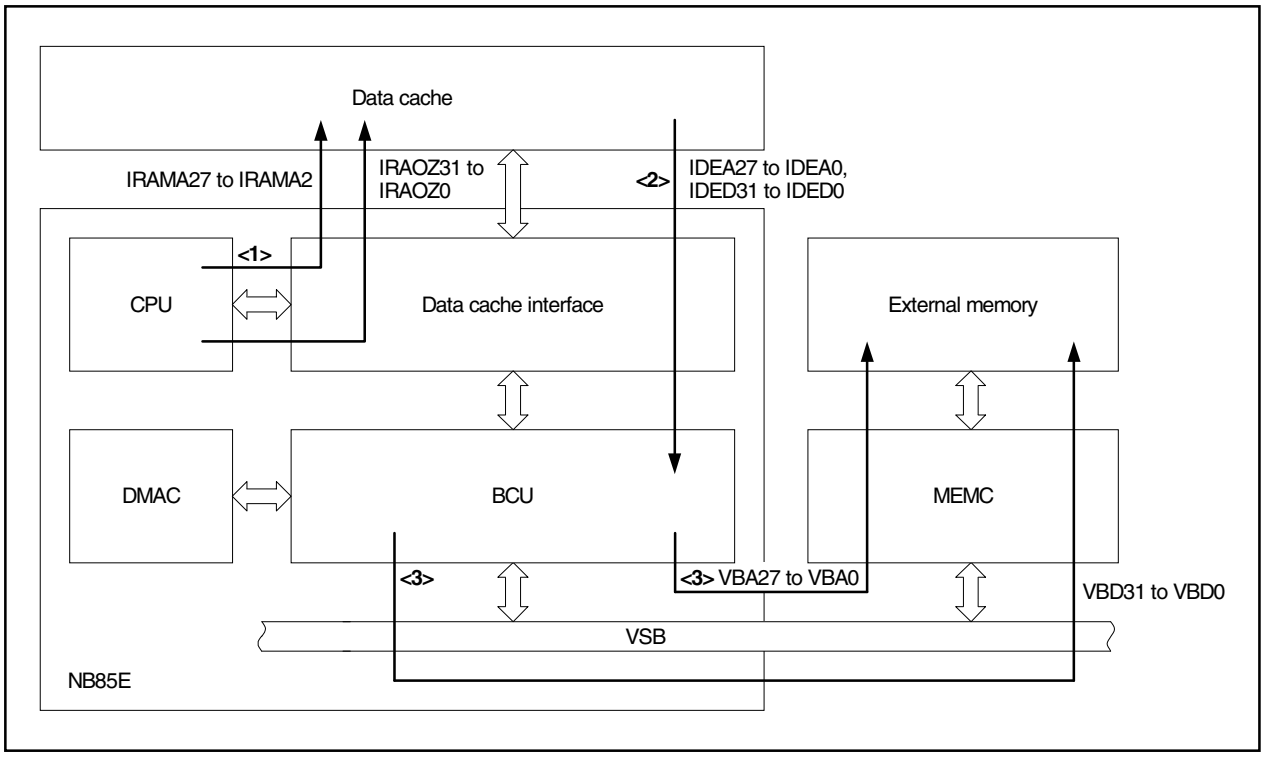
(b) Data cache miss

<1> When writing data to external memory, output the address (IRAMA27 to IRAMA2) to the data cache. If a miss occurs due to the address not existing in the data cache, data is not written to the data cache.

<2> The data cache outputs the address (IRAMA27 to IRAMA2) and data (IRAOZ31 to IRAOZ0) received from the CPU in IDEA27 to IDEA0 and IDED31 to IDED0, respectively.

<3> The BCU of the internal NB85E outputs the address (VBA27 to VBA0) to external memory via the VSB and writes the data to be written to external memory by passing through VBD31 to VBD0.

Figure 2-15. Operation on Data Cache Miss (Writeback Mode, Write Allocate Disabled, Write)



2.6.3 Writeback mode (write allocate enabled)

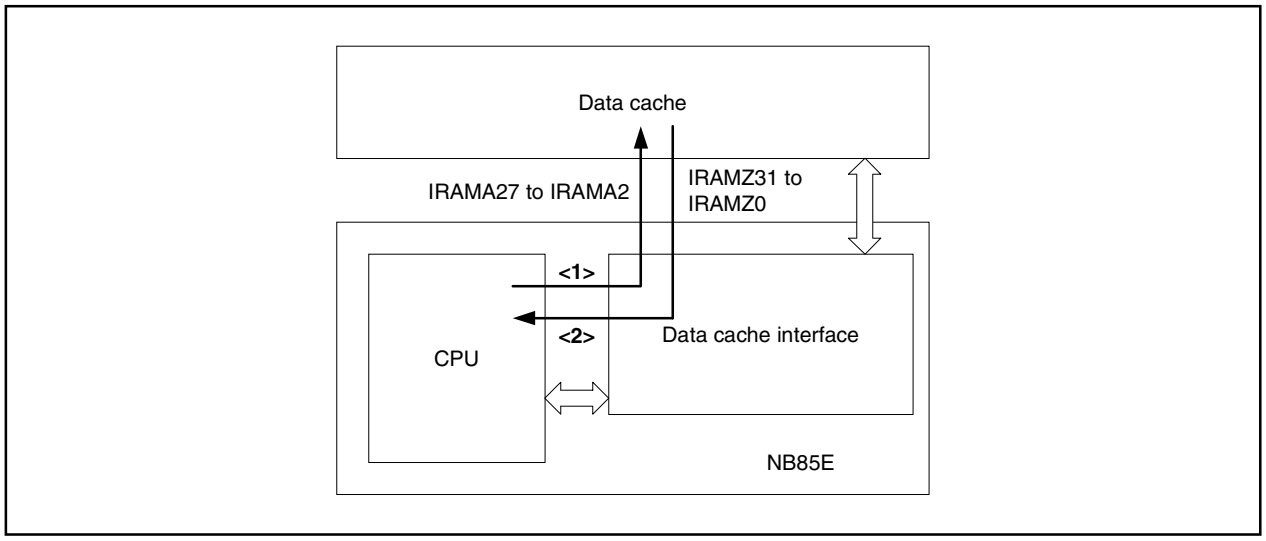
(1) On a read

(a) Data cache hit

<1> When reading data from external memory, output the address (IRAMA27 to IRAMA2) to the data cache.

<2> If a hit occurs due to the address existing in the data cache, read the data by passing through IRAMZ31 to IRAMZ0 from the data cache.

Figure 2-16. Operation on Data Cache Hit (Writeback Mode, Write Allocate Enabled, Read)



(b) Data cache miss

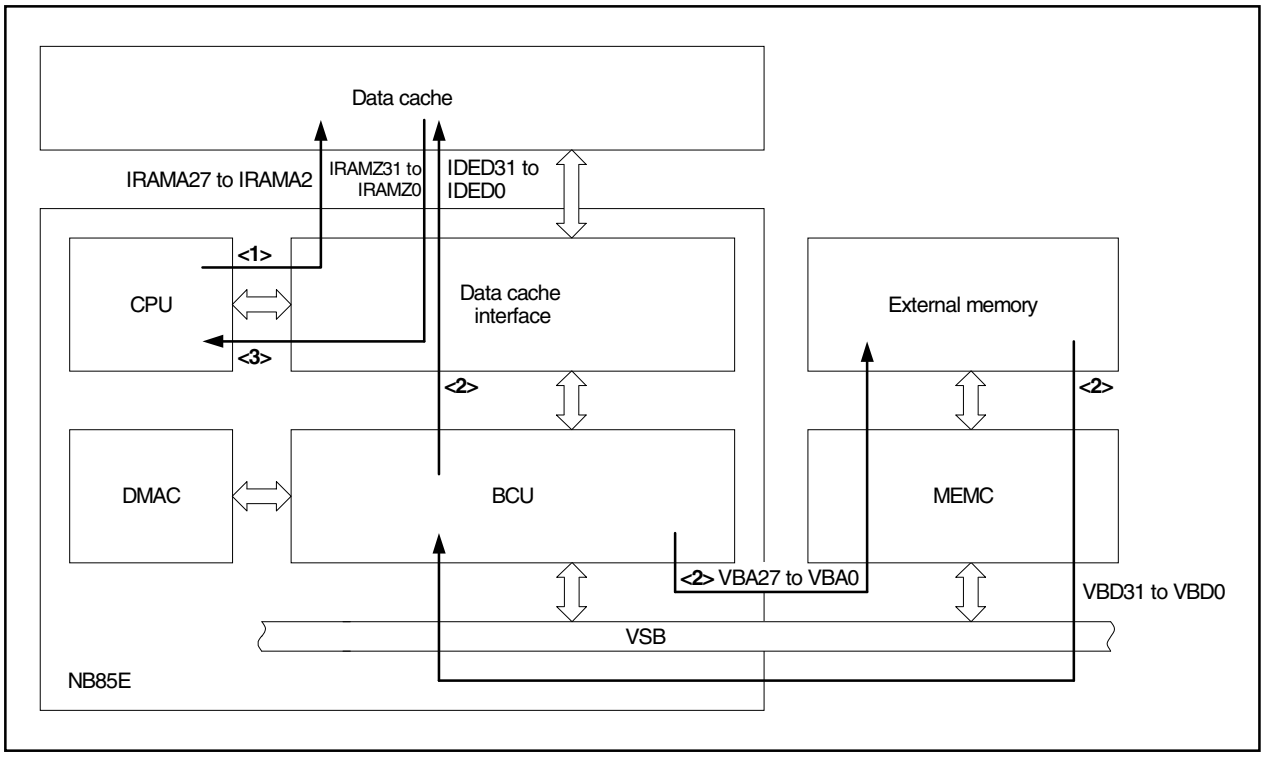
(i) When data being replaced is clean data

<1> When reading data from external memory, output the address (IRAMA27 to IRAMA2) to the data cache. If a miss occurs due to the address not existing in the data cache, perform tag and data replacement. If the data being replaced is clean data, a write operation to external memory is not performed at this time.

<2> The BCU of the internal NB85E outputs the address to be read (VBA27 to VBA0) to external memory via the VSB and refills the data cache with one line (4 words) of data of that address.

<3> The data cache then transfers the required data among the 4 words of refill data to the CPU by passing through IRAMZ31 to IRAMZ0.

Figure 2-17. Operation on Data Cache Miss (Writeback Mode, Write Allocate Enabled, Read, Clean Data)



(ii) When data being replaced is dirty data

<1> When reading data from external memory, output the address (IRAMA27 to IRAMA2) to the data cache. If a miss occurs due to the address not existing in the data cache, perform tag and data replacement.

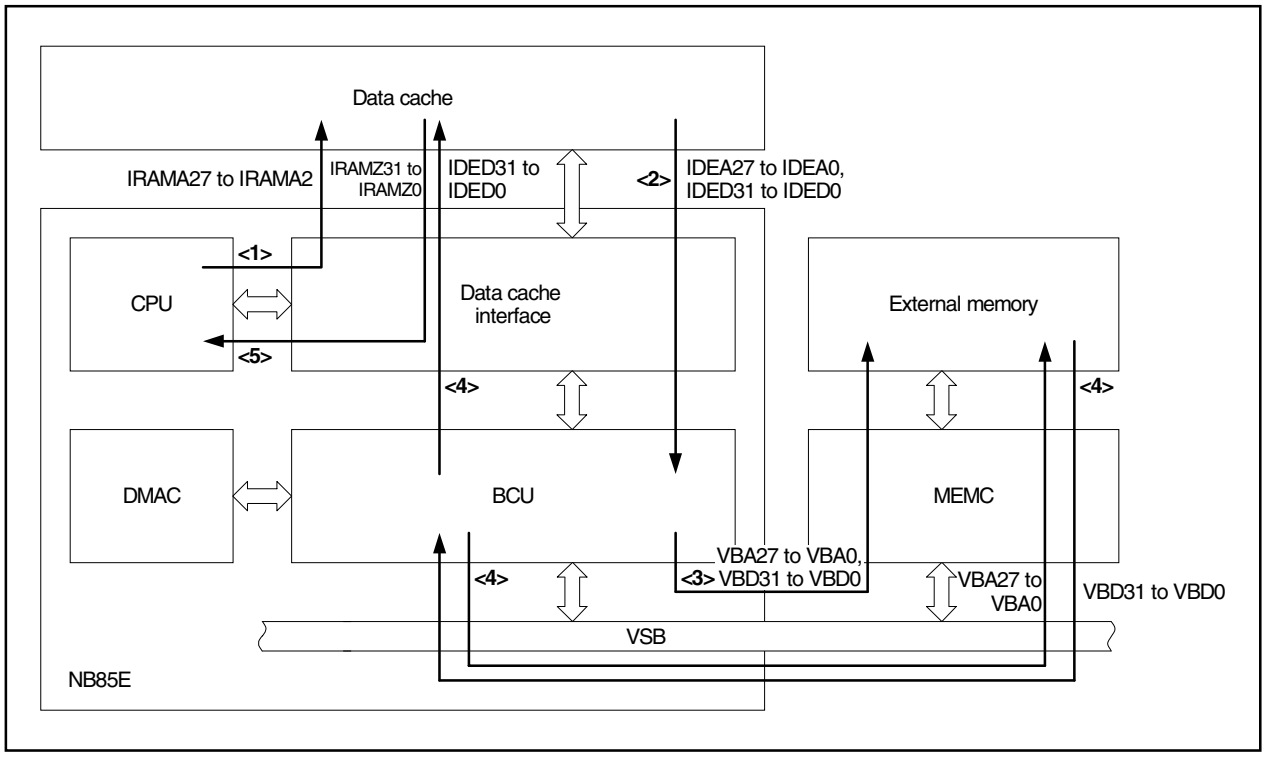
<2> If the data being replaced is dirty data, read the address corresponding to the line that has the dirty data from the tag and output it in IDEA27 to IDEA0. At the same time, read the dirty data and output it in IDED31 to IDED0.

<3> The BCU of the internal NB85E outputs the address (VBA27 to VBA0) of the dirty data to external memory via the VSB and writes one line (4 words) of the dirty data to be replaced to external memory.

<4> It then outputs the address to be read (VBA27 to VBA0) to external memory and refills the data cache with 4 words of data of that address.

<5> The data cache then transfers the required data among the 4 words of refill data to the CPU by passing through IRAMZ31 to IRAMZ0.

Figure 2-18. Operation on Data Cache Miss (Writeback Mode, Write Allocate Enabled, Read, Dirty Data)



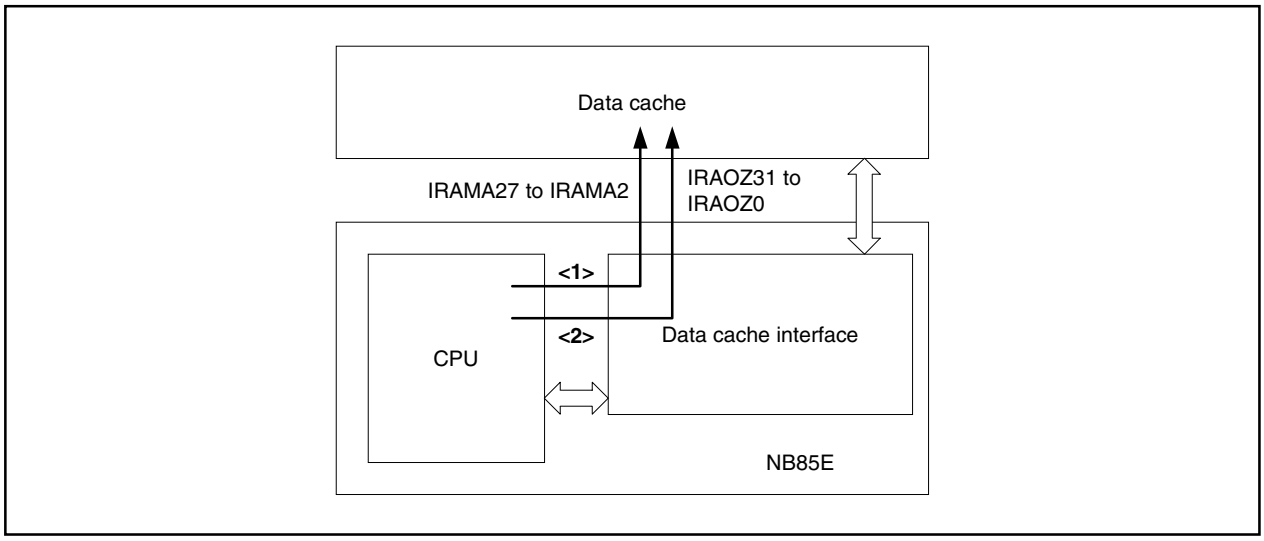
(2) On a write

(a) Data cache hit

<1> When writing data to external memory, output the address (IRAMA27 to IRAMA2) to the data cache.

<2> If a hit occurs due to the external memory address to be written existing in the data cache, write the data to the data cache by passing through IRAOZ31 to IRAOZ0.

Figure 2-19. Operation on Data Cache Hit (Writeback Mode, Write Allocate Enabled, Write)



(b) Data cache miss

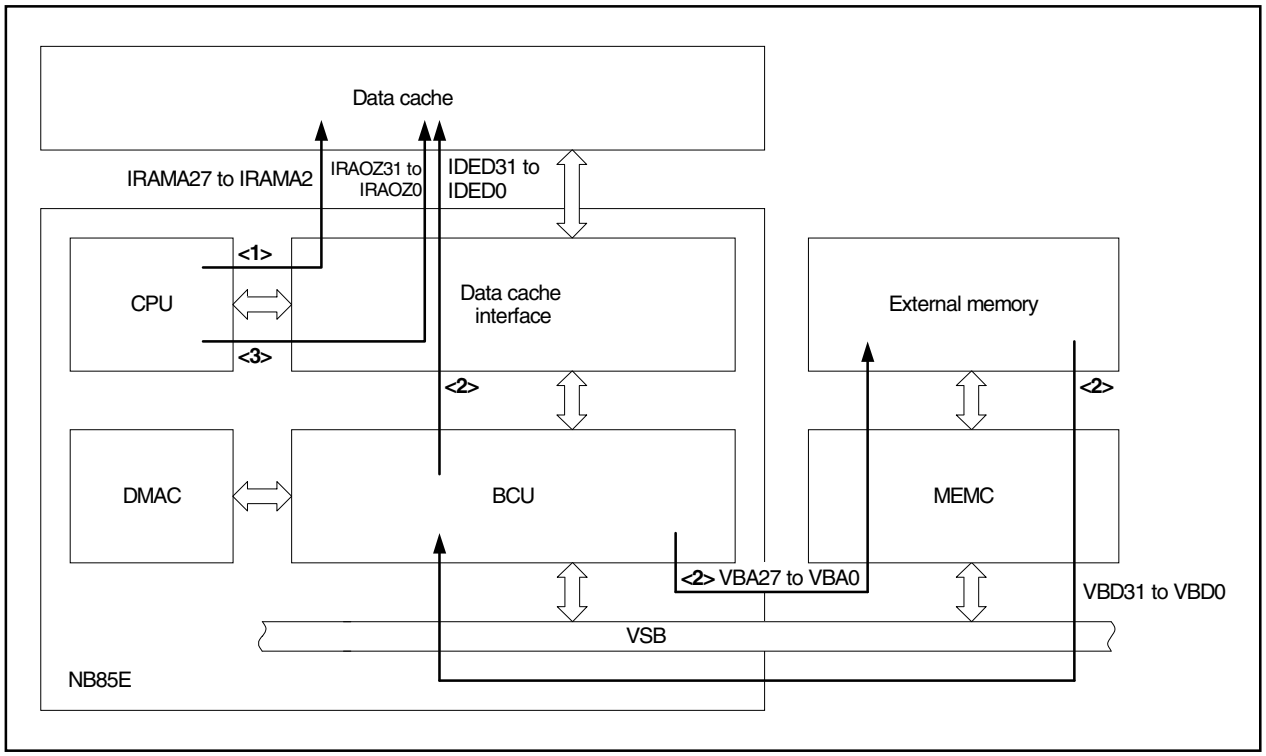
(i) When data being replaced is clean data

<1> When writing data to external memory, output the address (IRAMA27 to IRAMA2) to the data cache. If a miss occurs due to that address not existing in the data cache, perform tag and data replacement. If the data being replaced is clean data, a write operation to external memory is not performed at this time.

<2> The BCU of the internal NB85E outputs the address to be written (VBA27 to VBA0) to external memory via the VSB and refills the data cache with one line (4 words) of data of that address.

<3> Of the 4 words of refilled data, write the data of the address to be written to the data cache.

Figure 2-20. Operation on Data Cache Miss (Writeback Mode, Write Allocate Enabled, Write, Clean Data)



(ii) When data being replaced is dirty data

<1> When writing data to external memory, output the address (IRAMA27 to IRAMA2) to the data cache. If a miss occurs due to that address not existing in the data cache, perform tag and data replacement.

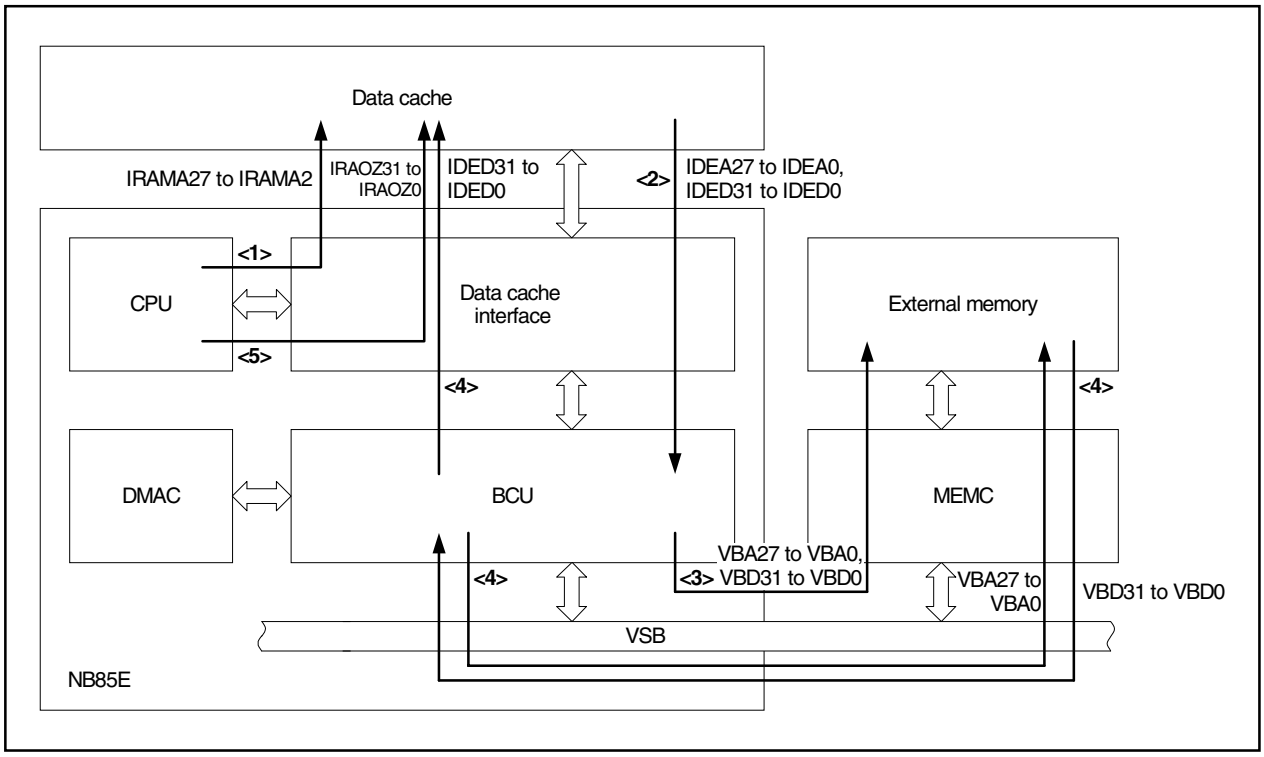
<2> If the data being replaced is dirty data, read the address corresponding to the line that has the dirty data from the tag and output it in IDEA27 to IDEA0. At the same time, read the dirty data and output it in IDED31 to IDED0.

<3> The BCU in the internal NB85E outputs the address (VBA27 to VBA0) of the dirty data to external memory via the VSB and writes one line (4 words) of the dirty data to be replaced to external memory.

<4> It then outputs the address to be written (VBA27 to VBA0) to external memory and refills the data cache with 4 words of data of that address.

<5> Of the 4 words of refilled data, write the data of the address to be written to the data cache.

Figure 2-21. Operation on Data Cache Miss (Writeback Mode, Write Allocate Enabled, Write, Dirty Data)



2.7 Bus Cycles Issued by Data Cache

The data cache issues the bus cycles shown in Table 2-3 depending on the operating mode.

Figures 2-22 to 2-27 show timing examples in the case of a 32-bit data bus and a 16-bit data bus for each operating mode.

The bus cycles indicated in Figures 2-22 to 2-27 (a) 32-bit data bus are 4 times greater when an 8-bit data bus is used as a result of bus sizing.

Remarks 1. The timing example is when no waits are used.

2. All signals in the timing example are NB85E signals.

3. The broken-line levels of the VBTTYP1, VBTTYP0, VBD31 to VBD0, VBWAIT, VBAHLD, and VBLAST signals indicate the undefined state (weak unknown) driven by the bus holder in the NB85E.

4. The circles indicate the sampling timing.

5. For details on the VSB signals (VBxxx, VDxxx), refer to the **NB85E Hardware User's Manual (A13971E)**.

Table 2-3. Operating Modes and Bus Cycles

Operating Mode	Access Type			Refill Mode	Bus Cycle	Reference
WT	Read	Hit		–	None	–
		Miss		Sequential	4R	Figure 2-22
				Critical first	4R (IRAMA3, IRAMA2 = 00)	–
					2R-2R (IRAMA3, IRAMA2 = 10)	Figure 2-23
					1R-2R-1R (IRAMA2 = 1)	Figure 2-24
	Write	Hit		–	1W	–
		Miss		–	1W	–
WB	Read	Hit		–	None	–
		Miss	Clean data	Sequential	4R	Figure 2-22
				Critical first	4R (IRAMA3, IRAMA2 = 00)	–
					2R-2R (IRAMA3, IRAMA2 = 10)	Figure 2-23
					1R-2R-1R (IRAMA2 = 1)	Figure 2-24
			Dirty data	Sequential	4W + 4R	Figure 2-25
				Critical first	4W + 4R (IRAMA3, IRAMA2 = 00)	–
					4W + 2R-2R (IRAMA3, IRAMA2 = 10)	Figure 2-26
					4W + 1R-2R-1R (IRAMA2 = 1)	Figure 2-27
	Write	Hit		–	None	–
		Miss		–	1W	–
WA	Read/Write	Hit		–	None	–
		Miss	Clean data	Sequential	4R	Figure 2-22
				Critical first	4R (IRAMA3, IRAMA2 = 00)	–
					2R-2R (IRAMA3, IRAMA2 = 10)	Figure 2-23
					1R-2R-1R (IRAMA2 = 1)	Figure 2-24
			Dirty data	Sequential	4W + 4R	Figure 2-25
				Critical first	4W + 4R (IRAMA3, IRAMA2 = 00)	–
					4W + 2R-2R (IRAMA3, IRAMA2 = 10)	Figure 2-26
					4W + 1R-2R-1R (IRAMA2 = 1)	Figure 2-27
Lock	Read	Hit		–	None	–
		Miss		–	1R (word access)	–
	Write	Hit (WT)		–	1W	–
		Hit (WB, WA)		–	None	–
		Miss		–	1W	–

Remarks 1. The meanings of the items in Operating Mode and Access Type columns are as follows.

WT: Write through mode

WB: Writeback mode (write allocate disabled)

WA: Writeback mode (write allocate enabled)

2. The meanings of the items in the Bus Cycle column are as follows.

1: Single transfer

2: 2-word burst

4: 4-word burst

R: Read

W: Write

Figure 2-22. Sequential Refill Read Cycle (4R) (1/2)

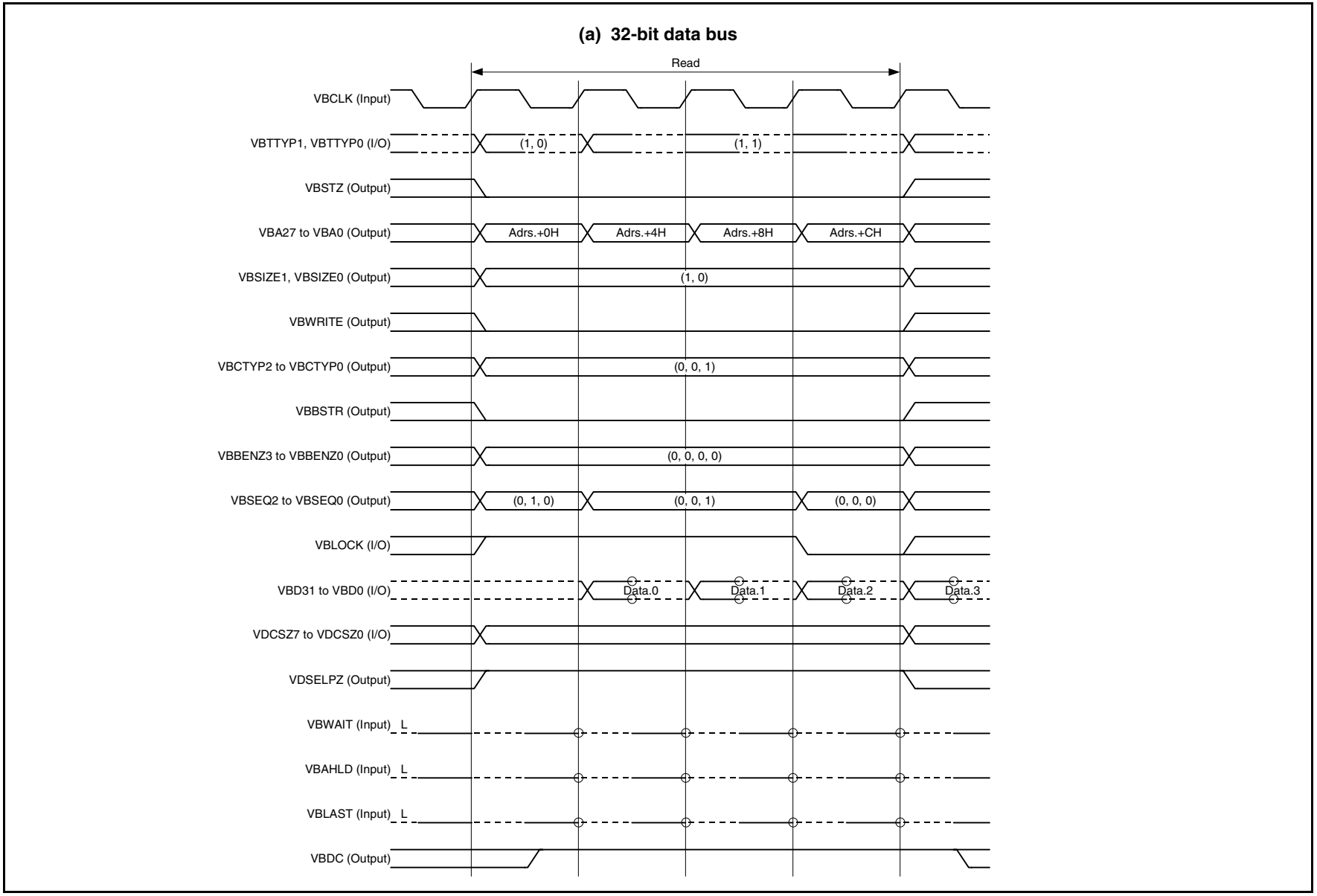


Figure 2-22. Sequential Refill Read Cycle (4R) (2/2)

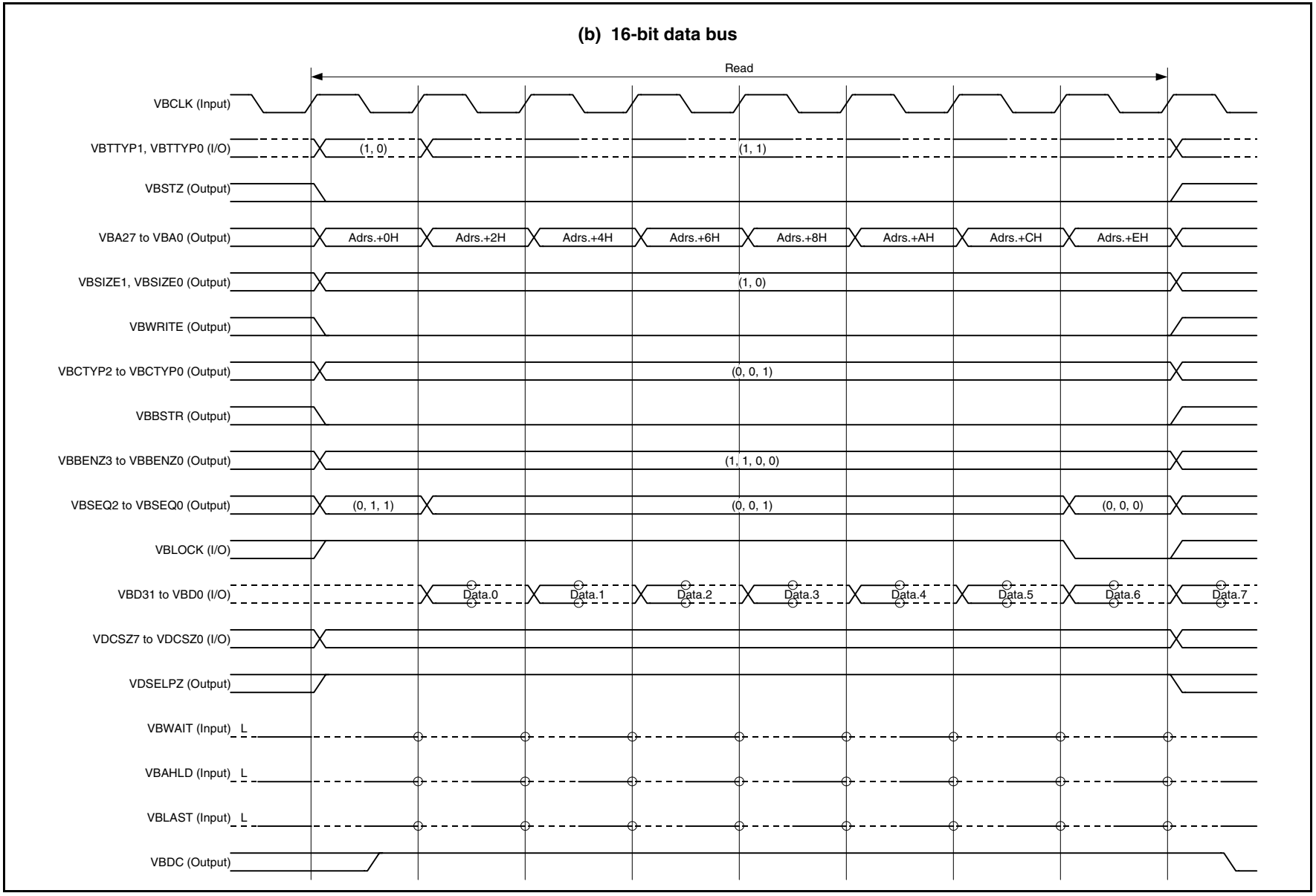


Figure 2-23. Critical First Refill Read Cycle (2R-2R) (1/2)

(a) 32-bit data bus

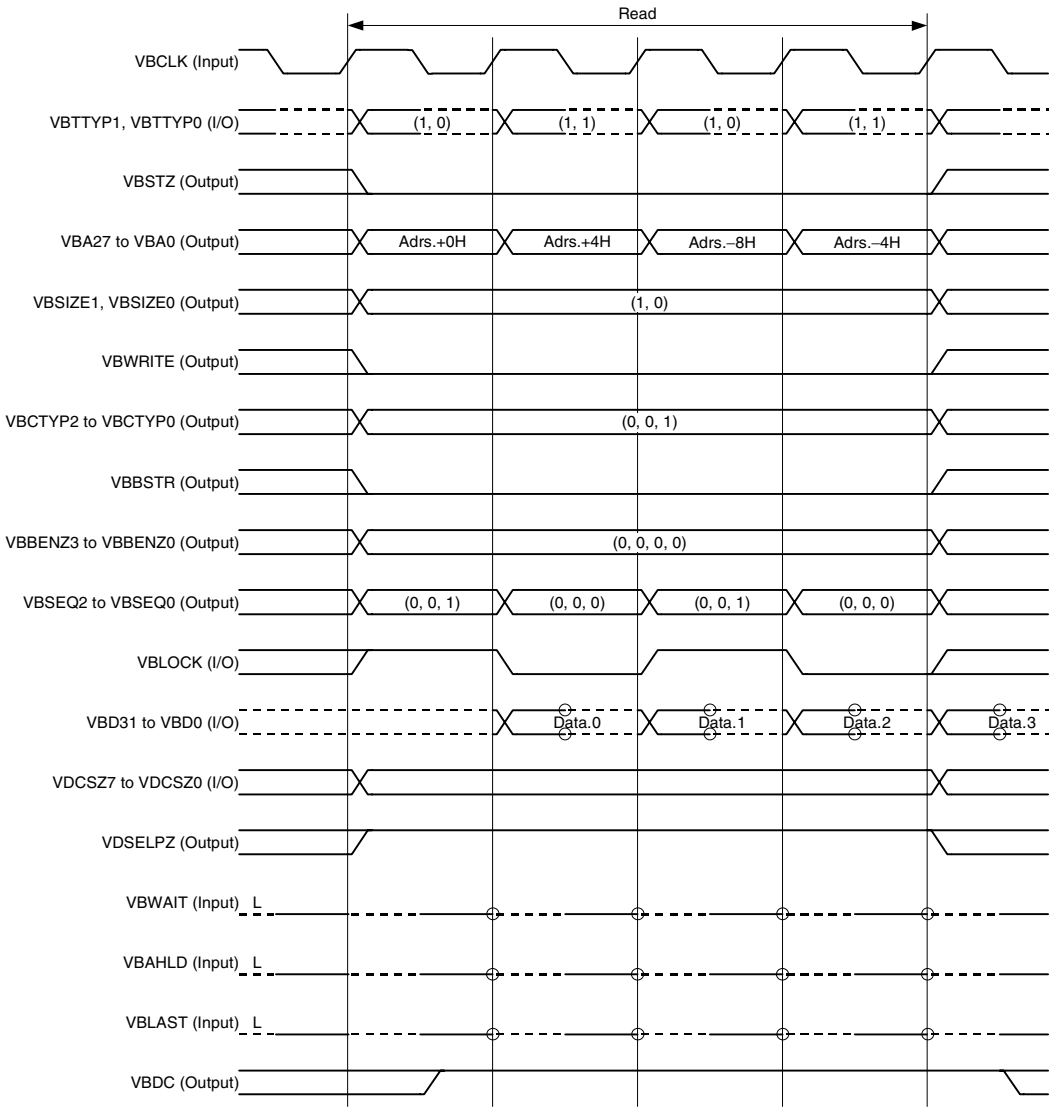


Figure 2-23. Critical First Refill Read Cycle (2R-2R) (2/2)

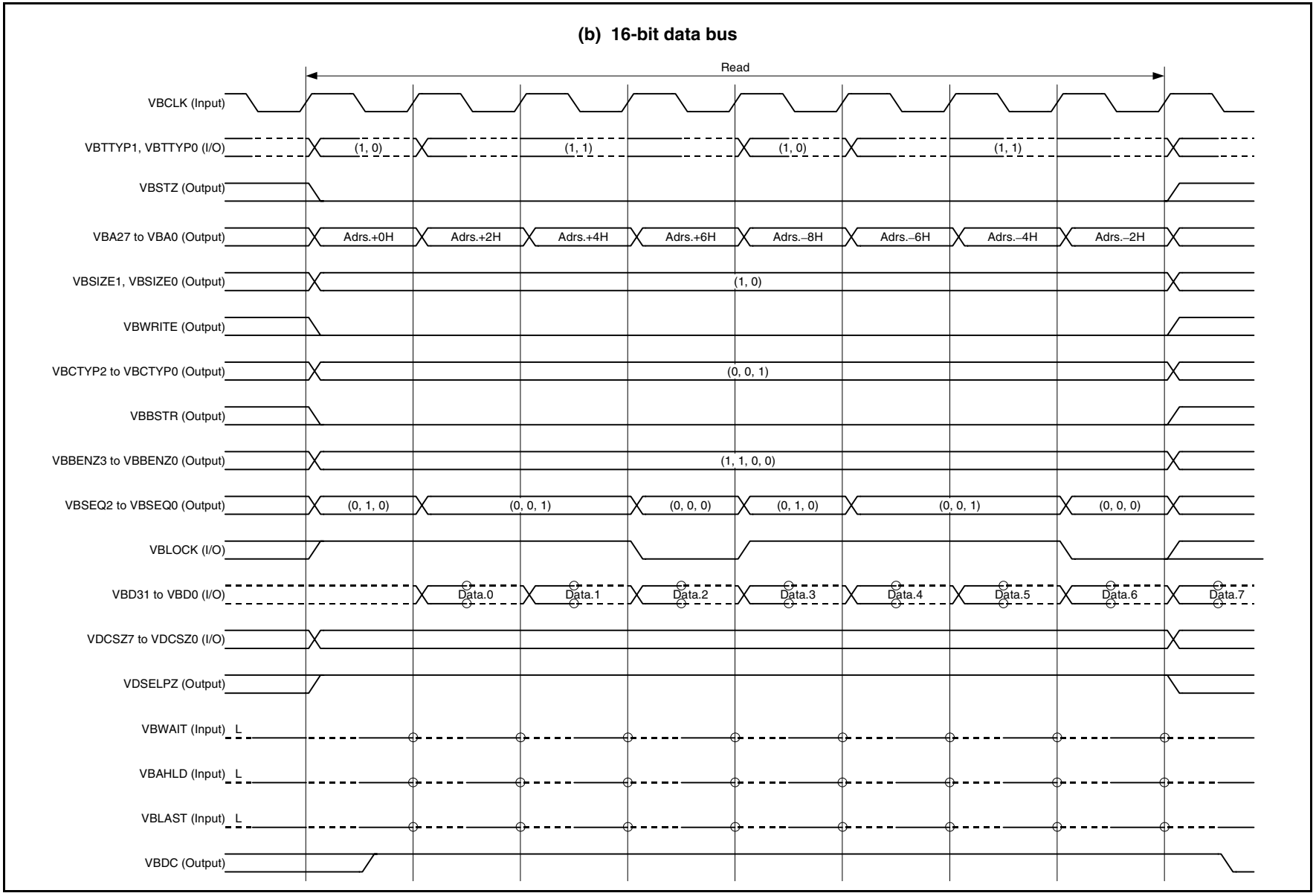


Figure 2-24. Critical First Refill Read Cycle (1R-2R-1R) (1/4)

(a) 32-bit data bus (IRAMA3, IRAMA2 = 01)

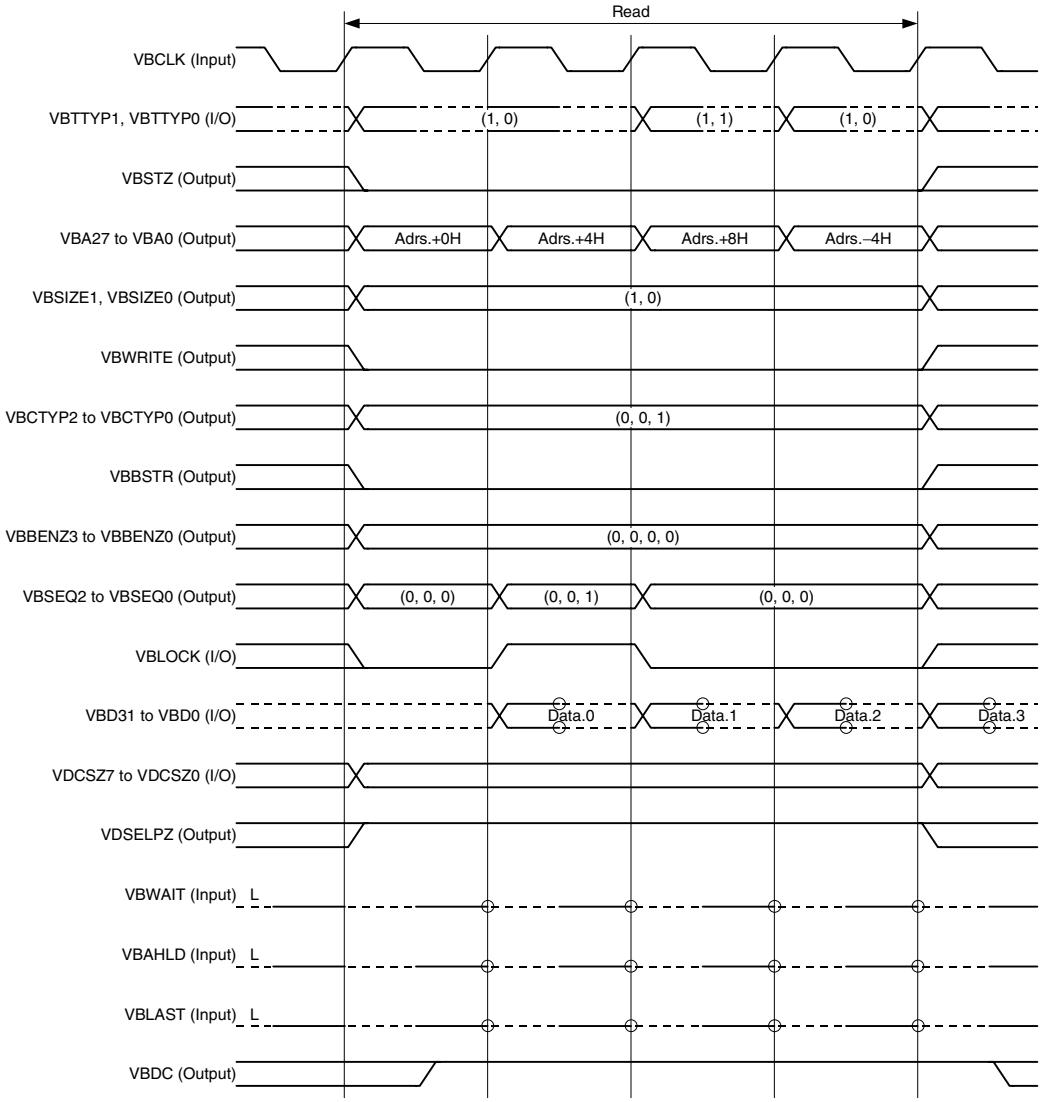


Figure 2-24. Critical First Refill Read Cycle (1R-2R-1R) (2/4)

(b) 32-bit data bus (IRAMA3, IRAMA2 = 11)

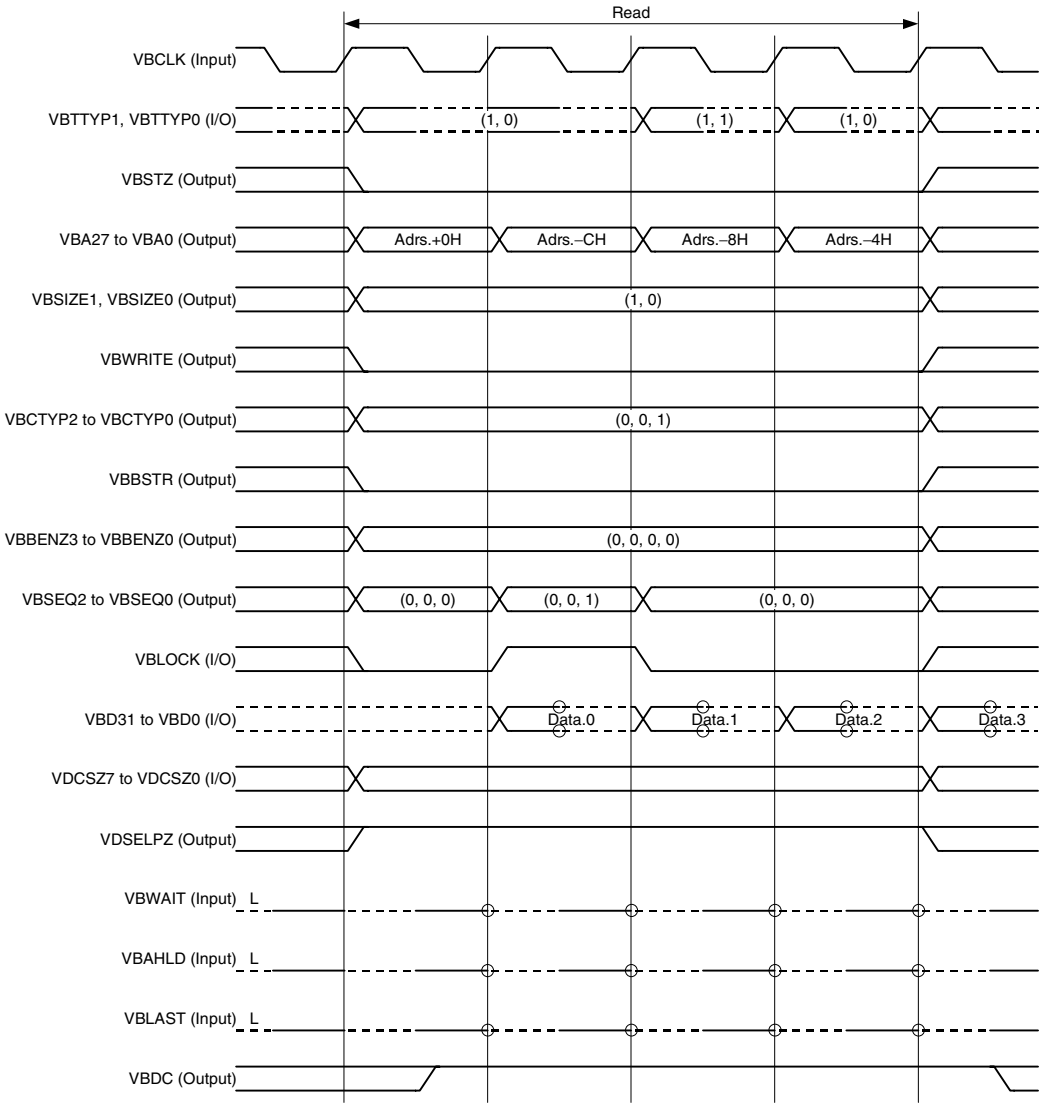


Figure 2-24. Critical First Refill Read Cycle (1R-2R-1R) (3/4)

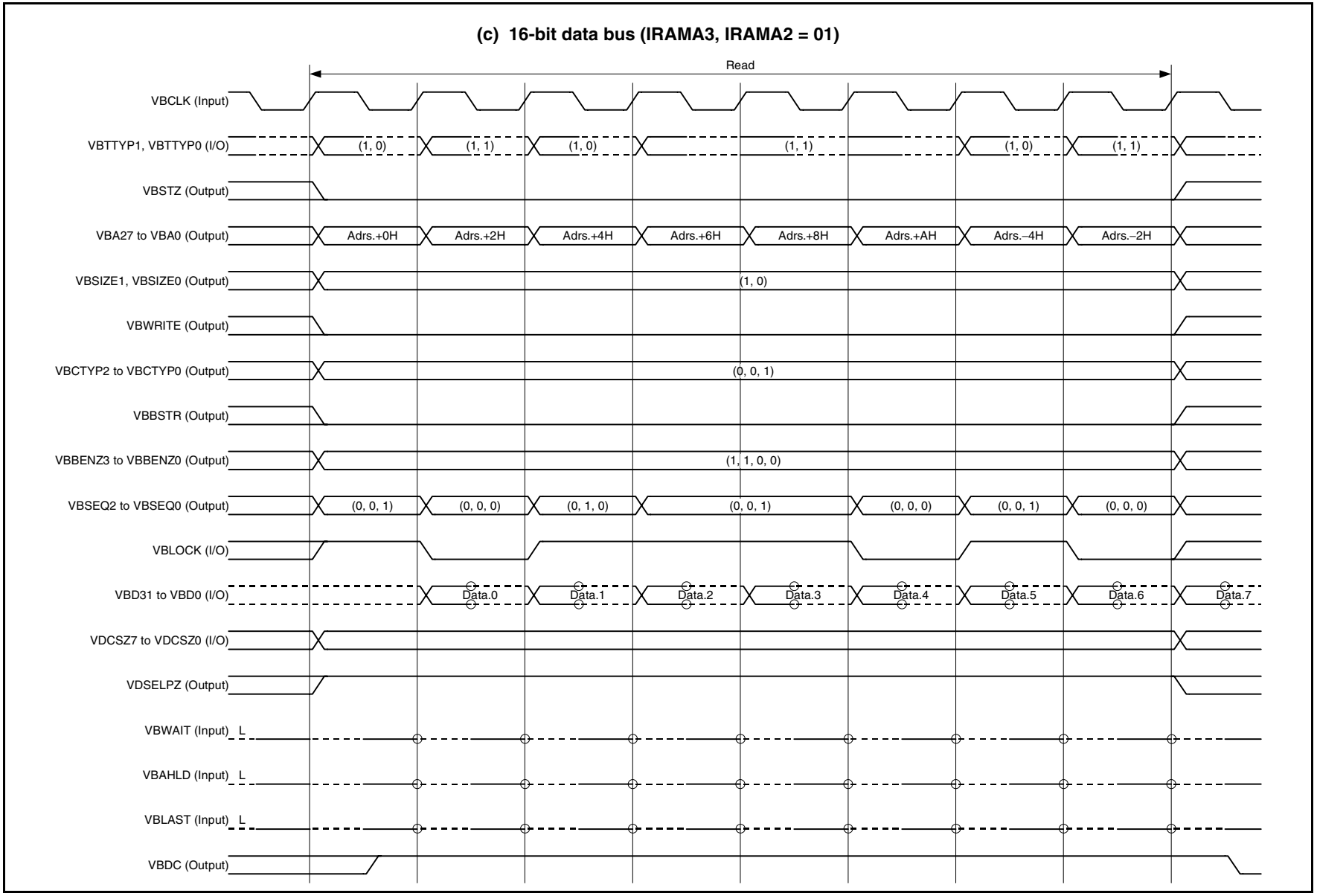


Figure 2-24. Critical First Refill Read Cycle (1R-2R-1R) (4/4)

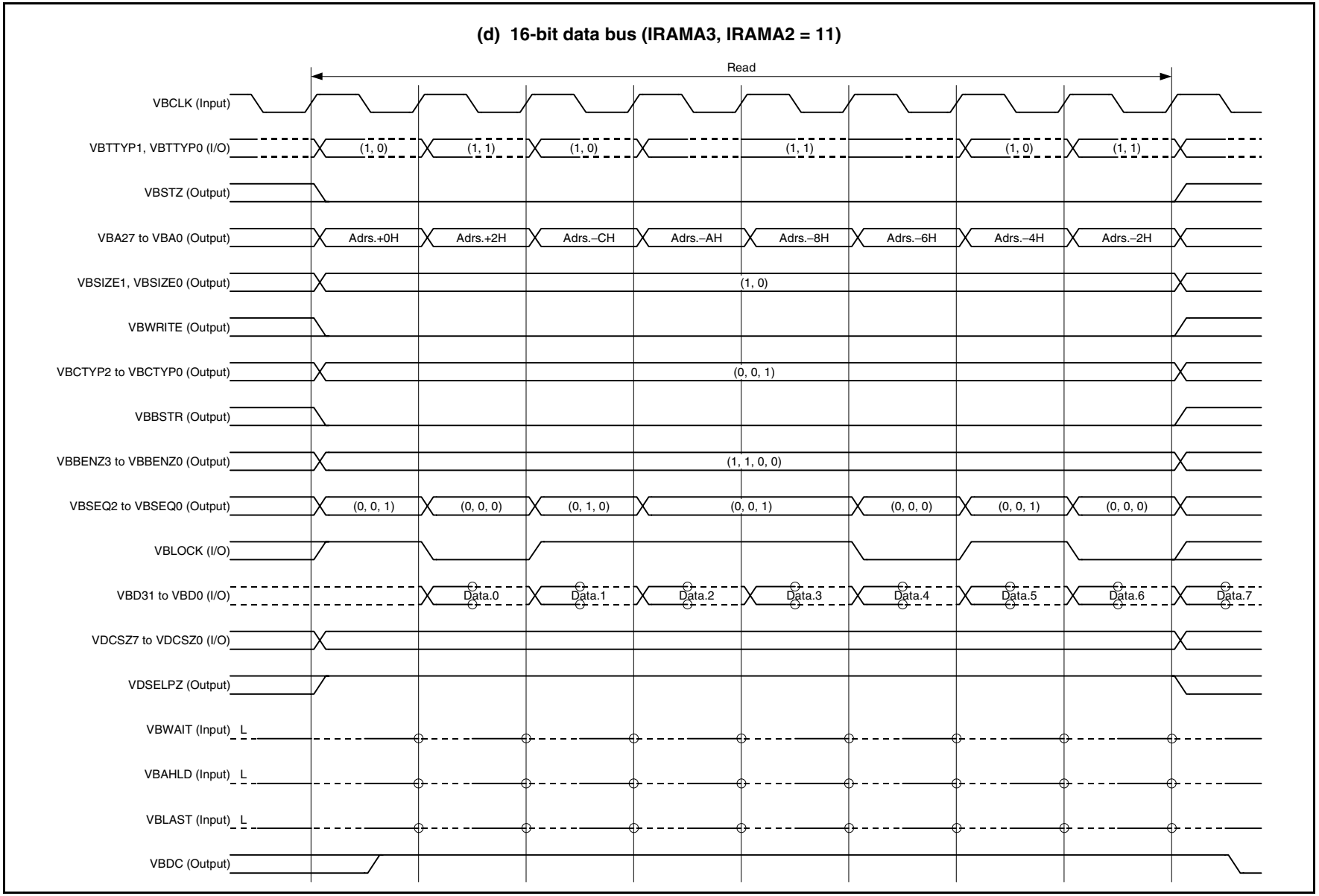


Figure 2-25. Sequential Refill Read Cycle in Writeback Mode (When Data Being Replaced Is Dirty Data) (4W + 4R) (1/2)

(a) 32-bit data bus

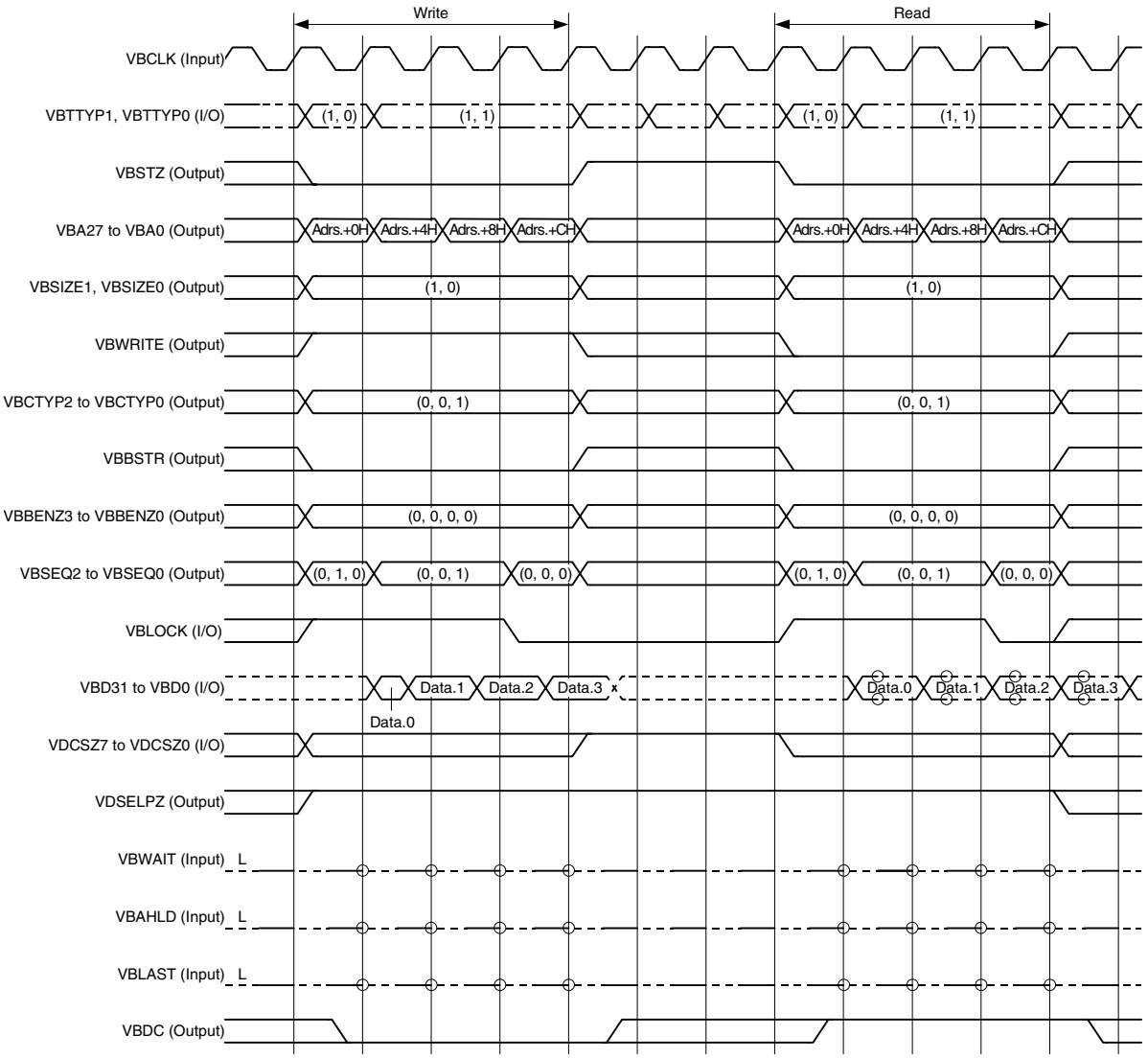


Figure 2-25. Sequential Refill Read Cycle in Writeback Mode (When Data Being Replaced Is Dirty Data) (4W + 4R) (2/2)

(b) 16-bit data bus

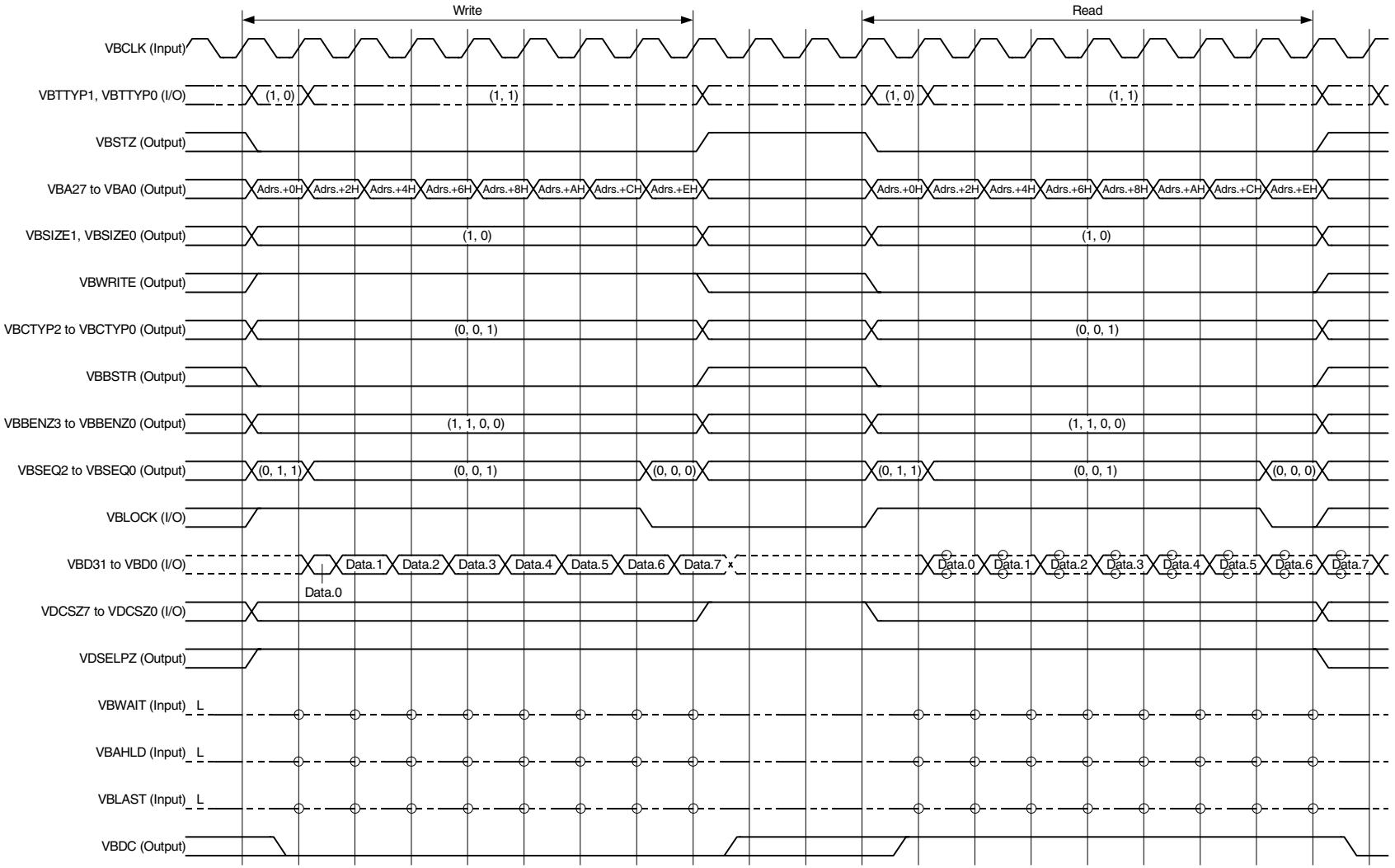


Figure 2-26. Critical First Refill Read Cycle in Writeback Mode (When Data Being Replaced Is Dirty Data) (4W + 2R-2R) (1/2)

(a) 32-bit data bus

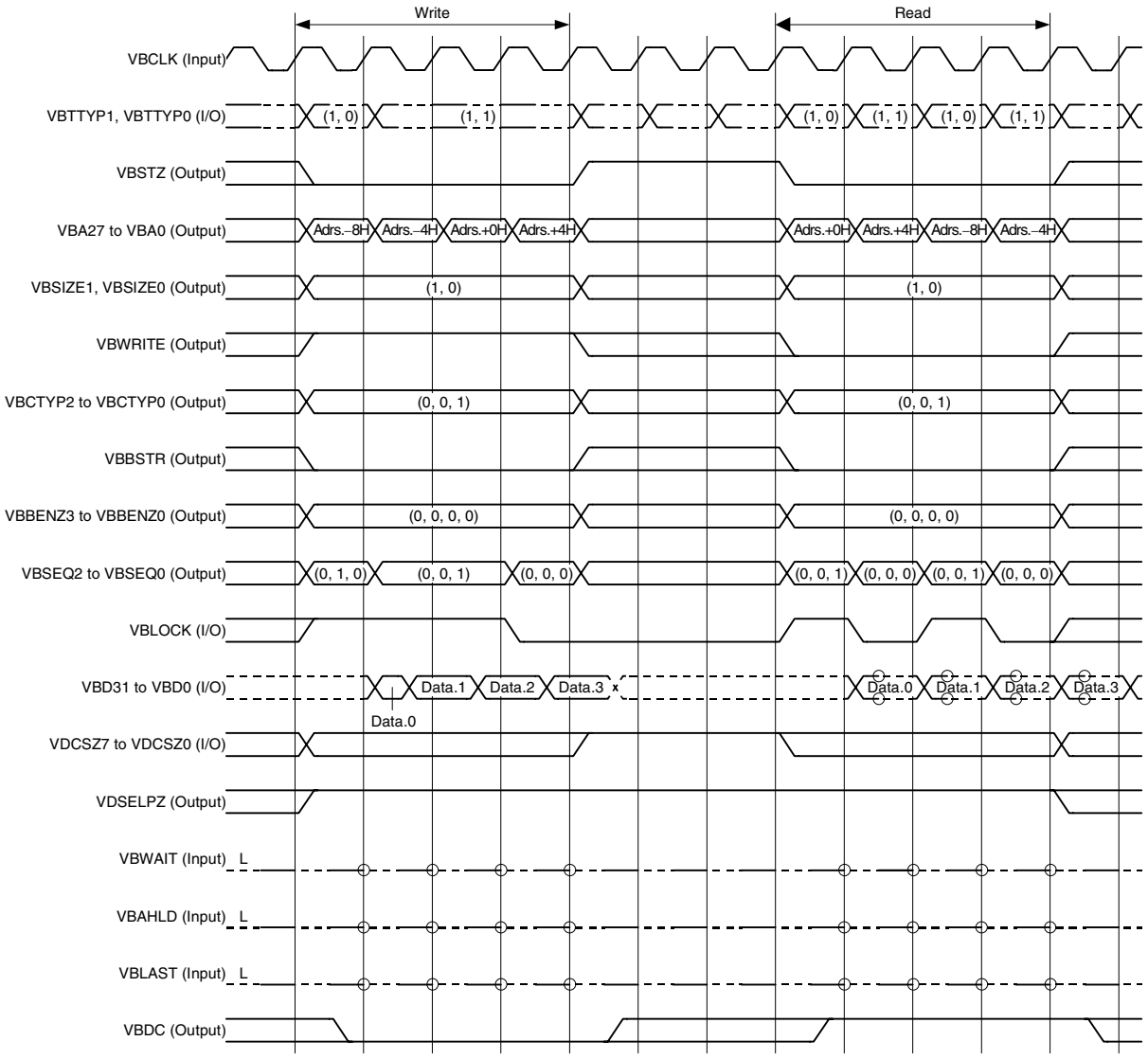


Figure 2-26. Critical First Refill Read Cycle in Writeback Mode (When Data Being Replaced Is Dirty Data) (4W + 2R-2R) (2/2)

(b) 16-bit data bus

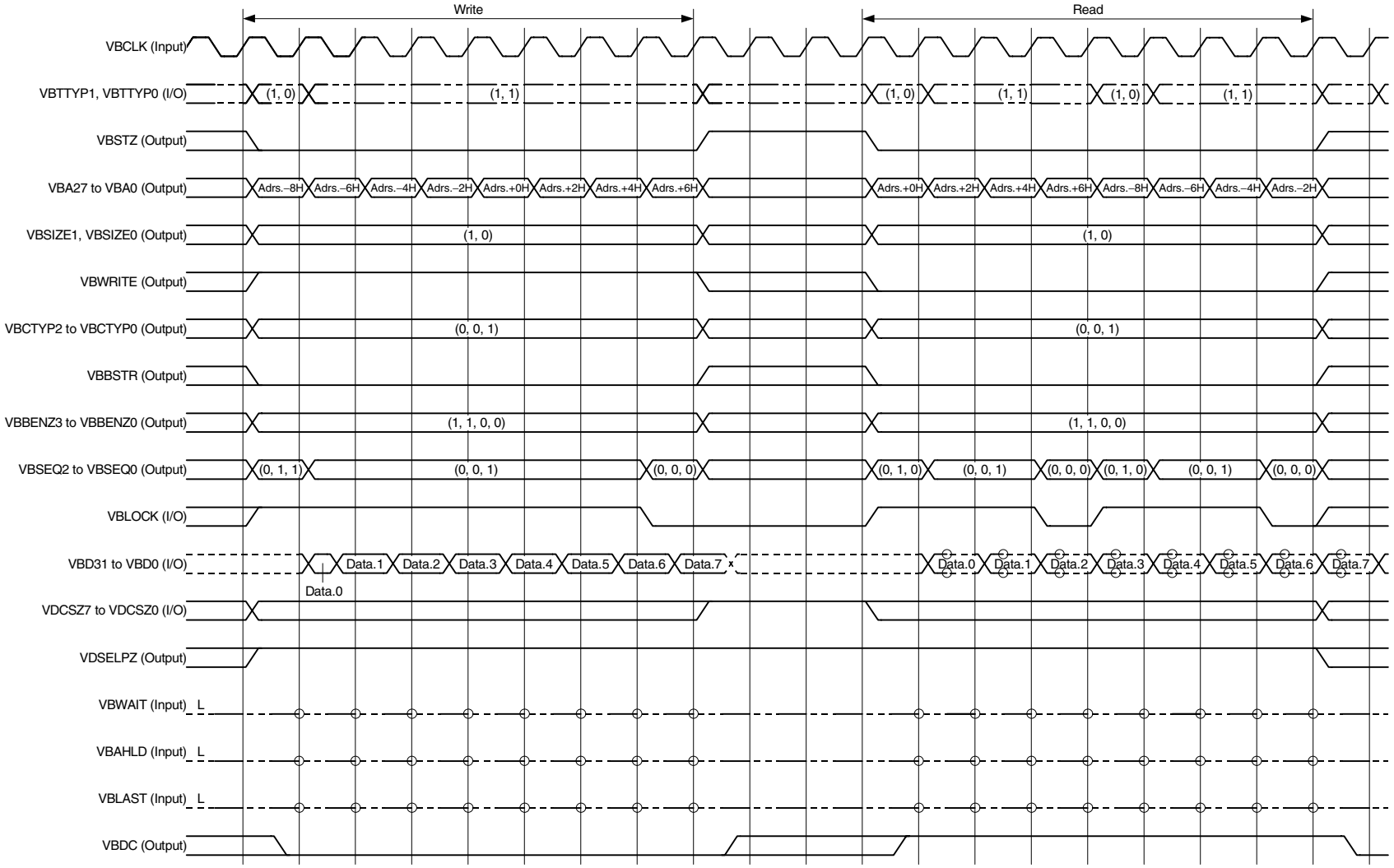


Figure 2-27. Critical First Refill Read Cycle in Writeback Mode (When Data Being Replaced Is Dirty Data) (4W + 1R-2R-1R) (1/4)

(a) 32-bit data bus (IRAMA3, IRAMA2 = 01)

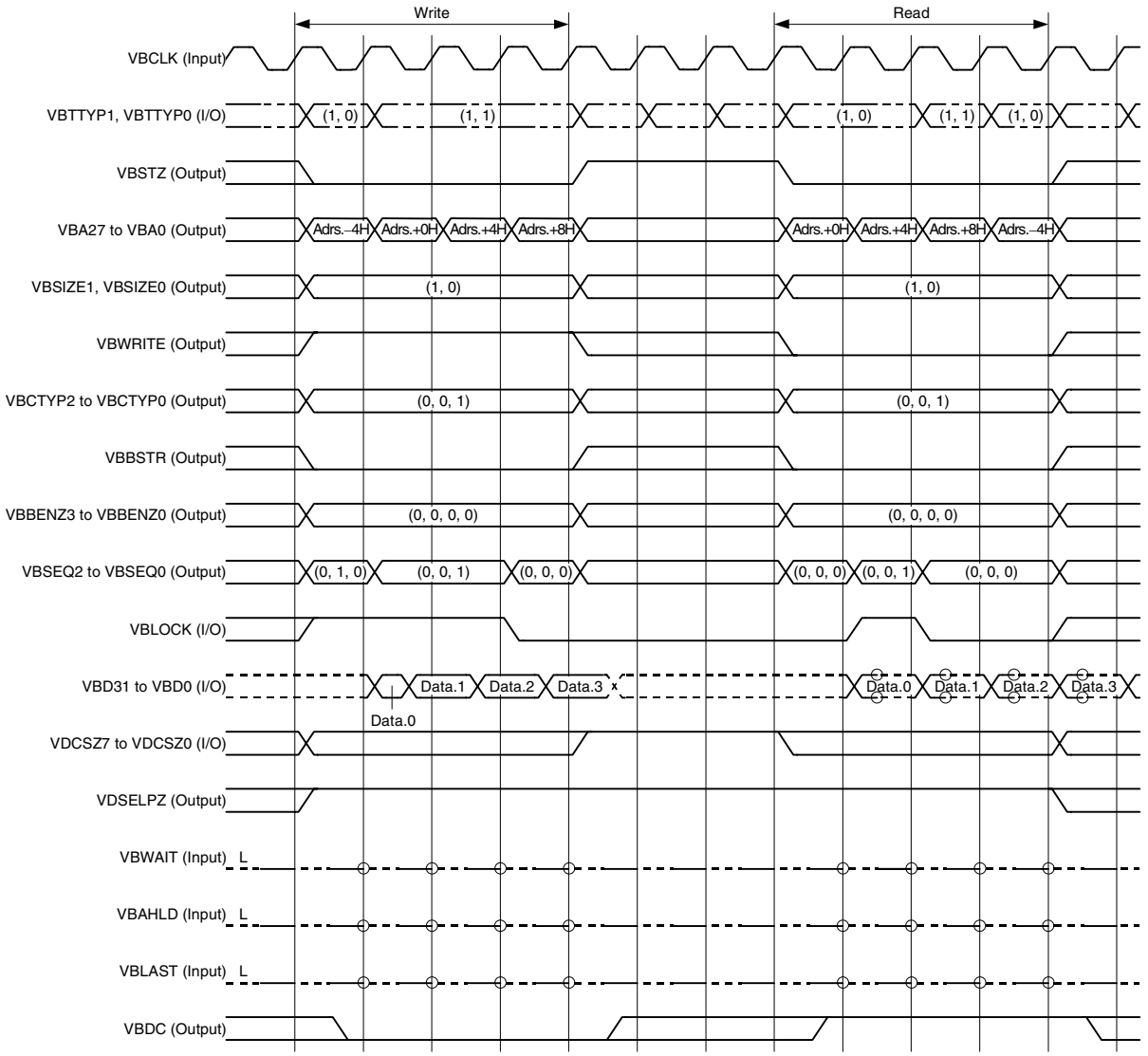


Figure 2-27. Critical First Refill Read Cycle in Writeback Mode (When Data Being Replaced Is Dirty Data) (4W + 1R-2R-1R) (2/4)

(b) 32-bit data bus (IRAMA3, IRAMA2 = 11)

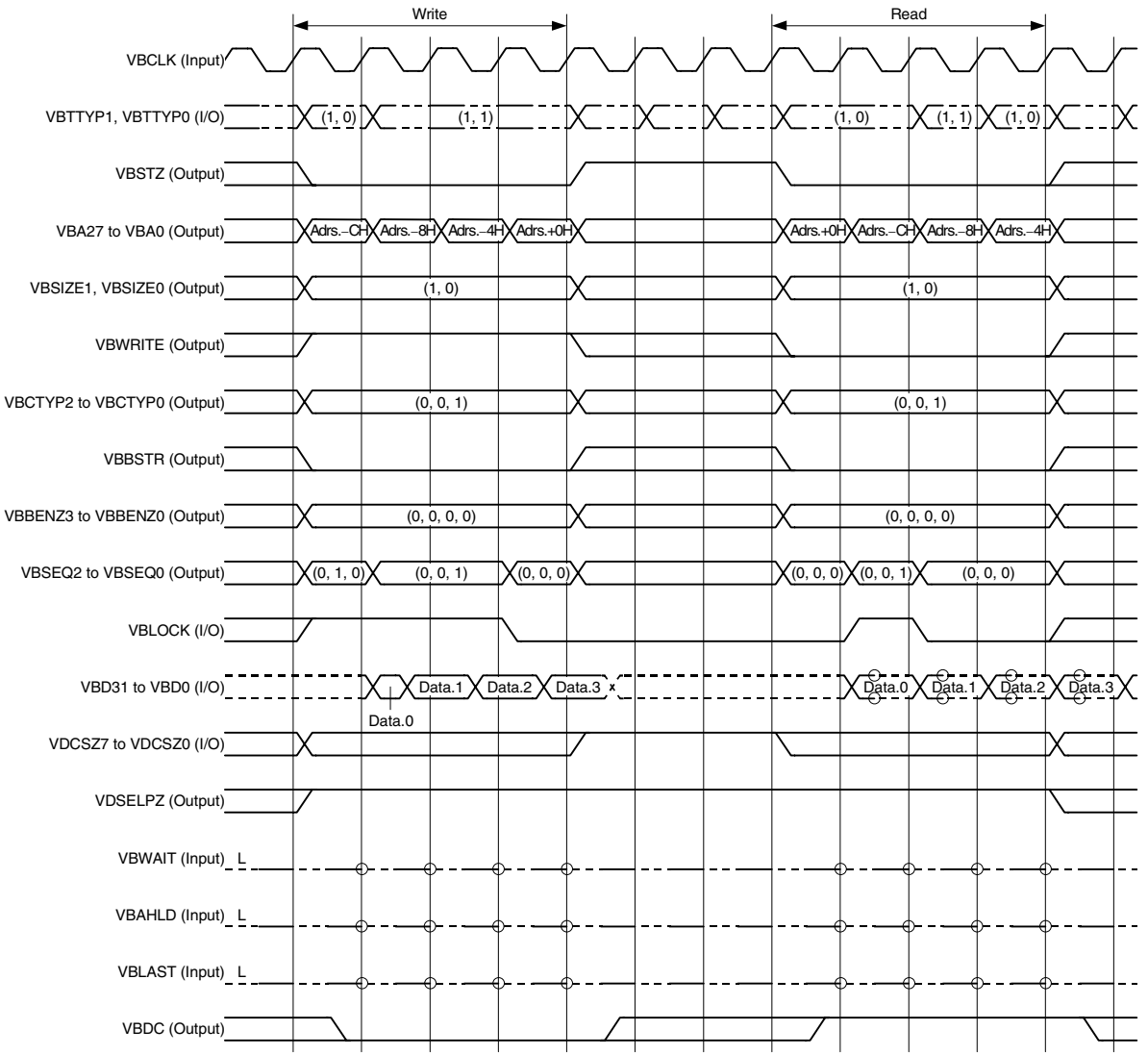


Figure 2-27. Critical First Refill Read Cycle in Writeback Mode (When Data Being Replaced Is Dirty Data) (4W + 1R-2R-1R) (3/4)

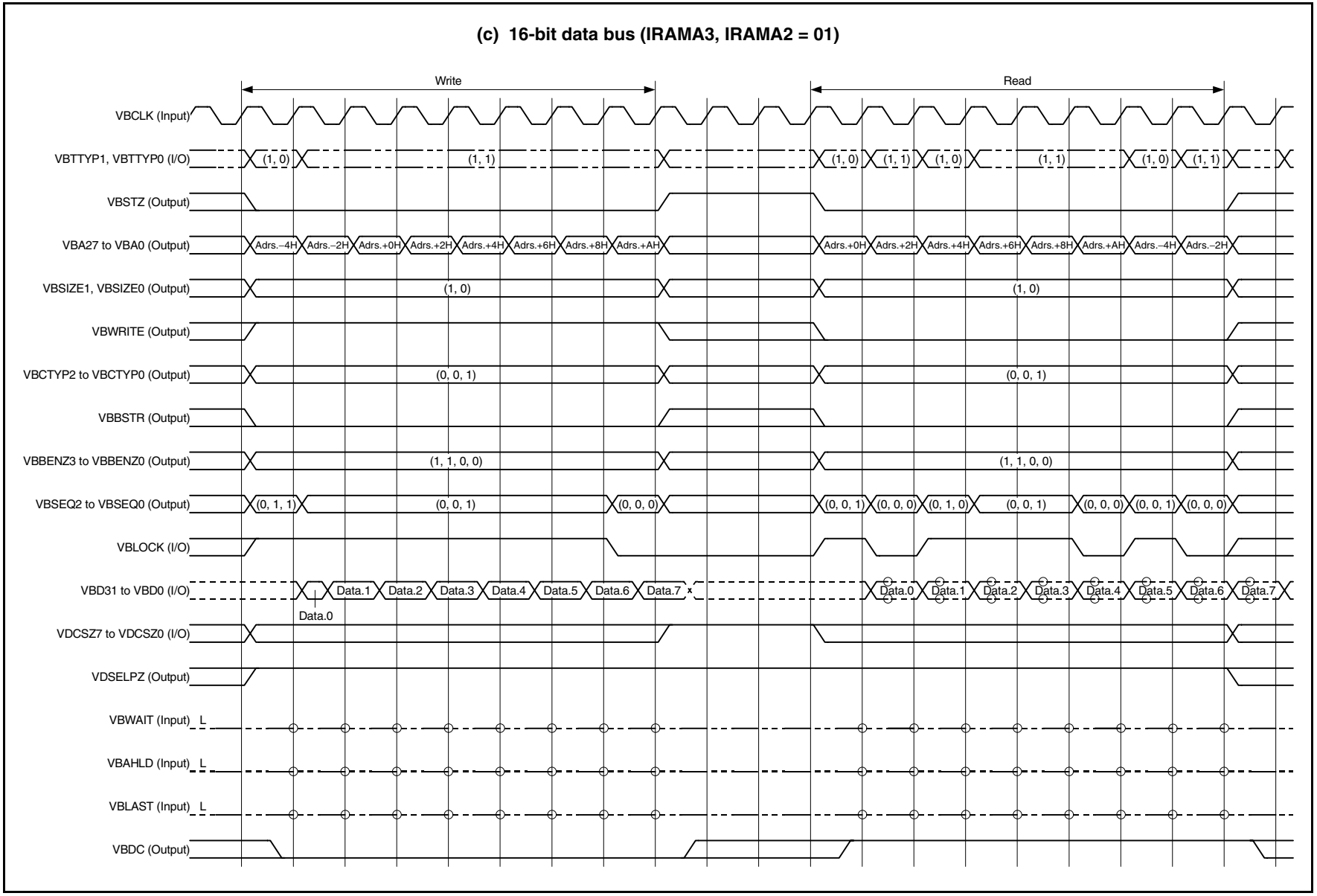
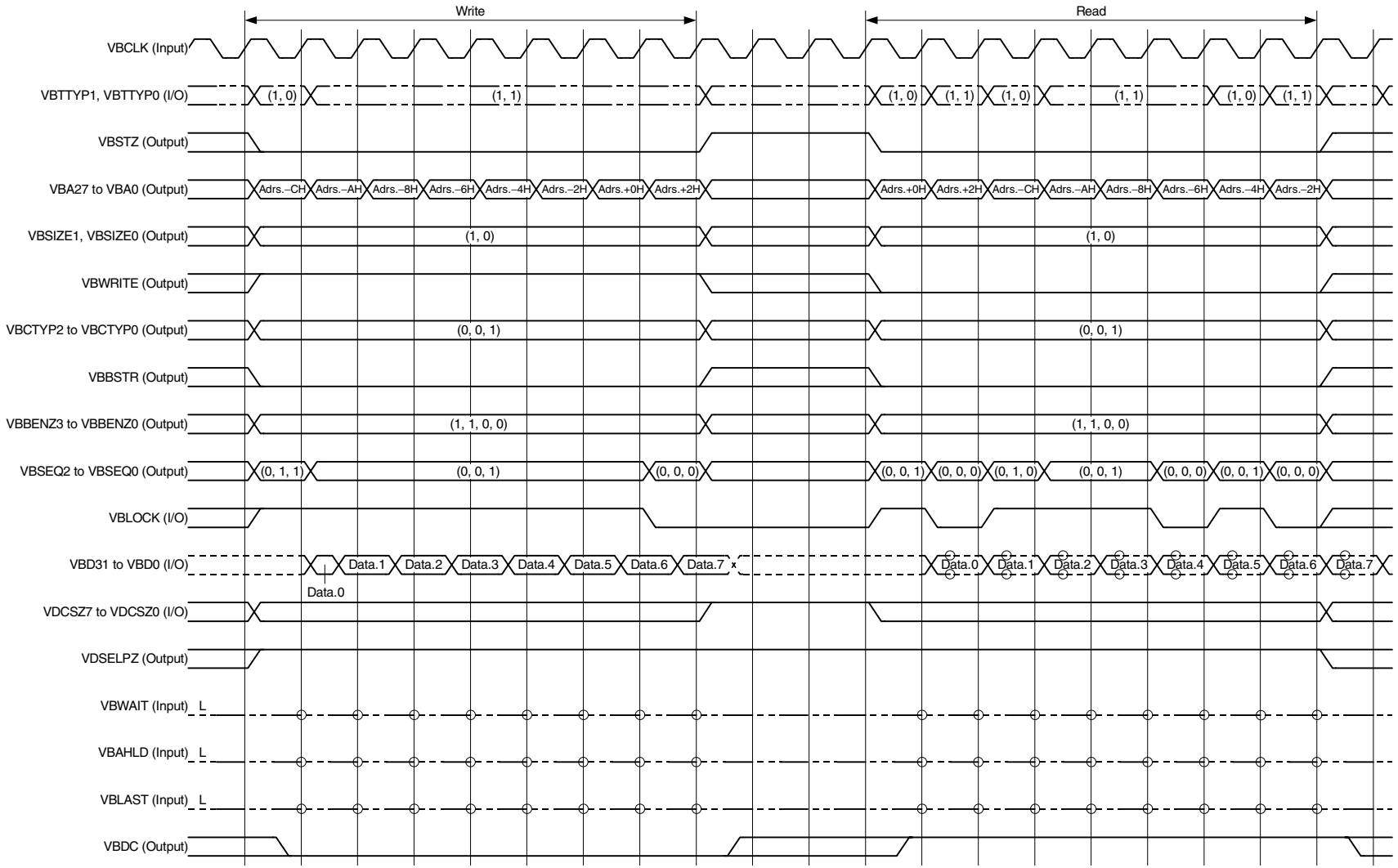


Figure 2-27. Critical First Refill Read Cycle in Writeback Mode (When Data Being Replaced Is Dirty Data) (4W + 1R-2R-1R) (4/4)

(d) 16-bit data bus (IRAMA3, IRAMA2 = 11)



2.8 Timing of Refill from SDRAM to Data Cache

Figures 2-28 to 2-30 show the refill timing examples from SDRAM to the data cache for each refill mode.

Remarks 1. These timing examples assume the following conditions.

- 32-bit data bus
- CAS latency = 2
- Number of wait states set with bits BCW1, BCW0 of SDRAM configuration register n (SCRn) of the SDRAM controller (NU85E502) = 1 (n = 7 to 0)

2. The broken-line levels of the VBD31 to VBD0 signals indicate the undefined state (weak unknown) driven by the bus holder in the NB85E.

3. The circles indicate the sampling timing.

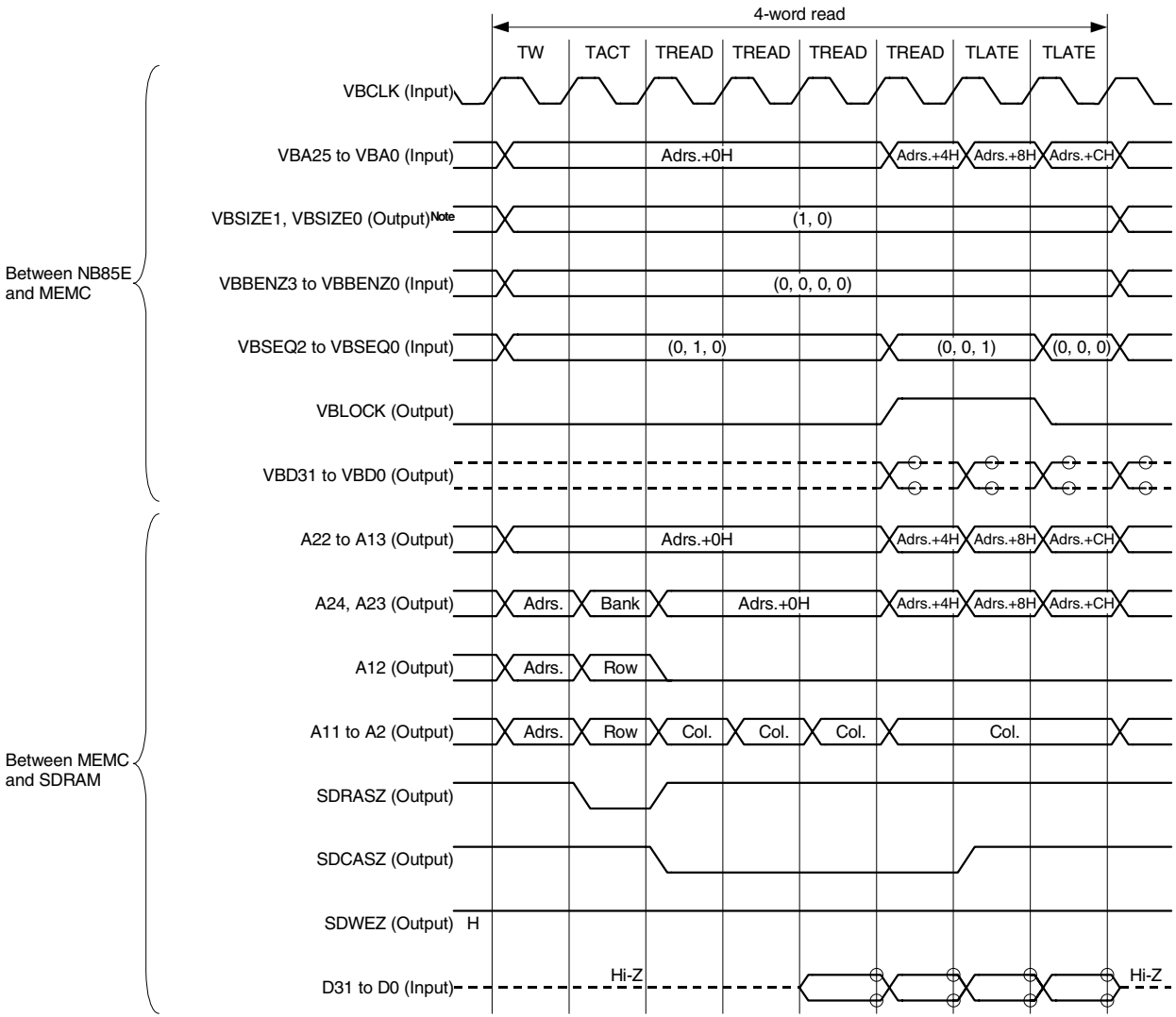
4. For details on the VSB signals (VBxxx), refer to the **NB85E Hardware User's Manual (A13971E)**.

5. The inputs and outputs as seen from the memory controller (MEMC) side are shown.

6. The various state abbreviations have the following meanings.

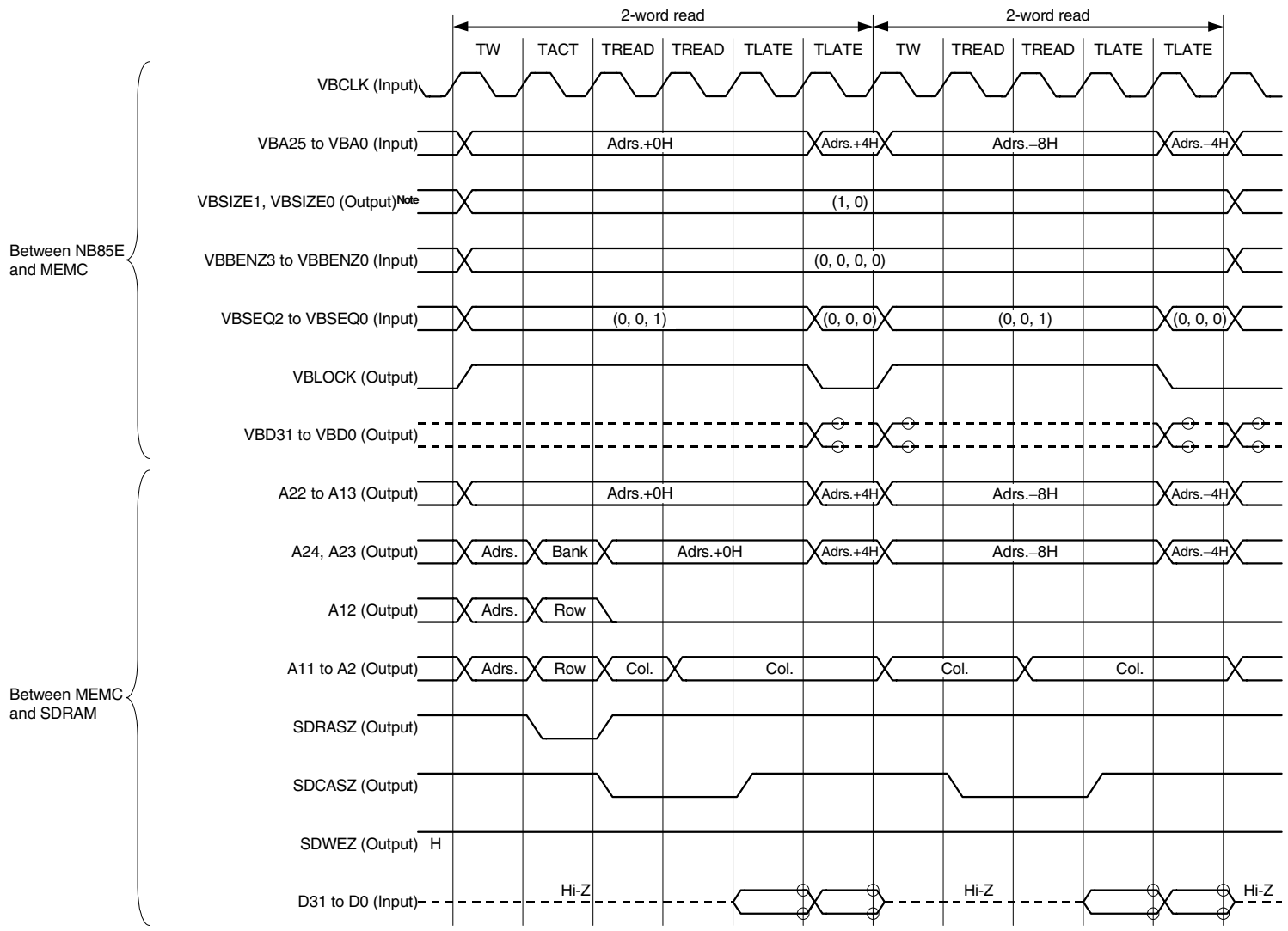
TW: Wait state
TACT: Bank active command state
TREAD: Read command state
TLATE: Latency wait state

Figure 2-28. Refill Timing Example from SDRAM to Data Cache (Sequential Refill (4R), Critical First Refill (4R))



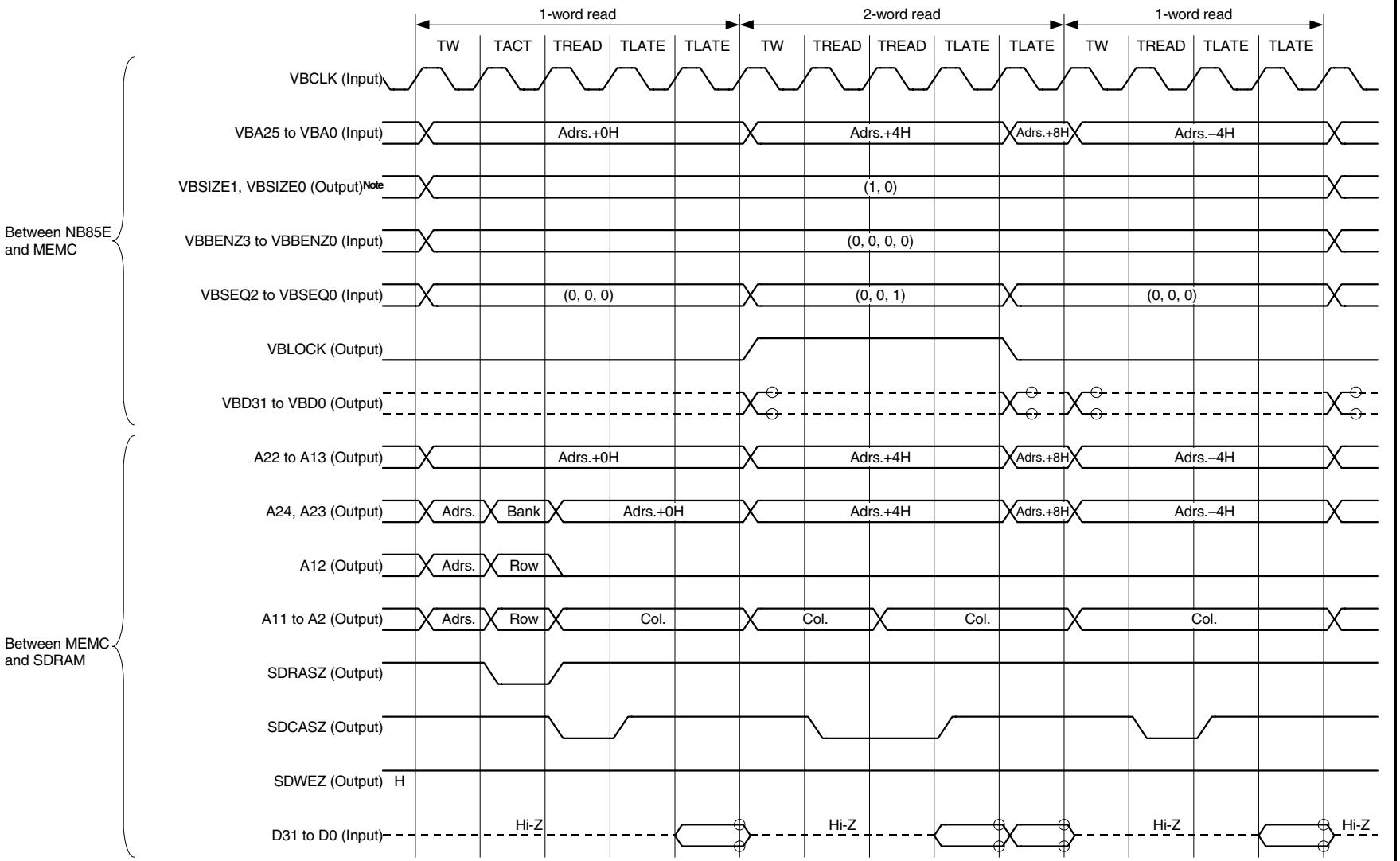
Note NB85E signal. This signal is not connected to MEMC.

Figure 2-29. Refill Timing Example from SDRAM to Data Cache (Critical First Refill (2R-2R))



Note NB85E signal. This signal is not connected to MEMC.

Figure 2-30. Refill Timing Example from SDRAM to Data Cache (Critical First Refill (1R-2R-1R))



Note NB85E signal. This signal is not connected to MEMC.

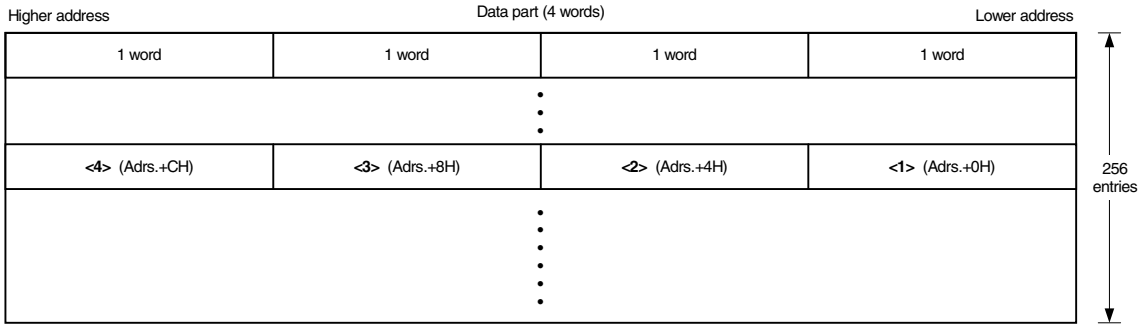
Remark This is an example when IRAMA3, IRAMA2 = 01.

2.9 Refill Sequence to Data Cache

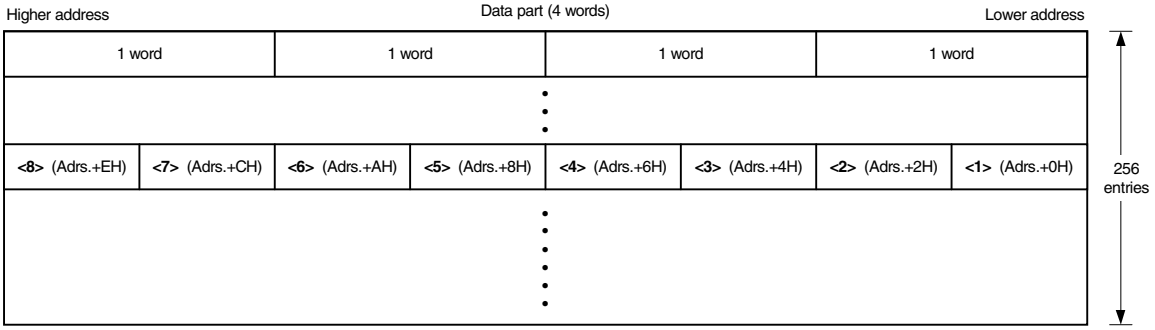
The refill sequence to the data part of a data cache when a miss occurs differs depending on the refill mode. Figures 2-31 to 2-33 show the refill sequence for each refill mode.

Figure 2-31. Refill Sequence to Data Cache (Sequential Refill (4R), Critical First Refill (4R))

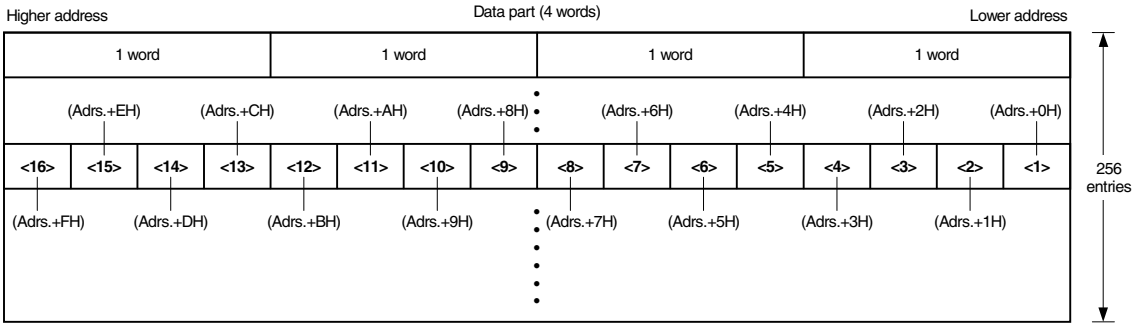
(a) 32-bit data bus



(b) 16-bit data bus



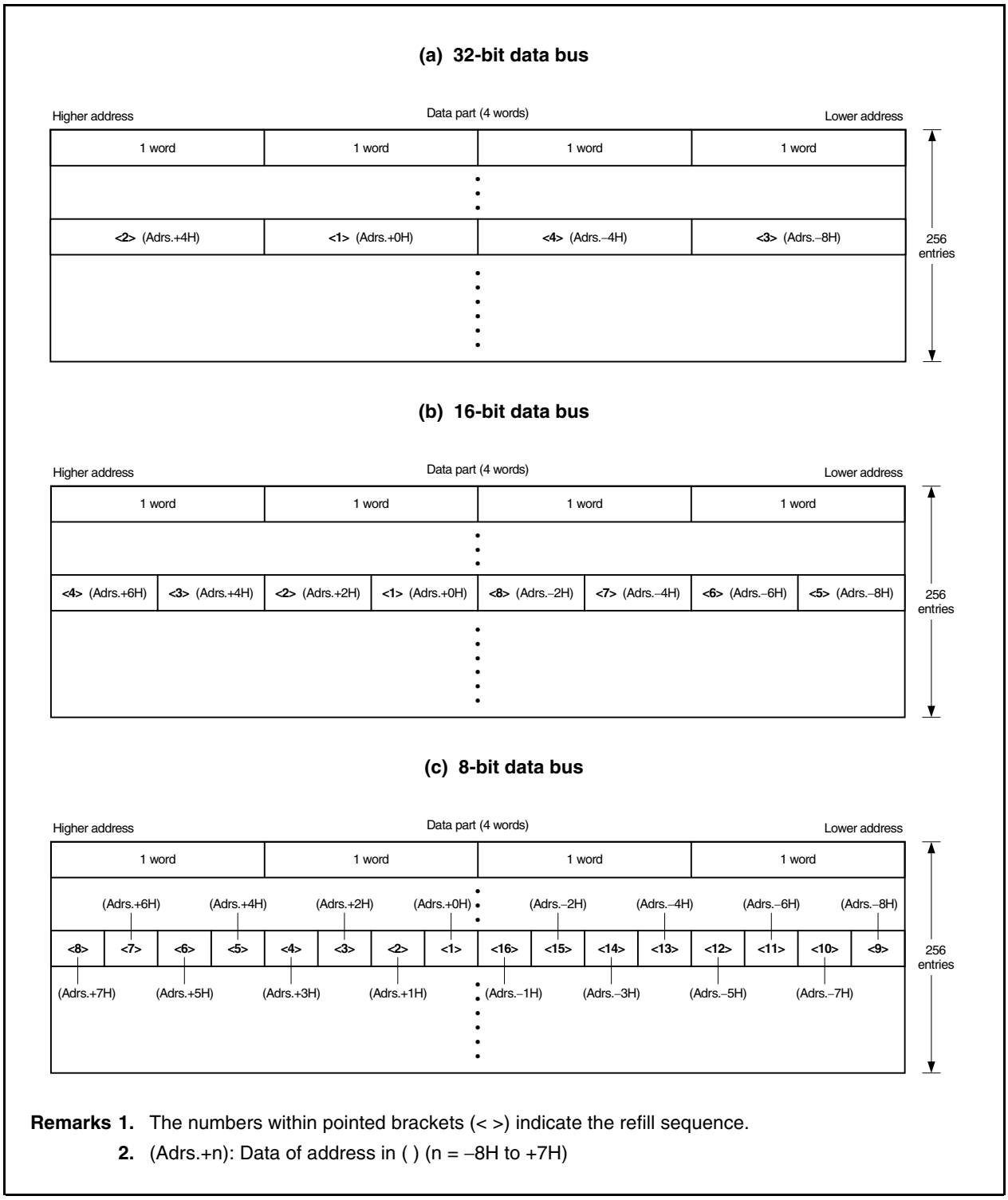
(c) 8-bit data bus



Remarks 1. The numbers within pointed brackets (< >) indicate the refill sequence.

2. (Adrs.+n): Data of address in ( ) (n = 0H to FH)

Figure 2-32. Refill Sequence to Data Cache (Critical First Refill (2R-2R))



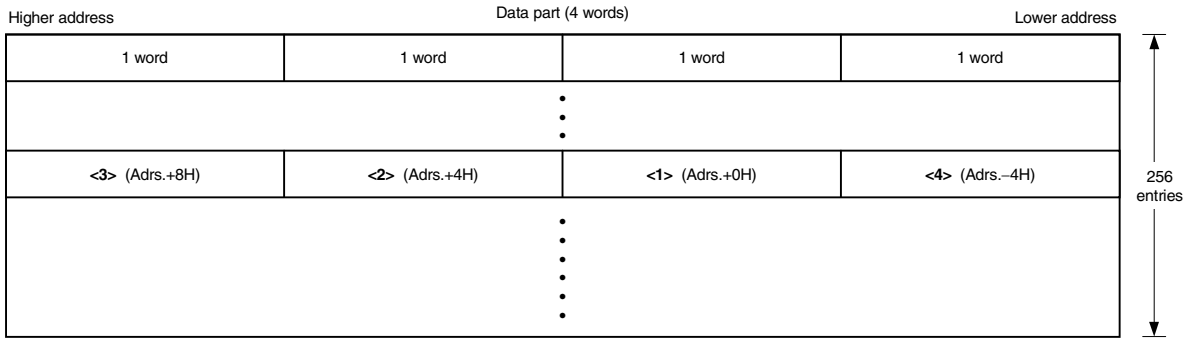
Remarks 1. The numbers within pointed brackets (< >) indicate the refill sequence.

2. (Adrs.+n): Data of address in ( ) (n = –8H to +7H)

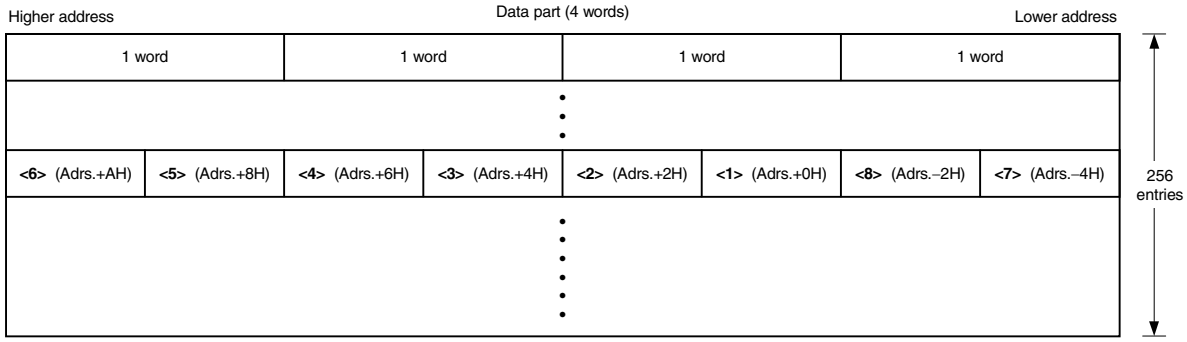
Figure 2-33. Refill Sequence to Data Cache (Critical First Refill (1R-2R-1R)) (1/2)

(1) IRAMA3, IRAMA2 = 01

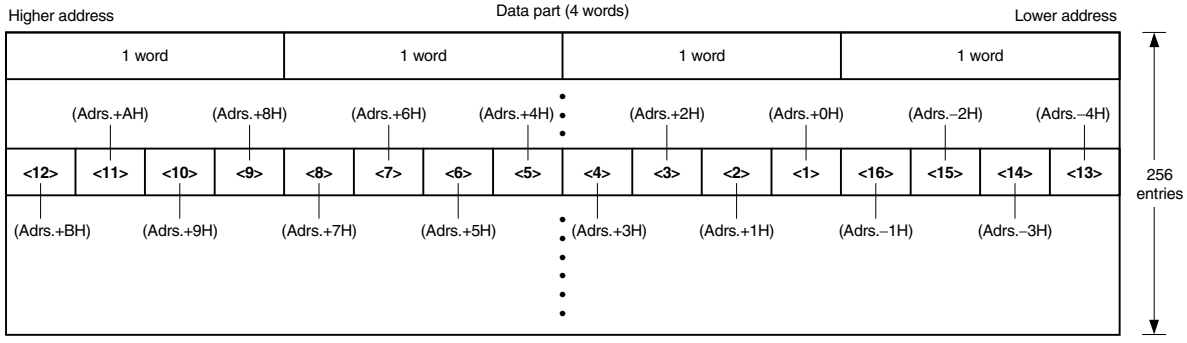
(a) 32-bit data bus



(b) 16-bit data bus



(c) 8-bit data bus



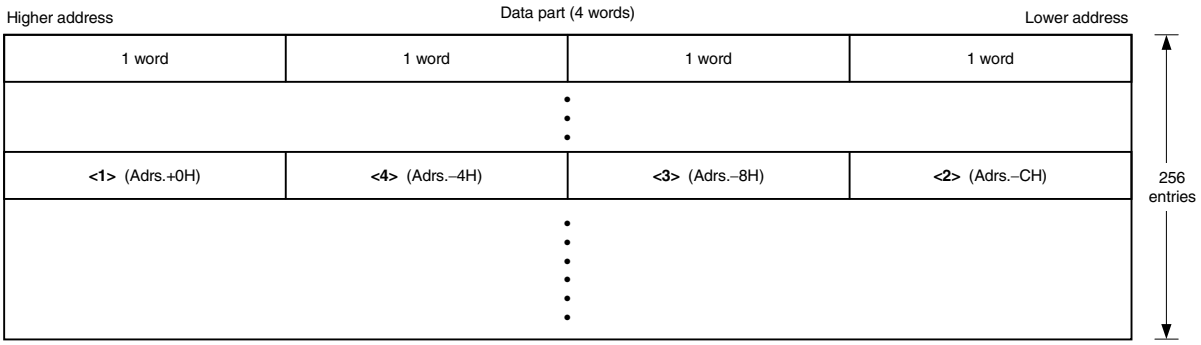
Remarks 1. The numbers within pointed brackets (< >) indicate the refill sequence.

2. (Adrs.+n): Data of address in ( ) (n = –4H to +BH)

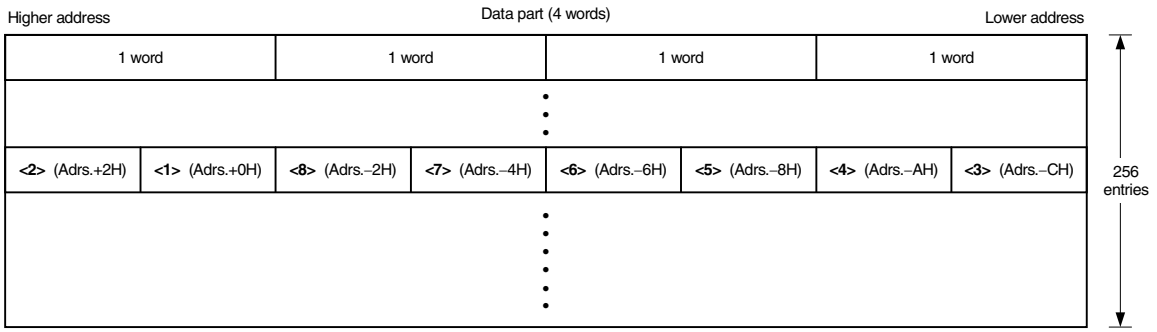
Figure 2-33. Refill Sequence to Data Cache (Critical First Refill (1R-2R-1R)) (2/2)

(2) IRAMA3, IRAMA2 = 11

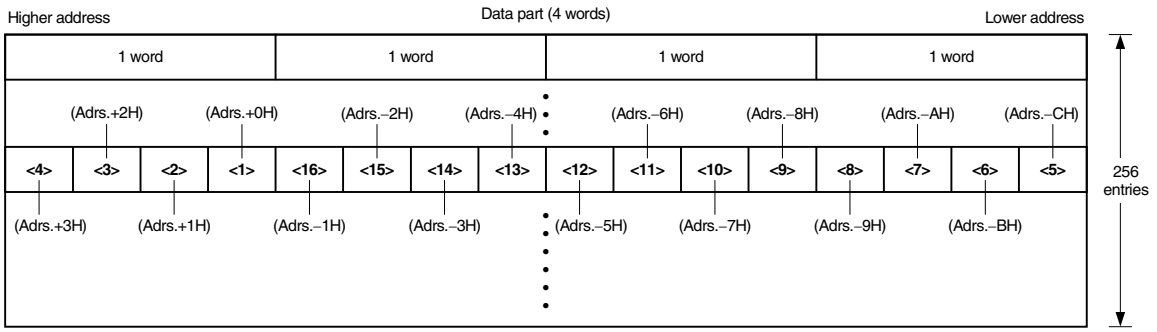
(a) 32-bit data bus



(b) 16-bit data bus



(c) 8-bit data bus



Remarks 1. The numbers within pointed brackets (< >) indicate the refill sequence.

2. (Adrs.+n): Data of address in ( ) (n = –CH to +3H)

2.10 Cautions

(1) Connection to NB85E

Connect pins that have the same pin names.

(2) Setting cache type selection pins

Input the levels shown below to cache type selection pins beginning with IFI.
However, connect the IFIUNCH1 pin and the IFIWRTH pin to the NB85E.

Pin Name	Input Level	
	NB85E252	NB85E263
IFIASEQ	Don't care	Don't care
IFIRABE	Low level	Low level
IFIDRCT[Note]	–	Low level
IFIOECT	Low level	Low level

Note NB85E263 only

(3) Bus cycle status

For all read cycles of an area for which the data cache setting is set to cacheable by the cache configuration register (BHC) of the NB85E and write cycles in writeback mode (write allocate enabled), the VBCTYP2 to VBCTYP0 signals of the NB85E always indicate a data access and do not indicate a misalign access.

(4) Operation on reset

At the time of a reset, tags are automatically cleared (invalidated), which puts the next data replacement in a state of being performed from way 0. Therefore, if there is an access to the data cache within a period of as many clock cycles as the number of lines after a reset, the CPU stops until the tags are cleared (become invalid).

(5) Test bus auto wiring tool support

This data cache does not support test bus auto wiring tools because although it has a BUNRI pin, it does not have test buses (TBOx, TBIx).

(6) Other

This data cache does not have a bus snoop circuit (which monitors the bus operation). Note that data in the data cache in the cases shown in the following examples is dirty data even when there is no write access to the data cache, and is data that has lost its coherency. To avoid this status, be sure to clear tags.

★ **Examples 1.** When DMA transfer is performed to the external memory of a cacheable area
(Transfer data is not reflected in the data cache)

2. When the external memory contents of a cacheable area are overwritten by the external bus master

★ **(7) Operation during debugging**

This data cache does not operate during debugging using an N-wire type in-circuit emulator.
When accessing external memory in the cacheable area during debugging, only the external memory is accessed even if the data cache is valid, and data loses its coherency. To avoid this, be sure to clear tags.
In addition, when using the in-circuit emulator (IE-V850E-MC-A), debugging of the data cache cannot be performed.

APPENDIX A CACHE PERFORMANCE COMPARISON TABLE

This section indicates the relative processing performance of the instruction and data caches, using 100 as the reference value for the performance obtained when executing the Dhrystone benchmark program with ROM connected to the VFB of the NB85E and RAM connected to the VDB of the NB85E.

Caution The values listed in the cache performance comparison table are have been obtained under the following operation environment. Performance values vary according to the operation environment of the user.

<Operation environment>

- Co-simulation tool: Virtual ICE™ (Ver. 2.2.1d) NB85E model
- Workstation: Sun Ultra2 (300 MHz, 2CPU/768 MB memory)
- OS: Solaris™2.6 (SunOS™5.6)
- HDL simulator: Verilog-XL™ Ver. 2.2.1 (no-turbo)
- Software development environment: C compiler for V850E core (Ver. 1.8.9) made by Green Hills Software™
- Used program: Dhrystone Ver. 2.1
  1-loop instructions: 413 (2-byte instructions: 254,
  4-byte instructions: 156,
  6-byte instructions: 3)
  1-loop program capacity: 1150 bytes

How to read the comparison table is explained on the following page.

<How to read cache performance comparison table>

- **Program loop number**
  The cache performance for the first loop and for the second and subsequent loops of the Dhrystone benchmark program are compared.
- **Program placement**
  Where to place the Dhrystone benchmark program is indicated.
- **Work data placement**
  Where to place the work RAM area to be used for data access is indicated.
- **Access conditions**
  (a) For ROM/RAM, external ROM/SRAM
  The number of waits and the total number of clocks required for the bus cycle are indicated.
  In the case of ROM/RAM, 1 clock is required as the bus cycle for VFB/VDB connection.
  In the case of external ROM/SRAM, the total number of clocks increases by the number of waits. However, the number of waits described here is not the value set with data wait control registers 0 and 1 (DWC0, DWC1) of the memory controller (MEMC), but the number of waits after which the VBWAIT signal of the NB85E becomes active (a high level is output).
  (b) For external SDRAM
  The value of CAS latency (CL) is indicated.
- **Relative value**
  The relative processing performance is shown with "*" locations in the table representing 100.
- **Hit rate**
  The hit rate of the cache is indicated.

Remark ROM: ROM directly connected to NB85E (connected to VFB)
RAM: RAM directly connected to NB85E (connected to VDB)
External ROM/SRAM/SDRAM: ROM/SRAM/SDRAM connected via VSB

(1) Instruction cache only (1/2)

Connected instruction cache				No	Yes			
Program loop number				–	1st loop		2nd and subsequent loops	
Program placement		Work data placement		Relative value	Relative value	Hit rate (%)	Relative value	Hit rate (%)
Bus width (bits)	Access conditions	Bus width (bits)	Access conditions					
ROM		RAM		–				
32	0 waits 1 clock	32	0 waits 1 clock	100 *	–	–	–	–
External ROM		RAM		–				
32	0 waits 1 clock	32	0 waits 1 clock	82.2	58.4	88.0	89.1	100.0
	1 wait 2 clocks			63.7	48.6	88.0	88.3	100.0
	2 waits 3 clocks			45.7	41.6	88.0	87.4	100.0
16	0 waits 1 clock			65.9	48.7	88.0	88.4	100.0
	1 wait 2 clocks			35.9	35.4	88.0	85.8	100.0
	2 waits 3 clocks			24.1	29.1	88.0	85.0	100.0
8	0 waits 1 clock			35.9	35.4	88.0	85.8	100.0
	1 wait 2 clocks			18.2	24.2	88.0	83.3	100.0
	2 waits 3 clocks			12.2	18.1	88.0	80.0	100.0
External ROM		External SRAM		–				
32	0 waits 1 clock	32	0 waits 1 clock	75.5	51.9	84.3	82.4	100.0
	3 waits 4 clocks		3 waits 4 clocks	28.4	27.2	84.3	59.9	100.0
16	0 waits 1 clock	16	0 waits 1 clock	54.0	40.7	84.3	74.6	100.0
	3 waits 4 clocks		3 waits 4 clocks	14.7	16.4	84.3	42.7	100.0

(1) Instruction cache only (2/2)

Connected instruction cache				No	Yes			
Program loop number				–	1st loop		2nd and subsequent loops	
Program placement		Work data placement		Relative value	Relative value	Hit rate (%)	Relative value	Hit rate (%)
Bus width (bits)	Access conditions	Bus width (bits)	Access conditions					
External ROM		External SDRAM		–				
32	3 waits 4 clocks	32	CL = 2	28.4	27.2	84.3	60.0	100.0
		16		16.0	17.8	84.3	53.4	100.0
16				27.0	26.0	84.3	55.2	100.0
External SDRAM		External SDRAM		–				
32	CL = 2	32	CL = 2	28.4	36.8	84.3	59.9	100.0
16		16		23.1	30.1	84.3	54.8	100.0

(2) Instruction cache and data cache (write through mode)

Connected instruction cache				No	Yes							
Connected data cache				No	Yes							
Program loop number				–	1st loop				2nd and subsequent loops			
Program placement		Work data placement		Relative value	Relative value	Hit rate (%)			Relative value	Hit rate (%)		
						Instruction cache	Data cache			Instruction cache	Data cache	
Bus width (bits)	Access conditions	Bus width (bits)	Access conditions				Read	Write			Read	Write
ROM		RAM		–								
32	0 waits 1 clock	32	0 waits 1 clock	100 *	–	–	–	–	–	–	–	–
External ROM		External SRAM		–								
32	0 waits 1 clock	32	0 waits 1 clock	75.5	50.7	84.3	86.6	64.1	81.1	100.0	100.0	77.3
	3 waits 4 clocks		3 waits 4 clocks	28.4	28.6	84.3	86.6	64.1	76.7	100.0	100.0	77.3
16	0 waits 1 clock	16	0 waits 1 clock	54.0	40.9	84.3	86.6	64.1	80.8	100.0	100.0	77.3
	3 waits 4 clocks		3 waits 4 clocks	14.7	17.6	84.3	86.6	64.1	67.5	100.0	100.0	77.3
External ROM		External SDRAM		–								
32	3 waits 4 clocks	32	CL = 2	28.4	28.8	84.3	86.6	64.1	76.7	100.0	100.0	77.3
		16		16.0	18.8	84.3	86.6	64.1	72.2	100.0	100.0	77.3
16				27.0	28.2	84.3	86.6	64.1	75.3	100.0	100.0	77.3
External SDRAM		External SDRAM		–								
32	CL = 2	32	CL = 2	28.4	39.7	84.3	86.6	64.1	76.7	100.0	100.0	77.3
16		16		23.1	32.8	84.3	86.6	64.1	74.6	100.0	100.0	77.3

(3) Instruction cache and data cache (writeback mode (write allocate disabled))

Connected instruction cache				No	Yes							
Connected data cache				No	Yes							
Program loop number				–	1st loop				2nd and subsequent loops			
Program placement		Work data placement		Relative value	Relative value	Hit rate (%)			Relative value	Hit rate (%)		
Bus width (bits)	Access conditions	Bus width (bits)	Access conditions			Instruction cache	Data cache			Instruction cache	Data cache	
							Read	Write			Read	Write
ROM		RAM		–								
32	0 waits 1 clock	32	0 waits 1 clock	100 *	–	–	–	–	–	–	–	–
External ROM		External SRAM		–								
32	0 waits 1 clock	32	0 waits 1 clock	75.5	50.7	84.3	86.6	64.1	81.8	100.0	100.0	77.3
	3 waits 4 clocks		3 waits 4 clocks	28.4	29.2	84.3	86.6	64.1	79.3	100.0	100.0	77.3
16	0 waits 1 clock	16	0 waits 1 clock	54.0	41.1	84.3	86.6	64.1	81.4	100.0	100.0	77.3
	3 waits 4 clocks		3 waits 4 clocks	14.7	18.4	84.3	86.6	64.1	75.7	100.0	100.0	77.3
External ROM		External SDRAM		–								
32	3 waits 4 clocks	32	CL = 2	28.4	29.5	84.3	86.6	64.1	79.3	100.0	100.0	77.3
		16		16.0	29.2	84.3	86.6	64.1	79.3	100.0	100.0	77.3
16				27.0	19.4	84.3	86.6	64.1	75.7	100.0	100.0	77.3
External SDRAM		External SDRAM		–								
32	CL = 2	32	CL = 2	28.4	41.1	84.3	86.6	64.1	79.3	100.0	100.0	77.3
16		16		23.1	34.2	84.3	86.6	64.1	78.3	100.0	100.0	77.3

(4) Instruction cache and data cache (writeback mode (write allocate enabled))

Connected instruction cache				No	Yes							
Connected data cache				No	Yes							
Program loop number				–	1st loop				2nd and subsequent loops			
Program placement		Work data placement		Relative value	Relative value	Hit rate (%)			Relative value	Hit rate (%)		
						Instruction cache	Data cache			Instruction cache	Data cache	
Bus width (bits)	Access conditions	Bus width (bits)	Access conditions				Read	Write			Read	Write
ROM		RAM		–								
32	0 waits 1 clock	32	0 waits 1 clock	100 *	–	–	–	–	–	–	–	–
External ROM		External SRAM		–								
32	0 waits 1 clock	32	0 waits 1 clock	75.5	50.9	84.3	100.0	86.7	81.8	100.0	100.0	100.0
	3 waits 4 clocks		3 waits 4 clocks	28.4	28.4	84.3	100.0	86.7	79.3	100.0	100.0	100.0
16	0 waits 1 clock	16	0 waits 1 clock	54.0	40.3	84.3	100.0	86.7	81.4	100.0	100.0	100.0
	3 waits 4 clocks		3 waits 4 clocks	14.7	17.4	84.3	100.0	86.7	75.7	100.0	100.0	100.0
External ROM		External SDRAM		–								
32	3 waits 4 clocks	32	CL = 2	28.4	29.9	84.3	100.0	86.7	79.3	100.0	100.0	100.0
		16		16.0	29.2	84.3	100.0	86.7	79.3	100.0	100.0	100.0
16				27.0	19.4	84.3	100.0	86.7	75.7	100.0	100.0	100.0
External SDRAM		External SDRAM		–								
32	CL = 2	32	CL = 2	28.4	41.9	84.3	100.0	86.7	79.3	100.0	100.0	100.0
16		16		23.1	34.4	84.3	100.0	86.7	78.3	100.0	100.0	100.0

APPENDIX B REVISION HISTORY

The major revisions up to the previous edition are shown below. “Page” indicates the pages in older edition to which the revision was applied.

(1) 1st → 2nd

Page	Description
pp.15, 25, 28	Modification of instruction cache autofill function to be applied to way 0 only
p.25	Modification of **1.4.1 (1) Instruction cache control register (ICC)**
p.27	Addition of Caution to **1.4.1 (2) Instruction cache data configuration register (ICD)**
p.27	Modification of initial value in **Figure 1-6 Instruction Cache Data Configuration Register (ICD)**
p.28	Modification of **1.4.2 Tag clear function**
p.29	Addition of **1.5 Instruction Cache Setting Procedure**
pp.29, 30	Modification of **1.6.1 Operation on instruction cache hit** and **1.6.2 Operation on instruction cache miss**
p.31 p.33	**1.7 Bus Cycle Issued by Instruction Cache** • Modification of Remark • Addition of timing example in case of 16-bit data bus
p.34	Addition of **1.8 Refill Sequence to Instruction Cache**
p.35	Modification of **1.9 (5) Setting registers**
p.36	Addition of **1.9 (7) Initial program settings** and **(8) Setting BHC register of NB85E**
p.37	Modification of **2.1.1 Features**
p.39	Modification of **Figure 2-1 NB85E and Data Cache Connection Example**
p.42	Modification of **2.2.2 (1) (g) IRAMWR3 to IRAMWR0 (input)** and **(l) IDHUM (output)**
p.44	Modification of **2.2.2 (2) (b) IFIRABE, IFIDRCT, and IFIOECT (input)**
p.44	Modification of **2.2.2 (3) (a) IDHIT (output)**
p.49	Addition of Caution to **2.4.1 (1) Data cache control register (DCC)**
p.51	Addition of Caution to **2.4.1 (2) Data cache data configuration register (DCD)**
p.51	Modification of initial value in **Figure 2-6 Data Cache Data Configuration Register (DCD)**
p.53	Addition of **2.5 Data Cache Setting Procedure**
p.70 p.71 pp.73, 75, 78, 79, 81, 83, 86, 87	**2.7 Bus Cycles Issued by Data Cache** • Modification of Remark • Modification of Bus Cycle column in **Table 2-3 Operating Modes and Bus Cycles** • Addition of timing example in case of 16-bit data bus
pp.88 to 91	Addition of **2.8 Timing of Refill from SDRAM to Data Cache**
pp.92 to 95	Addition of **2.9 Refill Sequence to Data Cache**
p.96	Addition of **2.10 (5) Others**
pp.97 to 103	Addition of **APPENDIX CACHE PERFORMANCE COMPARISON TABLE**

(2) 2nd → 3rd

Page	Description
p.27	Modification of bit description in **Figure 1-6 Instruction Cache Data Configuration Register (ICD)**
pp.28, 29	Addition of **Caution** and **Remark** and modification of description of tag clear procedure in **1.4.2 Tag clear function**
p.29	Addition of **Remark** to **1.4.3 Autofill function (way 0 only)**
p.30	Modification of description of initial setting procedure in **1.5 Instruction Cache Setting Procedure**
p.37	Addition of **1.9 (9) Test bus auto wiring tool support**
p.38	Addition of **1.9 (10) Tag clear procedure**
p.45	Modification of description in **2.2.2 (1) (s) IFIUNCH1**
p.53	Modification of bit description in **Figure 2-6 Data Cache Data Configuration Register (DCD)**
p.55	Deletion of **Caution** from **2.5 Data Cache Setting Procedure**
p.91	Modification of output value of VBA25 to VBA0 and A24 to A13 signals in TLATE state in **Figure 2-28 Refill Timing Example from SDRAM to Data Cache (Sequential Refill (4R), Critical First Refill (4R))**
pp.91 to 93	Modification of address bus timing between MEMC and SDRAM in **Figures 2-28** to **2-30**
p.98	Addition of **2.10 (5) Test bus auto wiring tool support**
p.98	Modification of description in **2.10 (6) Other**
p.107	Addition of **APPENDIX B REVISION HISTORY**

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
Market Communication Dept.
Fax: +49-211-6503-274

South America
NEC do Brasil S.A.
Fax: +55-11-6462-6829

Hong Kong, Philippines, Oceania
NEC Electronics Hong Kong Ltd.
Fax: +852-2886-9022/9044

Korea
NEC Electronics Hong Kong Ltd.
Seoul Branch
Fax: +82-2-528-4411

Taiwan
NEC Electronics Taiwan Ltd.
Fax: +886-2-2719-5951

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-250-3583

Japan
NEC Semiconductor Technical Hotline
Fax: +81- 44-435-9608

I would like to report the following error/make the following suggestion:

Document title: _____

Document number: _____ Page number: _____

If possible, please fax the referenced page or drawing.

Document Rating	Excellent	Good	Acceptable	Poor
Clarity	❑	❑	❑	❑
Technical Accuracy	❑	❑	❑	❑
Organization	❑	❑	❑	❑