

To our customers,

---

## Old Company Name in Catalogs and Other Documents

---

On April 1<sup>st</sup>, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: <http://www.renesas.com>

April 1<sup>st</sup>, 2010  
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (<http://www.renesas.com>)

Send any inquiries to <http://www.renesas.com/inquiry>.

## Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
  - “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.
  - “High Quality”: 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; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.



## **Preliminary User's Manual**

# **Instruction Cache, Data Cache**

**NU85E, NU85ET**

---

**NU85E212**

**NU85E213**

**NU85E252**

**NU85E263**

Document No. A15241EJ2V1UM00 (2nd edition)  
Date Published January 2002 NS CP(N)

© NEC Corporation 2001  
Printed in Japan

[MEMO]

## NOTES FOR CMOS DEVICES

### ① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note:

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

### ② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note:

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

### ③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note:

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

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

### Major Revisions in This Edition

Pages	Description
p.27	Addition of description to <b>Caution</b> in <b>1.4.2 Tag clear function</b>
p.28	Addition of <b>Caution 2</b> in <b>1.4.2 Tag clear function</b>
p.28	Addition of description to <b>Caution</b> and addition of <b>Remark 3</b> in <b>1.4.3 Autofill function (way 0 only)</b>
p.29	Addition of description to <b>Caution 2</b> in <b>1.5 Instruction Cache Setting Procedure</b>
p.30	Addition of description to <b>Caution</b> in <b>1.6.2 Operation on instruction cache miss</b>
p.37	Addition of <b>1.9 (9) Simultaneous operation of refill read cycle and cache access by specific instruction that performs branch</b>
p.98	Modification of <b>Example 1</b> in <b>2.10 (5) Other</b>

The mark ★ shows major revised points.



## PREFACE

**Target Readers** This manual is intended to give users an understanding the functions of the instruction caches (NU85E212 and NU85E213) and data caches (NU85E252 and NU85E263) for the NU85E and NU85ET 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 NU85E212 and NU85E213, which are instruction caches.

### CHAPTER 2 DATA CACHE

This chapter explains the NU85E252 and NU85E263, 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 NU85E and NU85ET, refer to:

→ **NU85E Hardware User's Manual (A14874E)** and **NU85ET Hardware User's Manual (A15015E)**

In this manual, unless specifically noted, the NU85E is described as the typical CPU core product. When using the NU85ET, read "NU85E" as "NU85ET".

**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)
<b>Note:</b>	Footnote for item marked with <b>Note</b> in the text
<b>Caution:</b>	Information requiring particular attention
<b>Remark:</b>	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)
- NU85E Hardware User's Manual (A14874E)
- NU85ET Hardware User's Manual (A15015E)
- Memory Controller NU85E, NU85ET User's Manual (A15019E)
- CB-10 Family VX Type NU85E, NU85ET Design Manual (A15401E)

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

# CONTENTS

<b>CHAPTER 1 INSTRUCTION CACHE</b> .....	<b>14</b>
<b>1.1 Outline</b> .....	<b>14</b>
1.1.1 Features.....	14
1.1.2 Symbol diagram.....	15
1.1.3 NU85E connection example.....	16
<b>1.2 Pin Functions</b> .....	<b>17</b>
1.2.1 List of pin functions.....	17
1.2.2 Explanation of pin functions.....	18
1.2.3 Pin status.....	20
<b>1.3 Configuration of Instruction Cache</b> .....	<b>21</b>
1.3.1 4 KB 2-way set-associative instruction cache.....	22
1.3.2 8 KB 2-way set-associative instruction cache.....	23
<b>1.4 Instruction Cache Control Functions</b> .....	<b>24</b>
1.4.1 Control registers.....	24
1.4.2 Tag clear function.....	27
1.4.3 Autofill function (way 0 only).....	28
<b>1.5 Instruction Cache Setting Procedure</b> .....	<b>29</b>
<b>1.6 Operation</b> .....	<b>29</b>
1.6.1 Operation on instruction cache hit.....	29
1.6.2 Operation on instruction cache miss.....	30
<b>1.7 Bus Cycle Issued by Instruction Cache</b> .....	<b>31</b>
<b>1.8 Refill Sequence to Instruction Cache</b> .....	<b>34</b>
<b>1.9 Cautions</b> .....	<b>35</b>
<b>CHAPTER 2 DATA CACHE</b> .....	<b>38</b>
<b>2.1 Outline</b> .....	<b>38</b>
2.1.1 Features.....	38
2.1.2 Symbol diagram.....	39
2.1.3 NU85E connection example.....	40
<b>2.2 Pin Functions</b> .....	<b>41</b>
2.2.1 List of pin functions.....	41
2.2.2 Explanation of pin functions.....	42
2.2.3 Pin status.....	46
<b>2.3 Configuration of Data Cache</b> .....	<b>47</b>
2.3.1 4 KB directly mapped data cache.....	48
2.3.2 8 KB 2-way set-associative data cache.....	49
<b>2.4 Data Cache Control Functions</b> .....	<b>50</b>
2.4.1 Control registers.....	50
2.4.2 Tag clear function.....	53
2.4.3 Tag fill function.....	53
2.4.4 Lock function.....	53
2.4.5 Data flush function.....	54

<b>2.5</b>	<b>Data Cache Setting Procedure</b> .....	<b>55</b>
2.5.1	Setting to validate data cache.....	55
2.5.2	Setting to validate, invalidate, and revalidate data cache.....	55
<b>2.6</b>	<b>Operation</b> .....	<b>56</b>
2.6.1	Write through mode.....	57
2.6.2	Writeback mode (write allocate disabled).....	61
2.6.3	Writeback mode (write allocate enabled).....	66
<b>2.7</b>	<b>Bus Cycles Issued by Data Cache</b> .....	<b>72</b>
<b>2.8</b>	<b>Timing of Refill from SDRAM to Data Cache</b> .....	<b>90</b>
<b>2.9</b>	<b>Refill Sequence to Data Cache</b> .....	<b>94</b>
<b>2.10</b>	<b>Cautions</b> .....	<b>98</b>

## LIST OF FIGURES (1/2)

Figure No.	Title	Page
1-1	NU85E and Instruction Cache Connection Example .....	16
1-2	Instruction Cache Configuration Example .....	21
1-3	Configuration of 4 KB 2-Way Set-Associative Instruction Cache .....	22
1-4	Configuration of 8 KB 2-Way Set-Associative Instruction Cache .....	23
1-5	Instruction Cache Control Register (ICC) .....	25
1-6	Instruction Cache Data Configuration Register (ICD) .....	26
1-7	Operation on Instruction Cache Hit.....	29
1-8	Operation on Instruction Cache Miss.....	30
1-9	Sequential Refill Read Cycle (4R) .....	32
1-10	Refill Sequence to Instruction Cache.....	34
1-11	Cache Area Setting Example.....	36
2-1	NU85E and Data Cache Connection Example .....	40
2-2	Data Cache Configuration Example .....	47
2-3	Configuration of 4 KB Directly Mapped Data Cache.....	48
2-4	Configuration of 8 KB 2-Way Set-Associative Data Cache .....	49
2-5	Data Cache Control Register (DCC).....	51
2-6	Data Cache Data Configuration Register (DCD) .....	52
2-7	Operation on Data Cache Hit (Write Through Mode, Read) .....	57
2-8	Operation on Data Cache Miss (Write Through Mode, Read) .....	58
2-9	Operation on Data Cache Hit (Write Through Mode, Write) .....	59
2-10	Operation on Data Cache Miss (Write Through Mode, Write) .....	60
2-11	Operation on Data Cache Hit (Writeback Mode, Write Allocate Disabled, Read).....	61
2-12	Operation on Data Cache Miss (Writeback Mode, Write Allocate Disabled, Read, Clean Data).....	62
2-13	Operation on Data Cache Miss (Writeback Mode, Write Allocate Disabled, Read, Dirty Data).....	63
2-14	Operation on Data Cache Hit (Writeback Mode, Write Allocate Disabled, Write).....	64
2-15	Operation on Data Cache Miss (Writeback Mode, Write Allocate Disabled, Write).....	65
2-16	Operation on Data Cache Hit (Writeback Mode, Write Allocate Enabled, Read) .....	66
2-17	Operation on Data Cache Miss (Writeback Mode, Write Allocate Enabled, Read, Clean Data) .....	67
2-18	Operation on Data Cache Miss (Writeback Mode, Write Allocate Enabled, Read, Dirty Data) .....	68
2-19	Operation on Data Cache Hit (Writeback Mode, Write Allocate Enabled, Write).....	69
2-20	Operation on Data Cache Miss (Writeback Mode, Write Allocate Enabled, Write, Clean Data).....	70
2-21	Operation on Data Cache Miss (Writeback Mode, Write Allocate Enabled, Write, Dirty Data).....	71
2-22	Sequential Refill Read Cycle (4R) .....	74

## LIST OF FIGURES (2/2)

Figure No.	Title	Page
2-23	Critical First Refill Read Cycle (2R-2R).....	76
2-24	Critical First Refill Read Cycle (1R-2R-1R).....	78
2-25	Sequential Refill Read Cycle in Writeback Mode (When Data Being Replaced Is Dirty Data) (4W + 4R) ....	82
2-26	Critical First Refill Read Cycle in Writeback Mode (When Data Being Replaced Is Dirty Data) (4W + 2R-2R) .....	84
2-27	Critical First Refill Read Cycle in Writeback Mode (When Data Being Replaced Is Dirty Data) (4W + 1R-2R-1R).....	86
2-28	Refill Timing Example from SDRAM to Data Cache (Sequential Refill (4R), Critical First Refill (4R)) .....	91
2-29	Refill Timing Example from SDRAM to Data Cache (Critical First Refill (2R-2R)) .....	92
2-30	Refill Timing Example from SDRAM to Data Cache (Critical First Refill (1R-2R-1R)) .....	93
2-31	Refill Sequence to Data Cache (Sequential Refill (4R), Critical First Refill (4R)).....	94
2-32	Refill Sequence to Data Cache (Critical First Refill (2R-2R)).....	95
2-33	Refill Sequence to Data Cache (Critical First Refill (1R-2R-1R)) .....	96

## LIST OF TABLES

Table No.	Title	Page
1-1	Pin Status in Each Operating Mode.....	20
2-1	Pin Status in Each Operating Mode.....	46
2-2	List of Operating Modes.....	56
2-3	Operating Modes and Bus Cycles .....	73

## CHAPTER 1 INSTRUCTION CACHE

### 1.1 Outline

The NU85E212 and NU85E213 are instruction cache memories for the NU85E.

They can be directly connected to the instruction cache interface incorporated in the NU85E.

The following two types of instruction caches are available.

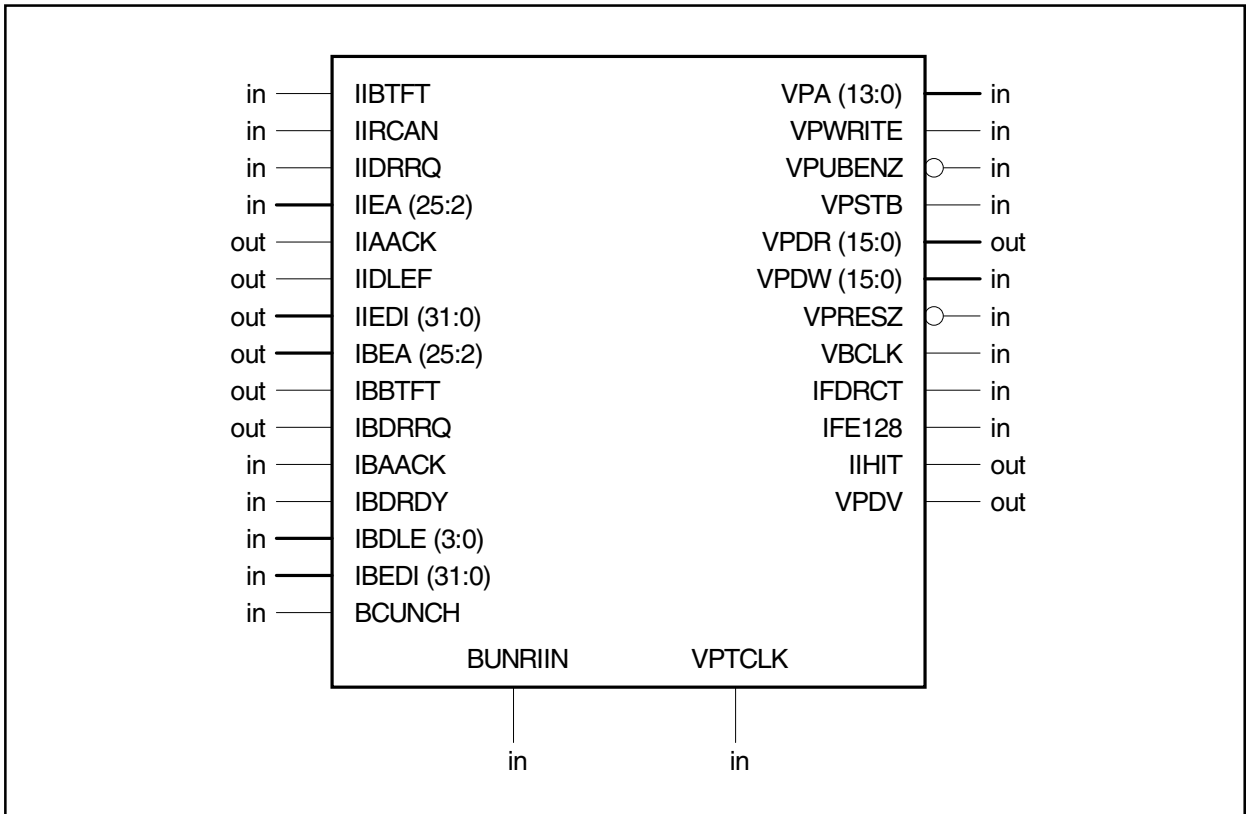
- NU85E212 ... 4 KB 2-way set-associative instruction cache (4 words  $\times$  128 entries  $\times$  2 ways = 4 KB)
- NU85E213 ... 8 KB 2-way set-associative instruction cache (4 words  $\times$  256 entries  $\times$  2 ways = 8 KB)

#### 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.



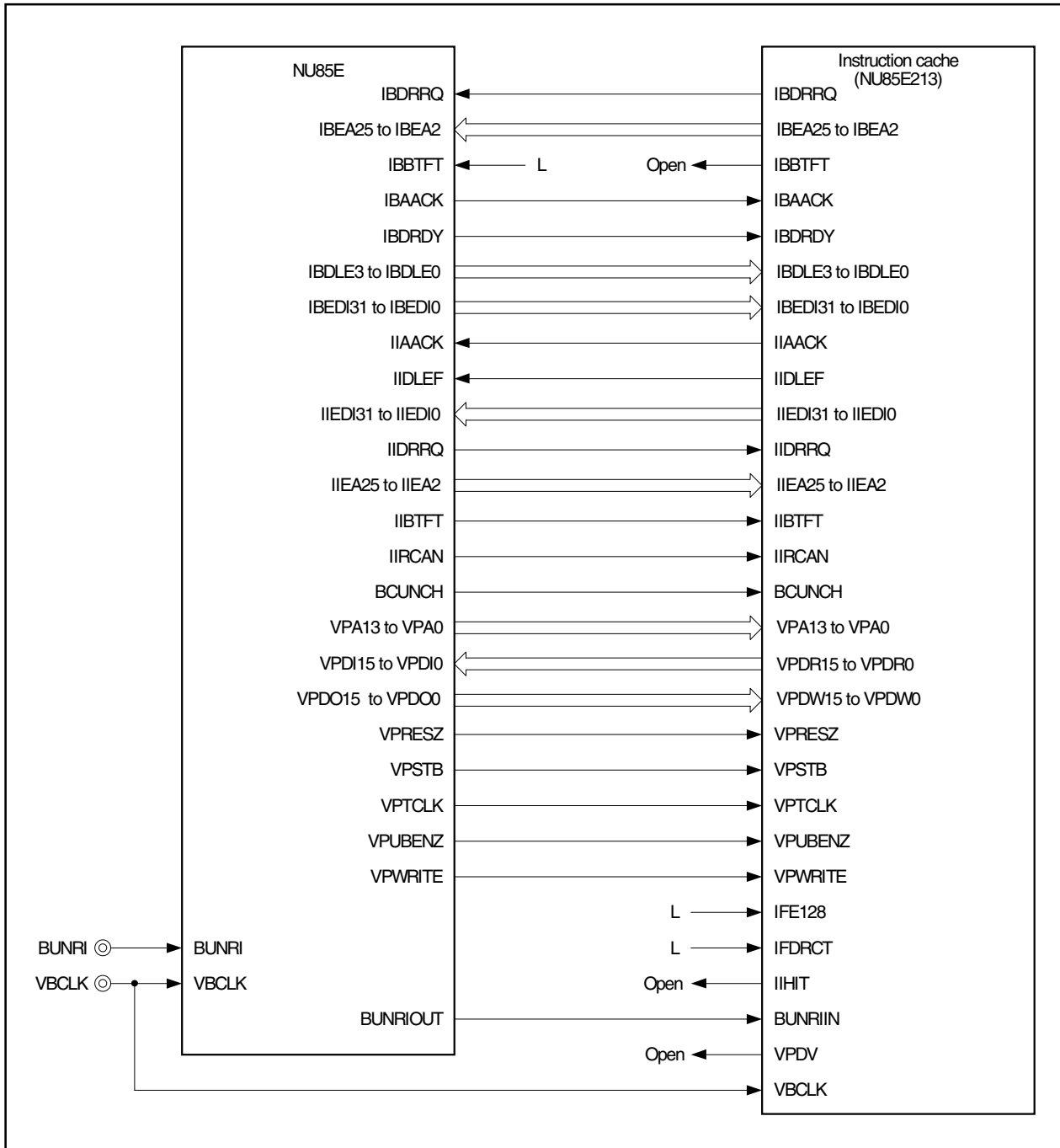
1.1.2 Symbol diagram



1.1.3 NU85E connection example

The following figure shows an example of the connection of an instruction cache to the NU85E.

Figure 1-1. NU85E and Instruction Cache Connection Example



## 1.2 Pin Functions

### 1.2.1 List of pin functions

Pin Name	I/O	Function	
NU85E connection pins	IIBTFT	Input	Input branch target fetch status from NU85E
	IIRCAN	Input	Input code cancel status from NU85E
	IIDRRQ	Input	Input fetch request from NU85E
	IIEA25 to IIEA2	Input	Input fetch address from NU85E
	IIAACK	Output	Output address acknowledge to NU85E
	IIDLEF	Output	Output data latch enable to NU85E
	IIEDI31 to IIEDI0	Output	Output data to NU85E
	IBAACK	Input	Input address acknowledge from NU85E
	IBDRDY	Input	Input data ready from NU85E
	IBDLE3 to IBDLE0	Input	Input data latch enable from NU85E
	IBEDI31 to IBEDI0	Input	Input data from NU85E
	BCUNCH	Input	Input uncache status from NU85E
	IBEA25 to IBEA2	Output	Output fetch address to NU85E
	IBBTFT	Output	NEC reserved pin (leave open)
	IBDRRQ	Output	Output fetch request to NU85E
	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)
	VPDR15 to VPDR0	Output	Output data (for NPB)
	VPDW15 to VPDW0	Input	Input data (for NPB)
	VPRESZ	Input	Input reset
	VBCLK	Input	Input internal system clock
	VPDV	Output	Output data bus direction control (for NPB)
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	BUNRIIN	Input	Input normal/test mode selection
	VPTCLK	Input	Input clock for test

**Remark** NPB: NEC peripheral I/O bus

## 1.2.2 Explanation of pin functions

### (1) NU85E connection pins

#### (a) IIBTFT (input)

IIBTFT inputs the branch target fetch status from the NU85E.

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 NU85E.

This is the signal for canceling the preceding request when data becomes unnecessary due to a branch or interrupt after the NU85E outputs a fetch request to the instruction cache.

#### (c) IIDRRQ (input)

IIDRRQ inputs a fetch request from the NU85E.

#### (d) IIEA25 to IIEA2 (input)

IIEA25 to IIEA2 constitute a bus that inputs a fetch address from the NU85E.

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 NU85E.

This signal is output to the NU85E when a fetch address from the NU85E (IIEA25 to IIEA2) is recognized.

#### (f) IIDLEF (output)

IIDLEF outputs a data latch enable to the NU85E.

#### (g) IIEDI31 to IIEDI0 (output)

IIEDI31 to IIEDI0 constitute a bus that outputs data to the NU85E.

This bus outputs the data that the NU85E is to read.

#### (h) IBAACK (input)

IBAACK inputs an address acknowledge from the NU85E.

This signal is input when the NU85E recognizes the IBEA25 to IBEA2 signals output from the instruction cache.

#### (i) IBDRDY (input)

IBDRDY inputs a data ready from the NU85E.

This is input when the NU85E 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 NU85E.

#### (k) IBEDI31 to IBEDI0 (input)

IBEDI31 to IBEDI0 constitute a bus that inputs data from the NU85E.

Refill data is input from the NU85E when a miss occurs.

**(l) BCUNCH (input)**

BCUNCH inputs the uncached status from the NU85E.

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 NU85E is accessed.

**(m) IBEA25 to IBEA2 (output)**

IBEA25 to IBEA2 constitute a bus that outputs a fetch address to the NU85E.

This bus outputs the address that the NU85E should read when a miss occurs.

**(n) IBBTFT (output)**

IBBTFT is reserved for NEC. Leave it open.

Fix the IBBTFT pin of the NU85E to low level.

**(o) IBDRRQ (output)**

IBDRRQ outputs a fetch request to the NU85E.

This pin outputs a signal requesting that the NU85E perform a fetch from external memory.

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

Refer to the **NU85E Hardware User's Manual (A14874E)**.

**(q) VPDR15 to VPDR0 (output)**

VPDR15 to VPDR0 output data to the NU85E.

Connect these pins to the VPDI15 to VPDI0 pins of the NU85E.

**(r) VPDW15 to VPDW0 (input)**

VPDW15 to VPDW0 input data from the NU85E.

Connect these pins to the VPDO15 to VPDO0 pins of the NU85E.

**(s) VPRESZ (input)**

VPRESZ inputs a reset.

**(t) VBCLK (input)**

VBCLK inputs the internal system clock.

**(u) VPDV (output)**

VPDV outputs data bus direction control for the NPB.

When connecting to the NU85E, leave this pin open since it is not used.

**(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 NU85E213 to low level)
- High level: 128 entries (Fix NU85E212 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) BUNRIIN (input)**

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

Connect this pin to the BUNRIOUT pin of the NU85E.

**(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
NU85E 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
	VPDR15 to VPDR0	L	Maintained	Operating	Operating
	VPDV	Undefined	Maintained	Operating	Operating
Status pin	IIHIT	L	Maintained	Operating	Operating

**Remark** L: Low-level output

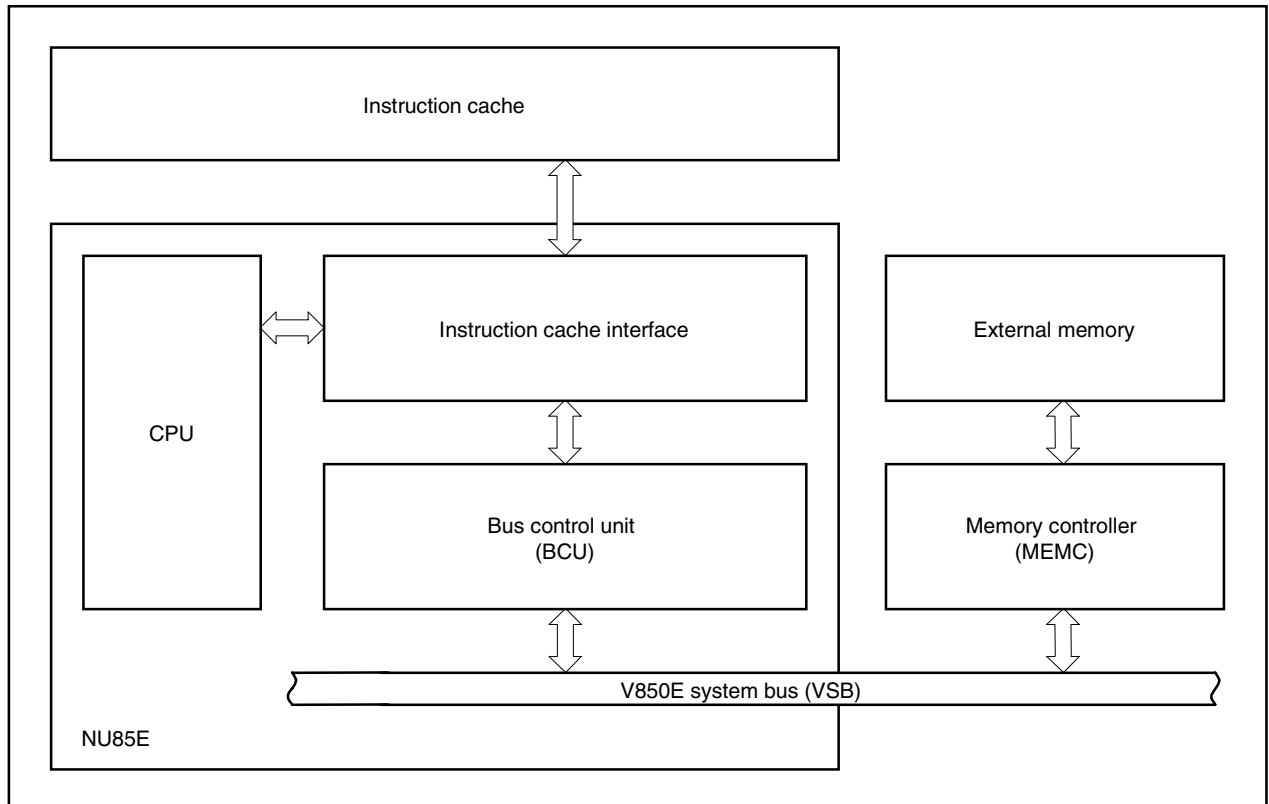
Maintained: The previous status is maintained

### 1.3 Configuration of Instruction Cache

The following two types of instruction caches are available. The NU85E can be accessed by one of these instruction caches in one cycle.

- 4 KB 2-way set-associative instruction cache (NU85E212)
- 8 KB 2-way set-associative instruction cache (NU85E213)

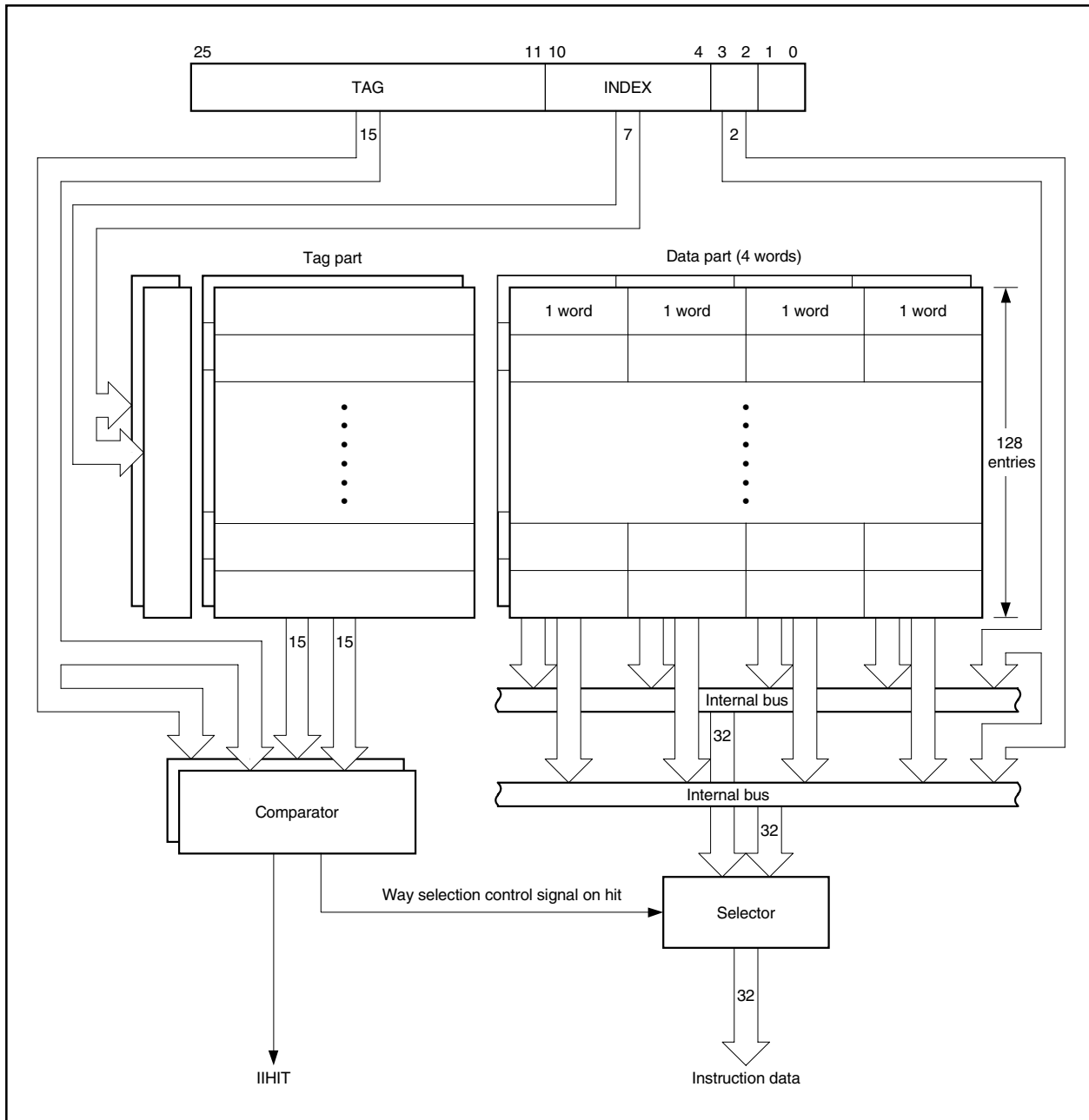
**Figure 1-2. Instruction Cache Configuration Example**



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

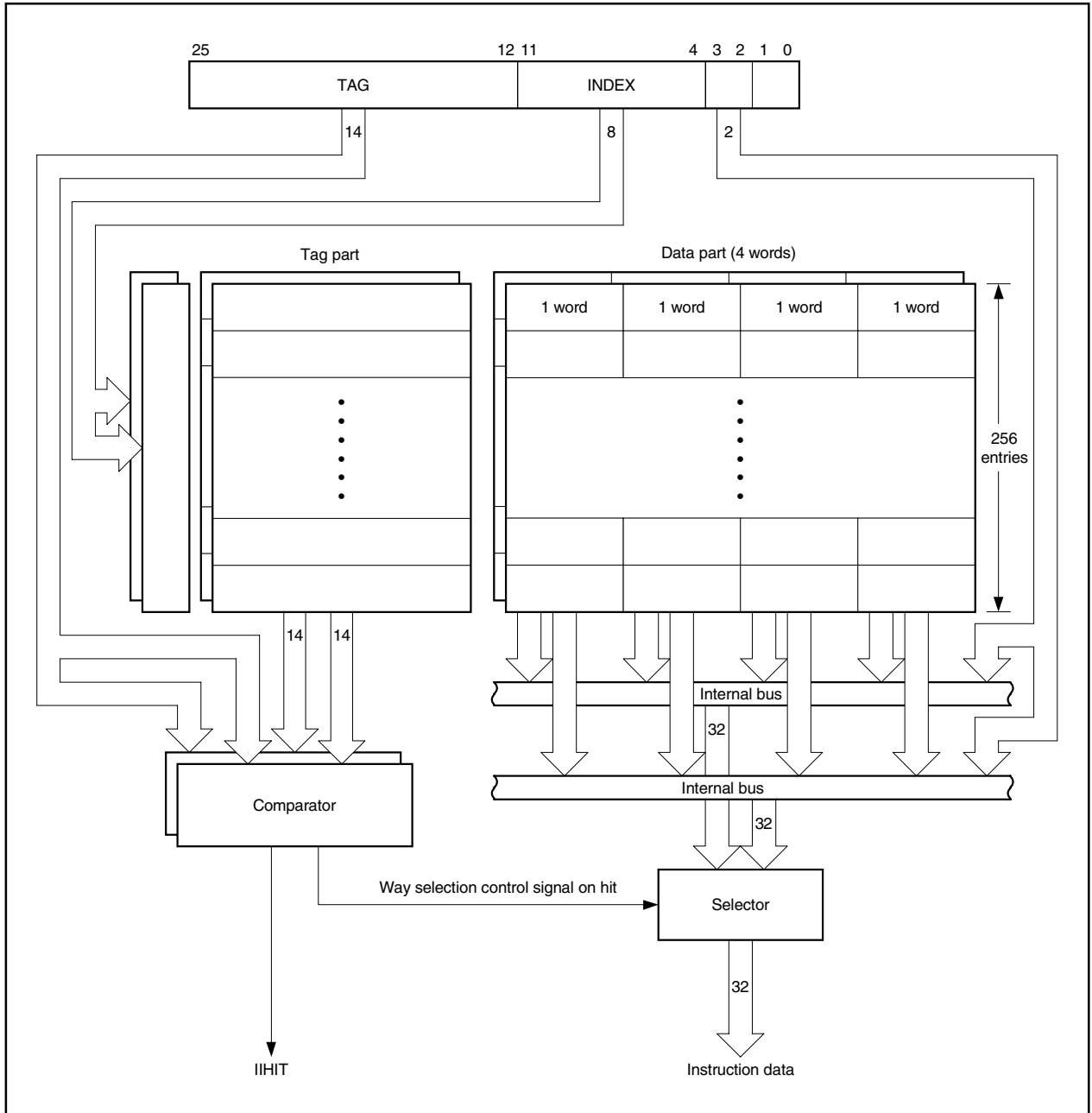




1.3.2 8 KB 2-way set-associative instruction cache

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	
FFFFFF070H	Instruction cache control register	ICC	R/W	–	–	√	0003H <sup>Note 1</sup>
FFFFFF070H	Instruction cache control register L	ICCL	R/W	√	√	–	03H <sup>Note 2</sup>
FFFFFF071H	Instruction cache control register H	ICCH	R/W	√	√	–	00H
FFFFFF074H	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 NU85E.

#### (1) Instruction cache control register (ICC)

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

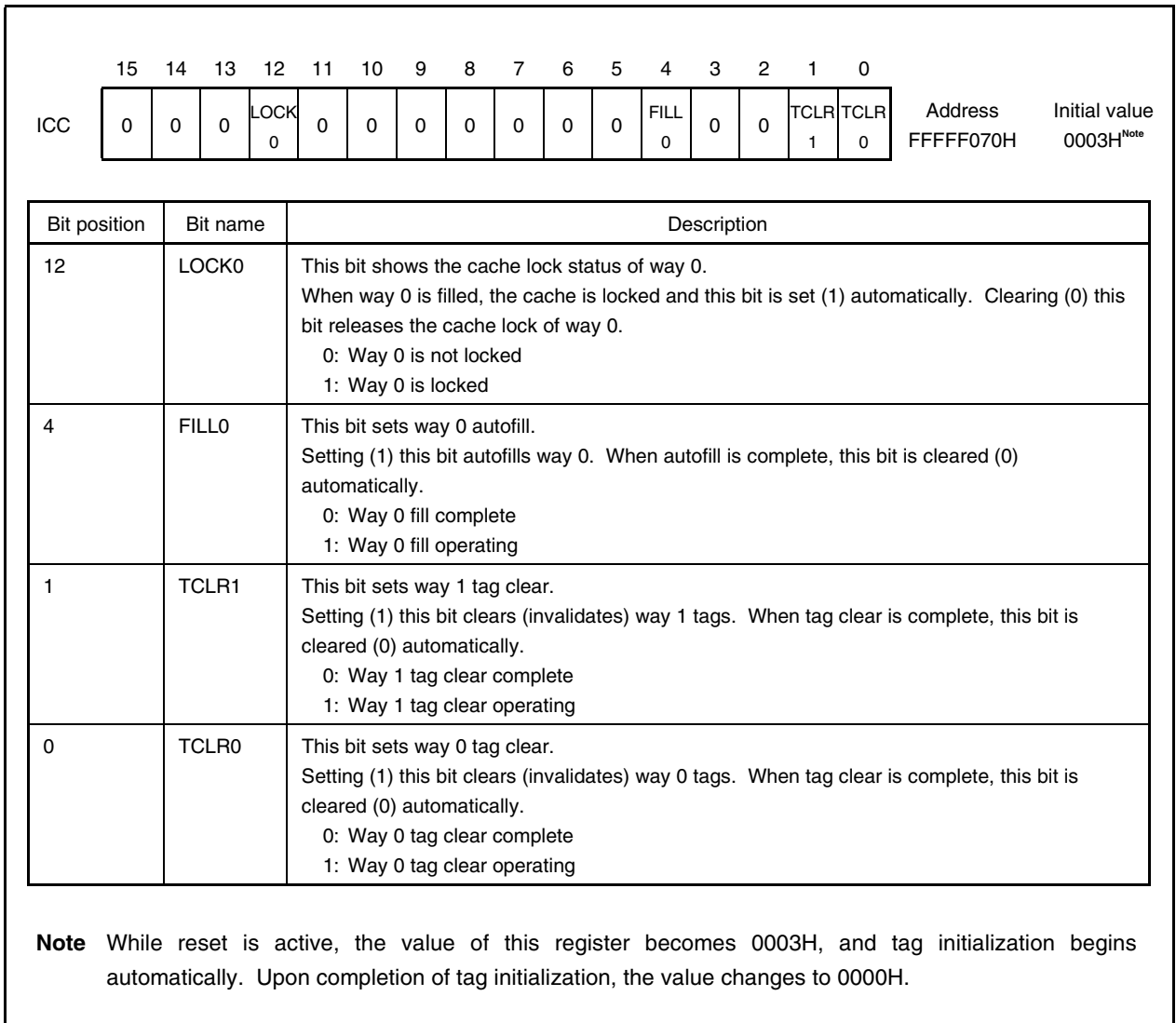
The ICC register can be read or written in 16-bit units.

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)



**(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	DATA 14	DATA 13	DATA 12	DATA 11	DATA 10	DATA 9	DATA 8	DATA 7	DATA 6	DATA 5	DATA 4	DATA 3	DATA 2	DATA 1	DATA 0	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	NU85E212: This bit sets the LSB of the tag information (bit 11 of the start address of the memory area to be autofilled). NU85E213: 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.

★ **Caution** 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).
- When clearing way 0 and way 1 individually<sup>Note</sup>:
  - (a) Set (1) the TCLR0 bit.
  - (b) Read the TCLR0 bit to confirm that this bit is cleared (0).
  - (c) Set (1) the TCLR1 bit.
  - (d) Read the TCLR1 bit to confirm that this bit is cleared (0).

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

#### 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:
                                ld.h         ICC[r0], r1
                                cmp          r0, r1
                                bnz         LOP1

```

- ★ **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.

- NU85E212: 128 clocks
- NU85E213: 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 bits, and 32-bit (1 word) data is filled in 1 clock).

- NU85E212: 512 clocks
- NU85E213: 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> Make the instruction cache setting of the BHC register of the NU85E “cacheable”.

**Cautions 1. Always input a low level to the IFIUNCHO pin of the NU85E (instruction cache enable).**

**If a low level is not input to the IFIUNCHO 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).**

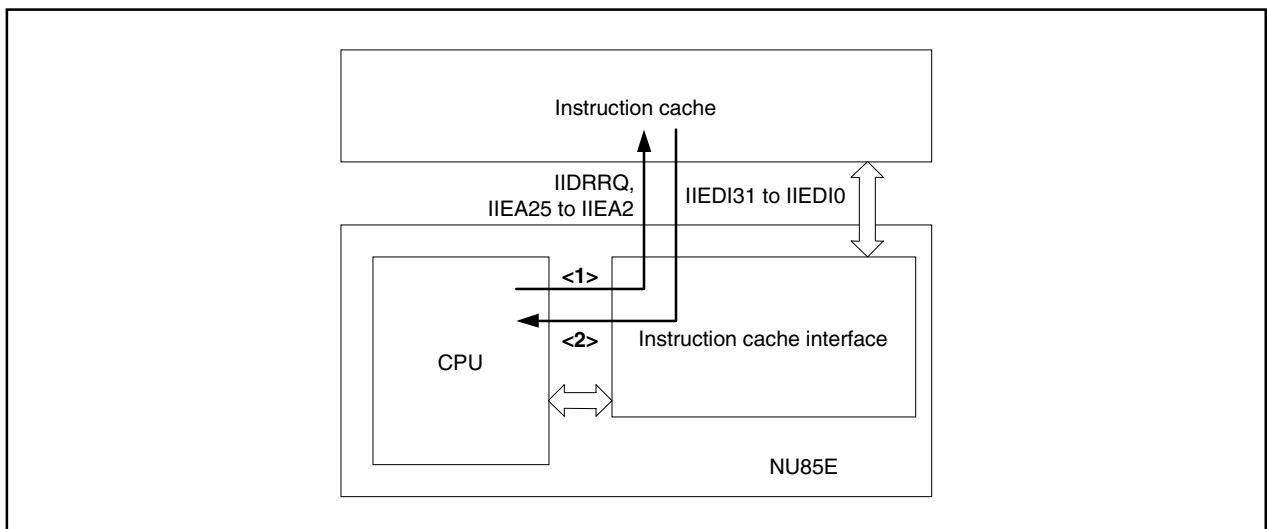
## 1.6 Operation

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 NU85E.

### 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**

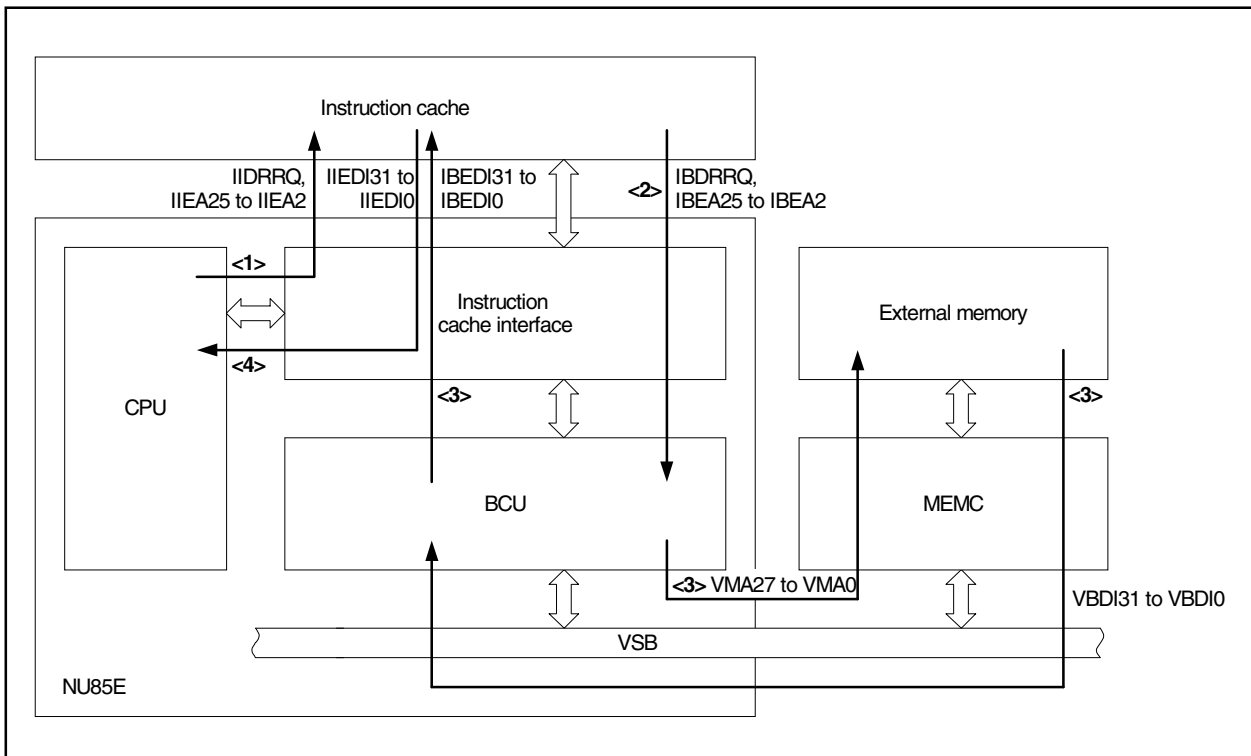


1.6.2 Operation on instruction cache miss

- <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 NU85E outputs the address (VMA27 to VMA0) 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 NU85E signals.
  3. The circles indicate the sampling timing.
  4. For details of the VSB signals (VMxxx, VDxxx), refer to the **NU85E Hardware User's Manual (A14874E)**.
  5. XXX: Undetermined state (for output), or arbitrary level (for input)

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

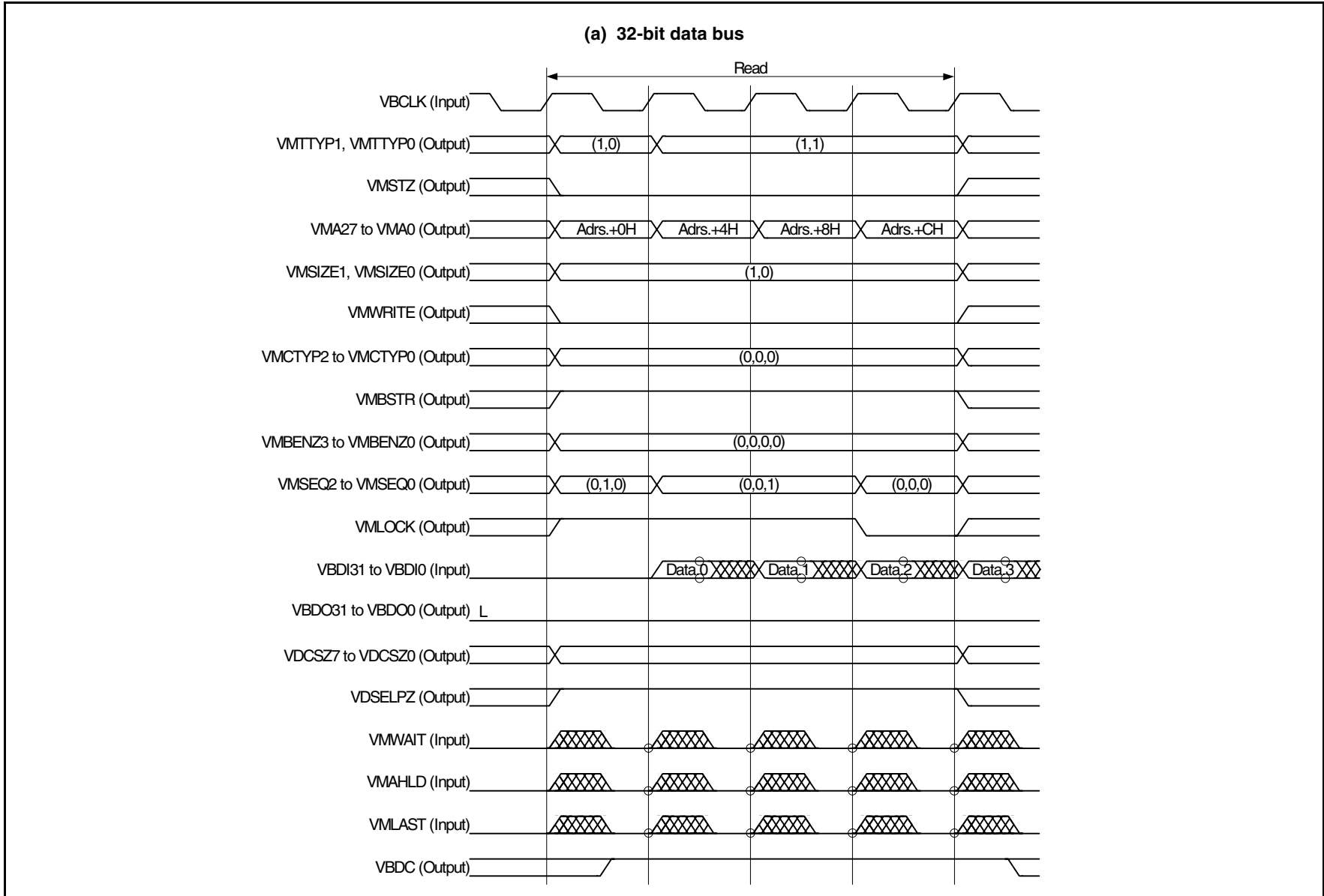
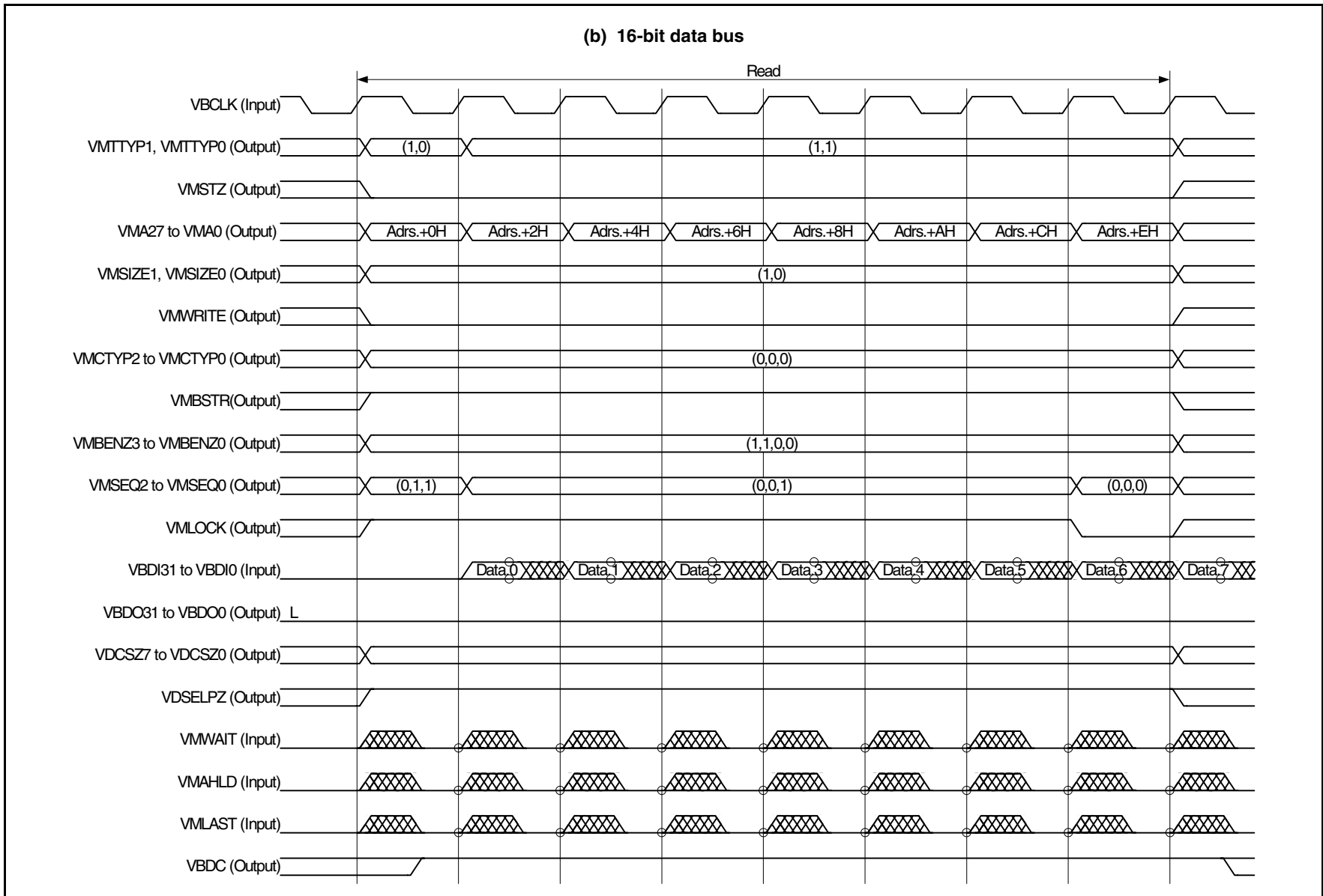


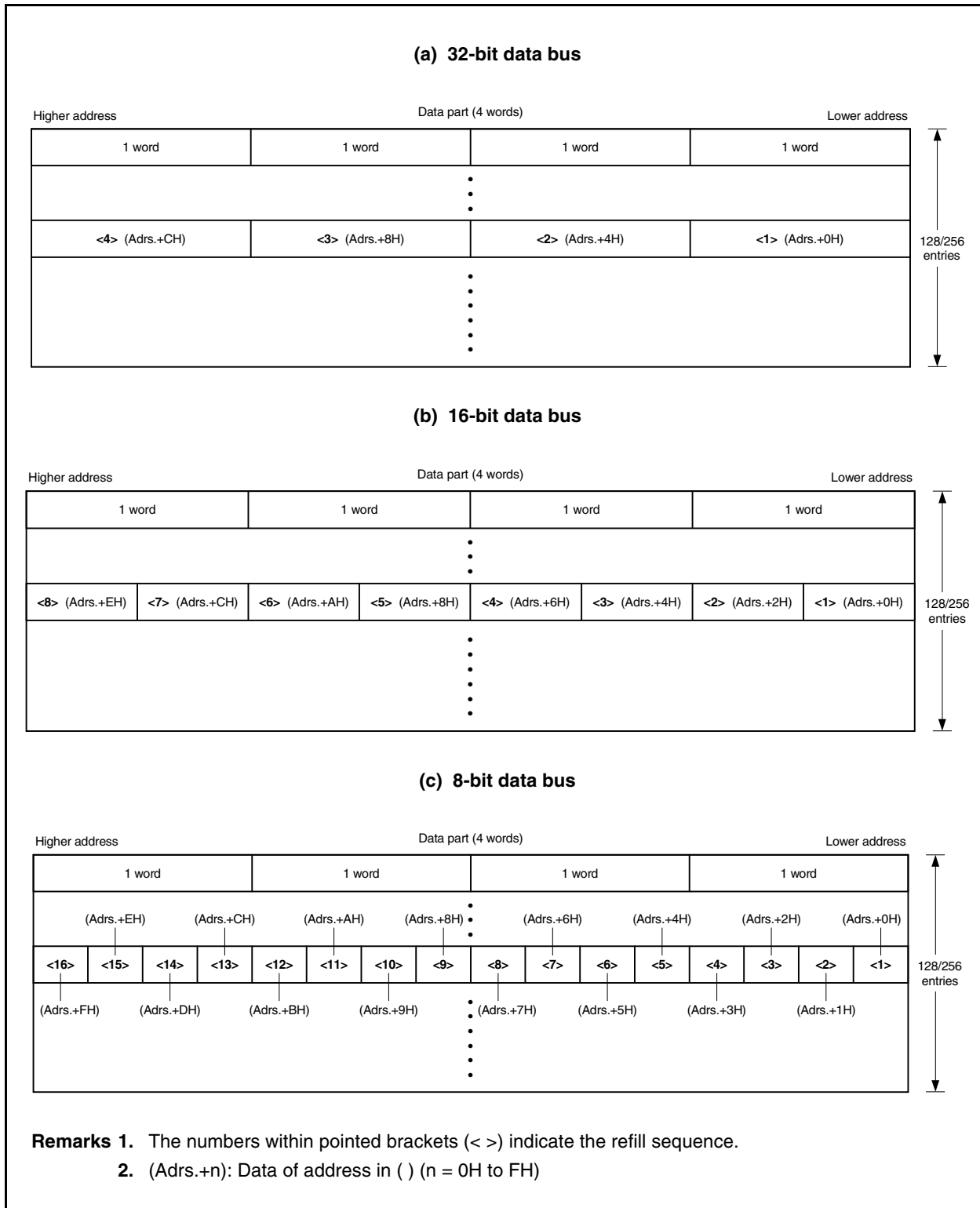
Figure 1-9. Sequential Refill Read Cycle (4R) (2/2)



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



## 1.9 Cautions

### (1) Connection to NU85E

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 NU85E to low level.

The following pins should be connected as follows (connect to pins that do not have the same pin names).

VPDR15 to VPDR0: Connect to the VPDI15 to VPDI0 pins of the NU85E.

VPDW15 to VPDW0: Connect to the VPDO15 to VPDO0 pins of the NU85E.

BUNRIIN: Connect to the BUNRIOUT pin of the NU85E.

### (2) Setting cache type selection pins

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

Pin Name	Input Level	
	NU85E212	NU85E213
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 NU85E, the VMCTYP2 to VMCTYP0 signals of the NU85E 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 NU85E 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)<sup>Note</sup>
- Instruction cache data configuration register (ICD)

**Note** Excluding bit 4

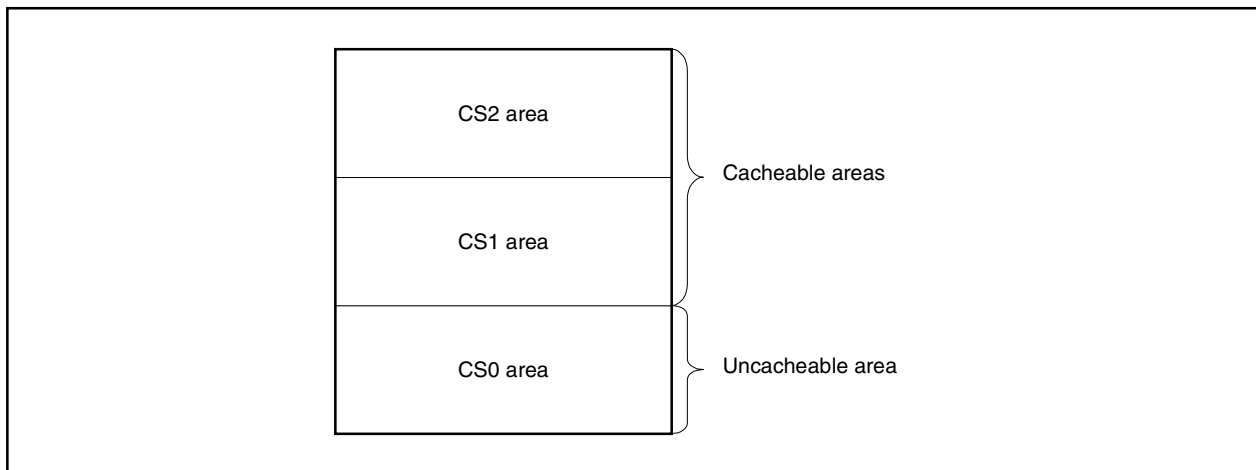
**(6) Access to memory boundary**

If adjacent chip select (CS<sub>n</sub>) 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 CS<sub>0</sub> area to CS<sub>1</sub> area, access is possible only by using a branch instruction.
- From CS<sub>1</sub> area to CS<sub>2</sub> area, continuous access is possible.

**Figure 1-11. Cache Area Setting Example**

**(7) Setting BHC register of NU85E**

In the case of CS<sub>n</sub> 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 CS<sub>n</sub> areas for which no instruction for setting the BHC register exists.

For example, if a BHC register setting instruction exists in the CS<sub>0</sub> area, the instruction cache of the CS<sub>0</sub> area cannot be set (cache enable/disable settings). In this case, only the instruction cache settings for areas CS<sub>1</sub> to CS<sub>7</sub> are possible.

However, instruction cache settings for all CS<sub>n</sub> 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)

**(8) 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.

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

In the NU85E 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<sup>Note</sup> 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 NU85E252 and NU85E263 are data cache memories for the NU85E.

They can be directly connected to the data cache interface incorporated in the NU85E.

The following two types of data cache are available.

- NU85E252 ... 4 KB directly mapped data cache (4 words  $\times$  256 entries = 4 KB)
- NU85E263 ... 8 KB 2-way set-associative data cache (4 words  $\times$  256 entries  $\times$  2 ways = 8 KB)

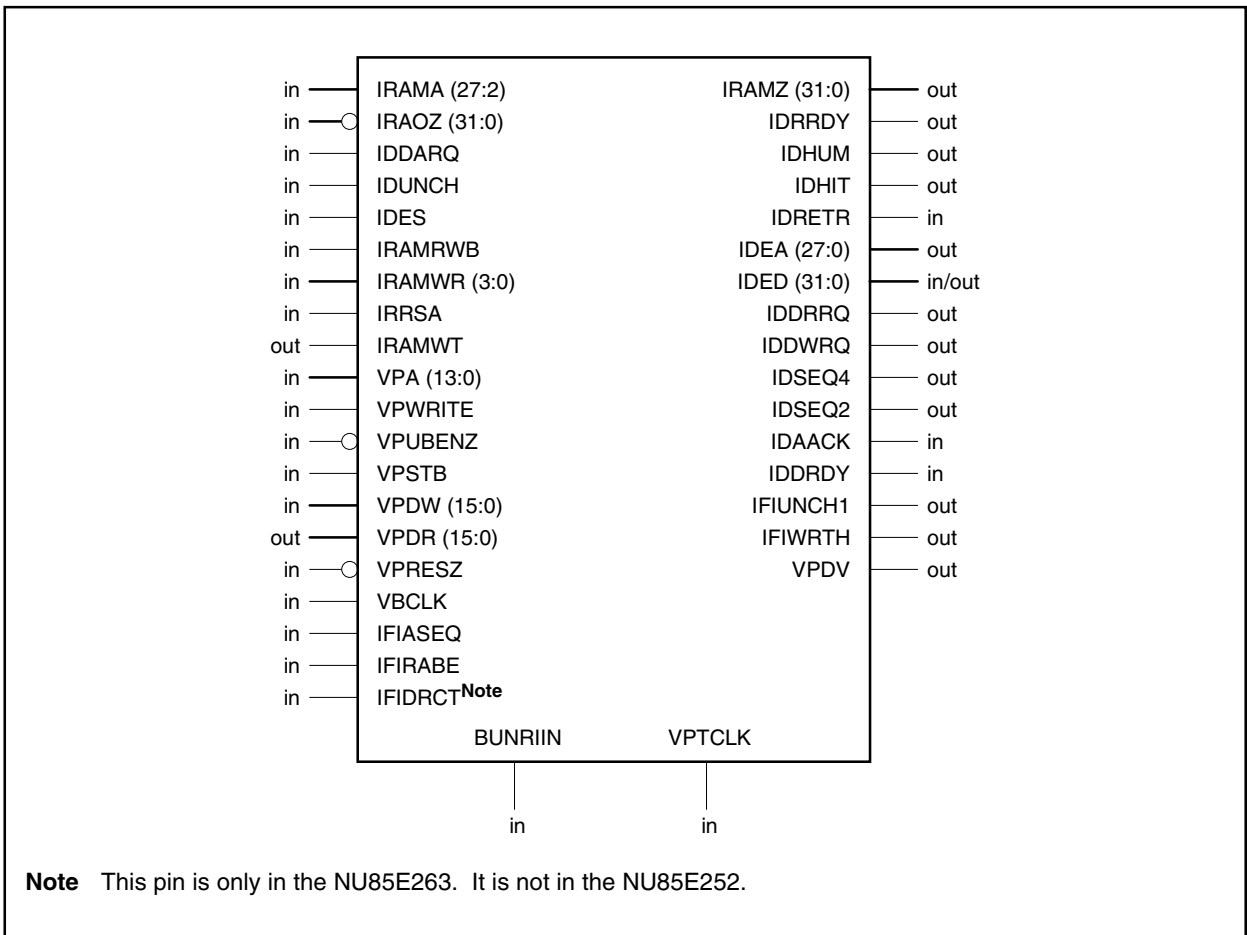
#### 2.1.1 Features

- 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 NU85E 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.



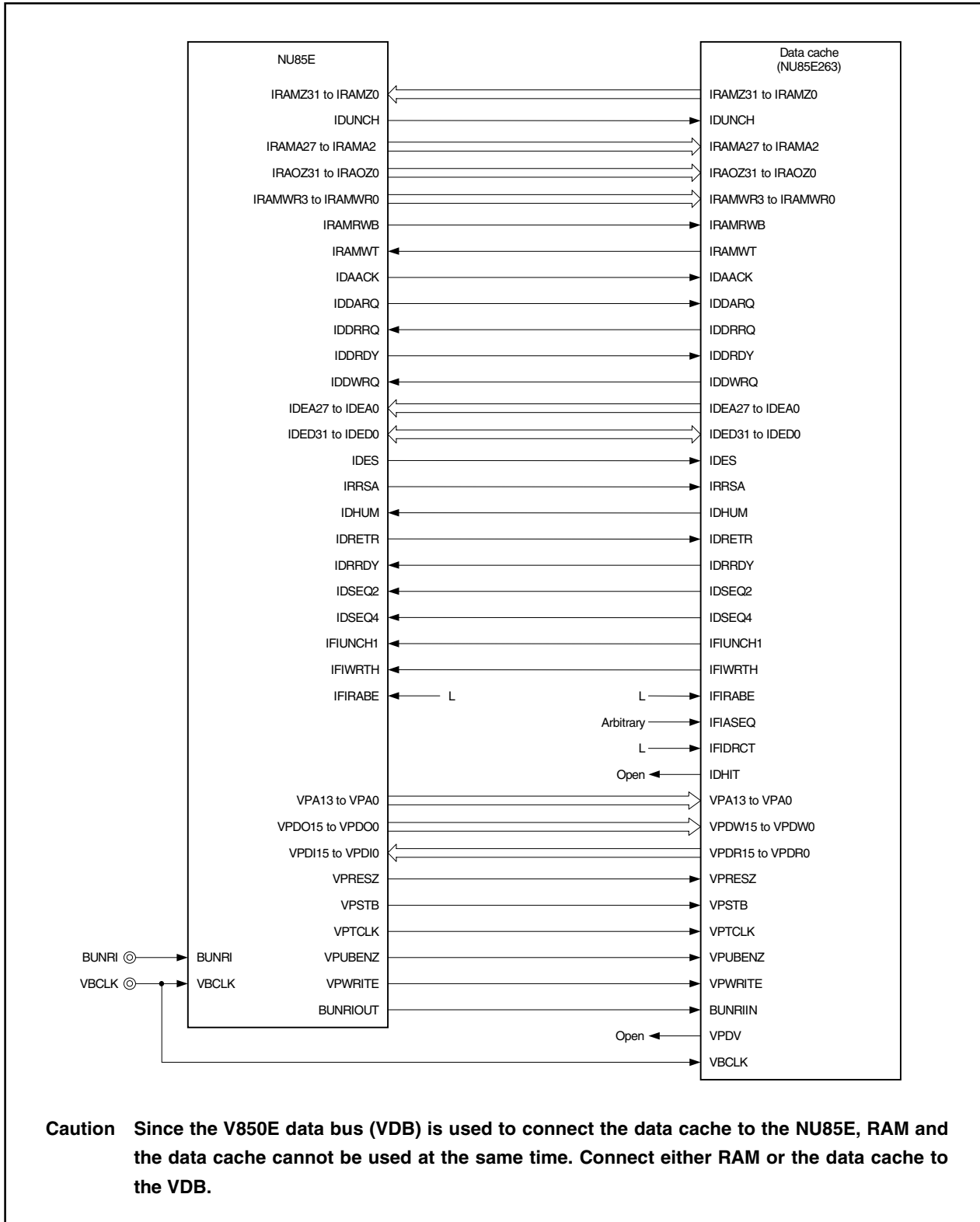
2.1.2 Symbol diagram



2.1.3 NU85E connection example

The following figure shows an example of the connection of a data cache to the NU85E.

Figure 2-1. NU85E and Data Cache Connection Example



**Caution** Since the V850E data bus (VDB) is used to connect the data cache to the NU85E, 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	
NU85E connection pins	IRAMA27 to IRAMA2	Input	Input address from NU85E
	IRAOZ31 to IRAOZ0	Input	Input data from NU85E
	IDDARQ	Input	Input read/write access request from NU85E
	IDUNCH	Input	Input uncache status
	IDES	Input	NEC reserved pin (connect to IDES pin of NU85E)
	IRAMRWB	Input	Input read/write status from NU85E
	IRAMWR3 to IRAMWR0	Input	Input write enable from NU85E
	IRRSA	Input	Input VDB hold status
	IRAMWT	Output	Output wait to NU85E
	IRAMZ31 to IRAMZ0	Output	Output data to NU85E
	IDRRDY	Output	Output read data ready to NU85E
	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
	IDRRRQ	Output	Output VSB read operation request to NU85E
	IDDWRQ	Output	Output VSB write operation request to NU85E
	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 NU85E
	IFIUNCH1	Output	Output data cache setting to NU85E
	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)
	VPDW15 to VPDW0	Input	Input data (for NPB)
	VPDR15 to VPDR0	Output	Output data (for NPB)
	VPDV	Output	Output data bus direction control (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

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 <sup>Note</sup>	Input	NEC reserved pin (input a low level)
Status pin	IDHIT	Output	Output tag hit status
Test mode pins	BUNRIIN	Input	Input normal/test mode selection
	VPTCLK	Input	Input clock for test

**Note** This pin is only in the NU85E263. It is not in the NU85E252.

## 2.2.2 Explanation of pin functions

### (1) NU85E connection pins

#### (a) IRAMA27 to IRAMA2 (input)

IRAMA27 to IRAMA2 constitute a bus that inputs an address from the NU85E.

#### (b) IRAOZ31 to IRAOZ0 (input)

IRAOZ31 to IRAOZ0 constitute a bus that inputs data from the NU85E.

#### (c) IDDARQ (input)

IDDARQ inputs a read or write access request from the NU85E.

#### (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 NU85E is accessed.

#### (e) IDES (input)

IDES is reserved for NEC. Be sure to connect it to the IDES pin of the NU85E.

#### (f) IRAMRWB (input)

IRAMRWB inputs read or write status from the NU85E.

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 NU85E.

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 NU85E.

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 NU85E.

**(k) IDRRDY (output)**

IDRRDY outputs a read data ready to the NU85E.

**(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 NU85E 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 NU85E.

This bus outputs the address the NU85E should access when a miss occurs.

**(o) IDDED31 to IDDED0 (I/O)**

IDDED31 to IDDED0 constitute a bus that inputs and outputs data from and to the NU85E.

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

**(p) IDRRQ, IDWRQ, IDSEQ4, and IDSEQ2 (output)**

IDRRQ, IDWRQ, IDSEQ4, and IDSEQ2 output operation type settings to the NU85E.

IDRRQ	IDWRQ	IDSEQ4	IDSEQ2	Operation Type
1	0	1	0	4-word sequential read
1	0	0	1	2-word sequential read
1	0	0	0	1-word read
0	1	1	0	4-word sequential write
0	1	0	1	2-word sequential write
0	1	0	0	1-word write
1	1	1	1	1-word write
1	1	1	0	1-halfword write
1	1	0	0	1-byte write
Other than above				Setting prohibited

**Remark** 0: Low-level output  
 1: High-level output

**(i) IDRRQ (output)**

IDRRQ outputs a VSB read operation request to the NU85E.

**(ii) IDWRQ (output)**

IDWRQ outputs a VSB write operation request to the NU85E.

**(iii) IDSEQ4, IDSEQ2 (output)**

IDSEQ4 and IDSEQ2 output read and write operation type settings to the NU85E.

**(q) IDAACK (input)**

IDAACK inputs an acknowledge.

This signal is input when the NU85E recognizes IDEA27 to IDEA0 signals output from the data cache.

**(r) IDRDY (input)**

IDRDY inputs a read data ready from the NU85E.

This is input when the NU85E 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 NU85E.

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 NU85E.

**(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 (NPB pins)**

Refer to the **NU85E Hardware User's Manual (A14874E)**.

**(v) VPDR15 to VPDR0 (output)**

VPDR15 to VPDR0 output data to the NU85E.

Connect these pins to the VPDI15 to VPDI0 pins of the NU85E.

**(w) VPDW15 to VPDW0 (input)**

VPDW15 to VPDW0 input data from the NU85E.

Connect these pins to the VPDO15 to VPDO0 pins of the NU85E.

**(x) VPDV (output)**

VPDV outputs data bus direction control for the NPB.

When connecting to the NU85E, leave this pin open since it is not used.

**(y) VPRESZ (input)**

VPRESZ inputs a reset.

**(z) 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 (input)**

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

The IFIDRCT pin is only in the NU85E263. It is not in the NU85E252.

**(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) BUNRIIN (input)

BUNRIIN is an input pin that selects normal or test mode.  
Connect to the BUNRIOUT pin of the NU85E.

(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
NU85E 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
	IDRRRQ	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
	VPDR15 to VPDR0	L	Maintained	Operating	Operating
	VPDV	Undefined	Maintained	Operating	Operating
Status pin	IDHIT	Undefined	Maintained	Operating	Operating

**Remark** H: High-level output  
L: Low-level output  
Maintained: The previous status is maintained

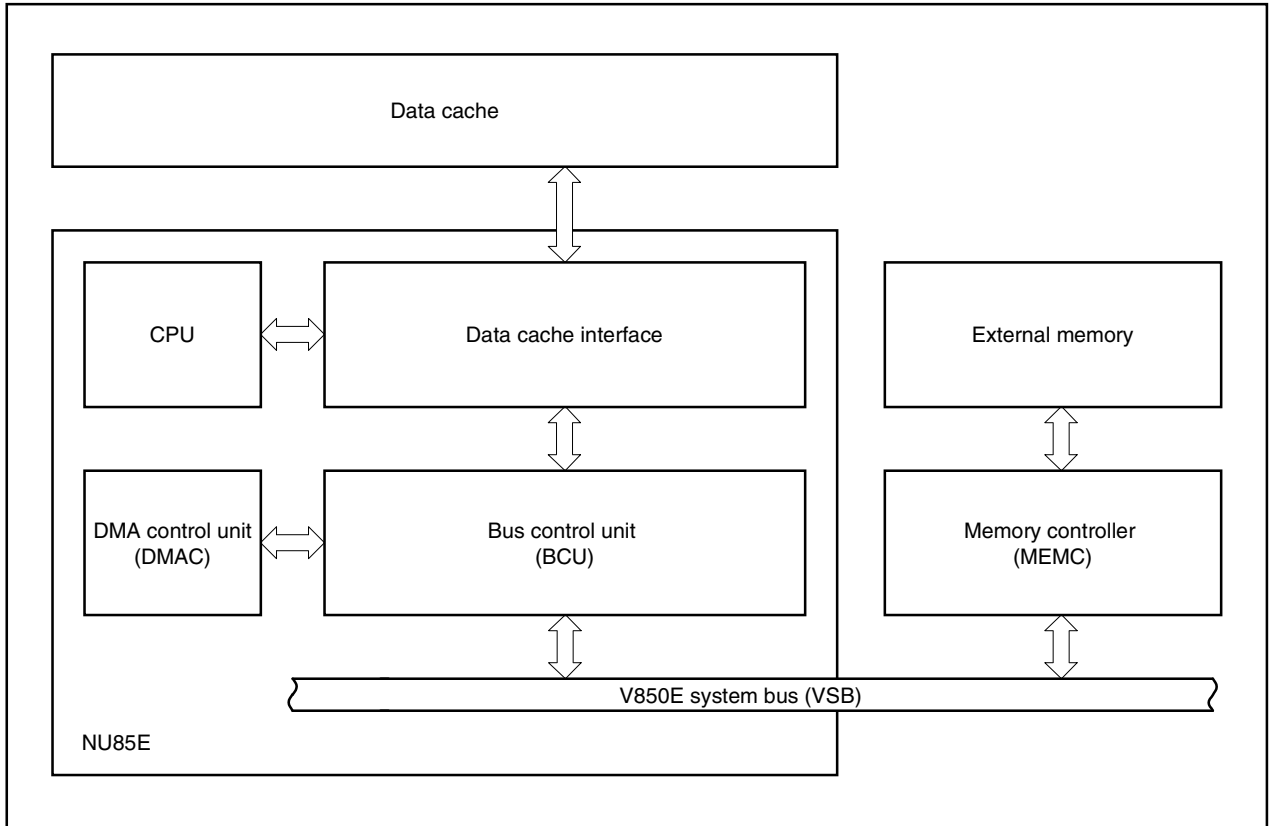


## 2.3 Configuration of Data Cache

The following two types of data caches are available. The NU85E can be accessed by one of these data caches in one cycle.

- 4 KB directly mapped data cache (NU85E252)
- 8 KB 2-way set-associative data cache (NU85E263)

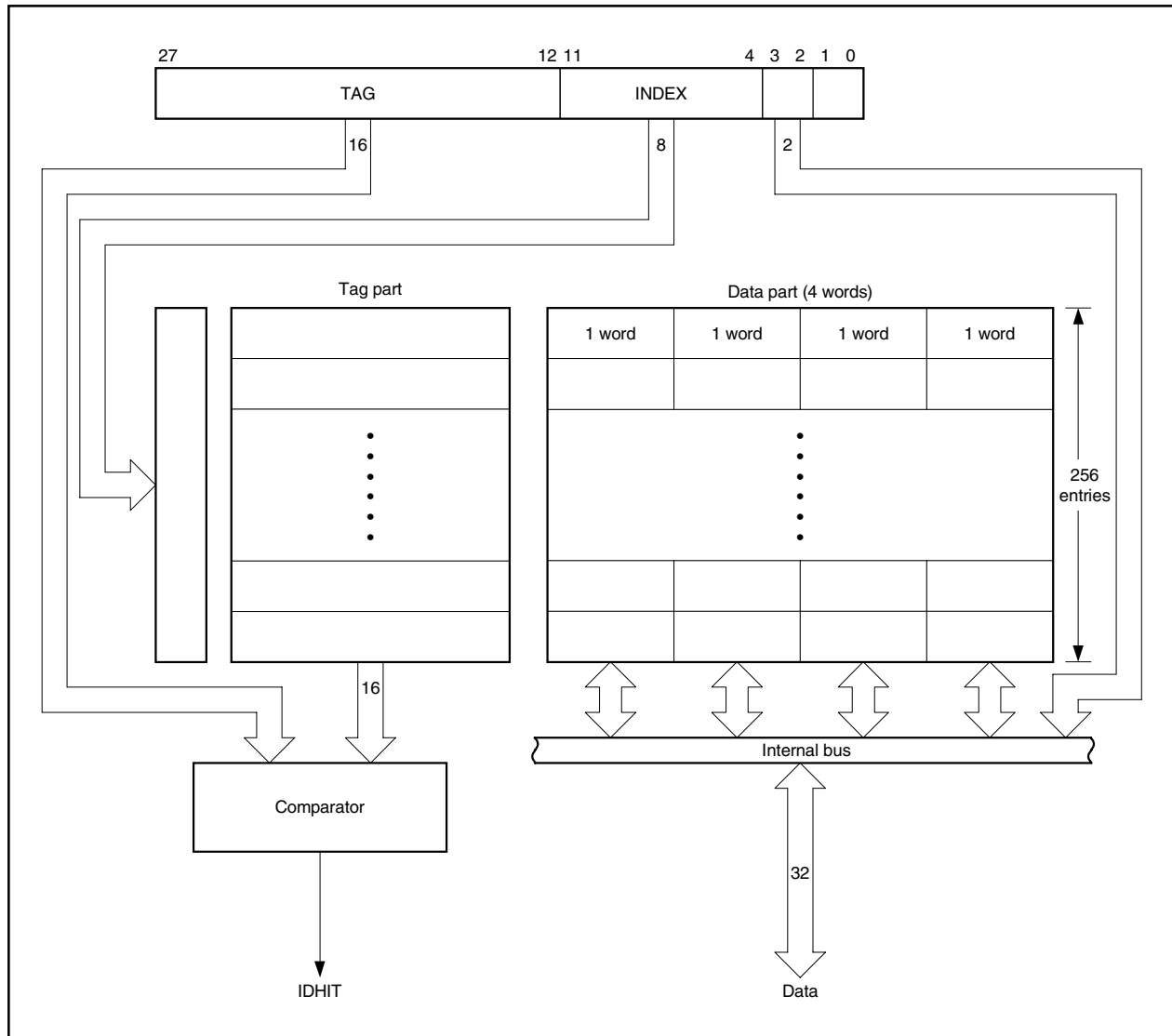
**Figure 2-2. Data Cache Configuration Example**



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.

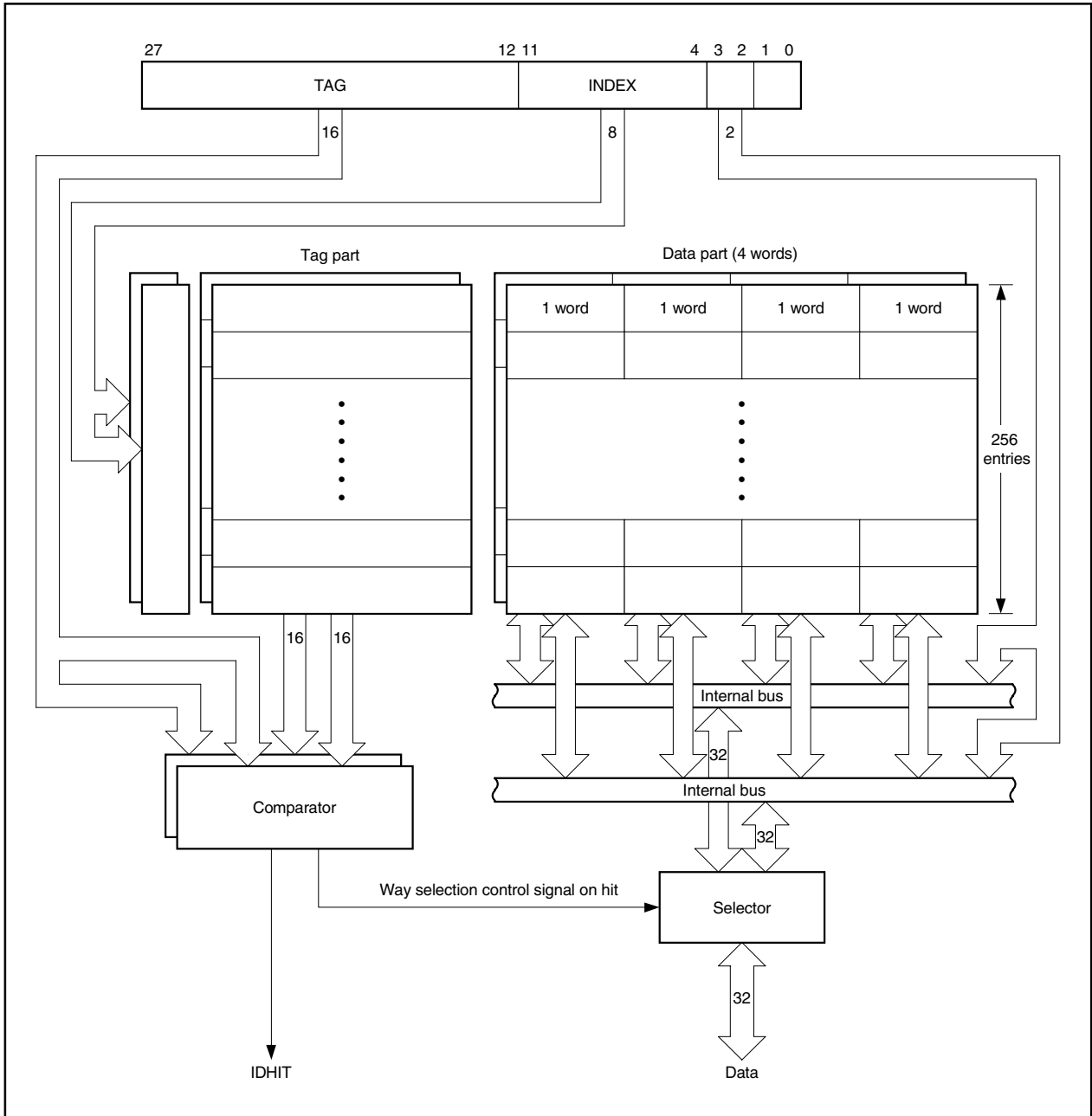
Figure 2-3. Configuration of 4 KB Directly Mapped Data Cache



**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 <sup>Note</sup>
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 NU85E.

#### (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 NU85E263. Be sure to set these bits to 0 in the NU85E252.
  3. Be sure to set the data cache enable area (using the NU85E'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 NU85E 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		
DCC	0	0	DC13	DC12	DC11	DC10	0	0	0	0	DC05	DC04	0	0	DC01	DC00	Address FFFFFF078H	Initial value 0003H

Bit position	Bit name	Description															
13	DC13	<p>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.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">DC13</th> <th style="width: 40%;">Cache lock</th> <th style="width: 50%;">Tag clear/fill</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Way 1 is not locked</td> <td>Way 1 tag clear function is valid</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Way 1 is locked</td> <td>Way 1 tag fill function is valid</td> </tr> </tbody> </table>	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						
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															
12	DC12	<p>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.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">DC12</th> <th style="width: 40%;">Cache lock</th> <th style="width: 50%;">Tag clear/fill</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Way 0 is not locked</td> <td>Way 0 tag clear function is valid</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Way 0 is locked</td> <td>Way 0 tag fill function is valid</td> </tr> </tbody> </table>	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						
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															
11, 10	DC11, DC10	<p>These bits set the operating mode.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">DC11</th> <th style="width: 10%;">DC10</th> <th style="width: 80%;">Operating mode</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>Cache access disabled</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>Write through mode</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>Writeback mode (write allocate disabled)</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>Writeback mode (write allocate enabled)</td> </tr> </tbody> </table> <p><b>Remark</b> 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.</p>	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)
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)															
5	DC05	<p>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</p>															

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 <b>Remark</b> 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 <b>Remark</b> 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		
DCD	DD15	DD14	DD13	DD12	DD11	DD10	DD9	DD8	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0	Address	Initial value
																	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 NU85E “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 NU85E, 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 NU85E

- <1> Make the data cache setting of the BHC register of the NU85E “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 NU85E “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 NU85E.

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	
	Write		Hit	–
		Miss	–	
Writeback mode (write allocate disabled)	Read	Hit	–	
		Miss	Clean data	Sequential
			Dirty data	Sequential
		Write		Hit
	Miss		–	
	Writeback mode (write allocate enabled)	Read	Hit	–
Miss			Clean data	Sequential
			Dirty data	Sequential
Write				Hit
		Miss	Clean data	Sequential
			Dirty data	Sequential

Data cache operation is described below for each operating mode.

### 2.6.1 Write through mode

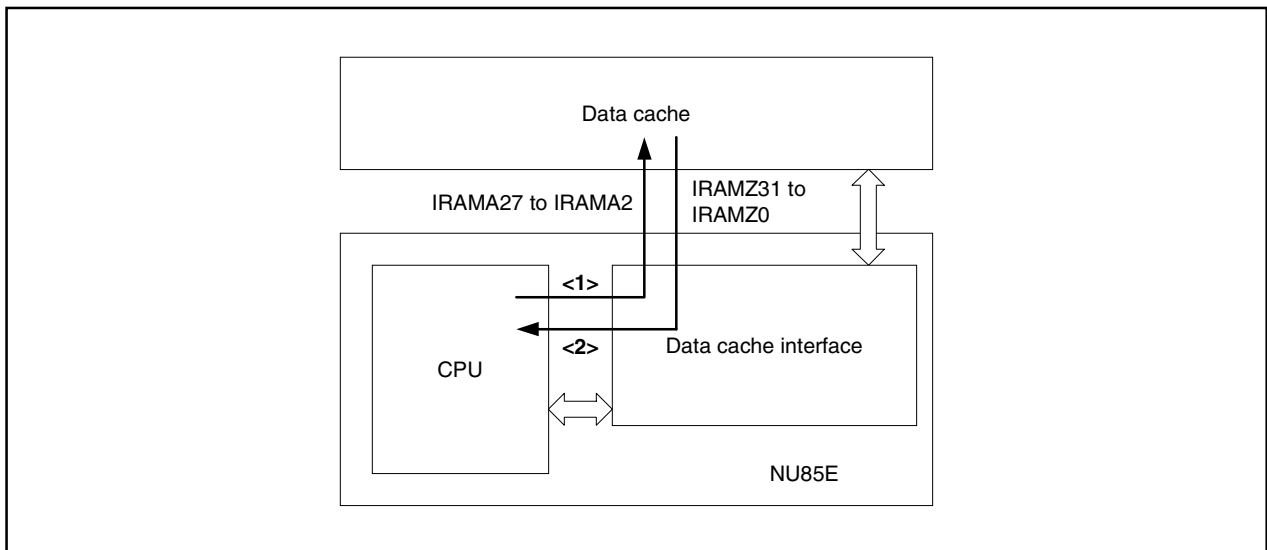
#### (1) On a read

##### (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.

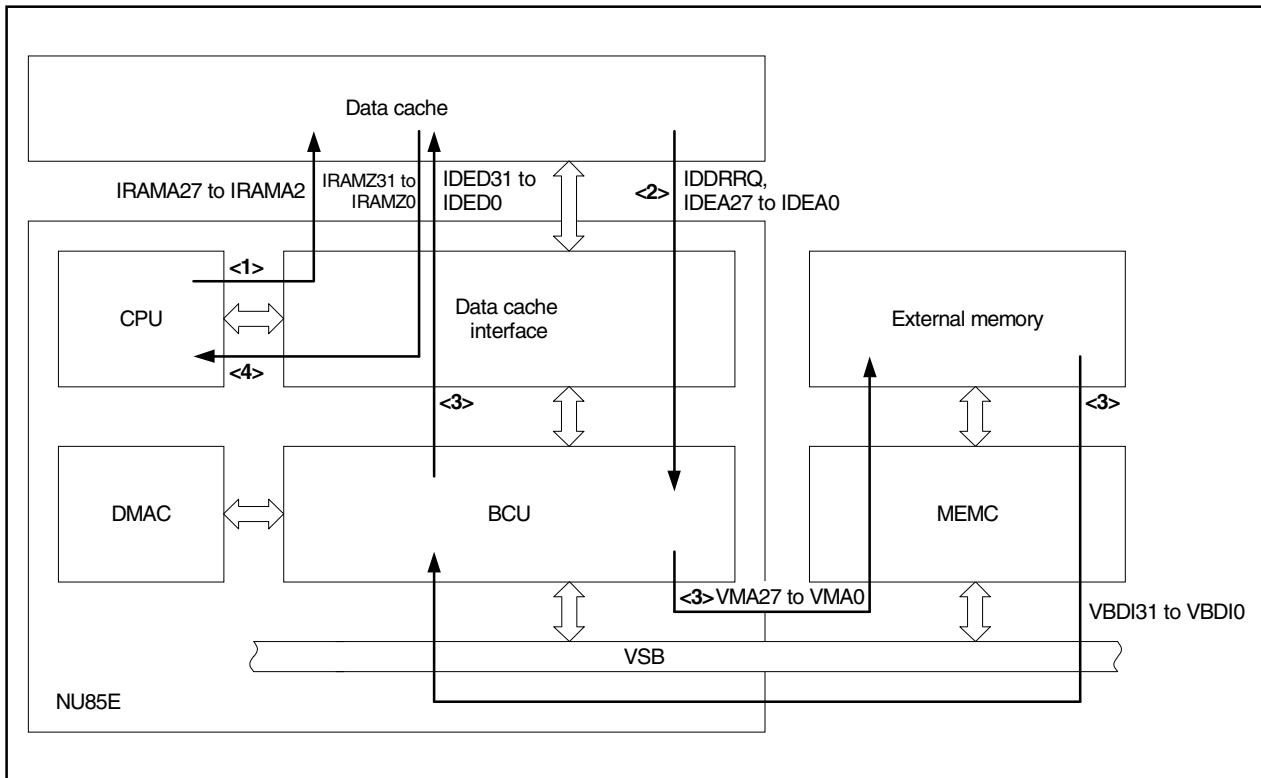
**Figure 2-7. Operation on Data Cache Hit (Write Through Mode, Read)**



(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 (IDRRQ) and the address to be read (IDEA27 to IDEA0) from the data cache to the BCU.
- <3> The BCU of the internal NU85E outputs the address (VMA27 to VMA0) 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)

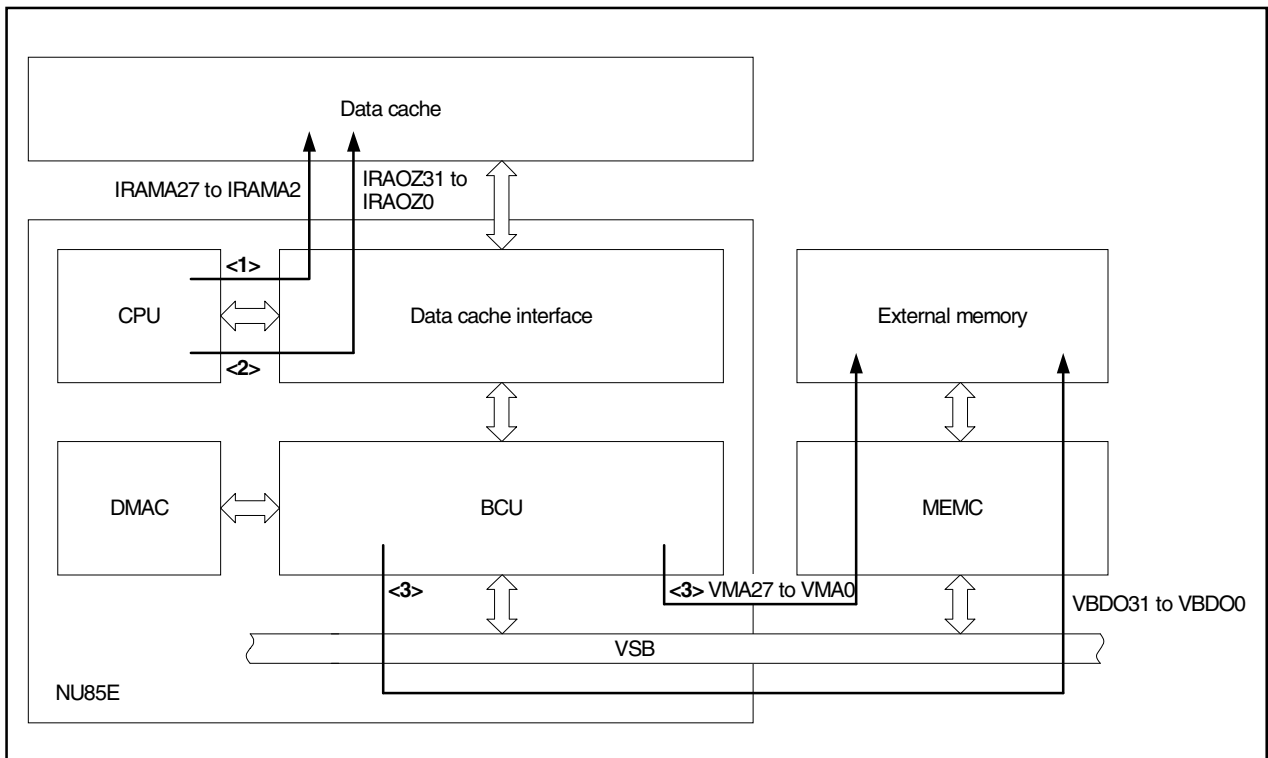


(2) On a write

(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 NU85E outputs the address (VMA27 to VMA0) to external memory via the VSB and writes the same data as IRAOZ31 to IRAOZ0 to external memory by passing through VBDO31 to VBDO0.

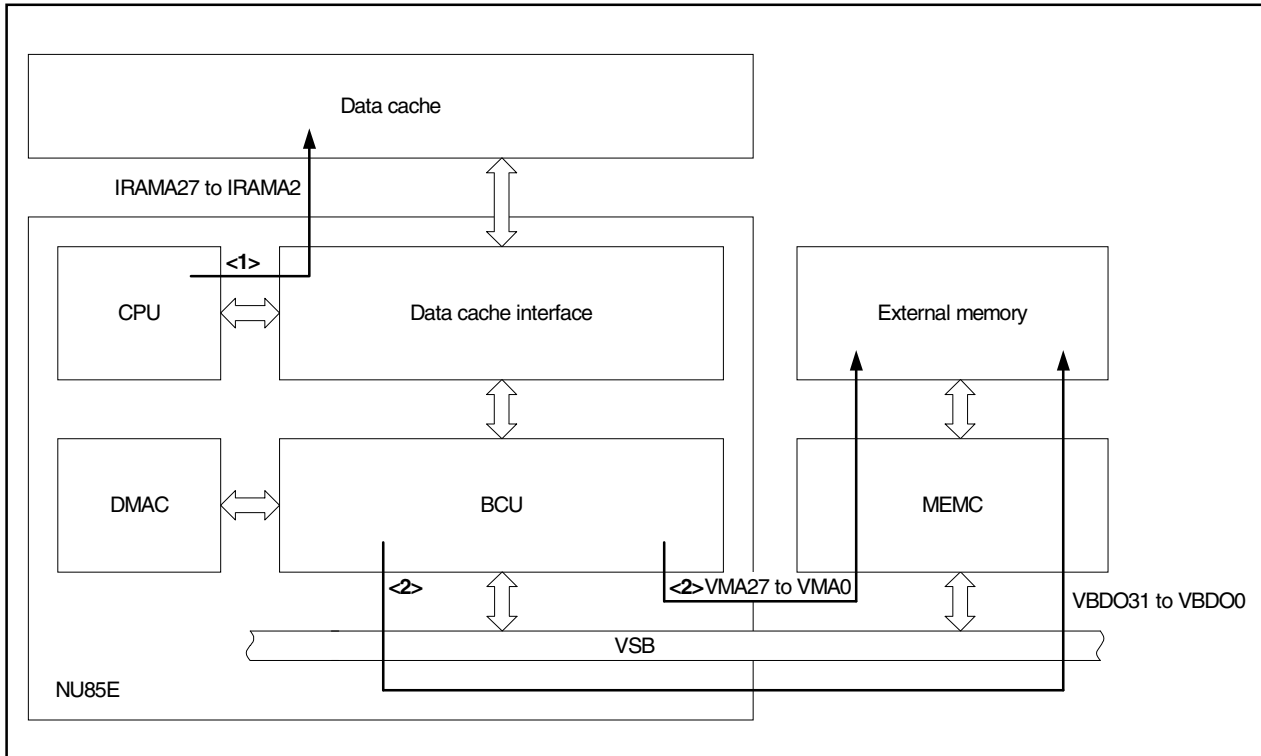
Figure 2-9. Operation on Data Cache Hit (Write Through Mode, Write)



**(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 NU85E outputs the address (VMA27 to VMA0) to external memory via the VSB and writes the data to be written to external memory by passing through VBDO31 to VBDO0.

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



## 2.6.2 Writeback mode (write allocate disabled)

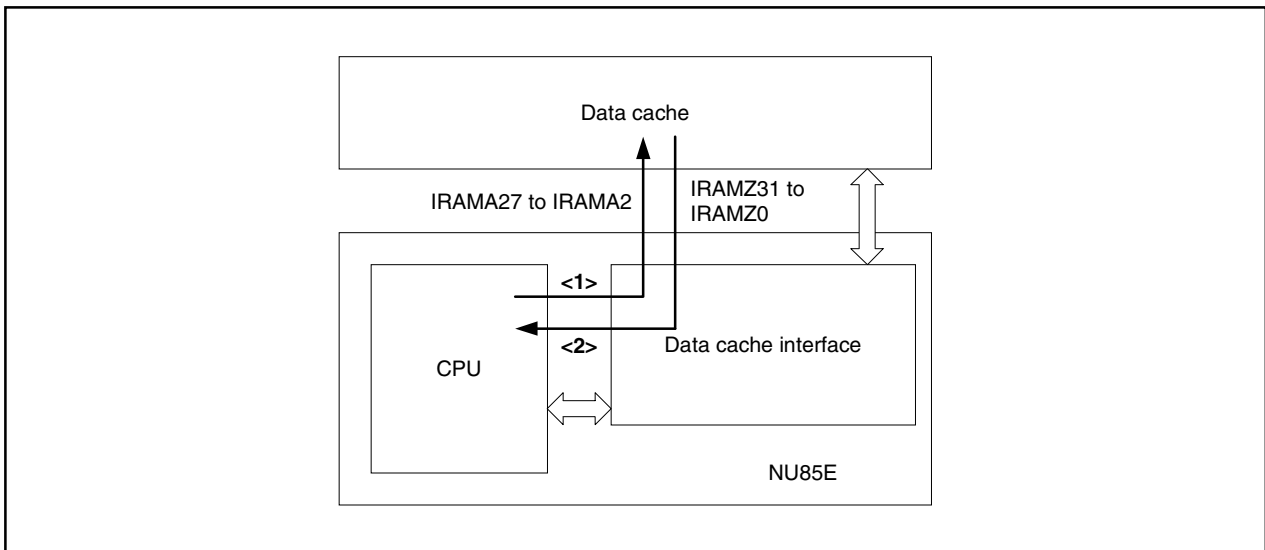
### (1) On a read

#### (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)**

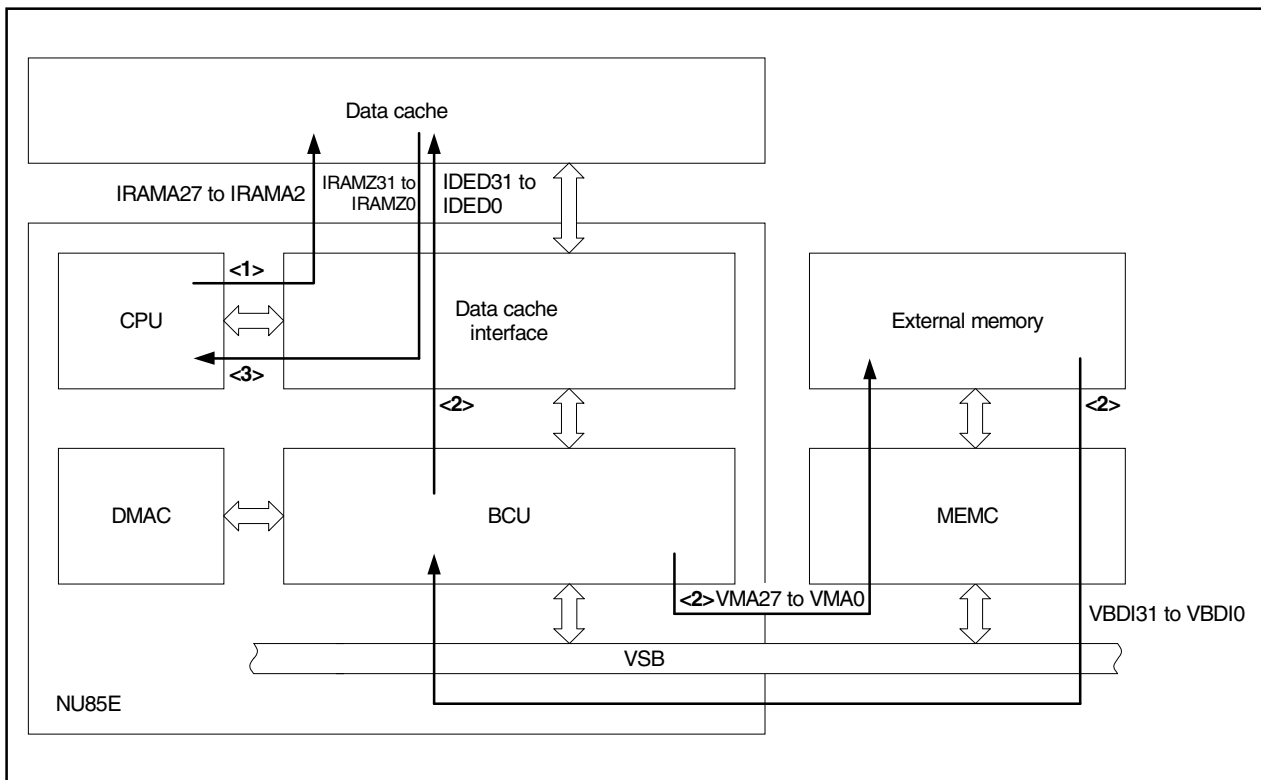


(b) Data cache miss

(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 NU85E outputs the address to be read (VMA27 to VMA0) 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)

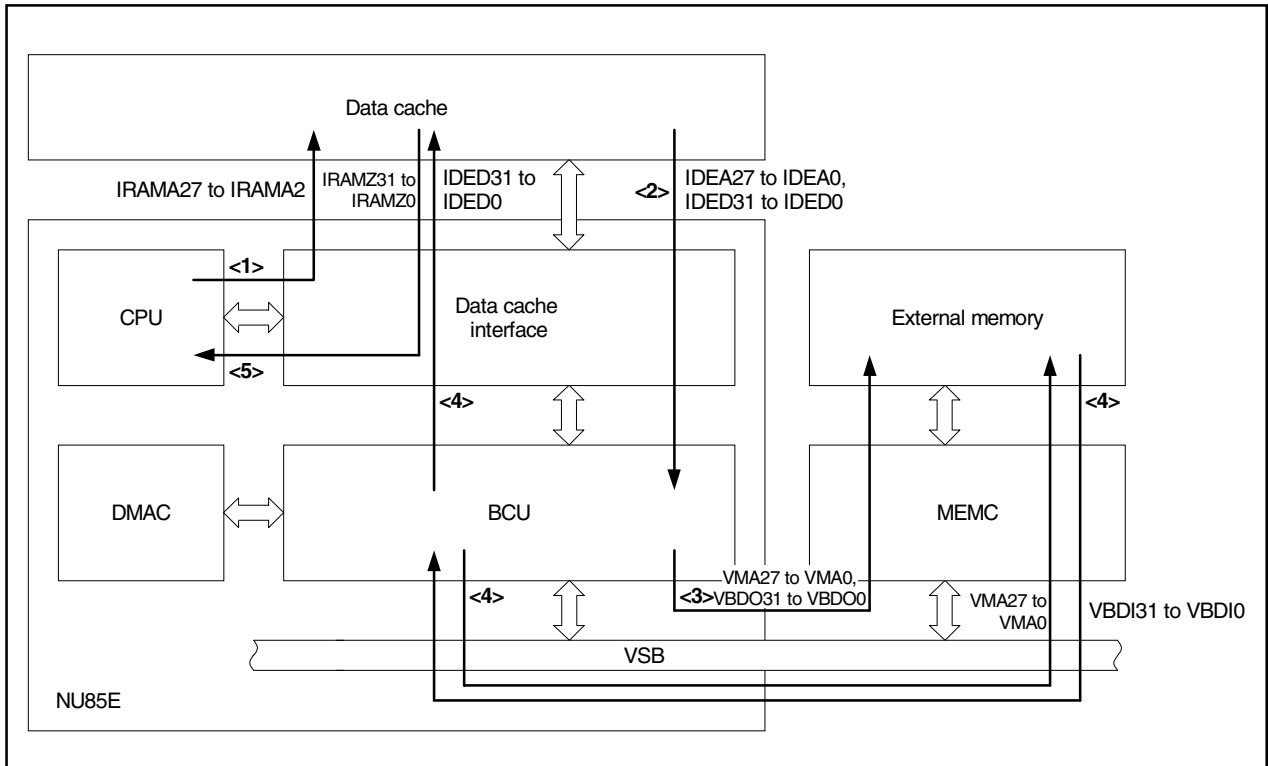




(ii) When data being replaced is dirty data

- <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 IDDED31 to IDDED0.
- <3> The BCU of the internal NU85E outputs the address (VMA27 to VMA0) 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 (VMA27 to VMA0) 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)

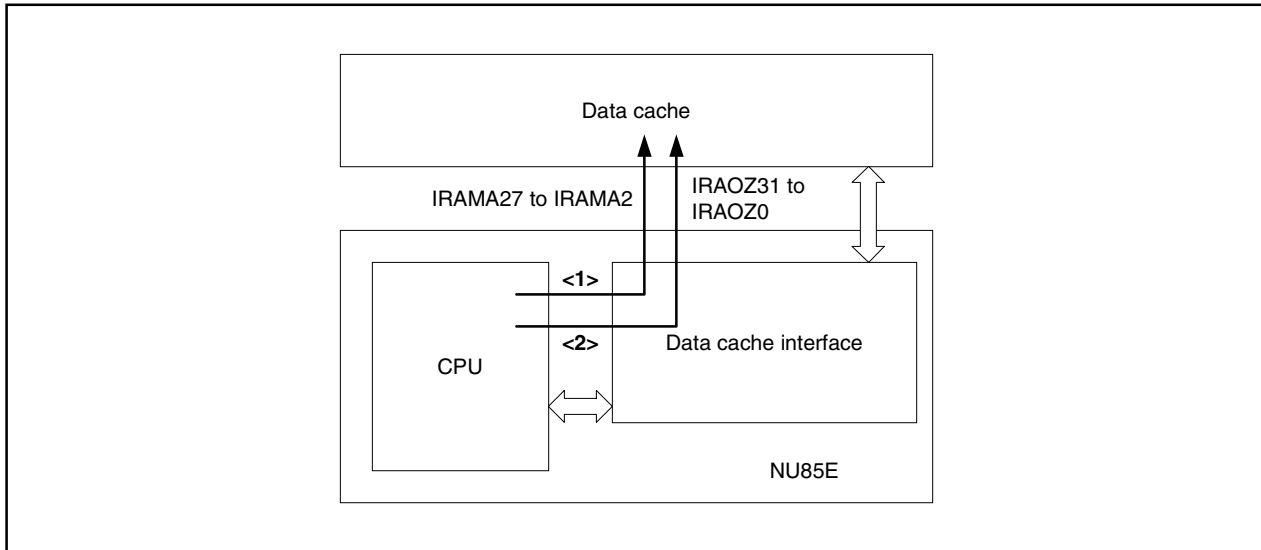


(2) On a write

(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.

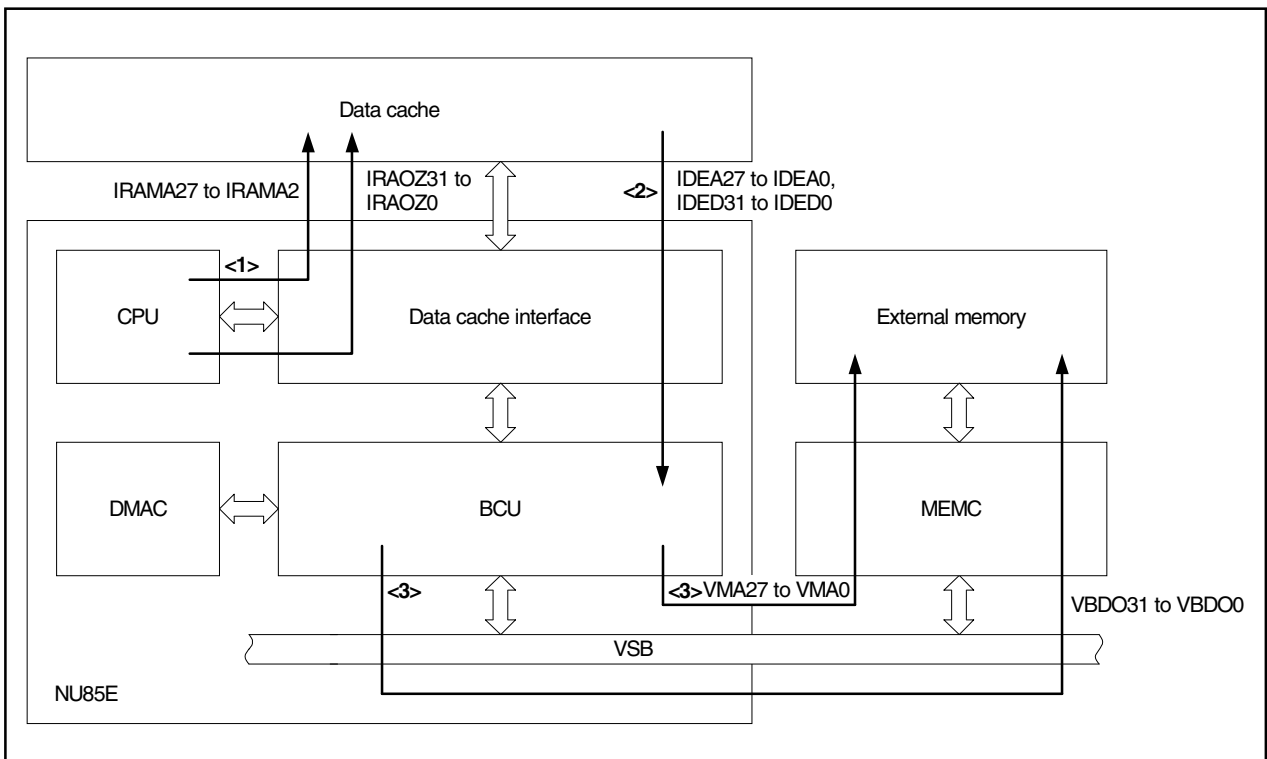
Figure 2-14. Operation on Data Cache Hit (Writeback Mode, Write Allocate Disabled, Write)



**(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, IDED31 to IDED0, respectively.
- <3> The BCU of the internal NU85E outputs the address (VMA27 to VMA0) to external memory via the VSB and writes the data to be written to external memory by passing through VBDO31 to VBDO0.

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



### 2.6.3 Writeback mode (write allocate enabled)

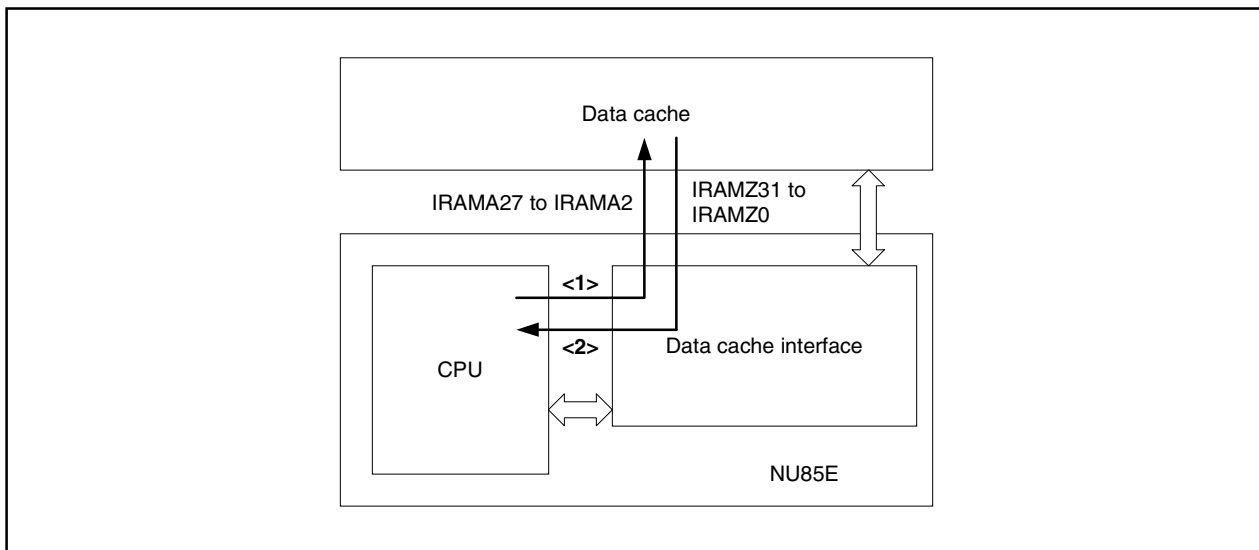
#### (1) On a read

##### (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)**

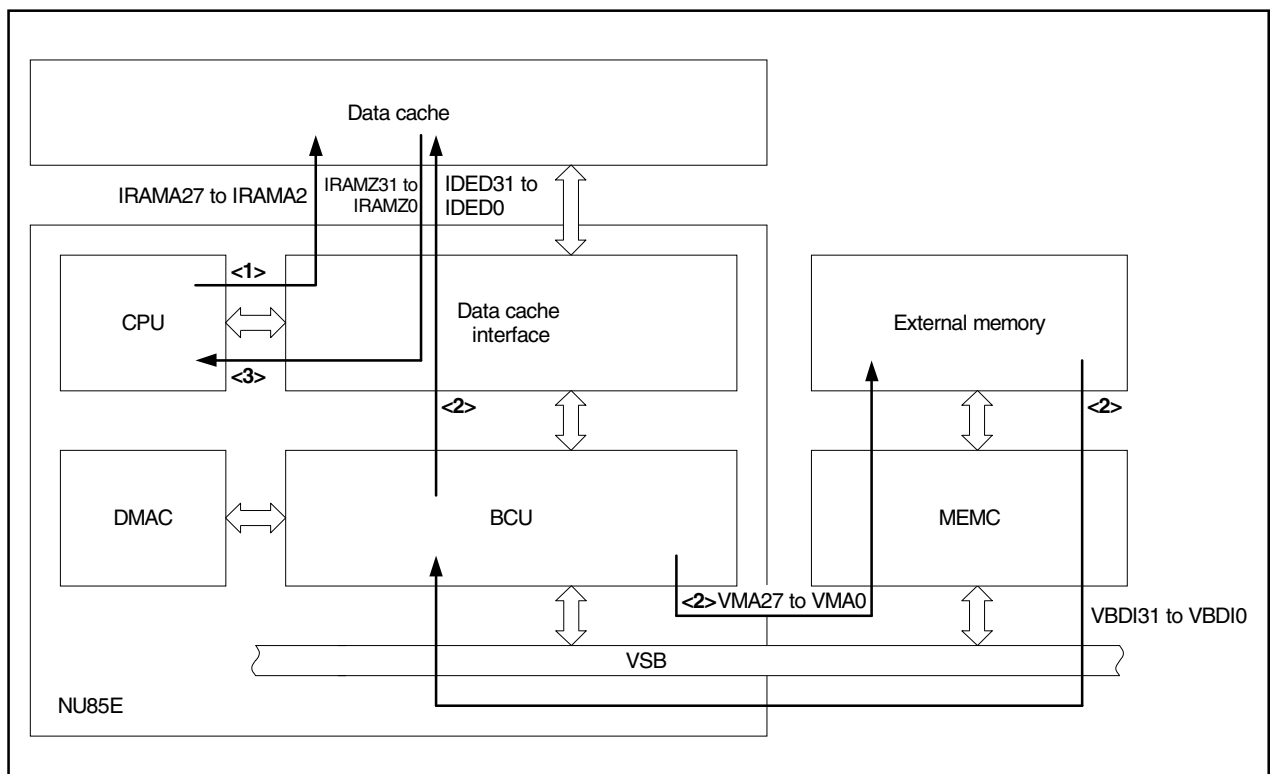


(b) Data cache miss

(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 NU85E outputs the address to be read (VMA27 to VMA0) 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.

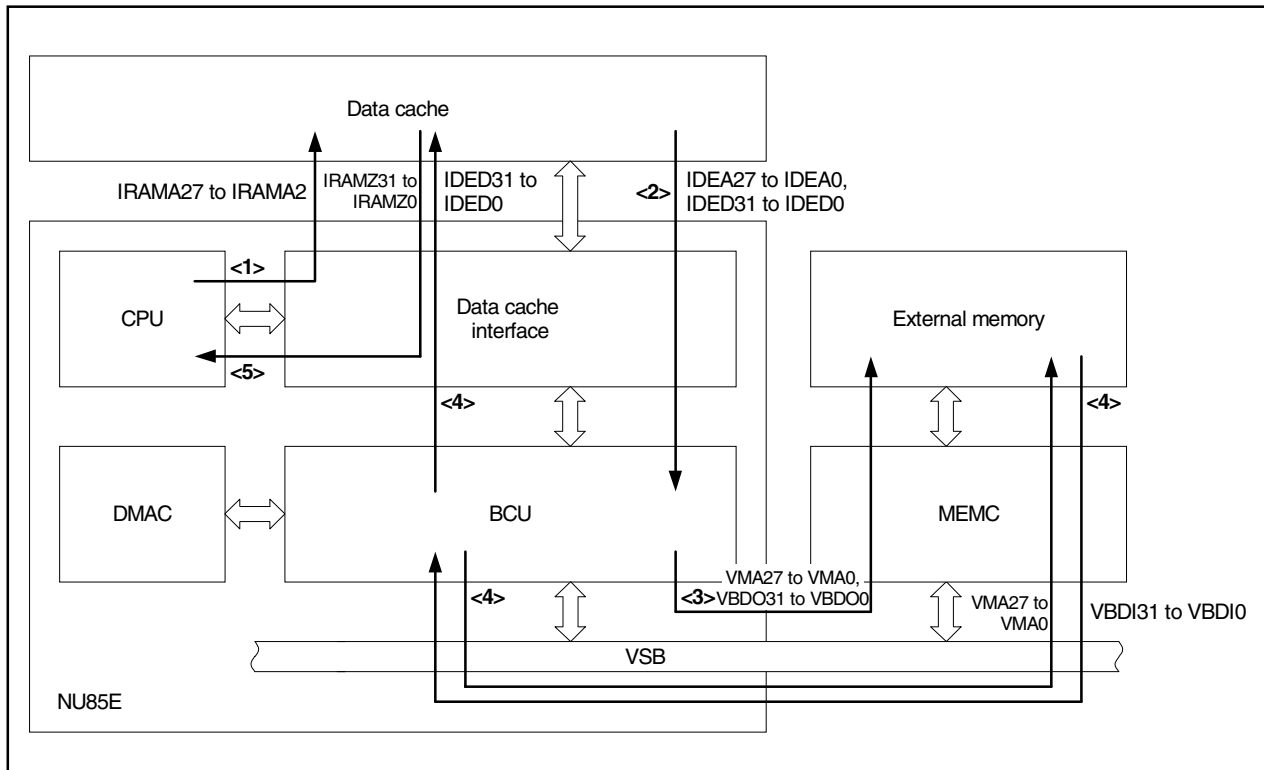
Figure 2-17. Operation on Data Cache Miss (Writeback Mode, Write Allocate Enabled, Read, Clean Data)



(ii) When data being replaced is dirty data

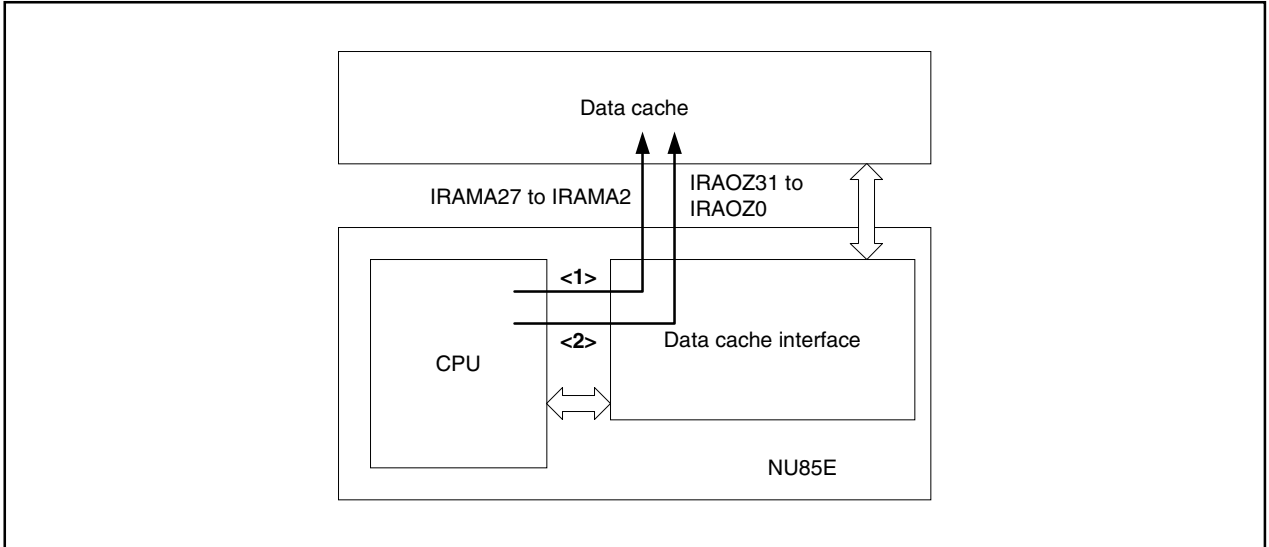
- <1> When reading data from external memory, output the address (IRAMA27 to IRAMA2) to the data cache. If a miss occurs due to the address not existing in the data cache, perform tag and data replacement.
- <2> If the data being replaced is dirty data, read the address corresponding to the line that 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 NU85E outputs the address (VMA27 to VMA0) 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 (VMA27 to VMA0) 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.

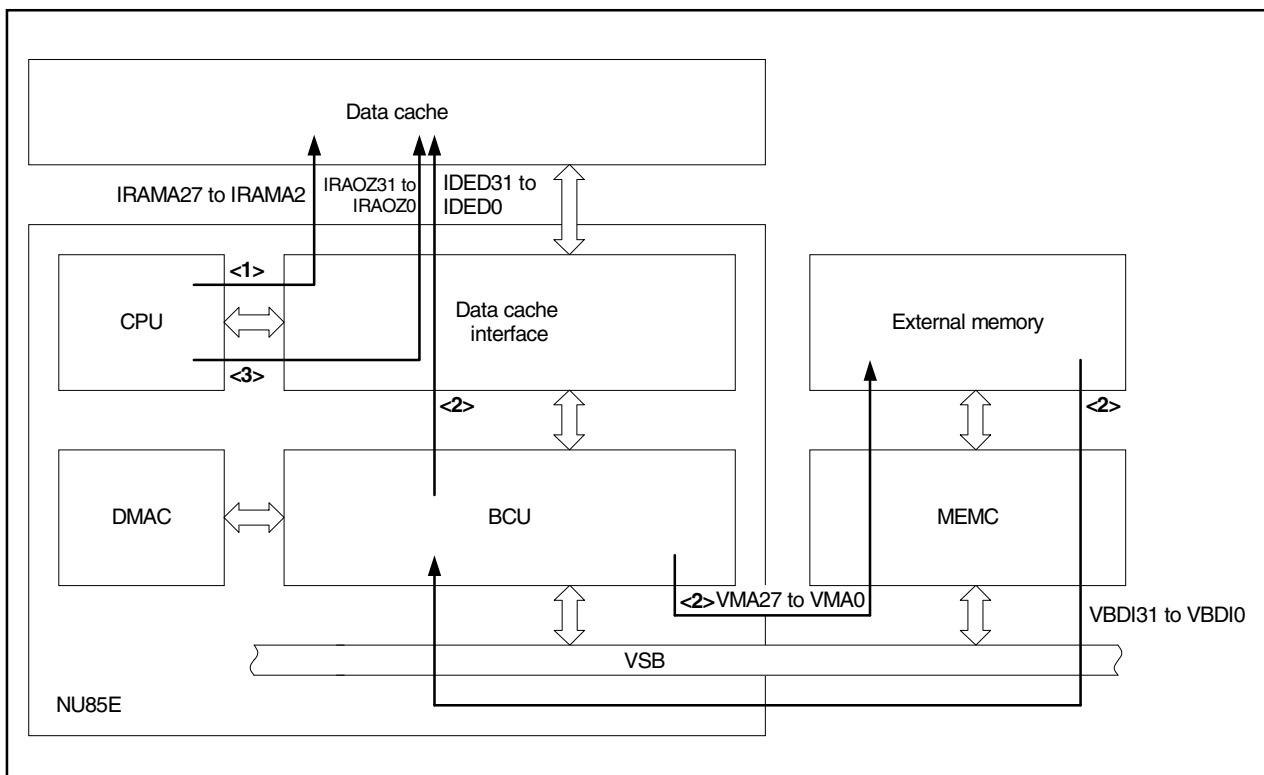
**Figure 2-19. Operation on Data Cache Hit (Writeback Mode, Write Allocate Enabled, Write)**

(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 NU85E outputs the address to be written (VMA27 to VMA0) 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)

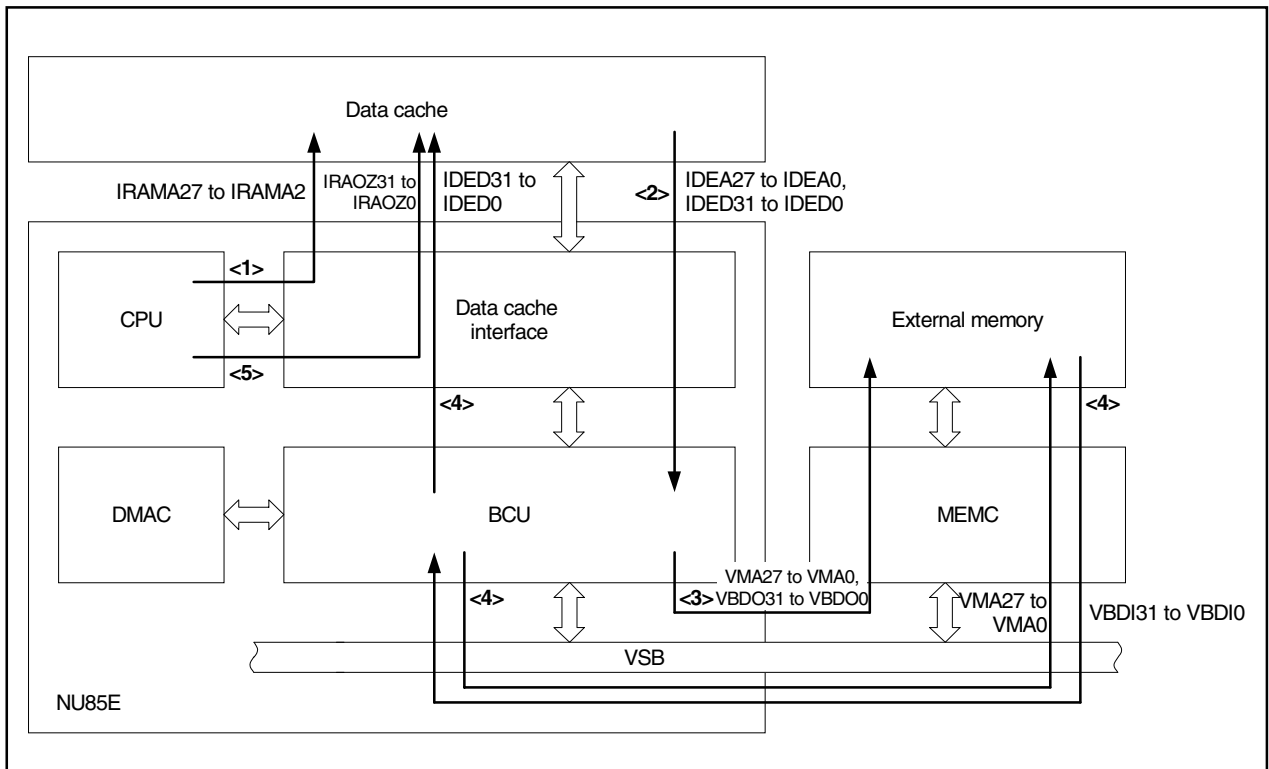




(ii) When data being replaced is dirty data

- <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 IDEDED31 to IDEDED0.
- <3> The BCU in the internal NU85E outputs the address (VMA27 to VMA0) 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 (VMA27 to VMA0) 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 NU85E signals.
  3. The circles indicate the sampling timing.
  4. For details on the VSB signals (VMxxx, VDxxx), refer to the **NU85E Hardware User's Manual (A14874E)**.
  5. XXX: Undetermined state (for output), or arbitrary level (for input)

**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)

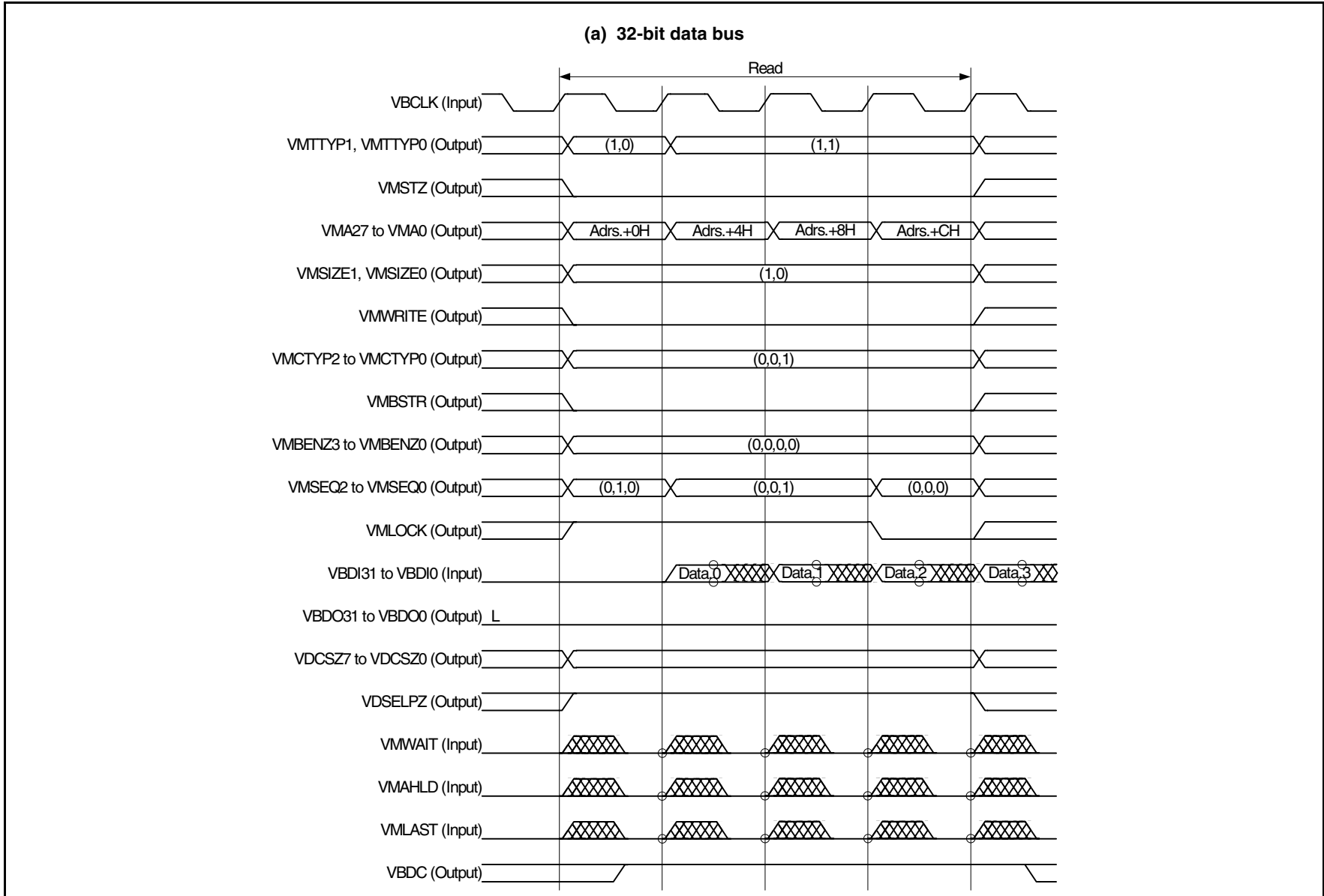


Figure 2-22. Sequential Refill Read Cycle (4R) (2/2)

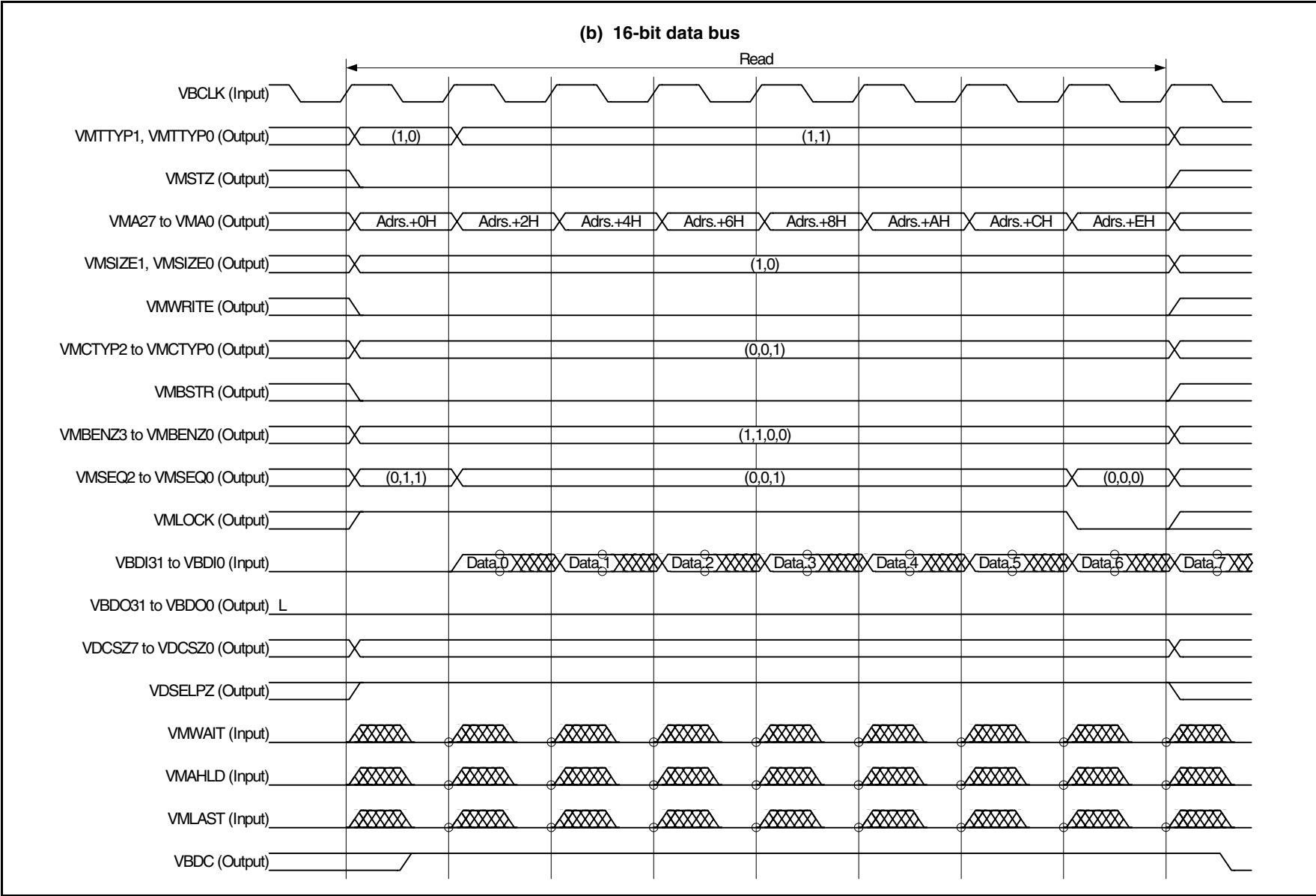


Figure 2-23. Critical First Refill Read Cycle (2R-2R) (1/2)

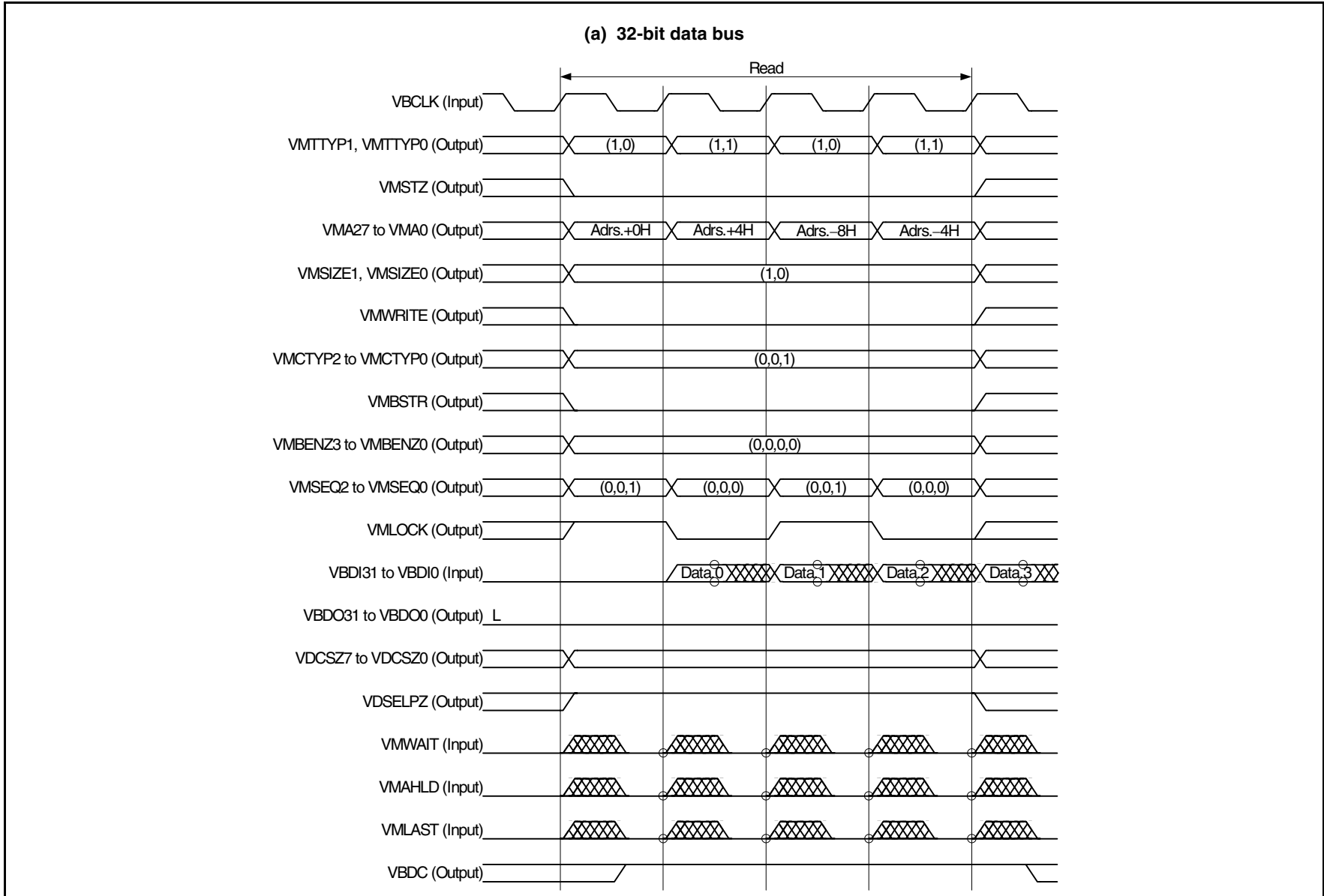


Figure 2-23. Critical First Refill Read Cycle (2R-2R) (2/2)

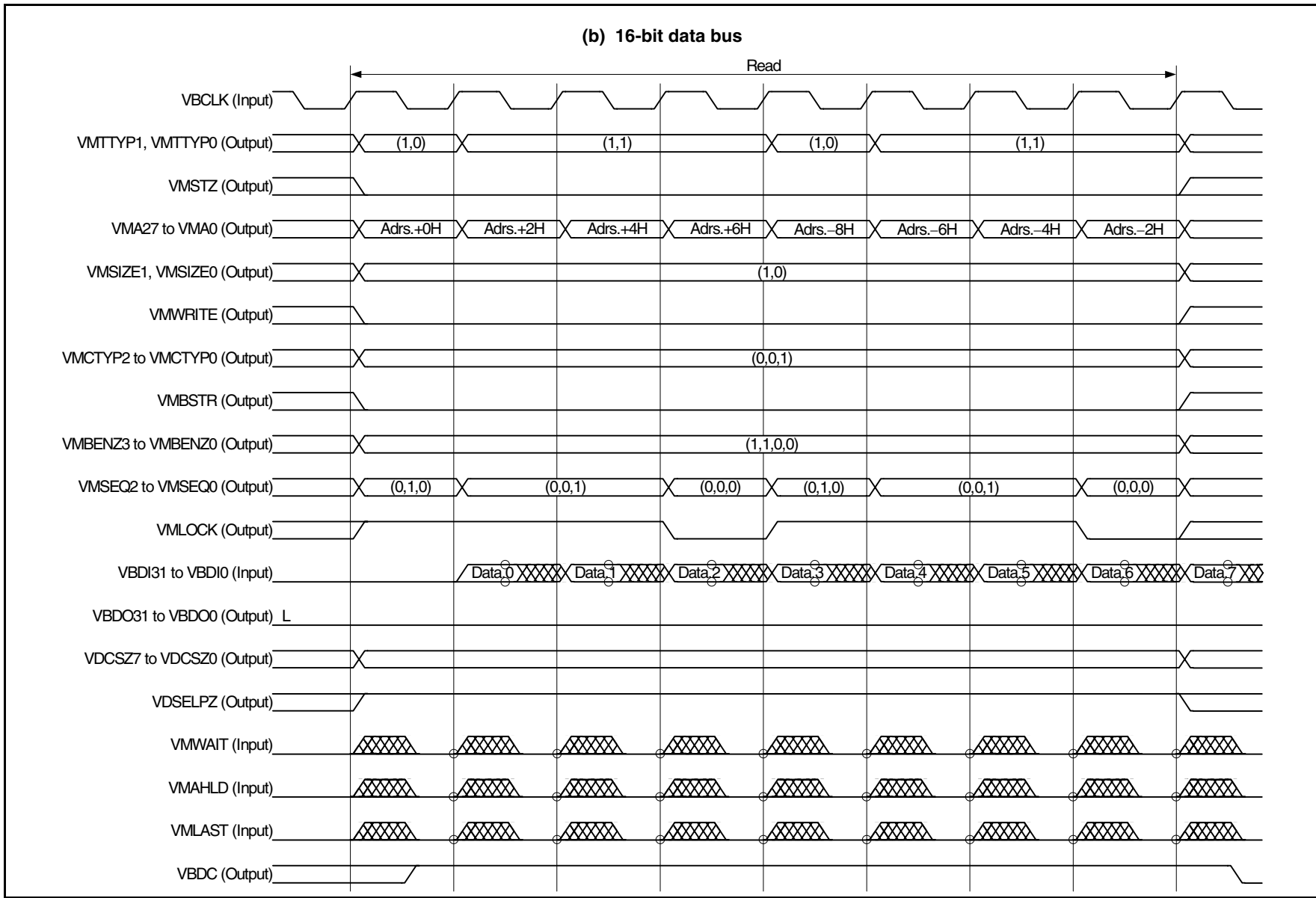


Figure 2-24. Critical First Refill Read Cycle (1R-2R-1R) (1/4)

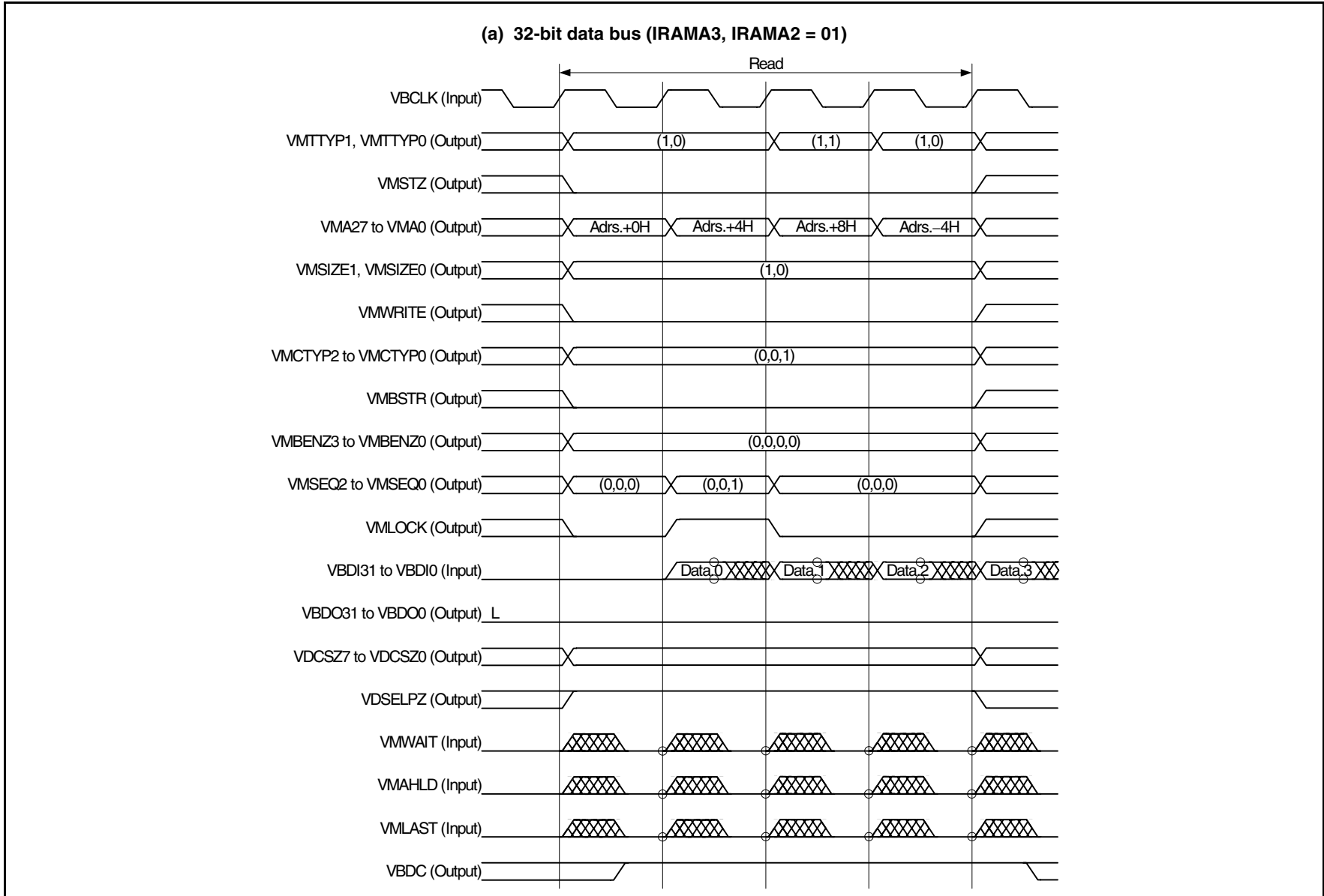




Figure 2-24. Critical First Refill Read Cycle (1R-2R-1R) (2/4)

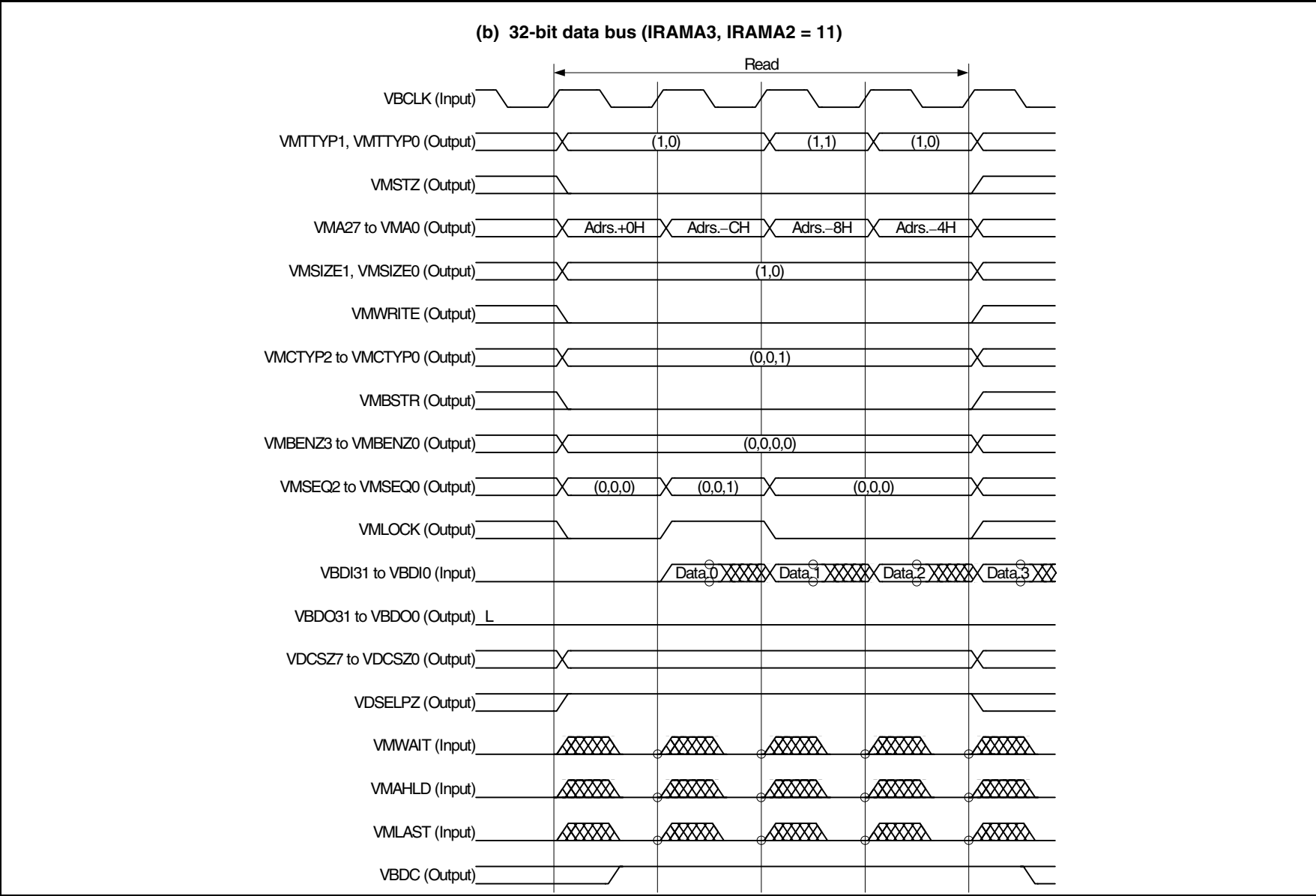


Figure 2-24. Critical First Refill Read Cycle (1R-2R-1R) (3/4)

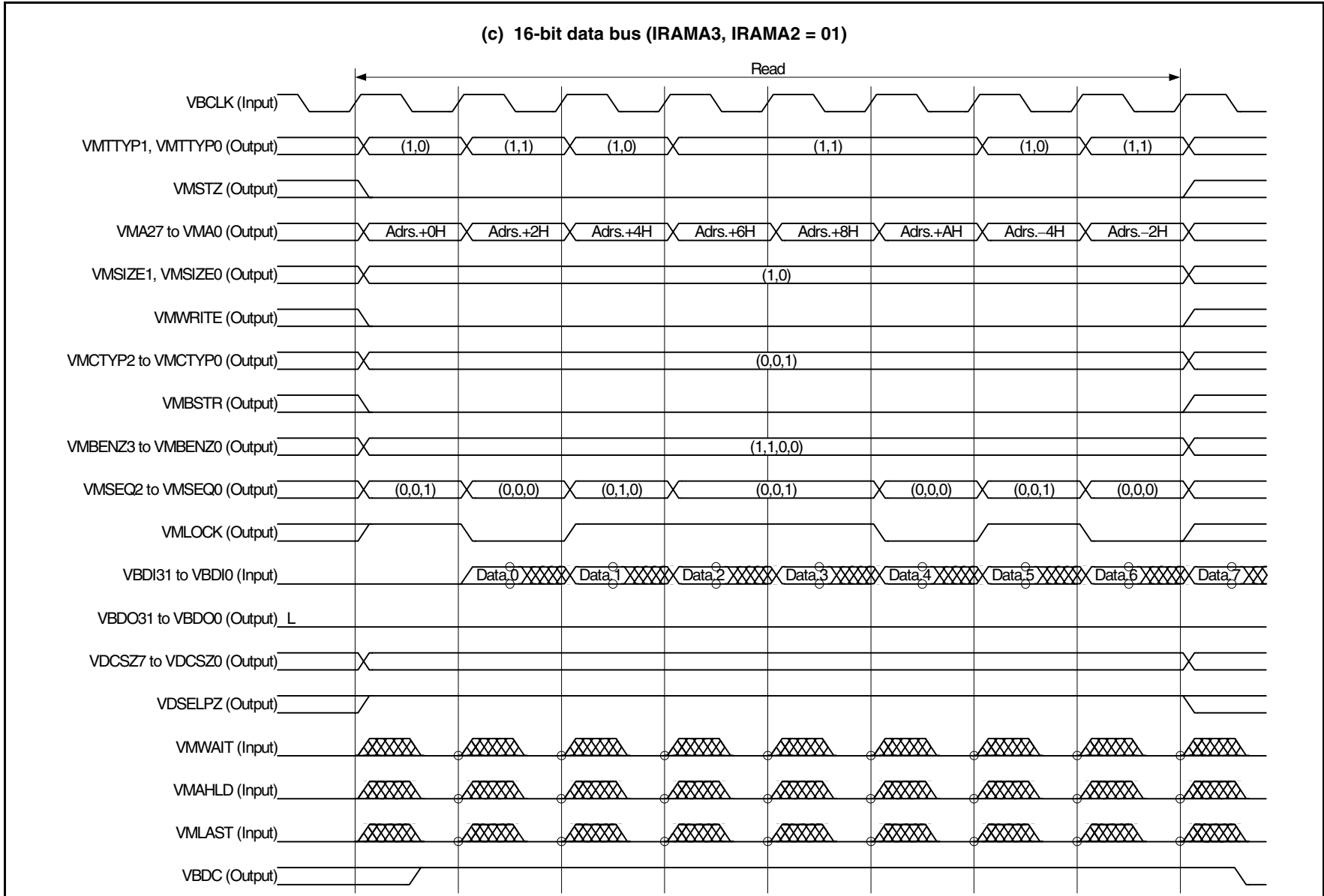


Figure 2-24. Critical First Refill Read Cycle (1R-2R-1R) (4/4)

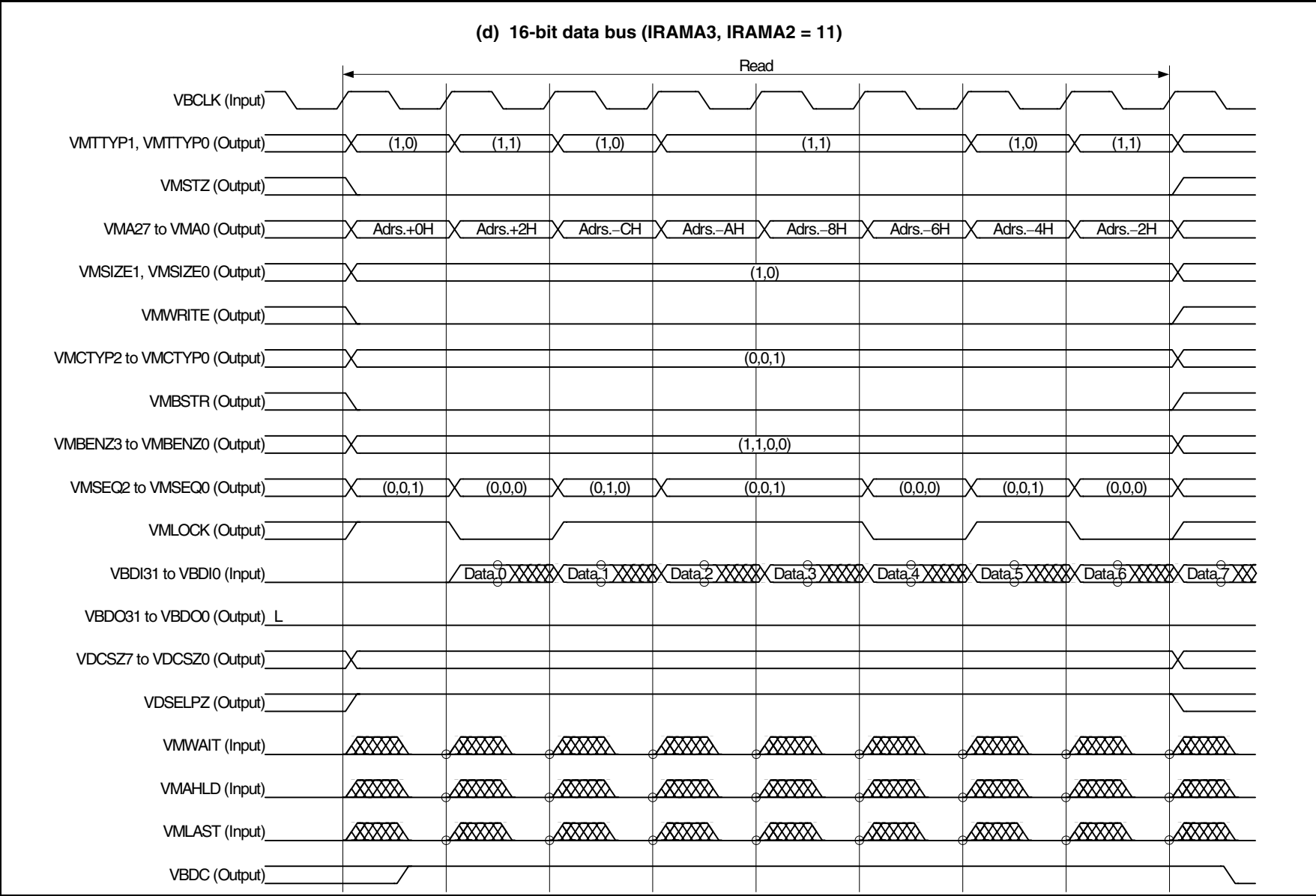


Figure 2-25. Sequential Refill Read Cycle in Writeback Mode (When Data Being Replaced Is Dirty Data) (4W + 4R) (1/2)

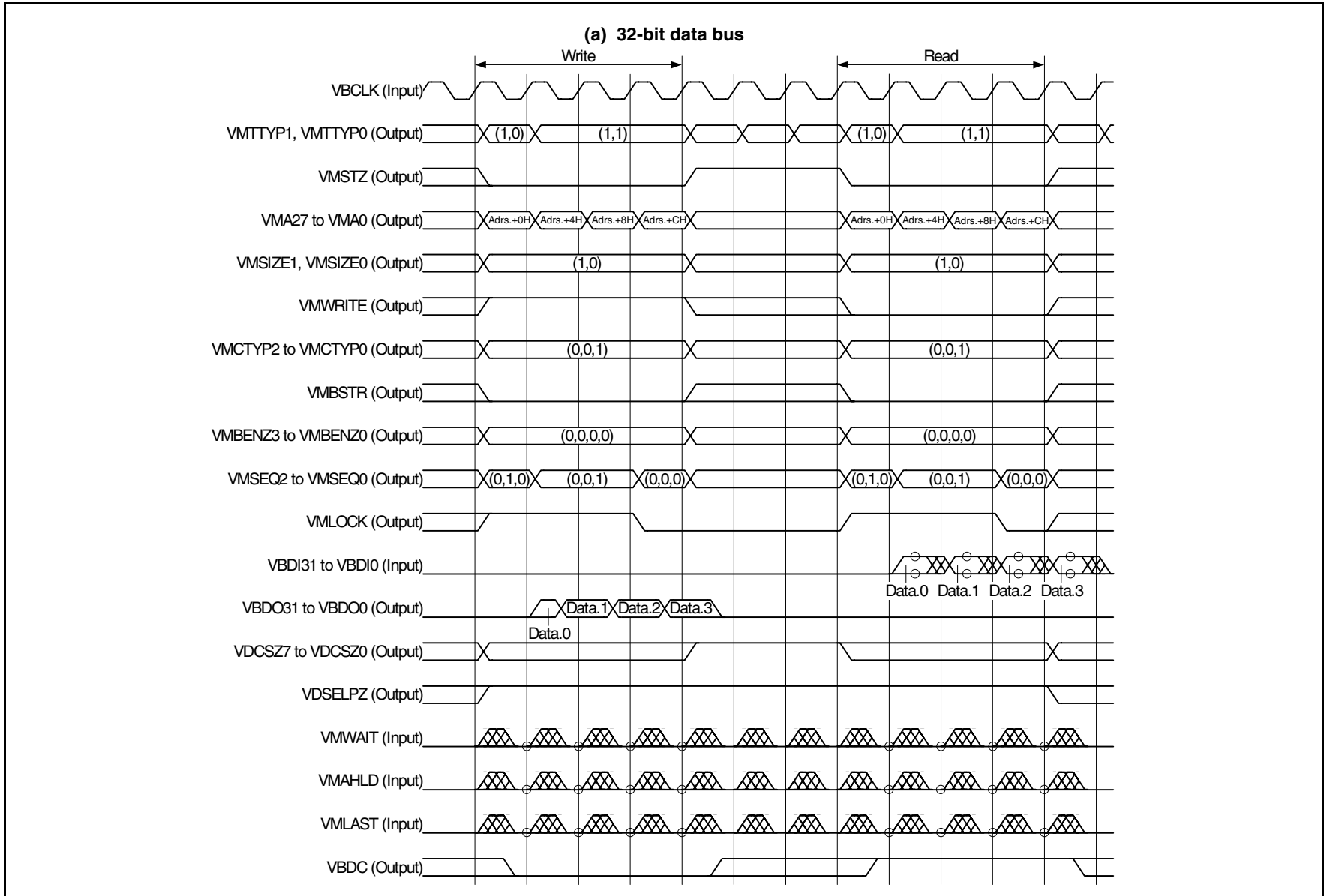


Figure 2-25. Sequential Refill Read Cycle in Writeback Mode (When Data Being Replaced Is Dirty Data) (4W + 4R) (2/2)

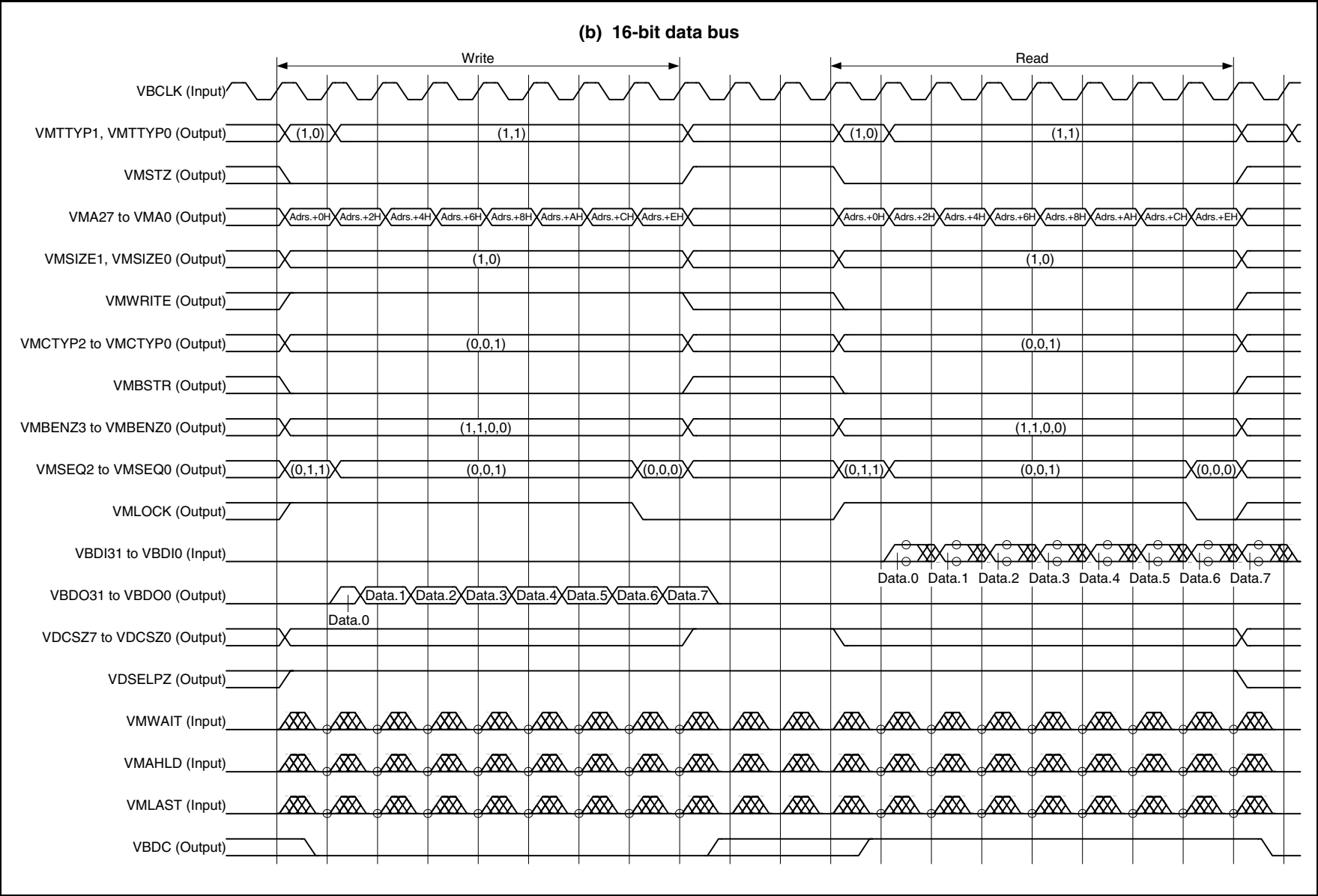


Figure 2-26. Critical First Refill Read Cycle in Writeback Mode (When Data Being Replaced Is Dirty Data) (4W + 2R-2R) (1/2)

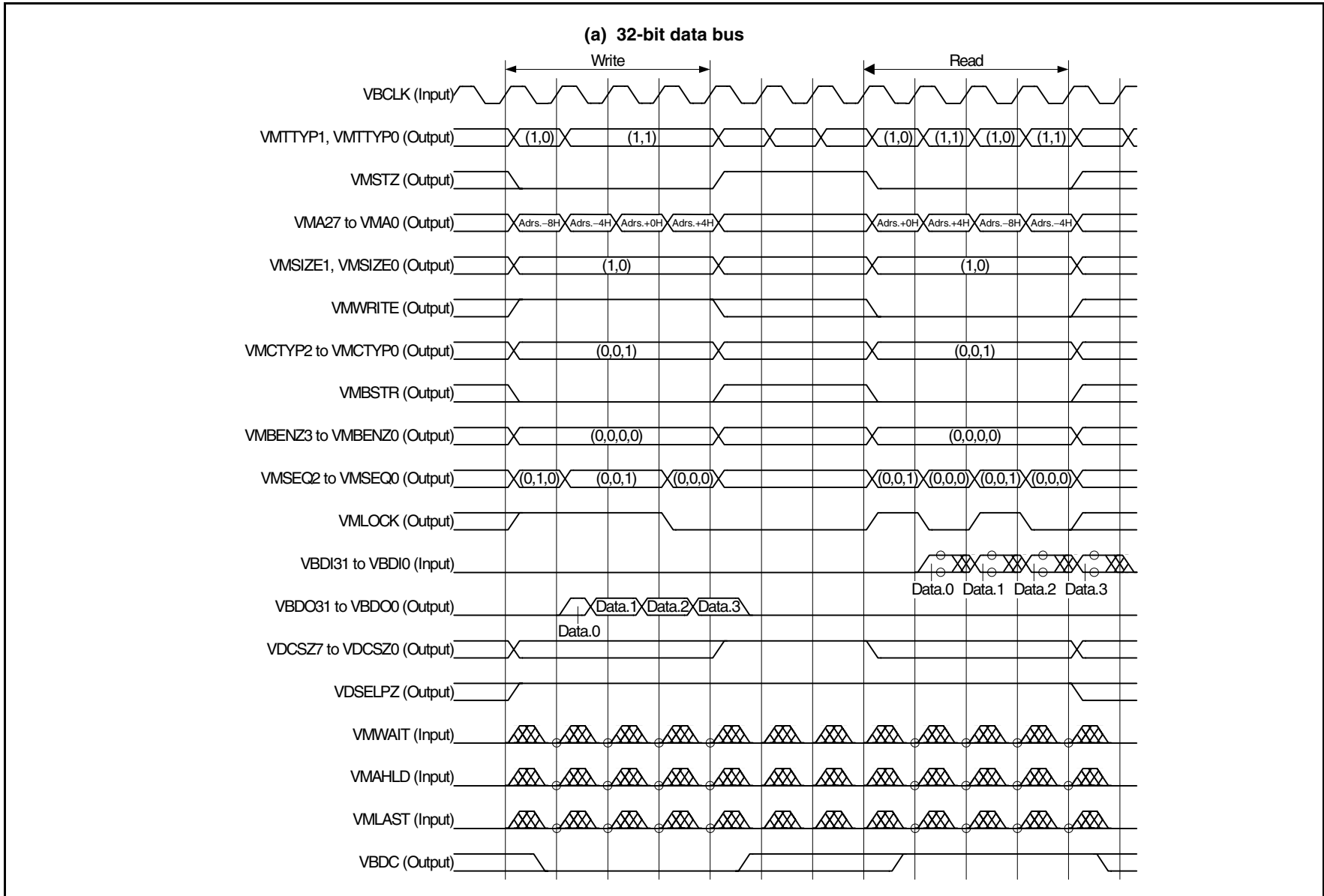


Figure 2-26. Critical First Refill Read Cycle in Writeback Mode (When Data Being Replaced Is Dirty Data) (4W + 2R-2R) (2/2)

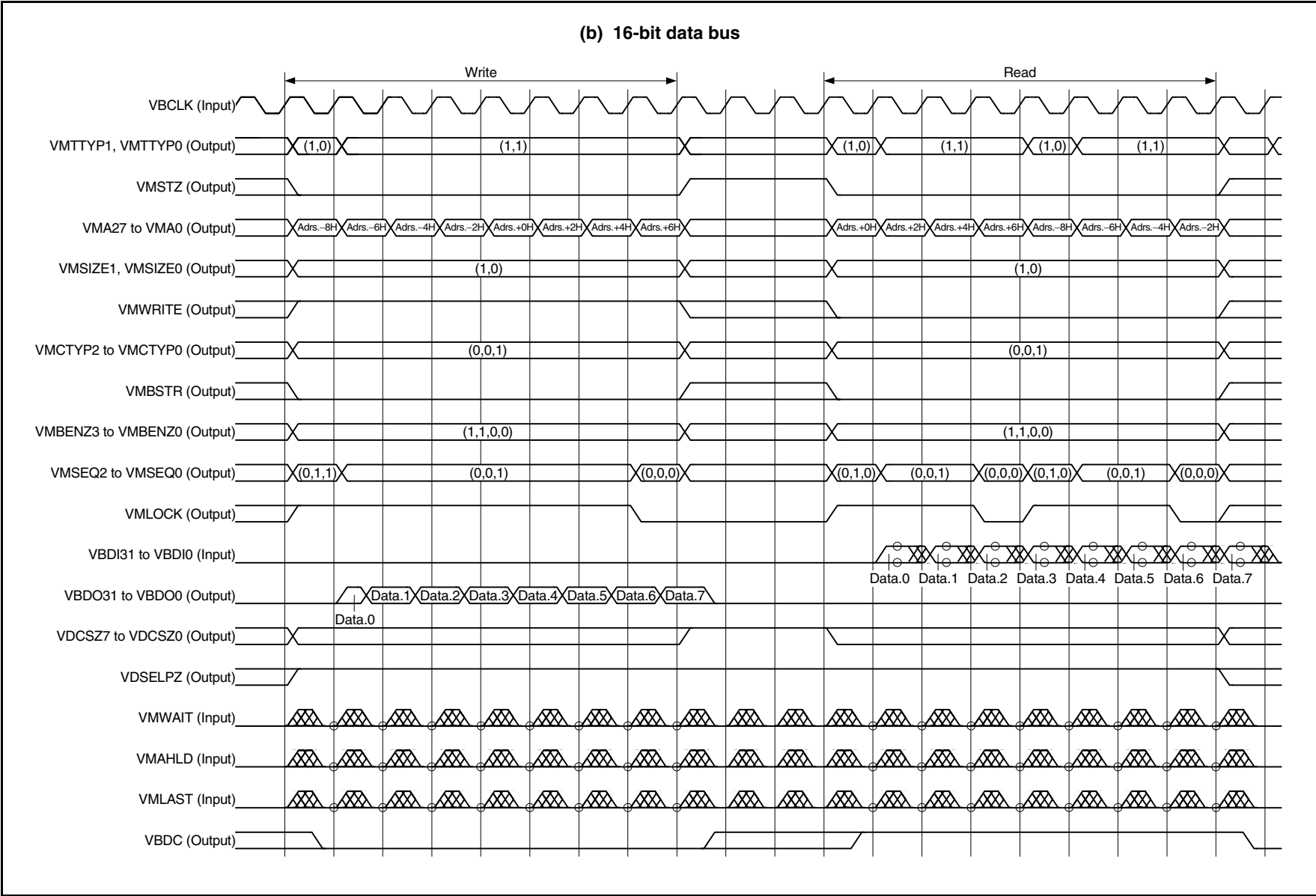


Figure 2-27. Critical First Refill Read Cycle in Writeback Mode (When Data Being Replaced Is Dirty Data) (4W + 1R-2R-1R) (1/4)

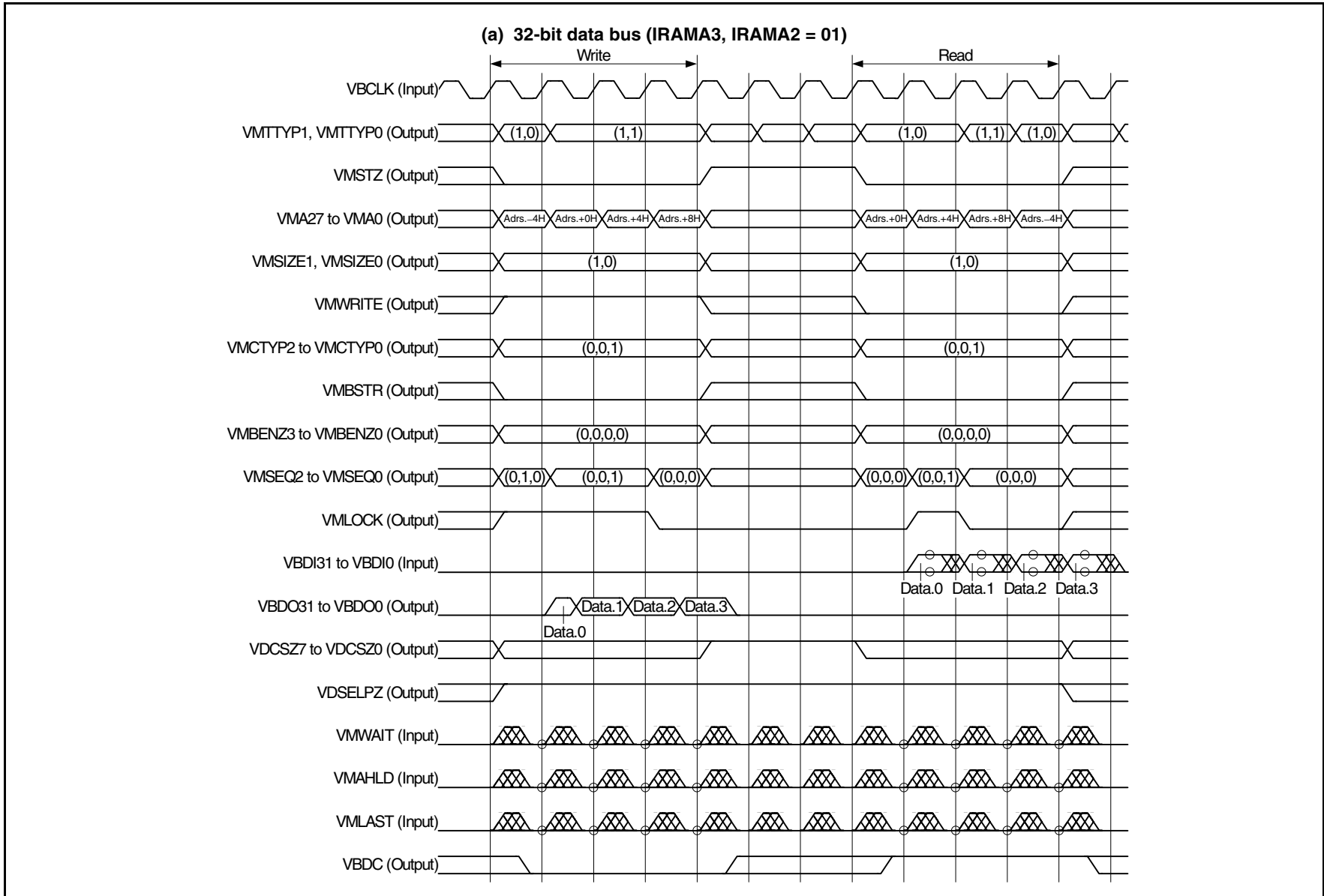




Figure 2-27. Critical First Refill Read Cycle in Writeback Mode (When Data Being Replaced Is Dirty Data) (4W + 1R-2R-1R) (2/4)

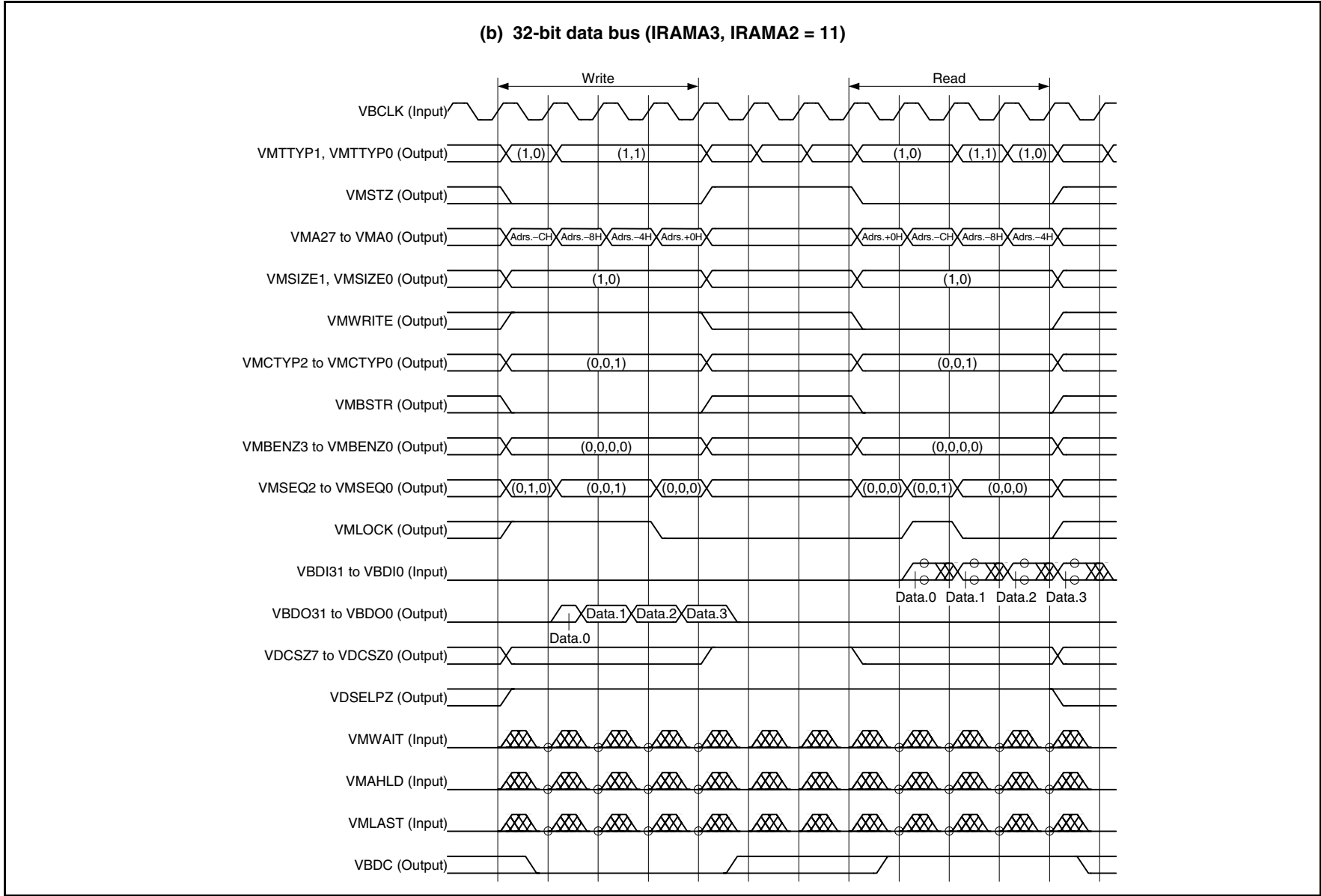


Figure 2-27. Critical First Refill Read Cycle in Writeback Mode (When Data Being Replaced Is Dirty Data) (4W + 1R-2R-1R) (3/4)

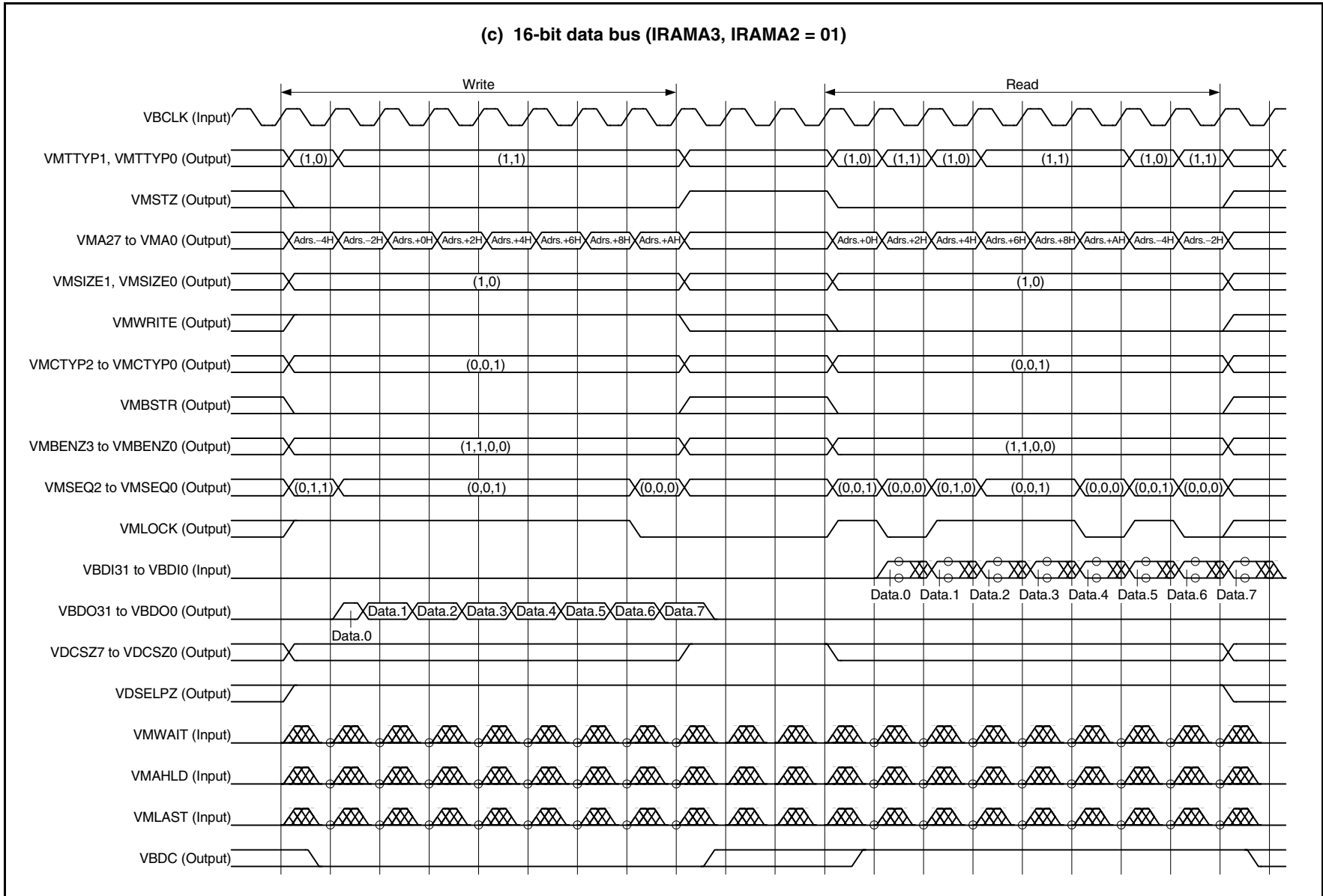
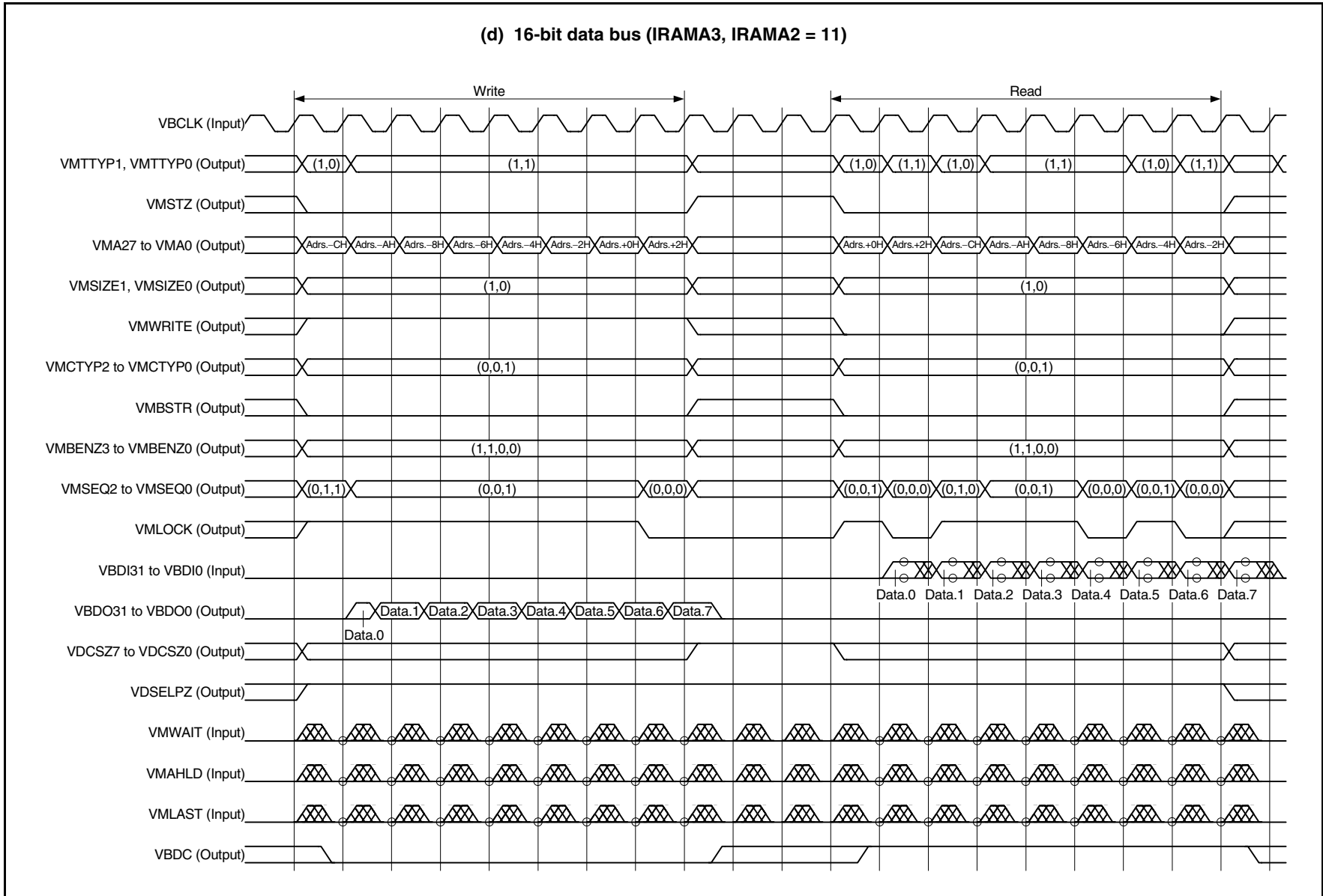


Figure 2-27. Critical First Refill Read Cycle in Writeback Mode (When Data Being Replaced Is Dirty Data) (4W + 1R-2R-1R) (4/4)



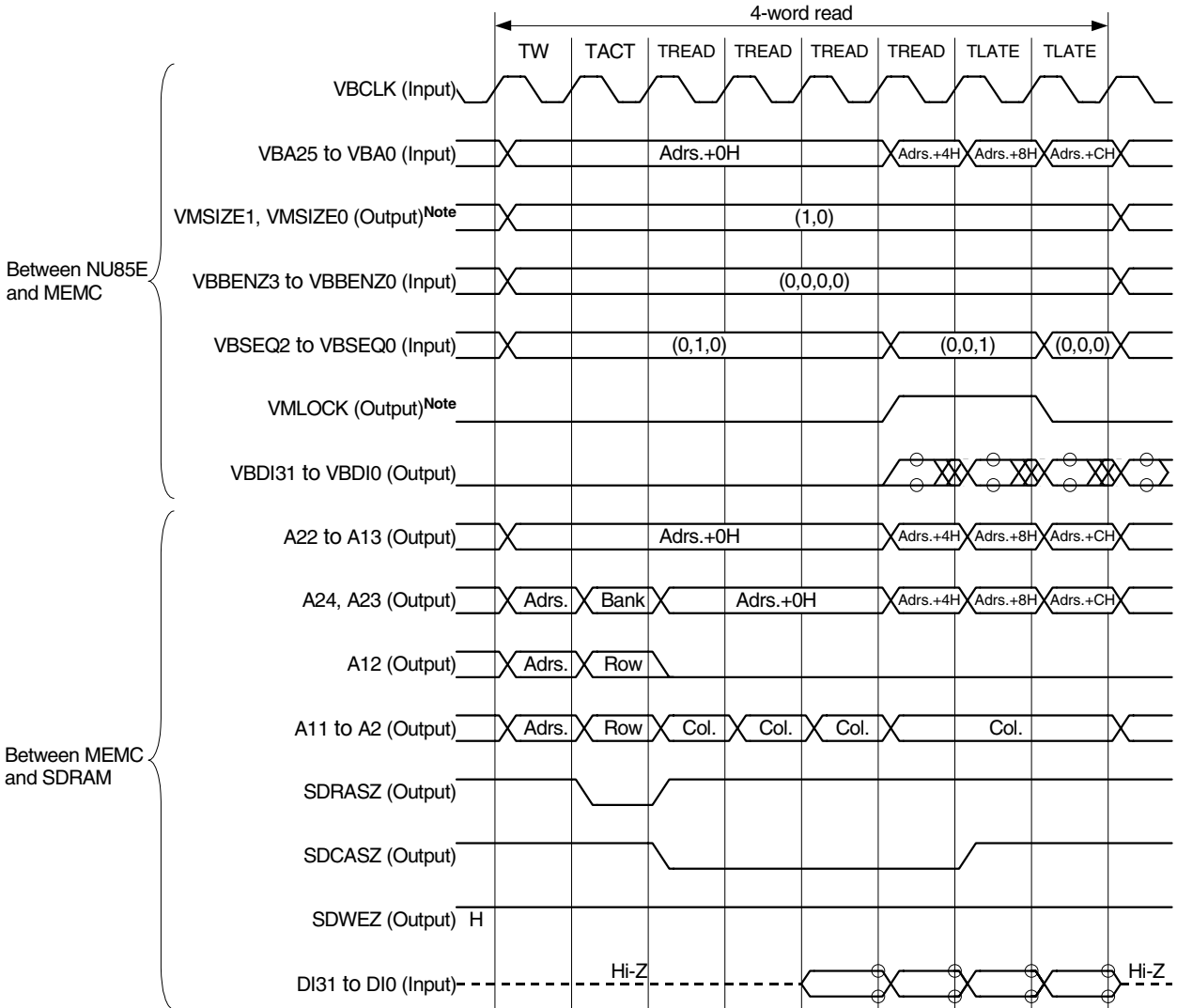
## 2.8 Timing of Refill from SDRAM to Data Cache

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 (NT85E502) = 1 (n = 7 to 0)
2. The circles indicate the sampling timing.
  3. For details on the VSB signals (VMxxx), refer to the **NU85E Hardware User's Manual (A14874E)**.
  4. The inputs and outputs as seen from the memory controller (MEMC) side are shown.
  5. The various state abbreviations have the following meanings.
    - TW: Wait state
    - TACT: Bank active command state
    - TREAD: Read command state
    - TLATE: Latency wait state
  6. XXX: Undetermined state (for output), or arbitrary level (for input)

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



**Note** NU85E signal. This signal is not connected to MEMC.

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

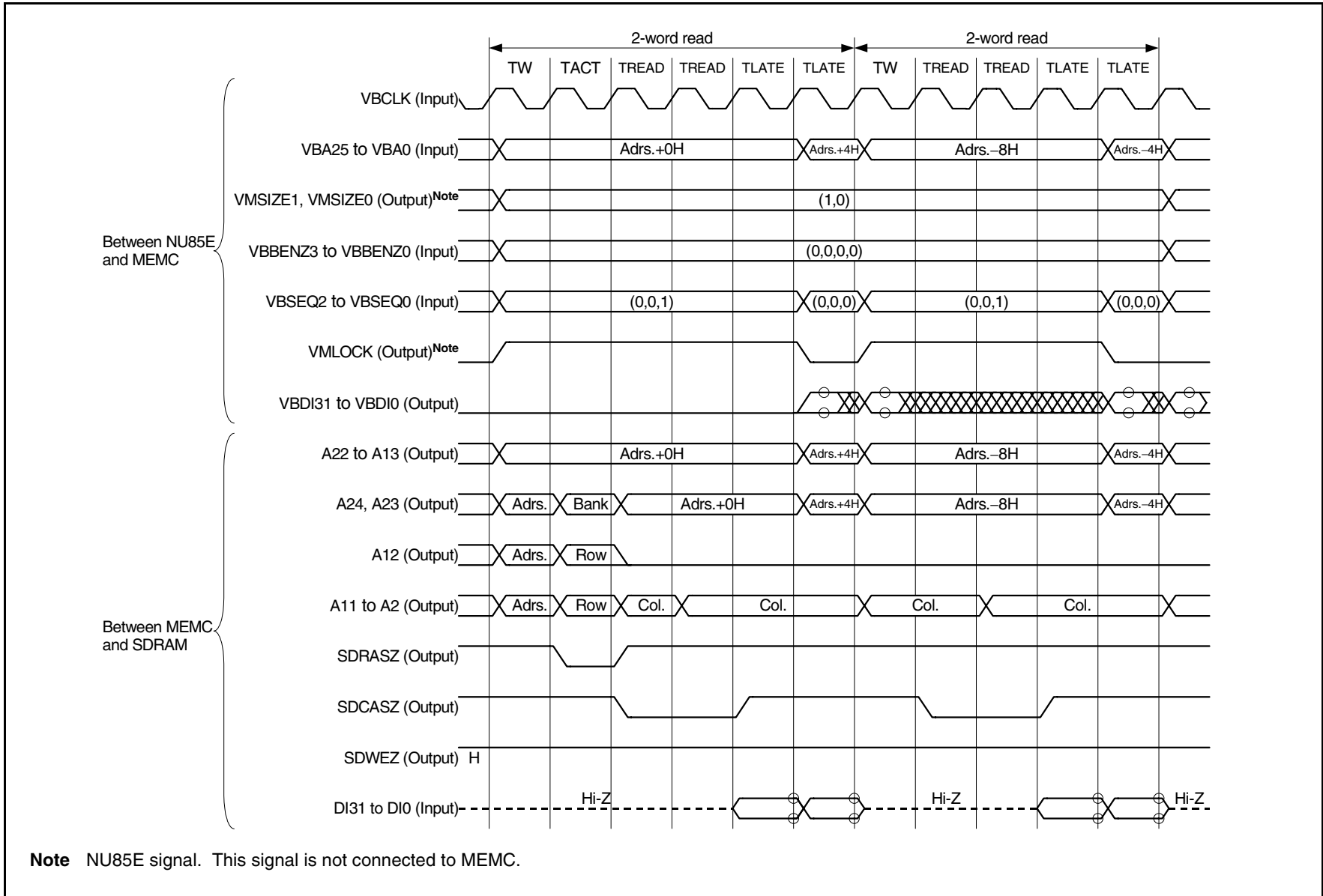
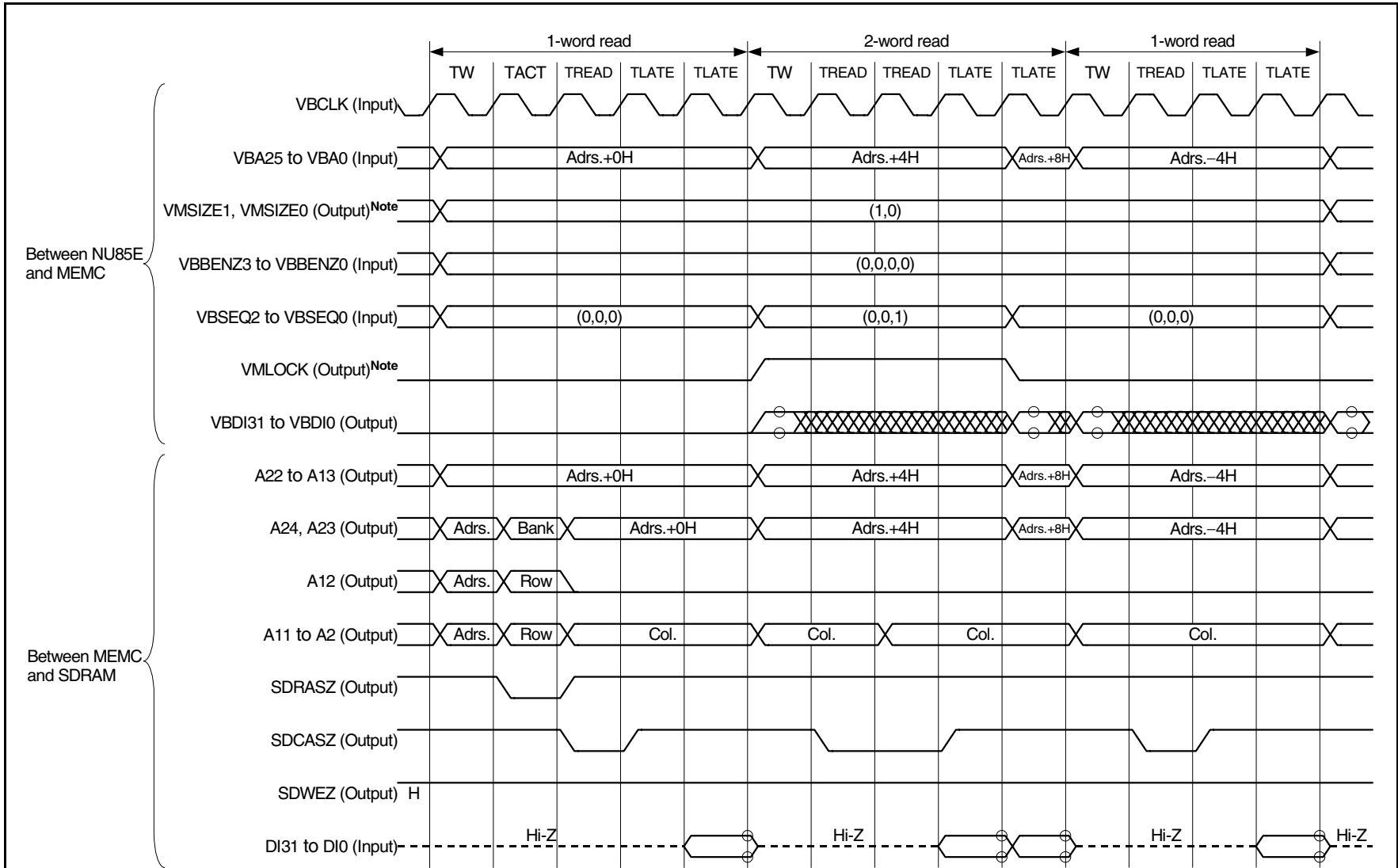


Figure 2-30. Refill Timing Example from SDRAM to Data Cache (Critical First Refill (1R-2R-1R))



**Note** NU85E 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))**

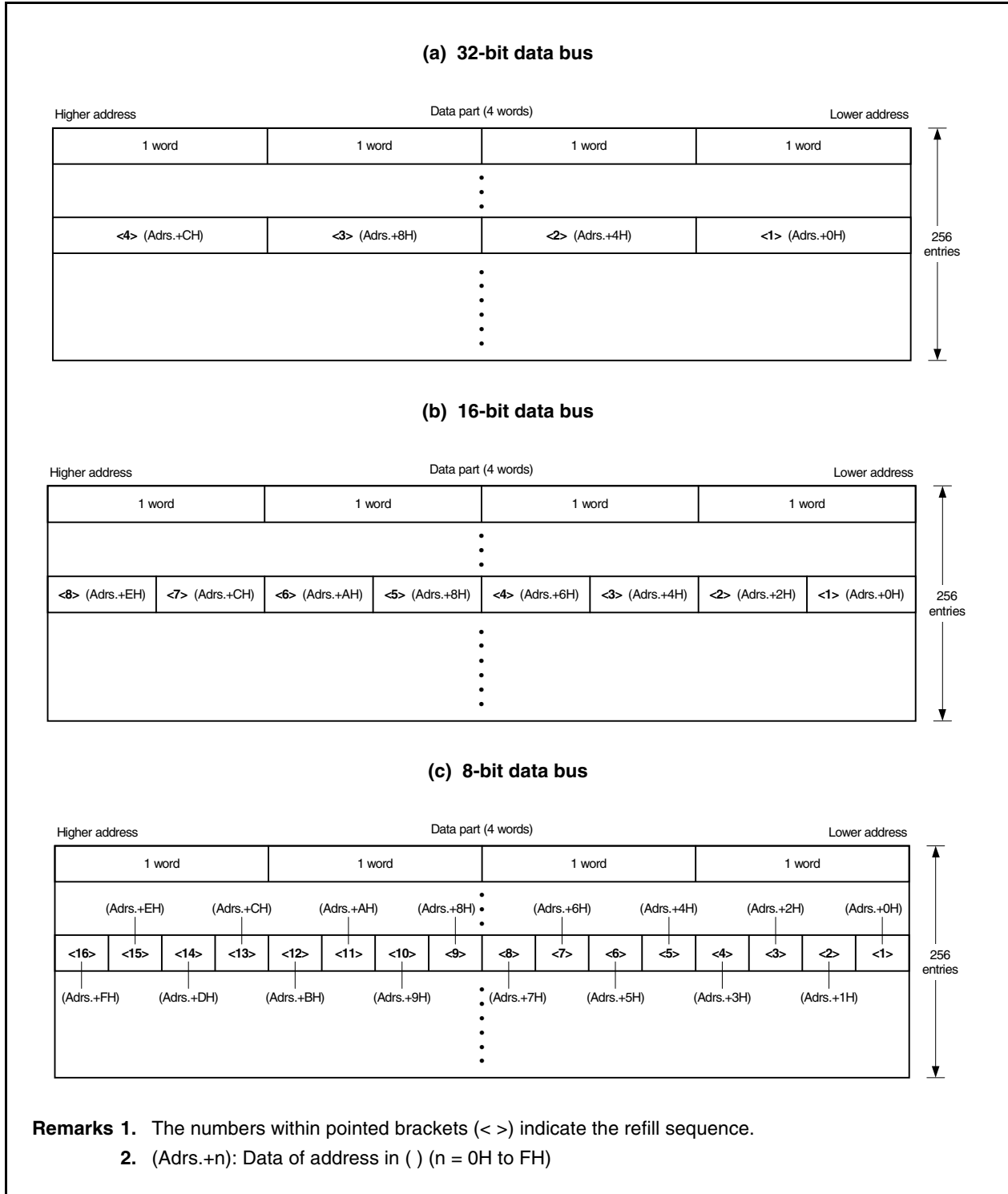




Figure 2-32. Refill Sequence to Data Cache (Critical First Refill (2R-2R))

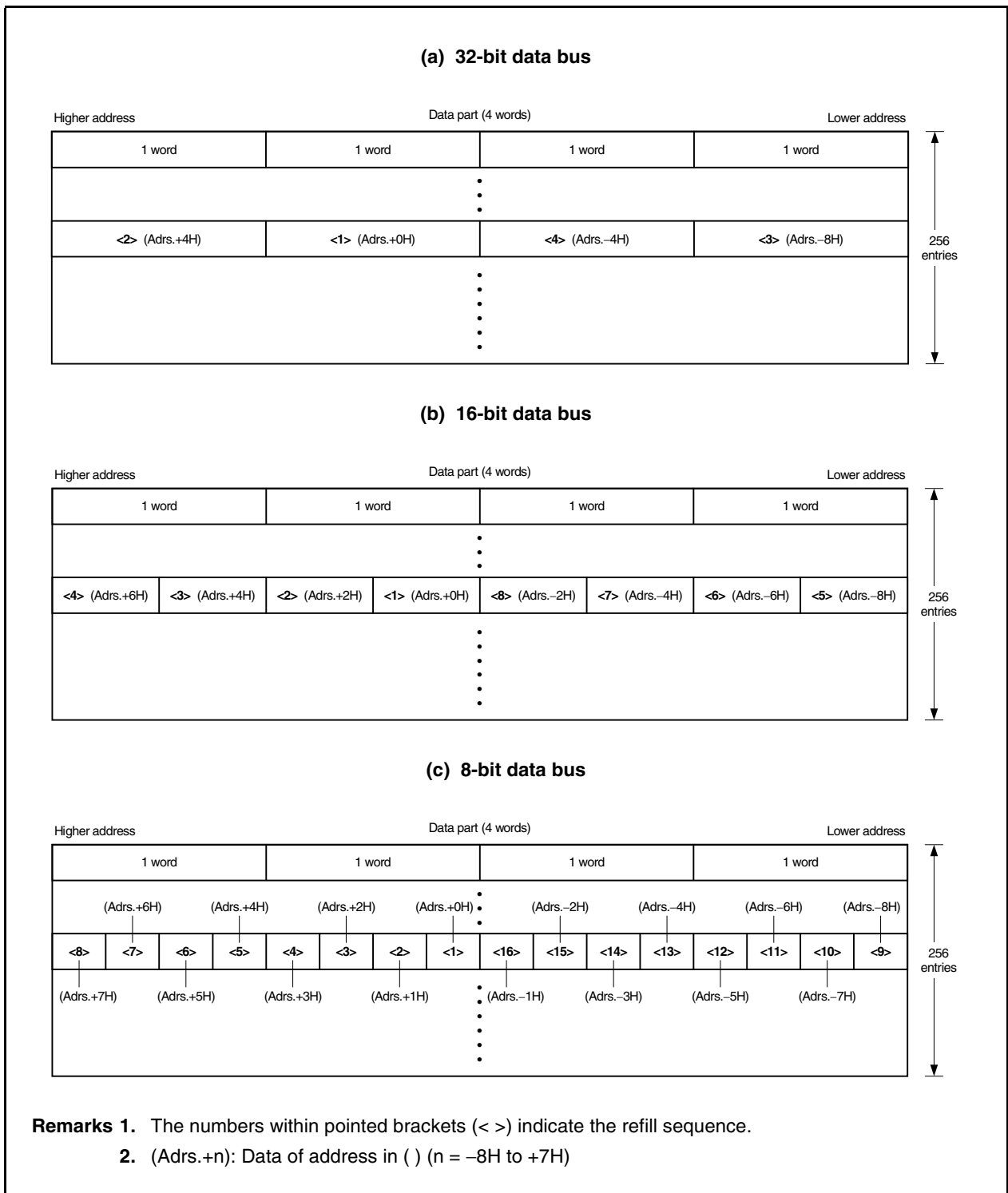
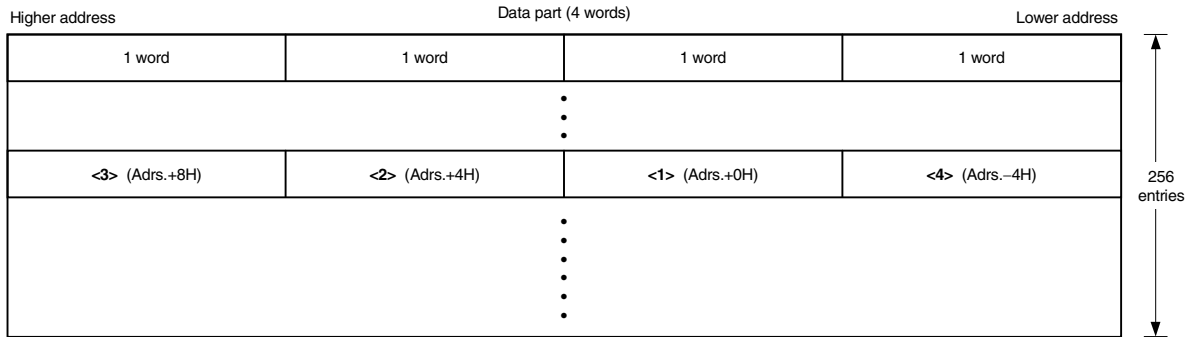


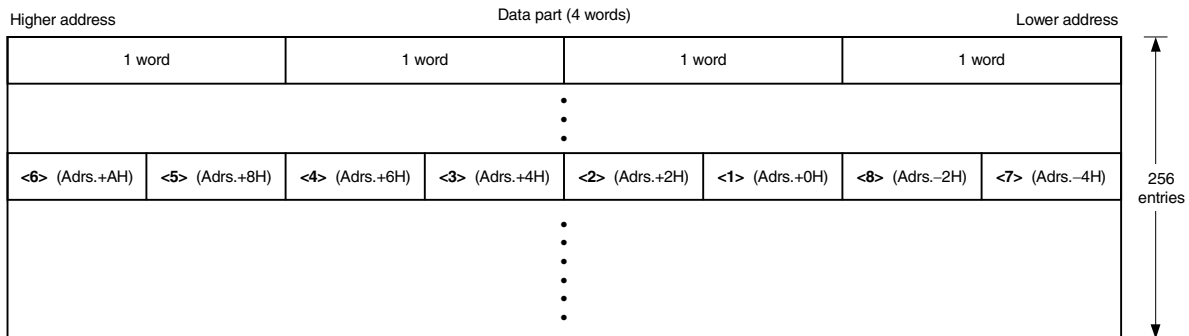
Figure 2-33. Refill Sequence to Data Cache (Critical First Refill (1R-2R-1R)) (1/2)

(1) IRAMA3, IRAMA2 = 01

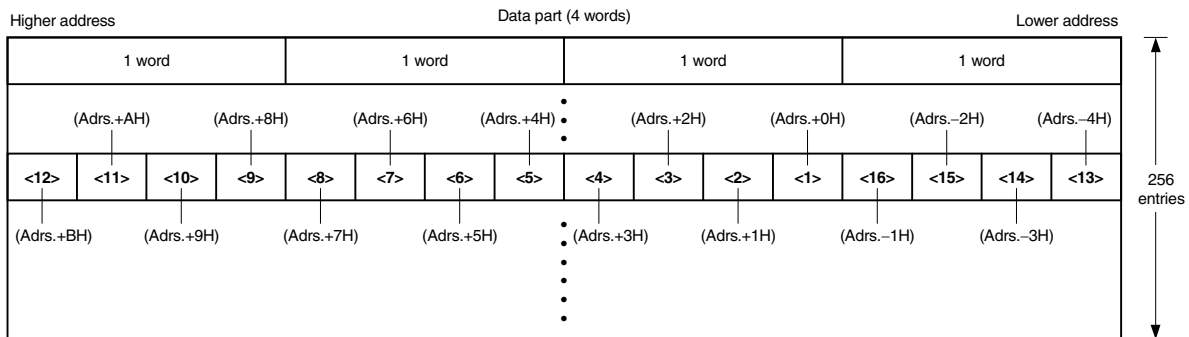
(a) 32-bit data bus



(b) 16-bit data bus

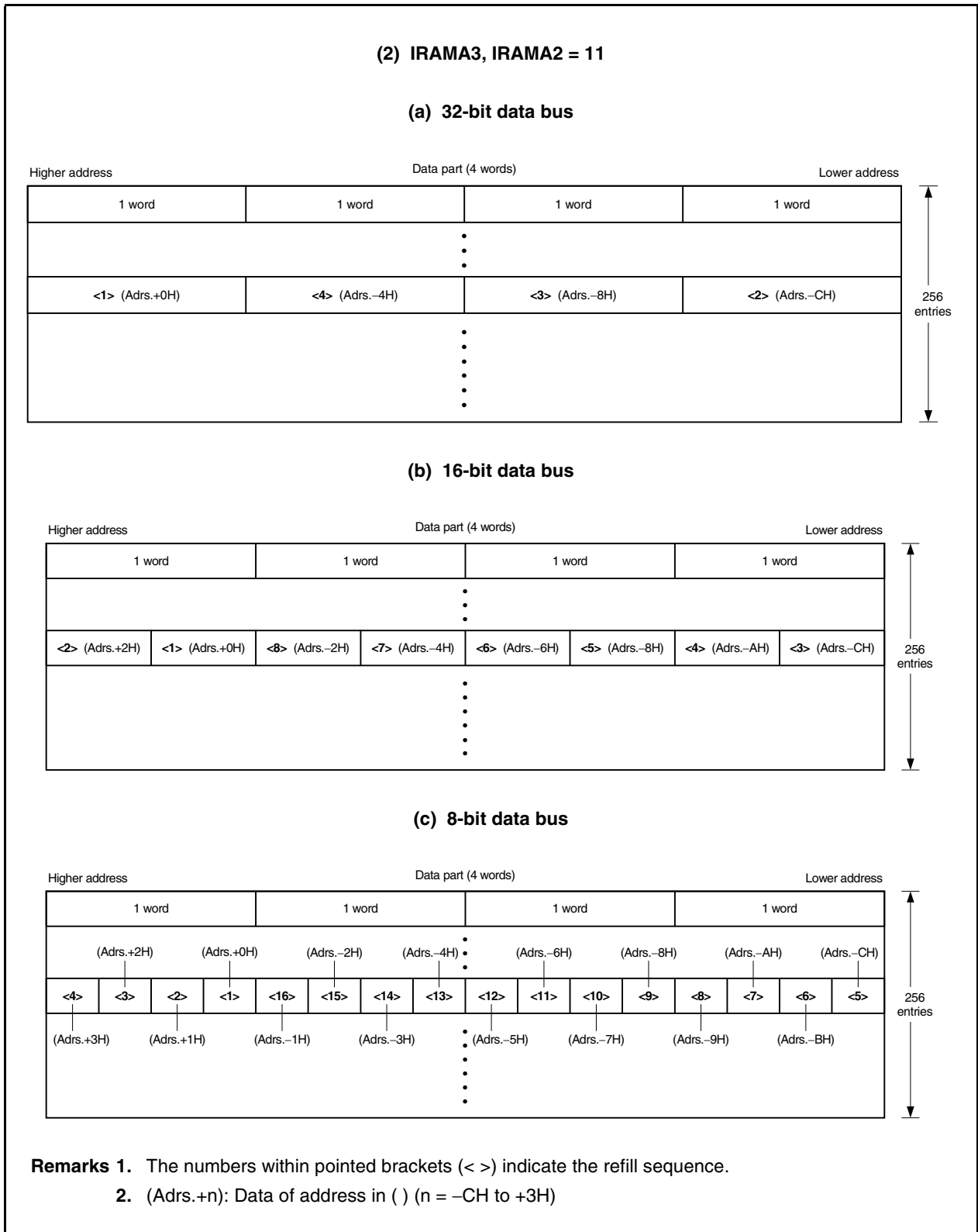


(c) 8-bit data bus



- 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)



## 2.10 Cautions

### (1) Connection to NU85E

Connect pins that have the same pin names. However, the following pins should be connected as follows (connect to pins that do not have the same pin names).

VPDR15 to VPDR0: Connect to the VPDI15 to VPDI0 pins of the NU85E.

VPDW15 to VPDW0: Connect to the VPDO15 to VPDO0 pins of the NU85E.

BUNRIIN: Connect to the BUNRIOUT pin of the NU85E.

### (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 NU85E.

Pin Name	Input Level	
	NU85E252	NU85E263
IFIASEQ	Don't care	Don't care
IFIRABE	Low level	Low level
IFIDRCT <sup>Note</sup>	–	Low level

**Note** NU85E263 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 NU85E and write cycles in writeback mode (write allocate enabled), the VMCTYP2 to VMCTYP0 signals of the NU85E 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) 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

### (6) 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.

## 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

**Hong Kong, Philippines, Oceania**

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

**Asian Nations except Philippines**

NEC Electronics Singapore Pte. Ltd.  
Fax: +65-250-3583

**Europe**

NEC Electronics (Europe) GmbH  
Market Communication Dept.  
Fax: +49-211-6503-274

**Korea**

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

**Japan**

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

**South America**

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

**Taiwan**

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

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	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Technical Accuracy	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organization	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>