To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 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 1st, 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.

RENESAS

User's Manual

μPD98409

(NEASCOT-S40C™)

ATM LIGHT SAR CONTROLLER

Document No. S12776EJ4V0UM00 (4th edition)
Date Published July 2005 NS CP (K)

[MEMO]

NOTES FOR CMOS DEVICES

① VOLTAGE APPLICATION WAVEFORM AT INPUT PIN

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (MAX) and V_{IH} (MIN) due to noise, etc., the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (MAX) and V_{IH} (MIN).

② HANDLING OF UNUSED INPUT PINS

Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to V_{DD} or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

③ PRECAUTION AGAINST ESD

A 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 when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up 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 benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

④ STATUS BEFORE INITIALIZATION

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

⑤ POWER ON/OFF SEQUENCE

In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current.

The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.

⑥ INPUT OF SIGNAL DURING POWER OFF STATE

Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements.

Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

EEPROM and NEASCOT-S40C are trademarks of NEC Electronics Corporation.
MICROWIRE is a trademark of National Semiconductor Corporation.

- **The information in this document is current as of July, 2005. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".
  The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.
  "Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots.
  "Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support).
  "Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

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

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

M8E 02. 11-1

MAJOR REVISIONS IN THIS EDITION

Page	Contents
p. 15	Addition of μPD98409GN-LMU-A in **1.2 Ordering Information**

The mark ★ shows major revised points.

[MEMO]

PREFACE

Readers

This manual is intended for user engineers who wish to understand the functions of the μPD98409 and design application systems using it.

Purpose

This manual explains the hardware functions of the μPD98409 in the following organization.

Organization

This manual consists of the following chapters.

- General
- Pin function
- Functional outline
- Interfaces
- Operations of μPD98409
- Commands
- Internal registers
- JTAG boundary scan

How to Read This Manual

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

To understand the overall functions of the μPD98409
→ Read this manual in the order of Table of Contents.

For the electrical characteristics of the μPD98409
→ Refer to the **Data Sheet** separately available.

Legend

Data significance	:	Left: high-order digit, right: low-order digit
Active low	:	××××_B (_B following pin or signal name)
Memory map address	:	Top: high-order, bottom: low-order
Note	:	Explanation of part of text marked **Note**
Caution	:	Important information
Remark	:	Supplementary information
Numeric notation	:	Binary ... ×××× or ×××B Decimal ... ×××× Hexadecimal ... ××××H

Related documents

Some of the related documents are preliminary editions but are not so specified below.

- μPD98409 Pamphlet: S12897E
- μPD98409 Data sheet: S12775E

[MEMO]

CONTENTS

CHAPTER 1 GENERAL 15

1.1 Features 15
1.2 Ordering Information 15
1.3 Example of System Configuration 16
1.4 Block Diagram 16

CHAPTER 2 PIN FUNCTION 17

2.1 Pin Configuration (Top View) 17
2.2 Pin Function 19
2.2.1 PHY device interface pin 19
2.2.2 Bus interface pins 21
2.2.3 Serial EEPROM interface pins 23
2.2.4 JTAG boundary scan pins 23
2.2.5 Other pins 24
2.2.6 Power and ground pins 24
2.2.7 Pin status during and after reset 25

CHAPTER 3 FUNCTIONAL OUTLINE OF μPD98409 27

3.1 Functional Outline 27
3.2 AAL-5 SAR Sublayer Function 29
3.3 ATM Layer Functions 30
3.3.1 Creation of cells 30
3.3.2 Setting of cell transmission rate 32
3.3.3 Support of non-AAL-5 traffic 33

CHAPTER 4 INTERFACES 35

4.1 Bus Interface 35
4.1.1 Features of PCI bus interface 35
4.1.2 Configuration register 36
4.1.3 PCI bus transaction 39
4.2 Serial EEPROM Interface 52
4.2.1 Serial EEPROM interface 52
4.2.2 Accessing EEPROM 53
4.2.3 EEPROM format 54
4.3 PHY Device Interface 55
4.3.1 UTOPIA interface 55
4.3.2 PHY device control interface 60

CHAPTER 5 OPERATIONS OF μPD98409 63

5.1 Initialization 63
5.2 Setting of Control Memory 65

5.3 Setting of Mailbox 68
5.3.1 Setting of mailbox 68
5.3.2 Operation of mailbox 69
5.4 Transmission Function 70
5.4.1 Transmission processing flow 70
5.4.2 Structure of transmit data 70
5.4.3 Transmit channel (transmit VC) 78
5.4.4 Traffic control 81
5.4.5 Transmit operation 95
5.4.6 Support of non-AAL-5 traffic 97
5.4.7 Issuance of transmit indication 98
5.5 Reception Function 99
5.5.1 Reception processing flow 99
5.5.2 Structure of receive data 101
5.5.3 Receive channel (receive VC) 106
5.5.4 Setting of receive look-up table 111
5.5.5 Receive operation 113
5.5.6 Issuance of receive indication 126
5.5.7 Reception of non-AAL-5 traffic 126
5.6 Transmit/Receive Indication 128
5.7 Interrupt Function 131
5.8 Loopback Function 131
5.9 Global Shutdown Function 131

CHAPTER 6 COMMANDS 133

6.1 Command Register (CMR) and Command Extension Register (CER) 133
6.2 Busy Flag 134
6.3 Commands Returning Command Indication 135
6.4 Commands Using Command Extension Register 135
6.5 Commands and Command Indication 136
6.6 Command Details 137

CHAPTER 7 INTERNAL REGISTERS 147

7.1 Register List 147
7.2 Direct Address Registers 149
7.3 Indirect Address Registers 159

CHAPTER 8 JTAG BOUNDARY SCAN 163

8.1 Features 163
8.2 Internal Configuration of Boundary Scan Circuit 164
8.2.1 Instruction register 164
8.2.2 TAP (Test Access Port) controller 164
8.2.3 Bypass register 164
8.2.4 Boundary scan register 164
8.3 Pin Function 165

8.3.1 JCK (JTAG Clock) pin 165
8.3.2 JMS (JTAG Mode Select) pin 165
8.3.3 JDI (JTAG Data Input) pin 165
8.3.4 JDO (JTAG Data Output) pin 165
8.3.5 JRST_B (JTAG Reset) pin 165
8.4 Operation Description 166
8.4.1 TAP controller 166
8.4.2 TAP controller state 166
8.5 TAP Controller Operation 171
8.6 Initializing TAP Controller 174
8.7 Instruction Register 174
8.7.1 BYPASS instruction 175
8.7.2 EXTEST instruction 175
8.7.3 SAMPLE/PRELOAD instruction 175
8.7.4 Boundary scan data bit definition 175

LIST OF FIGURES (1/2)

Figure No.	Title	Page
3-1	Functions of μPD98409	27
3-2	System Outline of μPD98409	28
3-3	PDU Format of ALL-5	29
3-4	Cell Structure of User Network Interface (UNI)	30
4-1	Bus Interface Signals	35
4-2	Layout of Configuration Register	36
4-3	Slave Transaction Timing	39
4-4	Master Transaction Timing	41
4-5	Transmit Cell Data Byte Alignment	49
4-6	AD[1:0] and PCBE_B[3:0] Pins during Byte Alignment Write Transfer	50
4-7	Example of Storing Cell Data by Byte Alignment	51
4-8	EEPROM Interface Timing	52
4-9	PHY Layer Interface for Data and Control	55
4-10	Octet-Level Handshake Transmission Timing	57
4-11	Cell-Level Handshake Transmission Timing	58
4-12	Octet-Level Handshake Reception Timing	59
4-13	Cell-Level Handshake Reception Timing	60
4-14	PHY Device Control Signal Timing	61
5-1	Control Memory after Initialization (free block pool)	64
5-2	Initialization Sequence	65
5-3	Configuration of Control Memory	66
5-4	Stack of Free Block Pool	67
5-5	Structure of Mailbox	69
5-6	Outline of Transmission Flow	70
5-7	Example of Structure of Transmit Data in System Memory	71
5-8	Format of Packet Descriptor	72
5-9	Location of Blank Descriptor	73
5-10	Example of Arrangement of Transmit Queue	73
5-11	Single-Buffer Mode	74
5-12	Multi-Buffer Mode	74
5-13	Insertion of CRC-10	76
5-14	Format of Packet Descriptor	77
5-15	Setting of LAST Bit	77
5-16	Structure of Transmit VC Table	79
5-17	Status of Transmit Channel	81
5-18	Example of Transmitting Cell (where 1 VC transmission, I/M = 1/2, P = 1, C = 1, ICM = “0”)	82
5-19	Concept of Scheduling Function	83
5-20	Concept of Scheduler Operation	83
5-21	Example of Actions of Parameters	86
5-22	Pointers Used for Linking	87
5-23	Linking/Releasing Transmit VC to/from Traffic Shaper	87
5-24	Shaper Pointer Entry of Control Memory	89

LIST OF FIGURES (2/2)

Figure No.	Title	Page
5-25	Shaper Status	89
5-26	Cell Transmission Sequence According to Priority	94
5-27	Outline of Transmit Operation	96
5-28	Outline of Reception Flow	99
5-29	Structure of Receive Pool in System Memory	101
5-30	Configuration of Pool	103
5-31	Location of Pool Descriptor in Receive Free Buffer Pool Pointer	104
5-32	Format of Pool Descriptor	104
5-33	Raw Cell Pool	106
5-34	Receive VC Table (for reception of packets other than MPEG-TS packets)	107
5-35	Receive VC Table (for reception of MPEG-TS packet)	108
5-36	Receive Channel Status	110
5-37	Receive Lookup Table (where SHIFT = 4)	112
5-38	Receive Operation Flowchart	114
5-39	Format of MPEG-TS Packet	115
5-40	MPEG-TS Packet Processing Flowchart	117
5-41	Configuration Example of Receive Data	118
5-42	Transfer Address of MPEG-TS Packet	120
5-43	Structure of Raw Cell Data Storage Pool (if raw cell data of VC#1 and VC#J are to be stored to the same pool)	120
5-44	Reassembly Timer (T1)	122
5-45	Format of Raw Cell Data	127
6-1	Command Issuance and B Bit	134
6-2	Closing Transmit Channel	140
6-3	Closing Receive Channel	141
6-4	Work of Host before Issuance of Add_Batches Command	143
8-1	Block Diagram of Boundary Scan Circuit	164
8-2	State Transition of TAP Controller	166
8-3	Operation Timing in Controller State	167
8-4	Operation of Test Logic (instruction scan)	172
8-5	Operation of Test Logic (data scan)	173

LIST OF TABLES

Table No.	Title	Page
4-1	Configuration Register	37
4-2	Selecting Enabled Burst Size	45
4-3	DMA Transfer by μPD98409	45
4-4	Burst Transfer Transition in Each Case	48
4-5	EEPROM Command List	53
4-6	EEPROM Format	54
5-1	Mailbox Operation of μPD98409	69
5-2	Function of Packet Descriptor	72
5-3	Description of Bits of Scheduler Register	85
5-4	Summary of Data Supported by μPD98409 Transmission Function	98
5-5	Types of Receive Data and Usable Pool	105
5-6	Errors Occurring in Any of First, Intermediate, and Last Cells of Packet	125
5-7	Error Taking Precedence When Occurring with Other Errors	125
6-1	Command Type	133
8-1	Operation in Each Controller State	171

CHAPTER 1 GENERAL

The μPD98409 is a high-performance SAR chip for segmentation and assembly of ATM cells. Provided with a PCI (Peripheral Component Interconnect) bus interface control memory and supporting a MPEG packet transfer engine function to mitigate the workload of the CPU in transferring compressed image data, this chip has ideal specifications for use in a set top box (STB) to interface with an ATM network. The μPD98409 conforms to ATM Forum recommendations and has AAL-SAR sublayer and ATM layer functions.

1.1 Features

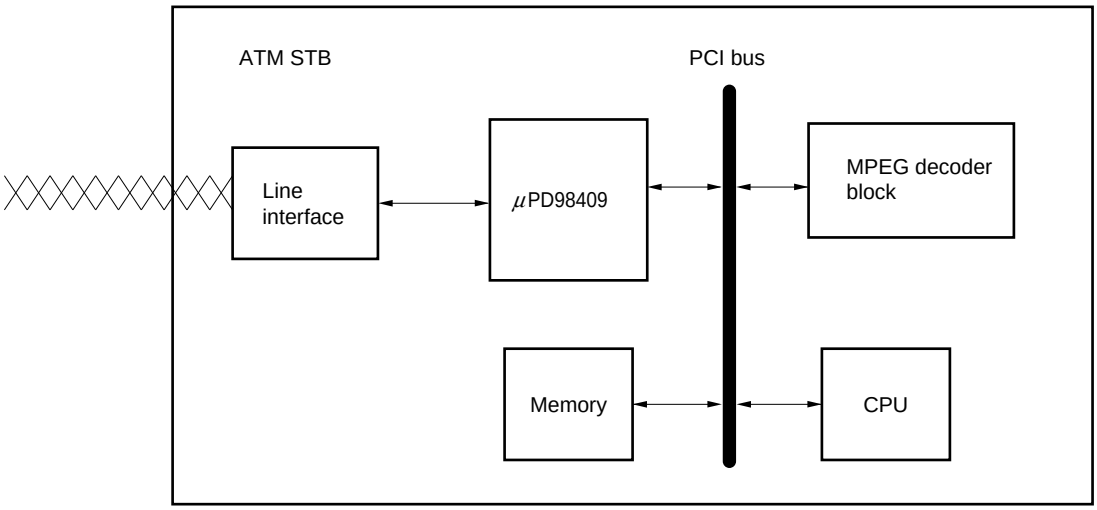
- Conforms to ATM Forum
- PCI bus interface (5/3.3 V, 32 bits, 33 MHz)
  Conforms to PCI Local Bus Specification Revision 2.1
- AAL-5 SAR sublayer and ATM layer functions
- Hardware support of AAL-5 processing (non-AAL-5 processing can be supported in software)
- Supports up to 64 virtual channels (VC) (64-VC control memory)
- Two traffic shapers for transmission scheduling
- MPEG packet transfer engine mitigating the workload of compressed image data transfer by CPU
- Receive FIFO of 12 cells
- PHY device I/F: UTOPIA Level-1 (octet/cell level handshake)
- JTAG boundary scan test function (IEEE 1149.1)
- 0.35-μm CMOS process, +5/+3.3-V power supply
  - Bus interface +5 V : +5/+3.3-V power supply
  - Bus interface +3.3 V : +3.3-V single power source

1.2 Ordering Information

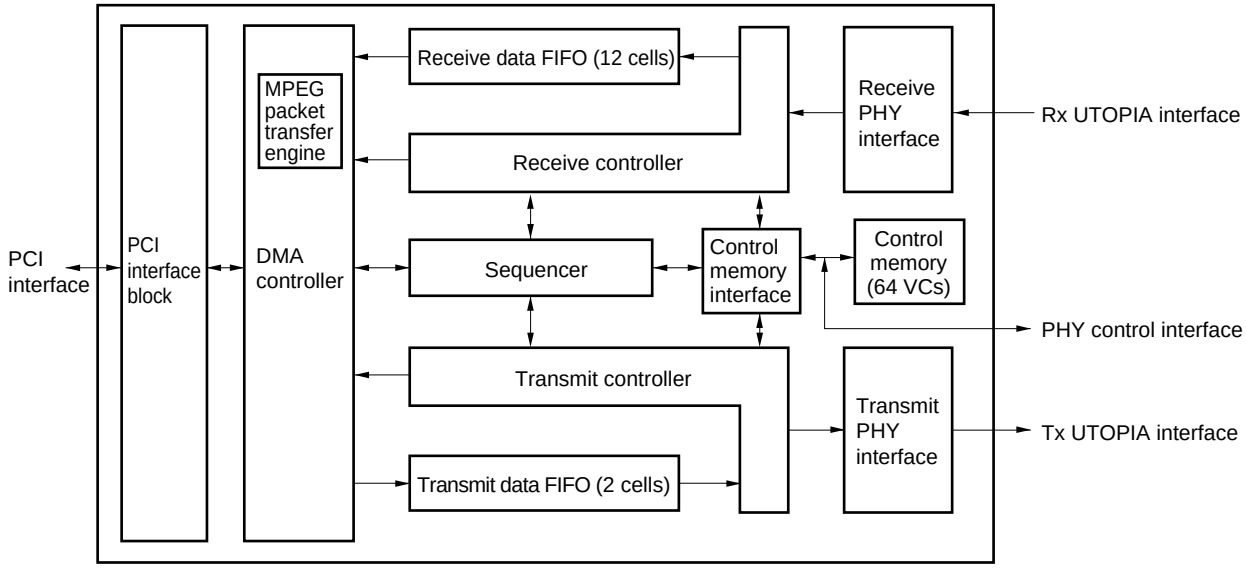
Part Number	Package
μPD98409GN-LMU	240-pin plastic QFP (fine pitch) (32 × 32)
★ μPD98409GN-LMU-A	240-pin plastic QFP (fine pitch) (32 × 32)

Remark The μPD98409GN-LMU-A is a lead (Pb)-free product.

1.3 Example of System Configuration

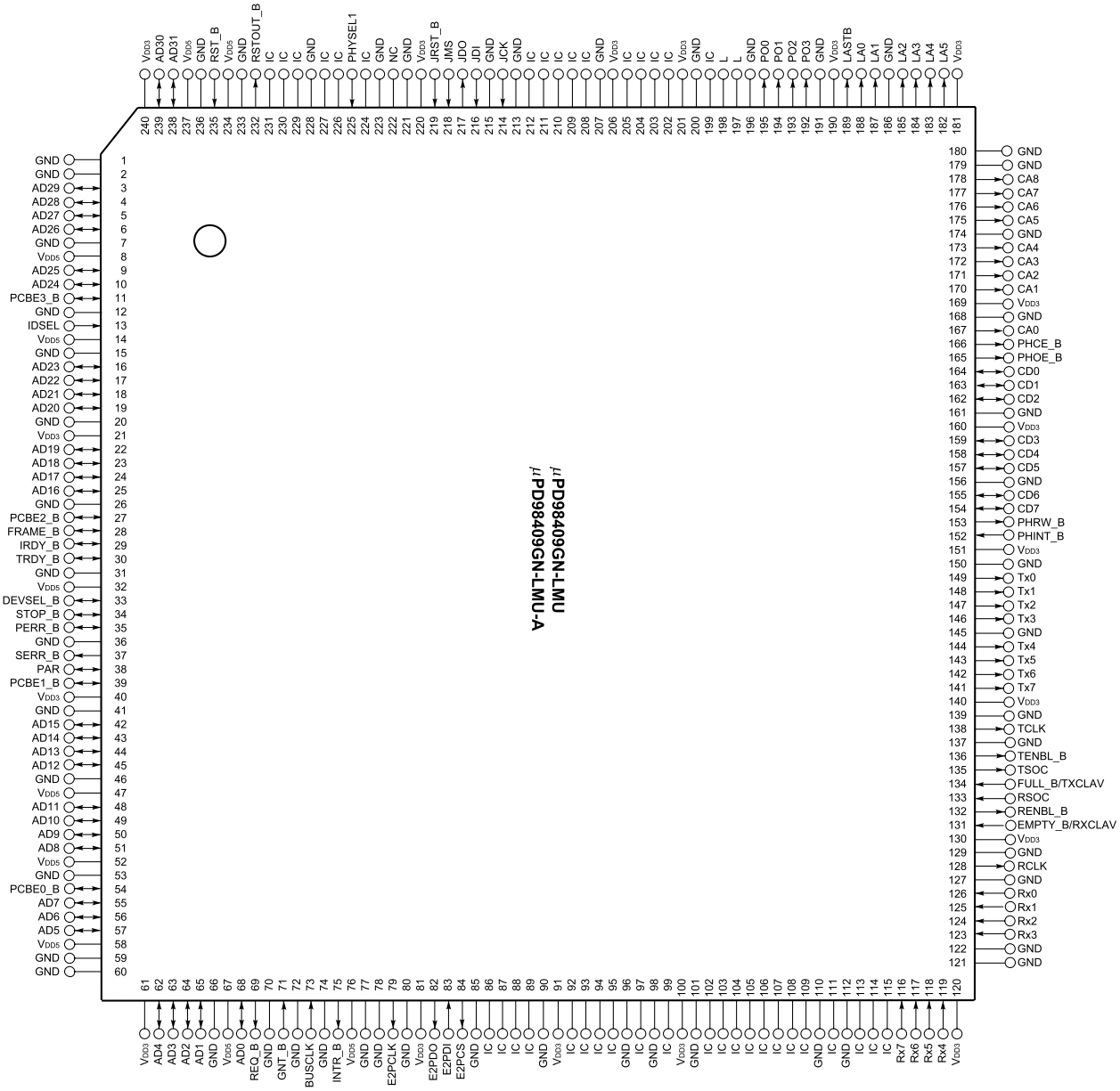


1.4 Block Diagram



2.1 Pin Configuration (Top View)

- **240-pin plastic QFP (fine pitch) (32 × 32)**



NC : No connection. Leave this pin open.

IC : Input pin with pull-down resistor for internal test. It is recommended to fix this pin to the low level.

L : Fix this pin to the low level.

PIN NAMES

AD31-AD0 : Address/Data
BUSCLK : Bus Clock
CA8-CA0 : PHY Device Address
CD7-CD0 : PHY Device Data
DEVSEL_B : Device Select
EMPTY_B/RxCLAV : PHY Empty / Rx Cell Available
E2PCLK : Clock for EEPROM™
E2PCS : EEPROM Chip Select
E2PDI : Serial Data Input from EEPROM
E2PDO : Serial Data Output to EEPROM
FRAME_B : Cycle Frame
FULL_B/TxCLAV : PHY Buffer Full / Tx Cell Available
GND : Ground
GNT_B : Grant
IDSEL : ID Select
INTR_B : Interrupt
IRDY_B : Initiator Ready
JCK : JTAG Test Pin
JDI : JTAG Test Pin
JDO : JTAG Test Pin
JMS : JTAG Test Pin
JRST_B : JTAG Test Pin
LA5-LA0 : Internal Test Pin
LASTB : Internal Test Pin
PAR : Parity
PCBE_B3-PCBE_B0 : Bus Command and Byte Enables
PERR_B : Parity Error
PHCE_B : PHY Chip Enable
PHINT_B : PHY Interrupt
PHOE_B : PHY Output Enable
PHRW_B : PHY Read/Write
PHYSEL1 : PHY Select
PO3-PO0 : Generic Output Port
RCLK : Receive Clock
RENBL_B : Receive Enable
REQ_B : Request
RSOC : Receive Start of Cell
RST_B : Reset
RSTOUT_B : Reset Output
Rx7-Rx0 : Receive Data Bus
SERR_B : System Error
STOP_B : Stop
TCLK : Transmit Clock
TENBL_B : Transmit Enable
TRDY_B : Target Ready
TSOC : Transmit Start of Cell
Tx7-Tx0 : Transmit Data Bus
V_{DD3} : +3.3 V Power Supply
V_{DD5} : +5 V Power Supply

2.2 Pin Function

The pin function of the μPD98409 is descibed below. A detailed explanation of how to use each pin, and the points to be noted in using the pins are given in **CHAPTER 4 INTERFACES**. Be sure to refer to this chapter.

Remark LV-TTL input : Can be connected to 5 V CMOS output
TTL output : Can be connected to 5 V TTL input, V_{OH} = 3.3 V, I_{OL} = 6 mA
CMOS output : 3.3 V CMOS output, V_{OH} = 3.3 V, I_{OL} = 12 mA
PCI input : 5/3.3 V PCI input
PCI output : 5/3.3 V PCI output

2.2.1 PHY device interface pin

PHY device interfaces include a UTOPIA interface through which the μPD98409 transfers ATM cells with a PHY device, and a PHY control interface by which the μPD98409 controls the PHY device.

(1) UTOPIA interface

(1/2)

Pin Name	Pin No.	I/O	I/O Level	Function
Rx7-Rx0	116 - 119 123 - 126	I	LV-TTL	Receive Data Bus. Rx7 through Rx0 constitute an 8-bit input bus which inputs data received from a network in byte format from a PHY device. The μPD98409 loads data in at the rising edge of RCLK.
RSOC	133	I	LV-TTL	Receive Start Cell. The RSOC signal is input in synchronization with the first byte of the cell data from a PHY device. This signal remains high while the first byte of the header is input to Rx7 through Rx0.
RENBL_B	132	O	TTL	Receive Enable. The RENBL_B signal indicates to a PHY device that the μPD98409 is ready to receive data in the next clock cycle.
EMPTY_B/ RxCLAV	131	I	LV-TTL	PHY Output Buffer Empty/Rx Cell Available. This signal notifies the μPD98409 that there is no cell data to be transferred in the receive FIFO and that no receive data can be supplied to the PHY device. When the UTOPIA interface is in the octet-level handshake mode, this signal serves as EMPTY_B, indicating that the data on Rx7 through Rx0 are invalid in the current clock cycle. In the cell-level handshake mode, it serves as RxCLAV, indicating that there is no cell to be supplied next after the transfer of the current cell is completed.
RCLK	128	O	TTL	Receive Clock. This is a synchronization clock used to transfer cell data with the PHY cell device at the receive side. The system clock input to the BUSCLK pin is output from this pin as is.
Tx7-Tx0	141 - 144 146 - 149	O	TTL	Transmit Data Bus. Tx7 through Tx0 constitute an 8-bit output bus which outputs transmit data in byte format to a PHY device. The μPD98409 outputs data at the rising edge of TCLK.
TSOC	135	O	TTL	Transmit Start of Cell. The TSOC signal is output in synchronization with the first byte of transmit cell data.

(2/2)

Pin Name	Pin No.	I/O	I/O Level	Function
TENBL_B	136	O	TTL	Transmit Enable. The TENBL_B signal indicates to a PHY device that data has been output to Tx7 through Tx0 in the current clock cycle.
FULL_B/ TxCLAV	134	I	LV-TTL	PHY Buffer Full/Tx Cell Available. This signal notifies the μPD98409 that the input buffer of the PHY device is full and that the device can receive no more data. When the UTOPIA interface is in the octet-level handshake mode, the PHY device inputs an inactive level to receive cell data. In the cell-level handshake mode, the PHY device inputs a signal that indicates that the PHY device can receive all the next one cell of data after the current cell has been completely transferred.
TCLK	138	O	TTL	Transmit Clock. This is a synchronization clock used to transfer cell data with the PHY device at the transmission side. The system clock input to the CLK pin is output from this pin as is.

(2) PHY device control interface

Pin Name	Pin No.	I/O	I/O Level	Function
PHRW_B	153	O	TTL	PHY Read/Write. The μPD98409 indicates the direction in which the PHY device is controlled, by using PHRW_B. 1: Read 0: Write
PHOE_B	165	O	TTL	PHY Output Enable. The μPD98409 enables output from the PHY device by making PHOE_B low
PHCE_B	166	O	TTL	PHY Chip Enable. The μPD98409 makes PHCE_B low to access a PHY device. This signal goes high after reset.
PHINT_B	152	I	LV-TTL	PHY Interrupt. This is an interrupt input signal from a PHY device. The PHY device indicates to the μPD98409 that it has an interrupt source, by inputting a low level to PHINT_B.
RSTOUT_B	232	O	TTL	Reset Output. This is a signal to reset a PHY device. The μPD98409 makes this pin low for the duration of 11 to 22 clock cycles when a low level is input to the RST_B pin or a software reset is executed.
CD7-CD0	154 - 155 157 - 159 162 - 164	I/O 3-state	LV-TTL in TTL out	PHY device data. CD7 through CD0 constitute an 8-bit data bus. These pins are three-state I/O pins. They are used to transfer data with a PHY device.
CA8-CA0	178 - 175 173 - 170 167	O	TTL	PHY device address. CA8 through CA0 constitute a 9-bit address bus that outputs an address to a PHY device during read/write operation.

2.2.2 Bus interface pins

The μPD98409 employs a 32-bit PCI bus interface as a bus interface with the host. This interface conforms to "PCI Local Bus Specification Revision 2.1".

(1/2)

Pin Name	Pin No.	I/O	I/O Level	Function
AD31-AD0	238 - 239 3 - 6 9, 10 16 - 19 22 - 25 42 - 45 48 - 51 55 - 57 62 - 65 68	I/O 3-state	PCI	Address/data. AD31 through AD0 are 32 bits of multiplexed address and data bus signals. When the μPD98409 operates as the bus master, it drives an address at the first one clock, and transfers data at the second clock and onward.
PCBE3_B PCBE2_B PCBE1_B PCBE0_B	11 27 39 54	I/O 3-state	PCI	Bus command and byte enable. These signals define "bus commands" (generated bus transaction) in an address phase. In a data phase, they indicate which byte lane holds valid data. The PCBE3_B pin corresponds to byte 3 (bits 31 through 24), and PCBE0_B pin corresponds to byte 0 (bits 7 through 0).
PAR	38	I/O 3-state	PCI	Parity. This signal inputs/outputs an even parity on the AD31 through AD0 and PCBE3_B through PCBE0_B pins including the PAR signal. When the μPD98409 operates as the master, the PAR signal is output in the address and write data phases. When the μPD98409 operates as a target, the PAR signal is output in the read data phase.
FRAME_B	28	I/O Sustained 3-state	PCI	Frame. This signal indicates the start and period of bus transaction. When this signal becomes active, it indicates the start of bus transaction. While it is active, data is transferred. When the next data transfer phase is for the last data of the transaction, this signal becomes inactive.
TRDY_B	30	I/O Sustained 3-state	PCI	Target ready. This signal goes low when the target device is ready to complete the transaction of the current data phase. This signal is used in pairs with IRDY_B. When both IRDY_B and TRDY_B are low, read/write data transfer is executed.
IRDY_B	29	I/O Sustained 3-state	PCI	Initiator ready. This signal goes low when the initiator is ready to complete the transaction of the current data phase. This signal is used in pairs with TRDY_B. When both IRDY_B and TRDY_B are low, read/write data transfer is executed. If both FRAME_B and IRDY_B are inactive, the bus cycle is not executed, and wait cycles are inserted until both IRDY_B and TRDY_B become active.

(2/2)

Pin Name	Pin No.	I/O	I/O Level	Function
STOP_B	34	I/O Sustained 3-state	PCI	Stop. This signal goes low when the target device requests the master device to stop the current transaction.
DEVSEL_B	33	I/O Sustained 3-state	PCI	Device select. This signal goes low when the μPD98409 operates as a target and recognizes an address after the FRAME_B signal has become active. When the μPD98409 operates as the master, it samples this signal to check to see whether a target device has been selected.
IDSEL	13	I	PCI	Initialization device select. This signal inputs a high level when the configuration register of the μPD98409 is read/written.
REQ_B	69	O[Note]	PCI	Request. The μPD98409 requests the arbiter for the bus mastership by making this signal low.
GNT_B	71	I	PCI	Grant. This signal goes low when the arbiter grants the μPD98409 the bus mastership.
PERR_B	35	I/O Sustained 3-state	PCI	Parity error. This signal indicates that the μPD98409 has detected a parity error. It is enabled when the “Parity Error Response” bit of the configuration register is set to 1.
SERR_B	37	O	N-ch open-drain	System error. This signal indicates that the μPD98409 has detected an address parity error. It is enabled when both the “Parity Error Response” and “System Error Enable” bits of the configuration register are set to 1.
INTR_B	75	O	N-ch open-drain	Interrupt output. Pull up this pin because it outputs an open-drain signal. INTR_B informs the CPU that the interrupt bit (not masked) of the GSR register is set.
BUSCLK	73	I	PCI	PCI bus clock. Bus clock input pin. It inputs a clock of up to 233 MHz.
RST_B	235	I	PCI	Reset. The RST_B signal initializes the μPD98409 (on starting). When a low level is input to RST_B, the internal state machine and registers of the μPD98409 are reset, and all the 3-state signals go into a high-impedance state. When this signal is input while the μPD98409 is operating, the operating status at that time is lost. Keep the input to RST_B low at least for the duration of 1 clock cycle. Do not access the μPD98409 at least for 20 clocks after it has been reset.

Note Although the “PCI Local Bus Specification Revision 2.1” specifies that the REQ_B pin go into a high-impedance state while a low level is input to the RST_B pin, the REQ_B pin of the μPD98409 outputs a high level.

2.2.3 Serial EEPROM interface pins

The μPD98409 has an interface for serial EEPROM supporting the MICROWIRE™ interface. Some of the contents of the PCI configuration register can be loaded from the EEPROM connected.

As the EEPROM, “NM93C46L” of National Semiconductor Corp. is recommended.

Pin Name	Pin No.	I/O	I/O Level	Function
E2PCS	84	O	TTL	EEPROM chip select. A chip select signal for EEPROM. Leave this pin open when it is not used.
E2PDI	83	I	TTL Internally pulled up	EEPROM data input. This pin is connected to the data output pin of the EEPROM. Pull up or open this pin when it is not used.
E2PDO	82	O	TTL	EEPROM data output. This pin is connected to the data input pin of the EEPROM. Pull up or open this pin when it is not used.
E2PCLK	79	O	TTL	EEPROM clock. This pin supplies a clock necessary for data transfer with the EEPROM. It outputs the clock input to the BUSCLK pin divided by 36. Leave this pin open when it is not used.

2.2.4 JTAG boundary scan pins

Pin Name	Pin No.	I/O	I/O Level	Function
JDI	216	I	LV-TTL	JTAG Test Data Input. The JDI pin is used to input data to the JTAG boundary scan circuit register. Normally, fix this pin to high or low level.
JDO	217	O 3-state	TTL	JTAG Test Data Output. The JDO pin is used to output data from the JTAG boundary scan circuit register. It changes output at the falling edge of the clock input to the JCK pin. Normally, leave this pin open.
JCK	214	I	LV-TTL	JTAG Test Clock. This pin is used to supply a clock to the JTAG boundary scan circuit register. Normally, fix this pin to a high or low level.
JMS	218	I	LV-TTL	JTAG Test Mode Select. Normally, fix this pin to a high or low level.
JRST_B	219	I	LV-TTL	JTAG Test Reset. This pin initializes the JTAG boundary scan circuit register. Normally, fix this pin to a low level.

Remark Processing of JTAG boundary scan pins not used (during normal operation)

The reason that the JRST_B pin is grounded when it is not used (during normal operation) is to better prevent malfunctioning of the JTAG logic. The JTAG pin may be also processed in either of the following ways:

- Reset the JTAG logic without using the JRST_B pin
  Reset the JTAG logic by using the JMS and JCK pins and keep it in the reset status (the JRST_B pin is pulled up).
  Fix the JMS pin to 1 (pull up) and input 5 clock cycles or more to the JCK pin.

- Reset the JTAG logic by using the JRST_B pin
  Input a low pulse of the same width as RST_B of the μPD98409 to the JRST_B pin. If both the JMS and JRST_B pins are pulled up and kept high, the JTAG logic is not released from the reset status. Therefore, the normal operation is not affected. Fix the input level of the JDI and JCK pins by pulling them down or up.

2.2.5 Other pins

Pin Name	Pin No.	I/O	I/O Level	Function
PHYSEL1	225	I	LV-TTL	Internal test pins. Input a low level to this pin.
PO3-PO0	192 - 195	O	CMOS	General-purpose output port. General-purpose output port pins. These pins output the value written to the GPOR register.
LA5-LA0	182 - 185 187 - 188	O	TTL	Internal test pins. Leave these pins open during normal operation.
LASTB	189	O	TTL	Internal test pin. Leave this pin open during normal operation.

2.2.6 Power and ground pins

Pin Name	Pin No.	I/O	Function
V_{DD3}	21, 40, 61, 81, 91, 100, 120, 130, 140, 151, 160, 169, 181, 190, 201, 206, 220, 240	—	+3.3-V power supply. These pins supply +3.3 V to the chip.
V_{DD5}	8, 14, 32, 47, 52, 58, 67, 76, 234, 237	—	+5 V power supply. These pins supply +5 V to the chip when a +5-V bus interface is used. Supply +3.3 V to these pins when a +3.3-V bus interface is used.
GND	1, 2, 7, 12, 15, 20, 26, 31, 36, 41, 46, 53, 59, 60, 66, 70, 72, 74, 77, 78, 80, 85, 90, 96, 98, 101, 110, 112, 121, 122, 127, 129, 137, 139, 145, 150, 156, 161, 168, 174, 179, 180, 186, 191, 196, 200, 207, 213, 215, 221, 223, 228, 233, 236	—	Ground.

2.2.7 Pin status during and after reset

Pin Name	During Reset	After Reset
RENBL_B	1	1
RCLK	CLK output	CLK output
Tx7-Tx0	0	0
TSOC	0	0
TENBL_B	1	1
TCLK	CLK output	CLK output
PHRW_B	0	0
PHOE_B	1	1
PHCE_B	Hi-Z	Hi-Z
AD31-AD0	Hi-Z (input)	Hi-Z (input)
PCBE3_B-PCBE0_B	Hi-Z (input)	Hi-Z (input)
PAR	Hi-Z (input)	Hi-Z (input)
FRAME_B	Hi-Z (input)	Hi-Z (input)
TRDY_B	Hi-Z (input)	Hi-Z (input)
IRDY_B	Hi-Z (input)	Hi-Z (input)
STOP_B	Hi-Z (input)	Hi-Z (input)
DEVSEL_B	Hi-Z (input)	Hi-Z (input)
REQ_B	1	1
PERR_B	Hi-Z (input)	Hi-Z (input)
SERR_B	Hi-Z	Hi-Z
INTR_B	Hi-Z	Hi-Z
E2PCS	0	0
E2PDO	0	0
E2PCLK	0	0
CD7-CD0	0	0
CA8-CA0	0	0
PO3-PO0	1	1
LA5-LA0	0	0
LASTB	0	1
JDO[Note]	Hi-Z	Hi-Z

Note When JRST_B input

[MEMO]

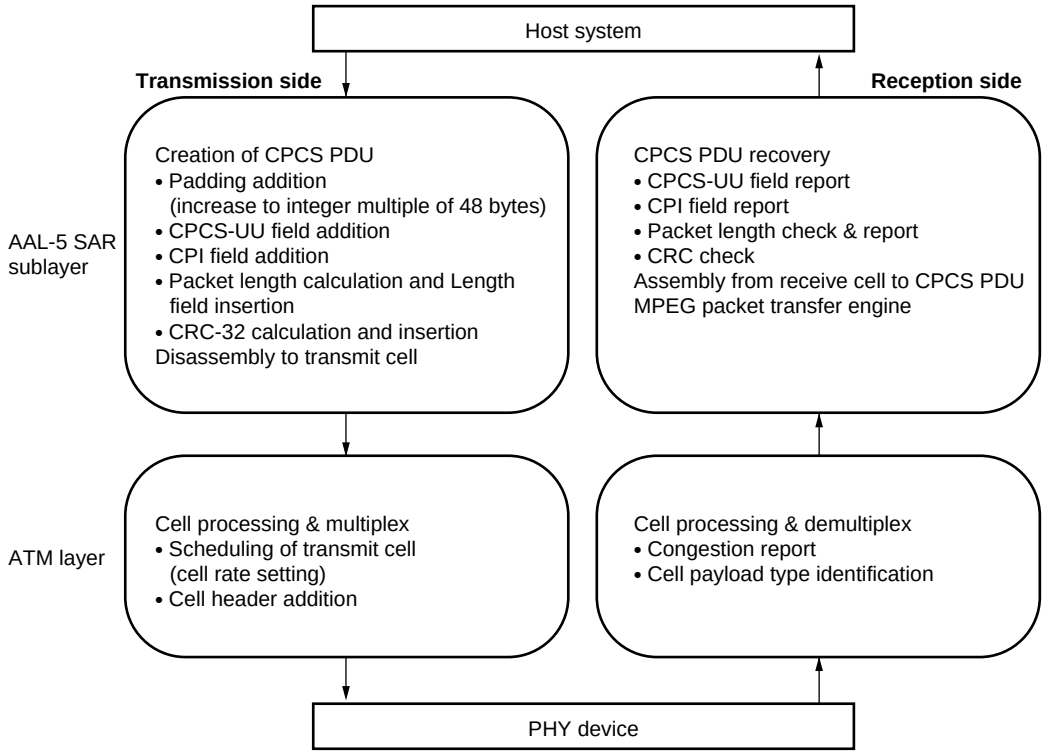
CHAPTER 3 FUNCTIONAL OUTLINE OF μPD98409

This chapter outlines the functions of the μPD98409. For details on each function, refer to **CHAPTER 5 OPERATIONS OF μPD98409**.

3.1 Functional Outline

The ATM function of the μPD98409 supports the AAL-5 SAR sublayer and ATM layer of the ATM adaptation layer of the ATM protocols in hardware.

Figure 3-1. Functions of μPD98409



The μPD98409 is placed between the host system and a PHY device and is controlled by the host by accessing the internal registers of the μPD98409 via the PCI bus interface. The transmit/receive data is directly transferred with the system memory under the management of the host, by using the internal DMA controller. An indication for each packet is transferred to the system memory to indicate the status of completion of transmission/reception. Therefore, the host must allocate the following three areas to the system memory to transfer or receive data using the μPD98409.

(a) Transmit buffer area : Stores transmit data.
(b) Receive buffer area : Stores receive data.
(c) Mailbox area : Stores transmit/receive indication.

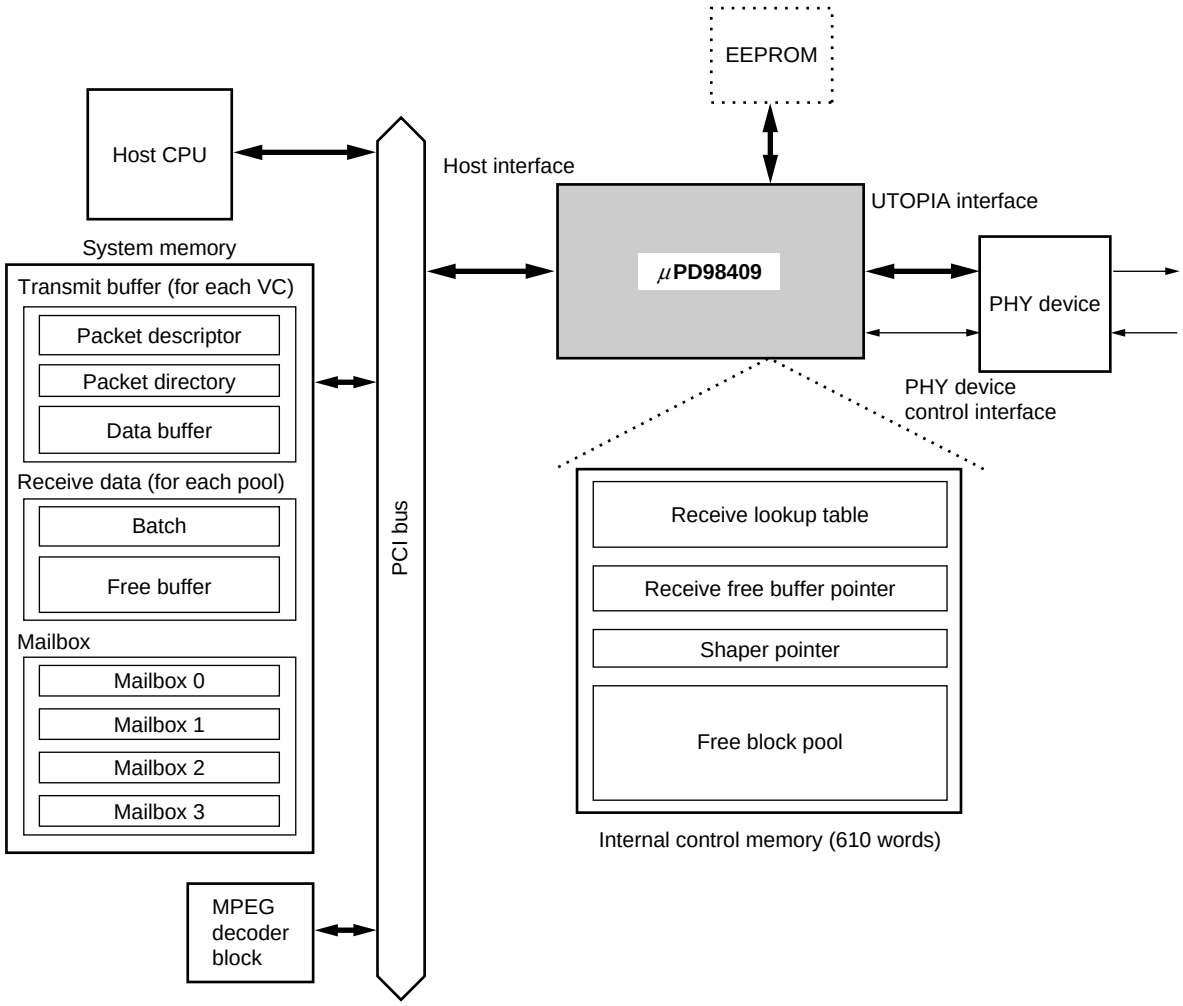
The μPD98409 uses the internal control memory to execute transmission/reception. A control memory of 64 VC (610 words) is provided. This memory is divided into the following four areas. The boundaries of the four areas are set by the host in a register of the μPD98409.

<1> Receive lookup table area
<2> Receive free buffer pool pointer area
<3> Shaper pointer area
<4> Free block pool area

For the details of the control memory, refer to **5.2 Setting of Control Memory**.

The μPD98409 and a PHY device transfer data with each other in cell format. The μPD98409 makes the transmit data read in segment units into cells by internal processing, and transfers these cells to the PHY device via the UTOPIA interface. The cells received from the PHY device are transferred to the system memory under management of the host in cell units after the reception processing by the μPD98409. The system outline of the μPD98409 is shown below.

Figure 3-2. System Outline of μPD98409

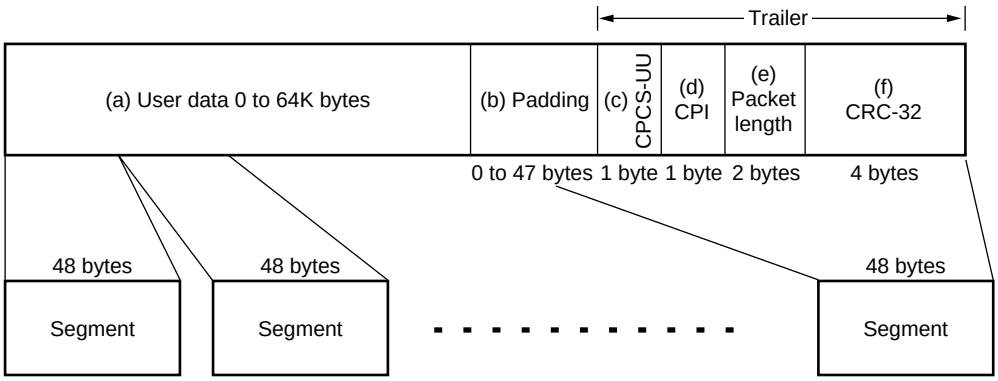


3.2 AAL-5 SAR Sublayer Function

When transmitting data, the μPD98409 appends padding to the user data of different lengths (0 to 65535 bytes) prepared by the host system so that the data is an integer multiple of 48 bytes, and creates a CPCS-PDU of AAL-5 by adding the trailer shown in Figure 3-3. The created a CPCS-PDU is disassembled into 48-byte segments.

When receiving data, the μPD98409 deletes the overhead from the received cell, and assembles the CPCS-PDU in the receive buffer of the system memory. It also checks the trailer of the packet, detects errors if any, and reports to the host. The CPCS-PDU is stored in the receive buffer with not only user data (a) but also fields (b) through (f) appended.

Figure 3-3. PDU Format of ALL-5



(a) User data field:
Data of up to 65535 bytes in length.

(b) Padding field:
A field of 0 to 47 bytes inserted between the user data and trailer to make the CPCS-PDU an integer multiple of 48 bytes. The μPD98409 automatically inserts data of all zeros.

(c) CPCS user-user information (CPCS-UU) field:
Used to transfer CPCS user-user information. The μPD98409 can set and transmit any data.

(d) Common part identifier (CPI) field:
This field is used as a all-zero field to make the CPCS-PDU trailer 8 bits. The other usage and set values are pending. This field of the packet transmitted by the μPD98409 can be set by the user at will.

(e) Packet length (Length) field:
Displays the user data length (0 to 65535) of the CPCS-PDU in binary in byte units. This field is calculated and inserted by the μPD98409 at the transmission side. At the reception side, this received field and the result of the packet size actually received are verified.

(f) CRC-32 field:
Sets a CRC code, justifying it to the right.
The creation polynomial is as follows:

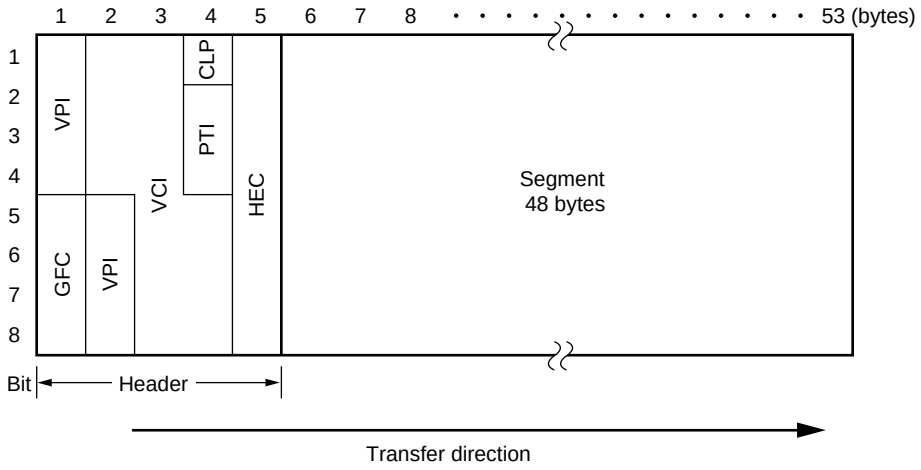
$$\text{Expression} = 1 + x + x^2 + x^4 + x^5 + x^7 + x^8 + x^{10} + x^{11} + x^{12} + x^{16} + x^{22} + x^{23} + x^{26} + x^{32}$$

3.3 ATM Layer Functions

3.3.1 Creation of cells

The μPD98409 creates a cell by appending the 5-byte header information shown in Figure 3-4 to a segment.

Figure 3-4. Cell Structure of User Network Interface (UNI)



GFC : Generic Flow Control
VPI : Virtual Path Identifier
VCI : Virtual Channel Identifier
PTI : Payload Type Identifier
CLP : Cell Loss Priority
HEC : Header Error Control

The function of each field of the header, and the functions supported by the μPD98409 are explained next.

(1) GFC (generic flow control) field

The GFC field is used for control information to prevent a cell conflict.

<Function of μPD98409>

Transmission : Inserts the pattern set by the user for each packet to this field.
Reception : Ignores this field, and does not report its contents to the host (except when receiving a raw cell).

(2) VPI/VCI field (virtual path identifier/virtual channel identifier)

VPI and VCI are routing bits used for identification when data is multiplexed at virtual path (VP) level and virtual channel (VC) level.

<Function of μPD98409>

Transmission : Inserts the 24 bits of the entire range of VPI/VCI set by the user.

Reception : Supports up to 15 bits of VPI/VCI. Some low-order bits of VPI and some low-order bits of VCI are concatenated. For the algorithm to reduce from 24 bits to 15 bits, refer to **5.5.4 Setting of receive look-up table**.

In the μPD98409, the settings for one channel differ for transmit VC and receive VC.

The μPD98409 can support up to 64 active VCs (virtual channels) (in any combination of receive and transmit VCs). A VC (virtual channel) is used to identify the transmit and receive channels in the μPD98409, and is different from the VCI field in meaning.

(3) PTI field (payload type identifier)

This 3-bit field indicates whether the payload type of a cell is user data or layer management information. This field also includes forward congestion indication (EFCI), indicating that a cell has passed through a congested network node.

The codes in the PTI field are allocated as follows:

PTI	Usage
000	User data cell, without congestion, SDU type = 0
001	User data cell, without congestion, SDU type = 1
010	User data cell, with congestion, SDU type = 0
011	User data cell, with congestion, SDU type = 1
100	OAM F5 cell (segment support)
101	OAM F5 cell (end-end support)
110	Resource management cell
111	Reserved for future functions

SDU type = 0: All segments except the last cell of an AAL-PDU

SDU type = 1: Last cell of an AAL-PDU. This cell includes a trailer.

OAM F5 cell: Specific OAM cell having VCC (Virtual Channel Connection) operation information

Resource management cell (RM cell): Specific cell having network resource information

<Function of μPD98409>

Transmission : The μPD98409 asigns the pattern set by the user to this field as is and transmits it. When a user data cell is set and processing of AAL-5 type is selected, the μPD98409 changes the least significant bit to “1” to transmit the last segment. When the pattern of an OAM cell is set, raw cell transmission processing is executed.

Reception : The μPD98409 monitors the PTI field of a receive cell, identifies the received cell as an OAM cell, resource management cell, or user data cell, and performs processing accordingly. If the received cell is an OAM or resource management cell, it is processed in cell units as a raw cell. If a user data cell is received, processing as an AAL-5 packet is performed. In addition, it can be specified that user data other than AAL-5 be received as a raw cells. When an AAL-5 packet is received, the μPD98409 checks the least significant bit of the PTI field to determine the last segment of the packet.

PTI Field	Processing of μPD98409
000	Can receive cells as user data
001	
010	
011	
100	Can receive cells as raw cell
101	
110	
111	

(4) CLP (Cell Loss Priority)

This field is used to indicate whether the cell takes precedence in being lost when the network is congested. When CLP = 1, it indicates that the cell takes precedence and is lost.

<Function of μPD98409>

Transmission : The user can select the following three modes for each transmit packet.

- CLP = 0 for all cells
- CLP = 1 for all cells
- CLP = 1 for cell except last cell of packet, CLP = 0 for last cell

Reception : The μPD98409 monitors each receive packet and when it has received a cell with CLP = 1, it reports the receive indication of that packet to the host.

(5) HEC field (header error control)

This field is processed by the TC sublayer of a physical layer and is used for cell synchronization, and header error detection and correction.

<Function of μPD98409>

Transmission : The μPD98409 inserts dummy data “00H” to this field and transmits it.

Reception : The μPD98409 ignores this field.

3.3.2 Setting of cell transmission rate

The μPD98409 has 2 shapers that control the transmission rate of created cells. Each shaper executes a dual leaky bucket algorithm. The parameters for the algorithm are set by the user for each shaper. The user can select which of the shapers is used for each channel.

Of the 2 shapers, one shaper can be set as an unassigned cell/idle cell generator. The shaper specified as an unassigned cell/idle cell generator functions as a shaper that transmits only an unassigned cell/idle cell at the rate given by the user. By using this unassigned cell/idle cell generator function, the sum of the bandwidth at which all the channels transmit data can be limited.

3.3.3 Support of non-AAL-5 traffic

The μPD98409 has a function to process received cells as raw cells to support traffic other than AAL-5 (AAL cells, OAM cells, and resource management cells (RM cells) other than AAL-5). The VC set to execute raw cell processing does not execute processing such as appending a trailer as in AAL-5 packet transmission, but simply makes transmit data into cells and transmits the data. During reception, the VC stores the 53-byte receive cells with headers in the system memory along with 11-byte indication information in cell units (for the format, refer to **5.5.7 (2) Raw cell data**). Processing such as header analysis for each cell and trailer verification is executed by the host via software.

The μPD98409 also has a function to insert/verify CRC-10 to mitigate the workload of the host in processing AAL-3/4 cells, OAM cells, and RM cells.

In the normal mode, each time a raw cell has been received, it is reported by an interrupt. The host executes the software processing each time one cell has been received. In the VC mode, the raw cell of the packet of AAL-3/4 cell is received, the μPD98409 checks the ST field of the AAL-3/4 cell, and an interrupt is generated only if the last cell or a packet with only one cell has been received.

[MEMO]

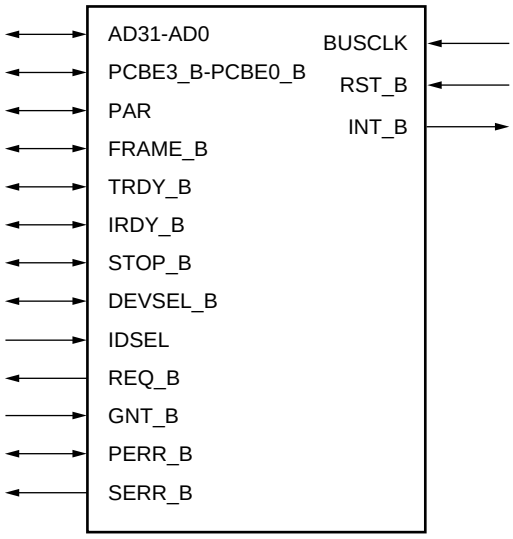
CHAPTER 4 INTERFACES

The μPD98409 has a host bus interface, PHY device interface. This chapter explains in detail the function and operation of each interface.

4.1 Bus Interface

The bus interface of the μPD98409 is a 32-bit PCI bus interface.

Figure 4-1. Bus Interface Signals



4.1.1 Features of PCI bus interface

- Conforms to “PCI Local Bus Specification Revision 2.1”
- Maximum operating frequency: 33 MHz
- Command response as medium-speed device
- Equipped with configuration register space conforming to PCI specifications and supports configuration cycle
- Accesses internal registers of μPD98409 by both memory and I/O commands
- Processes all memory read commands (memory read, memory read multiple, and memory read line) as the same memory read
- Processes all memory write commands (memory write, memory write multiple, and memory write line) as the same memory write
- Can respond to each target end (retry, disconnect, target abort) when μPD98409 operates as master
- 0-wait transfer (as master device)
- Support of serial EEPROM interface
- Retry Timer and TRDY Timer functions detect abnormal operation of the target.
- Operation as 5V PCI or 3.3 V PCI with PCI supply voltage (V_{DD5} pin)

4.1.2 Configuration register

The configuration register is mapped to a PCI configuration space. The contents of the configuration register are not initialized by software reset. All the contents of the configuration register are initialized to the default values by hardware reset.

For the contents of the configuration register, refer to **PCI Specifications** "**PCI Local Bus Specification Revision 2.1**".

Offset 40H is allocated to a register peculiar to the μPD98409 that sets an additional function.

Figure 4-2. Layout of Configuration Register

Address	31 24	23 16	15 8	7 0
00H	Device ID		Vendor ID	
04H	Status		Command	
08H	Class code			Revision ID
0CH	BIST	Header type	Latency timer	Cache line size
10H	IO base address			
14H	Memory base address			
18H 1CH 20H 24H 28H	Reserved			
2CH	Subsystem ID		Subsystem vendor ID	
30H 34H 38H	Reserved			
3CH	Max_Lat	Min_Gnt	Interrupt pin	Interrupt line
40H	Reserved		Retry timer	TRDY timer

Remark BIST :Built-in self test

Table 4-1. Configuration Register (1/2)

Offset	Name	Bit	R/W	Default	Description
00H	Device ID	31-16	R	0036H	Device ID of μPD98409
	Vendor ID	15-0	R	1033H	Vendor ID of NEC
04H	Status	31	R/W	0	Detected Parity Error bit
		30	R/W	0	Signaled System Error bit
		29	R/W	0	Received Master Abort bit
		28	R/W	0	Received Target Abort bit
		27	R/W	0	Signaled Target Abort bit
		26-25	R	01	DEVSEL_B timing. μPD98409 supports Medium.
		24	R/W	0	Data Parity Error Reported bit
		23	R	1	Fast Back-to-Back Capable bit
		22-16	R	00H	Reserved
	Command	15-10	R	00H	Reserved
		9	R	0	Fast Back-to-Back Enable bit
		8	R/W	0	System Error Enable bit
		7	R	0	Wait Cycle Enable bit
		6	R/W	0	Parity Error Response bit
		5	R	0	VGA Palette Snoop Enable bit
		4	R/W	0	Memory Write and Invalidate Enable bit
		3	R	0	Special Cycle Recognition bit
		2	R/W	0	Bus Master Enable bit
		1	R/W	0	Memory Access Enable bit
		0	R/W	0	I/O Access Enable bit
08H	Class Code	31-24	R	02	Basic class: Network controller
		23-16	R	03	Subclass: ATM controller
		15-8	R	00	Programming interface
	Revision ID	7-0	R	02H	Revision information of device
0CH	BIST	31-24	R	00H	Used to control built-in self-test function and to check status. Not supported by μPD98409.
	Header Type	23-16	R	00H	Header type of PCI configuration space.
	Latency Timer	15-8	R/W	00H	Sets master latency timer value. Bits [2:0] are read-only bits of “000“.
	Cache Line Size	7-0	R/W	00H	Specifies cache line size of system in word (32 bits) units.
10H	IO Base Address	31-8	R/W	00H	IO base address. 256-byte support.
		7-1	R	00H	Reserved
		0	R	1	IO Space Indicator bit

Table 4-1. Configuration Register (2/2)

Offset	Name	Bit	R/W	Default	Description
14H	Memory Base Address	31-12	R/W	00H	Memory base address. 4K-byte support
		11-4	R	0H	Reserved
		3	R	0	Prefetchable. Disabled by μPD98409.
		2, 1	R	00	Type (base address can be mapped to any position of 32-bit width). Fixed to 00
		0	R	0	Memory Space Indicator bit.
18H	Reserved	31-0	R	all 0	
1CH					
20H					
24H					
28H					
2CH	Subsystem ID	31-16	R	0000H	These registers sets ID peculiar to board on which PCI device is mounted, or to subsystem. These registers can be loaded from the external serial EEPROM
	Sybsystem Vender ID	15-0	R	0000H	
30H	Reserved	31-0	R	all 0	
34H					
38H					
3CH	Max_Lat	31-24	R	00H	Specify setting of latency timer value. These registers can be loaded from the external serial EEPROM
	Min_Gnt	23-16	R	00H	
	Interrupt Pin	15-8	R	01H	Specifies interrupt pin. INTA_B is used.
	Interrupt Line	7-0	R/W	00H	Specifies interrupt line of interrupt controller to which interrupt signal of μPD98409 is connected.
40H	Reserved	31-16	R	all 0	
	Retry Timer	15-8	R/W	00H	This register is peculiar to the μPD98409. It specifies the maximum number of times that successive retry is permitted by the μPD98409. After reset, 00H is loaded to the timer and the Retry Timer function is disabled. To enable the function, assign a value other than 0 to this register. If a transaction with the target cannot be performed after retry has been repeated the specified number of times with the Retry Timer function enabled, the FERR bit of the GSR register is set and master operation stops until it is reset.
	TRDY Timer	7-0	R/W	00H	This register is peculiar to the μPD98409. It specifies the maximum number of clocks during which the μPD98409 waits for TRDY_B. After reset, 00H is loaded to the timer and the TRDY Timer function is disabled. To enable the function, assign a value other than 0 to this register. If TRDY_B does not become active the specified number of clocks after FRAME_B has been made active with the TRDY Timer function enabled, the FERR bit of the GSR register is set and the master operation stops until it is reset.

If the μPD98409 detects a serial EEPROM after power application, the Subsystem Vendor ID, Subsystem ID, and Min_Gnt/Max_Lat fields of the above configuration register are loaded from the external serial EEPROM.

4.1.3 PCI bus transaction

(1) Slave transaction

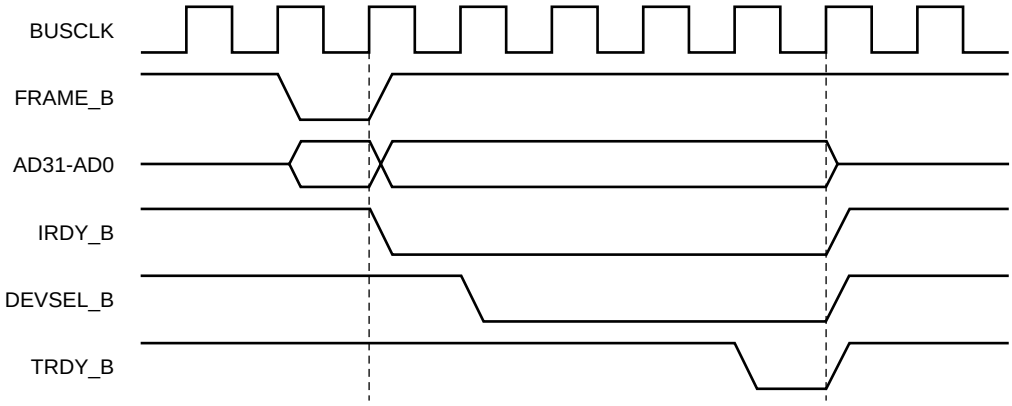
During a slave transaction, the μPD98409 performs 1-word burst transfer and responds to all memory commands/IO commands provided that all the data byte enable signals 3 through 0 (PCBE3_B through PCBE0_B) are 0.

During a slave transaction using the μPD98409, all memory read commands (memory read, memory read multiple, and memory read line) are processed as memory read commands. All memory write commands (memory write and memory-write-and-invalidate) are processed as memory write commands.

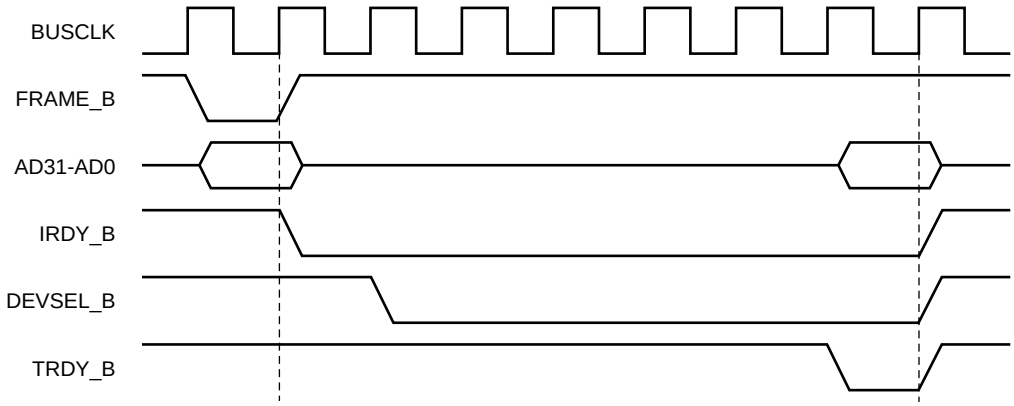
If an abnormal address is received when the slave is being accessed, the μPD98409 does not return an acknowledgment (does not make DEVSEL_B active). Figure 4-3 shows the slave transaction timing.

Figure 4-3. Slave Transaction Timing

(a) Write transaction



(b) Read transaction



(2) Master transaction

The μPD98409 supports bursts of 1, 2, 4, 8, 12, 13, and 16 words for master transaction. In particular, 13-word burst is executed for byte-alignment transfer of data of one cell to the system memory. The μPD98409 automatically sets the burst size to 1 to 47 words during MPEG-TS (Transport Stream) packet transfer. Note that byte alignment transfer cannot be performed during MPEG-TS packet transfer.

The μPD98409 requests to serve as the master of the PCI bus by making REQ_B active. The bus arbiter allows the μPD98409 to serve as the master by making GNT_B active. The μPD98409 samples FRAME_B and IRDY_B at the rising edge of BUSCLK and waits until the PCI bus enters the idle status. When both the signals become inactive and the μPD98409 detects the idle status of the PCI bus, it starts the transaction.

In a write transaction in which the μPD98409 transfers data to the system memory, the μPD98409 indicates that it is starting the transaction by making FRAME_B active. FRAME_B is kept active until the data unit before the last is transferred. The first clock edge at which the μPD98409 has made FRAME_B active is an address phase in which it drives an address onto AD31 through AD0, and a transaction type onto PCBE3_B through PCBE0_B. The next clock edge is a data phase in which data is driven onto AD31 through AD0. In addition, PCBE3_B through PCBE0_B are driven to indicate the position of the valid byte on AD31 through AD0. When the μPD98409 detects that both TRDY_B and IRDY_B have become active, it recognizes that the first data phase has been completed, and drives the next data onto AD31 through AD0.

In a read transaction in which data is transferred from the system memory to the μPD98409, the μPD98409 indicates the start of the transaction by making FRAME_B active. FRAME_B is kept active until the data unit before the last is transferred. At the first clock edge at which FRAME_B is made active, an address is driven onto AD31 through AD0 and a transaction type, onto PCBE3_B through PCBE0_B. At the next clock edge, the μPD98409 stops driving AD31 through AD0, and allows the target to control the bus. At the same clock edge, the μPD98409 changes the information it drives onto PCBE3_B through PCBE0_B to indicate the position of the valid byte on AD31 through AD0. The μPD98409 also indicates that it is ready to receive the first data from the target by making IRDY_B active. When the μPD98409 samples that TRDY_B and IRDY_B are active, it latches the first data on AD31 through AD0. The target drives the next data, and makes TRDY_B active to indicate this.

The μPD98409 issues a read/write command according to the following rules during master transaction.

(a) Read transaction

- Memory read : When number of transfer words = 1
- Memory read line : When number of transfer words > 1 and when μPD98409 reads up to next cache boundary line
- Memory read multiple : When data block to be read straddles over cache boundary

(b) Write transaction

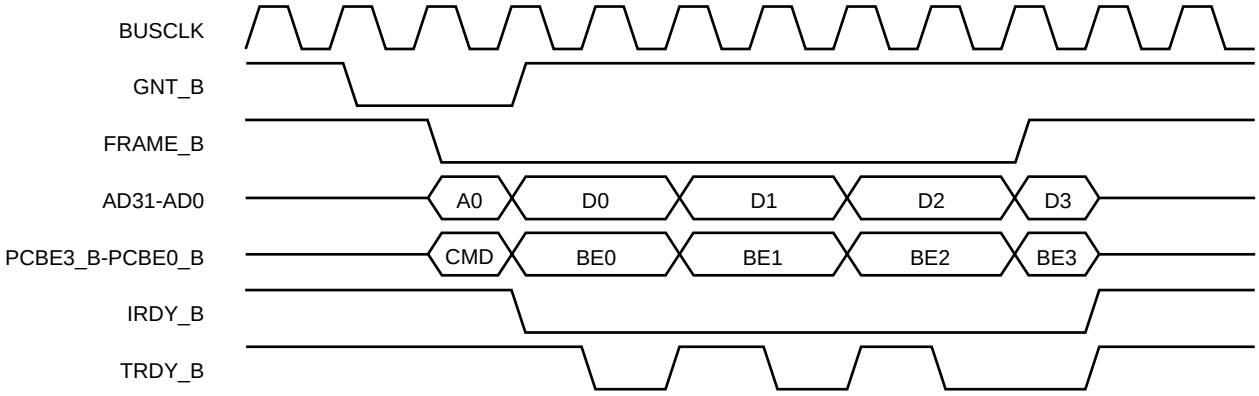
- Memory write and invalidate: When all the three conditions below are satisfied

<1> When the number of transfer words is the same as the number of cache lines
<2> When the “Memory Write and Invalidate Enable” bit of the configuration register is set to 1
<3> If the start address of write transaction is at a cache boundary.

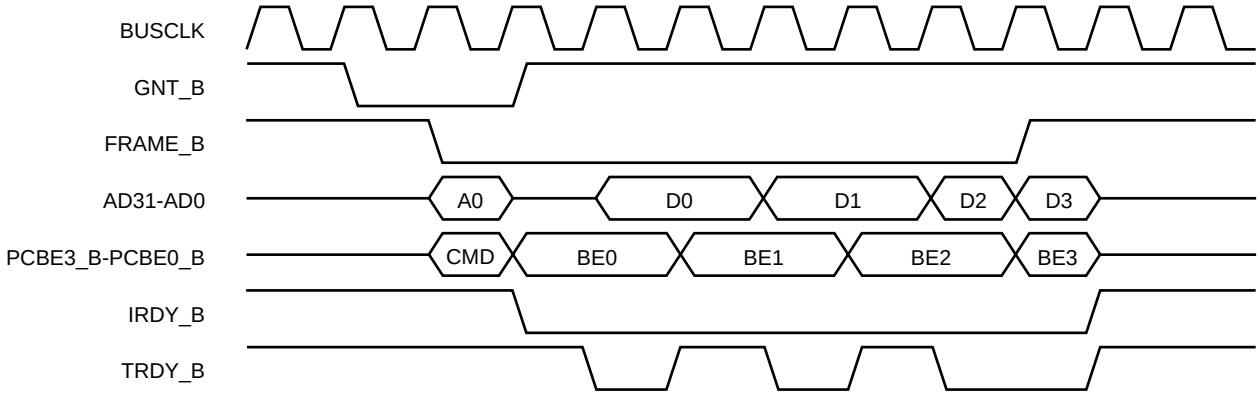
- Memory write: Conditions other than above

Figure 4-4. Master Transaction Timing

(a) PCI write transaction (from μPD98409 to target)



(b) PCI read transaction (from target to μPD98409)



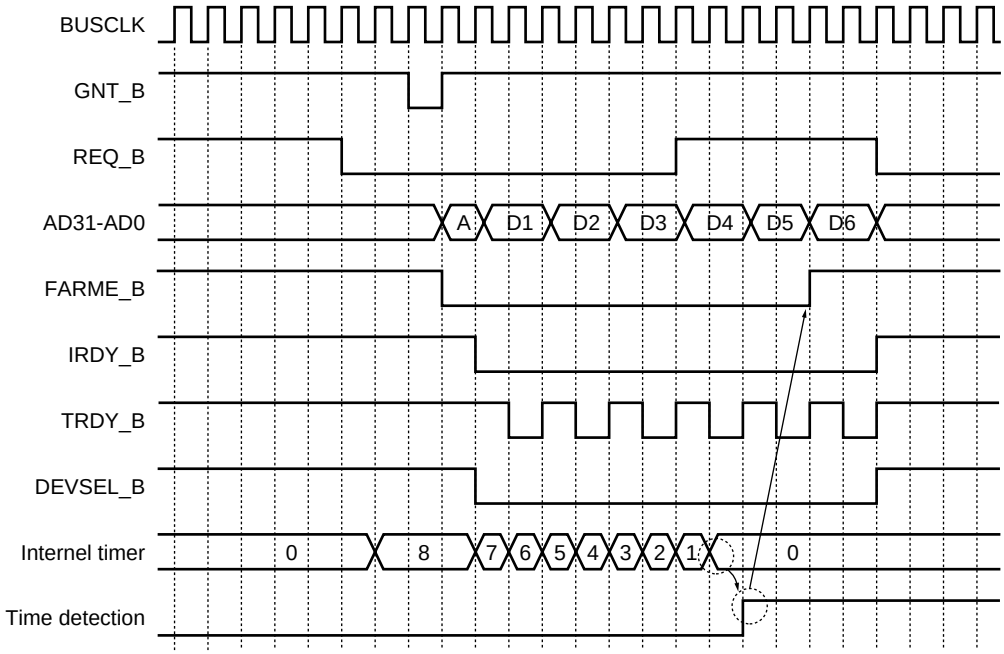
(3) End of transaction with μPD98409 as master

The transaction ends in the following three ways when the μPD98409 serves as the master.

<1> When transaction ends normally (normal end)

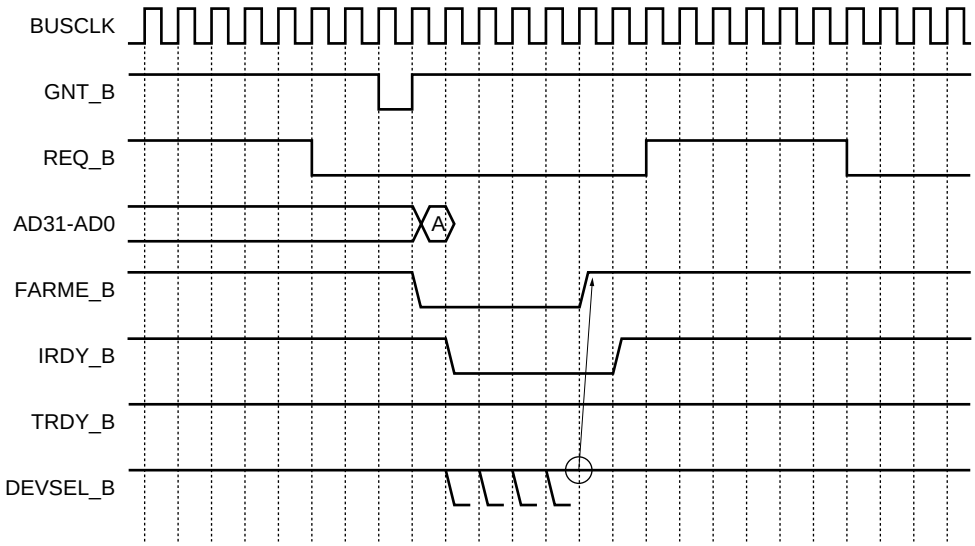
<2> If Latency Timer time has elapsed after another master makes GNT_B of the μPD98409 inactive to acquire the bus (time out)

Example: End of transaction when Latency Timer is eight



<3> If there is no target that returns a response to an address (master abort)

Example: Transaction when DEVSEL_B does not become active.



(4) End of transaction with μPD98409 as target

When the μPD98409 serves as a target, it may request the end of transaction by executing retry, disconnect, or target abort.

(a) Retry

The host issues a command to the ECCR register of the μPD98409 when it accesses an external EEPROM. Because it takes the μPD98409 a certain time to access the external EEPROM, the μPD98409 cannot successively accept commands. Therefore, the μPD98409 requests a retry when its ECCR register is accessed successively. The μPD98409 asks the host to retry if the host has performed high-speed back-to-back transaction with an internal register of the μPD98409 after it has accessed an internal register of the μPD98409 (direct address register or configuration register).

On power application, the μPD98409 detects the connection of the serial EEPROM, and asks the host to retry if the host has made an access (including the configuration cycle) while the configuration register is being automatically loaded.

(b) Disconnect

If an internal register of the μPD98409 (direct address register or configuration register) is accessed by means of burst transfer, the μPD98409 makes STOP_B active when one word has been transferred, and disconnects the transaction.

(c) Target abort

- The μPD98409 executes target abort if the address it has received includes a parity error.
- Target abort is also executed if data with PCBE3_B through PCBE0_B not being all 0 is transacted.

(5) Status information of PCI

The μPD98409 has two status bits related to the PCI bus operation in the GSR register. These bits are set only when the μPD98409 operates as the master.

(a) PERR: bit 22

This bit indicates a state of the PCI bus interface related to a parity error, and is set to 1 under the following conditions:

- If the μPD98409 detects that the target has made the PERR_B signal active during master write.
- If the μPD98409 detects a parity error in data phase during master read.

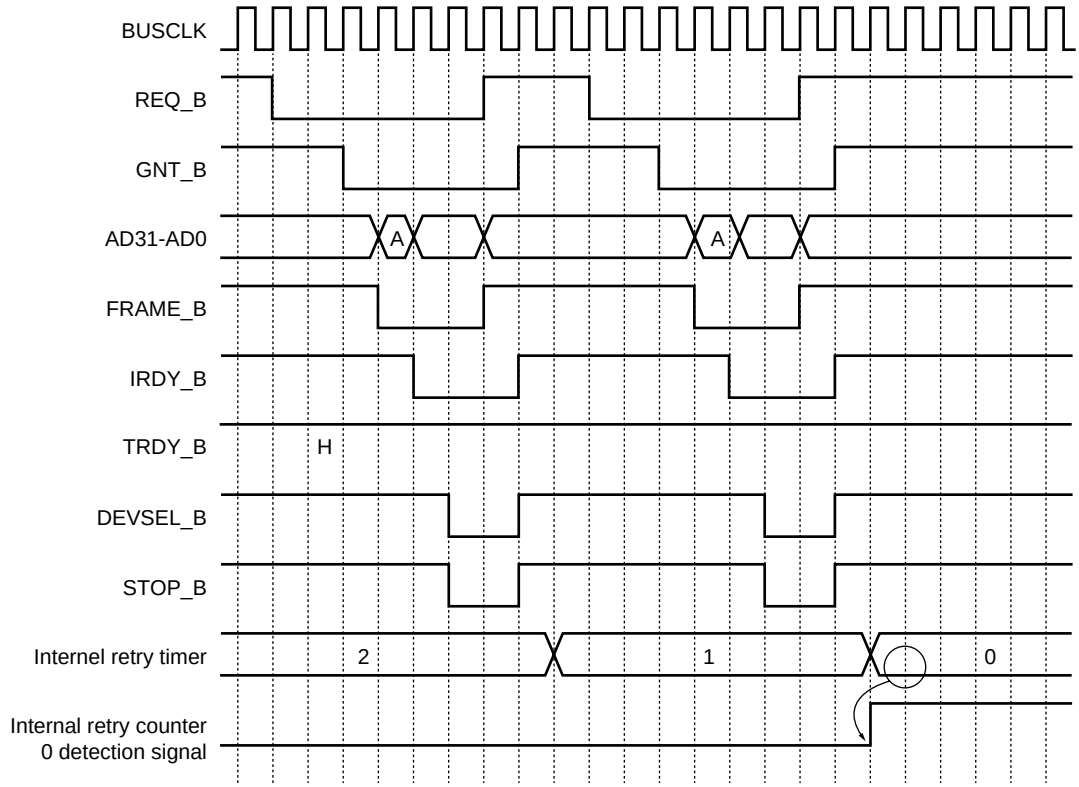
Caution The operation of the μPD98409 on and after detection of a parity error cannot be guaranteed. If it detects a parity error, reset the μPD98409.

(b) FERR: bit 21

This bit reports to the host that one of the following fatal errors has been detected during data transfer. When the FERR bit is set, the μPD98409 stops the operations related to the bus other than slave access. When the FERR bit is set, execute reset.

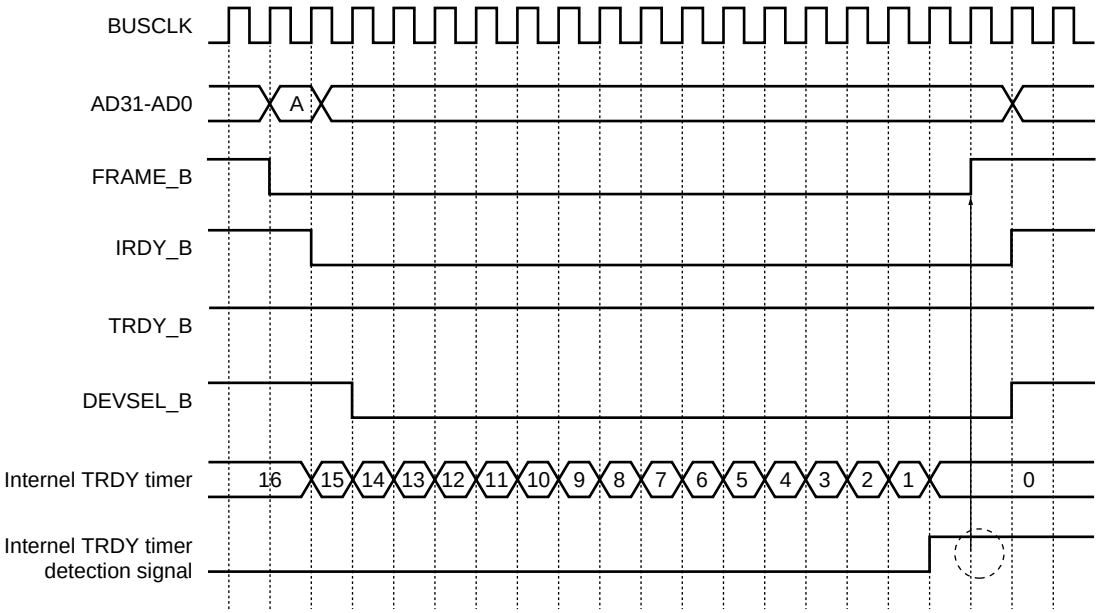
<1> The μPD98409 executes master abort because the target did not make DEVSEL_B active.
<2> The target has executed target abort by making STOP_B active.
<3> The retry timer counts the number of retries up to the number of times specified by the Retry Timer register of the configuration register, and then the transaction ends.

Example: Timing chart when retry timer register is set to “2”



<4> The TRDY Timer counts the number of clocks specified by the TRDY Timer register of the configuration register, and then the transaction ends.

Example: Timing chart when TRDY Timer register is set to 16



(6) Burst transfer

The μPD98409 supports burst transfer of 1, 2, 4, 8, 12, 13, and 16 words. The user can select the burst size to be enabled by using the "SZ" or "TBE" field of the GMR register. During MPEG-TS packet transfer, however, the burst size is automatically set by the μPD98409 to 1 to 47 words, regardless of the setting of the "SZ field" or "TBE field" of the GMR register.

Table 4-2. Selecting Enabled Burst Size

Selects burst size to be executed (GMR register, bits 11 through 8: SZ field, bit 16: TBE field, bit 7: AD field)		
SZ field	Bit 11	1: Enables 16-word burst, 0: Disables
	Bit 10	1: Enables 8-word burst, 0: Disables
	Bit 9	1: Enables 4-word burst, 0: Disables
	Bit 8	1: Enables 2-word burst, 0: Disables
TBE bit	Bit 16	1: Enables 12-word burst, 0: Disables (AD bit must always be set to 1.)
Default = all 0 (Only 1-word transfer is supported.)		

The host can enable two or more burst sizes at the same time. One-word transfer is always enabled regardless of the setting of the "SZ" and "TBE" fields. To enable 12-word burst by setting the TBE bit to 1, the AD bit of the GMR register must also be set to 1, and the burst size select function must be disabled. 13-word burst is executed for byte alignment transfer of data of 1 cell (12 words).

When 16-word transfer is enabled, the μPD98409 executes 16-word burst only when it writes raw cells to the system memory.

Table 4-3 shows the data transferred by the master (DMA) operation of the μPD98409. For the meaning of each operation, refer to **CHAPTER 5 OPERATIONS OF μPD98409**. Some data must start from a word (32-bit) boundary, and others can start from a byte boundary when it is transferred.

Table 4-3. DMA Transfer by μPD98409

Read/Write	Data Type	Number of Words	Byte Boundary
Read	Packet descriptor	4 words	Disabled
	Buffer descriptor	2 words	Disabled
	Transmit cell data	1 to 12 words	Enabled
	Buffer address in receive batch	1 word	Disabled
Write	Transmit indication	1 word	Disabled
	Receive indication	4 words	Disabled
	Receive cell data (other than MPEG-TS packet)	12 words	Enabled
	Receive cell data (MPEG-TS packet)	1 to 47 words	Disabled
	Receive batch link pointer	1 word	Disabled
	Raw cell data	16 words	Disabled

(7) Burst size select function

When two or more burst sizes are enabled, the μPD98409 checks the address field at the transfer destination, and automatically selects and determines the burst size to be executed from the enabled burst size. This function is valid during both read and write transactions with the μPD98409 serving as the master. However, it cannot be used when 12-word burst is enabled. During MPEG-TS packet transfer, the burst size is automatically set by the μPD98409 to 1 to 47 words, regardless of the setting of this function.

Transfer Destination Address	Executable Burst Size
xxxxxxxx xxxxxxxx xxxxxxxx xx0000xx	16-[Note], 8-, 4-, 2-, 1-word burst
xxxxxxxx xxxxxxxx xxxxxxxx xxx000xx	8-, 4-, 2-, 1-word burst
xxxxxxxx xxxxxxxx xxxxxxxx xxx100xx	4-, 2-, 1-word burst
xxxxxxxx xxxxxxxx xxxxxxxx xxxx10xx	2-, 1-word burst
xxxxxxxx xxxxxxxx xxxxxxxx xxxxx1xx	1-word transfer

Note 16-word burst takes precedence only when raw cell data is transferred.

Burst size	16	8	4	2	1
Priority	High → low				

The priorities of the burst size are 8-, 4-, 2-, and 1-word in that order, with 8-word burst having the highest priority. The μPD98409 selects the size with the highest priority from the executable and enabled burst sizes for execution.

This function can be enabled or disabled by the “AD bit” in the GMR register.

Selects burst transfer mode		
AD bit (GMR: bit 7)	0	When the μPD98409 executes master transfer, the function to check the address field at the transfer destination and determine the actual burst size from the enabled size according to the address is enabled.
	1	When the μPD98409 executes master transfer, the function to check the address field at the transfer destination is enabled. The μPD98409 simply selects the biggest burst size from the enabled burst sizes for execution.
	Default = 0	

Caution If 12-word burst is enabled by setting TBE to 1, the AD bit must be always set to 1 to disable the above function.

Next, an example of an operation where this function is enabled (AD = 0) or disabled (AD = 1) is shown below.

<Example> The μPD98409 stores each of the cells it receives in the system memory by means of master transaction. Because the payload of an ATM cell is 48 bytes long, the cell data to be transferred by the μPD98409 is 12 words long. How the receive cell data is stored in each case is shown below on the assumption that the start address of the receive buffer is “00000000H”.

Example	AD Bit	Enabled Burst Size
Case <1>	0	4-word, 8-word
Case <2>	1	4-word, 8-word
Case <3>	0	8-word only
Case <4>	0	2-word only (In this example, the operation is the same as when AD = 1 because the start address starts from all 0.)
Case <5>	0	Disables all burst sizes (In this example, the operation is the same as when AD = 1 because the start address starts from all 0.)

Table 4-4. Burst Transfer Transition in Each Case

Cell	Word	Address	Burst Size Generated in Each Case				
			<1>	<2>	<3>	<4>	<5>
First cell	1	00000000	8	8	8	2	1
	2	00000004					1
	3	00000008				2	1
	4	0000000C					1
	5	00000010				2	1
	6	00000014					1
	7	00000018				2	1
	8	0000001C					1
	9	00000020	4	4	1	2	1
	10	00000024			1		1
	11	00000028			1	2	1
	12	0000002C			1		1
Second cell	1	00000030	4	8	1	2	1
	2	00000034			1		1
	3	00000038			1	2	1
	4	0000003C			1		1
	5	00000040	8		8	2	1
	6	00000044					1
	7	00000048				2	1
	8	0000004C					1
	9	00000050		4		2	1
	10	00000054					1
	11	00000058				2	1
	12	0000005C					1
Third cell	1	00000060	8	8	8	2	1
	2	00000064					1
	3	00000068				2	1
	4	0000006C					1

(8) Byte alignment transfer

The μPD98409 can place transmit/receive cell data, except raw cell data, from a byte boundary of the system memory. When it executes master transaction, the μPD98409 checks the low-order 2 bits, AD1 and AD0, of the start address. If these bits are other than 00, the μPD98409 executes byte alignment.

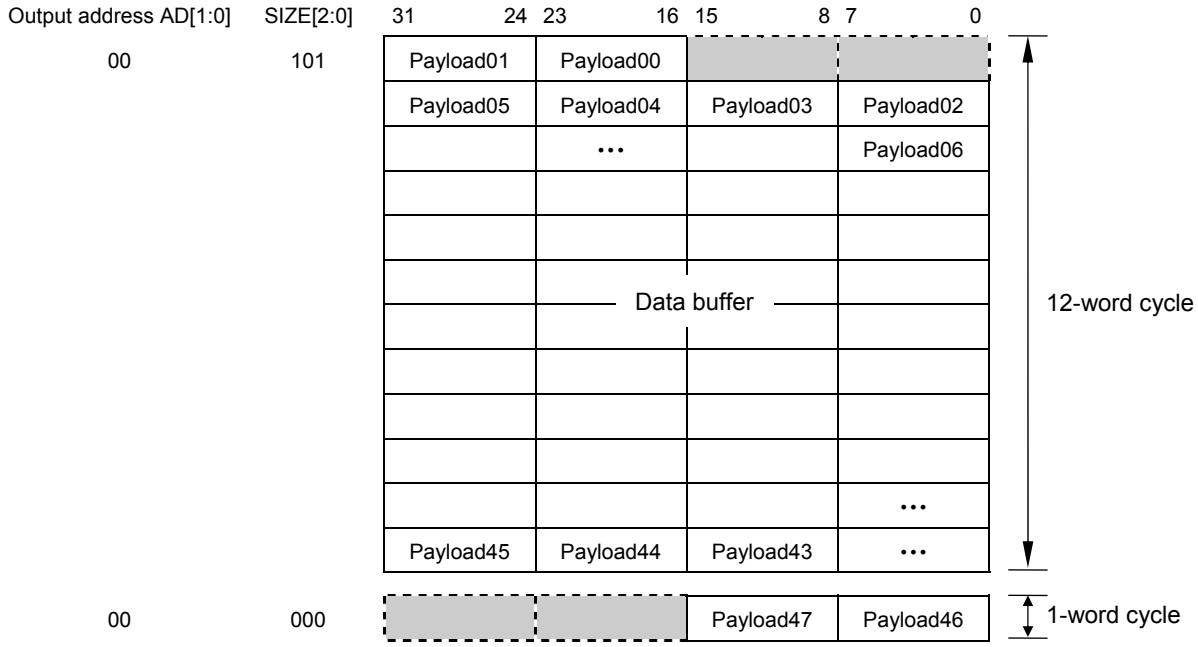
Because the MPEG-TS packet transfer address cannot be set at a byte boundary, byte alignment transfer does not take place during MPEG-TS packet transfer. Because data other than cell data cannot be assigned from a byte boundary, byte alignment transfer does not take place.

(a) Byte alignment transfer of transmit cell data

Transmit cell data is loaded in segment units (48 bytes) from the data buffer on the system memory by means of read transaction. The start address of the data buffer does not have to be placed at a 32-bit boundary. Even if read transaction of transmit data does not start from a 32-bit boundary, the μPD98409 reads data in word units (32 bits), and internally ignores unnecessary bytes.

<Example> If 12-word data of one cell is stored in the data buffer, and its start address starts from a byte boundary as shown in Figure 4-5 with the 12-word burst enabled

Figure 4-5. Transmit Cell Data Byte Alignment



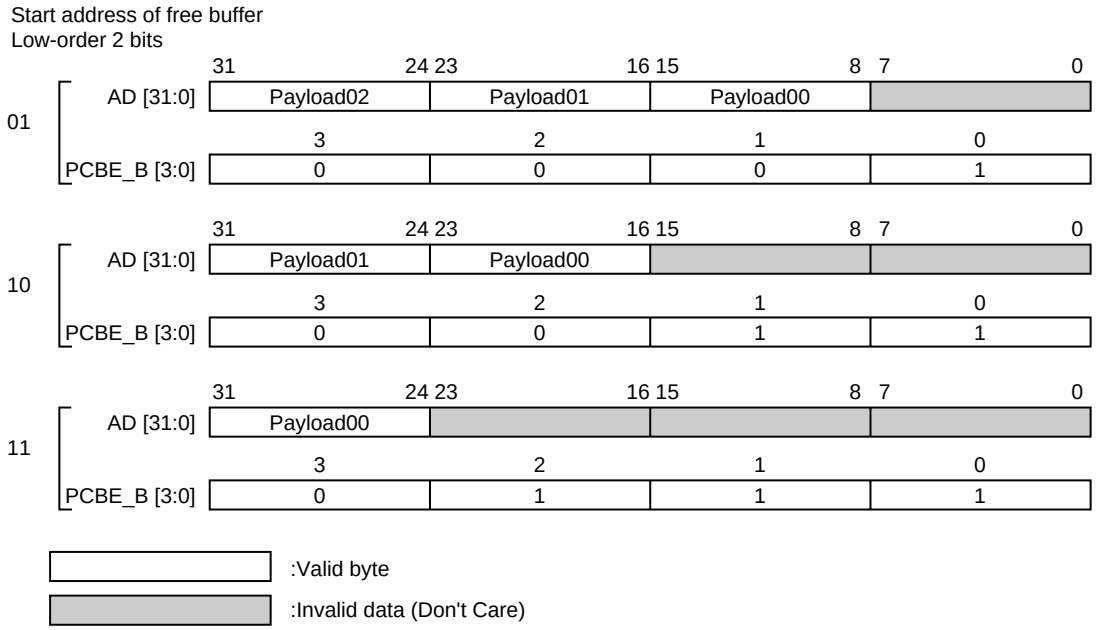
The μPD98409 first executes a read burst cycle of 12 words. The high-order 2 bytes of the first byte are internally dropped by the μPD98409. Next, one word is read. The low-order 2 bytes are dropped. In a transaction that is executed on a one-by-one basis like this, the μPD98409 outputs a 32-bit boundary address whose low-order 2 bits are 00.

(b) Byte alignment transfer of receive cell data

The μPD98409 stores receive cell data other than MPEG-TS packet in the free buffer in the system memory by means of a write transaction. The start address of the free buffer does not have to start from a 32-bit boundary but can also start from a byte boundary. If the low-order 2 bit of the start address of the free buffer (start address of the free buffer set in a batch) are other than 00, such as 01, 10, and 11, the μPD98409 writes data in 32-bit word units, and outputs signals indicating the valid byte to PCBE3_B through PCBE0_B.

Because the MPEG-TS packet transfer address cannot be set at a byte boundary, byte alignment transfer does not take place during MPEG-TS packet transfer.

Figure 4-6. AD[1:0] and PCBE_B[3:0] Pins during Byte Alignment Write Transfer



Next, an example where the low-order 2 bits of the start address of the free buffer are ‘10’ and the 12-word burst cycle is enabled is shown below.

Figure 4-7. Example of Storing Cell Data by Byte Alignment

Step-1 Buffer start address AD[1:0] = 10, execution of 13-word burst

First cell	31 24	23 16	15 8	7 0	PCBE_B [3:0] 3 2 1 0
Buffer address[1:0]=00	Payload1	Payload0			0 0 1 1
01	Payload5	Payload4	Payload3	Payload2	0 0 0 0
02	Payload9	Payload8	Payload7	Payload6	0 0 0 0
03	Payload13	Payload12	Payload11	Payload10	0 0 0 0
0a	Payload41	Payload40	Payload39	Payload38	0 0 0 0
0b	Payload45	Payload44	Payload43	Payload42	0 0 0 0
0c			Payload47	Payload46	1 1 0 0

Step-2 Execution of 13-word burst from last word of first cell

Second cell					
Buffer address[1:0]=0c	Payload1	Payload0			0 0 1 1
0d	Payload5	Payload4	Payload3	Payload2	0 0 0 0
0e	Payload9	Payload8	Payload7	Payload6	0 0 0 0
0f	Payload13	Payload12	Payload11	Payload10	0 0 0 0
16	Payload41	Payload40	Payload39	Payload38	0 0 0 0
17	Payload45	Payload44	Payload43	Payload42	0 0 0 0
18			Payload47	Payload46	1 1 0 0

don't care

If the start address is at a byte boundary when 12-word bust is enabled and receive cell data is written, 13-word burst is executed. 13-word burst is executed only in this case.

4.2 Serial EEPROM Interface

4.2.1 Serial EEPROM interface

The μPD98409 has an interface to connect an external serial EEPROM. In the external EEPROM, the Subsystem Vendor ID, Subsystem ID, and Min_Gnt/Max_Lat fields of the configuration register are stored. After reset, these fields are automatically loaded to the internal configuration register of the μPD98409. The μPD98409 checks whether an EEPROM is connected after hardware reset. If it is connected, the μPD98409 automatically loads the EEPROM contents. If no EEPROM is connected, automatic loading is not performed.

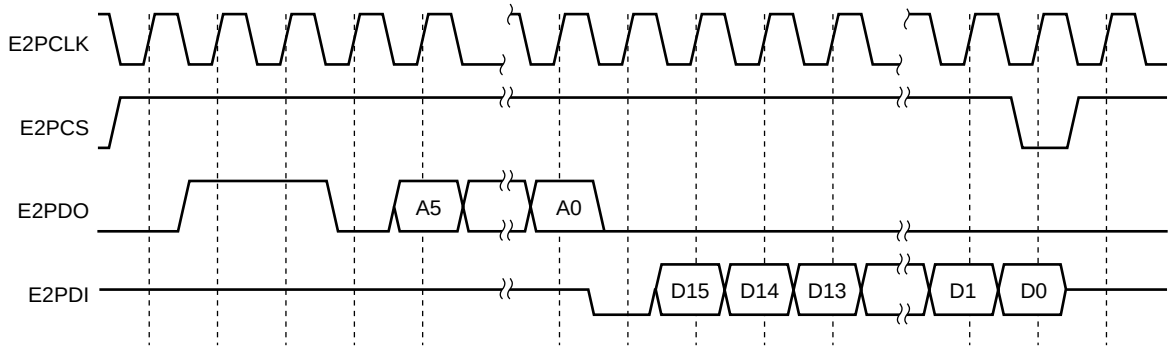
The μPD98409 does not perform automatic loading by software reset. The EEPROM interface uses the following signal lines:

- E2PCS (output) : Chip select line
- E2PDI (input) : Serial data input line
- E2PDO (output) : Serial data output line
- E2PCLK (output) : Clock output line (system clock divided by 36)

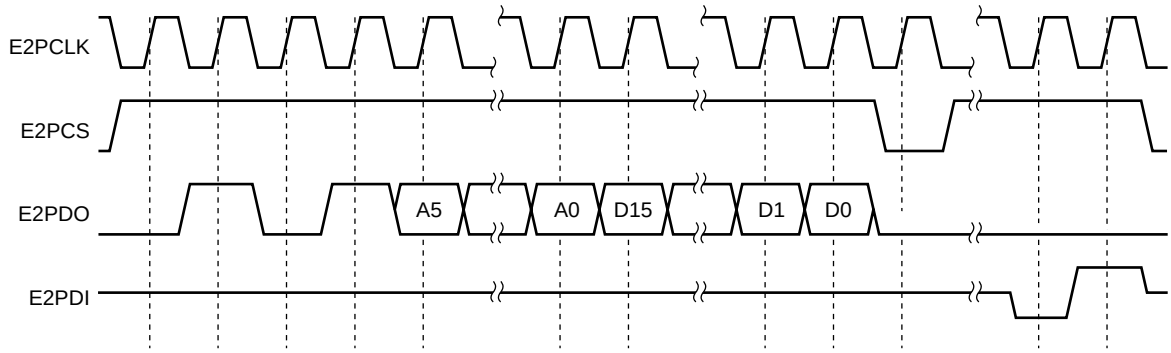
Remark The EEPROM interface of the μPD98409 employs MICROWIRE serial interface and is assumed to be connected to the National Semiconductor Corp. NM93C46 serial EEPROM.

Figure 4-8. EEPROM Interface Timing

Read timing



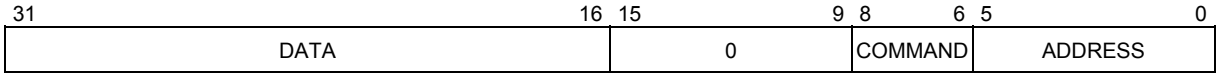
Write timing



4.2.2 Accessing EEPROM

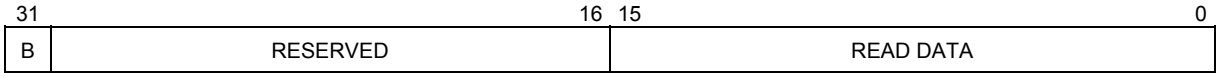
The EEPROM is accessed via two direct address registers of the μPD98409: ECCR and ERDR registers. The ECCR register sets the command and write data for the EEPROM, and the ERDR register stores the data read from the EEPROM.

ECCR register (address = 30H)



DATA : Sets 2-byte data to be written to the EEPROM.
COMMAND : Sets commands for the EEPROM.
ADDRESS : Sets the address of the EEPROM.

ERDR register (address = 34H)



B : 0 = Data of READ DATA field is valid.
1 = EEPROM is executing a command, or data of READ DATA field is still valid.
READ DATA : Read data from EEPROM.

Table 4-5. EEPROM Command List

Command	Setting		Description
	COMMAND	ADDRESS	
READ	110	Address specification	Read. Reads data at a specified address from the EEPROM, and stores it in the low-order 16 bit of the ERDR register.
WRITE	101	Address specification	Write. Writes the data set to the DATA field to the EEPROM.
ERASE	111	Address specification	Erase. Erases the data of the EEPROM at a specified address.
EWEN	100	11xxxx	Erase/write enable. Enables writing and erasing the EEPROM.
ERAL	100	10xxxx	Erase all. Erases all the areas of the EEPROM.
WRAL	100	01xxxx	Write all. Writes the data set to the data field to all the areas of the EEPROM.
EWDS	100	00xxxx	Erase/write disable. Disables writing and erasing the EEPROM.

When the host accesses the EEPROM, it writes a command to the ECCR register of the μPD98409. When the μPD98409 has received the command, it executes the command operation via the EEPROM interface. In a write transaction of one word in which the command is set, the μPD98409 releases the bus to the other masters by making TRDY_B active even if the set command is not completed. If the host starts transaction to write another command to the ECCR register when the μPD98409 has not yet completed the previously received command, the μPD98409 makes STOP_B active to request a retry.

To read data from the EEPROM, an address and the READ command are set to the ECCR register. When the μPD98409 receives a command, it starts reading data via the EEPROM interface. While the μPD98409 reads the data, the B bit of the ERDR register is set to 1. When the μPD98409 has completed reading, it clears the B bit to 0, and stores the data to the “READ DATA” field. The host confirms that the B bit of the ERDR register is 0 after it has issued the READ command, and acquires the data.

To write data to the EEPROM, or to erase its data, the write/erase operation must be first enabled by using the EWEN command.

If no EEPROM is connected, accessing these registers is meaningless.

4.2.3 EEPROM format

The EEPROM stores the Subsystem vendor ID, Subsystem ID, and Min_Gnt, Max_Lat fields of the configuration register. The format of the EEPROM is as shown below.

Table 4-6. EEPROM Format

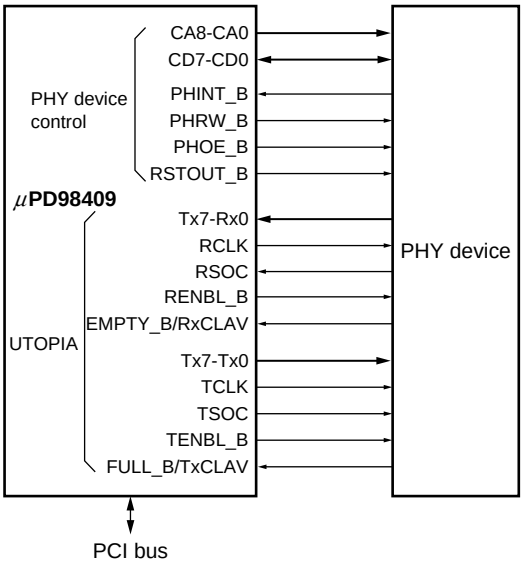
Address (H)	Contents
00	a5a5 (H) This code is necessary for checking to see if the μPD98409 is connected to EEPROM. If this code is not present, automatic loading is not performed even if EEPROM is physically connected.
01	Value to be stored to Subsystem Vendor ID area
02	Value to be stored to Subsystem ID area
03	High-order 8 bits: Value to be stored to Max Lat area, Low-order 8 bits: Value to be stored to Min Gnt area
04-3F	Not used. Can be used freely by applications.

Remark On power application, it takes the μPD98409 about 600 clocks to check whether EEPROM is connected. If the EEPROM is connected, about 2400 clocks is necessary until automatic loading is completed after power application. If the host makes an access during this period (including the configuration cycle), the μPD98409 asks the host to retry.

4.3 PHY Device Interface

This is an interface between the μPD98409 and a PHY device. To interface the PHY device, a UTOPIA interface that transfers cell data between the μPD98409 and the PHY device, and a PHY device control interface that controls the PHY device or obtains statuses, are used.

Figure 4-9. PHY Layer Interface for Data and Control



4.3.1 UTOPIA interface

The μPD98409 employs a UTOPIA interface conforming to the ATM Forum Recommendation to transfer cell data with a PHY device. This interface supports two modes: octet-level and cell-level. These modes are selected by using the UOC bit of the GMR register, as follows:

Selects UTOPIA interface mode		
UOC bit (GMR: bit 26)	0	Octet-level handshake
	1	Cell-level handshake
	Default = 0	

Caution Fix the PHYSEL1 pin of the μPD98409 to low.

The transmission and reception sides of the UTOPIA interface consist of a clock signal line through which the clock is supplied by the μPD98409, 8 bits of data signal lines, and three control signal lines.

- TCLK (output) : Transmit clock. The system clock input to the BUSCLK pin is output as is from this pin.
- TENBL_B (output) : Transmit enable signal. This signal indicates to the PHY device that data is output to Tx7 through Tx0 with the current clock signal.

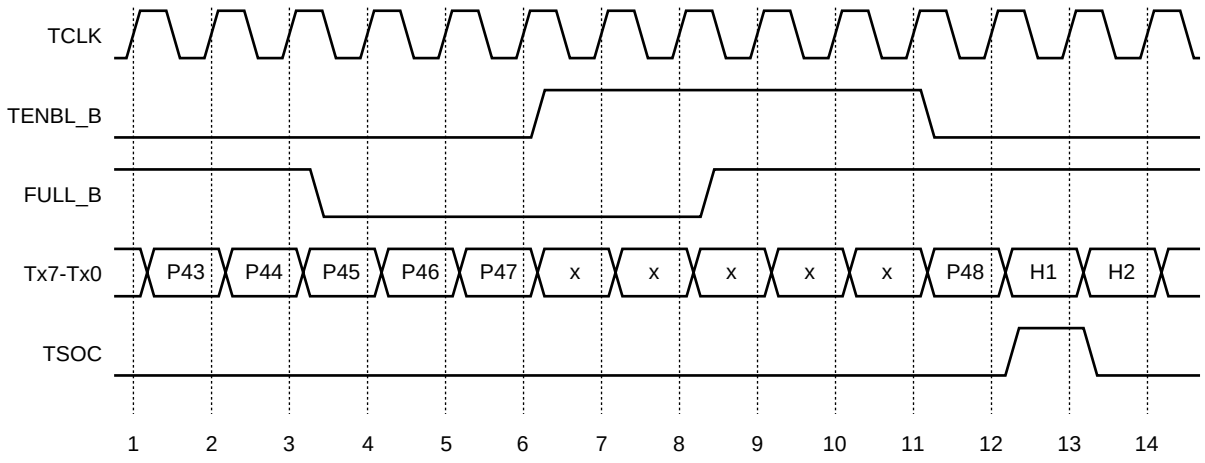
- FULL_B/TxCLAV (input) : This pin differs in meaning depending on whether the octet-level handshake or cell-level handshake mode is used. In the octet-level handshake mode, this pin functions as a FULL_B signal by which the PHY device reports to the μPD98409 that it can receive no more data because the buffer is full. In the cell-level handshake mode, this pin functions as a TxCLAV signal to report to the μPD98409 whether the PHY device can receive the next cell.
- Tx7 through Tx0 (output) : Transmit data bus.
- TSOC (output) : Transmit cell position start signal. Output in synchronization with the first byte of a cell.
- RCLK (output) : Receive clock. The system clock input to the BUSCLK pin is output as is from this pin.
- RENBL_B (output) : Receive enable signal by which the μPD98409 reports to the PHY device that it is ready to receive data at the next clock cycle.
- EMPTY_B/RxCLAV (input) : This pin differs in meaning depending on whether the octet-level handshake or cell-level handshake mode is used. In the octet-level handshake, it functions as an EMPTY_B signal that reports to the μPD98409 that the current data on Rx7 through Rx0 are invalid because there is no receive data to be supplied.
  In the cell-level handshake mode, it functions as an RxCLV_B signal that reports to the μPD98409 whether there is cell data to be supplied next.
- Rx7 through Rx0 (input) : Receive data bus.
- RSOC (input) : Receive cell start position signal. Input in synchronization with the first byte of the cell from the PHY device.

(1) Transmit interface

(a) Octet-level handshake

The μPD98409 outputs the system clock input to the BUSCLK pin as is from the TCLK pin as a clock to synchronize transmit cell data with the PHY device. The cell data in the transmit FIFO are sent to the 8-bit data line of Tx7 through Tx0 at the rising edge of TCLK. While valid data is being sent to Tx7 through Tx0, the enable signal TENBL_B is made low. In addition, the TSOC signal is made high in synchronization with the first byte of the cell header. When the PHY device has detected the low level of the TENBL_B signal, it reads data from Tx7 through Tx0, and learns the first start position of the cell by using the TSOC signal. When the internal FIFO of the PHY device has become full and therefore, the PHY device can receive no more transmit data, it makes the FULL_B signal low to report to the μPD98409. The μPD98409 samples the FULL_B signal at the rising edge of the TCLK clock. When it has detected the low level of the FULL_B signal, the μPD98409 makes the TENBL_B signal high at the rising edge two clock cycles after the detected edge, and stops transmitting valid cell data. When the FULL_B signal becomes inactive (high), the μPD98409 makes the TENBL_B signal low at the rising edge two clocks after the rising edge at which the high level of the TENBL_B signal has been detected, and resumes transmitting cell data.

Figure 4-10. Octet-Level Handshake Transmission Timing

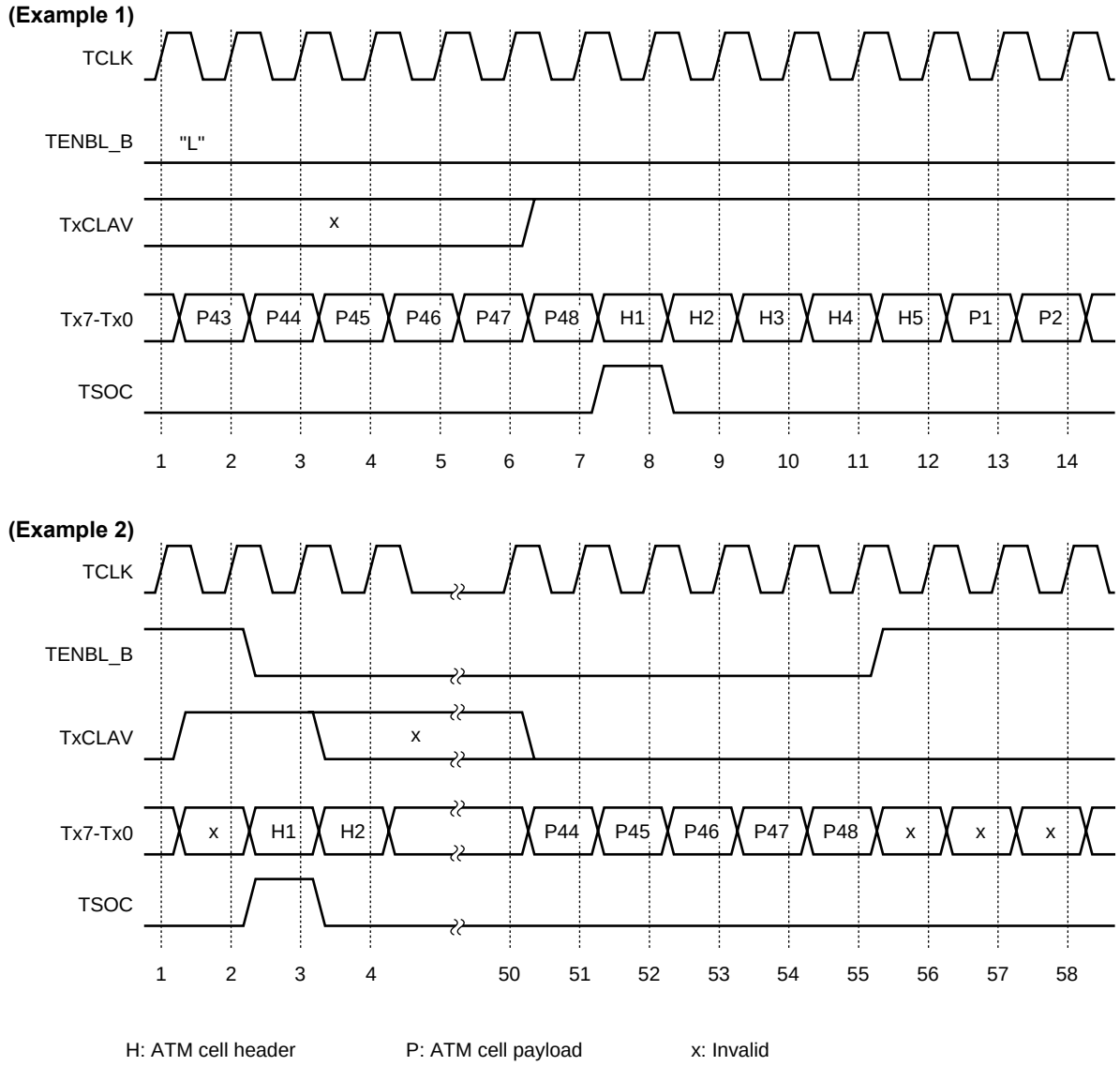


H: ATM cell header P: ATM cell payload x: Invalid

(b) Cell-level handshake

In the cell-level handshake mode, only the FULL_B signal in the octet-level handshake mode is replaced with the TxCLAV signal, and other signal timing is the same. The PHY device keeps the TxCLAV signal high until the current cell is completely transferred, if there is a vacancy in the transmit buffer to receive the next cell. If there is no vacancy in the buffer and therefore, the next cell cannot be received, the TxCLAV signal is made low at least four cycles before the end of transfer of the current cell.

Figure 4-11. Cell-Level Handshake Transmission Timing



H: ATM cell header P: ATM cell payload x: Invalid

(2) Receive interface

(a) Octet-level handshake

The μPD98409 supplies the system clock input to the BUSCLK pin as is to the PHY device from the RCLK pin as a receive cell data synchronization clock. The PHY device must output receive cell data to the data line of Rx7 through Rx0 in synchronization with this RCLK clock supplied from the μPD98409.

The μPD98409 latches the data at the rising edge of RCLK. It reports to the PHY device whether it can receive cell data, by using the RENBL_B signal. If the internal receive FIFO of the μPD98409, which has a capacity of 21 cells, has become full and therefore, the μPD98409 can receive no more data, the μPD98409 makes the RENBL_B signal high in a cycle one clock before to report to the PHY device. The PHY device must stop transmitting the receive cell data when it has detected the high level of RENBL_B. When there is a vacancy in the FIFO of the μPD98409 and therefore, the μPD98409 is ready for reception, it makes the RENBL_B signal low again in a cycle one clock before to report to the PHY device.

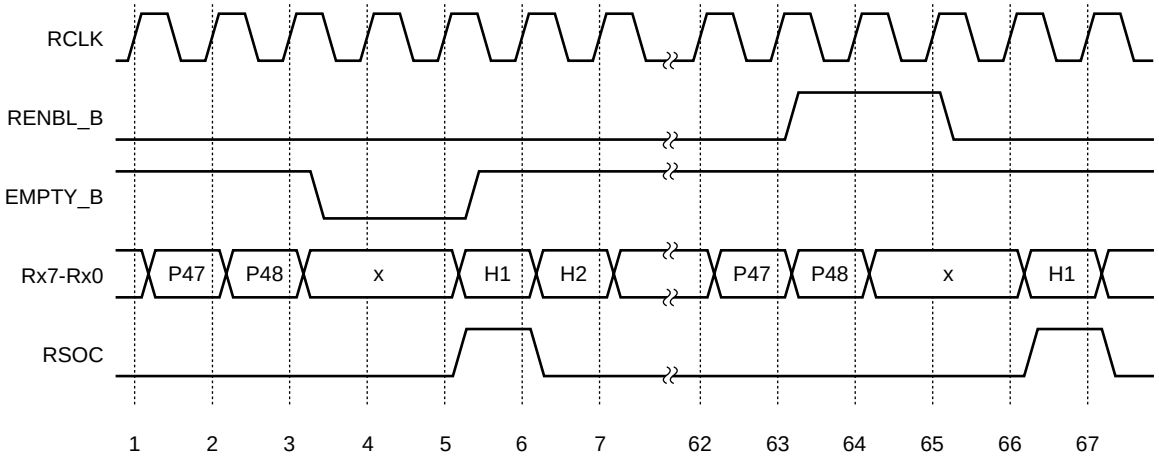
The operation of the RENBL_B signal of the μPD98409 differs depending on whether the DROP mode or No DROP mode is set. These modes can be selected by the DR bit of the GMR register, as follows.

DROP mode/No DROP mode (GMR register: DR bit)	
DROP mode (DR bit = 0)	The μPD98409 keeps the RENBL_B signal active (low) and does not request the PHY device to stop transmitting the receive cell data, even if the receive FIFO has become full. Therefore, there is a possibility that receive FIFO overrun will occur. The receive cell that is responsible for overrun is internally dropped. Do not use the DROP mode when receiving MPEG-TS packets. The operation is not guaranteed if the receive cell of a MPEG-TS packet is internally dropped.
No DROP mode (DR bit = 1)	When the receive FIFO becomes full and therefore, the μPD98409 can receive no more receive cell data, it makes the RENBL_B signal high in a cycle one clock before. When the PHY device detects the high level of the RENBL_B signal, it must stop transmitting receive cell data. When there is a vacancy in the receive FIFO of the μPD98409, the μPD98409 makes the RENBL_B signal low again.

The PHY device makes the EMPTY_B signal low and stops outputting the valid receive data when there is no more valid cell data to be output in the receive FIFO. The μPD98409 samples the EMTPY_B signal at the rising edge of the RCLK clock. Even when it detects that the EMTPY_B signal has gone low, the μPD98409 does not latch the data at the rising edge of the clock. The PHY device must input a high level to the RSOC pin in synchronization with the first byte (first byte of the header) of the receive cell header. The μPD98409 samples the RSOC signal at the rising edge of RCLK. When it detects the high level of the RSOC signal, the μPD98409 starts counting the valid data input at the same rising edge. When the μPD98409 has counted 53 bytes, it assumes that it has received one cell, and executes receive cell processing.

Caution Be sure to select the No DROP mode when receiving MPEG-TS packets. The operation is not guaranteed if a MPEG-TS packet is internally dropped by the μPD98409.
Make sure that the input levels to the Rx7 through Rx0 pins of the UTOPIA receive interface of the μPD98409 do not go into a high-impedance state.

Figure 4-12. Octet-Level Handshake Reception Timing

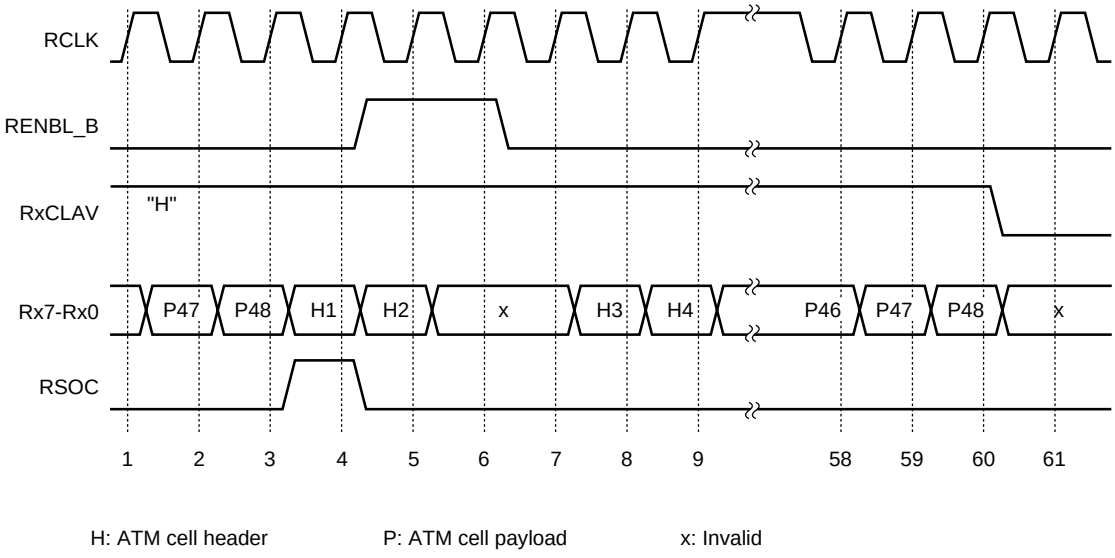


H: ATM cell header P: ATM cell payload x: Invalid

(b) Cell-level handshake

In the cell-level handshake mode, only the EMPTY_B signal in the octet-level handshake mode is replaced with the RxCLAV signal, and other signal timing is the same. The RxCLAV signal is used by the PHY device to indicate whether the device has cell data to be transmitted next. If the PHY device has no more cell data to be supplied, it makes the RxCLAV signal low in the cycle next to the last octet of the data. The μPD98409 does not receive data on Rx7 through Rx0 in the cycle in which the RxCLAV signal is low. The input timing of the RxCLAV signal is the same as that of the EMPTY_B signal in the octet-level handshake mode.

Figure 4-13. Cell-Level Handshake Reception Timing



H: ATM cell header P: ATM cell payload x: Invalid

4.3.2 PHY device control interface

The μPD98409 has an interface to control reading/writing the registers of the PHY device. When this function is used, no interface circuit has to be provided between the host and PHY device.

The control interface of the PHY device is a simple memory type slave interface.

The host uses the Indirect_Access command, one of the commands of the μPD98409, to access the PHY device. If the Indirect_Access command is issued to the PHY device, the μPD98409 starts a PHY control cycle to manipulate the following control signals.

For the details on the Indirect_Access command, refer to **CHAPTER 6 COMMANDS**.

- PHCE_B (output) : Chip enable signal to the PHY device. If the host issues the Indirect_Access command to the PHY device via the μPD98409, this pin outputs a low level.
- PHRW_B (output) : This pin indicates whether the PHY device is accessed for read or write. It outputs a high level if the device is accessed for read, and a low level if it is accessed for write.
- PHOE_B (output) : Output enable signal to make the output lines of the PHY device active.
- PHINT_B (input) : This pin inputs the interrupt request signal of the PHY device. When a low level is input to this pin, the μPD98409 sets the PHY interrupt (PI) bit in the GSR register and generates an interrupt to the host.
- RSTOUT_B (output) : This pin supplies a reset signal to the PHY device. When hardware or software reset is executed, the μPD98409 keeps this signal low for the duration of 18 to 19 clocks.

- CD7 through CD0 (input/output) : Eight data input/output signal lines.
- CA8 through CD0 (output) : Nine address signal lines.

When this interface is not used, leave the output pins open, and pull up the input pin and the output pin.

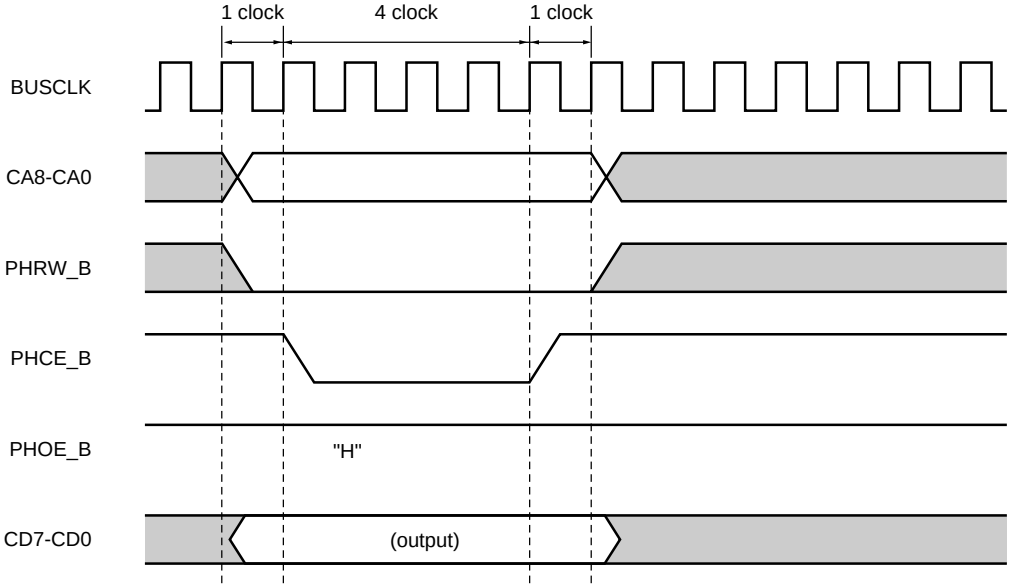
Remark PHINT_B can be used as a general-purpose port pin to detect a level via the μPD98409, in addition to as a pin to input an interrupt from the PHY device.

Caution The operation of the PHY device control interface of the μPD98409 is the same as that of the μPD98401, except for the following:

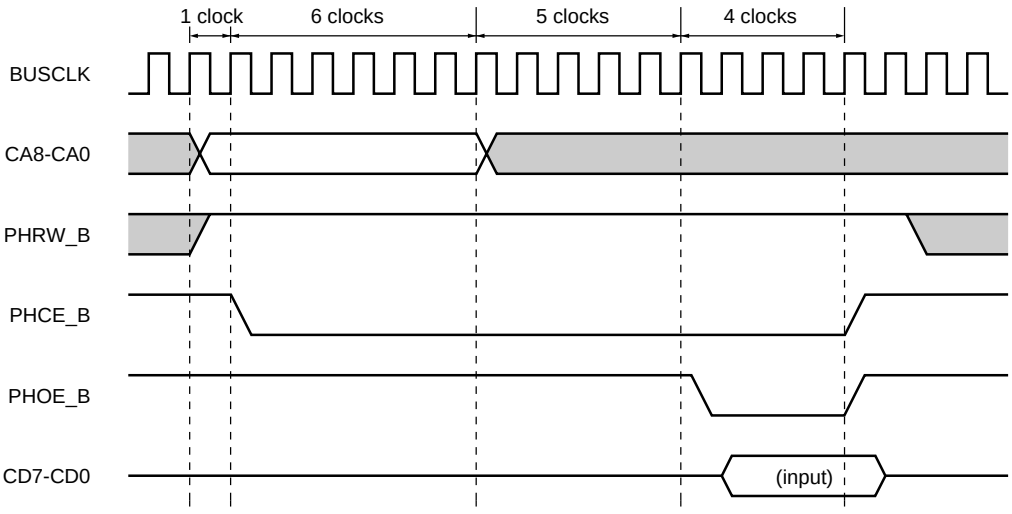
- **The CD7 through 0 pins go into a high-impedance state only when the PHOE_B pin outputs a low level.**

Figure 4-14. PHY Device Control Signal Timing

(a) Write operation



(b) Read operation



[MEMO]

CHAPTER 5 OPERATIONS OF μPD98409

This chapter explains the functions of the μPD98409.

5.1 Initialization

(1) Initializing the chip

So that the μPD98409 may function appropriately, the chip must be initialized to set the internal state machine and registers in specific status. Two types of reset can be executed: hardware reset that is executed by inputting a low level to the RST_B pin, and software reset that is executed when the host writes the SWR register. Software reset initializes all the registers of the μPD98409 except the configuration register. Hardware reset initializes all the internal state machines and registers of the μPD98409.

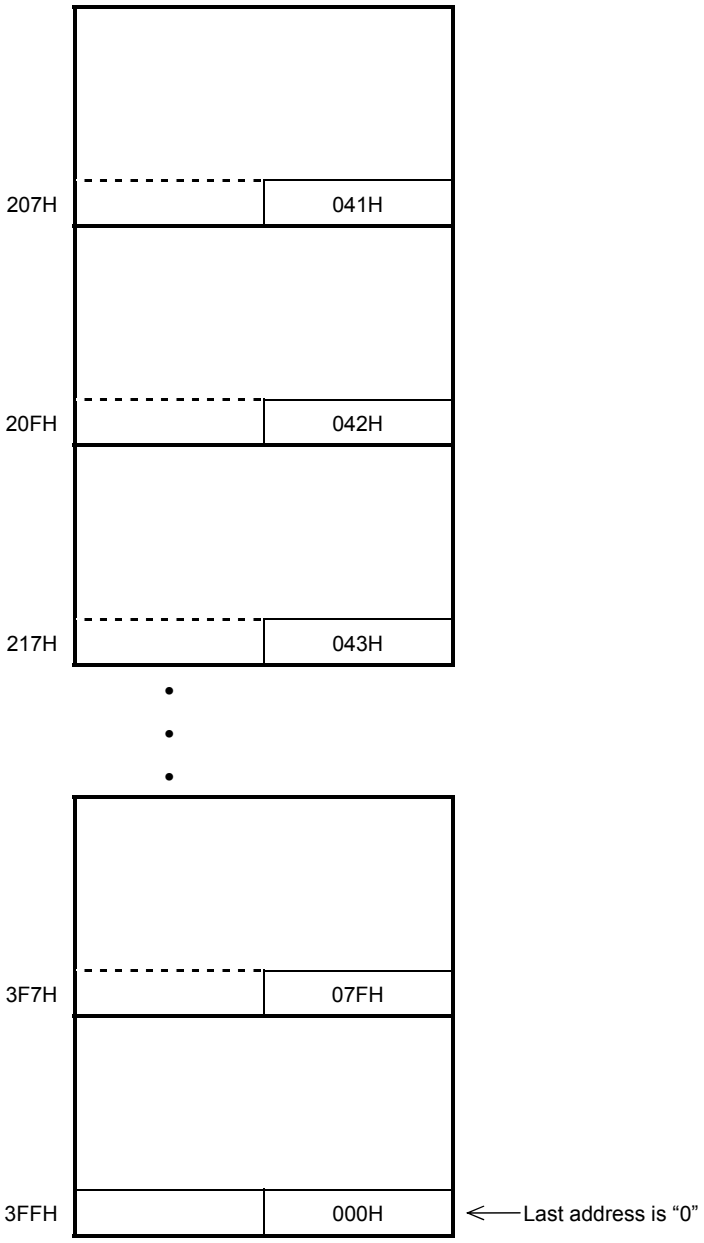
Caution The μPD98409 requires a time of 20 clock cycles to initialize the internal circuitry of the chip. Therefore, do not execute slave access to the μPD98409 within 20 clock cycle time after reset.

Remark On power application, it takes the μPD98409 about 600 clocks to check whether EEPROM is connected. If the EEPROM is connected, about 2400 clocks is necessary until automatic loading is completed after power application. If the host makes an access during this period (including the configuration write cycle), the μPD98409 asks the host to retry.

(2) Initializing control memory

The control memory incorporated in the μPD98409 is automatically initialized after hardware or software reset. The μPD98409 first clears all the areas of the control memory to 0 and then divides the free block pool area into blocks by writing an 8-word block number (pointer) to each block. The written block number functions as a pointer that indicates the first address of the next block, chaining the blocks into which the free block pool area has been divided. This chain information is used by the μPD98409 to manage the free block pool area of the control memory, and is rewritten to open/close VC or transmit/receive data. Only the free block pool area is divided into blocks and these blocks are linked, and the other fields are cleared to 0. The linking of each block is illustrated below.

Figure 5-1. Control Memory after Initialization (free block pool)



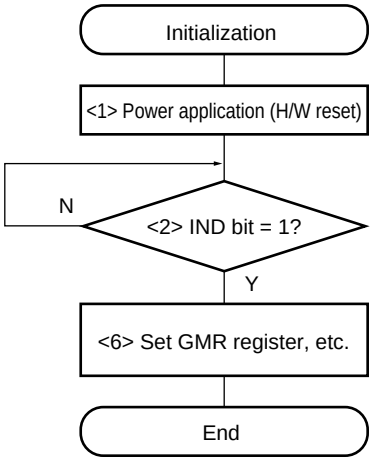
It takes about 1024 clocks to initialize the control memory after reset. During this period, access cannot be made by using the control memory interface (by using the Indirect_Access command), and only the direct address registers of a slave, apart from CMR and CER, can be accessed.

The μPD98409 sets the IND bit of the GSR register to 1 after initialization of the control memory has been completed, and an interrupt, if not masked, occurs.

(3) Example of initialization sequence

Figure 5-2 shows an example of the sequence to initialize the chip.

Figure 5-2. Initialization Sequence



<1> On power application, execute hardware reset to initialize the μPD98409. Do not access the μPD98409 as a slave for the duration of 20 clocks after reset.

<2> Wait until the IND bit of the GSR register is set (completion of control memory initialization) (It takes about 1024-clock cycles). To detect that the IND bit has been set to 1, use polling or an interrupt. To use an interrupt, the interrupt must be unmasked by using the interrupt mask register (IMR).

<3> If the IND bit is set to 1, the Indirect_Access command can be used. The host sets the contents of registers, GMR, TOS, SMA, PMA, and so on.

5.2 Setting of Control Memory

The size of the internal control memory of the μPD98409 is 610 words (1 word = 32 bits wide), and up to 64 VC can be supported. The control memory is divided into the following four areas (“1 block” in the explanation below means 8 words).

(1) Receive lookup table:

This is an area for storing the patterns of received VPI/VCI (up to 15 bits wide) and an enable bit. It has a size of 4 blocks (32 words) and can register up to 64 received VC. For details, refer to **5.5.4 Setting of receive look-up table**.

(2) Receive free buffer pool pointer:

This area stores “pool descriptors”. It has a size of 8 blocks (64 words), and up to 32 pool descriptors can be set. For details, refer to **5.5.2 (2) Receive free buffer pool pointer**.

(3) Transmit shaper pointer:

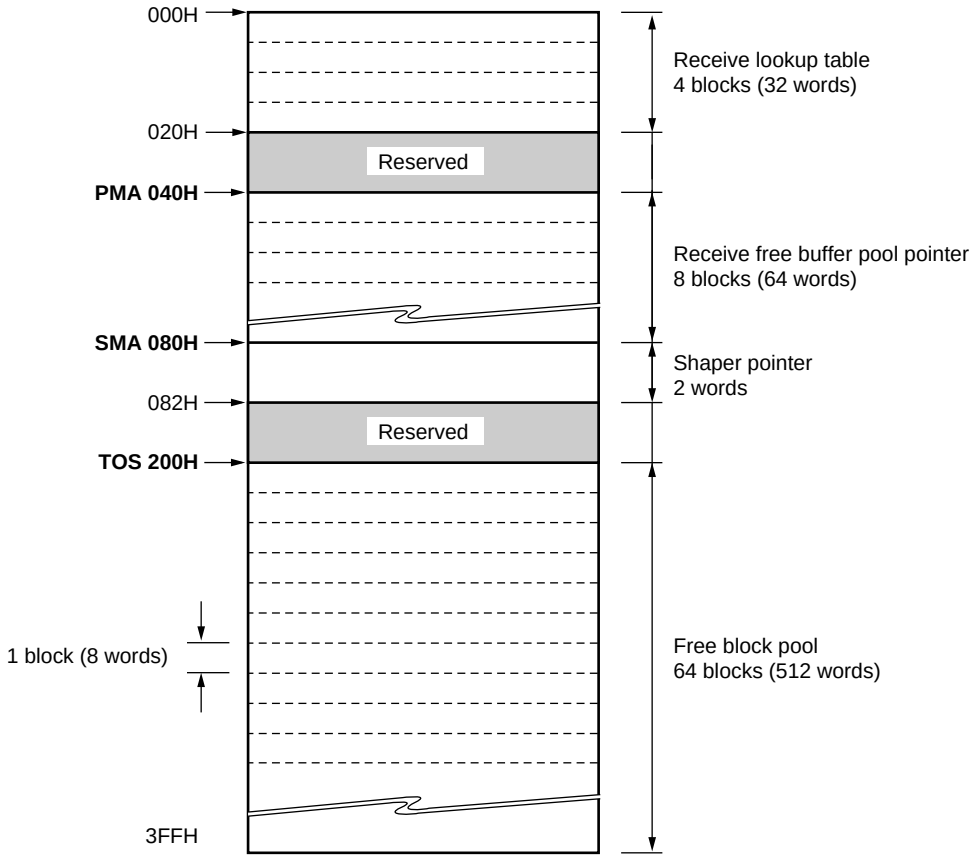
This is an area used by the μPD98409 for transmission processing. It has a size of 2 words, and two shaper pointer entries are stored. For details, refer to **5.4.4 (5) Shaper link list**.

(4) Free block pool:

This area stores transmit/receive VC tables. Its size is 64 blocks (512 words) and this area can accommodate up to 64 VC tables with transmit and receive tables combined.

For details, refer to **5.4.3 (2) Setting of transmit VC table** and **5.5.3 (2) Setting of receive VC table** in the description on the transmit/receive functions.

Figure 5-3. Configuration of Control Memory

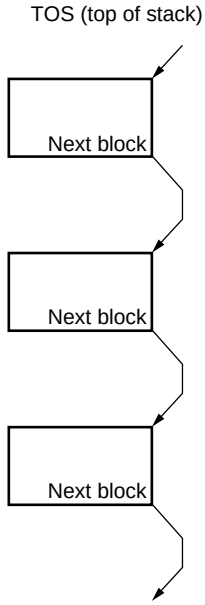


The host must write the start address (default value) of each area to the indirect address register of the μPD98409 on initialization, to divide the areas of the control memory. The start address is written to the following three registers:

- PMA register (set value : 040H) : Sets the start address of the receive free buffer pool descriptor area.
- SMA register (set value : 080H) : Sets the start address of the shaper pointer.
- TOS register (set value : 200H) : Sets the start address of the free block pool

The initial value (200H) of TOS (top of stack) is set by the user. After that, the μPD98409 updates TOS as a stack pointer that indicates the beginning of the free block pool. The μPD98409 chains the blocks of the free block pool on starting. (refer to **5.1 (2) Initializing control memory**). Each time the host has issued the Open_Channel command, the μPD98409 returns the address indicated by TOS from the free block pool by using command indication, and allocates blocks. At this time, the μPD98409 updates the contents of the TOS register to the pointer of the next block. Each time the host has issued the Close_Channel command, the μPD98409 returns a block to the free block pool and allocates it to TOS.

Figure 5-4. Stack of Free Block Pool



Caution Be sure to initialize the PMA, SMA, and TOS registers with their default values. The operation is not guaranteed if a value other than the default value is written to these registers.

Remark The μPD98409 accesses the PMA, SMA, and TOS registers related to the control memory more than once during transmission/reception. The host must write appropriate values to these registers only on initialization after reset, and must not change their contents after the transmission/reception operation has been started. If the register contents are changed during transmission/reception operation, malfunctioning may occur. The registers can be read at any time, however.

5.3 Setting of Mailbox

5.3.1 Setting of mailbox

A mailbox is an area prepared by the host in the system memory to store the indications issued by the μPD98409. An indication is status information the μPD98409 issues for each transmit/receive packet. For the details of the contents of each transmit/receive indication, refer to **5.6 Transmit/Receive Indication**.

The μPD98409 can support two mailboxes each for transmission and reception, totaling four mailboxes. These four mailboxes are assigned numbers 0 to 3 and are classified by number into those for transmission and those for reception (it is not necessary to use all the four mailboxes).

- Receive mailboxes : Nos. 0 and 1
- Transmit mailboxes: Nos. 2 and 3

The μPD98409 uses the mailboxes as ring buffers in the system memory. These buffers are defined by the following addresses.

- Mailbox start address high (MSH) : High-order 16 bits of the first address of the mailbox
- Mailbox start address low (MSL) : Low-order 16 bits of the first address of the mailbox
- Mailbox bottom address (MBA) : Low-order 16 bits of the address after the last address of the mailbox
- Mailbox write address (MWA) : Low-order 16 bits of the write pointer managed by the μPD98409
- Mailbox tail address (MTA) : Low-order 16 bits of the address updated up to where the host has read

All the pointers of a mailbox are 16 bits long. The μPD98409 concatenates a pointer with the 16 bits of the mailbox start address high to create an actual 32-bit address. The maximum size of one mailbox is 64K bytes (for one indication). All the pointers of a mailbox are allocated to the direct address register of the μPD98409. The host sets the initial value of each pointer of the mailbox to the register. The initial values of MWA and MTA are set in the same manner as MSL.

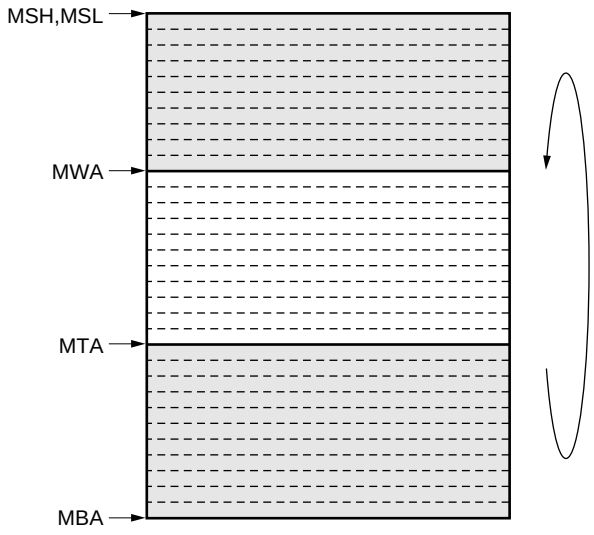
Cautions 1. One receive indication consists of 4 words. Therefore, the size of a receive mailbox must be an integer multiple of 4 words.

2. Do not set MSL and MBA to the same value. If MSL is set to "0000H" and the mailbox to be used has the maximum size of 64K bytes, MBA is "0000H", the same as MSL, because it is the address after the last address. To use the maximum area, set MBA to an address one indication less than 64K bytes.
Transmit: FFFCH, Receive: FFF0H

3. Of the registers related to the mailboxes, the host updates and manages only MTA. Write MSH, MSL, MWA, and MBA only during initialization, and do not change them while the μPD98409 is executing a transmission/reception operation. The μPD98409 accesses and manages these registers more than once during the transmission/reception operation. If the register contents are changed in the middle, malfunctioning may take place.

Remark Since this is raw cell reception, the μPD98409 does not use a mailbox.

Figure 5-5. Structure of Mailbox



5.3.2 Operation of mailbox

The μPD98409 increments the write pointer (MWA) when it has written an indication. Each time the μPD98409 has written an indication, it sets the MM bit of the GSR register corresponding to the mailbox and issues an interrupt if not masked.

If the bottom address (MBA) is reached when the μPD98409 updates the write pointer (MWA), it jumps MWA to the start address (MSL).

The read pointer (MTA) is used to prevent the μPD98409 from overwriting an indication that has not yet been read by the host. The read pointer (MTA) of each mailbox is managed and updated by the host. Each time the host has read an indication from a mailbox, it writes the address next to the address of the indication that has been read to the read pointer (MTA). MTA is changed only by the host and is read by the μPD98409.

The μPD98409 checks the pointer before writing an indication. If the write pointer (MTA) indicates an address equal to that of the read pointer (MTA), the μPD98409 assumes that the mailbox is full (MF) and sets the MF bit of the GSR register corresponding to the mailbox and issues an interrupt if not masked. The μPD98409 does not issue the next indication to the mailbox in the MF status (MWA = MTA). The host, therefore, must read indication from a mailbox in the MF status and quickly update MTA.

Caution If the μPD98409 waits for DMA transfer of indication because a mailbox is in the MF status, all DMA operations are stopped until transfer of the indication has been completed. If the MF status takes place frequently, therefore, the transmission/reception operation is affected.

Table 5-1. Mailbox Operation of μPD98409

(a)	Updates MWA to current MWA + number of bytes of indication if indication is issued.
(b)	Changes MWA to MSL if MWA = MBA.
(c)	Sets MF bit of GSR register if MWA = MTA.
(Number of bytes of indication: Receive = 16 bytes (4 words), Transmit = 4 bytes (1 word))	

5.4 Transmission Function

5.4.1 Transmission processing flow

The μPD98409 transmits a cell in several steps.

Figure 5-6. Outline of Transmission Flow

	Host processing
(1)	Sets scheduler register
(2)	Sets transmit data
(3)	Issues Open_Channel command
(4)	Sets VC table
(5)	Issues Tx_Ready command
⋮	
(13)	Reads transmit indication
(14)	Updates mailbox pointer (MTA)

	Operation of μPD98409
(6)	Makes shaper active. Links VC with shaper
(7)	Fetches packet descriptor
(8)	Scheduling
(9)	Fetches data from system memory in segment units of VC
(10)	Creates cell (inserts AAL-5 trailer and CRC-10 as necessary)
(11)	Transmits cell to PHY device
(12)	Stores transmit indication and issues interrupt

The μPD98409 can support up to 32K channels in any combination of transmit VC and receive VC. (1) The host sets the scheduler register to the indirect address register of the μPD98409 to determine the transmission rate before starting transmission. (2) The transmit data is stored in the system memory and is managed by the host by using a transmit queue that is set for each transmit channel. The transmit queue stores the descriptor of each packet. The host issues Open_Channel command (3) for each connection to be set to open a channel (VC). The μPD98409 allocates a block to be used as a VC table from the free block pool of the control memory, and returns its address to the host. The host assigns an initial value to the VC table (4). This completes the preparation for starting transmission.

The host issues the Tx_Ready command (5) to allow the μPD98409 to read the transmit queue. The μPD98409 makes the shaper, which is the link destination of the VC, active when it has received the command, and adds the VC to the shaper link list (6). It then reads the descriptor (7) and wirtes it to the VC table. After that, the μPD98409 fetches data in segment units (9) in accordance with scheduling (8), internally creates a cell (10), and transmits the cell to the PHY device (11). The processing (8) through (11) is repeated until transmission of one packet has been completed.

The μPD98409 writes transmit indication (12) to the mailbox as status information, and issues an interrupt. The host reads the mailbox of transmit indication (13) and updates the read pointer of the mailbox in the μPD98409 (14).

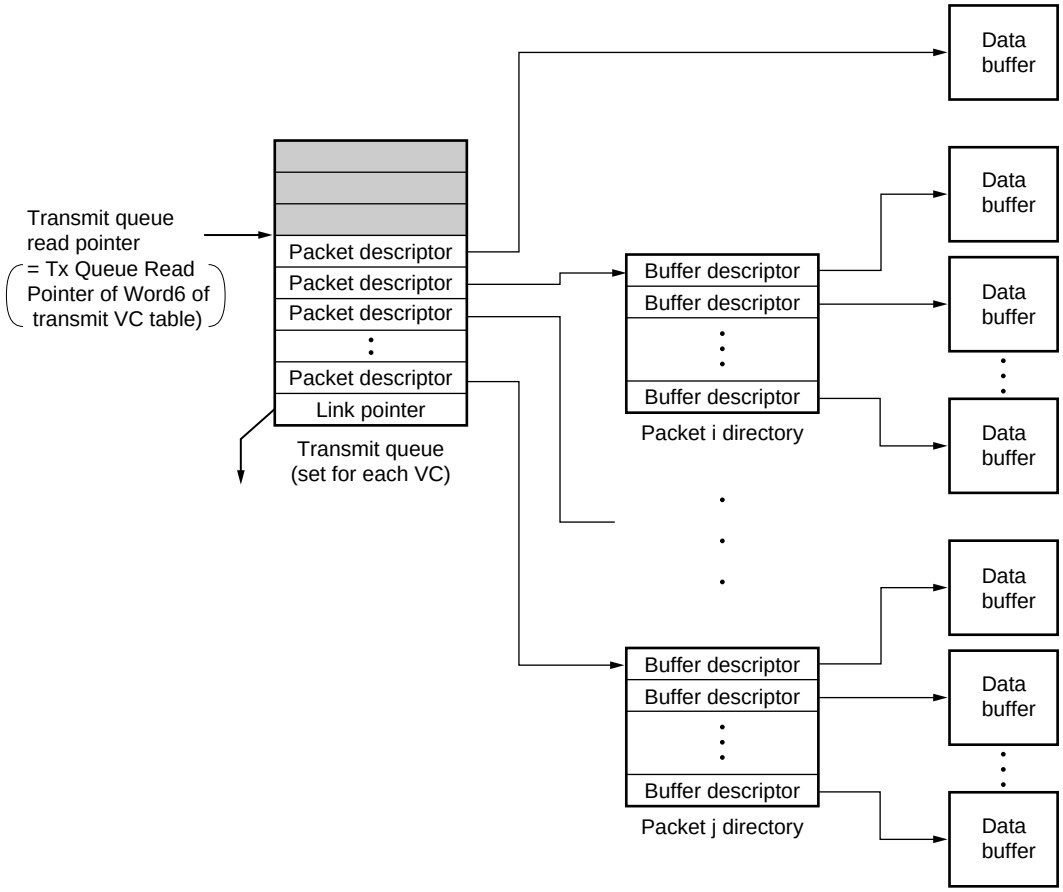
The transmission processing flow in Figure 5-6 is shown only for the purpose of illustration.

5.4.2 Structure of transmit data

Transmit data is stored in the system memory managed by the host. The transmit data set in the system memory consists of the following three elements for each VC. Figure 5-7 shows the structure of transmit data.

(1) Transmit queue
(2) Packet descriptor
(3) Data buffer

Figure 5-7. Example of Structure of Transmit Data in System Memory



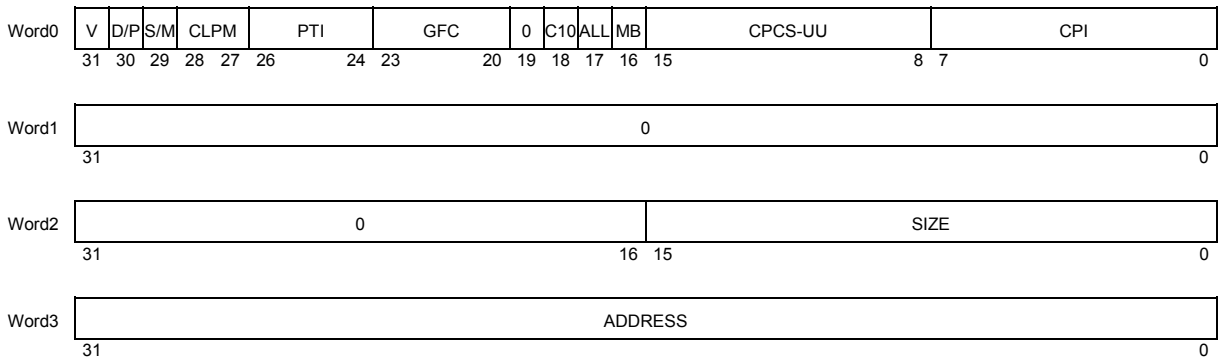
(1) Transmit queue

The host creates and manages a transmit queue for each channel (transmit VC). The transmit queue consists of the packet descriptor of the corresponding channel. A packet has one or more data buffer. The read pointer of the transmit queue is initialized to the corresponding transmit VC table of the control memory by the host. After that, the read pointer is corrected and managed by the μPD98409 each time the transmit packet descriptor has been fetched.

The initial value of the read pointer (Tx Queue Read Pointer) of the packet descriptor is set to the transmit VC table by the host. After that, the value of the read pointer is updated each time the μPD98409 has transmitted a packet. The μPD98409 reads the packet descriptor in the system memory in accordance with this transmission. If the host suddenly changes the transmit queue area in the system memory, the μPD98409 cannot correctly obtain information from the packet descriptor, and may malfunction. To release or change the transmit queue area, close the VC by using the Close_Channel command.

Figure 5-8 shows the format of the packet descriptor.

Figure 5-8. Format of Packet Descriptor



A “packet descriptor” consists of 4 words and must be located at a 32-bit boundary of the system memory. Set one packet descriptor for each transmit packet. The packet descriptor can store a pointer of the data buffer or a pointer of the packet directory.

The meaning of each field of the packet descriptor is described below.

<1> V, D/P, S/M bits

Table 5-2. Function of Packet Descriptor

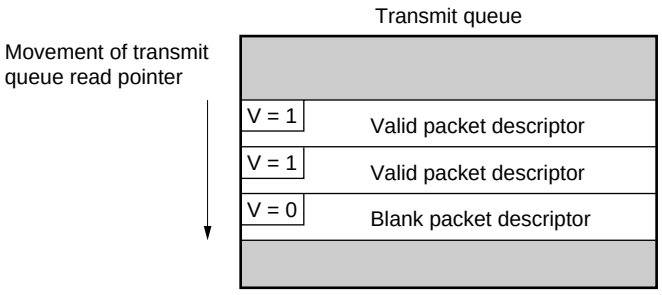
V	D/P	S/M	Function
0	–	–	Blank packet descriptor
1	0	–	Link pointer
1	1	0	Packet descriptor in multi-buffer mode
1	1	1	Packet descriptor in single-buffer mode

The V bit is set to “1” if the packet descriptor has data to be transmitted, or if the packet descriptor functions as a link pointer, indicating “valid packet descriptor”. If the V bit is set to 0, the packet descriptor is a “blank packet descriptor”.

The blank packet descriptor is necessary for the μPD98409 to recognize the last valid packet descriptor in a chain. Therefore, a blank packet descriptor must be located at the end of one or successive valid packet descriptors. If the V bit is “0”, indicating that the packet descriptor is “blank”, the areas other than the V bit of that packet descriptor are meaningless, and the μPD98409 ignores these areas. Because the μPD98409 reads the packet descriptor in 4-word units, however, the blank packet descriptor must also consist of a 4-word area.

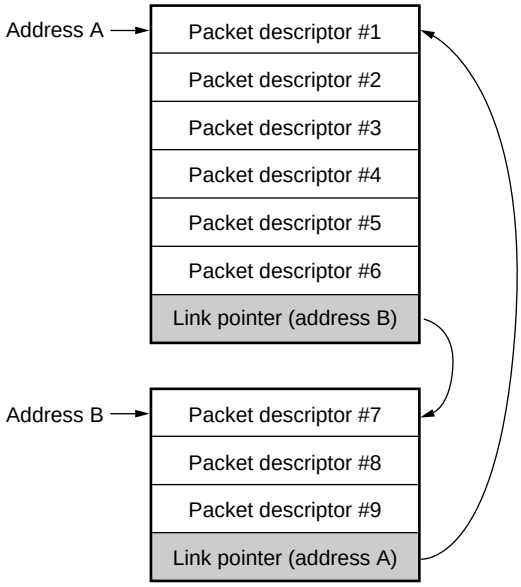
Caution The host must set the V bit to 1 after it has set all the areas other than the V bit of the packet descriptor. If Word1 through Word3 are set after the V bit of Word0 has been set to 1, the μPD98409 completes the processing of the preceding packet descriptor before the host has set all the valid packet descriptors, causing malfunctioning.

Figure 5-9. Location of Blank Descriptor



The packet descriptor can function as a “link pointer” if the D/P bit is so set. By using this, the user can handle the transmit queue as if it were arranged as a line or ring. When the packet descriptor is set as a link pointer, the “ADDRESS” field stores the first address of the next valid packet descriptor. When a packet descriptor is used as a link pointer, the fields of the packet descriptor other than the V and D/P bits and ADDRESS field have no meaning, and the μPD98409 ignores these fields.

Figure 5-10. Example of Arrangement of Transmit Queue

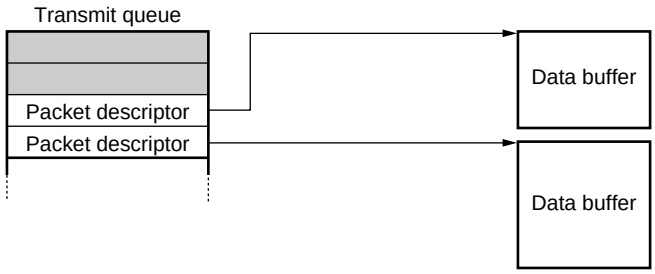


The S/M bit of the packet descriptor indicates whether this transmit packet is constructed in the single-buffer mode or multi-buffer mode.

- **In single-buffer mode**

  In the single-buffer mode, one transmit packet consists of one data buffer. In this case, the “ADDRESS” field of the packet descriptor stores the start address of the data buffer.

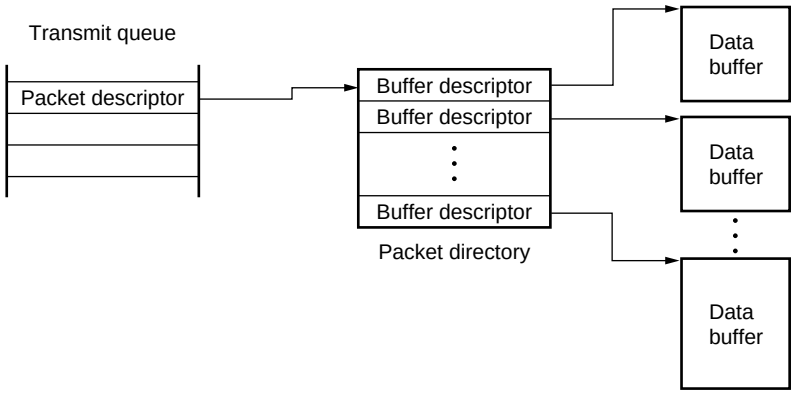
Figure 5-11. Single-Buffer Mode



- **In multi-buffer mode**

  In the multi-buffer mode, one transmit packet consists of two or more data buffers scattered in the system memory. In this case, the data buffers are bundled with a “packet directory”. The “ADDRESS” field of the packet descriptor stores the start address of the packet directory.

Figure 5-12. Multi-Buffer Mode



Packet descriptors in the single-buffer mode and packet descriptors in the multi-buffer modes can exist in one transmit queue together.

<2> CLPM

This field selects the transmission mode indicated by the “CLP” bit in the cell headers of the cells in which a packet is transmitted. The μPD98409 changes and transmits the CLP bit in accordance with the setting of this field.

Bit	Mode
00	Clears the CLP bit of all the cells in which this packet is to be transmitted to “0”.
11	Sets the CLP bit of all the cells in which this packet is to be transmitted to “1”.
01	Sets the CLP bit other than that of the last cell of this packet to “1”, and clears the CLP bit of only the last cell to “0”.
10	Setting prohibited

Do not assign code “10” to the CLPM field. When transmitting a raw cell or OAM cell, either “00” or “11” is valid. “01” must not be used.

<3> PTI

The μPD98409 inserts the 3 bits assigned by the user to this field to the "PTI" field in each cell header when it disassembles a transmit packet into cells for transmission.

The μPD98409 recognizes the pattern assigned to this field when it performs transmission processing. The transmission operation differs depending on whether the packet is the code (100 or 101) of an OAM F5 cell.

- **If pattern of OAM F5 cell (100 or 101) is set:**

  The μPD98409 unconditionally transmits the packet as an OAM cell. The transmission processing of the OAM cell differs from the transmission operation of an AAL-5 or raw cell. Refer to **5.4.6 Support of non-AAL-5 traffic**.

  If the pattern of an OAM F5 cell is set, be sure to select a raw cell by clearing the AAL bit to "0".

  To use a packet descriptor to transmit an OAM F5 cell, one OAM F5 cell must be set for one packet descriptor. The buffer specified by the packet descriptor for transmitting an OAM F5 cell must be always in the single-buffer mode, and must not be in the multi-buffer mode.

- **If pattern other than OAM F5 cell is set:**

  Whether the packet is processed as a transmit packet of AAL-5 or as a raw cell depends on the setting of the AAL bit.

  - **AAL-5 type (AAL bit = 1):**

    The μPD98409 inserts the code set in this field as is to the cells other than the last cell. The last cell of a packet is transmitted with the μPD98409 changing the low-order 1 bit to "1".
  - **AAL type other than AAL-5 (AAL bit = 0):**

    The μPD98409 inserts the contents of this field as is to all the headers of the transmit packet. The μPD98409 transmits the PTI field of the last cell without changing the low-order 1 bit to "1".

<4> GFC

This field stores the GFC pattern of all the cells of this transmit packet. The μPD98409 inserts the contents set by the user as is to all the GFC fields for transmission.

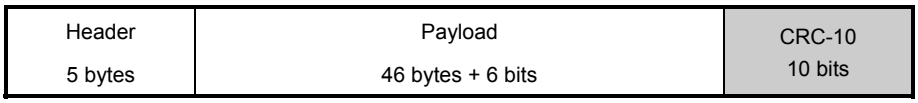
<5> C10

This bit specifies whether a CRC-10 error detection code is inserted.

C10	1	The μPD98409 performs 10-bit CRC operation and inserts an error detection code to each cell.
	0	The μPD98409 does not insert CRC-10.

If this bit is set to "1", the μPD98409 performs a CRC operation on fields except the last 10 bits of the payload for each cell, and inserts the result to the end of the cell as a 10-bit CRC error detection code.

Figure 5-13. Insertion of CRC-10



The generation polynomial of CRC-10 is as follows:

$$G(X) = 1 + X + X^4 + X^5 + X^9 + X^{10}$$

When using the CRC-10 error detection function of μPD98409, the following points must be noted.

<Cautions>

- **The μPD98409 reads every 48 bytes which are the data of 1 cell, from the data buffer. However, the last 10 bits of the 48-byte data is overwritten with a CRC-10 error detection code. Therefore, processing of the data buffer such as inserting dummy data in the portion to which the CRC code is to be inserted is necessary, taking the overwriting into consideration.**
- **If the data of one packet cannot be divided into units of 48 bytes, the μPD98409 cannot execute CRC-10 operation on cell data of less than 48 bytes. Therefore, dummy data (data of all 0) must be inserted to the transmit packet to which CRC-10 is to be inserted, to make the size a multiple of 48 bytes.**
- **Be sure to set the C10 bit to 1 to enable the CRC-10 insertion function when the AAL bit is 0.**

<6> AAL

This bit specifies whether a transmit packet is transmitted as a packet of AAL-5 type or other type.

AAL	1	The μPD98409 processes this packet as AAL-5-PDU.
	0	The μPD98409 processes this packet as a raw cell.

<7> MB

This bit selects mailbox 2 or 3 to store the transmit indication of this packet.

MB	1	Selects mailbox 3.
	0	Selects mailbox 2.

For the explanation on the mailbox, refer to **5.3 Setting of Mailbox**.

<8> CPCS-UU and CPI

The μPD98409 inserts the pattern set by the user in these fields to the CPCS-UU and CPI fields in the AAL-5 trailer. These fields are valid only when a packet of AAL-5 is transmitted (AAL bit = 1); otherwise, the μPD98409 ignores these fields.

CPCS-UU : CPCS user-user indication
CPI : Common part identifier

<9> SIZE

In single-buffer mode : Stores the size of the buffer in byte units.
In multi-buffer mode : This field is meaningless, and the μPD98409 ignores this field.

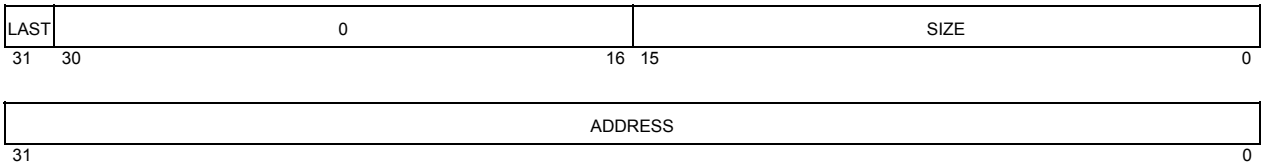
<10> ADDRESS

In single-buffer mode	Stores the start address of the data buffer. Because the data buffer can be allocated from a byte boundary in the system memory, this field stores a byte address 32 bits wide.
In multi-buffer mode	Stores the start address of the first buffer descriptor of a packet directory. The buffer descriptor must be located at a 32-bit boundary in the system memory. Therefore, the low-order 2 bits of the address stored to this field must be “00”.
In the case of link pointer (D/P = 0)	Stores the start address of the first packet descriptor of the next transmit queue. Because the packet descriptor must be located at a 32-bit boundary in the system memory, the low-order 2 bits of the address stored to this field must be “00”.

(3) Packet directory

A packet directory is set when a transmit packet is configured in the multi-buffer mode, and must be always located at a 32-bit boundary in the system memory. It does not have to be set in the single-buffer mode. A packet directory consists of successive buffer descriptors configured in the 2-word format shown in Figure 5-14. The buffer descriptors are pointers indicating the first addresses of data buffers.

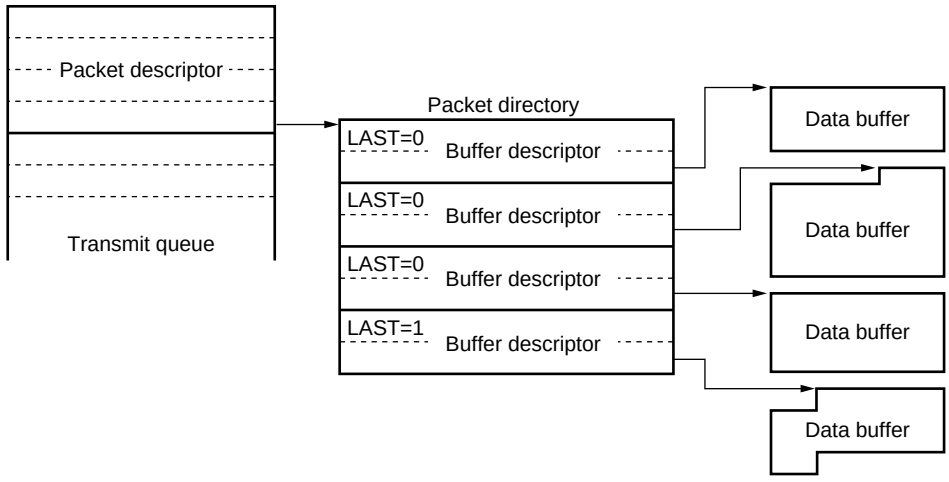
Figure 5-14. Format of Packet Descriptor



<1> LAST

This bit is set to “1” if the packet descriptor indicates the last data buffer of the transmit packet.

Figure 5-15. Setting of LAST Bit



<2> SIZE

This field sets the size of the data buffer in byte units.

<3> ADDRESS

This field sets the start address of the data buffer. Because the data buffer can be located from a byte boundary, the address of this field is a byte address 32 bits wide.

Remark Packet descriptors for AAL-5, OAM F5 transmission, and raw cell transmission can exist in the transmit queue of one VC together.

<4> Data buffer

The data buffer stores transmit data that is actually transferred. The transmit data can be located at a byte boundary location in the system memory and its size can be set up to 64K bytes. If the data is of AAL-5 type, the transmit data is equivalent to the “user data” field of CPCS-PDU.

5.4.3 Transmit channel (transmit VC)

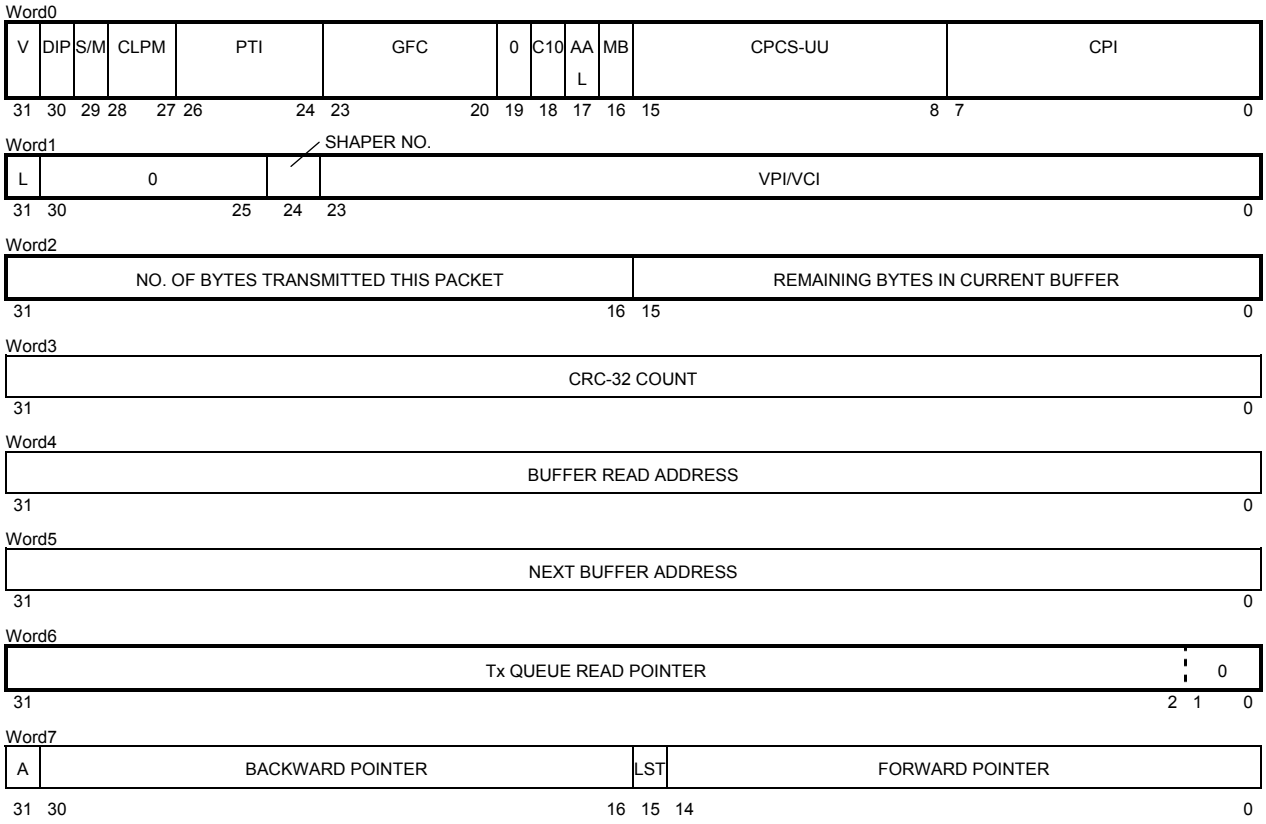
(1) Opening transmit channel

When the host issues the Open_Channel command, the μPD98409 allocates the block indicated by TOS (top of stack) from the free block pool of the control memory, and returns its first address to the host by using command indication. The host sets the allocated block as a transmit VC table.

(2) Setting of transmit VC table

The host assigns an 8-word block allocated from the free block pool of the control memory for each VC as a transmit VC table. The structure of the transmit VC table is as shown in Figure 5-16. The host assigns initial values to the locations indicated by the thick solid line in this figure. The host writes data to the VC table by using the Indirect_Access command. The areas not indicated by the thick solid line are used by the μPD98409 for transmission. The host can read the table at any time and use it as status information.

Figure 5-16. Structure of Transmit VC Table



Word0 stores the first word of the packet descriptor the μPD98409 fetches from the system memory, and its contents are as set by the user.

Word0	:	Stores the contents of Word0 of the packet descriptor which the μPD98409 has read from the system memory as is. The initial value must be cleared to all 0.
L	:	This bit is used by the μPD98409 as a flag. Be sure to set “1” to this bit as the initial value.
SHAPER NO.	:	Sets the shaper number (0 or 1) with which this VC is to be linked.
VPI/VCI	:	VPI/VCI field of cell header.
NO. OF BYTES TRANSMITTED THIS PACKET	:	The number of bytes transmitted by this packet so far. Be sure to set this field to “0” as the initial value.
REMAINING BYTES IN CURRENT BUFFER	:	The number of bytes in the current buffer that have not been transmitted yet. Be sure to set this field to “0” as the initial value.
CRC-32 COUNT	:	Indicates the result of the CRC-32 operation of this packet so far. The final operation result is inserted in the CRC-32 field of the AAL-5 trailer.
BUFFER READ ADDRESS	:	Pointer indicating the byte to be transferred next to the current buffer
NEXT BUFFER ADDRESS	:	Pointer indicating the next buffer of the current packet directory. In the single-buffer mode, 0 is set.
Tx QUEUE READ POINTER	:	Pointer indicating the first address of the next packet descriptor. The initial value is set by the host. The packet descriptor must be always located from a word boundary, and the low-order 2 bits of this pointer must be cleared to “00”. This pointer is updated each time the μPD98409 has transmitted a packet. Do not change the value of this pointer except when assigning the initial value to it.

A : Active bit indicating whether this VC table is in the active or idle status.
1: Active status
0: Idle status

BACKWARD POINTER : Pointer indicating the preceding VC on the shaper link list.

LST : "1" indicates that this VC is the last VC on the shaper link list.

FORWARD POINTER : Pointer indicating the next VC on the shaper list.

Remark Do not change Word7. Word7 of the transmit VC table stores pointers and flags necessary for the μPD98409 to manage the VC table. If this word is changed by an external host, the μPD98409 cannot correctly manage the VC table.

(3) Status transition of transmit channel

A transmit channel may be in non-existent, idle, or active status. The host opens a channel by issuing the Open_Channel command. When the μPD98409 has received this command, it reports the address of the transmit VC table from the free block pool of the control memory to the host as command indication. Next, the host writes appropriate parameters (such as VPI/VCI, shaper used, and transmit queue read pointer) to a block allocated by using the Indirect_Access command. This makes this block a transmit VC table, and the channel enters the idle status.

When the channel is in the idle status (after the parameters have been set), the VC is ready for transmission. If the data to be transmitted is created and a valid packet descriptor has been prepared in this status, the host issues the Tx_Ready command to the μPD98409. When the μPD98409 receives this command, it sets the "A" bit of Word7 of the VC table to "1", and adds that VC table to the link list of the shaper. As a result, the channel enters the active status.

The VC in the active status takes turns to transmit a cell at the rate determined by the scheduler (refer to **5.4.4 Traffic control**). When this VC first transmits, the μPD98409 reads the packet descriptor in the system memory and updates Word0 of the VC table in accordance with the transmit queue read pointer (Tx Queue Read Pointer: Word6 of VC table). At this time, if the packet descriptor is a blank packet descriptor (V = 0), the μPD98409 clears the "A" bit of the VC table to "0" to returns the VC to the idle status. If the packet descriptor is valid (V = 1), the μPD98409 starts transmitting a cell. The μPD98409 also add 4 words so that the transmit queue read pointer of the transmit VC table indicates the next packet descriptor.

When the μPD98409 completes reading all the data of the packet, it fetches the next packet descriptor. If this packet descriptor is valid, the VC remains in the active status and continues transmitting the packet. While a valid packet descriptor exists in the transmit queue, the channel remains in the active status.

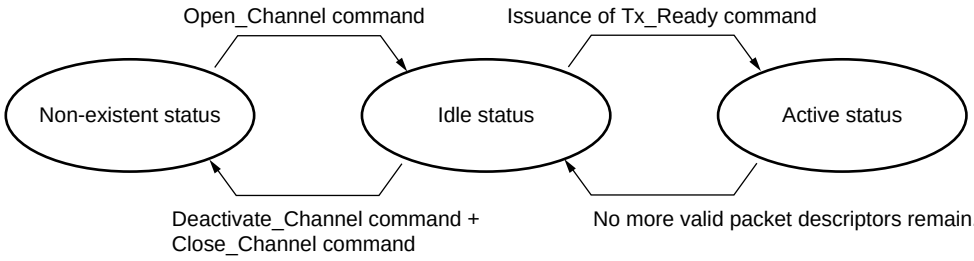
When the μPD98409 has read a blank packet descriptor (V = 0), it recognizes that there no more packets remain. When the μPD98409 has read all the data of the last packet, it stores the Tx indication, removes the transmit VC table from the link list of the shaper, and clears the "A" bit to "0". As a result, this channel returns to the idle status. The channel returns to the idle status when the μPD98409 has completed reading from the system memory, not when it has completed transmitting all the data of the packet.

The transmit queue read pointer of the channel that has returned to the idle status indicates the last blank packet descriptor. Setting of the next valid packet desciptor is started from this blank packet descriptor. Therefore, the host does not have to change the transmit queue read pointer. The host accesses the transmit queue read pointer only when it sets an initial value. If the Tx_Ready command is issued after the valid packet descriptor has been set, the channel enters the active status again.

To end using this channel, the host issues the Deactivate_Channel command and Close_Channel command when the channel is in the idle status, and returns the VC table to the free block pool. As a result, the channel no longer exists and enters the non-existent status. The host issues the Deactivate_Channel and Close_Channel command only when the channel is in the idle status. If these commands are issued when the channel is in the

active status, the μPD98409 malfunctions. To return a channel from the active status to the idle status, wait until the channel enters the idle status through recognition of a blank packet descriptor after no more valid packet descriptors remain in the transmit queue.

Figure 5-17. Status of Transmit Channel



5.4.4 Traffic control

(1) Cell Transmission mode

The μPD98409 transfers cells to the PHY device via the UTOPIA interface.

Two cell transmit modes can be selected by using the UCE bit of the GMR register.

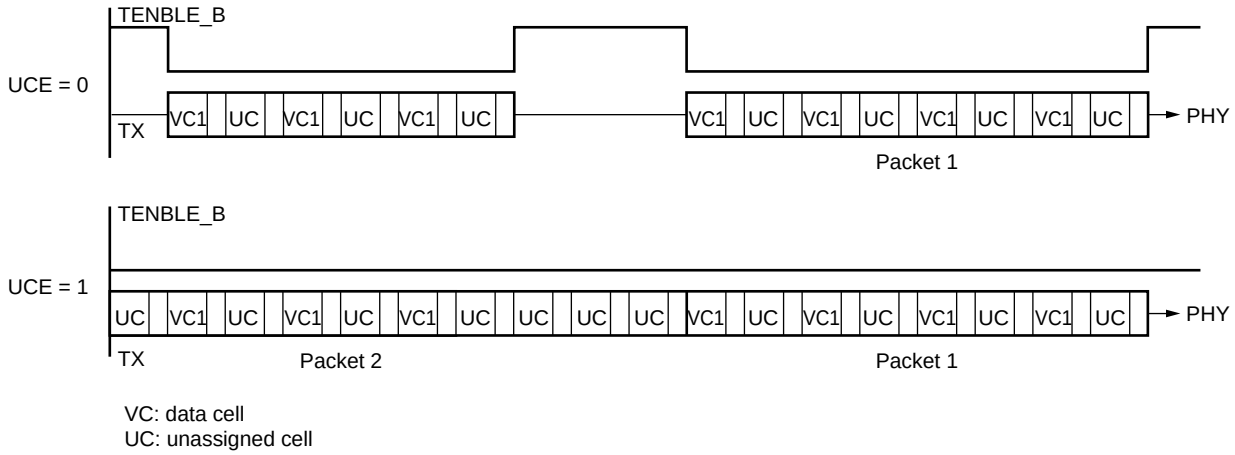
When the UCE bit is “1”, cells are successively transmitted if at least one transmit VC is active or if there is a shaper that is set as an unassigned cell/idle cell generator. Otherwise, the TENBL_B signal is made inactive and cells are not transmitted.

When the UCE bit is “1”, and if there is no data to be transmitted, an unassigned cell/idle cell selected by the ICM bit of the GMR register is output, and the TENBL_B signal is kept low.

Selects unassigned/idle cell insertion mode between packets		
UCE bit (GMR: bit 29)	1	Makes TENBL_B signal high and unassigned cell/idle cell cannot be transmitted if there is no active transmit VC or if there is no shaper set as unassigned/idle cell generator.
	0	Transmits unassigned cell/idle cells selected by ICM bit of GMR register if there is no active transmit VC or if there is no shaper set as an unassigned/idle cell generator.
	Default = 1	

Figure 5-18 shows an example of differences between the methods of transmitting unassigned cells depending on the setting of the UCE bit.

Figure 5-18. Example of Transmitting Cell (where 1 VC transmission, I/M = 1/2, P = 1, C = 1, ICM = "0")



The cell transmission rate is controlled in cell units, and up to two types of rates can be set.

(2) Scheduling

Scheduling is to determine the sequence in which cells are to be transmitted. The channel (transmit VC) how which the next cell is to be transmitted is determined by scheduling. Scheduling is implemented by the operations of the 2 traffic shapers and scheduler.

Each shaper has a corresponding scheduler register. The host assigns a parameter that determines the peak rate and average rate at which the cells are to be transmitted, to the scheduler register. Each shaper executes a dual leaky bucket algorithm operation based on the given parameter to generate cell transmission timing. The host selects the rate of a shaper to transmit cells to an opened transmit VC, and assigns the rate to the transmit VC table. The transmit VC that has prepared a valid packet and has been made active by the Tx_Ready command is linked to the set shaper. This link operation is controlled by using the shaper link list and shaper pointer entry.

The scheduler checks the parameters of all the shapers enabled, once every 24 system clocks, detects the shapers ready for transmission, and selects the shaper that has the highest priority. The cell of the transmit VC linked to this shaper is transferred. These operations are repeated to control traffic.

Figure 5-19. Concept of Scheduling Function

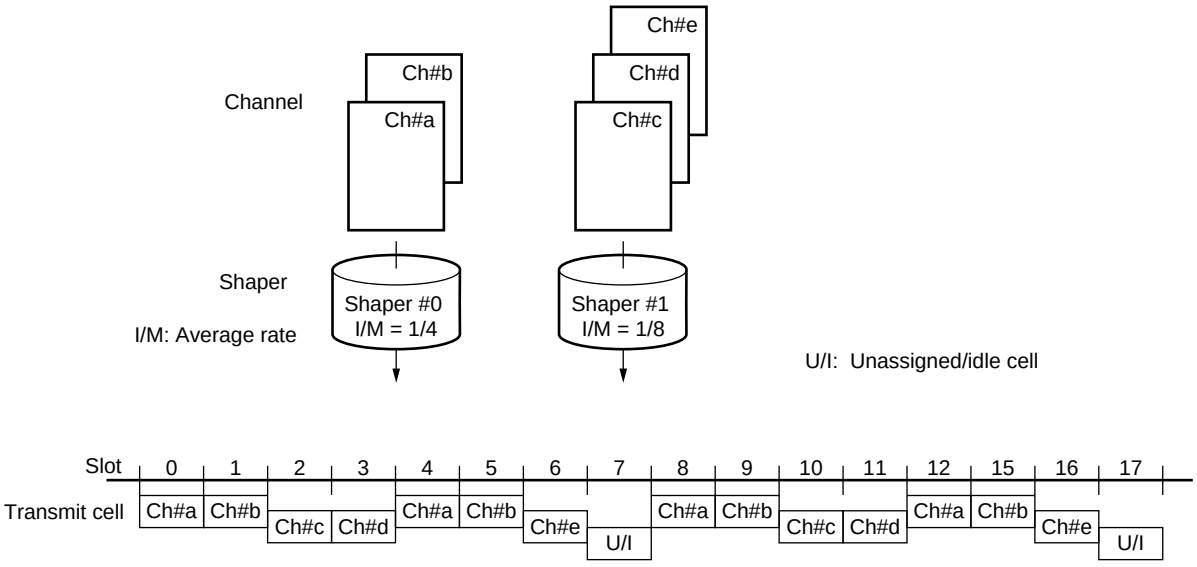
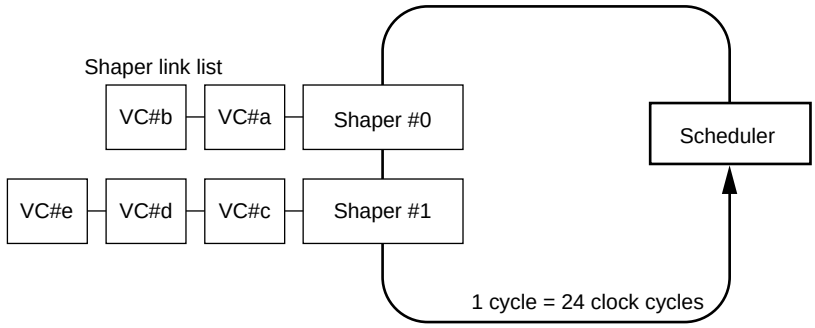


Figure 5-20. Concept of Scheduler Operation



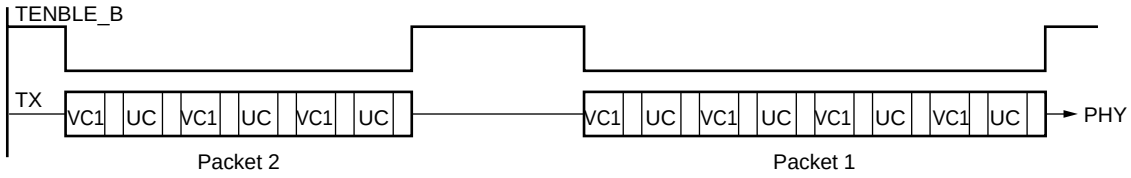
(3) Transmitting vacant cell

The μPD98409 successively transmits cells if at least one transmit VC is active. The data cell of the active VC is transmitted in accordance with the average rate and peak rate assigned to the shapers. If no shaper is ready to transmit data cells, the μPD98409 transmits a vacant cell. The user can select the type of vacant cell transmitted by the μPD98409 from two types: unassigned cells and idle cells. This selection is made by setting a bit of the GMR register. In the default mode, unassigned cells are transmitted.

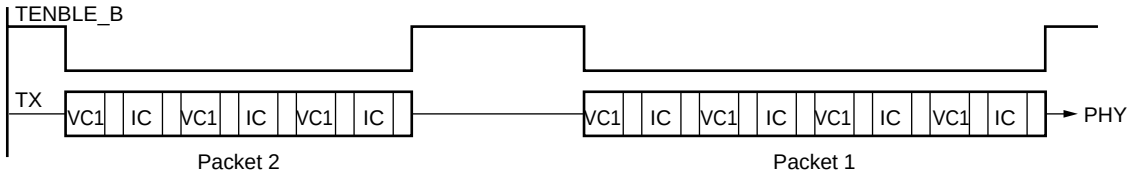
Selects vacant cell		
ICM bit (GMR: bit 28)	0	Inserts unassigned cell
	1	Inserts idle cell
	Default = 0	

<Example> Transmission with 1 VC only. Shaper setting: I/M = 1/2, C = 1, P = 0

Where ICM = "0"



Where ICM = "1"



VC1: Data cell of VC1, UC: Unassigned cell, IC: Idle cell

- Format of vacant cell

  Unassigned cell (Header: all "00H", Payload: all "00H")

	Header					Payload				
	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 1	Byte 2	• • •	Byte 47	Byte 48
Contents	00	00	00	00	00	00	00	• • •	00	00

Idle cell (Header: CLP = 1, all "00H" for others, Payload: all "6AH")

	Header					Payload				
	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 1	Byte 2	• • •	Byte 47	Byte 48
Contents	00	00	00	01	00	6A	6A	• • •	6A	6A

(4) Scheduler register

Each of the 2 shapers has a scheduler register that stores the parameters set by the host and the variables managed by the μPD98409.

The user assigns the parameters to the scheduler register of the shaper to be used, to determine rates before starting transmission. The scheduler registers are in the indirect address register, and the host can read or write these registers by using the Indirect_Access command.

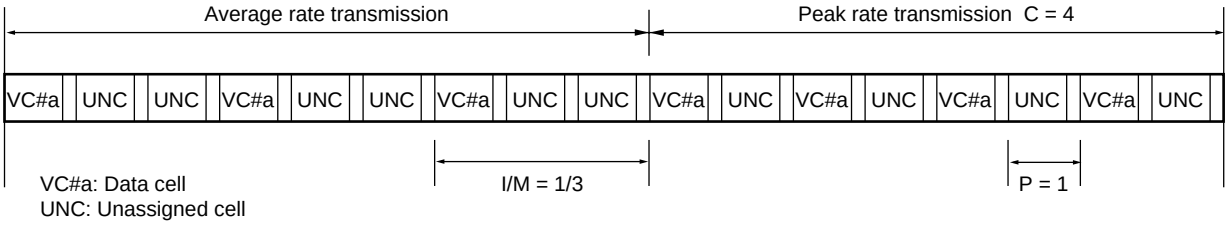
Table 5-3. Description of Bits of Scheduler Register

Bit Name	Description
I (8 bits), M (24 bits)	Average rate. Sets average rate parameter in cell units. Sets I/M to indicate I cells transmitted per M cell.
P (peak, 8 bits)	Peak cell rate. Sets minimum interval between two successive cells of channel linked to shaper in cell units.
C (credit, 8 bits)	Credit. Sets maximum number of credits shaper can accumulate. Maximum burst size is determined by parameters I, M, P, and C.
PRIORITY (priority, 1 bit)	Sets priority to be assigned to shaper. 0 is highest priority, and 1 is lowest. **Remark** The user must set a priority higher than or the same as that of shaper 1 to shaper 0. Shaper 1 must not be assigned a priority higher than that of shaper 0.
E (enable, 1 bit)	This bit enables the shapers.

Figure 5-21. Example of Actions of Parameters

<Example>

I/M = 1/3 (average rate)
P = 1 (peak rate)
C = 4 (credit)



The μPD98409 internally generates variables for each shaper from the parameters set by the user in Figure 5-21, and executes an algorithm operation. Each shaper has flags that recognize the status of the shaper. These variables and flags are managed by the μPD98409, using each scheduler register. If the host sets the shaper enable bit "SE" of the GMR register and if the enable bit "E" of each shaper is set to 1, the operation is started and the variables and flags are updated.

Variable	x (32 bits)
	y (32 bits)
	p (8 bits)
	c (8 bits)
Flag	S (scan, 1 bit)
	R (round-robin, 1 bit)
	A (active, 1 bit)

Caution The I, M, P, and C parameters and all the contents of the scheduler registers must not be changed while the shaper is active (A =1) and is transmitting a packet. If they are changed, the μPD98409 cannot keep the rate. To change the values of I, M, P, or C to change the set rate, clear enable bit E to 0 and clear all the register contents to 0 while the shaper is active (A = 0) (however, without manipulating the "A" bit when the unassigned cell generator function is used).

(5) Shaper link list

The user selects a shaper at the link destination for each transmit VC and the host assings the number of the selected shaper to the 4-bit “Shaper No.” in the transmit VC table. Several VCs can be linked to a specific shaper. The μPD98409 automatically links the transmit VC to the shaper or releases the VC from the link.

The μPD98409 generates a link list of the VC tables for each shaper, and manages a multiplexed active VC. The link list of a shaper is managed by using the seventh word of the VC table. This word has a “Forward Pointer”, “Backward Pointer”, and “Lst” bit that indicates the end of the list. A code indicating the beginning of the VC table, “VC number”, is written to the pointers. The μPD98409 links the VC and shaper, or releases the VC from the link, by changing this code as necessary.

Figure 5-22. Pointers Used for Linking

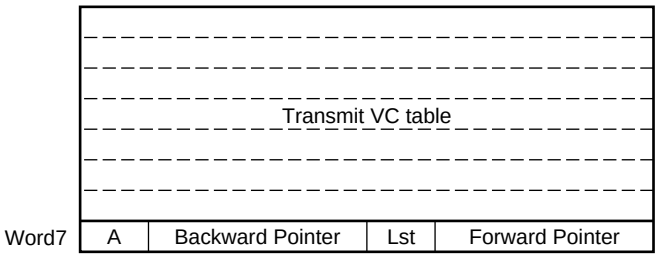
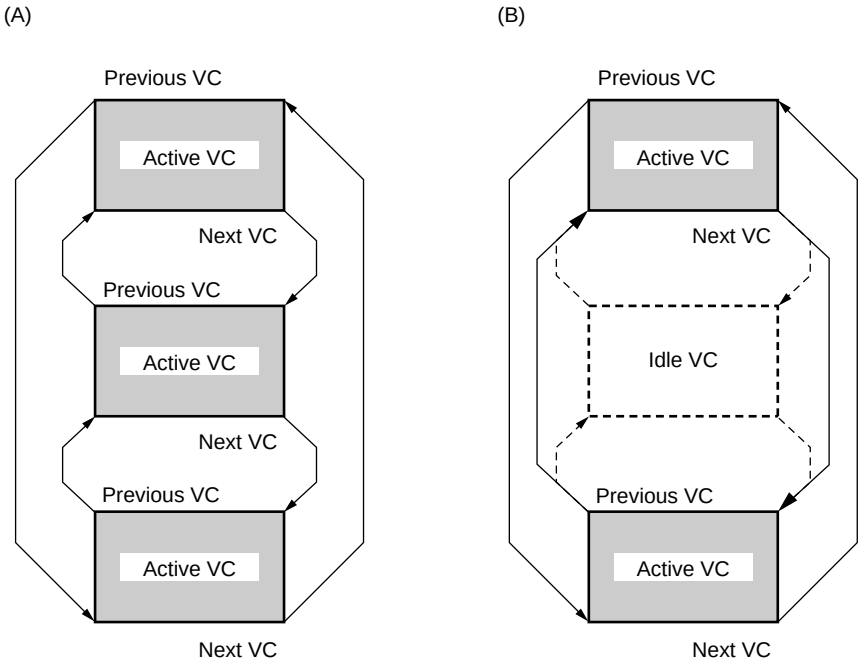


Figure 5-23. Linking/Releasing Transmit VC to/from Traffic Shaper



(A) Linking to shaper

A VC is linked to a shaper when the host issues the Tx_Ready command to a transmit VC in the idle status (A bit = 0). The μPD98409 obtains the number of the shaper to which the VC is to be linked from the "Shaper No." field of the VC table, and rewrites the VC linked to the end of the link list of the shaper, and the "Forward Pointer" and "Backward Pointer" of the VC to be added. At this time, the active bit A of the VC table is set to 1 to set the VC in the active status. If no VC is linked to the shaper, and if the VC to be linked is at the beginning of the link list, the active flag "A" bit of the scheduler register of the shaper and the "a bit" at the shaper pointer entry are set to 1 to make the VC active.

If the Tx_Ready command is issued to an active VC that already exists in the link list of the shaper, the μPD98409 does nothing.

(B) Releasing from link

A VC is released from the link when the VC returns to the active status from the idle status. If the packet descriptor fetched next is a blank packet descriptor, and if all the data of the packet being transmitted has been completely read, the "Forward Pointers" and "Backward Pointers" of the VC and those before and after that are rewritten. As a result, the VC is released from the link list. At this time, the active bit A of the VC table is cleared to 0.

If the VC to be released from the link is the last VC linked to the shaper, the "A bit" in the scheduler register and "a bit" of the shaper pointer entry are cleared to 0 as soon as the VC has released from the link, and the shaper is made inactive.

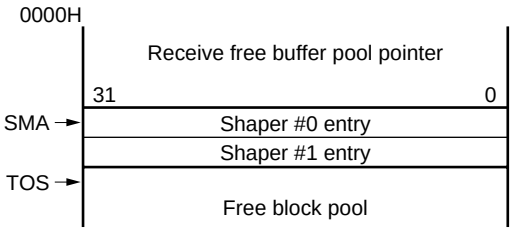
The μPD98409 traces the "Forward Pointer" of the link list, and proceeds with transmission processing for each VC. When executing the transmission processing of a certain VC, the μPD98409 assigns the "VC Number" stored in the "Forward Pointer" of the VC table, i.e., the "VC Number" of the VC subject to the next processing, to the shaper pointer of the control memory. In this way, the μPD98409 determines for which VC it should perform processing when it has moved between shapers.

(6) Shaper pointer entry

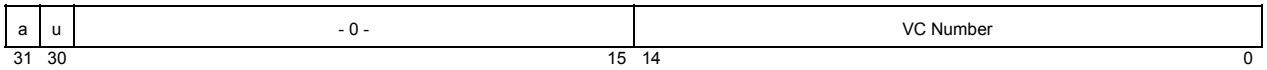
The shaper pointer entry is a table that stores a record indicating up to which VC the μPD98409 has performed processing when the μPD98409 has moved between shapers, and is located in the shaper pointer area of the control memory. The shaper pointer area starts from the address assigned by the user to the SMA register, and its size is 2 words. The μPD98409 uses this area as the entry of shaper 0, entry of shaper 1, starting from the address of SMA.

The shaper pointer entry is mainly used and managed by the μPD98409 as a table. The host can read or write this area by using the Indirect_Access command. However, the host need to access this area only when it uses the unassigned cell generator function; otherwise, the host does not need access the area.

Figure 5-24. Shaper Pointer Entry of Control Memory



The format of the shaper pointer entry is as follows:



a bit : Active flag bit. This bit is mainly used as an internal flag by the μPD98409.
1: At least one active VC is linked to the shaper.
0: There is no active VC.
When this shaper is used as an unassigned cell generator, the host sets this bit and u bit to 1.

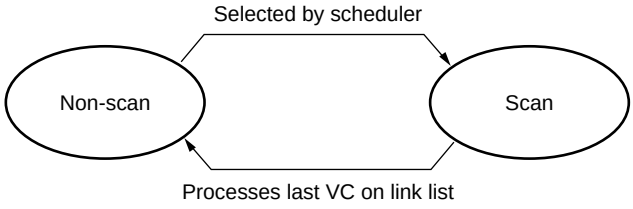
u bit : Unassigned cell generator. This bit is set to 1 by the host only when this shaper is used as an unassigned cell generator (for details, refer **to 5.4.4 (9) Unassigned (idle) cell generator function**).

VC Number : This field is used as a table to which the μPD98409 stacks the "VC Number" of the VC to be transmitted next. It is all 0 if no VC is linked to the shaper.

(7) Algorithm operation

The cell rate is controlled by executing a dual leaky bucket algorithm operation with the scheduler selecting a shaper. The scheduler checks the variable parameters of all the shapers once every 24 system clocks, and select one of the shapers for transmission processing. The status of the selected shaper is called the scan status, while the status of the shaper not selected is called the non-scan status. These two statuses are indicated by the "S flag" of the scheduler register. The shapers for which the S flag is set to 1 executes cell transmission processing of the linked VC.

Figure 5-25. Shaper Status



A shaper becomes a candidate to be selected by the scheduler when it satisfies all the following conditions.

(i) SE = 1 (GMR register)

All the shapers are enabled to operate. This is set by the host to the GMR register.

(ii) E = 1 (scheduler register)

The selected shaper is enabled. The host sets the E bit of the scheduler register corresponding to the shaper to be used to 1.

(iii) A = 1 (scheduler register)

An active VC to be transmitted is linked to the shaper. This is set by the μPD98409.

(iv) c variable > 0 (scheduler register)

The μPD98409 updates the c variable according to the I, M, and C parameters given by the user.

(v) p variable = 0 (scheduler register)

The μPD98409 updates the p variable according to the P parameter given by the user.

Conditions (i) through (iii) indicate whether the VC to be transmitted is linked to the shaper and do not directly affect the transmission rate. The transmission rate is determined when the c parameter, which is the condition of (v), is incremented to a value greater than 0 and when the p parameter reaches to 0.
The conditions in which the 8-bit parameter "c" of the scheduler register (iv) is incremented are as follows:

(a) SE = 1
(b) E = 1
(c) $x \geq y$
(d) $c < C$

The "c" parameter is incremented depending on the relation between variables x and y managed by the μPD98409, as well as under the conditions (a) and (b), in which the shaper operation is enabled. The upper-limit to which the c parameter is to be incremented is up to the C parameter given by the user.
Variable parameters x and y are updated according to the following rules in every cycle (24 clocks) in which the scheduler checks all the shaper parameters in accordance with the I and M parameters given by the user.

- When $x = y \rightarrow$ I and M are loaded to x and y, respectively.
- When $x > y \rightarrow$ I is added to x and M is added to y.
- When $x < y \rightarrow$ I is added to x.

"c" is incremented where $x \geq y$. If $c > 0$, and $p = 0$, the shaper becomes a candidate for selection. When the shaper is selected, it enters the scan status and transmits cells.
Each time the shaper changes its status from scan to non-scan (when transmission of all linked VCs has been completed), the c parameter is decremented.
The conditions under which "c" is incremented do not include the condition that the shaper is active (scheduler register: A bit = 1). Therefore, the "c" parameter is incremented at a cycle of I/M until it reaches to "C", regardless of whether an active VC is linked to the shaper.

"p" is loaded by "P" given by the user when the shaper has been selected and is in the scan status, and is decremented by one each timer the scheduler makes a round of the shapers (in 24 clocks). So that the scheduler selects the same shaper, p must be decremented to 0. Consequently, the interval between cells of the same VC is equal to at least the peak rate "P".

Example: When only shaper 0 is used and the following parameters are set to the scheduler register:

I = 3, M = 10, P = 0, C = 4

Open a transmit channel (VC#a) and link it to shaper 0.
UN: Unassigned cell or idle cell
VC: Data cell

Slot	x Value	y Value	x ? y	c	Scan	Transmit Cell
1 initial value	0	0	x = y	1		
2	3	10	x < y	1		
3	6	10	x < y	1		
4	9	10	x < y	1		
5	12	10	x > y	2		
6	15	20	x < y	4		
7	18	20	x < y	2		
8	21	20	x > y	3		
9	24	30	x < y	3		
10	27	30	x < y	3		
11	30	30	x = y	4		
12	3	10	x < y	4		
13	6	10	x < y	4		
14	9	10	x < y	4		
15	12	10	x > y	4		
16	15	20	x < y	4	●	VC
17	18	20	x < y	3	●	VC
18	21	20	x > y	3	●	VC
19	24	30	x < y	2	●	VC
20	27	30	x < y	1	●	VC
21	30	30	x = y	1	●	VC
22	3	10	x < y	0		U/I
23	6	10	x < y	0		U/I
24	9	10	x < y	0		U/I
25	12	10	x > y	1	●	VC
26	15	20	x < y	0		C
27	18	20	x < y	0		U/I
28	21	20	x > y	1	●	VC
29	24	30	x < y	0		U/I
30	27	30	x < y	0		U/I
31	30	30	x = y	1	●	VC
32	3	10	x < y	0		U/I
33	6	10	x < y	0		U/I
34	9	10	x < y	0		U/I
35	12	10	x > y	1	●	VC
36	15	20	x < y	0		U/I

Tx_Ready command issued →

Slot	x Value	y Value	x ? y	c	Scan	Transmit Cell
37	18	20	x < y	0		U/I
38	21	20	x > y	1	●	VC
39	24	30	x < y	0		U/I
40	27	30	x < y	0		U/I
41	30	30	x = y	1	●	VC
42	3	10	x < y	0		U/I
43	6	10	x < y	0		U/I
44	9	10	x < y	0		U/I
45	12	10	x > y	1	●	VC
46	15	20	x < y	0		U/I
47	18	20	x < y	0		U/I
48	21	20	x > y	1	●	VC
49	24	30	x < y	0		U/I
50	27	30	x < y	0		U/I
51	30	30	x = y	1	●	VC
52	3	10	x < y	0		U/I
53	6	10	x < y	0		U/I
54	9	10	x < y	0		U/I
55	12	10	x > y	1	●	VC
56	15	20	x < y	0		U/I
57	18	20	x < y	0		U/I
58	21	20	x > y	1	●	VC
59	24	30	x < y	0		
60	27	30	x < y	0		
61	30	30	x = y	1		
62	3	10	x < y	1		
63	6	10	x < y	1		
64	9	10	x < y	1		
65	12	10	x > y	2		
66	15	20	x < y	2		
67	18	20	x < y	2		
:	:	:	:	:		:
:	:	:	:	:		:

3/10

Packet transmission completed →

"C" specifies the number of cells that can be successively transmitted at the peak rate. If C = 1, therefore, an unassigned cell is inserted because the shaper cannot successively transmit two or more cells at the peak rate.

<Example> Setting of shaper (Priority is not considered.)

- I/M = 1/1
- P = 0

- When C = 1

An unassigned cell is inserted because two or more cells cannot be transmitted successively.

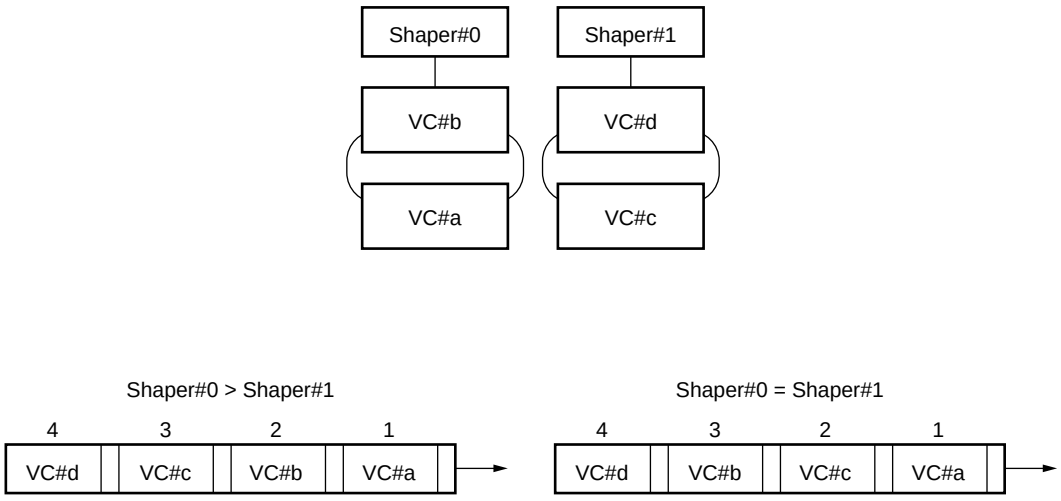
- When C = 2

Two or more cells can be transmitted successively.

When the scheduler makes a round of 2 shapers, it selects the shaper having the highest priority from all the candidates that satisfy the given condition. The priority of the shaper is set by the user, using the "Priority" parameter. The user must set a priority higher than or the same as that of shaper 1 to shaper 0. Shaper 1 must not be assigned a priority higher than that of shaper 0.
If two shapers having the same priority exist, the scheduler selects one shaper at a time by means of round robin. The μPD98409 uses the round-robin (R) bit of the scheduler register to manage the round-robin algorithm.

<Example> When two shapers are used with two VCs are linked to each, the priority of cell transmission is as shown in Figure 5-26 according to the priority setting.

Figure 5-26. Cell Transmission Sequence According to Priority



(8) Cell transmission by shaper in scan status

A shaper that has been set in the scan status sequentially transmits the cells of all the VCs linked on a one-by-one basis, and returns to the non-scan status when it has transmitted the cell of the VC (LST bit = 1) linked last. This mode means that the transmit rate set to the shaper is controlled in units of each VC. However, the band occupied by one shaper changes depending on the number of VCs linked to the shaper.

(9) Unassigned cell generator function

The user can use one or more shaper as an unassigned cell generator to limit the band used by the active VC. The shaper set as an unassigned cell generator functions as if it were always linked to the VC that transmits an unassigned (idle) cell. By assigning a priority higher than that of the shaper for data transmission to the shaper set as an unassigned cell generator, the band used by data can be limited.

The unassigned cell to be transmitted can be changed to an idle cell by using the ICM bit of the GMR register.

The unassigned cell generator is set in the following procedure.

<1> Set bits 31 and 30 (“a” and “u”) of the shaper pointer entry in the control memory of the shaper to be set as an unassigned cell generator to 1.

<2> Set parameters to the scheduler register. At this time, however, set enable bit E and active bit “A” to 1 at the same time.

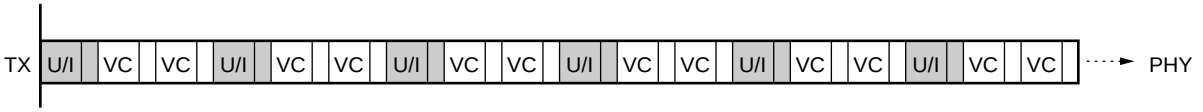
Cautions 1. A normal VC cannot be linked to the shaper set as an unassigned cell generator.

2. The host does not have to set the “a” bit of the shaper pointer entry and “A” bit of the scheduler register to “1” when the unassigned cell generator function is not used. These bits are automatically set to “1” by the μPD98409 for the shaper for data transmission.

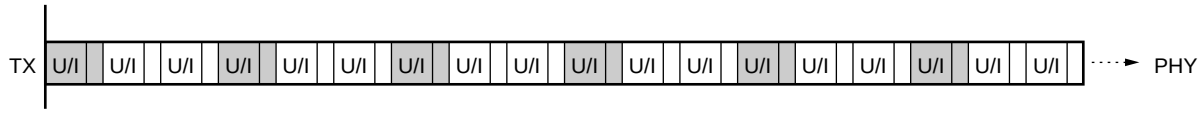
3. Use the unassigned cell generator function only when the UCE bit of the GMR register is “0”. If the UCE bit is “1” so the mode where an unassigned cell/idle cell is not transmitted between packets is selected, the unassigned cell generator function cannot be used.

<Example> Shaper 0 : Set as unassigned cell generator I/M = 1/3
Shaper 1 : Sets normal cell data I/M = 1/1

Data cell to be transmitted can be embedded between unassigned cells.



When there is no data cell to be transmitted.



: Cell generated by unassigned cell generator

U/I : Unassigned cell where ICM = 0
Idle cell where ICM = 1
VC : Data cell

5.4.5 Transmit operation

The μPD98409 takes a transmit segment (payload data of cell: 48 bytes) from the packet stored to the system memory, and appends an AAL-5 trailer or CRC-10 to this segment, as necessary, and transmits the cell to the PHY device.

The μPD98409 has a FIFO of 2 cells. During normal operation, this FIFO stores transmit data from the system memory, and becomes empty after the cells have been transmitted to the PHY device.

The transmit machine reads a segment from the system memory in accordance with the order of VCs determined by the scheduler.

The μPD98409 organizes cells by using the segment (payload data: 48 bytes) taken out from the memory and the cell header information stored in the control memory. At this time, it inserts dummy data “00H” in the HEC field of the cell header.

The GFC, VPI/VCI, PTI, and CLP fields of the cell header are taken from Word0 of the VC table and generated. The first word of the VC table is updated by the μPD98409 each time a packet descriptor has been received from a transmit packet.

To the “Remaining Bytes in Current Buffer” of the VC table, the contents of the “SIZE” field of the packet descriptor are stored in the single-buffer mode. In the multi-buffer mode, the contents of the “SIZE” field of each buffer descriptor are stored in this field.

In the case of the last cell, the μPD98409 sets “1” to the LSB of the PTI field, and the CLP bit is set in accordance with the CLP mode indicated by the VC table.

The 48 bytes starting from the “Buffer Read Address” field of the VC table are read from the system memory and embedded in the payload of the cell.

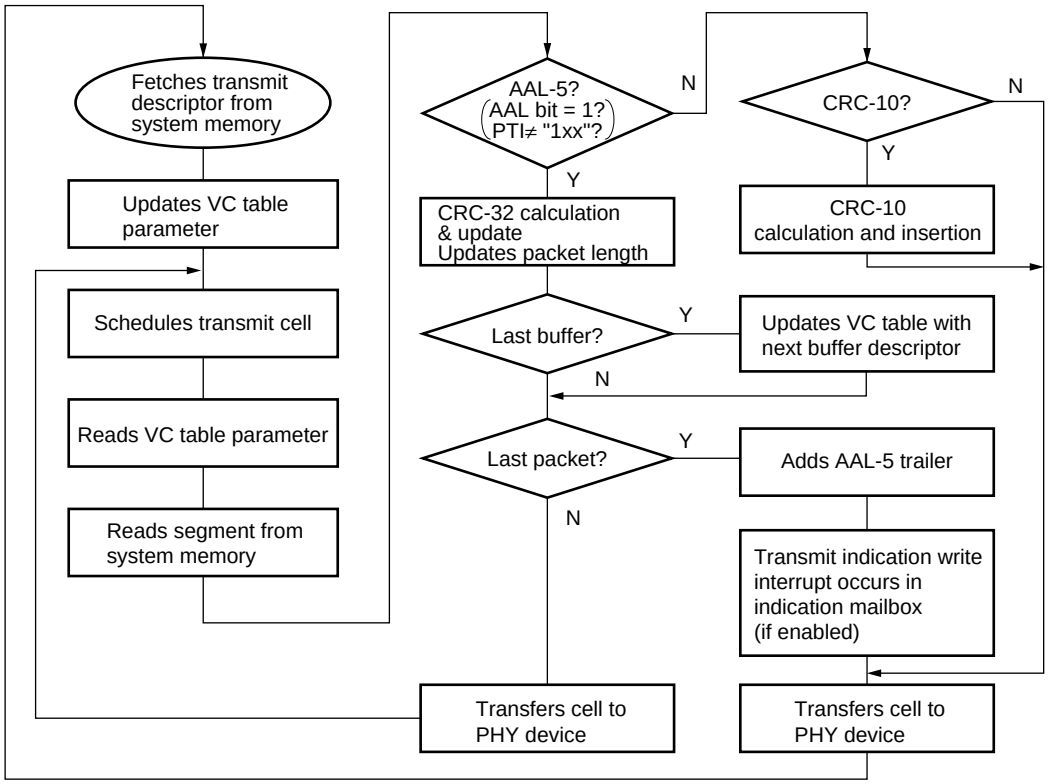
Next, 48 bytes are added to this field, and the 48 bytes are subtracted from the “Remaining Bytes in Current Buffer” field. In the multi-buffer mode, the “Remaining Bytes in Current Buffer” field becomes 0. When no more capacity is available in the buffer, the “Remaining Bytes in Current Buffer” field is updated from the SIZE field of a new buffer descriptor. The μPD98409 manages the information on the current transmit buffer by using the VC table.

If the field of the VC table is less than 40 bytes when the L flag of the current transmit buffer indicates the last buffer, the current cell is the last cell of an AAL-PDU. If the current cell is the last cell of an AAL-PDU, and has a room to add an 8-byte AAL-5 trailer, the μPD98409 adds the trailer with padding (data of all 0) of an appropriate number. If there is no room to add 8 bytes of the trailer to the last cell, a cell containing only padding and trailer information is added and transmitted.

During AAL-5 traffic (packet descriptor AAL bit = 1), the transmit machine stores the preliminary result of CRC-32 operation and packet length of each segment in the “CRC-32” field and “No. Of Bytes Transmitted This Packet” field of the VC table each time it has read a segment from the system memory. When the last segment of AAL-5 PDU has been read, the last value of CRC-32 and packet length are inserted in the trailer of AAL-5 PDU, the contents of the first word are inserted in the CPCS-UU and CPI fields, and an AAL-5 trailer is generated.

The transmit machine makes the VC inactive when there are no more valid packet descriptors in the transmit queue. In order to be able to detect that no more valid packet descriptors remain in the transmit queue, the host must locate a vacant packet descriptor with V bit = 0 to the end of the list of the valid packet descriptors.

Figure 5-27. Outline of Transmit Operation



5.4.6 Support of non-AAL-5 traffic

To support non-AAL-5 traffic, the μPD98409 has a function to transmit raw cells without appending an AAL-5 trailer. The μPD98409 executes non-AAL-5 transmission in two cases: (1) to transmit an OAM F5/RM cell, and (2) to transmit raw cell.

(1) Transmitting OAM F5/RM cell

When the host assigns the pattern of an OAM F5/RM cell (100, 101) to the PTI field of the packet descriptor, the μPD98409 does not append an AAL-5 trailer. In this case, the μPD98409 reads the 48-byte data of one cell from the beginning of the data buffer and ignores the rest of the data even if the host assigns more than 48 bytes to the "SIZE" field of the packet descriptor. Do not assign less than 48 bytes to the PTI field. When the transmit data is less than 48 bytes, extends the data length to 48 bytes by inserting dummy data (such as all 0). To transmit an OAM F5/RM cell, the host assigns one packet descriptor per OAM F5/RM cell.

CRC-10 operation can be inserted when transmitting an OAM F5/RM cell. If the "C10 bit" of the packet descriptor is set to 1, the μPD98409 executes a CRC-10 operation on the 46 bytes and 6 bits of the 48 bytes of the segment of the OAM F5/RM cell, and writes the result over the last 10 bits as a CRC-10 error detection code. When the CRC-10 operation insertion function is enabled, therefore, the host must assume that the last 10 bits are overwritten and dummy data (such as all 0) must be located in that portion in the data buffer. The CRC-10 operation insertion function can be executed only when the "SIZE field" of the packet descriptor is 48 bytes long. If the data to be actually transmitted is less than 48 bytes, and if the CRC-10 operation insertion function is to be used, the host must append dummy data to the data to extend the data length to 48 bytes.

When setting the packet descriptor to transmit an OAM F5/RM cell, the following points must be noted.

- The "AAL" bit must be always cleared to 0 when transmitting an OAM F5/RM cell.
- The multi-buffer mode cannot be used. Always use the single-buffer mode.

(2) Transmitting raw cell

When transmitting a user data packet other than that of AAL-5 type, the host clears the "AAL" bit of the packet descriptor to 0, and assigns "0xx" (000 to 011) indicating the code of a user data cell to the PTI field. The μPD98409 does not add an AAL-5 trailer to the packet if the "AAL" bit of the packet descriptor is 0. It repeatedly reads 48 bytes of the transmit segment from the data buffer and transmits them to the PHY device as a cell. The transmit data of one packet can be set to a length of up to 65535 bytes, and a data buffer can be configured in the multi-buffer mode. If the "SIZE" field of the packet descriptor is not an integer multiple of 48 bytes, the μPD98409 adds padding (all 0) to extend the segment of the last cell to 48 bytes.

The host can set the "C10" bit of the packet descriptor to 1 as necessary, and enable the CRC-10 operation insertion function of the μPD98409. When this function has been enabled, the μPD98409 executes the CRC-10 operation on the first 46 bytes and 6 bits of the 48 bytes of the segment read from the data buffer of the system memory, and writes the result of the operation over the last 10 bits of the cell as an error detection code. To prevent the CRC-10 code from overwriting the data, the host must insert dummy data in segment units to the 10 bits which are to be overwritten, when it prepares the data in the system memory.

If the length of the data buffer is not an integer multiple of 48 bytes, the μPD98409 cannot execute a CRC-10 operation on the last segment of less than 48 bytes. Therefore, the host must always extend the user data length to an integer multiple of 48 bytes when it uses the CRC-10 operation insertion function.

When raw cells are transmitted, the pattern set to the PTI field of the packet descriptor is transmitted as is even when the last cell is to be transmitted, and the least significant bit of the PTI field is not automatically changed to "1".

5.4.7 Issuance of transmit indication

The μPD98409 writes transmit indication to a mailbox for each packet descriptor. Mailbox 2 or 3 is used for transmission, and the transmission indication is stored in either of these mailboxes in accordance with the “MB” bit of the packet descriptor set by the host.

The μPD98409 writes the transmit indication when all the data of a packet has been completely read. Even if the transmit indication has been issued, transmission of the packet to the PHY device may not have been completed.

After storing the transmit indication in a mailbox, the μPD98409 sets the corresponding MM bit of the GMR register to 1, and, if not masked, issues an interrupt.

Unlike the receive indication, the transmit indication is issued to each packet for all the data of AAL-5 packets and non-AAL-5 packets (OAM F5/RM cell, RM cell, and raw cell transmit packets).

For the details and processing of the indication, refer to **5.6 Transmit/Receive Indication**.

The μPD98409 can transmit the three types of data shown in Table 5-4. It selects the data type by using the packet descriptor.

Table 5-4. Summary of Data Supported by μPD98409 Transmission Function

	Setting	Data Length	Indication	AAL-5 Trailer	CRC-10	Remark
AAL-5 data	PTI=“0xx” AAL=1	65535 bytes MAX.	Issued	Automatically added	Cannot be inserted	
Packet transmission other than AAL-5 type	PTI=“0xx” AAL=0	65535 bytes MAX.	Issued	None	Can be inserted	• Transmit data length must be integer multiple of 48 bytes when CRC-10 is added. • CLPM field cannot be set to “01” • Least significant 1 bit of PTI field of last cell is not changed
OAM F5/RM cell	PTI=“1xx” AAL=0	48 bytes MAX.	Issued	None	Can be inserted	• Data length must be always 48 bytes when CRC-10 is added. • CLPM field cannot be set to “01” • Can be used only in single-buffer mode

5.5 Reception Function

5.5.1 Reception processing flow

The μPD98409 receives a cell in several steps.

Figure 5-28. Outline of Reception Flow

Host Processing	
(1)	Sets VRR register
(2)	Sets receive pool in system memory
(3)	Sets receive pool descriptor in control memory
(4)	Sets mailbox
(5)	Issues Open_Channel command
(6)	Sets receive VC table
(7)	Sets look-up table entry in control memory

(18)	Reads receive indication
(19)	Updates mailbox read pointer (MTA)
(20)	Reads data from receive pool
(21)	Manages buffer of receive pool (Add_Batches command)

Operation of μPD98409 (AAL-5 packet reception)	
(8)	Receives first cell of packet
(9)	Assigns batch of pool
(10)	Links VC to T1 timer list (monitors T1 error)
(11)	Transfers data to system memory in segment units
(12)	Assigns new batch as necessary
(13)	Receives last cell of packet
(14)	Checks error from AAL-5 trailer
(15)	Issues receive indication and interrupt

Operation of μPD98409 (for MPEG-TS packet reception)	
(8)*	Receives first cell of packet
(9)*	Links VC to T1 timer list (monitors T1 error)
(10)*	Transfers data to TS address
(11)*	Receives last cell of MPEG-TS packet
(12)*	Checks error from AAL-5 trailer
(13)*	Updates number of receive packets and number of error packets
(14)*	Checks validity of MPEG-TS packet
(15)*	Deletes AAL-5 padding and trailer, and transfers them to TS address
(16)*	Transfers dummy cells as necessary
(17)*	Issues receive indication and interrupt only if error occurs

Operation of μPD98409 (raw cell reception)	
(8)**	Receives cell
(9)**	CRC-10 check
(10)**	Transfers data to system memory as raw cell data
(11)**	Sets RCR bit of GSR register and issues interrupt

The μPD98409 supports 64 VC (in any combination of receive VC and transmit VC). Of the 24 bits of the receive VPI/VCI, the μPD98409 can use up to 15 bits to identify a receive VC. The user determines the method to convert these 15 bits into information and selects this conversion method using the VRR register (1). Before executing reception, prepare a reception pool to store the receive data in the system memory (2). Write information on the system memory such as the receive pool size and address to the free buffer pool pointer area of the control memory as a pool descriptor (3). In addition, assign a mailbox area in the system memory in which the receive indication is to be stored.

The host issues the Open_Channel command for connection to be received and opens a channel (VC) (5). The μPD98409 allocates a block to be used as a VC table from the free block pool of the control memory, and returns its address (VC number) to the host. In turn, the host initializes parameters in the receive VC table (6). Next, it assigns the 15-bit information obtained from the receive VPI/VCI and a flag that enables reception by the VPI/VCI to a 16-bit area corresponding to the VC number of the lookup table, in order to actually start reception (7).

The μPD98409 searches the receive VPI/VCI registered in the lookup table when it has received the first cell of a packet from the PHY device (8). If the receive VPI/VCI is registered, it determines, by using the enable flag, whether to receive or drop the cell. To receive the cell, the μPD98409 accesses the VC table using the VC number obtained from the lookup table, and assigns the batch of the pool to the VC (9). It also adds the VC to the list of the T1 timer and monitors time out (10). Each time a cell is received, transfer to the system memory in segment units is repeated (11). If the batch becomes full in the middle, a new batch is fetched and assigned (12). When the last cell of the packet is received (13), an error check is performed based on the trailer information included in the cell (14), receive indication is issued to the mailbox specified by the VC table (15), and an interrupt is issued to the host if not masked (15). The host reads the receive indication (18), updates the read pointer of the mailbox (19), and receives the receive data from the pool (20). The host issues the Add_Batches command to add the number of batches if the batches remaining in the pool of the system memory has run out (21).

The μPD98409 can dedicate one VC to reception of an MPEG-TS packet by setting the VC table. When an MPEG-TS packet is received, receive data is transferred to the TS address set in the receive VC table. Setting the receive pool in the system memory (2), setting the receive pool descriptor in the control memory (3), reading data from the receive pool (19), and management of the buffer of the receive pool (21) are not necessary. When the μPD98409 receives the cell of the MPEG-TS packet (8)*, it links the VC to the T1 timer list (9)*, and transfers data to the TS address (10)*. The transfer burst size at this time is automatically determined by the μPD98409. When the last cell of the MPEG-TS packet has been received (11)*, an error check is performed from the information on the AAL-5 trailer (12)*, and the receive packet counter and error packet counter on the VC table are updated (13)*. In addition, the validity of the MPEG-TS packet is checked (14)*. After that, AAL-5 padding and trailer are deleted (15)*, only the payload is transferred to the TS address, and a dummy cell is transferred as necessary if the MPEG-TS packet is inappropriate because a cell has been dropped (16)*.

If an error occurs, the receive indication is issued to the mailbox specified by the VC table (17)*, and an interrupt is issued to the host if not masked. If no error occurs, neither interrupt nor indication is issued. Issuing of the receive indication and interrupt of (17)* can be disabled by setting the VC table. In this case, the host does not have to perform processing step (18) and those that follow.

The μPD98409 has a raw cell receive function that transfers received cells in cell units to the system memory as is, in order to receive traffic other than AAL-5 type. This function can be selected by setting the VC table. To receive a raw cell, when a cell is received (8)**, a CRC-10 check is executed (9)**, and raw cell data is stored in the free buffer of a specified pool (10)**. When a raw cell is received, processing is completed in cell units, and the receive indication is not stored in a mailbox, unlike when an AAL-5 packet is received. Each time a cell is stored, or after the last cell of a packet is received, the RCR bit of the GSR register is set and an interrupt occurs (11)** according to the setting of the VC table.

Please bear in mind that the above description of the receive process is an just outline to explain the flow of the reception processing.

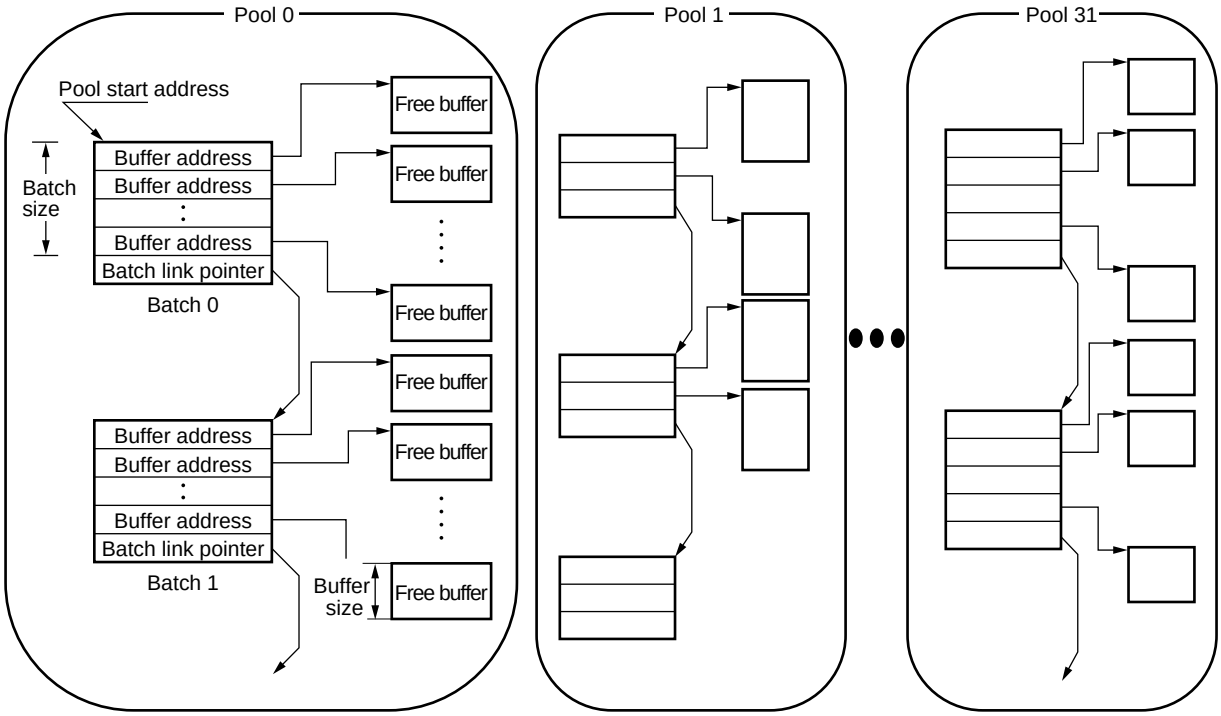
5.5.2 Structure of receive data

When receiving an AAL-5 packet other than an MPEG-TS packet or a raw cell packet, the following receive pool must be configured in the system memory. To receive an MPEG-TS packet, the receive pool is not necessary because receive data is transferred to the TS address set in the receive VC table.

(1) Receive pool

The μPD98409 stores receive data other than the MPEG-TS packet to a pool provided in the system memory. The user must install up to 32 pools in the system memory in accordance with the rules described below before letting the μPD98409 start reception, and assign the information on the pools, such as addresses and size, to the "receive free buffer pool pointer" area in the control memory.

Figure 5-29. Structure of Receive Pool in System Memory



Each pool consists of the following elements.

- Batch : A block that bundles the start addresses of the free buffers. One batch stores the addresses of one to 255 buffers. The number of buffers bundled by each batch of the same pool must be equal. The batch must be always located from a 32-bit boundary in the system memory. The last word is used as a link pointer, and must store the first address of the next batch to chain all the batches.

The format of the free buffer address and link pointer constituting a batch is as follows:

Buffer address

FREE BUFFER ADDRESS : This field stores the 32 bits of the start address of the free buffer that actually stores the receive data. The start address of the free buffer can be located at any byte boundary. If the low-order 2 bits of the address are not set to “00”, the μPD98409 executes a byte alignment DMA write.
For the byte alignment transfer, refer to **(8) (b) Byte alignment transfer of receive cell data** in **4.1.3 PCI bus transaction**

Link pointer

<1> AAL-5 type reception pool

NEXT BATCH ADDRESS 1 : This field stores the 32-bit start address of the next batch. Because a batch is located at a 32-bit boundary, the low-order 2 bits of this field must be always “00”.

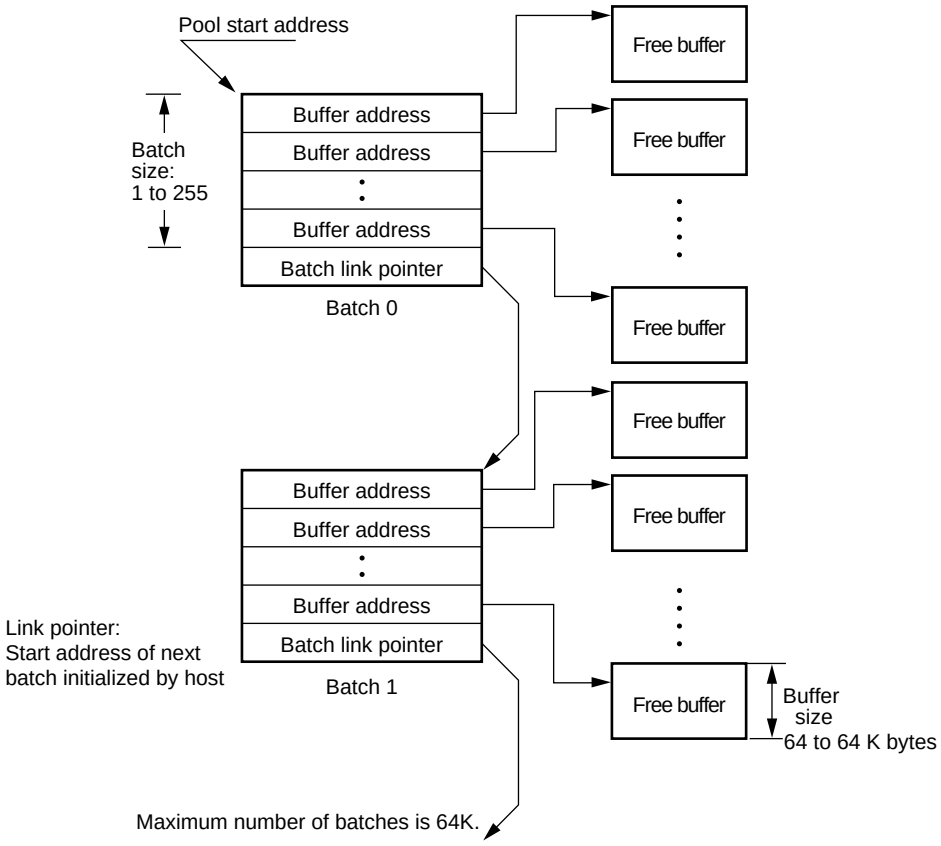
<2> Raw cell reception pool

NEXT BATCH ADDRESS 2 : This field stores the 32-bit start address of the next batch. Because the batch is allocated at a 32-bit boundary, the low-order 2 bits of the actual address are always “00”. However, the μPD98409 must set the least significant bit of the link pointer of the pool for raw cell reception to “1” as a flag to recognize the link pointer. Refer to **5.5.2 (3) Pool storing raw cells**.

Remark The link pointer of the batch at the end of the chain may be any address. However, it is recommended, to ensure protection of the other areas for operating the system memory, that the start address of the first batch or the start address of any batch in the can be set.

- Free buffer : This area is used by the μPD98409 to store the actual received data. The size of one buffer can be set from 64 bytes to 64K bytes. The size of all the free buffers in the same pool must be the same, however.

Figure 5-30. Configuration of Pool

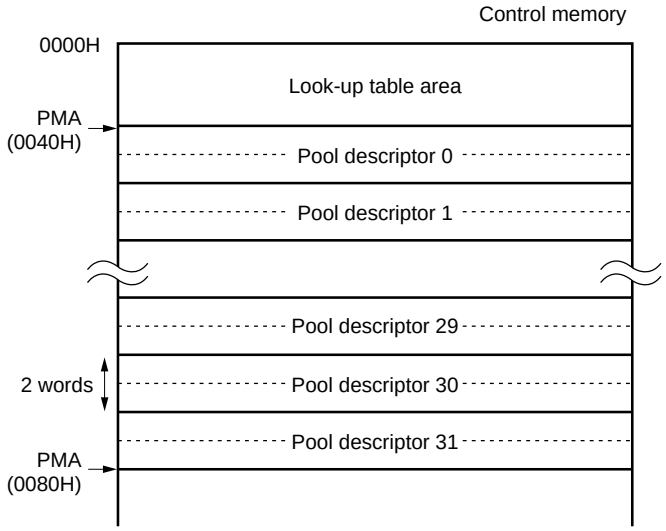


(2) Receive free buffer pool pointer

The maximum number of pools prepared by the host in the system memory is 32. These pools are numbered from 0 to 31. The information on each pool such as address and size is set in a “receive free buffer pool pointer” area by a pool descriptor consisting of 2 words. The “receive free buffer pool pointer” area starts from the PMA address of the control memory, and the μPD98409 sequentially assigns pools 0, 1, 2, and so on, starting from the PMA address. The PMA address of the μPD98409 is fixed to ‘0040H’. For example, the address of the low-order word of the descriptor of pool 3 is ‘0046H’, and the address of the high-order word is ‘0047H’. The μPD98409 has a receive free buffer pool pointer area of 64 words.

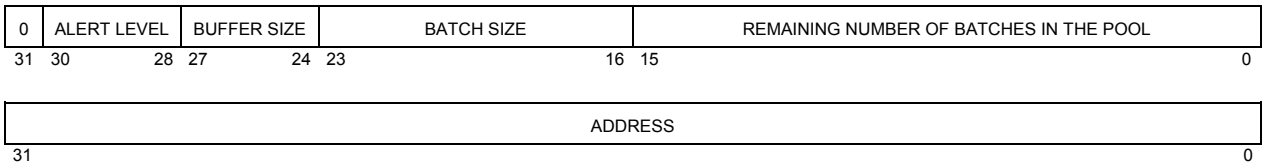
To receive an MPEG-TS packet, it is not necessary to set the receive free buffer pool pointer.

Figure 5-31. Location of Pool Descriptor in Receive Free Buffer Pool Pointer



The format of the pool descriptor is as follows.

Figure 5-32. Format of Pool Descriptor



<1> ALERT LEVEL

This field sets the "alert level" of the number of remaining batches. If the number of batches remaining in the pool is equal to the value set in this field, the μPD98409 sets the corresponding bit of the RQA register and the RQA bit of the GSR register to "1". If not masked, it issues an interrupt to report to the host. This function is valid only for the pool for AAL-5 type reception. This field is meaningless for the pool for raw cell data storage, and the μPD98409 ignores this field.

Setting : When "n" is written to this field, it means that "n x 4" (number of remaining batches: 4, 8, 12 ... 28) is specified. When "000" is written, this function is disabled, and no report is made to the host.

<2> BUFFER SIZE

This field specifies the size of all the buffers located in this pool. The buffer size that can be specified is 64 bytes to 64K bytes.

Setting : When "n" is written to this field, "64×2^n" (64, 128, 256, ... 64K bytes) is specified.
For the pool for raw cell data, n = 0, setting the buffer size to 64 bytes.

<3> BATCH SIZE

This field specifies the number of buffers of one batch located in this pool.

Setting : The end of the batch is always a "batch link pointer". Specify the number of buffers n, excluding the batch link pointer, to this field. The size of the system memory actually used by one batch is n + 1 because the "link pointer" is added to the number of buffers n.
Be sure to set n to 1 or more. In the case of a pool for AAL-5 packet reception, the μPD98409 recognizes the position of the link pointer from this "BATCH SIZE".

<4> REMAINING NUMBER OF BATCHES

The host writes the number of batches n prepared for a pool to this field during initialization. After that, this field is managed by the μPD98409, and indicates the number of batches remaining in the pool.

The μPD98409 decrements the number of batches each time it has fetched a batch from the pool, and increments the number each time it has received the Add_Batches command.

<5> ADDRESS

The host writes the first address of the first batch in a pool to this field during initialization. After that, this field is used by the μPD98409 as a pointer that indicates the next batch.

(3) Pool storing raw cell

The pool numbers that can be stored differ depending on the type of the data that can be received by the μPD98409.

When receiving data of an AAL-5 type other than an MPEG-TS packet, the user can specify any of pools 0 to 31 for each VC. For raw cell data, pools 0 to 7 can be specified. To receive a cell with the pattern of the PTI field being "1xx", however, the μPD98409 unconditionally stores this cell in pool 0. If the user has set the μPD98409 to receive OAM F5 cell/RM cells, therefore, pool 0 must be always selected to store the raw cell data of OAM cells.

Table 5-5. Types of Receive Data and Usable Pool

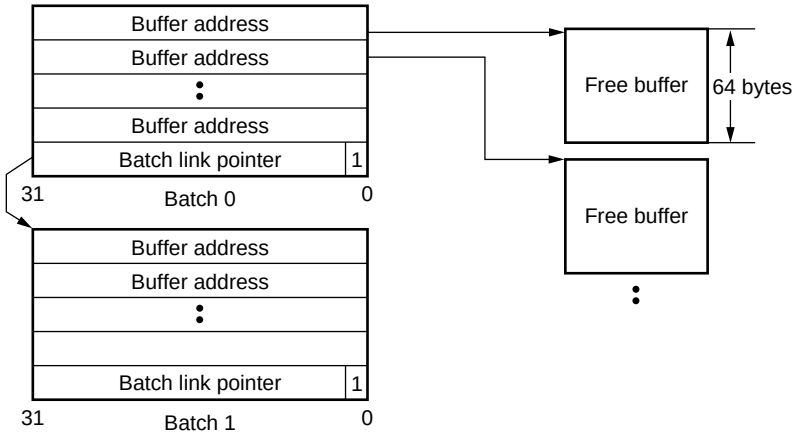
Data Type	Usable Pool
AAL-5 type packet reception	Pool 0 to 31 can be used.
Raw cell data	Pool 0 to 7
OAM F5/RM cell/raw cell data of RM cell	Pool 0 only

The setting of the "link pointer" of the pool specified to store raw cell data differs from the pool to store AAL-5 data. In the case of a raw cell data pool, the μPD98409 monitors the least significant bit of the buffer address read from the batch. If the least significant bit is "1", the μPD98409 recognizes that the address is a link pointer, and moves to the next batch indicated by the address with the least significant bit changed to "0". Therefore, the host must set the least significant bit of the address stored in the link pointer of the batch to "1" only in the case of a pool for storing raw cell data.

When a pool storing raw cells are used, the RQA interrupt function that alerts the number of remaining batches is disabled, and the μPD98409 ignores the setting of the "ALERT LEVEL" field.

The μPD98409 stores one raw cell data consisting of 64 bytes in one free buffer. Therefore, the raw cell data storage pool sets the free buffer size to 64 bytes. If the size is set to more than 64 bytes, the μPD98409 ignores the area.

Figure 5-33. Raw Cell Pool



5.5.3 Receive channel (receive VC)

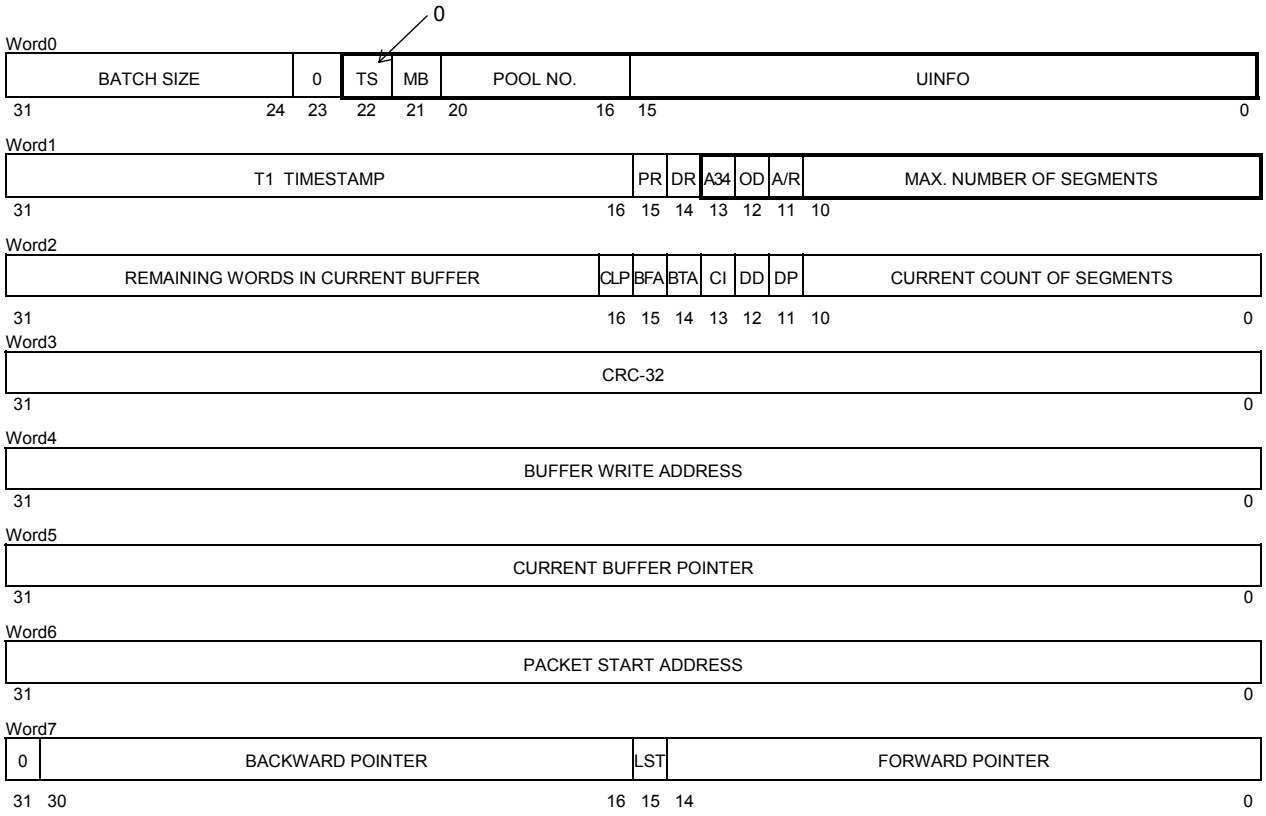
(1) Opening receive channel

The host opens a receive channel by issuing the Open_Channel command. When receiving this command, the μPD98409 allocates a block indicated by TOS (Top Of Stack) from the free block pool of the control memory, and returns its first address to the host by using command indication. The host sets the allocated block as a receive VC table.

(2) Setting of receive VC table

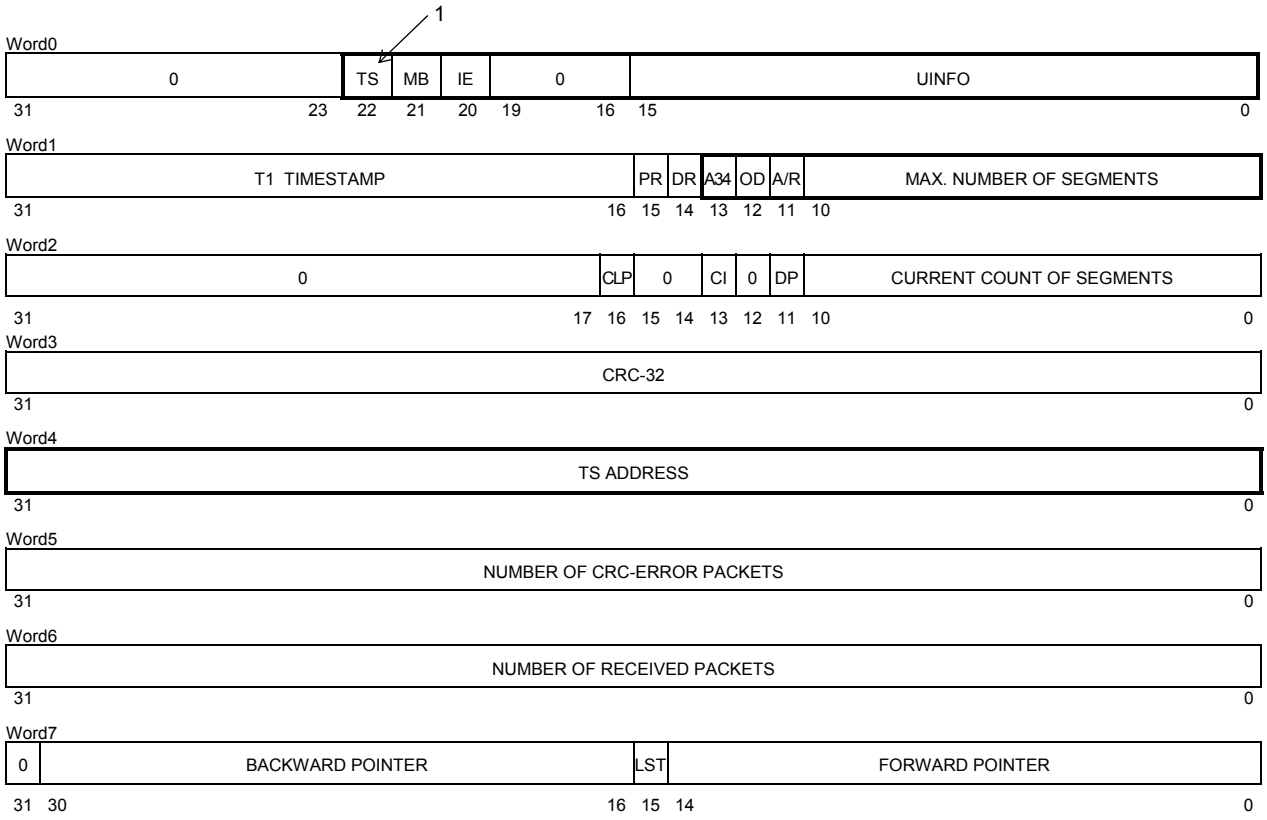
The host initializes a block of 8 words allocated from the free block pool of the control memory as a receive VC table. Figure 5-34 shows the structure of the receive VC table. The host sets initial values to the locations indicated by the solid bold line in this figure, and clears the other areas to 0. The host uses the Indirect_Access command to write data to the VC table. The areas other than those indicated by the solid bold line are used for reception. The host can access the table as necessary, and can use it as status information.

Figure 5-34. Receive VC Table (for reception of packets other than MPEG-TS packets)



Caution Clear the areas of Word0 and Word1 other than those indicated by the solid bold line to 0 during initialization.

Figure 5-35. Receive VC Table (for reception of MPEG-TS packet)



Caution Clear the areas of Word0 and Word1 other than those indicated by the solid bold line to 0 during initialization.

BATCH SIZE : This field indicates the number of free buffers currently remaining in the batch.

TS : MPEG-TS packet reception. Sets whether this VC receives MPEG-TS packets. Only one VC can be set to receive MPEG-TS packets.

- 1: Receives MPEG-TS packets.
- 0: Does not receive MPEG-TS packets.

MB : Mail box. This bit selects a mailbox that stores receive indication to this VC.

- 1: Mailbox 1
- 0: Mailbox 0

POOL NO. : Pool No. One of the 32 pools is selected for this VC (invalid when MPEG-TS packets are received).

IE : Enables MPEG-TS receive indication. When an MPEG-TS packet is received, receive indication is issued only if a packet including an error is received and if the IE bit I “1”. The receive indication is not issued even if the IE bit is “1” if a packet without any errors are received.

- 1: Writes receive indication to the mailbox and issues an interrupt if not masked only when MPEG-TS packets including an error is received. The receive indication is not issued if no receive error occurs.
- 0: Does not issue indication or interrupt when an MPEG-TS packet is received.

UINFO : User information. The user can assign any pattern to this field. The pattern assigned to this field is returned by the μPD98409 with receive indication.

T1 TIMESTAMP : This area is used by the μPD98409 to calculate T1 timer.

PR : Packet reception. This bit is set to “1” while this VC is receiving a packet; otherwise, it is 0.

DR : Drop FIFO. This bit is set to “1” if the cell of the packet is dropped because of overrunning of the receive FIFO. Once a cell has been dropped in an AAL-5 packet, the incoming cells belonging to the packet are dropped.

A34 : This bit is set as follows when this VC receives an AAL-3/4 cell as a raw cell.

- 1: Checks the “ST field” of the receive AAL-3/4 cell. Only if the pattern is “01” or “11”, issues an RCR interrupt; otherwise, does not issue the interrupt.
- 0: Normally receives a raw cell. Each time a cell has been received, an RCR interrupt is issued.

  When this bit is set to 1, the A/R bit must be always reset to 0.

OD : This bit selects whether an OAM cell/RM cell is received or dropped. When a cell having a PTI field pattern of “1XX” is received to this VC, this bit selects whether the cell is received or dropped.

- 1: Ignores and drops the cell (OAM F5 cell/RM cell) with a pattern of PTI = 1XX.
- 0: Receives the cell (OAM F5 cell/RM cell) with a pattern of PTI = 1XX as a raw cell.

A/R : This bit selects whether the cell received to this VC is processed as an AAL-5 cell or raw cell.

- 1: Receives as an AAL-5 cell.
- 0: Receives as a raw cell.

MAX. NO. OF SEGMENTS : Maximum number of segments in one packet. This field sets the maximum number of segments of a packet received by this VC. If the last cell of the received packet is not received despite the fact that the current number of segments of the packet has reached the number of segments specified by this field, a receive indication including an error status is immediately issued.

REMAINING WORDS IN CURRENT BUFFER : Number of words in vacant area remaining in the current buffer. (invalid when MPEG-TS packets are received).

CLP : CLP = 1 reception. This bit is set to “1” if even one cell with header CLP = 1 is received in the packet being received.

BFA : This bit is set to “1” if there is a free buffer allocated to this VC. (invalid when MPEG-TS packets are received).

BTA : This bit is set to “1” if there is a batch allocated to this VC. (invalid when MPEG-TS packets are received).

CI : Congestion indication. This bit is set to “1” if even one cell with a PTI field pattern of “01X” that indicates congestion is received in the packet being received.

DD : DMA drop. This bit is set to “1” if the cell is dropped because no vacant free buffer is allocated. If a cell is dropped while an AAL-5 packet is received, all the incoming cells belonging to the packet are dropped. (invalid when MPEG-TS packets are received).

DP : Packet reception in progress. This bit is set to “1” while a packet is being received; otherwise, it is “0”.

CURRENT COUNT OF SEGMENTS : This field indicates the number of segments received so far from the packet being received.

CRC-32 : Temporary buffer used by the μPD98409 to perform CRC-32 operation on the packet.

BUFFER WRITE ADDRESS : Address of the currently allocated buffer to which data is to be stored next. (invalid when MPEG-TS packets are received).

TS-ADDRESS	:	First address of the DMA transfer destination of the MPEG-TS packet. Be sure to set this address at a word boundary (32 bits).
CURRENT BUFFER POINTER	:	First address of the free buffer to be allocated next. (invalid when MPEG-TS packets are received).
NUMBER OF CRC-ERROR PACKETS	:	32-bit free running counter that indicates the total number of packets that have been received so far with CRC errors since the start of MPEG-TS packet reception.
PACKET START ADDRESS	:	Start address of packet. The first address of the batch that is allocated first (invalid when MPEG-TS packets are received).
NUMBER OF RECEIVED PACKETS	:	32-bit free running counter that indicates the total number of packets that have been received so far since the start of MPEG-TS packet reception.
BACKWARD POINTER	:	"VC NUMBER" of VC linked to the T1 list before this VC.
LST	:	This bit is set to "1" if the VC is the one linked to the end of the T1 list.
FORWARD POINTER	:	"VC NUMBER" of VC linked to the T1 list next to this VC.

(3) Status transition of receive channel

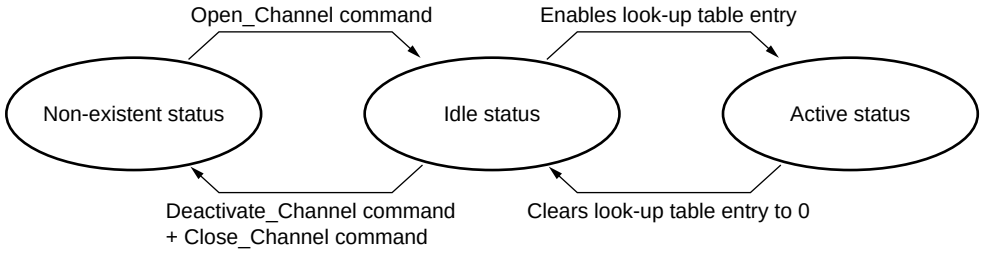
The receive channel may be in three statuses: non-existent, idle, and active. The host opens a channel by issuing the Open_Channel command. On receiving this command, the μPD98409 reports the address of the VC table from the free block pool of the control memory to the host as a command indication. Next, the host writes appropriate parameters (such as pool number used, AAL-5 processing/MPEG-TS/raw cell processing, and maximum number of segments enabled) to the block allocated by using the Indirect_Access command. As a result, this block is used as a receive VC table, and the channel enters the idle status.

To make the channel active, set the look-up table to set the enable bit. For an explanation of how to set the look-up table, refer to **5.5.4 Setting of receive look-up table**. As a result, the channel enters the active status. When a cell having the corresponding VPI/VCI is received from the PHY device, the μPD98409 starts reception processing. The channel remains in the active status as long as the enable bit of the look-up table entry is set to "1", but returns to the idle status again when the enable bit of the look-up table entry is disabled.

To terminate this channel, the host clears the entry in the look-up table to 0 to set the channel in the idle status. The host then issues the Deactivate_Channel command followed by the Close_Channel command. Then the VC table used is returned to the free block pool, and the channel enters the non-existent status because it no longer exists.

Before issuing the Deactivate_Channel command and Close_Channel command, be sure to wait for the duration of 48 clocks or more after clearing the entry in the look-up table to 0. Create this timing by issuing the NOP command two times.

Figure 5-36. Receive Channel Status



5.5.4 Setting of receive look-up table

(1) Look-up table

The look-up table is in the control memory and is used to map VPI/VCI to be received. The look-up table always exists in address 000H to 01FH of the control memory, and its size is determined by the setting of the VRR register to be received and the pattern of VPI/VCI to be received.

The μPD98409 internally converts the 24 bits of VPI/VCI included in a receive cell into a 15-bit logic code. This conversion is made according to the setting of the "SHIFT" and "MASK" fields of the VRR register. Based on the converted logic code, whether the cell of the VPI/VCI is received or not is set in the receive look-up table entry. This is done in the following procedure.

<1> VPI is shifted toward VCI by the number set in the "SHIFT" field of the VRR register. At this time, VPI can be shifted by a width of up to 15 bits. If the VPI is shifted by 8 bits or more, 0 is added to the high-order bits.

<2> The 15 bits created in<1>and the contents of the "MASK" field of the VRR register are ANDed.

<3> The 15-bit value resulting from ANDing (VPI/VCI reduction value) is the value to be assigned to the receive lookup table.

<4> The address of the receive lookup table to which the above VPI/VCI reduction value is assigned is determined by "VC NUMBER" of the receive VC (VC table) that receives a cell having the VPI/VCI value.

<5> The value excluding the most significant bit of "VC NUMBER" (value: "1") and the least significant bit is an address that sets the VPI/VCI reduction value. If the excluded least significant bit is "0", the receive lookup table is set to the low-order 16 bits of that address; if it is "1", the table is set to the high-order 16 bits.

<6> The enable bit of the same field ("ENBL" bit) is set to "1".
With this setting, the receive lookup table can be set.

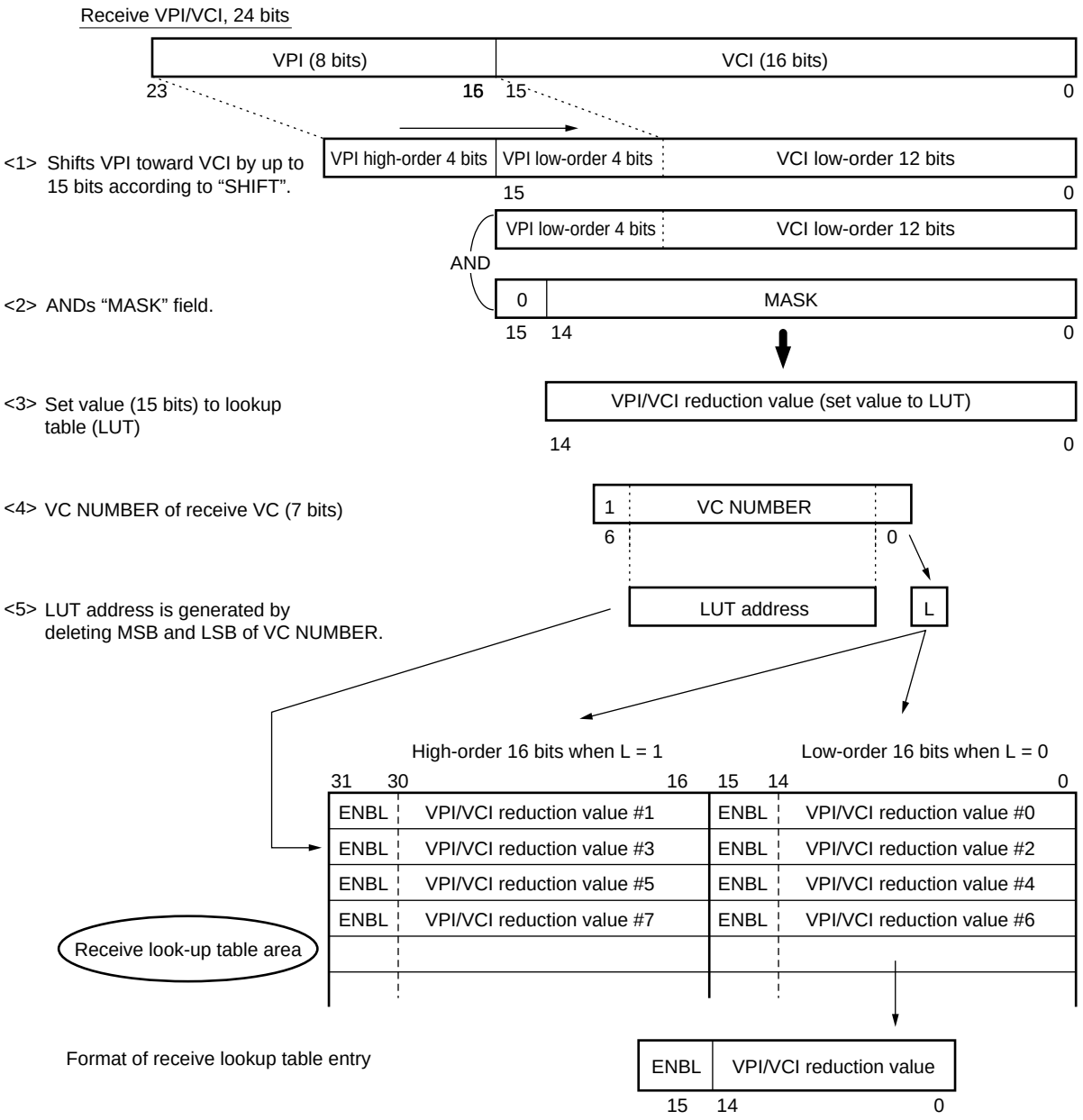
Caution "VC NUMBER" is not a physical address that indicates the beginning of the VC table in the free block pool, but a value derived from it by excluding the low-order 3 bits ("000").

Reference) Reception operation of μPD98409

When the μPD98409 receives a cell from a PHY device, it reduces the VPI/VCI in the header of the received cell to 15 bits by the same method as above, according to the setting of the VRR register. Then the μPD98409 searches for the VPI/VCI reduction value from the receive lookup table (LUT). If the VPI/VCI reduction value is found and the enable bit is "1" as a result of the search, the cell is received. At this time, the address of the receive VC table that is mapped is calculated from the address (LUT address, 5 bits) in which the VPI/VCI reduction value is stored. If the VPI/VCI reduction value is stored in the high-order 16 bits of LUT, "1" is appended to the low-order bit of the LUT address; if the VPI/VCI reduction value is stored in the low-order 16 bits of LUT, "0" is appended. In addition, 1 bit is appended in the high-order bit of the LUT address (value: "1"). Moreover, 3 bits of "000" are appended to the low-order bits. In this way, a 10-bit address for the VC table is generated. The μPD98409 accesses the VC table mapped to this address and checks or updates the settings for reception processing.

Remark When the same VPI/VCI reduction value is stored in LUT, the μPD98409 selects the one at the lower address after it is found.

Figure 5-37. Receive Lookup Table (where SHIFT = 4)



ENBL bit: 1 - Enable. Enables reception by making VC active.
0 - Disable. Does not receive cell of the VC.
VPI/VCI reduction value: 24-bit VPI/VCI value reduced to 15 bits by “SHIFT” and “MASK”.

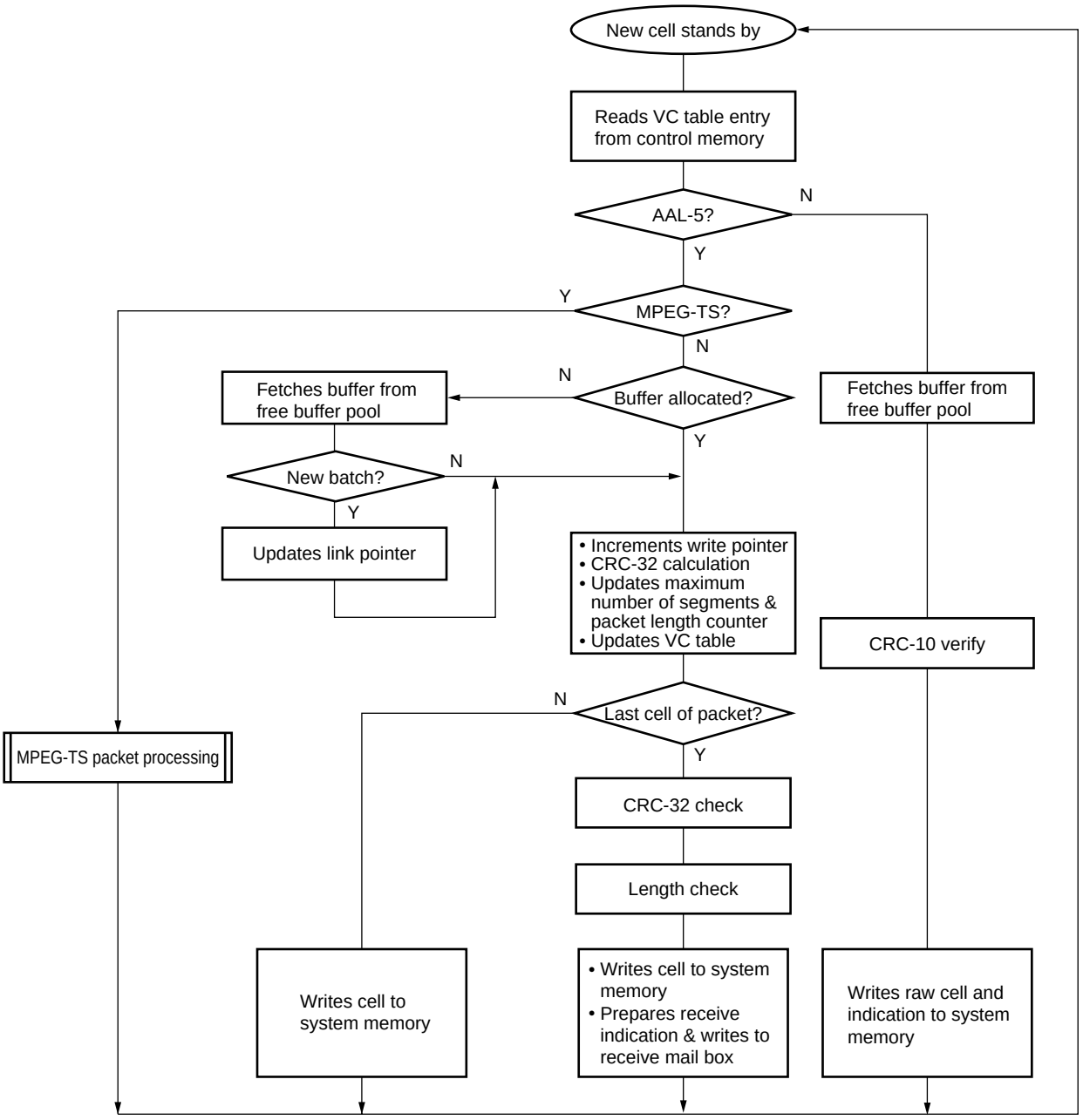
5.5.5 Receive operation

(1) Receive operation

- The μPD98409 receives a cell from the PHY device via the UTOPIA interface, and stores the cell in the receive FIFO. The receive FIFO has a capacity of 23 cells. If the VPI/VCI pattern at the header of the received cell is all 0, the cell is recognized as invalid, and it is not stored in the receive FIFO but dropped.
- If the cell is not an unassigned cell, an address indicating the look-up table of the control memory is generated from the 24-bit pattern of VPI/VCI according to the "SHIFT" and "MASK" fields of the VRR register.
- The μPD98409 reads the entry in the look-up table indicated by the generated address. If the enable bit (ENBL bit) is "1", it stores the cell in the receive FIFO and continues the processing. Any cell not mapped to the look-up table (ENBL bit = "0") is dropped.
- If the ENBL bit = 1, the μPD98409 reads the VC table of the free block pool from "VC NUMBER" stored in the look-up table, and obtains information indicating which raw cell processing is to be performed. When an MPEG-TS packet is received, MPEG-TS packet processing is performed. Otherwise, the pool number of the system memory and address information are read from the VC table, and the following processing is performed:
- If the cell is the first cell of a packet and is not assigned a batch of the receive pool, or if the current batch is used up, the μPD98409 fetches a new batch from the "ADDRESS" field of the pool descriptor in the receive free buffer pool pointer area of the control memory.

  If a new batch is fetched, the value of the "REMAINING NUMBER OF BATCHES IN THE POOL" field of the pool descriptor is decremented by one, so that the "ADDRESS" field indicates the first address of the next batch.
- μPD98409 transfers a segment (payload of 48 bytes of the receive cell) to the first buffer of the system memory by means of DMA. The free buffer address of the fetched buffer is stored to the VC table, and is updated each time the μPD98409 has transferred a segment.
- If the cell is the first cell of the packet, the T1 time stamp is stored to the VC table, and the VC table is added to the T1 link list.
- The μPD98409 transfers a segment (payload data) to the system memory each time it has received a cell of the VC, and updates the free buffer address. It also calculates CRC-32 and packet length for each segment, and updates the intermediate result to the VC table.
- If the free buffer used for the VC becomes full before the last cell of the packet is received, the μPD98409 fetches the address of a new free buffer from the batch of the system memory. When the batch is used up, a new batch is fetched from the pool descriptor.
- If one packet straddles two or more batches, the μPD98409 overwrites the link pointer of the batch used, to change the chain.
- If the μPD98409 receives a packet with the LSB bit of the PTI field of the cell header being 1, it recognizes the cell as the last cell, and compares the result of calculation of CRC-32 and the number of cell counts with the "CRC-32" and "Length" of the AAL-5 trailer included in the last cell.
- Next, the receive indication is stored in the mailbox specified by the VC table specified by the host. If an error occurs, that status information is included. If not masked, an interrupt is also generated.

Figure 5-38. Receive Operation Flowchart

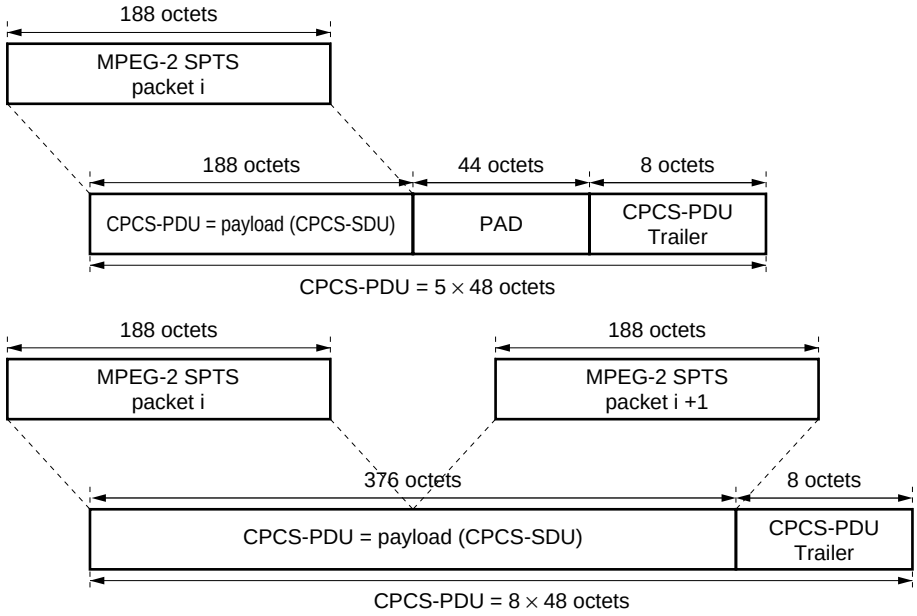


MPEG-TS packet processing

The format of the MPEG-TS packet the μPD98409 receives is as follows. The basic unit is a 188-byte MPEG-2 SPTS packet. An AAL-5 Packet that includes an MPEG-2 SPTS packet in its payload is divided into five cells, and an AAL-5 packet including two MPEG-2 SPTS packets is divided into eight cells. The μPD98409 can correctly execute the following MPEG-TS processing even for an AAL-5 packet including three or more MPEG-2 SPTS packets.

- Deletes the AAL-5 trailer and padding from the MPEG-TS packet and transfers just the payload to the MPEG-TS processing LSI.
- If the packet is invalid as an MPEG-TS packet including a dropped cell, generates dummy data and always writes data of integer multiple of 188 bytes to the MPEG-TS processing LSI.

Figure 5-39. Format of MPEG-TS Packet



MPEG-TS packet processing is performed as follows:

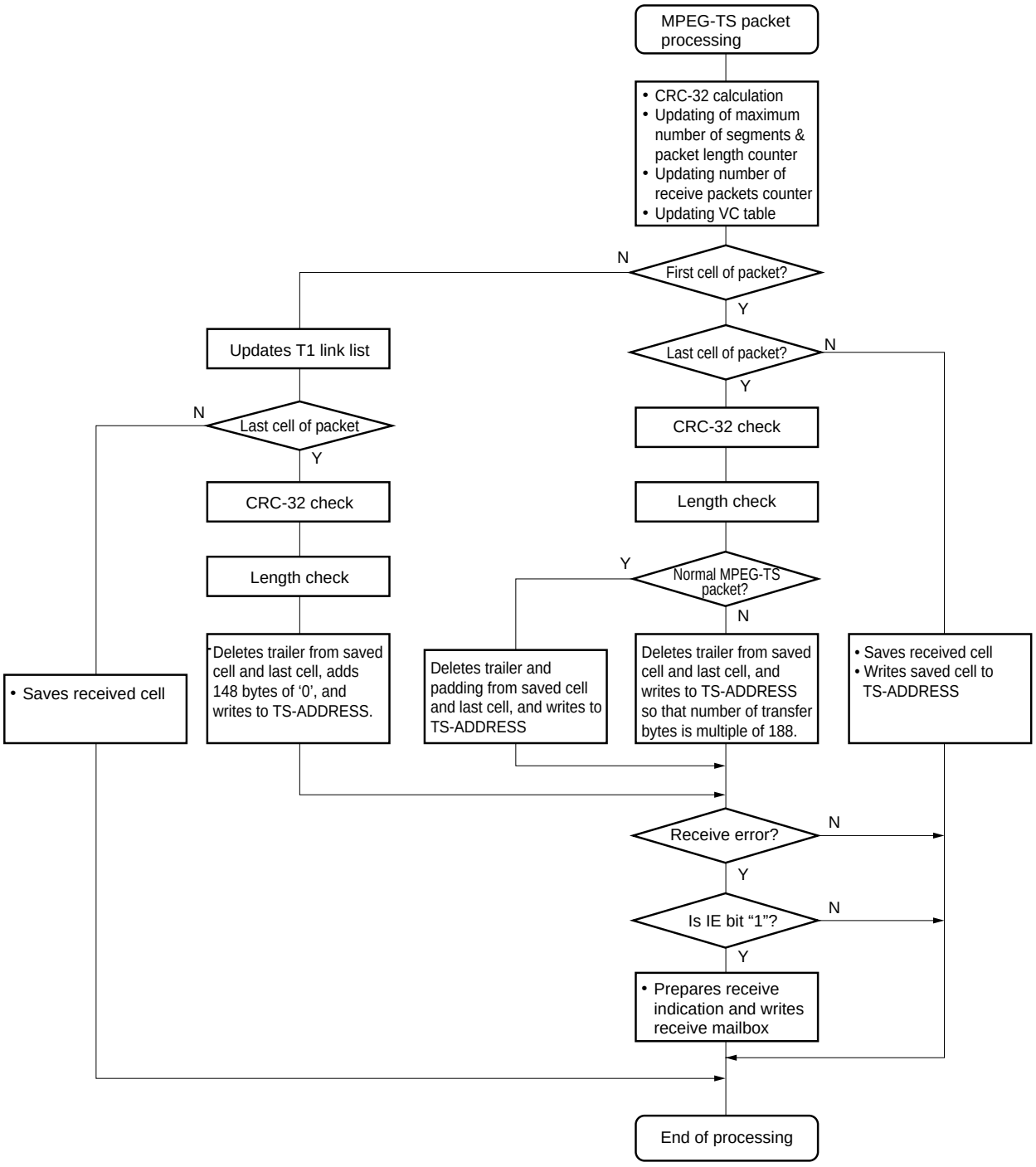
- Each time the μPD98409 has received a VC cell, it calculates CRC-32 and packet length for each segment (payload) and updates the intermediate valves in the VC table. It also updates the maximum number of segments counter, packet length counter, and number of received packets counter.
- If the cell is the start cell of the packet, the T1 time stamp is stored in the VC table, and that VC table is added to the T1 link list.
- If a packet in which the LSB of the PTI field of the cell header of the first cell is “0” is received, it is recognized that that packet has subsequent cells. The segment (48 byte payload of received cell) is saved to the internal buffer, and MPEG-TS processing is completed.
- The calculated CRC-32 and the cell count are compared with the “CRC-32” and “Length” valves of the AAL-5 trailer in the last cell.
- If a packet in which the LSB of the PTI field of the cell header of the first cell is “1” is received, it is recognized that the packet has only one cell. The μPD98409 updates the receive packet counter of the VC table, and compares the calculated CRC-32 and the cell count with the “CRC-32” and “Length” valves in the AAL-5 trailer in the last cell. If a CRC error occurs, the CRC error packet counter in the VC table is updated.
- Next, TS-ADDRESS is output to the PCI bus as an address, the 8-byte trailer is deleted from the segment (48 bytes of the payload of the received cell) as transfer data, and 188 bytes of data to which 148 bytes of “0s” have been added is output, and a master write operation is performed (47-word burst transfer).
- If a cell that is not the first cell of the packet where the LSB of the PTI field of the cell header is “0” is received, it is recognized that the packet has subsequent cells, a master write operation (12-word burst transfer) is executed to the address TS-ADDRESS using the previously received segment that has been saved in the internal buffer as transfer data, and the MPEG-TS packet processing is completed.
- If a cell that is not the first cell of the packet where the LSB of the PTI field of the cell header is “1” is received, it is recognized as the last cell. The μPD98409 updates the receive packet counter of the VC table and compares the calculated CRC-32 and the cell count with the “CRC-32” and “Length” values in the AAL-5 trailer in the last cell. If a CRC error occurs, the CRC error packet counter in the VC table is updated.

- Next, the validity of the packet as an MPEG-TS packet is checked. If the packet is valid as an MPEG-TS packet, TS-ADDRESS and the previously received segment saved in the internal buffer as transfer data plus the segment received this time apart from trailer and padding are output to the PCI bus as an address, and a master write operation is performed (11- to 22-word burst transfer).
- If the packet is not valid as an MPEG-TS packet, TS-ADDRESS and the previously received segment saved in the internal buffer as transfer data plus the segment received this time apart from trailer and padding are output to the PCI bus as an address, and a master write operation is performed (22-word burst transfer). As a result of this transfer, if the number of transfer bytes of the entire receive packet is not an multiple of 188, dummy data "0" is written to address TS-ADDRESS, until the number of transfer bytes is a multiple of 188 (1- to 46-word burst transfer).
- When the last cell has been written, it is checked whether a receive error has occurred. This check is independent of checking the validity of MPEG-TS packets, and the same as the receive processing of an AAL-5 packet other than MPEG-TS packet. If a receive error occurs, and if issuing of the receive indication is enabled by the IE bit of the VC table, the receive indication is stored in the mailbox specified by the VC table. If an error occurs, status information is included. If not masked, an interrupt occurs, and MPEG-TS packet processing is completed. If a receive error does not occur, of if issuing of the receive indication is disabled by the IE bit of the VC table, the receive indication is not issued, and the MPEG-TS packet processing is completed.

When the μPD98409 receives the last cell of the MPEG-TS packet, the validity of the MPEG-TS packet is checked based on the following criteria:

- Length error occurs as AAL-5 packet.
  (The number of words of padding calculated from the Length field and the number of received cells of the AAL-5 trailer is greater than 55 bytes, or less than 8 bytes).
- The Length field of the AAL-5 trailer is not a multiple of 188.
- $25 \leq \{(\text{Number of received cells} - 2) \times 12\} \bmod 47 \leq 36$

Figure 5-40. MPEG-TS Packet Processing Flowchart



(2) Storing receive data

This section explains the procedure in which the μPD98409 stores the receive data in the system memory, and the procedure of the host operation. The sequence in which the μPD98409 pool is used differs depending on whether a pool for AAL-5 packets other than MPEG-TS or a pool for raw cell data storage is used.

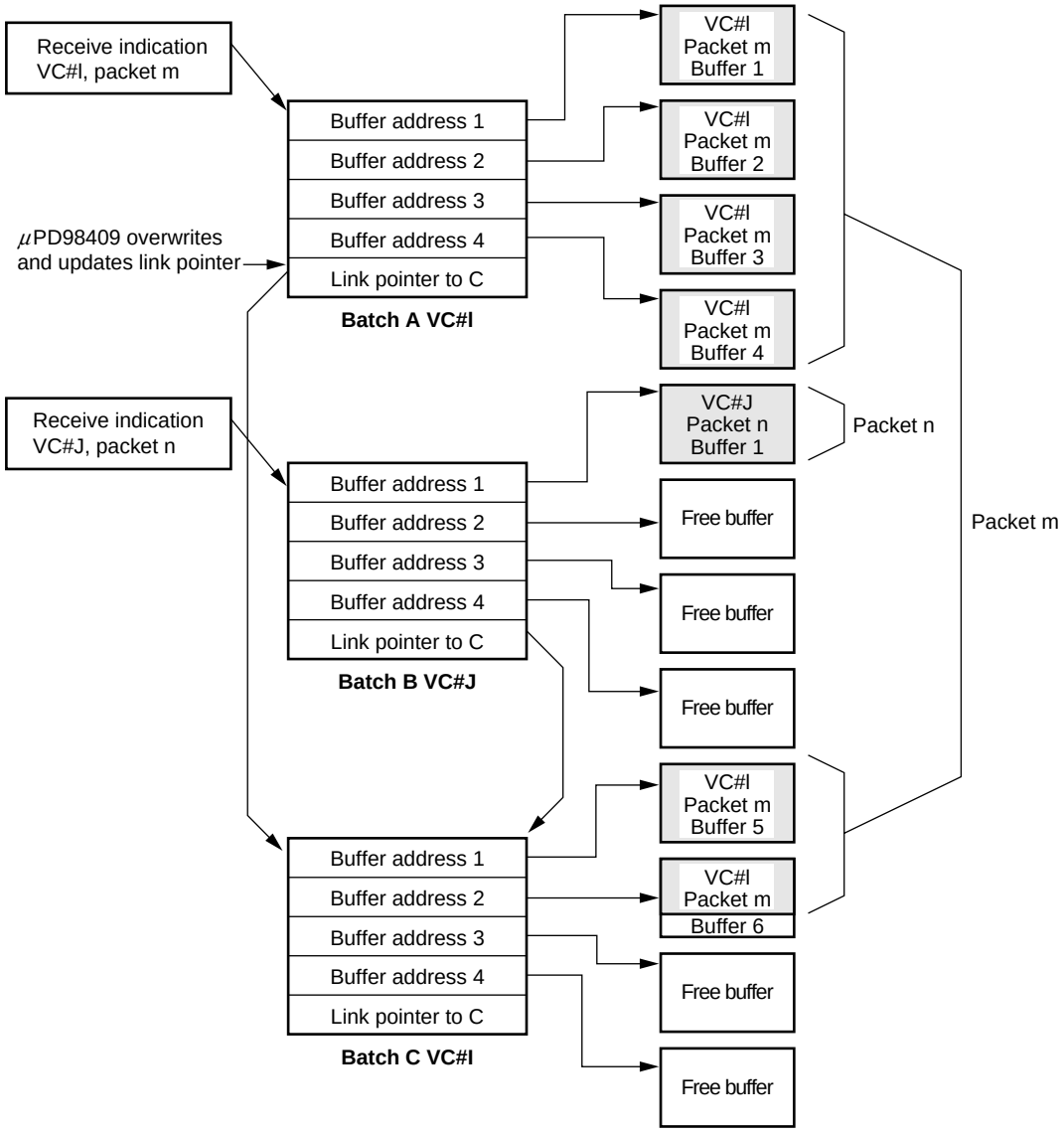
(a) Pool for AAL-5 packet (other than MPEG-TS)

The user can store each VC in one of the 32 pools by setting VC table "POOL NO.". One pool can store the receive packets of two or more VCs. For AAL-5 traffic, the VC fetches a free buffer in batch units. Therefore, each packet always begins with a new batch. The VC uses two or more free buffers to store a received packet. All the free buffers of one batch may be used, or the free buffers may be used straddling two or more batches. If packets are stored straddling multiple batches, the μPD98409 overwrites the link pointers of the batches by DMA, and updates the link information.

An example of reception operation of AAL-5 traffic is shown below.

The pool shown in Figure 5-38 has batches A, B, C, and so on. One batch has four free buffers. Both of the two VCs, #i and #j, are stored in the same pool.

Figure 5-41. Configuration Example of Receive Data



- The first cell of packet m of VC#I is received. The μPD98409 reads the pool descriptor to fetch a batch. At this time, because the first address of batch A is stored to the pool descriptor, batch A is allocated to packet m of VC#I.
- Because batch A is full, the μPD98409 updates the pool descriptor so that it indicates the beginning of batch B.
- While packet m is being received, the first cell of packet n of VC#J is received. The μPD98409 allocates batch B to packet n.
- VC#I makes the four free buffers of batch A full before the packet is received to the end, and allocates new batch C. At this time, the μPD98409 overwrites, by DMA, the link pointer of batch A that previously indicated batch B so that the pointer indicates the beginning of batch C, and updates the link information.
- To store all the packets of VC#I, two free buffers are necessary from batch C. When all the data of packet m have been stored to the free buffers, the μPD98409 creates the receive indication for packet m. In this indication, the first address of batch A as the start address of the packet, and packet size in cell units are stored. The host updates this information and processes the data received from the link pointer.
- Packet n requires only one free buffer of batch B. The first address and size of batch B are returned to the receive indication that is issued when all packet n has been received.

When the pool for AAL-5 traffic storage is used, an “alert level (ALERT LEVEL)” is assigned to the pool descriptor. When the number of remaining batches reaches this alert level, an interrupt signal can be issued to the μPD98409. The μPD98409 decrements the “REMAINING NUMBER OF BATCHES IN THE POOL” and compares it with the ALERT LEVEL each time it has used a batch. If these two are equal, the μPD98409 sets the corresponding bit of the RQA register to 1, then sets the RQA bit of the GSR register, to issue an interrupt to the host. In response, the host adds a new batch by issuing the Add_Batches command to the pool. The function of ALERT LEVEL is valid only for the pool for AAL-5 traffic.

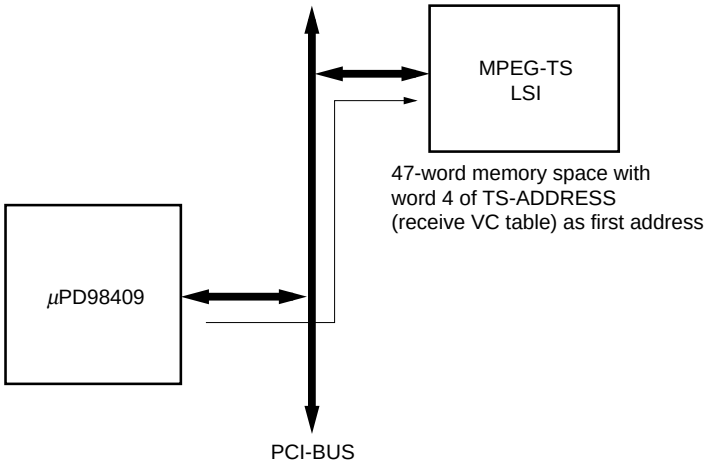
If “REMAINING NUMBER OF BATCHES IN THE POOL” of a specified pool has reached to 0 when a new cell is received and is about to be transferred to the specified pool, the receive queue underruns. The μPD98409 sets the corresponding bit of the RQU register to 1, sets the RQU bit of the GSR register, and issues an interrupt if not masked.

The initial information such as the number of batches of the receive pool and address is directly written to the pool descriptor of the control memory by the host by using the Indirect_Access command. After that, the host adds batches to the pool by using the Add_Batches command. On receiving the Add_Batches command, the μPD98409 updates the contents of the pool descriptor. Note that the host does not directly update the value of the pool descriptor of the control memory.

(b) Transfer of MPEG-TS packet

The MPEG-TS packet is directly written to the MPEG-TS processing LSI connected to the PCI bus by means of PCI memory write operation. The burst size is automatically determined by the μPD98409 in a range of 1 to 47 words. TS-ADDRESS (word 4 of the receive VC table) is the writing start address. Because the packet is written by means of memory write operation, if the MPEG-TS processing LSI is disconnected during transfer, the address output by the μPD98409 to the PCI bus on resumption of transfer is not the TS-ADDRESS address, but a value of the number of words written before disconnection plus TS-ADDRESS. Therefore, the MPEG-TS processing LSI must be of configuration so that input data written to any address of the 47-word space with TS-ADDRESS as the first address can be received.

Figure 5-42. Transfer Address of MPEG-TS Packet



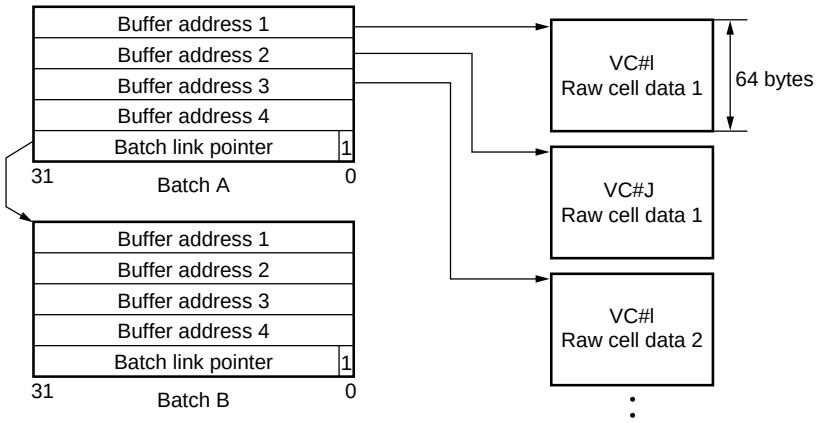
(c) Pool storing raw cell data

The pool for Raw cell data storage does not allocate batches to each packet, unlike the pool for AAL-5 traffic. It allocates one free buffer to one raw cell data.

When the μPD98409 has received cells, it sequentially stores raw cell data, starting from the first free buffer of the first batch of a pool. The μPD98409 does not issue a receive indication when it has received raw cell data. Instead, the μPD98409 sets the bits corresponding to the pool of the RCR7 through RCR0 bits of the GSR register to 1, and issues an interrupt if not masked, each time it has stored one unit of raw cell data (however, if the A34 bit is set to 1, the interrupt is issued only when the last cell of the AAL-3/4 packet has been received, instead of each time the raw cell data has been stored).

Because the μPD98409 does not report the address of the free buffer to which the data has been stored to the host, the host must record the address of the free buffer whose processing has been previously completed. One pool can be set to store the raw cell data of two or more VCs. In this case, the cells are stored in the free buffer in the order in which they have arrived, regardless of the differences of the VCs. In the pool for AAL-5 packet storage, the “ADDRESS” field of the pool descriptor is updated to the start address of the batch to be stored next by the μPD98409 and used. The raw cell data storage pool is used to store an address indicating the free buffer to be stored next. When the address of a free buffer is fetched from a batch, and if it is detected that the least significant bit is 1, the μPD98409 recognizes this address as a link pointer, and proceeds to the next batch.

Figure 5-43. Structure of Raw Cell Data Storage Pool
(if raw cell data of VC#1 and VC#J are to be stored to the same pool)



The raw cell data storage pool does not have a function to monitor the ALERT LEVEL that indicates the number of remaining batches. Only the RQU interrupt that is triggered by an underflow in the transmit queue is valid.

For an explanation of raw cell data, refer to **3.3.3 Support of non-AAL-5 traffic**. Note that an AAL-5 packet cannot be stored in a pool used to store raw cell data. Also note that both the data must not exist together in the same pool.

(3) T1 timer (reassembly timer)

The μPD98409 has a function to specify the time required for one packet to arrive, by using a hardware watchdog timer. This function is called the T1 timer (reassemble timer) function. The user sets the permissible time from the arrival of the first cell to the arrival of the last cell in the register. The μPD98409 monitors whether the packet being received exceeds this time. If the time is exceeded, the μPD98409 stops reception of the packet, and reports this to the host as a T1 error by using a receive indication.

The μPD98409 uses the following two registers to implement this T1 timer function.

- TSR register: This is a 32-bit counter that continuously counts up in the cycle of the system clock. The μPD98409 uses the value of this counter as the “current time”. After reset, this counter immediately starts counting up.
- T1R register: This register sets the permissible time (T1 time) for the host to receive one packet. Of the 32-bit value of this register, only the high-order 16 bits are set. The unit is the cycle of the system clock (refer to **TSR register** above). The low-order 16 bits are all 0 (0000H), and they are not used.

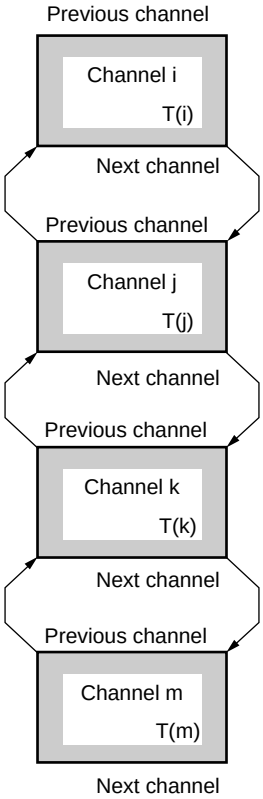
Note that the same set value of T1R may differ from the actual time depending on the frequency of the system clock at which the μPD98409 operates.

The μPD98409 monitors the permissible time by a link list method which is formed by using the “FORWARD POINTER” and “BACKWARD POINTER” at the seventh word in the VC table. These pointers store the “VC NUMBER” of a VC. Each time the first cell of a new packet has arrived, the μPD98409 writes the current contents of the TSR register to the “T1 TIME STAMP” field of the VC table. It also corrects the “FORWARD POINTER” and “BACKWARD POINTER” of the VC table, and the pointers of the VCs previously received, and then adds the VC table to the end of the link list. In other words, the link list always starts with the VC that started reception earliest and ends with the VC that started the reception last. Therefore, the μPD98409 only has to check the “content of the TSR register ≥ T (“T1 TIME STAMP”) of the VC at the beginning of the link list, i.e., the VC that has started reception earliest. When the last cell of the packet has arrived, the μPD98409 releases the VC table of the VC from the link list.

During normal operation, the T1 time is set so that the packet can be correctly received.

Figure 5-44. Reassembly Timer (T1)

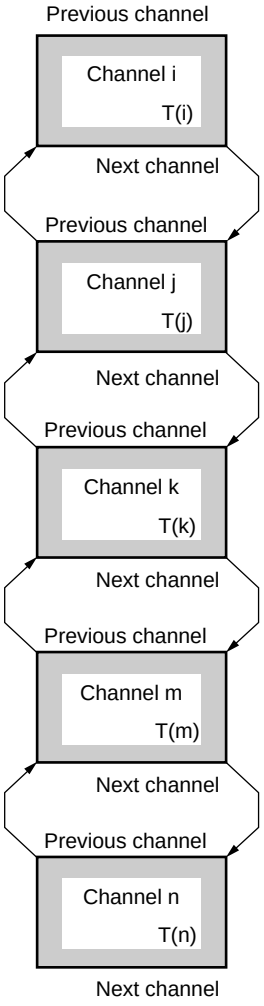
(a) Current status of T1 list



(b) Starts PDU with channel n

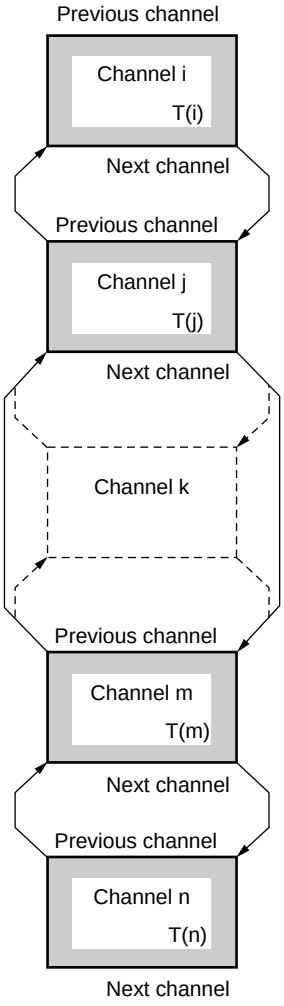
Adds channel n to list

T(n) = current time + T1 value



(c) Ends PDU with channel k

Removes channel k from list



(4) AAL-5 packet reception error detection

The μPD98409 checks for assembly errors in a packet from the AAL-5 trailer information during reception of the packet or on completion of reception. If it detects an error, the μPD98409 reports to the host the type of error, and the start address and size of the data that has been transferred to the system memory up to when the error occurred, by using the reception indication stored in the mailbox.

MPEG-TS receive VC reports an error by using receive indication only when the IE bit of the VC table is set to "1" in case of an error.

The host executes appropriate processing and drops the packet responsible for the error when it has received the receive indication including an error status.

The errors reported by the receive indication are as follows:

(a) Free buffer undeflow
(b) Receive FIFO overrun
(c) "MAX. NUMBER OF SEGMENTS" violation
(d) CRC-32 errir
(e) User abort
(f) "Length" error
(g) T1 time-out
(h) Execution of Deactivate_Channel command

Remark An error is not reported when the validity of the packet as an MPEG-TS packet is checked.

(a) Free buffer underflow

When the μPD98409 receives a cell while it is receiving an AAL-5 packet, if a vacant area of 48 bytes is not available from the free buffer when the μPD98409 tries to transfer the segment of the cell to the system memory, the μPD98409 drops the cell. If the dropped cell is an intermediate or last cell of the packet, the μPD98409 stops receiving the packet, and issues receive indication that reports a free buffer underflow error. Because the size of the free buffer is always an integer multiple of 64 bytes, the free buffer underflow occurs if 16 bytes of the 48-byte data of one segment have been stored. If a free buffer underflow occurs, an RQU interrupt in that pool also occurs (refer to **7.2 (4) RQU**).

If the dropped cell is the first cell of an AAL-5 packet, the receive indication of free buffer underflow is not issued. Only the RQU interrupt, which reports that no free buffer is available from the pool, occurs.

The host replenishes the pool that has generated the RQU interrupt with batches by using the Add_Batches command. If the remaining cells of the packet that caused the free buffer underflow arrive even after the pool has been replenished with the batches, these cells, including the last cell, are dropped.

(b) Receive FIFO overrun

If the receive FIFO having a capacity of 10 cells is full when a cell in the middle of a packet is to be received while an AAL-5 packet is being received in the DROP mode (GMR register: DR bit =0), the μPD98409 drops the cell, and the receive FIFO overruns. Once the cell has been dropped because of the occurrence of receive FIFO overrun while a packet is being received, the remaining cells belonging to the packet, including the last cell, are dropped when they have arrived. If the μPD98409 has exited from the FIFO overrun status when the last cell of the dropped packet has arrived, it issues the receive indication to report the occurrence of the overrun of the packet.

If the FIFO overrun status still persists when the last cell has arrived, the next packet is also dropped because the boundary with the next packet cannot be detected.

In the No DROP mode (GMR register: DR bit = 1), the FIFO overrun does not occur because the μPD98409 stops transfer of receive data to the PHY device when the FIFO has become full.

(c) "MAX. NUMBER OF SEGMENTS" violation

This error occurs if the last cell of the packet is not received even when the number of received cells has reached the "MAX. NUMBER OF SEGMENTS" specified by the user. The data up to the "MAX. NUMBER OF SEGMENTS" set by the user to the VC table are stored in the system memory. The receive indication is issued when the cell next to the one that has exceeded the "MAX. NUMBER OF SEGMENTS" has been received. After that, the cells belonging to the packet that is responsible for this error, including the last cell, are dropped.

If the user sets the "MAX. NUMBER OF SEGMENTS" to "100", for example, the last cell must be received at the 100th cell position at the latest. If the last cell is not received at the 100th cell position, the receive indication of the "MAX. NUMBER OF SEGMENTS" error is issued as soon as the 101st cell has been received. After that, the 101st cell and those that follow are dropped by the μPD98409, until the last cell is received.

(d) CRC-32 error

This error is reported if the result of calculation of CRC-32 and the value of the "CRC-32" field included in the receive trailer do not coincide with each other after all the data of the packet have been transferred to the system memory.

Because the CRC-32 error is added to the AAL-5 trailer that is received at the end of a packet, the μPD98409 detects the error after it has stored all the receive data of the packet to the receive buffer. Therefore, all the data is stored in the receive buffer even if the packet caused the CRC error.

If it has been detected that a packet having both a CRC-32 error and "Length" error" has been received as a result of checking an AAL-5 trailer, the CRC-32 error is reported as the error status of the receive indication.

(e) User abort

This error is reported if the value of "Length" field included in the receive trailer is found to be 0 after all the data of a packet have been transferred to the system memory. Usually, the received packet is dropped by the host as an invalid packet in this case.

(f) "Length" error

This error is reported if it is found that the following conditions are satisfied as a result of checking the "Length" field included in the calculated packet length and receive trailer, after all the data of a packet have been transferred to the system memory.

- ("Number of receive cells x 48 bytes" – ""Length value" in trailer") > 55 bytes
- ("Number of receive cells x 48 bytes" – ""Length value" in trailer") < 8 bytes

(g) T1 time-out

This error occurs if the last cell has not been received even after the user-set T1R time has elapsed after the first cell of a receive packet was received. Until the T1R time elapses, the receive data is stored in the system memory, and the start address and size of the packet are reported by the same receive indication.

The remaining cells in the packet that has caused the T1 error are received as a new packet. As a result, a Length error occurs.

(h) Execution of Deactivate_Channel command

When the host issues the Deactivate_Channel command to the receive VC, it is reported that the command processing has been completed with receive indication, regardless of whether a packet is being received or not.

If the command is issued while the packet of this VC is received, the cells written to the receive PHY until the issuance of the command are transferred to the system memory, and the start address and size of the packet are reported by the same receive indication. If no packet is being same received, size of 0 is reported.

Table 5-6. Errors Occurring in Any of First, Intermediate, and Last Cells of Packet

Error	Dropping Cell	Receive Indication Issuance Timing	Cell after Occurrence
Free buffer underrun	Received cell is dropped if free buffer is not available. Dropping is continued until free buffer is replenished.	If transfer can no longer continue during segment transfer.	Cells belonging to packet causing error, including last cell, are dropped.
Receive FIFO overrun	Cell received, when receive FIFO is full, is dropped.	FIFO overrun status is released after cell has been dropped, and last cell of dropped packet is received.	Cells belonging to packet causing error, including last cell, are dropped.
"MAX. NUMBER OF SEGMENTS" error	Cells, including last cell, received exceeding MAX. NUMBER OF SEGMENTS are dropped.	Cell next to one exceeding MAX. NUMBER OF SEGMENTS is received.	Cells belonging to packet causing error, including last cell, are dropped.
T1 error	None	When T1 time has elapsed.	Received as new packet. → "Length" error
Deactivate_Channel	Dropped.	After completion of command.	Dropped.

Two or more errors may simultaneously occur depending on the type of the error, but only one error is reported by the receive indication. The priority of reporting, depending on the combination of errors, is shown below.

Table 5-7. Error Taking Precedence When Occurring with Other Errors

	Underflow	Overrun	MAX. Error	CRC Error	Abort	Length	T1 error
Underflow	–	–	Underflow	Underflow	Underflow	Underflow	Underflow
Overrun	–	–	Overrun	Overrun	Overrun	Overrun	Overrun
MAX. error	–	–	–	MAX. Error	MAX. Error	MAX. Error	MAX. Error
CRC error	–	–	–	–	CRC Error	CRC Error	CRC Error
Abort	–	–	–	–	–	–	Abort
Length	–	–	–	–	–	–	Length
T1 error	–	–	–	–	–	–	–

5.5.6 Issuance of receive indication

When a packet of AAL-5 traffic (except MEPG-TS reception) is received, the μPD98409 issues a receive indication to inform the host that it has received a packet. When an MPEG-TS packet is received, the μPD98409 reports an error by using the receive indication only when the IE bit of the VC table is set to "1". As the receive indication, 4-word information is generated for each packet, and is stored in a mailbox. The mailbox for reception is mailbox 0 or 1. Which mailbox is to be used is selected by the "MB" bit of the VC table set by the host for each VC. When receive indication is stored, the μPD98409 sets the corresponding MM bit of the GSR register to 1, and issues an interrupt if not masked. The receive indication includes the start address and size of the batch used by the μPD98409 to store the packet. By reading the receive indication, the received packet data can be processed.

The receive indication is issued when all the data (including the AAL-5 trailer) have been stored to the receive buffer after the packet has been correctly received.

If a packet including an error is received, and if the error is the "CRC-32 error" or "Length error", the receive indication is issued when all the data (including the AAL-5 trailer) have been stored to the receive buffer. If the error is other than these, the receive indication is issued as soon as the error has been detected. The detected error is reported in the indication as an error status. When the host reads a receive indication including an error status, it performs appropriate processing, and drops the error packet.

For the format of the receive indication, refer to **5.6 (2) Receive indication**.

5.5.7 Reception of non-AAL-5 traffic

(1) Raw cell reception processing

Raw cell data of 64 bytes, including all the 53-byte raw cell data and 11-byte indication, is generated each time either of the following two types of cells has been received, and the cell is stored in an appropriate free buffer pool. After that, the corresponding bit of RCR7 through RCR0 of the GSR register is set, and an interrupt is issued, if not masked. When a raw cell has been received, no receive indication is stored in a mailbox because processing is performed in cell units, not in packet units.

When a raw cell is received, the function to verify the CRC-10 error code that is appended in cell units is always enabled. If an error is detected, the error bit in the raw cell data is set, and the detection is reported to the host.

<1> OAM F5 cell/RM cell

When the μPD98409 has received a cell where the PTI field of the header has the pattern "1XX" indicating that that the cell is an OAM F5 cell or resource management cell when the user clears the OD bit of the VC table to "0" (to receive an OAM F5 cell), it generates raw cell data and stores it to pool 0. The RCR0 bit of the GSR register is set to 1, and an interrupt is generated, if not masked. The μPD98409 always store the cell to pool 0. If the OD bit is cleared to "0", therefore, pool 0 must be specified as a pool to store raw cell data.

The OD bit can be cleared to "0", regardless of whether the A/R bit is 0 or 1. For example, if a cell whose PTI field is a user data code is received when OD = 0, A/R = 1, and "POOL NO." is set to 8, the cell is stored to pool 8 as an AAL-5 packet. If a cell having the code of an OAM F5 cell/RM cell in the PTI field is received, it is stored to pool 0 as raw cell data. However, if there is a channel that receives OAM F5 cell/RM cell, pool 0 is used to store raw cell data, and a pool other than 0 must be specified to store AAL-5 packets. Both the raw cell data and AAL-5 packets must not be stored to one pool together

<2> Non-AAL-5 traffic

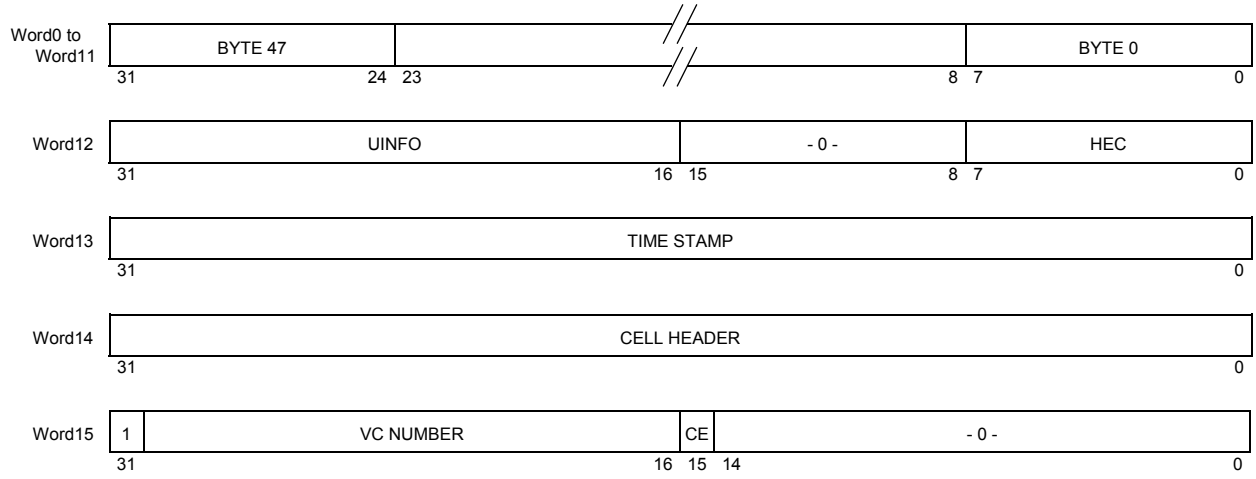
If the A/R bit of the VC table opened by the user is cleared to "0", the μPD98409 transfers a cell of the VC to the system memory as raw cell data, each time it has received such a cell. When a cell is received as raw cell data, the user can specify each of pool 0 to 7 as "POOL NO.". The specified pool must be prepared for storing raw cell data. Using this function, the user can receive non-AAL-5 traffic cells. Assembling a packet from the raw cell data and trailer processing are executed by software in the system memory. When non-AAL-5 traffic is received as raw cells, processing is completed in cell units, and no receive indication is issued for each packet, unlike the AAL-5 packet. Therefore, functions to report errors are not available, unlike when a AAL-5 packet is received. Even if the cell is dropped because a free buffer underflow or receive FIFO overrun error has occurred, the μPD98409 does not report this to the host, and the functions to monitor the MAX. NUMBER OF SEGMENTS and to detect the T1 error are not implemented.

However, indication that indicates occurrence of the RQU interrupt and completion of the Deactivate_Channel command is issued.

(2) Raw cell data

The format of the raw cell data is as follows (in big endian).

Figure 5-45. Format of Raw Cell Data



BYTE0 through BYTE47 : Segment data of receive cell

UINFO : User information. The pattern set by the user to the "UINFO" field of the VC table of this channel is stored as is.

HEC : Pattern of the HEC field included in the header of this cell.

TIME STAMP : Value of the TSR register when this cell is received.

CELL HEADER : 4-byte header of this cell (except HEC at the fifth byte position).

VC NUMBER : VC NUMBER of the VC of this cell.

CE : Verification result of CRC-10 error operation

0: No error

1: CRC-10 error is detected.

(3) AAL-3/4 packet raw cell reception assist function

Normally, each time a raw cell has been received, an RCR interrupt occurs, and the host must execute processing of the raw cell data in cell units. When an AAL-3/4 packet is received, the interrupt can be issued only when the last cell of a packet or 1 cell independent data has been received, in order to mitigate the processing by the host.

When the "A34" bit in Word1 of the receive VC table is set, the μPD98409 checks the segment type (ST field) of an AAL-3/4 cell to be received, and issues an RCR interrupt only if it has received a cell with the 2 bits of its ST field being "01" = end of message, or "11" = single segment message. If any other cell is received, the RCR interrupt is not issued, even though the raw cell data is stored in the system memory.

When the A34 bit is set to 1, the A/R bit must always be set to 1.

Format of AAL-3/4 cell

ST: Segment Type, LI: Length Indication
SN: Sequence Number, CRC: Cyclic Redundancy Check Code
MID: Multiplexed Identification

ST Bit	Segment Type
10	BOM (Beginning of Message)
00	COM (Continuation of Message)
01	EOM (End of Message)
11	SSM (Single-segment message)

5.6 Transmit/Receive Indication

The μPD98409 sends indication to the host as a status indicating completion of transmission/reception of each packet.

The timing of issuance of the transmit/receive indication is as follows:

- Transmit indication: When all the data of one transmit packet have been read.
- Receive indication :
  - When all the data of one receive packet have been transferred to the system memory
  - If reception of a packet is aborted because of an error
  - If the Deactivate_Channel command is received and the command processing has been completed

    During MPEG-TS reception, receive indication is issued only when a receive error occurs and when the IE bit of the receive VC table is set to "1".

The indication is stored in one of the four mailboxes under management of the system memory. Two mailboxes each are available for storing transmit indications and for storing receive indications. The user can select which of the mailboxes is to be used for each VC.

- Transmit indication : Mailboxes 2 and 3 Set to packet descriptor
- Receive indication : Mailboxes 0 and 1 Set to VC table

When indication is stored in a mailbox, the μPD98409 sets the corresponding MM bit of the GSR register to 1, and issues an interrupt, if the bit of the corresponding interrupt mask register IMR is enabled.

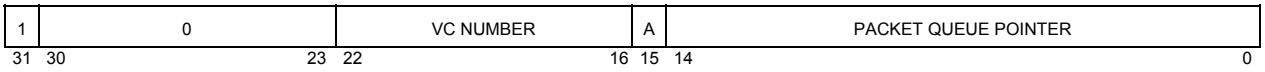
The MSB B of the last word of the transmit/receive indication stored by the μPD98409 is always set to "1" By resetting this bit of an indication that has been read to "0", it can be used as a flag indicating which indications in the mailbox have been processed and which have not been processed. If this bit is used as a flag, two or more indications can be processed at one time by continuing the processing as long as the bit or the indications continues to be "1" (this flag does not have to be changed).

The receive indication contains information on the batch and free buffer to which a packet is stored. The host appropriately processes the receive data based on the indication (such as padding and eliminating the AAL-5 trailer), and then passes the data to the higher application. After that, the host releases the batch and free buffer it has used to the system memory in order to recycle them, or adds them to the chain as a new batch and free buffer.

The format of the transmit/receive indication is explained next.

(1) Transmit indication

The transmit indication is always issued when a packet reception, regardless AAL-5 or non AAL-5, has been completed. Its format is as follows.



VC NUMBER : "VC NUMBER" used by this VC.

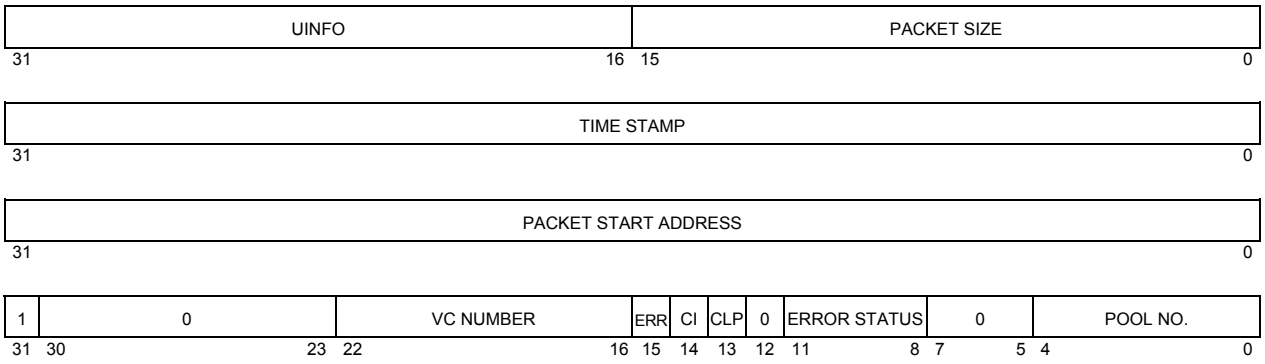
A (Active) : 0: The VC has entered the idle status because the next packet descriptor is a blank packet descriptor.

1: The VC remains in the active status because the next packet descriptor is a valid packet descriptor.

PACKET QUEUE POINTER : The low-order 15 bits of the start address of the next packet descriptor of the transmit queue.

(2) Receive indication

The receive indication is always issued for a packet of AAL-5 traffic. During MPEG-TS reception, receive indication is issued only when a receive error occurs and when the IE bit of the receive VC table is set to "1". If a raw cell of non-AAL-5 traffic is processed, the receive indication is issued only when free buffer underflow occurs or when the Deactivate_Channel command is executed. The format is as shown below.



UINFO : The pattern set by the host to the “UINFO” area is stored as is.

PACKET SIZE : Size of the receive packet.

How the size is reported differs depending on the setting of the PSM bit of the GMR register, as follows:

PSM Bit	Size Indication Mode
0	The entire size of CPCS-PDU including the padding and trailer of an AAL-5 packet is stored in segment units (48 bytes). This is the default mode.
1	The Length field of an AAL-5 packet received normally is stored in this area. Therefore, the user data of CPCS-PDU is indicated in byte units. However, a packet that has caused an error is stored in segment units up to the last portion that was received.

TIME STAMP : Value of the TSR register when this packet was received.

PACKET START ADDRESS : Start address of the batch used.
(invalid in the case of MPEG-TS packet)

VC NUMBER : “VC NUMBER” used by this VC.

ERR
1: Indicates that an error has occurred while the packet was being received.
0: Indicates that the packet was received normally.

CI : Congestion indication. Indicates that a congestion code was received in the PTI field of the header of at least one of the cells belonging to this packet.

CLP : CLP = 1. Indicates that the CLP field in the header of one of the cells belonging to this packet is “1”.

ERROR STATUS : Status of the error that has occurred.
0000 : No error
0001 : Free buffer underflow
0010 : Receive FIFO overrun
0011 : “MAX. NUMBER OF SEGMENTS” violation
0100 : CRC-32 error
0101 : User abort
0110 : “Length” error
0111 : T1 time-out
1000 : Execution of Deactivate_Channel command
For an explanation of how each error is detected, refer to **5.5.5 (4) AAL-5 packet reception error detection.**

POOL NO. : Number of pool used
(invalid in the case of MPEG-TS packet)

5.7 Interrupt Function

The μPD98409 has one open-drain interrupt output pin (INTR_B). This pin can be asserted active by several sources. Each interrupt source is assigned a bit of the GSR register. This bit is set to “1” when the corresponding interrupt occurs.

The bits of the GSR register correspond to the bits of each interrupt mask register IMR. When a bit of the GSR register is set, and when the corresponding bit of the IMR register is set, the interrupt pin becomes active.

When masking an interrupt is cleared by setting a bit of the IMR register to 1, the interrupt is active even if the corresponding bit of the GSR register has been already set to 1.

The GSR register is cleared to 0 each time it has been read by the host. If the same interrupt occurs internally after an interrupt is issued and before the host clears the GSR register by reading it, the bit corresponding to the interrupt source is overwritten.

After reset, all the bits of the GSR register and IMR register are cleared to “0”, and all the interrupt sources are masked.

Of the 32 bits of the GSR register, the PI bit for PHY interrupt is used to input an interrupt from an external device such as a PHY device. When it is detected that the PHY interrupt has become active, the PI bit is set, and an interrupt is generated to the host. In response, the host issues the Indirect_Access command, accesses the registers of the PHY device, and reads the detailed interrupt source as data.

For the details on the GSR register and IMR register, refer to **7.2 (2) GSR** and **(3) IMR**.

5.8 Loopback Function

The μPD98409 has a loopback function for debugging and testing. The loopback function is executed by setting the LP bit of the GMR register to 1. When the loopback function is active, the μPD98409 lets the data received from the host pass through the transmission block, loops it back through the internal circuit of the interface of the PHY device, and returns it to the host via the receive block. The transmit/receive indication is issued normally. In the loopback mode, data is not transmitted from the PHY device interface (UTOPIA interface) to an external device.

5.9 Global Shutdown Function

Global shutdown is executed by setting the MSB bit of the VRR register. When the μPD98409 receives this command, it stops processing all the receive VCs under way. After that, the μPD98409 does not receive cells, stopping its reception function completely. The transmission function is not affected and continues operating.

When the μPD98409 has stopped its reception function, it sets the RD bit of the GSR register to 1, and issues an interrupt to the host, if not masked.

The reception function of the μPD98409 cannot be enabled again once this command has been issued to the μPD98409. To activate the μPD98409 again, execute software reset or hardware reset to the μPD98409.

[MEMO]

CHAPTER 6 COMMANDS

The host issues the seven types of commands listed in Table 6-1 to the μPD98409. This chapter first explains the procedure to issue a command to the μPD98409, and then the details of each command.

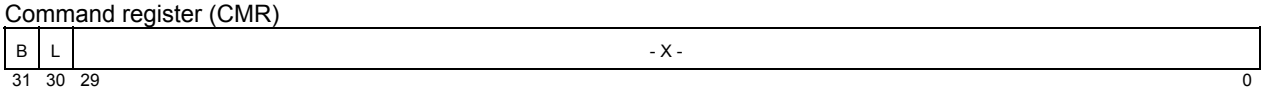
Table 6-1. Command Type

Command Name
Open_Channel command
Close_Channel command
Deactivate_Channel command
Tx_Ready command
Add_Batches command
NOP command
Indirect_Access command

6.1 Command Register (CMR) and Command Extension Register (CER)

All commands are input to the μPD98409 via the command register and command extension register. The address of each register is shown in the table below. The command register has a “busy flag” that functions as a command status flag.

Command register (CMR)



B bit : Busy flag
L bit : Bit used in the μPD98409 during command execution. The value is undefined when it is read.
X : Command code, parameter

Command extension register (CER)

Name	Register Name	Address
Command register	CMR	20H
Command extension register	CER	28H

6.2 Busy Flag

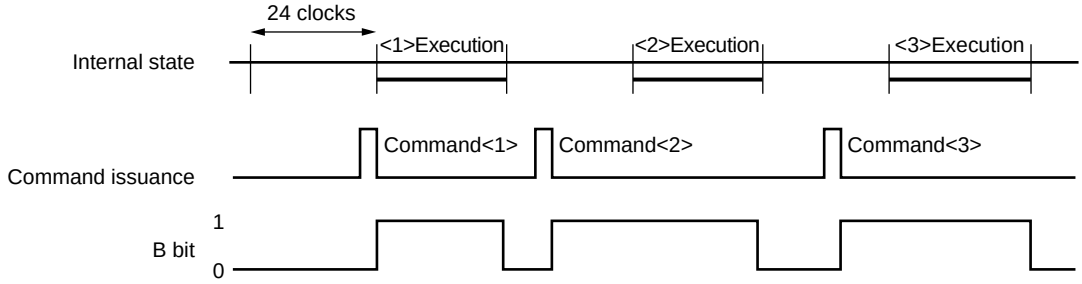
The host writes a command with various parameters to the command register.

Internally, the μPD98409 operates in units called states which are equivalent to the duration of 24 system clocks. The μPD98409 receives one command from the host in one state, and executes and completes it in the next state. Because the μPD98409 can only execute one command in one state, if it receives a command in a certain state, it sets bit 31 (B bit) of the command register to “1”, indicating that the next state is filled. In other words, this B bit functions as a busy flag that indicates that the μPD98409 is executing a command.

While the B bit is “1”, a new command cannot be received until the next state. Even if a new command is overwritten to the command register while the B bit is 1, the μPD98409 ignores and invalidates this command. The host can issue a new command only while the B bit is “0”.

The μPD98409 clears the B bit to “0” when it completes execution of a command in the state after to the one in which it received the preceding command.

Figure 6-1. Command Issuance and B Bit



When the host issues a new command, therefore, it must read the command register once and confirm that the “B bit” is “0” (execution of the preceding command has been completed). Alternatively, the host must wait at least for the duration of two states (48 clocks) after issuance of the preceding command, and then issue the next command.

6.3 Commands Returning Command Indication

When the Open_Channel or Close_Channel command is executed, the μPD98409 stores command indication in the command register.

When the host issues a command having a command indication, it must read the command register to receive the command indication. The μPD98409 stores the command indication in the command register when it has completed execution in the command. While the B bit is “1”, execution of the command has not been completed yet, and the command indication has not been stored in the command register. If the B bit is “0”, the command indication is stored in the command register.

The host either polls the command register or waits for the duration of 48 clocks after issuing the preceding command until the B bit is cleared to “0”.

6.4 Commands Using Command Extension Register

The “Indirect_Access command” and “Add_Batches command” use the command extension register because these commands require setting parameters.

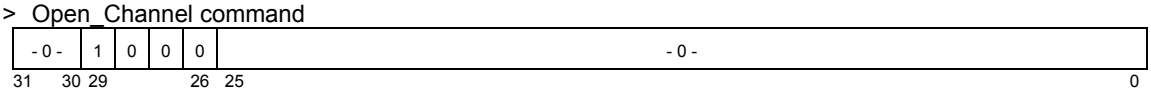
When data is to be written to a command extension device by either of these commands, the command is set in the command register. The μPD98409 starts executing the command when the command has been written to the command register. Therefore, the data of the command extension register must be changed before data is written to the command register.

When data is read, a command is written to the command register, and the data stored in the command extension register after execution of the command has been completed is read. The μPD98409 stores the data in the command extension register when execution of the command has been completed. The host reads the command extension register after it has confirmed that the B bit is “0”, or after waiting for a duration of more than 48 clocks.

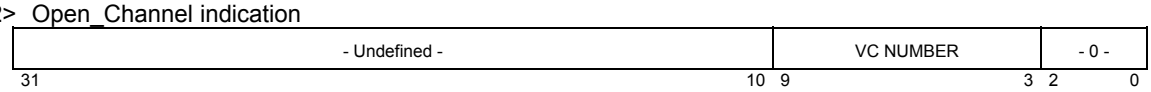
The parameter to be set in the CER register differs depending on the command. Refer to the explanation of each command.

6.5 Commands and Command Indication

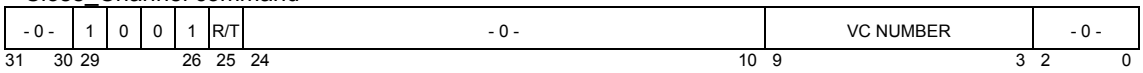
<1> Open_Channel command



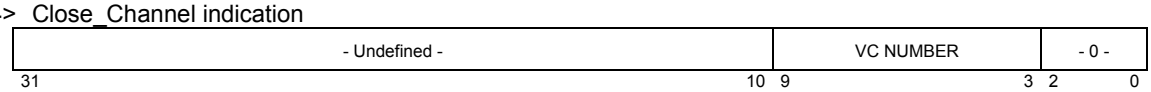
<2> Open_Channel indication



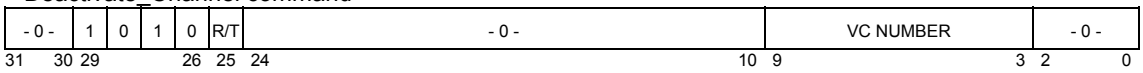
<3> Close_Channel command



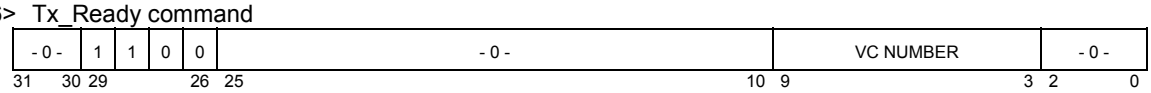
<4> Close_Channel indication



<5> Deactivate_Channel command

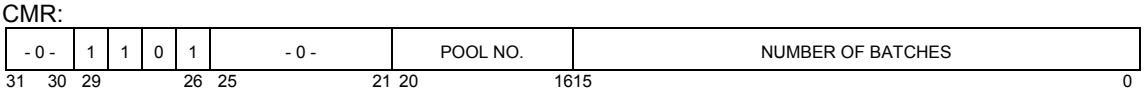


<6> Tx_Ready command

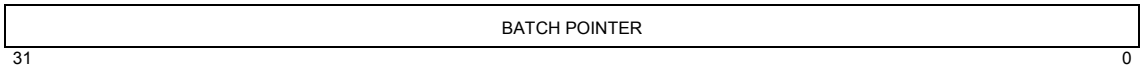


<7> Add_Batches command

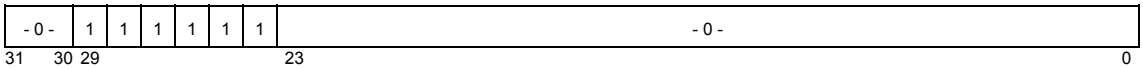
CMR:



CER:

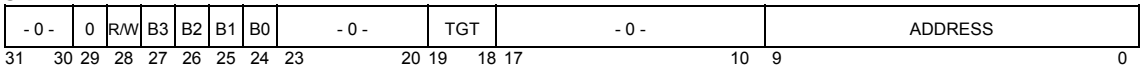


<8> NOP command



<9> Indirect_Access command

CMR:

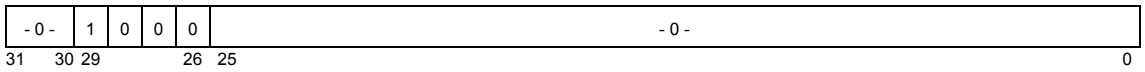


CER:

6.6 Command Details

<1>	Open_Channel command

Programming of command register

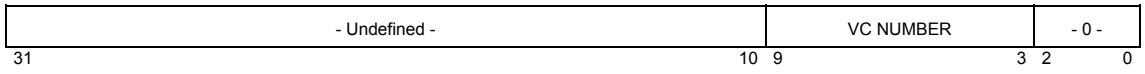


Command description

The Open_Channel command is used to open a new channel used for transmission or reception. When the μPD98409 receives this command, it stores the contents of the TOS register in the command register, and then stores them in the CMR register as Open_Channel indication. After that, the contents of the TOS register are updated to the value indicated by the “FORWARD POINTER” of the VC table that opened the contents of the TOS register. This command does not distinguish between transmission and reception.

<2>	Open_Channel indication

Contents of command register



Parameter description

VC NUMBER (bits 3 through 9): Bits 9 through 3 of the first address of the VC table for a new channel. If this area is 0, it indicates that no more new channels can be opened.

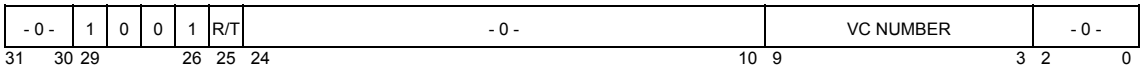
Command description

When the μPD98409 executes the Open_Channel command, it stores a part of the first address of the new VC table, VC NUMBER, in the command register as an indication. The μPD98409 stores this indication in the command register when execution of the Open_Channel command has been completed. The host confirms that the B bit is 0 after it has issued the Open_Channel command, and receives the indication.

If “000” is added to the low-order 3 bits of this parameter, a physical address indicating Word0 of the VC table in the control memory is formed.

<3> Close_Channel command

Programming of command register



Parameter description

R/T: Indicates whether of a transmit or receive channel is to be closed.

1 - Receive channel

0 - Transmit channel

VC NUMBER (bits 3 through 9): "VC NUMBER" of the channel to be closed.

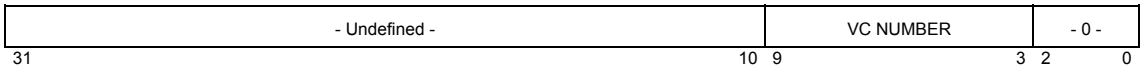
Command description

The Close_Channel command is used to close a receive channel or transmit channel. When the μPD98409 receives this command, it returns the specified VC table to the free block pool and updates the contents of the TOS register so that they indicate that VC table.

This command must be always used together with the Deactivate_Channel command. The procedure differs depending on whether a transmit or receive channel is to be closed by this command. For the differences, refer to the description of the Deactivate_Channel command.

<4> Close_Channel indication

Contents of command register



Parameter description

VC NUMBER (bits 9 through 3): Bits 9 through 3 of the first address of the VC table of the channel to be closed are stored.

If this area is 0, it indicates that the μPD98409 rejected the issued Close_Channel command. The host repeatedly issues the Close_Channel command until it obtains a VC NUMBER.

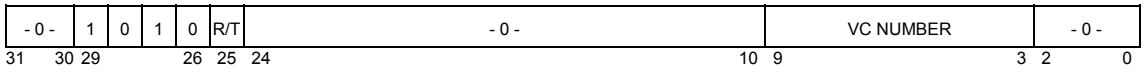
Command description

When the μPD98409 executes the Close_Channel command, it stores the VC NUMBER of the VC table of the channel closed in the command register as an indication. The μPD98409 stores this indication in the command register when execution of the Close_Channel command has been completed. The host confirms that the B bit is 0 after it has issued the Close_Channel command, and then receives the indication.

The μPD98409 may return "0" as the indication of the Close_Channel command. This indicates that the specified channel could not be closed. In this case, the host must repeatedly issue the Close_Channel command until it can obtain the "VC NUMBER" of the VC closed.

<5>	Deactivate_Channel command

Programming of command register



Parameter description

R/T: Indicates whether the channel to be deactivated is a receive or transmit channel.

1 - Receive channel

0 - Transmit channel

VC NUMBER (bits 3 through 9): "VC NUMBER" of the channel to be deactivated.

Command description

The Deactivate_Channel command is to get the μPD98409 ready for closing an active transmit channel or receive channel before the Close_Command is issued.

If the host issues the Close_Channel command when there is still cell data in the internal transmit/receive FIFO of the μPD98409 and the μPD98409 has not completed processing of the data, the contents of the VC table used by the channel are cleared and, as a result, the μPD98409 can no longer continue processing correctly. This Deactivate_Channel command is used to allow sufficient time to elapse during which all the processing related to the cell of the channel to be closed is completed.

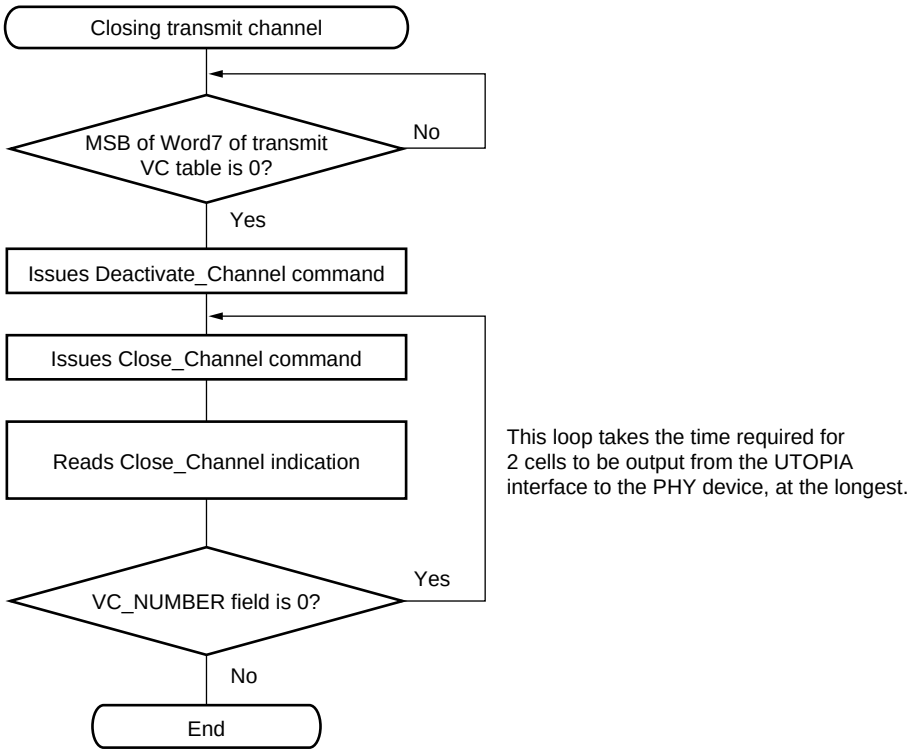
- **Transmit channel**

The targeted channel can be closed while it is in the idle status.

When the Close_Channel command is issued, the μPD98409 clears the VC table and returns it to the free block pool. The transmit channel changes its status from active to idle if the packet descriptor next to the one fetched by the μPD98409 is a blank descriptor. The channel enters the idle status by clearing the active bit (A bit of Word7) to "0". However, the processing related to the channel continues internally until all the cell data loaded to the internal transmit FIFO is output to the PHY device, and therefore, the contents of the VC table must be retained until then. The Deactivate_Channel command ensures the existence of the VC table as long as the internal processing continues. When the μPD98409 receives the Deactivate_Channel command, it measures the quantity of the cell data in the internal transmit FIFO, and returns "0" as Close_Channel indication, not the executing the Close_Channel command until the processing is completed. The Deactivate_Channel command measures the timing at which the Close_Channel command is internally accepted.

Figure 6-2 illustrates how a transmit channel is closed.

Figure 6-2. Closing Transmit Channel



<1> Check to see if the MSB (A bit) of Word 7 of the transmit VC table is '0' (inactive). If it is '1' (active), wait until packet transmission is completed.

<2> If the A bit is '0' (inactive), issue the Deactivate_Channel command.

<3> Issue the Close_Channel command.

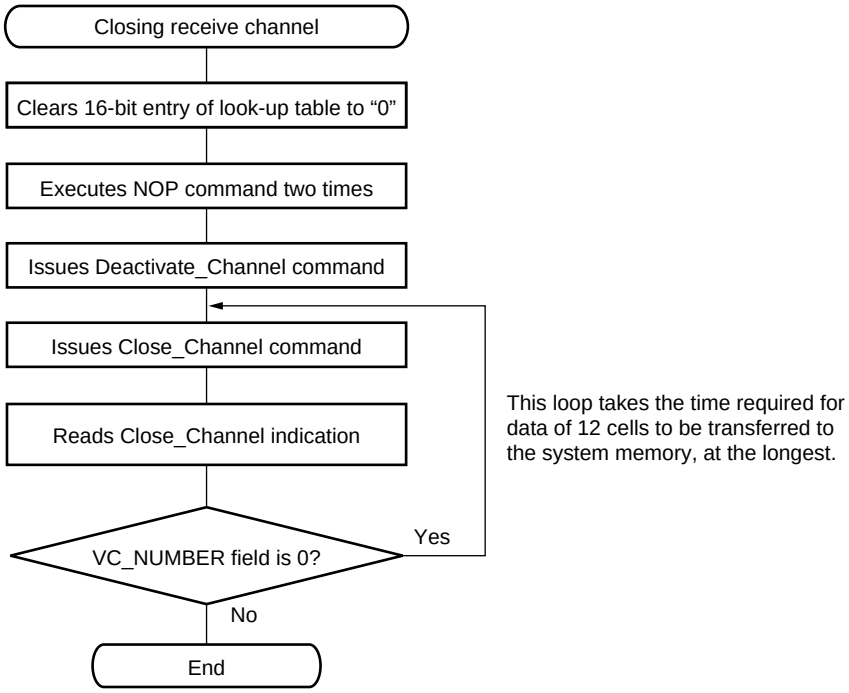
<4> Read the Close_Channel indication from the μPD98409.

<5> Check to see if the VC NUMBER field is '0'. If so, perform processing from <3>. If the VC NUMBER of the channel to be closed is stored in the VC NUMBER field, closing the transmit channel is completed.

- **Receive channel**

  Like that of a transmit channel, the Deactivate_Channel command for a receive channel is used to ensure the lapse of the time during which all the cell data of the channel are transferred from the receive FIFO to the host system. The receive channel no longer receives the cells of the VC and enters the idle status when the entry in the look-up table is disabled. Even in the idle status, however, the cells that were received immediately before the channel has entered the idle status still remain in the FIFO. The Deactivate_Channel command is used to delay execution of the Close_Channel command until the processing of these cells is completed.
  Figure 6-3 illustrates how a receive channel is closed.

Figure 6-3. Closing Receive Channel



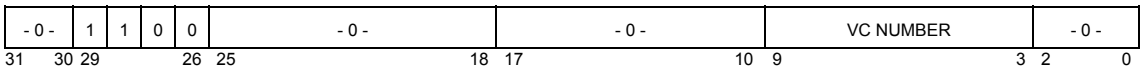
<1> Clear the 16-bit entry of the receive look-up table.
<2> Insert a wait state by issuing the NOP command two times.
<3> After that, issue the Deactivate_Channel command.
<4> Issue the Close_Channel command.
<5> Read the Close_Channel indication.
<6> Check to see if the VC NUMBER field is ‘0’. If ‘0’, perform processing from <4>. If the VC NUMBER of the channel to be closed is stored in the VC NUMBER field, closing the receive channel is completed.

Remarks 1. When the μPD98409 completes execution of the Deactivate_Channel command, it stores the receive indication including its status code to a specified receive mailbox. However, the host does not have to wait for this receive indication to be stored, to issue the Close_Channel command.

2. To close a receive channel set for MPEG-TS packets, the following processing is performed. If the number of transfer bytes of the packet is not a multiple of 188 after the cell data remaining in the internal receive FIFO has been output to the PCI bus, dummy data “0” is written to TS-ADDRESS address so that the number of bytes is a multiple of 188 (1- to 46-word burst transfer).

<6> Tx_Ready command

Programming of command register



Parameter description

VC NUMBER (bits 3 through 9): "VC NUMBER" of the channel to which a transmit operation has been started to the μPD98409.

Command description

The Tx_Ready command is used by the host to report to the μPD98409 that a transmit packet has been added to this channel (that a new packet descriptor has been added to the transmit queue). When the μPD98409 receives the Tx_Ready command, it activates the targeted transmit VC table.

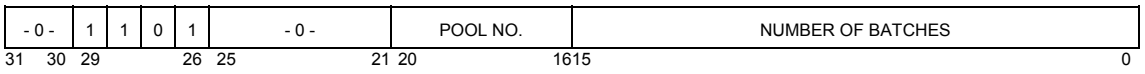
The active status of the channel is retained until a packet descriptor whose V bit is "0" arrives. Therefore, this Tx_Ready command does not have to be issued once to one transmit packet.

For example, if three valid packet descriptors are arranged in the transmit queue in a row and the Tx_Ready command is issued, the μPD98409 transmits the three packets and then enters the idle status. The μPD98409 ignores the Tx_Ready command issued to a channel in the active status, and does not influence the transmission/reception of that channel.

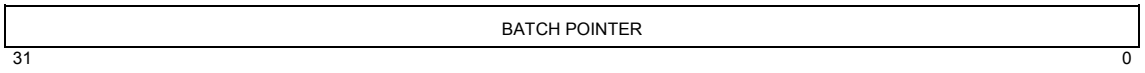
<7> Add_Batches command

Programming of command register

CMR:



CER:



Parameter description

POOL NO.	: Specifies number 0 to 31 (0000B to 11111B) of the targeted pool.
NUMBER OF BATCHES	: Writes the number of batches to be newly added.
BATCH POINTER	: Writes the first address of the first batch in the batch list to be newly added.

Command description

The host uses the Add_Batches command to add an unused batch to one of the 32 receive free buffer pools. This command uses the command extension register (CER). "BATCH POINTER" is written to the command extension register and then to the command register (CMR).

The operation of the μPD98409 when it has received the Add_Batches command differs depending on whether any unused batches remain in the targeted pool or not.

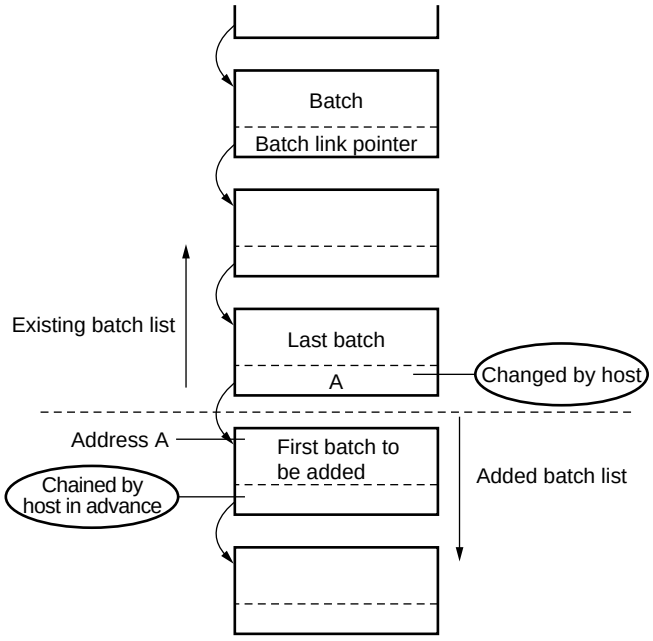
(a) If unused batches remain in pool

(If "REMAINING NUMBER OF BATCHES" of pool descriptor is not "0")

The μPD98409 adds the value of "NUMBER OF BATCHES" to "REMAINING NUMBER OF BATCHES" of the pool descriptor. At this time, the host must change the "batch link pointer" of the remaining batch to the first address of the batch to be newly added (refer to **Figure 6-4**).

In addition, the batch to be added must be chained by the link pointer.

Figure 6-4. Work of Host before Issuance of Add_Batches Command



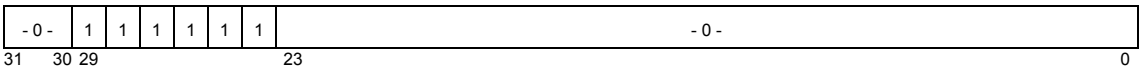
(b) If pool is empty

(If "REMAINING NUMBER OF BATCHES" of pool descriptor is "0" = in RQU status)

If the μPD98409 receives the Add_Batches command while the pool is empty, it loads the "NUMBER OF BATCHES" and "BATCH POINTER" of the command to the "REMAINING NO. OF BATCHES" and "ADDRESS" fields of the pool descriptor in the control memory.

<8>	NOP command

Programming of command register



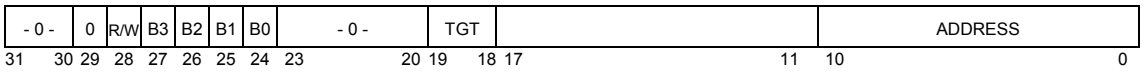
Command description

The host uses the NOP command to delay execution of the other commands (such as the Deactivate_Channel command). When the μPD98409 receives this command, it does nothing, except setting the B bit to 1 until the next command state is received (for the duration of 24 clocks).

<9>	Indirect_Access command

Programming of command register

CMR:



CER:

Parameter description

R/W : Specifies whether the target is accessed for read or write.
1: Read
0: Write

B0, B1, B2, B3 : The μPD98409 writes data only to the byte with this bit set to "1" when the control memory is accessed for a write. If the control memory is not accessed for a write, the setting of this bit is invalid.

1: Enables writing to corresponding byte.
0: Does not write to corresponding byte.

TGT (Target) : Specifies the target device.
00 : Control memory
01 : Indirect address register of μPD98409
11, 10 : PHY device

ADDRESS : Address output to the target device

DATA : Sets the contents to be written to the target device whose address is specified by the host during write. During read, the contents received from the target device whose address is specified are stored by the μPD98409.

Command description

The Indirect_Access command is used by the host to access the following target devices for read or write.

- Indirect address register of μPD98409
- Control memory
- PHY device

This command is used together with the command extension register (CER).

During write access, the host stores the data to be written to the target device in the command extension register. The host then writes the command to the command register. When the μPD98409 receives the command, it starts a write cycle for the target device.

During read access, the host writes the command to the command register. The μPD98409 executes a read cycle of the target device, and stores the requested data in the command extension register. The host confirms that the command execution has been completed, and then reads the command extension register.

The read/write cycle started by the Indirect_Access command differs depending on the target device. The cycle to access the internal indirect address register and control memory of the μPD98409 is executed by the internal

bus of the μPD98409. The cycle to the PHY device is executed by using the address lines (CA8 through CA0) and data lines (CD7 through CD0), and the control signals (PHOE_B, PHCE_B, and PHRW_B) of the PHY device interface.

[MEMO]

CHAPTER 7 INTERNAL REGISTERS

The internal registers of the μPD98409 are classified into two types: direct address registers and indirect address registers. The direct address registers are directly accessed via the bus interface. The indirect address registers are accessed by executing the Indirect_Access command.

Remarks 1. Do not access addresses to which a register is not allocated.

2. Do not write data to a register whose access mode is read only.

3. For registers whose access mode is read/write, write only 0 to fields marked "0" in the register field diagram in each register description.

7.1 Register List

(1) Direct address register

(1/2)

Address B (H)	Register	Function	Read/Write
00	GMR	General mode register	Read/Write
04	GSR	General status register	Read
08	IMR	Interrupt mask register	Read/Write
0C	RQU	Receive queue underrun	Read
10	RQA	Receive queue alert	Read
14	ADDR	Last burst address	Read
18	VER	Version number	Read
1C	SWR	Software reset	Write
20	CMR	Command register	Read/Write
28	CER	Command extension register	Read/Write
30	ECCR	EEPROM command control register	Read/Write
34	ERDR	EEPROM read data register	Read
40	MSH0	Mailbox 0 start address, high	Read/Write
44	MSH1	Mailbox 1 start address, high	Read/Write
48	MSH2	Mailbox 2 start address, high	Read/Write
4C	MSH3	Mailbox 3 start address, high	Read/Write
50	MSL0	Mailbox 0 start address, low	Read/Write
54	MSL1	Mailbox 1 start address, low	Read/Write
58	MSL2	Mailbox 2 start address, low	Read/Write
5C	MSL3	Mailbox 3 start address, low	Read/Write
60	MBA0	Mailbox 0 bottom address	Read/Write
64	MBA1	Mailbox 1 bottom address	Read/Write
68	MBA2	Mailbox 2 start address, high	Read/Write
6C	MBA3	Mailbox 3 start address, high	Read/Write
70	MTA0	Mailbox 0 tail address	Read/Write
74	MTA1	Mailbox 1 tail address	Read/Write
78	MTA2	Mailbox 2 tail address	Read/Write

(2/2)

Address B (H)	Register	Function	Read/Write
7C	MTA3	Mailbox 3 tail address	Read/Write
80	MWA0	Mailbox 0 write address	Read/Write
84	MWA1	Mailbox 1 write address	Read/Write
88	MWA2	Mailbox 2 write address	Read/Write
8C	MWA3	Mailbox 3 write address	Read/Write

(2) Indirect address registers

The indirect address registers are accessed by using the Indirect_Access command of the μPD98409.

Scheduler registers

Address (H)	Register	Function
40000 - 40001	I, M	I and M entries of schedulers 0 through 1
40010 - 40011	X	X entry of schedulers 0 through 1
40020 - 40021	Y	Y entry of schedulers 0 through 1
40030 - 40031	P, C, p, c	P, C, p, and c entries of schedulers 0 through 1
40040 - 40041	Pri & Status	Priority and status of schedulers 0 through 1

Other registers

Address (H)	Register	Function
40100	TOS	Control memory address of top of stack
40200	SMA	Control memory start address of shaper
40201	PMA	Control memory start address of receive pool
40300	T1R	T1 register
40301	VRR	VPI/VCI reduction register/global shutdown
40302	TSR	Time stamp register
40500	GPOR	General-purpose output port register

7.2 Direct Address Registers

(1) GMR (General Mode Register)

The GMR register is set by the host mainly to select the operation mode of the Time stamp register μPD98409 and to enable the transmission/reception function. The host sets this register first after resetting the device. Because this register sets the basic operation modes of the μPD98409, do not change its contents after the host has set the SE and RE bits of this register and the μPD98409 has started transmission/reception; otherwise, malfunctioning may occur.

Address : 00H
Access mode : Read/write

31	30	29	28	27	26	25	17	16	15	14	13	12	11	8	7	6	5	4	3	2	1	0
- 0 -		UCB	ICM	PSM	UOC	- 0 -		TBE	0	LP	- 0 -		SZ		AD	- 0 -				DR	SE	RE

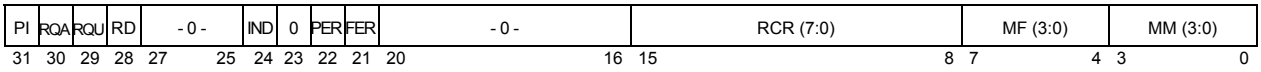
Field	Function	Value after Reset
UCE	Selects cell transfer function between packets 0: Always transmits cells selected by ICM field to PHY layer device regardless of status of VC channel. 1: Makes TENBL signal high if there is no active VC, and does not transmit cells. Transmits cells selected by ICM field only to adjust rate.	1: Does not transmit cells if there is no active VC
ICM	Idle cell mode. 0: Selects the use of unassigned cells to adjust the rate, to insert between packets, or to transmit using the unassigned cell function. 1: Selects the use of idle cells to adjust the rate, to insert between packets, or to transmit using the unassigned cell function.	0: Transmits unassigned cell
PSM	Receive packet size report mode of receive indication. 0: Reports the entire size of a receive AAL-5 packet in segment units. 1: Reports with the Length field of a receive packet. Only the user data length is reported in byte units. However, an error packet is reported in cell units.	0: Reports in cell units
UOC	UTOPIA interface mode 0: Octet-level mode 1: Cell-level mode	0: Octet-level mode
TBE	Enables 12-word burst. 0: Disable 1: Enable To enable 12-word burst, the AD bit must be also set to 1.	0: Disable
LP	Sets loopback mode 0: Normal operation 1: Loopback mode	0: Normal operation

Field	Function	Value after Reset
SZ	Enables burst size. Two or more burst sizes can be enabled. Bit 8 = '1' : Enables 2-word burst Bit 9 = '1' : Enables 4-word burst Bit 10 = '1' : Enables 8-word burst Bit 11 = '1' : Enables 16-word burst **Remark** The μPD98409 uses the 16-word burst only to store raw cell data.	All 0. All multi-word DMA transfer is disabled. Only 1-word transfer is enabled.
AD	Enables or disables the function to automatically check the address field at the transfer destination and select burst size when the μPD98409 executes DMA transfer. 0: Enable 1: Disable	0: Disable
DR	Selects receive or drop mode. 0: Drop mode. Overrun is assumed and cells are dropped internally if the receive FIFO is full. 1: No Drop mode. Requests PHY device to stop transfer of cells using the RENBL_B signal when the receive FIFO is full. The μPD98409 does not drop the cells. **Caution When using the MPEG-TS packet receive function, be sure to select the No Drop mode. Operation is not guaranteed if the Drop mode is selected**.	0: Drop mode
SE	Enables or disables the shapers. This bit enables the transmission function of the μPD98409. 1: Transmission enable 0: Transmission disable	0: Transmission disable
RE	Enables or disables the reception function of the μPD98409. 1: Reception enable 0: Reception disable	0: Reception disable

(2) GSR (General Status Register)

The GSR register indicates interrupt sources that have been generated. If an interrupt source is internally generated, the corresponding bit of this register is set to 1. If the interrupt is unmasked by the corresponding bit of the interrupt mask register IMR, the interrupt occurs. This register is cleared when it is read by the host. If the same source is generated before this register is cleared, 1 is overwritten.

Address : 04H
Access mode : Read only



Field	Function	Value after Reset
PI	Interrupt from the PHY device. '1' indicates that a low level is input to the PHINT_B pin from the PHY device and that an interrupt has been received.	0
RQA	Receive buffer alert. Indicates that a pool exceeding "ALERT LEVEL" exists in the "REMAINING NUMBER OF BATCHES IN THE POOL" of the pool descriptor. The host reads the RQA register to determine in which pool the error has occurred.	0
RQU	Receive free buffer underflow. Indicates that a pool with the "REMAINING NUMBER OF BATCHES IN THE POOL" of the pool descriptor being 0 (no unused batch) exists. The host reads the RQU register to determine in which pool the error has occurred.	0
RD	Receiver deactivate complete. '1' indicates that execution of global shutdown issued by the host has been completed and that the reception function has been stopped.	0
IND	Control memory initialization complete. '1' indicates that initialization of the control memory performed by the μPD98409 after reset has been completed. A time of about 32k system clocks is required until this bit is set. Until then, the host can only access the direct address registers other than the command register of the μPD98409.	0
PER	This is a status bit related to the parity error of the PCI bus and is set to 1 in the following cases: • If the μPD98409 detects that PERR_B is active during master write. • If the μPD98409 detects a parity error in the data phase during master read.	0

Field	Function	Value after Reset
FER	This is a status bit related to the transactions of the PCI bus and is set to 1 in the following cases: • The μPD98409 executes master abort because the DEVSEL_B signal does not become active. • The target executes target abort by making the STOP_B signal active. • The retry timer is full and transaction has ended. • The TRDY timer is full and transaction has ended. When this bit is set to 1, the transmission/reception function and master function of the μPD98409 are stopped. Execute software reset because target transaction to the direct address registers can be accepted.	0
RCR (7:0)	Raw cell data reception. A bit in this field which is '1' indicates that the μPD98409 has stored raw cell data in a pool. Bit 8 of this field corresponds to RCR0, indicating that the data has been stored to pool 0.	0
MF (3:0)	Mailbox full. A bit in this field which is '1' indicates that the write pointer (MWA) of the μPD98409 has caught up and coincides with the read pointer (MTA) of the host in the mailbox corresponding to the bit. Bit 4 of this field corresponds to MF0, indicating that mailbox 0 is full.	0
MM (3:0)	Stores indication in a mailbox. A bit in this field which is '1' indicates that the μPD98409 has stored a new indication in the mailbox corresponding to the bit. Bit 0 of this field corresponds to MM0, indicating that the indication has been stored in mailbox 0.	0

(3) IMR (Interrupt Mask Register)

The IMR register masks or unmasks issuance of the interrupt corresponding to each interrupt source. The mask bits corresponding to the bits of the GSR register are located at the same bit positions of the IMR register. When the bit of the GSR register corresponding to the bit that unmasks an interrupt is set to 1, the interrupt output pin is asserted active.

Address : 08H
Access mode : Read/write

MASK

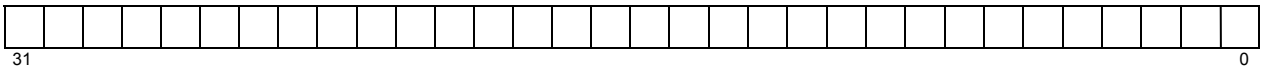
31 0

Field	Function	Value after Reset
MASK	The mask bits corresponding to the respective bits of the GSR register are located at the same bit positions in the mask register. 0: Mask 1: Unmask. When 1 is set in the GSR register, the corresponding interrupt occurs.	All 0. All interrupts are masked.

(4) RQU (free buffer underflow register)

The bit of the RQU register corresponding to a pool where no more free buffers exist is set to 1. If any of the bits of this register is set to 1, the RQU bit of the GSR register is set to 1. The μPD98409 detects that there are no free buffers after it has received a cell and when it transfers the data of the cell to the system memory. Until free buffers are replenished, the cells of the VC set to the pool without free buffers are continuously dropped, and each time a new packet has arrived, this bit is set.

Address : 0CH
Access mode : Read only



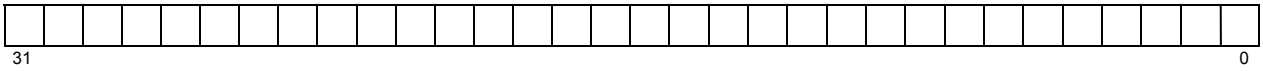
Field	Function	Value after Reset
Receive Queue Underrun	Bit 0 corresponds to pool 0, and bit 31 corresponds to pool 31. '1' indicates that there is no free buffer for the corresponding pool for the bit location.	All 0

(5) RQA (Receive Queue Alert register)

The RQA register is set to 1 if the batches of a pool have been consumed and the "REMAINING NUMBER OF BATCHES IN THE POOL" of the pool descriptor indicating the remaining number of pools has become equal to the number of batches of "ALERT LEVEL" set by the user.

If any of the bits of this register is set to 1, the RQA bit of the GSR register is set.

Address : 10H
Access mode : Read only

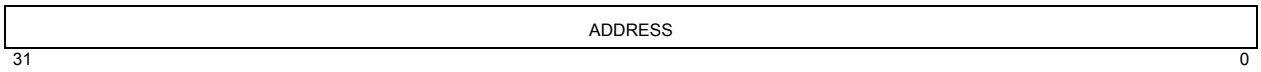


Field	Function	Value after Reset
Receive Queue Alert	Bit 0 corresponds to pool 0, and bit 31 corresponds to pool 31. '1' indicates that the number of remaining batches of the pool corresponding to the bit location has reached ALERT LEVEL. This means that the free buffers of the pool are about to be used up.	All 0

(6) ADDR (burst Address register)

The ADDR register is used to test the device. When the host reads this register, the address of the DMA cycle executed last by the μPD98409 is stored.

Address : 14H
Access mode : Read only

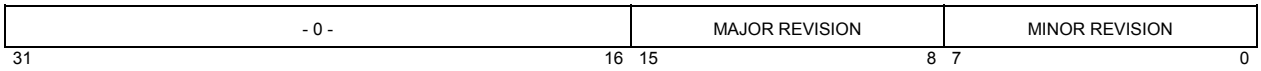


Field	Function	Value after Reset
Address	Last burst address. Transfer address of the last DMA cycle transfer executed up to the time the host reads this register.	All 0

(7) VER (Version register)

This register stores the code indicating the version of this chip. By reading this register, the version of the chip software can be identified.

Address : 18H
Access mode : Read only



Field	Function	Value after Reset
MINOR REVISION	Stores a code indicating the version of this chip.	Consult NEC distributor.
MAJOR REVISION		02H

(8) SWR (Software Reset virtual register)

Address : 1CH
Access mode : Write only

SWR is a virtual register. Any write operation to address 1CH causes software reset.

(9) CMR (Command Register)

Address : CMR : 20H
Access mode : Read/write

The CMR register is used by the host to write a command to the μPD98409 or to receive a command indication for the written command. For an explanation of how to use this register, refer to **CHAPTER 6 COMMANDS**.

(10) CER (Command Extension Register)

Address : CER : 28H

Access mode : Read/write

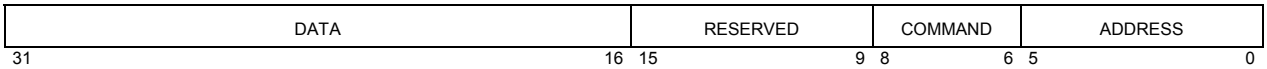
The CER register is used when the Indirect_Access or Add_Batches command is issued. For an explanation of how to use this register, refer to **CHAPTER 6 COMMANDS**.

(11) ECCR register

This register is used to set commands, data to be written, and addresses for the EEPROM.

Address : 30H

Access mode : Read/write



Field	Function	Value after Reset
DATA	EEPROM write data. Sets 2-byte data to be written to EEPROM.	0
COMMAND	EEPROM command. Sets command for EEPROM.	0
ADDRESS	EEPROM address. Sets address of EEPROM.	0

EEPROM command list

Bit				Command Description	
8	7	6	5-0		
1	1	0	Address	READ	Data read. Stored in low-order 16 bits of ERDR register.
1	0	1	Address	WRITE	Data write.
1	1	1	Address	ERASE	Data erase.
1	0	0	11xxxx	EWEN	Write/erase enable.
1	0	0	10xxxx	ERAL	Erases all areas.
1	0	0	01xxxx	WRAL	Writes data to all areas.
1	0	0	00xxxx	EWDS	Disables write/erase.

(12) ERDR register

This register stores data read from the EEPROM.

Address : 34H

Access mode : Read

B	0	READ DATA
31	16	15 0

Field	Function	Value after Reset
B	Indicates whether data read is valid or invalid. 0: READ DATA field data is valid. 1: READ DATA field is still invalid because read command is under execution.	0
READ DATA	Read data. Data read by executing read command is stored.	0

(13) MSH0 through MSH3 (Mailbox Start address, High) (4 registers)

Addresses : 40H - MSH0: Mailbox 0
44H - MSH1: Mailbox 1
48H - MSH2: Mailbox 2
4CH - MSH3: Mailbox 3

Access mode : Read/write

MSH0 through MSH3 are registers that set the high-order 16 bits of the 32-bit start addresses of the four mailboxes set in the system memory. Of the four mailboxes, the host sets the address of the mailbox to be used to the corresponding MSH register. The initial values of these register are undefined after reset.

- 0 -	MSH0
31 16	15 0

- 0 -	MSH1
31 16	15 0

- 0 -	MSH2
31 16	15 0

- 0 -	MSH3
31 16	15 0

(14) MSL0 through MSL3 (Mailbox Start address, Low) (4 registers)

Addresses : 50H - MSL0: Mailbox 0
54H - MSL1: Mailbox 1
58H - MSL2: Mailbox 2
5CH - MSL3: Mailbox 3

Access mode : Read/write

MSL0 through MSL3 are registers that set the low-order 16 bits of the 32-bit start addresses of the four mailboxes set in the system memory. Of the four mailboxes, the host sets the address of the mailbox to be used to the corresponding MSL register. The initial values of these register are undefined after reset.

- 0 -	MSL0
31 16	15 0

- 0 -	MSL1
31 16	15 0

- 0 -	MSL2
31 16	15 0

- 0 -	MSL3
31 16	15 0

(15) MBA0 through MBA3 (Mailbox Bottom Address) (4 registers)

Address : 60H - MBA0: Mailbox 0
64H - MBA1: Mailbox 1
68H - MBA2: Mailbox 2
6CH - MBA3: Mailbox 3

Access mode : Read/write

MBA0 through MBA3 store the low-order 16 bits of the 32-bit bottom addresses of the four mailboxes set in the system memory. The address next to the last word of the area used as a mailbox area is set as the bottom address. The host sets the address of the mailbox to be used to the corresponding MBA register. The high-order 16 bits of the addresses are set by the MSH register. The value set to the MBA register is not equal to the value set to the MSL register. The initial values of these registers are undefined after reset.

- 0 -	MBA0
31 16	15 0

- 0 -	MBA1
31 16	15 0

- 0 -	MBA2
31 16	15 0

- 0 -	MBA3
31 16	15 0

(16) MTA0 through MTA3 (Mailbox Tail Address) (4 registers)

Address : 70H - MTA0: Mailbox 0
74H - MTA1: Mailbox 1
78H - MTA2: Mailbox 2
7CH - MTA3: Mailbox 3

Access mode : Read/write

MTA0 through MTA3 store the low-order 16 bits of the read pointer of the four mailboxes read by the host. These registers are managed by the host. Each time the host completes processing of transmit/receive indication, it writes the start address of the next indication to update the MTA register. The initial values of these registers are undefined after reset. The initial values written by the host to these registers after reset are the same as those of the MSL registers. The host sets the same values as those of the MSL registers to these registers for initialization after reset.

31 - 16	15 - 0
- 0 -	MTA0
- 0 -	MTA1
- 0 -	MTA2
- 0 -	MTA3

(17) MWA0 through MWA3 (Mailbox Write Pointer) (4 registers)

Address : 80H - MWA0: Mailbox 0
84H - MWA1: Mailbox 1
88H - MWA2: Mailbox 2
8CH - MWA3: Mailbox 3

Access mode:Read/write

MWA0 through MWA3 store the low-order 16 bits of the write pointers of the four mailboxes. These registers are managed by the μPD98409. Each time the μPD98409 has stored indication, it increments and updates the addresses of these registers. The initial values of these registers are undefined after reset. The host sets the same values as those of the MSL registers to these registers for initialization after reset.

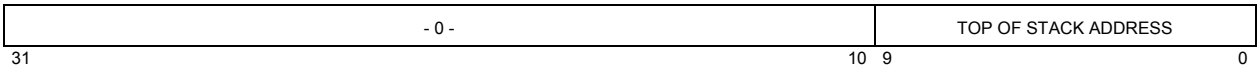
31 - 16	15 - 0
- 0 -	MWA0
- 0 -	MWA1
- 0 -	MWA2
- 0 -	MWA3

7.3 Indirect Address Registers

(1) TOS (Top Of Stack)

The host assigns the start address (200H) of a free block pool of the control memory to this register. After that, this register is managed by the μPD98409 and functions as a pointer that indicates the free block that can be allocated as a VC table next. The initial value of this register is 0 after reset.

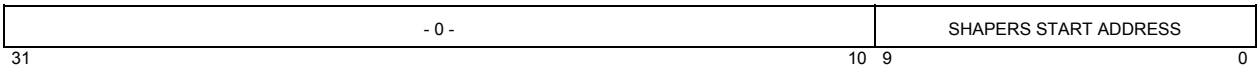
Address : 40100H
Access mode : Read/write



(2) SMA (Shaper Pointer Entry Start Address)

SMA is a register to which the host assigns the start address (080H) of a shaper pointer area of the control memory. The initial value of this register is 0 after reset.

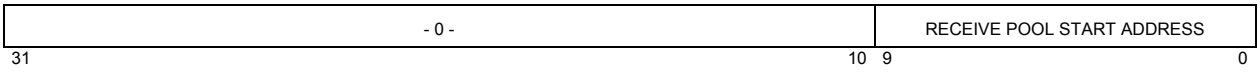
Address : 40200H
Access mode : Read/write



(3) PMA (Receive Free Buffer Pool Pointer Start Address)

PMA is a register to which the host assigns the start address (040H) of a receive free buffer pool pointer area of the control memory. The initial value of this register after reset is 0.

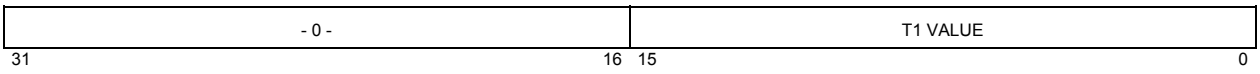
Address : 40201H
Access mode : Read/write



(4) T1R (T1 Time)

The user assigns the time permitted to receive one packet to T1R. Of the 32-bit value of this register, only the high-order 16 bits are defined. The time is set in multiple of 65536 system clock cycle. The low-order 16 bits are 0000H, and they are not used. The initial value after reset is “FFFFH”.

Address : 40300H
Access mode : Read/write



(5) VRR (VPI/VCI Reduction Register)

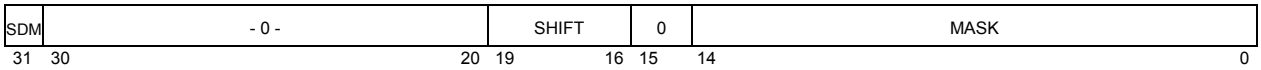
The VRR register is used to set a 4-bit “SHIFT” parameter and a 15-bit “MASK” parameter used to reduce the 24 patterns of the received VPI/VCI to an internal 15-bit logic code. For the conversion algorithm, refer **to 5.5.4 Setting of receive look-up table**.

The MSB bit of this register is assigned to a bit of the “global shutdown” command. When the host writes 1 to this bit, the μPD98409 stops all reception processing under execution, sets the RD bit of the GSR register to 1, and issues an interrupt, if not masked.

The initial value of this register after reset is “00007FFFH’.

Address : 40301H

Access mode : Read/write



SMD : Global shutdown bit

0: Normal operation

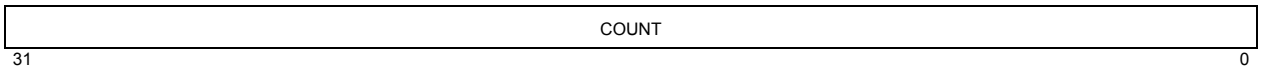
1: Executes shutdown

(6) TSR (Time Stamp Register)

The TSR register is a 32-bit counter register that is counted up by the μPD98409 in the cycle of the system clock. It is used to indicate the packet arrival start time of the T1 timer function. The initial value of this register after reset is 0. However, the μPD98409 immediately starts counting up after this register has been reset.

Address : 40302H

Access mode : Read/write

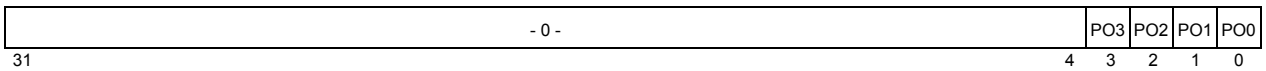


(7) GPOR (general-purpose output port register)

This register sets the level to be output by the general-purpose output port (PO3 through PO0) of the μPD98409. The PO3 through PO0 bits of the GPOR register correspond to the PO3 through PO0 pins. When these bits are cleared to “0”, the corresponding port pins output a low level; when the bits are set to “1”, the pins output a high level.

Address : 40500H

Access mode : Read/write



(8) Scheduler registers

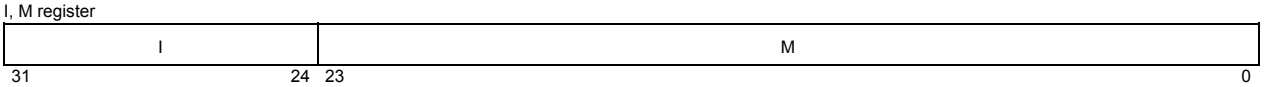
Five registers are assigned to each of the 2 shapers. The host sets a parameter that determines the average rate and peak rate of the shaper to be used to these registers. The μPD98409 stores the variables used for calculation of traffic control, and flags indicating the status of a shaper in these registers.

Cautions 1. Clear the enable bit (E) of the scheduler register only when the variable and flag used by the μPD98409 are cleared to 0 and after the parameters (I, M, P, C, and Priority) have been set.

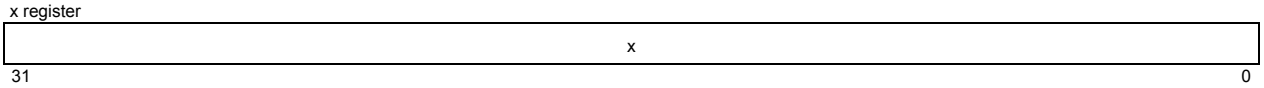
2. The contents of the scheduler registers cannot be changed if the corresponding shaper is active because the A bit = 1. A shaper becomes inactive when no VC table is linked. Before changing parameters I, M, P, and C, confirm that the shaper is inactive (A bit = 0) and clear the enable bit (E) to 0. At this time, clear all the variables and flags used by the μPD98409 to 0.

Address (H)	Register	Access Mode	Function
40000 - 40001	I, M	Read/write	I and M parameters of schedulers 0 through 1
40010 - 40011	x	Read/write	x value of schedulers 0 through 1
40020 - 40021	y	Read/write	y value of schedulers 0 through 1
40030 - 40031	P, C, p, c	Read/write	P and C parameters, and p and c values of schedulers 0 through 1
40040 - 40041	Pin & Status	Read/write	Priority and status of schedulers 0 through 1

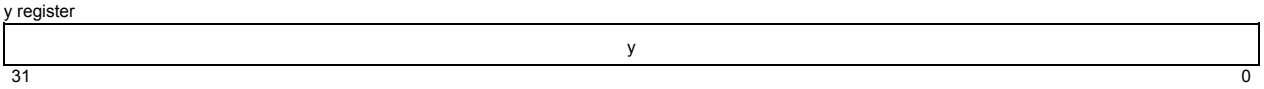
I, M register



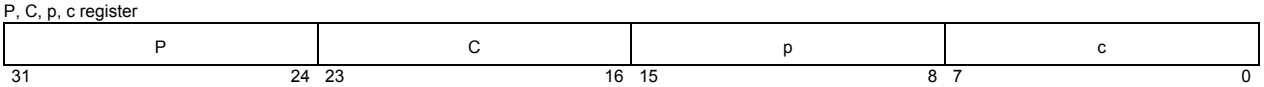
x register



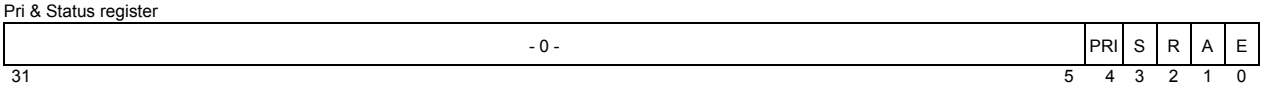
y register



P, C, p, c register



Pri & Status register



Field	Function	Value after Reset
I, M	Average rate. Sets average rate parameter in cell units. Sets I/M to indicate I cells transmitted per M cell.	0
x, y	Used by the μPD98409 to store temporary parameters.	0
P	Peak cell rate. Minimum time difference between two successive cells in the channel linked to the shaper. Set in cell units.	Undefined
C	Credit. Sets maximum number of credits shaper can accumulate. Maximum burst size is determined by parameters I, M, P, and C.	Undefined
p	Used by the μPD98409 to store temporary parameters.	0
c	Used by the μPD98409 to store temporary parameters.	0
PRI	Priority. Sets priority to be allocated to the shaper. 0 is the highest priority, and 1 is the lowest.	Undefined
S	Scan flag. Used by the μPD98409 to manage the shaper status. The user must not change the value of this field.	0
R	Round-robin flag. Used by the μPD98409 to manage the shaper status. The user must not change the value of this field.	0
A	Active flag. Used by the μPD98409 to manage the shaper status. 1 - Shaper is active. 0 - Shaper is inactive. This bit is set to 1 by the host only when the unassigned cell generator function is used.	0
E	Enable. The host sets this field to '1' once when this shaper is used. 1 - Enables shaper 0 - Disables shaper	0

[MEMO]

CHAPTER 8 JTAG BOUNDARY SCAN

The μPD98409 has a JTAG boundary scan circuit.

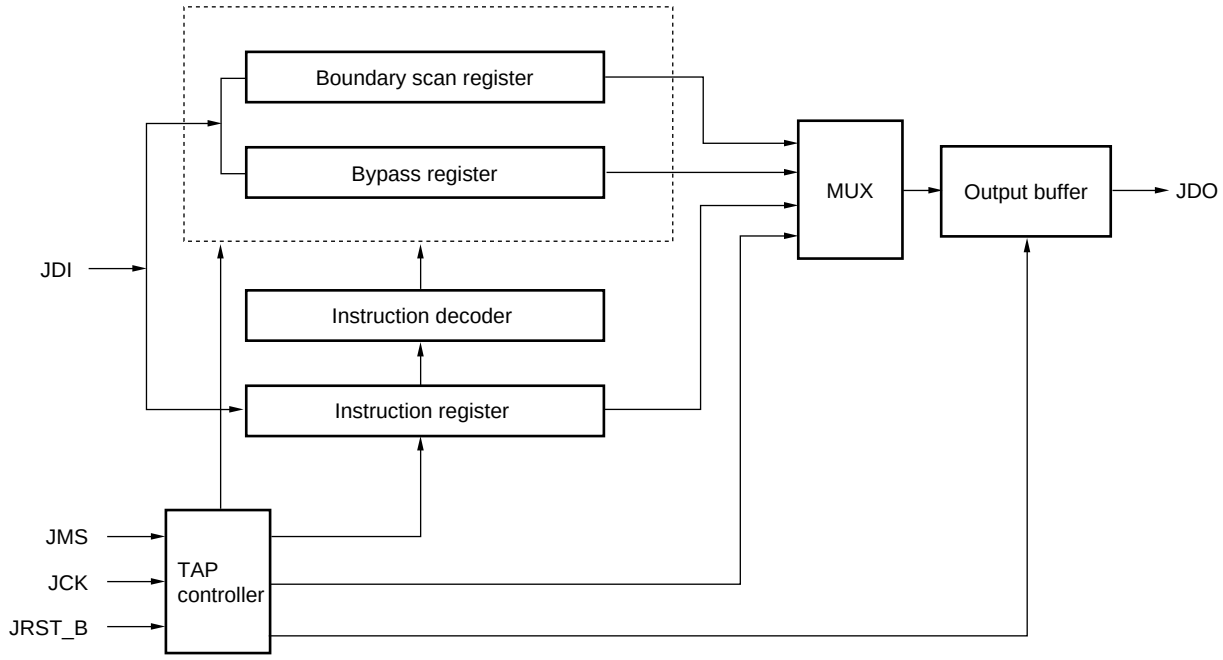
8.1 Features

- Conforms to IEEE1149.1 JTAG Boundary Scan Standard.
- Three registers dedicated to boundary scan
  - Instruction register
  - Bypass register
  - Boundary scan register
- Two instructions supported
  - BYPASS instruction
  - EXTEST instruction
  - SAMPLE/PRELOAD instruction
- Five pins dedicated to boundary scan (5 pins)
  - JCK (JTAG Clock)
  - JMS (JTAG Mode Select)
  - JDI (JTAG Data Input)
  - JDO (JTAG Data Output)
  - JRST_B (JTAG Reset)

8.2 Internal Configuration of Boundary Scan Circuit

Figure 8-1 shows the block diagram of the internal JTAG boundary scan circuit of the μPD98409.

Figure 8-1. Block Diagram of Boundary Scan Circuit



8.2.1 Instruction register

The instruction register consists of a 2-bit shift register and writes instruction data from the JDI pin. The register and instruction are selected by this instruction data.

8.2.2 TAP (Test Access Port) controller

The TAP controller changes operating state by latching the signal of the JMS pin at the rising edge of the clock input to the JCK pin.

8.2.3 Bypass register

The bypass register consists of a 1-bit shift register connected between the JDI and JDO pins when the TAP controller is in Shift-DR state. If this register is selected while the TAP controller is in Shift-DR state, data is shifted to the JDO pin at the rising edge of the clock input to the JCK pin.

When this register is selected, the operation of the JTAG boundary circuit does not influence on the operation of the μPD98409.

8.2.4 Boundary scan register

The boundary scan register is located between an external pin of the μPD98409 and internal logic circuit. When this register is selected, data is latched or loaded by the instruction of the TAP controller.

If this register is selected while the TAP controller is in Shift-DR state, data is output to the JDO pin starting from the LSB at the falling edge of the clock input to the JCK pin.

8.3 Pin Function

8.3.1 JCK (JTAG Clock) pin

The JCK pin is used to supply a clock signal to the JTAG boundary scan circuit (such as the bypass register, instruction register, and TAP controller. This clock signal is isolated so as not to be supplied to the other internal circuits of the μPD98409.

8.3.2 JMS (JTAG Mode Select) pin

Input to the JMS signal is latched at the rising edge of the clock input to the JCK pin and defines the operation of the TAP controller.

8.3.3 JDI (JTAG Data Input) pin

The JDI pin is an input pin that inputs data to the JTAG boundary scan circuit register.

8.3.4 JDO (JTAG Data Output) pin

The JDO pin is an output pin that outputs data from the JTAG boundary scan circuit. This pin changes its output at the falling edge of the clock input to the JCK pin. This pin is a three-state output pin and is controlled by the TAP controller.

8.3.5 JRST_B (JTAG Reset) pin

This pin asynchronously initializes the TAP controller. This reset signal sets the μPD98409 in the normal operation mode and the boundary register in non-operation state.

8.4 Operation Description

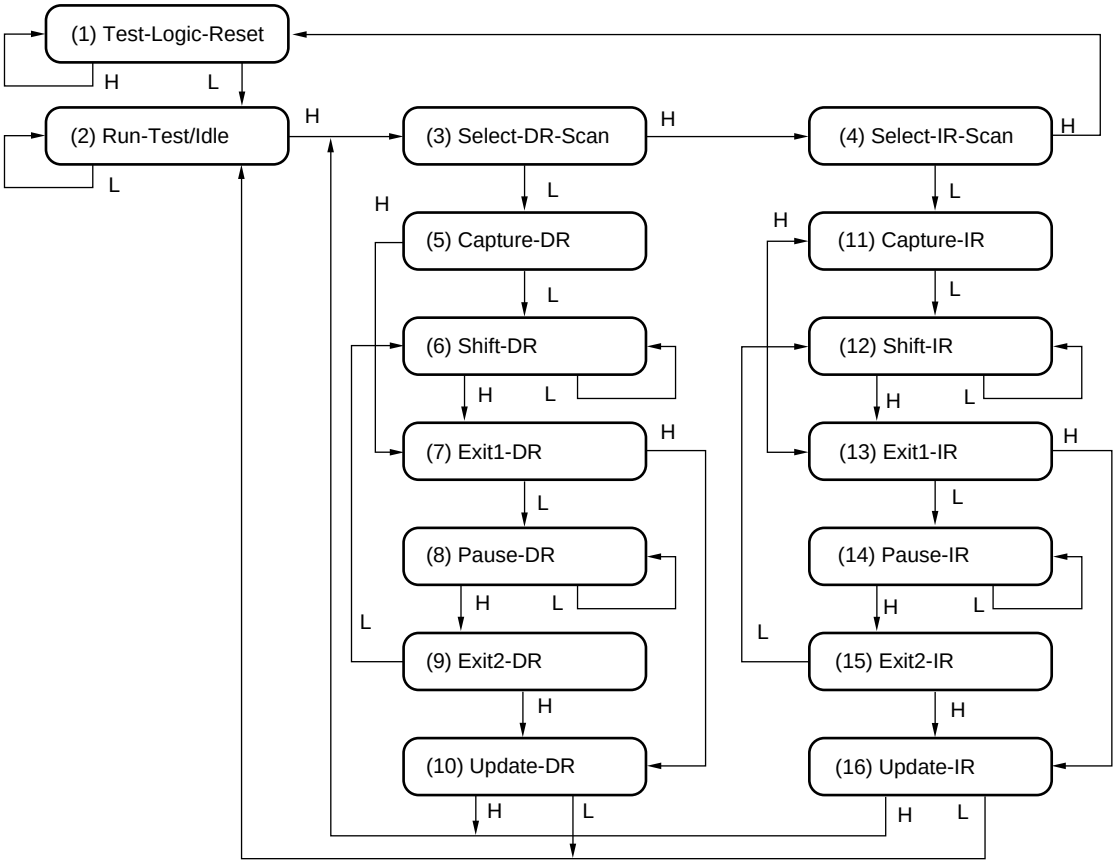
8.4.1 TAP controller

The TAP controller is a circuit having 16 states synchronized with changes of the JMS and JCK pins. Its operation is specified by IEEE Standard 1149.1.

8.4.2 TAP controller state

Figure 8-2 shows the state transition of the TAP controller. The state of the TAP controller is determined depending on the state of the JMS pin signal input at the rising edge of the clock input to the JCK pin. The operations of the instruction register, boundary scan register, and bypass register change at the rising or falling edge of the clock input to the JCK pin. (Refer to **Figure 8-3**).

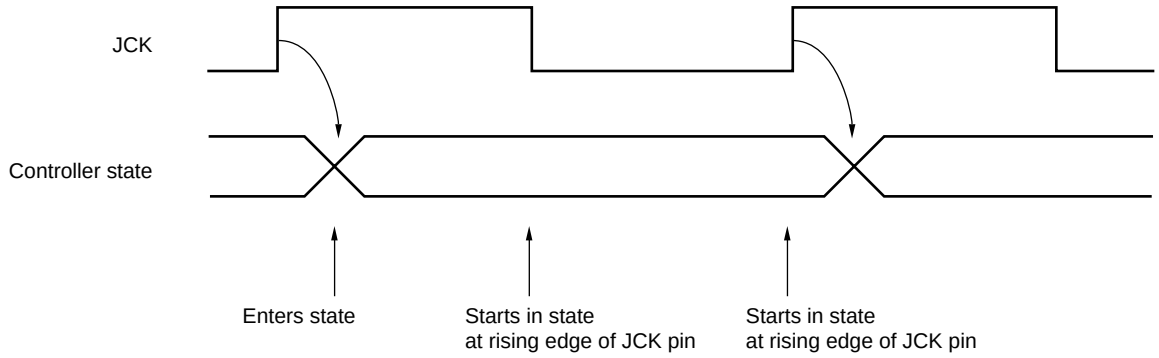
Figure 8-2. State Transition of TAP Controller



Remarks 1. "H" and "L" of the arrows indicating state transition in the above figure indicate the state of the JMS pin at the rising edge of the clock input to the JCK pin.

2. Numbers in ( ) in the above figure correspond to the explanation below.

Figure 8-3. Operation Timing in Controller State



(1) Test-Logic-Reset

The boundary scan circuit performs no operation on the μPD98409. Therefore, it does not affect the system logic of the μPD98409. This is because the bypass instruction is stored to the instruction register and executed on initialization. The TAP controller enters the Test-Logic-Reset state if the JMS pin holds the high level for the duration of at least five rising edges of the JCK pin signal, regardless of the current state of the controller. The TAP controller holds this state for the duration in which the JMS pin signal high.

If the TAP controller must be in the Test-Logic-Reset state, the controller returns to the original Test-Logic-Reset state even if a low-level signal is input once to the JMS pin by mistake (due to, for example, the influence of the external interface), if the JMS pin signal holds its high level status for the duration of three rising edges of the JCK pin signal.

The operation of the test logic does not hinder the logic operation of the μPD98409 due to the above error.

When the TAP controller exits from the Test-Logic-Reset state, the controller enters the Run-Test/Idle state. In this state, no operation is performed because the current instruction is set by the operation of the bypass register. The logic operation of the JTAG boundary scan circuit is inactive even in the Select-DR-Scan and Select-IR-Scan states.

(2) Run-Test/Idle

The TAP controller is in this state during scan operation (in Select-DR-Scan or Select-IR-Scan state). Once the controller has entered this state and if the JMS pin signal holds the low level, the controller remains in this state. The controller enters the Select-DR-Scan state if the JMS pin signal holds high level at one rising edge of the JCK pin signal.

All the test data registers (boundary register and bypass register) selected by the current instruction hold the previous status (Idle). While the TAP controller is in this state, the instruction does not change.

(3) Select-DR-Scan

This is a temporary boundary scan state. The boundary scan register and bypass register selected by the current instruction hold the previous state.

If the JMS pin signal is held low at the rising edge of the JCK pin signal while the TAP controller is in this state, the controller enters the Capture-DR state, and scan sequence to the selected registers is started.

If the JMS pin signal is held high at the rising edge of the JCK pin signal, the TAP controller enters the Select-IR-Scan state. While the controller is in this state, the instruction does not change.

(4) Select-IR-Scan

This is a temporary boundary scan state. The boundary scan register and bypass register selected by the current instruction hold the previous state.

If the JMS pin signal is held low at the rising edge of the JCK pin signal while the TAP controller is in this state, the controller enters the Capture-IR state, and scan sequence to the selected registers is started.

If the JMS pin signal is held high at the rising edge of the JCK pin signal, the TAP controller enters the Test-Logic-Reset state. While the controller is in this state, the instruction does not change.

(5) Capture-DR

In this controller state, data is loaded to the boundary scan register selected by the current instruction in parallel at the rising edge of the JCK pin signal (in this case, data is input from the input pin of each device to the corresponding boundary scan register). While the TAP controller is in this state, the instruction does not change.

If the TAP controller is in this state at the rising edge of the JCK pin signal, the controller enters the following state:

- If the JMS pin signal is held high : Exit1-DR state
- If the JMS pin signal is held low : Shift-DR state

(6) Shift-DR

In this controller state, JDI and JDO are connected (at either of the boundary scan register of bypass register) by the current instruction. The shift data is shifted one state at a time toward the serial output direction at each rising edge of the JCK pin signal.

The boundary scan register or bypass register selected by the current instruction holds the previous status without change if the controller is not on the serial bus (not in the Shift-DR state). While the controller is in this state, the instruction does not change.

If the TAP controller is in this state at the rising edge of the JCK pin signal, the controller enters the following state:

- If the JMS pin signal is held high : Exit1-DR state
- If the JMS pin signal is held low : Shift-DR state

(7) Exit1-DR

This is a temporary controller state. If the JMS pin signal is held high at the rising edge of the JCK pin signal with the TAP controller in this state, the controller enters the Update-DR state. This ends the scan process.

If the JMS pin signal is held low at the rising edge of the JCK pin signal, the TAP controller enters the Pause-DR state.

Both the bypass register and boundary scan register selected by the current instruction hold the previous status without change. While the TAP controller is in this state, the instruction does not change.

(8) Pause-DR

In this controller state, shifting between JDI and JDO connected by either the bypass register or boundary scan register is temporarily stopped. These registers selected by the current instruction hold the previous state without change.

The TAP controller remains in this state while the JMS pin signal is low. If the JMS pin signal is held high at the rising edge of the JCK pin signal, the TAP controller enters the Exit2-DR state. While the TAP controller is in this state, the instruction does not change.

(9) Exit2-DR

This is a temporary controller state. If the JMS pin signal is held high at the rising edge of the JCK pin signal with the TAP controller in this state, the controller enters the Update-DR state. This ends the scan process.

If the JMS pin signal is held low at the rising edge of the JCK pin signal, the TAP controller enters the Shift-DR state.

Both the bypass register and boundary scan register selected by the current instruction hold the previous status without change. While the TAP controller is in this state, the instruction does not change.

(10) Update-DR

The boundary scan register has a parallel output latch to prevent changes in parallel output (while shifted to the shift register path concatenated) by certain instructions (for example, EXTEST instruction).

In the Update-DR controller state, data is latched from the shift register to the parallel output latch of this register at the falling edge of the JCK pin signal.

The data retained latched to the parallel output latch changes depending on this controller state (the data does not change with the other controller states).

The previous states of all the shift register of the boundary scan register selected by the current instruction are retained.

While the TAP controller is in this state, the instruction does not change.

If the JMS pin signal is held high at the rising edge of the JCK pin signal with the TAP controller in this state, the controller enters the Select-DR-Scan state.

If the JMS pin signal is held low at the rising edge of the JCK signal, the TAP controller enters the Run-Test/Idle state.

(11) Capture-IR

In this controller state, the shift register loads the pattern of a fixed logic value “01(binary)” to the instruction register at the rising edge of the JCK pin signal.

The previous states of both the bypass register and boundary scan register selected by the current instruction are retained without change.

While the TAP controller is in this state, the instruction does not change.

If the JMS pin signal is held high at the rising edge of the JCK pin signal while the controller is in this state, the controller enters the Exit1-IR state.

If the JMS pin signal is held low at the rising edge of the JCK pin signal, the TAP controller enters the Shift-IR state.

(12) Shift-IR

In this controller state, JDI and JDO are connected by the shift register in the instruction register. The shift data is shift one state toward the serial output direction at each rising edge of the JCK pin signal.

The boundary scan register or bypass register selected by the current instruction holds the previous state without change.

While the TAP controller is in this state, the instruction does not change.

If the JMS pin signal is held high at the rising edge of the JCK pin signal with the TAP controller in this state, the controller enters the Exit1-IR state. If the JMS pin signal is held low, the controller remains the Shift-IR state.

(13) Exit1-IR

This is a temporary controller state. If the JMS pin signal is held high at the rising edge of the JCK pin signal, the TAP controller enters the Update-IR state. This ends the scan process.

If the JMS pin signal is held low at the rising edge of the JCK pin, the TAP controller enters the Pause-IR state.

Both the bypass register and boundary scan register selected by the current instruction retain their states without change.

While the TAP controller is in this state, the instruction does not change.

(14) Pause-IR

In this controller state, shift of the instruction register is temporarily stopped. The bypass register and boundary scan register selected by the current instruction hold the previous state without change.

While the TAP controller is in this state, the instruction does not change. The instruction register holds the current state.

While the JMS pin signal is low, the TAP controller remains in this state. If the JMS pin signal is held high at the rising edge of the JCK pin signal, the TAP controller enters the Exit2-IR state.

(15) Exit2-IR

This is a temporary controller state. If the JMS pin signal is held high at the rising edge of the JCK pin signal, the TAP controller enters the Update-IR state. This ends the scan process.

If the JMS pin signal is held low at the rising edge of the JCK pin, the TAP controller enters the Shift-IR state.

Both the bypass register and boundary scan register selected by the current instruction retain their states without change.

While the TAP controller is in this state, or while the instruction register holds the current state, the instruction does not change.

(16) Update-IR

In this controller state, the instruction shifted to the instruction register is latched to the parallel output latch from the shift register path at the falling edge of the JCK pin signal. Once a new instruction has been latched, it is used as the current instruction.

The bypass register or boundary scan register selected by the current instruction holds the previous state.

If the JMS pin signal is held high at the rising edge of the JCK pin signal while the TAP controller is in this state, the TAP controller enters the Select-DR-Scan state.

If the JMS pin signal is held low at the rising edge of the JCK pin signal, the TAP controller enters the Run-Test/Idle state.

The Pause-DR controller state in (8) and Pause-IR controller state in (14) temporarily stop shifting of data in the bypass register, the boundary scan register, or instruction register.

8.5 TAP Controller Operation

The TAP controller operates as follows.

The state of the controller is changed by either of (1) and (2) below.

(1) Rising edge of JCK pin signal
(2) JRST_B pin input

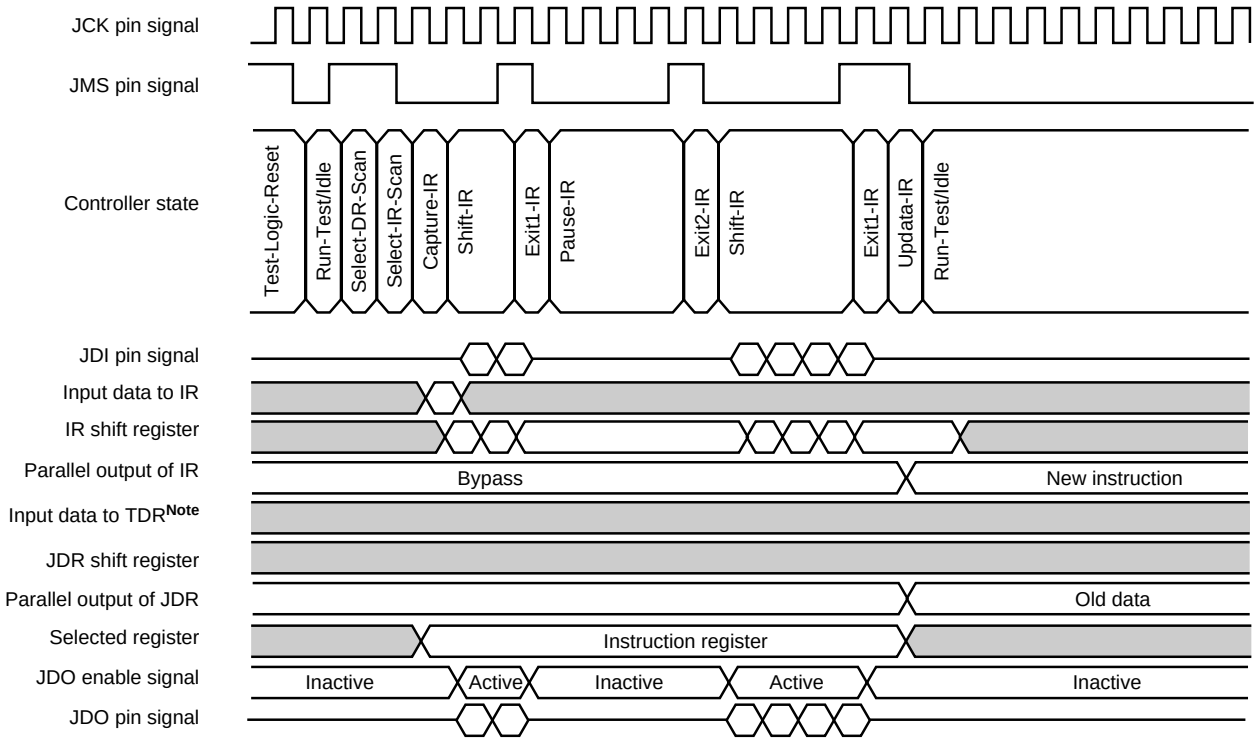
The TAP controller generates signals that control the operations of the bypass register, boundary scan register, and instruction register defined by the IEEE1149.1 JTAG Boundary Scan Standard (refer to **Figures 8-4** and **8-5**).

The JDO pin output buffer and the peripheral circuit that selects a register whose contents are to be output to the JDO pin are controlled as shown in Table 8-1. The JDO pin defined in this table changes at the falling edge of the JCK pin signal after it has entered each state.

Table 8-1. Operation in Each Controller State

Controller State	Selected Register to Be Driven to JDO Pin	JDO Pin Driver
Test-Logic-Reset	Undefined	High impedance
Run-Test/Idle		
Select-DR-Scan		
Select-IR-Scan		
Capture-IR		
Shift-IR	Instruction register	Active
Exit1-IR	Undefined	High impedance
Pause-IR		
Exit2-IR		
Update-IR		
Capture-DR		
Shift-DR	Data register (boundary scan register, bypass register)	Active
Exit1-DR	Undefined	High impedance
Pause-DR		
Exit2-DR		
Update-DR		

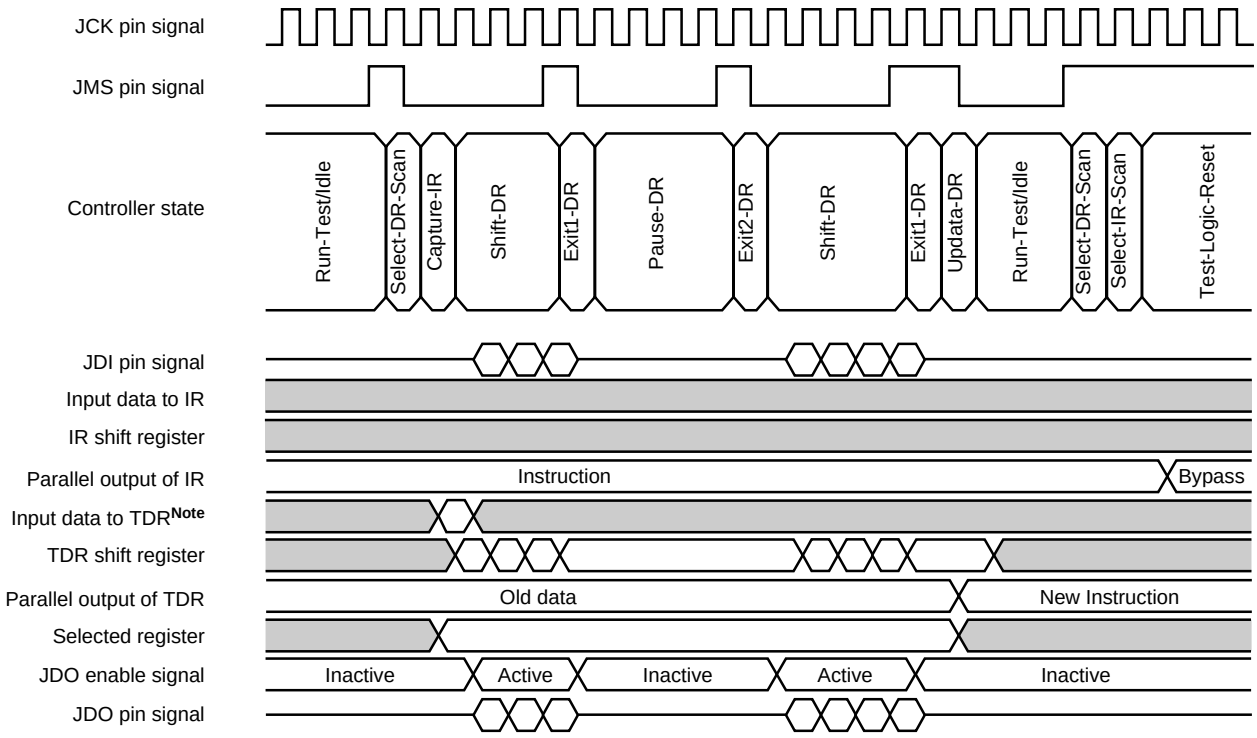
Figure 8-4. Operation of Test Logic (instruction scan)



Note TDR (Test Data Register): Boundary scan register and bypass register

Remark ▨ : Don't care or undefined

Figure 8-5. Operation of Test Logic (data scan)



Note TDR (Test Data Register): Boundary scan register and bypass register

Remark : Don’t care or undefined

8.6 Initializing TAP Controller

The TAP controller is initialized as follows:

(1) The TAP controller is not initialized by the operation of system input such as system reset.
(2) The TAP controller enters the Test-Logic-Reset controller state at the fifth rising edge of the JCK pin signal (while the JMS pin signal is held high).
(3) The TAP controller asynchronously enters the Test-Logic-Reset state when the JRST_B signal is input.

8.7 Instruction Register

This register is defined as follows (refer to **8.2 Internal Configuration of Boundary Scan Circuit**).

(1) The instruction shifted and input to the instruction register is latched so that it changes only in the Update-IR and Test-Logic-Reset controller states.
(2) Data is not inverted since it has been serially input to the instruction register until it is serially output.
(3) A fixed binary pattern data “01” (with LSB (Least Significant Bit) being “1”) is loaded to this register cell in the Capture-IR controller state.
(4) A fixed binary pattern data “01” (with LSB being “1”) is loaded to this register cell in the Test-Logic-Reset controller state.
(5) While this register is read, data is output from the JDO pin, starting from the LSB to the MSB (Most Significant Bit), at each falling edge of the JCK pin signal.

The JTAG boundary scan circuit of the μPD98409 can support only the following two instructions depending on the data set to the instruction register.

- BYPASS instruction
- EXTEST instruction
- SAMPLE/PRELOAD instruction

Instruction Register		Supported Instruction
D1	D0	
0	0	EXTEST instruction
0	1	SAMPLE/PRELOAD instruction
1	0	Unused (BYPASS instruction)
1	1	BYPASS instruction

8.7.1 BYPASS instruction

This instruction is specified by instruction data “11” or “10”. This instruction is used to select only the bypass register (to access between the JDI and JDO pins serially) in the Shift-DR controller state.

While this instruction is selected, the operation of the JTAG boundary scan circuit does not affect the operation of the μPD98409.

This bypass instruction is selected while the TAP controller is in the Test-Logic-Reset state.

8.7.2 EXTEST instruction

This instruction is specified by instruction data “00”. In the Shift-DR controller state, this instruction is used to select the boundary scan register of serial access between the JDI and JDO instructions.

- While this instruction is selected:

  The states of all the signals driven from the system output pins are completely defined by the data shifted to the boundary scan register. In the Update-DR controller state, the states of all the signals are changed only by the falling edge of the JCK pin signal.

  The states of all the signals input from the system input pins are loaded to the boundary scan register at the rising edge of the JCK pin signal while the TAP controller is in the Capture-DR state.

8.7.3 SAMPLE/PRELOAD instruction

This instruction is specified by instruction data “01”. This instruction is used to implement two functions: SAMPLE and PRELOAD.

8.7.4 Boundary scan data bit definition

If requested, NEC can provide a reference BSDL (Boundary Scan Description Language) file for the μPD98409. Contact the NEC Semiconductor Technical Hotline written in the Facsimile Message given at the end of this document.

[MEMO]