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

μPD98401A

(NEASCOT-S15™)

ATM SAR CHIP

Document No. S12054EJ4V0UM00 (4th edition)
Date Published June 2002 NS CP(K)

[MEMO]

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note:

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

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note:

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

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note:

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

NEASCOT-S15 is a trademark of NEC Corporation.

- **The information in this document is current as of June, 2002. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC's data sheets or data books, etc., for the most up-to-date specifications of NEC semiconductor products. Not all products and/or types are available in every country. Please check with an NEC 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 prior written consent of NEC. NEC assumes no responsibility for any errors that may appear in this document.
- NEC 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 semiconductor 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 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 customer's equipment shall be done under the full responsibility of customer. NEC assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC endeavours to enhance the quality, reliability and safety of NEC semiconductor 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 semiconductor products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment, and anti-failure features.
- NEC semiconductor products are classified into the following three quality grades:
  "Standard", "Special" and "Specific". The "Specific" quality grade applies only to semiconductor products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of a semiconductor product depend on its quality grade, as indicated below. Customers must check the quality grade of each semiconductor 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 semiconductor products is "Standard" unless otherwise expressly specified in NEC's data sheets or data books, etc. If customers wish to use NEC semiconductor products in applications not intended by NEC, they must contact an NEC sales representative in advance to determine NEC's willingness to support a given application.

  (Note)

  (1) "NEC" as used in this statement means NEC Corporation and also includes its majority-owned subsidiaries.

  (2) "NEC semiconductor products" means any semiconductor product developed or manufactured by or for NEC (as defined above).

M8E 00. 4

Major Revisions in this Edition

Page	Description
p.26	Addition of Remark in **2.2.5 JTAG boundary scan pins**

The mark ★ shows major revised points.

PREFACE

Readers

This manual is intended for user engineers who wish to understand the functions of the μPD98401A and design application systems using it.

Purpose

This manual explains the hardware functions of the μPD98401A in the following organization.

Organization

This manual consists of the following chapters.

- Overview
- Pin function
- Functional outline
- Interfaces
- Operations of μPD98401A
- 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 μPD98401A
→ Read this manual in the order of Table of Contents.

For the electrical characteristics of the μPD98401A
→ 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.

- Brochure: S11847E
- Data sheet: S12100E

CONTENTS

LIST OF FIGURES (1/3)

LIST OF FIGURES (2/3)

LIST OF FIGURES (3/3)

LIST OF TABLES

CHAPTER 1 OVERVIEW

The μPD98401A is a high-performance SAR chip that segments and reassembles ATM cells. This chip can interface with an ATM network when it is included in a workstation, computer, front-end processor, network hub, or router. The μPD98401A conforms to the ATM Forum Recommendation, and provides the functions of the AAL-5 SAR sublayer and ATM layer.

The μPD98401A is compatible with its predecessor, μPD98401, in terms of hardware and software.

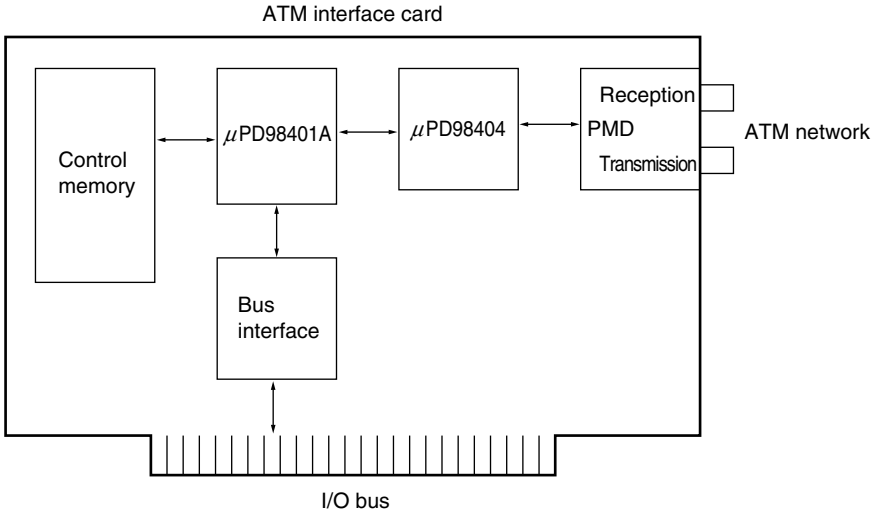
1.1 Features

- Conforms to ATM Forum
- AAL-5 SAR sublayer and ATM layer functions
- Hardware support of AAL-5 processing
- Processing of non-AAL-5 traffic (AAL-3/4 cell, OAM cell, RM cell) by software with raw cell processing function
- Hardware support of generation/verification of CRC-10 for non-AAL-5 traffic
- Supports up to 32K virtual channels (VC)
- Provided with 16 traffic shapers that carry out transmission scheduling (control of average rate/peak rate) so as to set different transmission rate for each VC
- Interface and commands for controlling PHY device
- Employs “UTOPIA interface” as cell transfer interface with PHY device
  - Octet-level handshake
  - Cell-level handshake
- 32-bit general-purpose bus interface
- High-speed DMAC (supports 1-, 2-, 4-, 8-, 12-, and 16-word burst)
- JTAG boundary scan test function (IEEE1149.1)
- CMOS process
- +5 V single power source

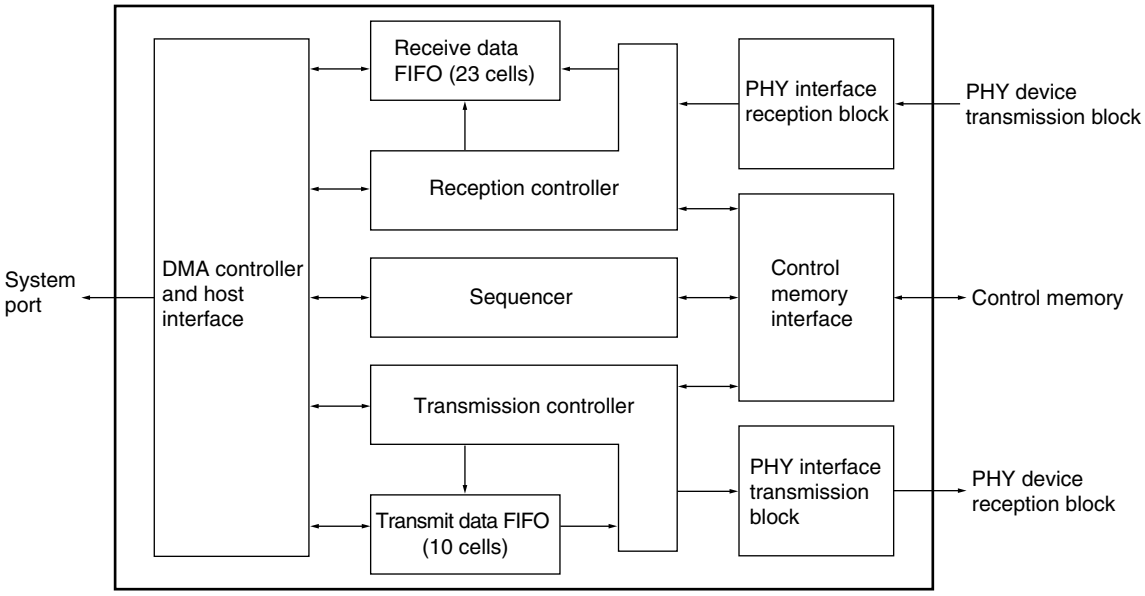
1.2 Ordering Information

Part Number	Package
μPD98401AGD-MML	208-pin plastic QFP (fine pitch) (28 × 28)

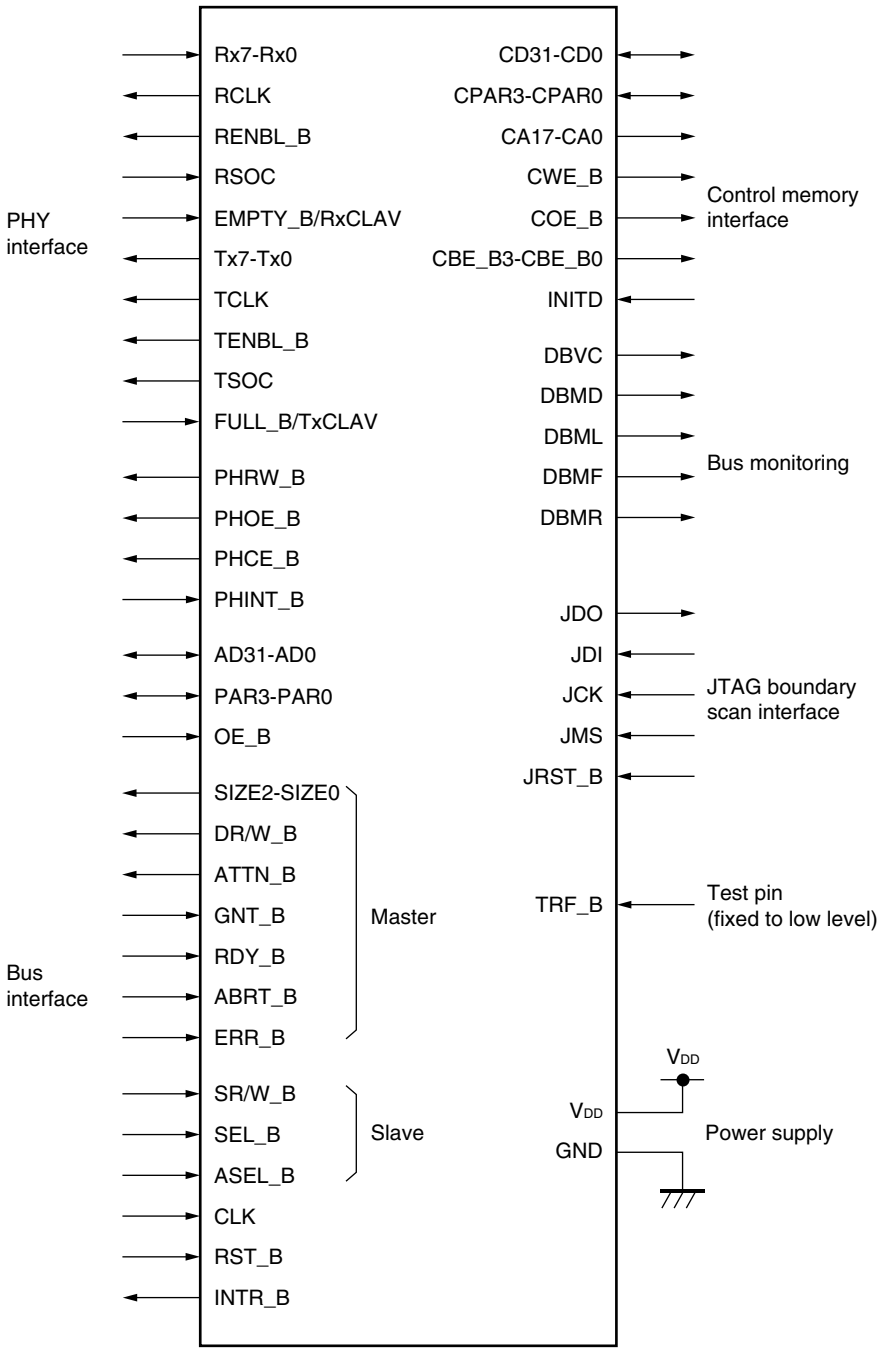
1.3 Example of System Configuration



1.4 Block Diagram



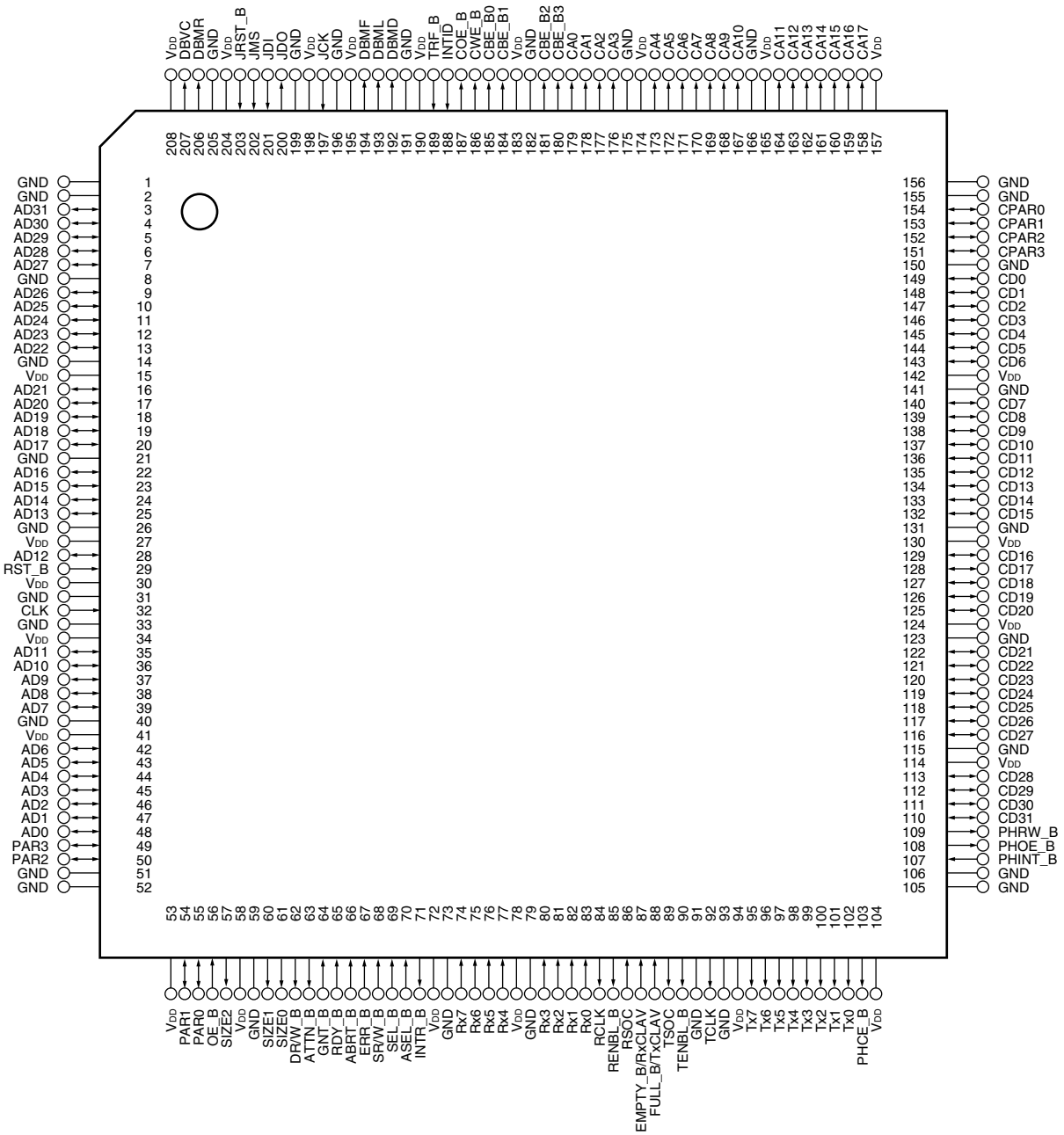
1.5 Pin Configuration



2.1 Pin Configuration (Top View)

208-pin plastic QFP (fine pitch) (28 × 28)

μPD98401AGD-MML



ABRT_B	:	Abort
AD31_AD0	:	Address/Data
ASEL_B	:	Slave Address Select
ATTN_B	:	Attention/Burst Frame
CA17-CA0	:	Control Memory Address
CBE_B3_CBE_B0	:	Local Port Byte Enable
CD31-CD0	:	Control Memory Data
CLK	:	Clock
COE_B	:	Control Memory Output Enable
CPAR3-CPAR0	:	Control Memory Parity
CWE_B	:	Control Memory Write Enable
DBMD	:	DMA Bus Monitor Data
DBMF	:	DMA Bus Monitor First
DBML	:	DMA Bus Monitor Last
DBMR	:	DMA Bus Monitor Remaining
DBVC	:	DMA Bus Monitor VC
DR/W_B	:	DMA Read/Write
EMPTY_B/RxCLAV	:	PHY Output Buffer Empty
ERR_B	:	Error
FULL_B/TxCLAV	:	PHY Buffer Ful
GND	:	Ground
GNT_B	:	Grant
INITD	:	Initialization Disable
INTR_B	:	Interrupt
JCK	:	JTAG Test Pin
JDI	:	JTAG Test Pin
JDO	:	JTAG Test Pin
JMS	:	JTAG Test Pin
JRST_B	:	JTAG Test Pin
OE_B	:	Output Enable
PAR3-PAR0	:	Bus Parity
PHCE_B	:	PHY Chip Enable
PHINT_B	:	PHY Interrupt
PHOE_B	:	PHY Output Enable
PHRW_B	:	PHY Read/Write
RCLK	:	Receive Clock
RDY_B	:	Target Ready
RENBL_B	:	Receive Enable
RSOC	:	Receive Start Cell
RST_B	:	Reset
Rx7-Rx0	:	Receive Data Bus
SEL_B	:	Slave Select
SIZE2-SIZE0	:	Burst Size
SR/W_B	:	Slave Read/Write
TCLK	:	Transmit Clock
TENBL_B	:	Transmit Enable
TSOC	:	Transmit Start of Cell
TRF_B	:	Delay Select
Tx7-Tx0	:	Transmit Data Bus
V_{DD}	:	Power Supply

2.2 Pin Function

The μPD98401A is a package having 208 pins, of which 152 pins are function pins and 56 pins are V_{DD} and GND pins. 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.

2.2.1 PHY device interface pin

PHY device interfaces include a UTOPIA interface through which the μPD98401A transfers ATM cells with a PHY device, and a PHY control interface by which the μPD98401A controls the PHY device.

(1) UTOPIA interface

(1/2)

Pin Name	Pin No.	I/O	I/O Level	Function
Rx7-Rx4 Rx3-Rx0	74 - 77 80 - 83	I	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 μPD98401A loads data in at the rising edge of RCLK.
RSOC	86	I	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	85	O	CMOS	Receive Enable. The RENBL_B signal indicates to a PHY device that the μPD98401A is ready to receive data in the next clock cycle. This signal goes high during and after reset.
EMPTY_B/ RxCLAV	87	I	TTL	PHY Output Buffer Empty/Rx Cell Available. This signal notifies the μPD98401A 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	84	O	CMOS	Receive Clock. This is a synchronization clock used to transfer cell data with the PHY cell device at the recieve side. The system clock input to the CLK pin is output from this pin as is, immediately after reset.
Tx7-Tx0	95 - 102	O	CMOS	Transmit Data Bus. Tx7 through Tx0 constitute an 8-bit output bus which outputs transmit data in byte format to a PHY device. The μPD98401A outputs data at the rising edge of TCLK.
TSOC	89	O	CMOS	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	90	O	CMOS	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. This signal remains high during reset, and goes high after reset.
FULL_B/ TxCLAV	88	I	TTL	PHY Buffer Full/Tx Cell Available. The FULL_B signal notifies the μPD98401A that the input buffer is full and that the PHY 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 of data. In the cell-level handshake mode, this signal indicates that the PHY device can receive all the next one cell of data after the current cell has been completely transferred
TCLK	92	O	CMOS	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	109	O	CMOS	PHY Read/Write. The μPD98401A indicates the direction in which the PHY device is controlled, by using PHRW_B. This signal goes low after reset. 1: Read 0: Write
PHOE_B	108	O	CMOS	PHY Output Enable. The μPD98401A enables output from the PHY device by making PHOE_B low
PHCE_B	103	O	CMOS	PHY Chip Enable. The μPD98401A makes PHCE_B low to access a PHY device. This signal goes high after reset.
PHINT_B	107	I	TTL	PHY Interrupt. This is an interrupt input signal from a PHY device. The PHY device indicates to the μPD98401A that it has an interrupt source, by inputting a low level to PHINT_B. This signal goes high after reset.

2.2.2 Bus interface pins

The bus interface is a general-purpose bus interface compatible with most generally used I/O buses (such as PCI, S bus, GIO, and AP bus).

(1/3)

Pin Name	Pin No.	I/O	I/O Level	Function
AD31-AD27 AD26-AD22 AD21-AD17 AD16-AD13 AD12 AD11-AD7 AD6-AD0	3 - 7 9 - 13 16 - 20 22 - 25 28 35 - 39 42 - 48	I/O 3-state	TTL in CMOS out	Address/Data. AD31 through AD0 constitute a 32-bit address/data bus. These pins are I/O pins multiplexing an address bus and a data bus. At the first clock of input/output, AD31 through AD0 transfer an address. They transfer data at the second clock and onward. The AD bus goes into a high-impedance state when the μPD98401A does not access the bus.
PAR3 PAR2 PAR1 PAR0	49 50 54 55	I/O 3-state	TTL in CMOS out	Bus Parity. PAR pins indicate the parity of AD31 through AD0. A parity check mode is set by GMR. Enabling or disabling parity, odd or even parity, and word or byte parity can be specified. If byte parity is specified, PAR3 indicates the parity of AD31 through AD24, and PAR0 indicates the parity of AD7 through AD0. If word parity is specified, PAR3 serves as an input/output pin. It serves as an output pin when an address is output and when data is written, and as an input pin when data is read. When the μPD98401A does not access the bus, PAR3 through PAR0 go into a high-impedance state. Pull up these pins when they are not used.
OE_B	56	I	TTL	Output Enable. When this pin is low, the μPD98401A uses AD31 through AD0 and PAR3 through PAR0 as 3-state I/O pins. These pins go into a high-impedance state while a high level is being input to OE_B. This pin is an option pin. Fix this pin to low level in a system where it is not necessary to forcibly set the bus of the μPD98401A in a high-impedance state by controlling this pin.
SIZE2 SIZE1 SIZE0	57 60 61	O	CMOS	Burst Size. SIZE2 through SIZE0 indicate the size of the current DMA transfer. These pins are used to interface a bus (such as S bus) requiring clear burst size.

SIZE2	SIZE1	SIZE0	Function
0	0	0	1-word transfer
0	0	1	2-word burst
0	1	0	4-word burst
0	1	1	8-word burst
1	0	0	16-word burst
1	0	1	12-word burst
1	1	0	Undefined
1	1	1	Reception side byte alignment

(2/3)

Pin Name	Pin No.	I/O	I/O Level	Function
DR/W_B	62	O	CMOS	DMA Read/Write. DR/W_B indicates the direction of DMA access. 1: Read access 0: Write access This pin is set to 1 after reset.
ATTN_B	63	O	CMOS	Attention/Burst Frame (DMA request). The μPD98401A makes the ATTN_B signal low when it performs a DMA operation. The ATTN_B signal becomes inactive at the rising edge of CLK when the data to be transferred by means of DMA has decreased to 1 word.
GNT_B	64	I	TTL	Grant The GNT_B signal inputs a low level when the bus arbiter grants the μPD98401A use of the bus in response to a DMA request from the μPD98401A. The μPD98401A recognizes that it has been granted use of the bus and starts DMA operation when the GNT_B signal goes low (active). Make sure that the GNT_B signal falls at least one system clock cycle after the rising of the ATTN_B signal. The GNT_B signal must be returned to the high (inactive) level before the μPD98401A makes the ATTN_B signal low (active) to issue the next DMA cycle request.
RDY_B	65	I	TTL	Target Ready. RDY_B indicates to the μPD98401A in the DMA cycle that the target device is ready for input/output. During the DMA read operation of the μPD98401A, the RDY_B signal is made low if valid data is on AD31 through AD0. During the DMA write operation of the μPD98401A, the RDY_B signal is made low if the target device is ready for receiving data. The sampling timing of the RDY_B and ABRT_B signals of the μPD98401A can be advanced by one clock (early mode) by using an internal register (GMR register).
ABRT_B	66	I	TTL	Abort. ABRT_B is used to abort a DMA transfer cycle. If this signal goes low while data is being transferred in a DMA transfer cycle, DMA transfer is aborted in that cycle and the ATTN_B signal is deasserted inactive. After that, the μPD98401A asserts the ATTN_B signal active again, and resumes burst starting from the aborted data. While a low level is input to ABRT_B, the RDY_B signal is ignored. The user can advance the sampling timing of the RDY_B and ABRT_B signals of the μPD98401A by one clock (early mode) by using an internal register (GMR register). Pull up this pin when it is not used.
ERR_B	67	I	TTL	Error. This pin is used by a device that manages the bus to forcibly stop the operation of the μPD98401A when occurrence of an error is detected on the system bus. When a low level is input to this pin, the μPD98401A stops all bus operations, sets the system bus error bit (bit 25) of the GSR register (when not masked), and generates an interrupt. Pull up this pin when it is not used.

(3/3)

Pin Name	Pin No.	I/O	I/O Level	Function
SR/W_B	68	I	TTL	Slave Read/Write. The SR/W_B signal determines the direction in which the slave is accessed. 1: Read access 2: Write access
SEL_B	69	I	TTL	Slave Select. This signal goes low (active) when the μPD98401A is accessed as a slave. The SEL_B signal must goes low as soon as or after the ASEL_B signal has gone low. An inactive period of at least 2 system clock cycles must be inserted between when the SEL_B signal has become inactive and when it becomes active again.
ASEL_B	70	I	TTL	Slave Address Select. The ASEL_B signal is used to select the direct address register of the μPD98401A. When a low level is input to ASEL_B, the μPD98401A samples the AD bus at the first rising edge of CLK.
CLK	32	I	TTL	Clock. This pin inputs the system clock. Input a clock in a range of 8 to 33 MHz.
RST_B	29	I	TTL	Reset. The RST_B signal initializes the μPD98401A (on starting, etc.). After reset, the μPD98401A can start normal operation. When a low level is input to RST_B, the internal state machine and registers of the μPD98401A are reset, and all 3-state signals go into a high-impedance state. The reset input is asynchronous. When this signal is input during operation, the operating status at that time is lost. Hold RST_B low at least for the duration of one clock. After reset, do not access the μPD98401A for at least 20 clock cycles.
INTR_B	71	O	Nch open-drain output	Interrupt. This is an open-drain signal and must be pulled up. INTR_B informs the CPU that the interrupt bit (unmasked) of the GSR register is set.

2.2.3 Bus monitor pins

The bus monitor pins indicate the type of data under DMA transfer. These five pins are enabled when the BME bit of the GMR register is set to 1; they go into a high-impedance state when the BME bit is 0.

Pin Name	Pin No.	I/O	I/O Level	Function
DBMD	192	O 3-state	CMOS	DMA Bus Monitor Data. This pin indicates that the payload of an AAL-5 cell is under DMA transfer. This pin is enabled when the BME bit of the GMR register is set to 1, and goes into a high-impedance state when the BME bit is 0. The DBMD signal changes in synchronization with the falling of the ATTN_B signal. The high level of this signal indicates that the payload of an ALL-5 packet transmit/receive cell is under DMA transfer, and low level indicates that the other data is being transferred.
DBML	193	O 3-state	CMOS	DMA Bus Monitor Last. If one-word data currently under DMA transfer satisfies any of the following conditions, this pin goes high in synchronization with output of the data. • Last 1 word of last cell of AAL-5 packet • 1-word data to be written to last word of receive buffer • Last 1-word data of last cell of receive packet in which MAX. NUMBER OF SEGMENTS error has occurred When this pin is low, it indicates that the data is other than above. This pin is enabled when the BME bit of the GMR register is set to 1; it goes into a high-impedance state when the bit is 0.
DBMF	194	O 3-state	CMOS	DMA Bus Monitor First. This pin indicates that the data under DMA transfer is the start cell of a receive AAL-5 packet. This pin is enabled when the BME bit of the GMR register is set to 1; it goes into a high-impedance state when the bit is 0. This pin goes high in synchronization with the last word data of the first cell of an AAL-5 packet.
DBMR	206	O 3-state	CMOS	DMA Bus Monitor Remaining. This pin indicates that the number of cells remaining in the transmit buffer is equal to, or has dropped below the value assigned to the RCS register. This pin is enabled when the BME bit of the GMR register is set to 1; it goes into a high-impedance state when the bit is 0.
DBVC	207	O 3-state	CMOS	DMA Bus Monitor VC. Reports that the data currently being transferred by means of DMA is of the VC for which the VCP bit of the receive VC table is set to 1. This pin is asserted active in synchronization with the falling of ATTN_B. It is enabled when the BME bit of the GMR register is set to 1; it goes into a high-impedance state when the BME bit is reset to 0.

2.2.4 Control memory interface pins

These pins constitute an interface through which the μPD98401A accesses an external control memory and a PHY device. A 18-bit address bus and a 32-bit data bus are used. The control memory of the host is accessed only via this interface.

Pin Name	Pin No.	I/O	I/O Level	Function
CD31-CD28 CD27-CD21 CD20-CD16 CD15-CD7 CD6-CD0	110-113 116-122 125-129 132-140 143-149	I/O 3-state	TTL in, CMOS out	Control Memory Data. CD31 through CD0 are 3-state I/O pins and constitute a 32-bit data bus which is used to transfer data with the control memory or a PHY device.
CPAR3- CPAR0	151-154	I/O	TTL in, CMOS out	Control Memory Parity. CPAR3 through CPAR0 indicate the parity of CD31 through CD0 in 8-bit units. In the read cycle, the μPD98401A checks the parity (when enabled). In the write cycle, CPAR3 through CPAR0 output the parity. Pull up these pins when they are not used.
CA17-C11 CA10-CA4 CA3-CA0	158-164 167-173 176-179	O	CMOS	Control Memory Address. CA17 through CA0 constitute an 18-bit address bus. They output an address to the control memory or a PHY device during read/write operation.
CWE_B	186	O	CMOS	Control Memory Write Enable. CWE_B signal indicates the direction in which the control memory is accessed. 1: Read access 2: Write access
COE_B	187	O	CMOS	Control Memory Output Enable COE_B enables or disables data output of the control memory.
CBE3_B CBE2_B CBE1_B CBE0_B	180 181 184 185	O	CMOS	Local Port Byte Enable. CBE3_B through CBE0_B indicate the byte on the control port to be read or written.
INITD	188	I	TTL	Initialization Disable. The INITD signal is used to disable automatic initialization of the control memory during chip test. During normal operation other than test, directly connect INITD to GND.

2.2.5 JTAG boundary scan pins

Pin Name	Pin No.	I/O	I/O Level	Function
JDI	201	I	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	200	O 3-state	CMOS	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	197	I	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	202	I	TTL	JTAG Test Mode Select. Normally, fix this pin to a high or low level.
JRST_B	203	I	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 RESET_B of the μPD98401A 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.

2.2.6 Test pin

Pin Name	Pin No.	I/O	I/O Level	Function
TRF_B	189	I	TTL	This pin is used to test the internal circuitry of the chip. 0: Normal operation 1: Test Normally, directly connect this pin to ground and fix it to a low level.

2.2.7 Power supply and ground pins

Pin Name	Pin No.	I/O	Function
V_{DD}	15, 27, 30, 34, 41, 53, 58, 72, 78, 94, 104, 114, 124, 130, 142, 157, 165, 174, 183, 190, 195, 198, 204, 208	—	Power supply (24 pins) These 24 V_{DD} pins supply a voltage of +5 V ± 5% to the chip.
GND	1, 2, 8, 14, 21, 26, 31, 33, 40, 51, 52, 59, 73, 79, 91, 93, 105 ,106, 115, 123, 131, 141, 150, 155, 156, 166, 175, 182, 191, 196, 199, 205	—	Ground (32 pins) Connect these pins to ground.

2.2.8 Pin status during and after reset

Pin	During Reset	After Reset
AD0-AD31	Hi-Z (input mode)	Hi-Z (input mode)
PAR0-PAR3	Hi-Z (input mode)	Hi-Z (input mode)
SIZE0-SIZE2	0	0
DR/W_B	1	1
ATTN_B	1	1
INTR_B	1 (however, pulled up)	1 (however, pulled up)
CA17-CA0	0	0
CD0-CD31	All 0 (output mode)	All 0 (output mode)
CWE_B	1	1
COE_B	1	1 (repetition of high/low)
CBE3_B-CBE0_B	All 1	All 1
PHRW_B	0	0
PHOE_B	1	1
PHCE_B	1	1
RCLK	CLK output	CLK output
RENBL_B	0	0
Tx0-Tx7	All 0	All 0
TCLK	CLK output	CLK output
TENBL_B	1	1
TSOC	0	0
JDO	Hi-Z (3-state)	Hi-Z (3-state)
DBMD	Hi-Z	Hi-Z
DBML	Hi-Z	Hi-Z
DBMF	Hi-Z	Hi-Z
DBMR	Hi-Z	Hi-Z

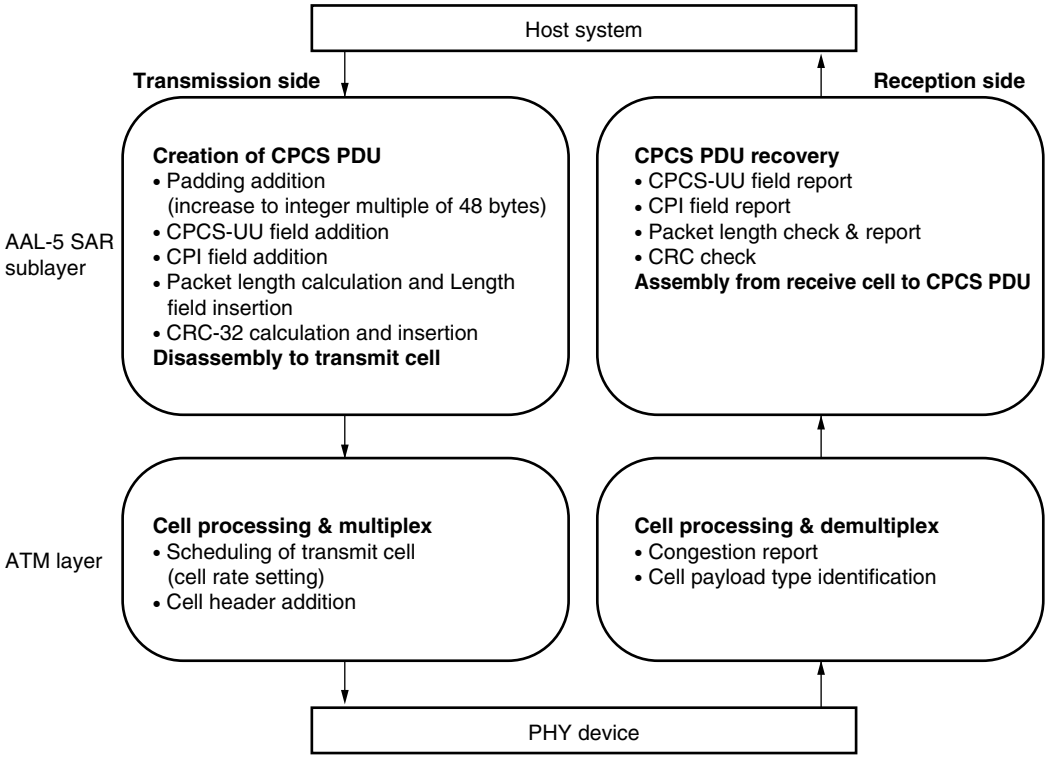
CHAPTER 3 FUNCTIONAL OUTLINE

This chapter outlines the functions of the μPD98401A. For details on each function, refer to **CHAPTER 5 OPERATIONS OF μPD98401A**.

3.1 Functional Outline of μPD98401A

The μPD98401A 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 μPD98401A



The μPD98401A is placed between the host system and a PHY device and is controlled by the host by accessing the internal registers of the μPD98401A via the 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 written to the system memory by means of DMA 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 through the μPD98401A

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

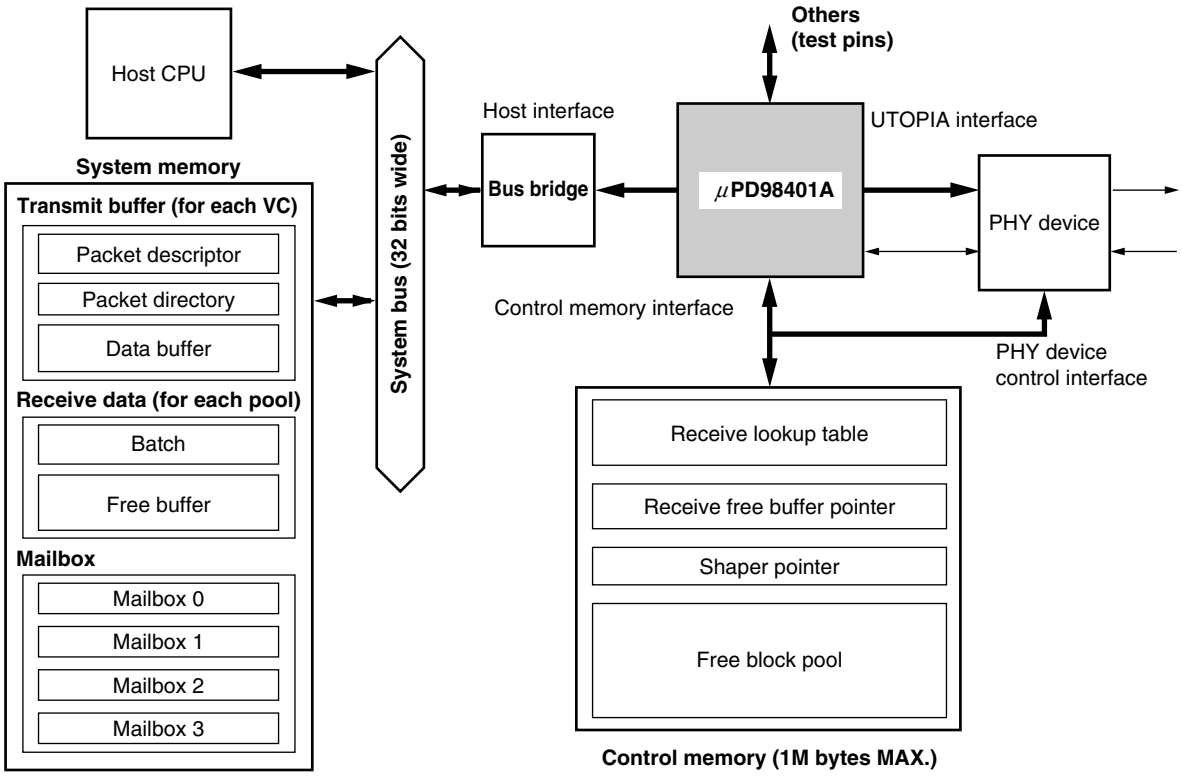
The μPD98401A uses a dedicated external memory as a control memory to execute transmission/reception processing. Up to 1M bytes of control memory can be used. The control memory is divided into the following four areas. The size of each area is mainly determined by the number of channels simultaneously transmitted/received by the μPD98401A, and the boundaries of the four areas are set by the host in a register of the μPD98401A.

<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 μPD98401A and a PHY device transfer data with each other in cell format. The μPD98401A makes the transmit data read by means of DMA in segment units (48 bytes as the payload section of the cell) into cells by internal processing, and transfers these cells to the PHY device via the UTOPIA interface. The cells the μPD98401A receives from the PHY device are transferred by means of DMA to the system memory under management of the host in cell units after the reception processing by the μPD98401A.

Figure 3-2. System Outline of μPD98401A

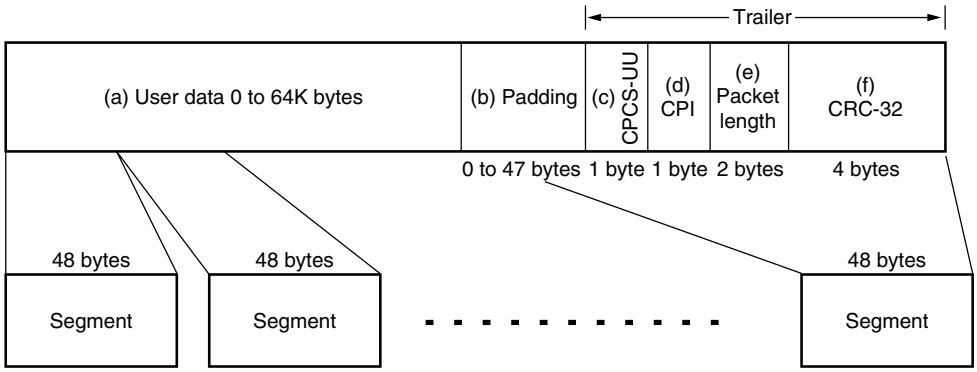


3.2 AAL-5 SAR Sublayer Function

When transmitting data, the μPD98401A 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 μPD98401A deletes the header 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 μPD98401A automatically inserts data of all zeros.

(c) CPCS user-user information (CPCS-UU) field:
Used to transfer CPCS user-user information. The μPD98401A 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 μPD98401A 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 μPD98401A 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:

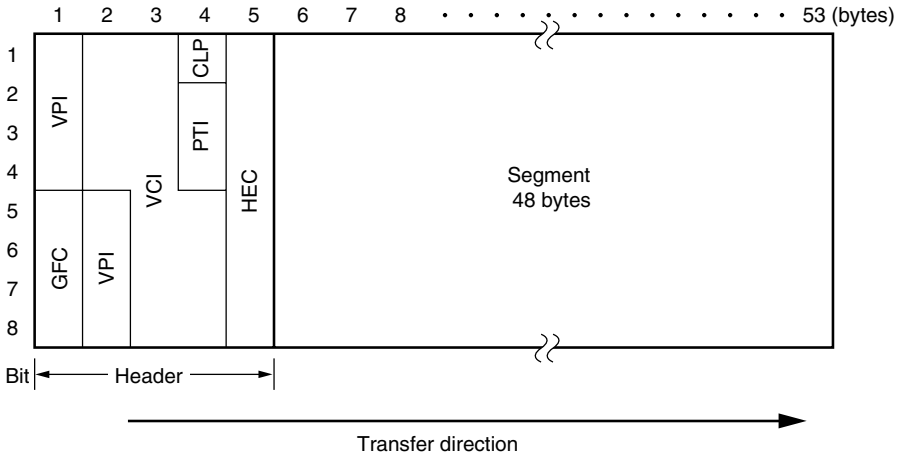
$$\text{Expression} = 1 + x + x^2 + x^4 + x^5 + x^7 + x^8 + x^{10} + x^{11} + x^{12} + x^{16} + x^{22} + x^{23} + x^{26} + x^{32}$$

3.3 ATM Layer Functions

3.3.1 Creation of cells

The μPD98401A 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 μPD98401A are explained next.

(1) GFC (generic flow control) field

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

<Function of μPD98401A>

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 (not processed) 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 μPD98401A>

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

Reception : Supports up to 16 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 16 bits, refer to **5.5.4 Setting of receive look-up table**.

In the μPD98401A, the settings for one channel are divided into transmit VC and receive VC.

The μPD98401A can support up to 32K 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 μPD98401A, 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 operation information

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

<Function of μPD98401A>

Transmission : The μPD98401A stores the pattern set by the user in this field as is and transmits it. When a processing of AAL-5 type is selected, the μPD98401A 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 μPD98401A 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 μPD98401A checks the least significant bit of the PTI field to determine the last segment of the packet.

PTI Field	Processing of μPD98401A
000	Can receive raw cell if reception of AAL-5 packet as user data is specified
001	
010	
011	
100	Receives 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 μPD98401A>

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 μPD98401A 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 μPD98401A>

Transmission : The μPD98401A inserts dummy data "00H" to this field and transmits it.

Reception : The μPD98401A ignores this field.

3.3.2 Setting of cell transmission rate

The μPD98401A has 16 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 set a shaper for each channel.

Of the 16 shapers, one or more shapers 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 bandwidth at which all the channels transmit data can be limited.

3.3.3 Support of non-AAL-5 traffic

The μPD98401A has a function to process transmit/receive 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 cell with header to the system memory along with 11-byte indication information (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 μPD98401A 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.

Each time a raw cell has been received, it is reported by an interrupt. When the raw cell of the packet of AAL-3/4 cell is received, the ST field of the AAL-3/4 cell is checked, and an interrupt is generated only if the last cell or a packet with only one cell has been received, in order to mitigate the workload of the host.

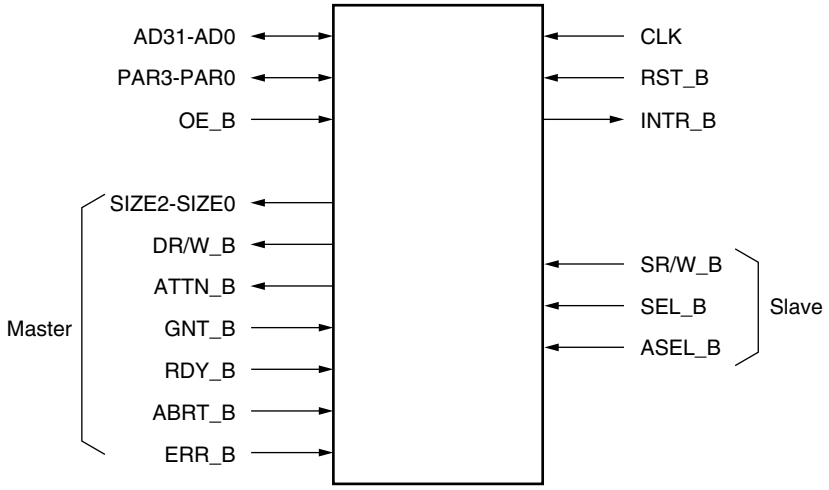
CHAPTER 4 INTERFACES

The μPD98401A has a host bus interface, PHY device interface, and control memory interface. This chapter explains in detail the function and operation of each interface.

4.1 Bus Interface

The bus interface of the μPD98401A is a 32-bit address/data multiplexed bus and has different control signals for master (DMA) operation and slave operation. This interface is a general-purpose bus interface that can be connected to a general I/O bus (PCI, S bus, GIO, or AP bus) with a minimal number of additional circuits.

Figure 4-1. Bus Interface Signals



4.1.1 Bus operation control pins

(1) OE_B pin

The OE_B pin inputs an output enable signal for AD31 through AD0 and PAR3 through PAR0 of the μPD98401A. The μPD98401A executes output from AD31 through AD0 and PAR3 through PAR0 only when a low level is input to the OE_B pin. If a high level is input to this pin, the μPD98401A makes AD31 through AD0 and PAR3 through PAR0 go into a high-impedance state.

This operation is performed independently of the internal state of the μPD98401A. For example, if the OE_B signal is made high during DMA write operation by the μPD98401A as the master, the μPD98401A immediately sets output to the bus in a high-impedance state. Internally, however, the μPD98401A continues the DMA cycle, and data being transferred will be lost.

This signal is used if it is necessary to forcibly stop the bus operation of the μPD98401A from an external source. Fix this signal to a low level (enable) unless there is a case in which it must be set in a high-impedance state.

(2) ERR_B pin

The ERR_B pin inputs a signal that allows the device managing the bus to stop the bus operation of the μPD98401A if an error is detected on the system bus. A low level can be input to this pin only when the μPD98401A is in the data phase of DMA transfer (master operation).

When a low level is input to this pin, the μPD98401A stops DMA transfer under execution, and immediately returns the ATTN_B signal to an inactive level. If the "SBE bit" (system bus error) of the GSR register is set to 1, and if interrupts are not masked, an interrupt is issued to the host. In a system bus error status, the μPD98401A no longer performs DMA transfer (master operation), and the transmission/reception operation is stopped. Because the slave can be accessed, execute software reset or hardware reset to restore the original status.

Caution The ERR_B signal can be made active only in the data phase during DMA transfer. If this signal is made low in other timing, the μPD98401A malfunctions.

Figure 4-2. Input Timing of ERR_B Signal (4-word burst)

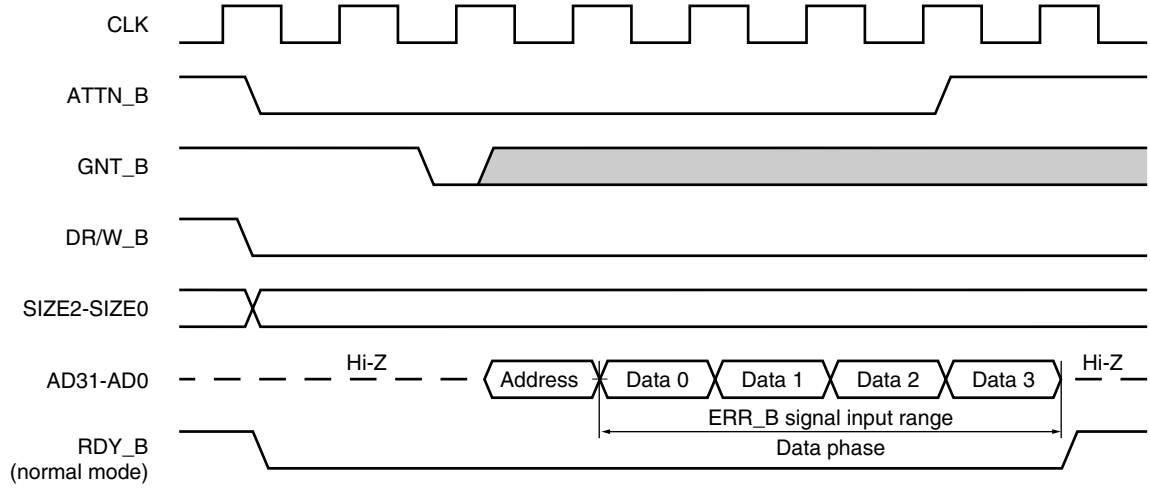
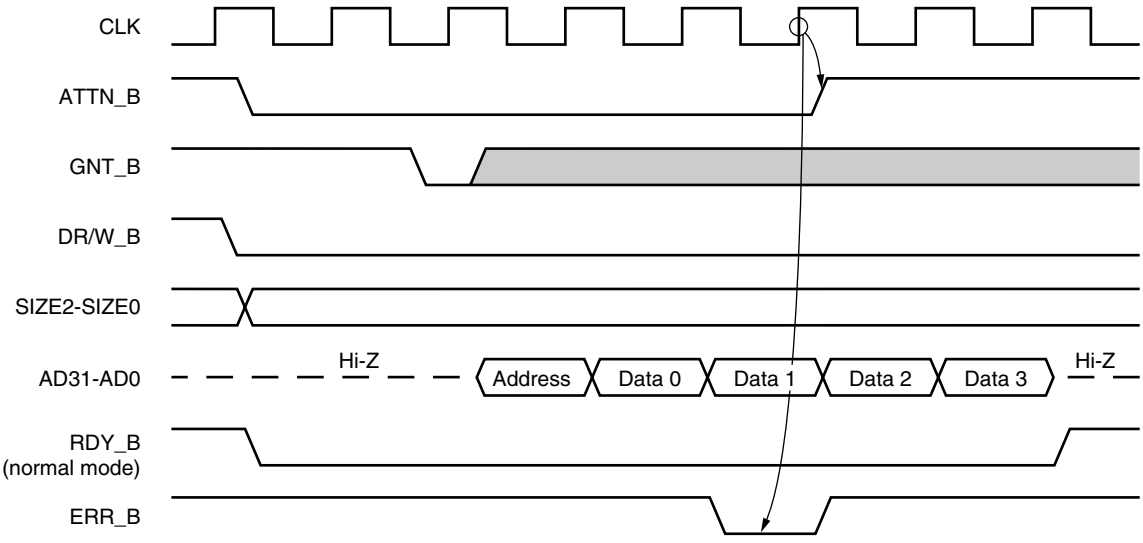


Figure 4-3. Input Example of ERR_B Signal (4-word burst)



4.1.2 Parity check function

The μPD98401A uses pins PAR3 through PAR0 to input/output parity signals to/from the bus interface, and a function to output/check parity bits of AD31 through AD0.

The user can enable or disable the parity function, and select byte or word parity, and odd or even parity, by setting the GMR register.

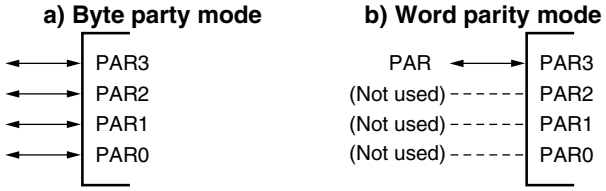
Table 4-1. Bus Parity Mode Select Bit

Selects enabling/disabling bus parity		
BPE bit (GMR: bit 3)	0	Disabled. μPD98401A does not check bus parity.
	1	Enabled. μPD98401A checks bus parity bit input. When it detects error, it sets SPE bit of GSR register and generates interrupt (if not masked).
	Default = 0	
Selects even/odd parity mode		
PC bit (GMR: bit 4)	0	Even parity mode
	1	Odd parity mode
	Default = 0	
Selects byte/word parity mode		
PM bit (GMR: bit 5)	0	Byte parity mode. All bus parity pins PAR3 through PAR0 are used.
	1	Word parity mode Only PAR3 of bus parity pins is used.
	Default = 0	

The parity bit is input/output from PAR3 through PAR0 pins. The pins used differ depending on whether the byte parity mode or word parity mode is used.

In the byte parity mode, all the PAR3 through PAR0 pins are used. PAR3 inputs/outputs the parity bit of AD31 through AD24, and PAR0 inputs/outputs the parity bit of AD7 through AD7. In the word parity mode, the PAR3 pin inputs/outputs the parity bit of AD31 through AD0. These pins function as output pins and output the parity bit when the μPD98401A outputs an address or writes data. When the μPD98401A reads data, the pins serve as input pins and receive the parity bits input from an external source and internally check the parity bits. When the μPD98401A does not access a bus, these pins go into a high-impedance state.

Figure 4-4. Difference in Pins Used Depending on Byte/Word Parity Mode



The μPD98401A generates output parity and checks input parity both during master (DMA) operation and slave operation. When the bus is in the input direction from the host to the μPD98401A, the μPD98401A checks the parity bits that are input along with address and data. When the BPE bit of the GMR register is set to 1 and therefore, the

check function is enabled, the SPE bit of the GSR register is set to 1 and an interrupt is generated (if not masked) if a parity error is detected.

Caution If the μPD98401A is executing a master (DMA) or slave operation when a parity error has been detected, the μPD98401A continues and completes the operation. If the slave operation has been performed, a new slave request can be accepted and executed. However, the operation of the μPD98401A when and after the parity error has been detected cannot be guaranteed. If a parity error has been detected, reset the μPD98401A.

The PAR3 through PAR0 pins always operate as three-state input/output pins regardless of the setting of the BPE bit of the GMR register (enabling or disabling the bus parity). Therefore, pull up these pins when they are not used.

Caution Even if the parity check function is disabled (BPE bit = 0), the parity bit output function of the μPD98401A is operating, and the parity bits are always output from PAR3 through PAR0 when the μPD98401A outputs an address or data to the bus. Therefore, if the host reads a register of the μPD98401A or if the μPD98401A executes DMA transfer before the host sets a parity mode to the GMR register after power application, the host detects a parity error.

4.1.3 Master (DMA) operation

The μPD98401A reads the transmit data and each descriptor in the external system memory, or writes the indication of the transmission/reception completion status or receive data to the external memory by using an internal 32-bit DMA controller, when it operates as the bus master.

(1) Master (DMA) operation

The following input/output pins are used to control the master operation of the μPD98401A.

- ATTN_B (output) : This is a DMA request signal the μPD98401A outputs to the host. If the μPD98401A has an internal source of DMA transfer, it makes this pin active to notify an external bus arbiter. This pin becomes inactive at the rising edge of the clock when only one word remains to be transferred.
- GNT_B (Input) : This signal is input by the bus arbiter to grant the μPD98401A the right to use the bus in response to a DMA request from the μPD98401A.
- DR/W_B (output) : Indicates the direction of DMA access.
- SIZE (output) : Indicates the size of the data under DMA transfer.
- RDY_B (input) : Wait cycle control signal.
- ABRT_B (input) : This signal is input when the host cancels the DMA cycle of the μPD98401A under execution.

Figure 4-5. 1-Word Write DMA Cycle Timing

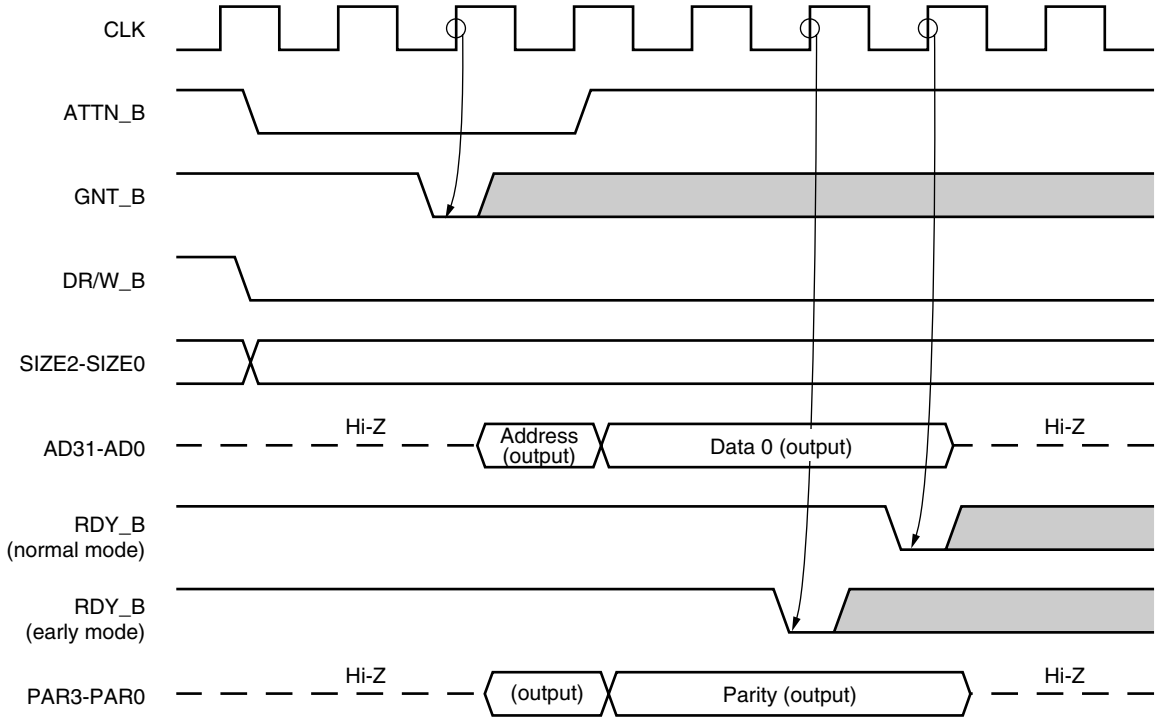
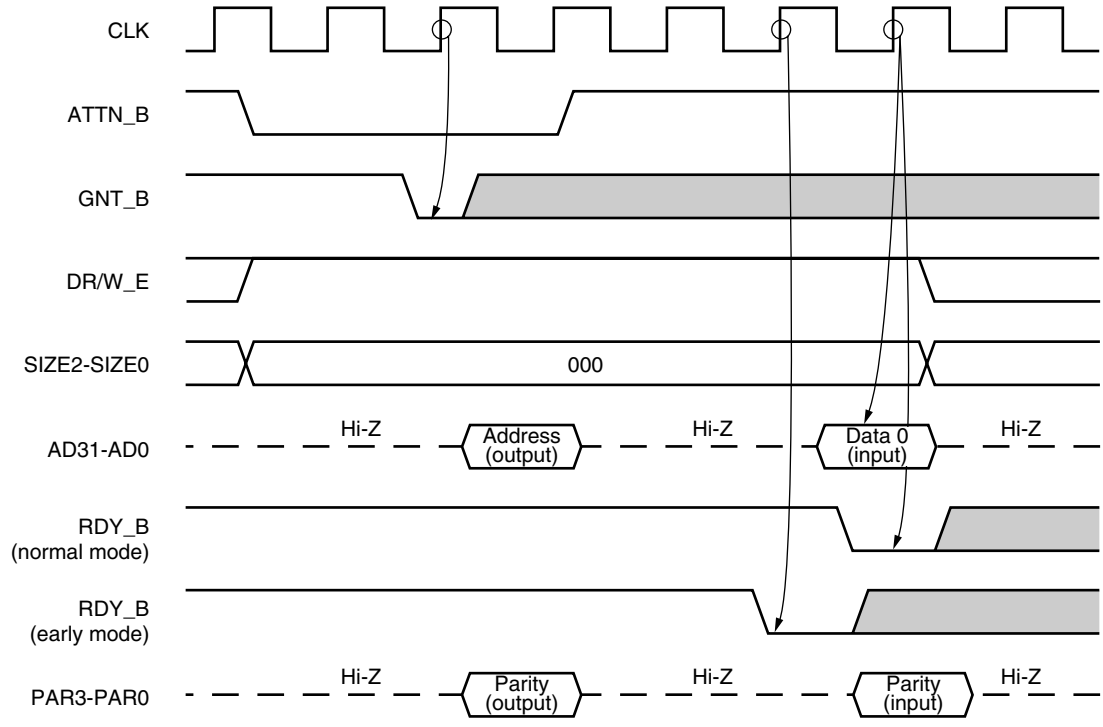


Figure 4-6. 1-Word Read DMA Cycle Timing



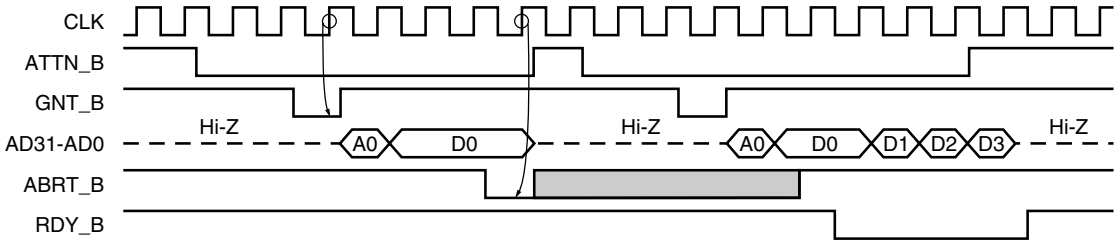
The μPD98401A requests an external bus arbiter for the right to use the bus by making the ATTN_B signal low. At the same time, it indicates the direction of transfer by using DR/W_B, and the bus size by using SIZE2 through SIZE0. The ATTN_B signal becomes active once for 1 burst transfer. The bus arbiter acknowledges the request made by the ATTN_B signal by making the GNT_B signal low, and grants the μPD98401A the right to use the bus. After making the ATTN_B signal low, the μPD98401A samples the GNT_B signal at the rising edge of the clock. When the μPD98401A detects that GNT_B has gone low, it outputs an address at the rising edge of the clock, and starts sampling the RDY_B signal. The address output by the μPD98401A is one clock cycle in both the read and write cycles. After outputting the address, the uμPD98401A starts sampling the RDY_B signal at the rising edge of the clock.

During DMA read operation, the μPD98401A latches data on AD31 through AD0 at the rising edge of the clock when a low level has been input to the RDY_B pin. During DMA write operation, the μPD98401A outputs data to AD31 through AD0 immediately after the address cycle. The data is retained until a low level is input to the RDY_B signal. The user can insert a wait cycle by controlling input of this RDY_B signal. The μPD98401A makes the ATTN_B signal inactive at the rising edge of the clock when there remains only one piece of data to be transferred. The μPD98401A also makes the GNT_B signal inactive, until the ATTN_B signal is made high next time to make a DMA cycle request. The GNT_B signal can be made inactive at any time, before the ATTN_B signal is made high.

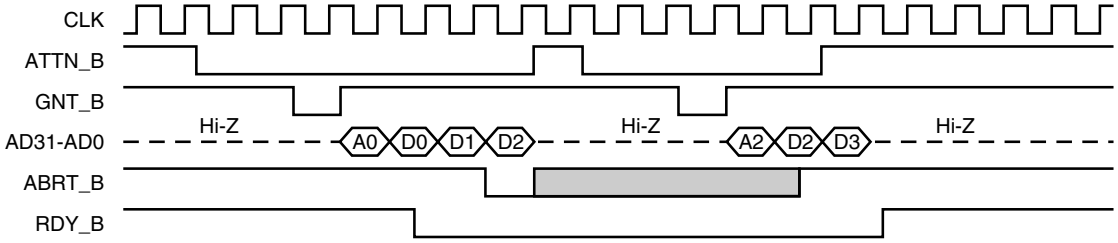
The ABRT_B signal is used to abort a data transfer cycle. The μPD98401A also samples the ABRT_B signal, as well as the RDY_B signal, in the data transfer cycle after address output. If a low level is input to the ABRT_B pin in a data transfer cycle, the μPD98401A aborts DMA transfer in that cycle, and makes the ATTN_B pin inactive once. After that, it makes the ATTN_B signal active again, and retries transfer from the aborted data. If a low level is input simultaneously to the ABRT_B and RDY_B pin, the μPD98401A gives priority to the ABRT_B signal. Figure 4-7 shows an example of input timing of the ABRT_B signal.

Figure 4-7. ABRT_B Signal Input Timing

(a) Aborted at first word during 4-word burst



(b) Aborted at third word during 4-word burst



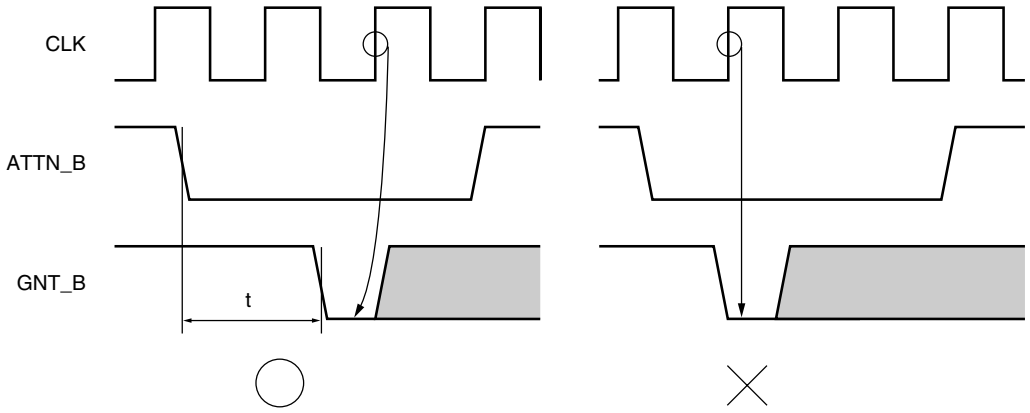
The user can set either “normal mode” or “early mode” as the μPD98401A’s sampling timing of the RDY_B and ABRT_B signals independently during DMA read and DMA write operations. This is done by using the RA and WA bits of the GMR register. In the early mode, the μPD98401A detects RDY_B and ABRT_B at the rising edge of a system clock one cycle earlier than in the normal mode.

Setting of input timing of RDY_B/ABRT_B signals during read operation		
RA bit (GMR: bit 12)	0	Normal mode
	1	Early mode. Input one clock earlier
	Default = 0	Normal mode
Setting of input timing of RDY_B/ABRT_B signals during write operation		
WA bit (GMR: bit 13)	0	Normal mode
	1	Early mode. Input one clock earlier
	Default = 0	Normal mode

Remark For the differences in input timing between the normal mode and early mode, refer to **Figure 4-5. 1-Word Write DMA Cycle Timing.**

Caution Make sure that the interval between the falling edge of the ATTN_B signal and the falling edge of the GNT_B signal (t in Figure 4-8) is at least one system clock cycle. The μPD98401A malfunctions if it detects the low level of the GNT_B signal at the rising edge of the clock immediately after the one at which the ATTN_B signal has been made low.

Figure 4-8. Relations between ATTN_B and GNT_B



(2) Burst transfer

The μPD98401A supports burst transfer of 1, 2, 4, 8, 12, and 16 words. The user can select the mode of burst transfer by setting the burst size to be enabled in the "SZ field" or "TBE field" of the GMR register.

Table 4-2. Selecting Burst Size to Be Enabled

Selects burst size to be executed (GMR register, bits 11 through 8: SZ field, bit 16: TBE field)		
SZ field	Bit 11	1: Enables 16-word burst, 0: Disabled
	Bit 10	1: Enables 8-word burst, 0: Disabled
	Bit 9	1: Enables 4-word burst, 0: Disabled
	Bit 8	1: Enables 2-word burst, 0: Disabled
TBE bit	Bit 16	1: Enables 12-word burst, 0: Disabled (The AD bit must also be set to 1.)
Default = All 0 (supports only 1-word transfer)		

Two or more burst size can be simultaneously enabled. The 1-word transfer mode is always enabled, regardless of the setting of the "SZ field" and "TBE field". To enable 12-word burst by setting the TBE bit to 1, the AD bit must be also set to 1 and the burst size select function must be disabled. When 16-word transfer is enabled, the μPD98401A executes 16-word burst only when it writes a raw cell to the system memory.

Table 4-3 shows the data that are transferred by the master (DMA) operation of the μPD98401A. For the meaning of each operation, refer to **CHAPTER 5 OPERATIONS OF μPD98401A**. Some of the data to be transferred must always start from a word (32-bit boundary) and the others can start from a byte boundary.

Table 4-3. DMA Transfer by μPD98401A

Read/Write	Type of Data	Number of Words	Byte Boundary
Read	Packet descriptor	4 words	Impossible
	Buffer descriptor	2 words	Impossible
	Transmit cell data	1 to 12 words	Possible
	Buffer address in receive batch	1 word	Impossible
Write	Transmit indication	1 word	Impossible
	Receive indication	4 words	Impossible
	Receive cell data	12 words	Possible
	Receive batch ring pointer	1 word	Impossible
	Raw cell data	16 words	Impossible

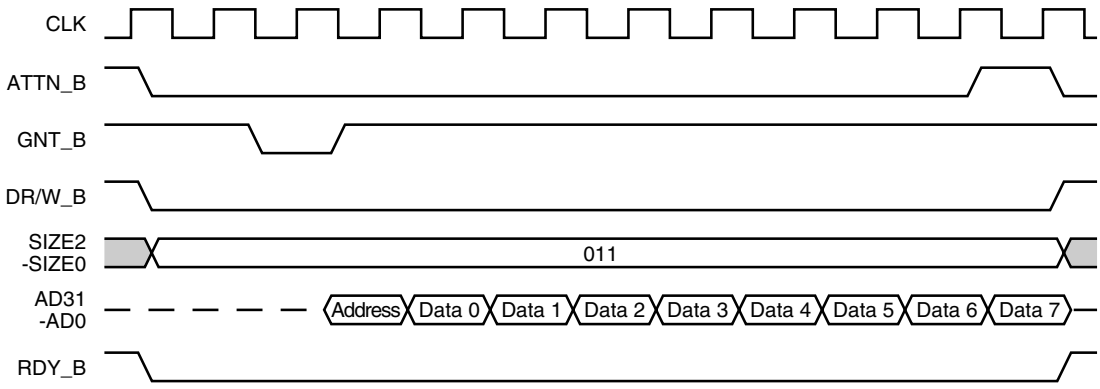
The burst size currently under execution is coded as shown in Table 4-4 and output from the SIZE2 through SIZE0 pins.

Table 4-4. Burst Size under Execution

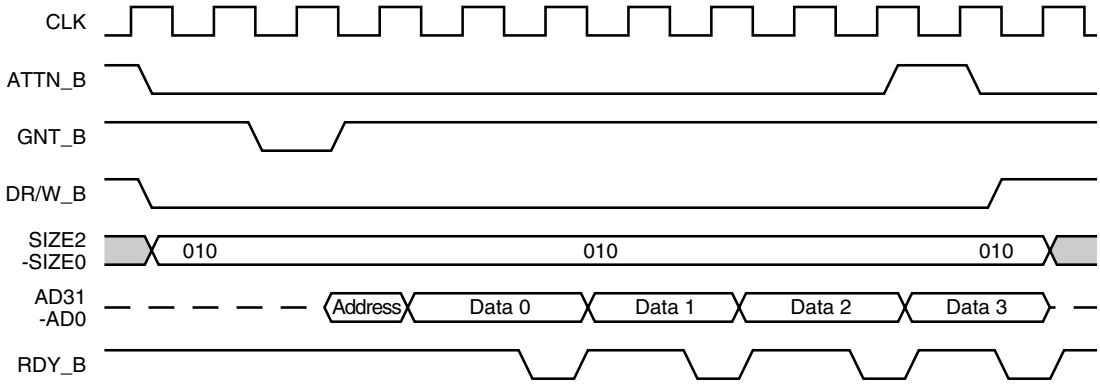
SIZE2-SIZE0	Burst Size under Execution
000	1-word transfer
001	2-word burst
010	4-word burst
011	8-word burst
100	16-word burst
101	12-word burst
110	Undefined
111	Receive byte alignment transfer

Figure 4-9. Burst Write Cycle Timing

(a) Example of 8-word burst write cycle (no wait)



(b) 4-word byte write cycle (with 1 wait cycle inserted)



(3) Burst size select function

If two or more burst sizes are enabled, the μPD98401A checks the address field at the transfer destination, and automatically selects the burst size to be executed from the enabled burst sizes. This function is valid in both the DMA read and write cycles executed by the μPD98401A. However, it cannot be used to enable 12-word burst.

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 is given the top priority when raw cell data is transferred.

Burst size	16	8	4	2	1
Priority	High → low				

The μPD98401A checks the number of '0s' in the low-order byte of the address fields AD2 through AD5 at the transfer destination when it executes DMA transfer. The executable burst size is determined according to the number of '0s'. Priorities are assigned to burst size sequentially, in the order 8-, 4-, 2-, and 1-word. The μPD98401A selects and executes the enabled burst size with the highest priority.
This function can be enabled or disabled by using the "AD bit" of the GMR register.

Selection of burst transfer mode		
AD bit (GMR: bit 7)	0	The μPD98401A checks the address field at the transfer destination when it executes DMA transfer, and the function to select the burst size actually to be executed from the enabled burst sizes is enabled.
	1	The function to check the address field at the transfer destination when the μPD98401A executes DMA transfer is disabled. The μPD98401A selects and executes the biggest burst size of the enabled burst sizes.
	Default = 0	

Caution If TBE is set to 1 and the 12-word burst is enabled, be sure to set the AD bit to 1 to disable the above function.

Next, an example of operation when this function is enabled (AD = 0) or disabled (AD = 1).
During a reception operation, the μPD98401A stores each 1 cell to the system memory by executing a master (DMA) operation. Because the payload of an ATM cell is 48 bytes, the cell data to be transferred by the μPD98401A is 12 words. Assuming the start address of the receive buffer is "00000000H", the operation to store the receive cell data in each case is shown below.

Table 4-5. Burst Transfer Transition in Each Case

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

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

(4) Byte alignment transfer

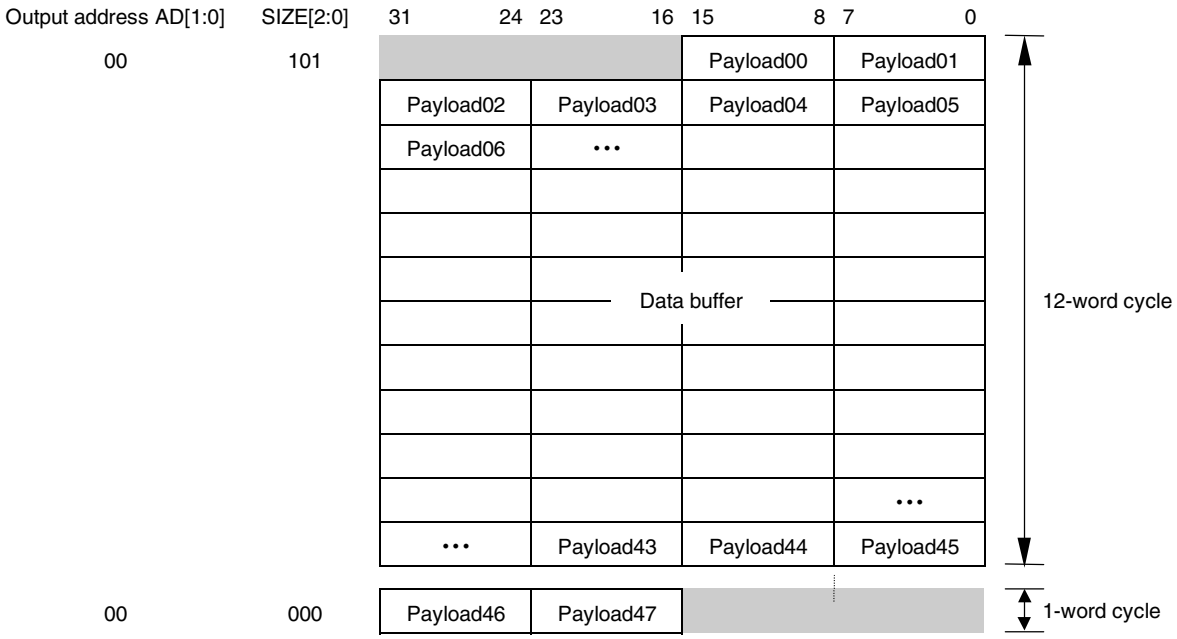
The μPD98401A can place transmit/receive cell data, excluding raw cell data, from a byte boundary of the system memory. The μPD98401A checks the low-order 2 bits of the start address, AD1 and AD0, when it executes a DMA cycle. If these bits are other than 00, the μPD98401A executes byte alignment transfer.

(a) Byte alignment transfer of transmit cell data

The transmit cell data is received from the data buffer on the system memory by means of DMA read. The start address of the data buffer does not have to be always at a 32-bit boundary. The μPD98401A executes DMA read in word (32-bit) units even if the transmit data does not start from a 32-bit boundary, and internally ignores unnecessary bytes.

As an example, Figure 4-10 shows a case where 12-word data of one cell is placed in the data buffer, and its start address starts from a byte boundary with 12-word burst enabled.

Figure 4-10. Transmit Cell Data Byte Alignment (big endian)



First, the μPD98401A executes a DMA read cycle of 12-word burst. It internally drops the high-order 2 bytes of the first word. Next, the μPD98401A executes a DMA read cycle of one word. It drops the low-order 2 bytes. In these two DMA cycles, the μPD98401A outputs a 32-bit boundary address whose low-order 2 bits are 00.

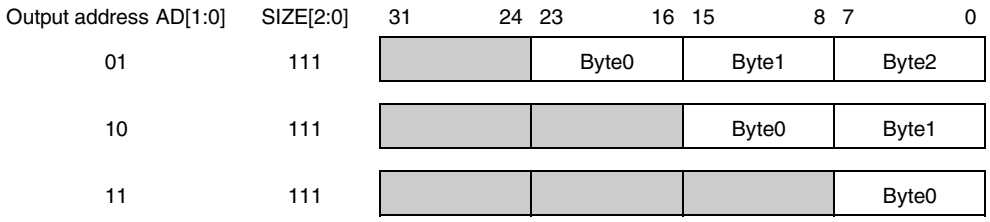
(b) Byte alignment transfer of receive cell data

The μPD98401A stores receive cell data to the free buffer in the system memory by means of DMA write. The start address of the free buffer does not always have to start from a 32-bit boundary, but can start from a byte boundary. If the low-order 2 bits 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, or 11, the μPD98401A executes 1-word DMA write. In this DMA cycle, the AD[1:0] pins output a byte boundary to be written, and the SIZE[2:0] pins output "111", indicating that DMA write to be executed is for a specific byte.

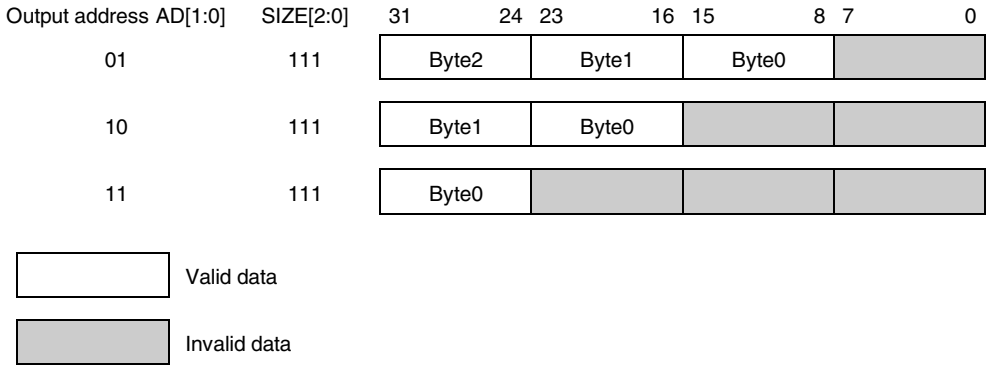
If the start address of the free buffer starts from a byte boundary and if SIZE[2:0] = 00 and AD[1:0] output ≠ 00, the user must decode these signals and append glue logic that controls writing to the system memory in byte units.

Figure 4-11. AD[1:0] Pin and SIZE[2:0] Pin Output during Byte Alignment DMA Write

(a) Big endian



(b) Little endian



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.

Figure 4-12. Example of Storing Cell Data by Byte Alignment

Step-1 Executes 1-word DMA write. Outputs SIZE = 111 and AD[1:0] = 10.

		31 24	23 16	15 8	7 0
First cell	AD[31:2]=00	(don't care)	(don't care)	Payload0	Payload1

Step-2 Executes 12-word burst. Outputs SIZE = 101 and AD[1:0] = 00.

01	Payload2	Payload3	Payload4	Payload5
02	Payload6	Payload7	Payload8	Payload9
03	Payload10	Payload11	Payload12	Payload13

0a	Payload38	Payload39	Payload40	Payload41
0b	Payload42	Payload43	Payload44	Payload45
0c	Payload46	Payload47	(don't care)	(don't care)

Step-3 Same as Step-1

Second cell	AD[31:2]=0c	(don't care)	(don't care)	Payload0	Payload1

Step-4 Same as Step-2

0d	Payload2	Payload3	Payload4	Payload5
0e	Payload6	Payload7	Payload8	Payload9
0f	Payload10	Payload11	Payload12	Payload13

16	Payload38	Payload39	Payload40	Payload41
17	Payload42	Payload43	Payload44	Payload45
18	Payload46	Payload47	(don't care)	(don't care)

Notice that the start address of the second cell is the same as the last word of the first cell. This address is overwritten unless writing of the system memory is controlled in byte units.

(5) Bus monitor signal output function

The μPD98401 has five bus monitor pins to identify the type of data currently under DMA transfer. All these pins are enabled if the BME bit of the GMR register is set to 1, and go into a high-impedance state when the BME bit is 0. The user can control AAL-5 data flowing on the bus by using this pin information in combination.

(a) DBMD pin

This pin indicates that the data under DMA read/write is the data of the payload of an AAL-5 cell. It changes its status at the falling edge of the ATTN_B signal. If the ABRT_B signal is input in the DMA cycle of the payload of transmit or receive cell data, and if the cycle is aborted, the DBMD signal retains its status until the next DMA cycle.

High level : Transfers payload data of AAL-5 cell.
Low level : Transfers other data.

Figure 4-13. DBMD Signal Output Timing Example 1

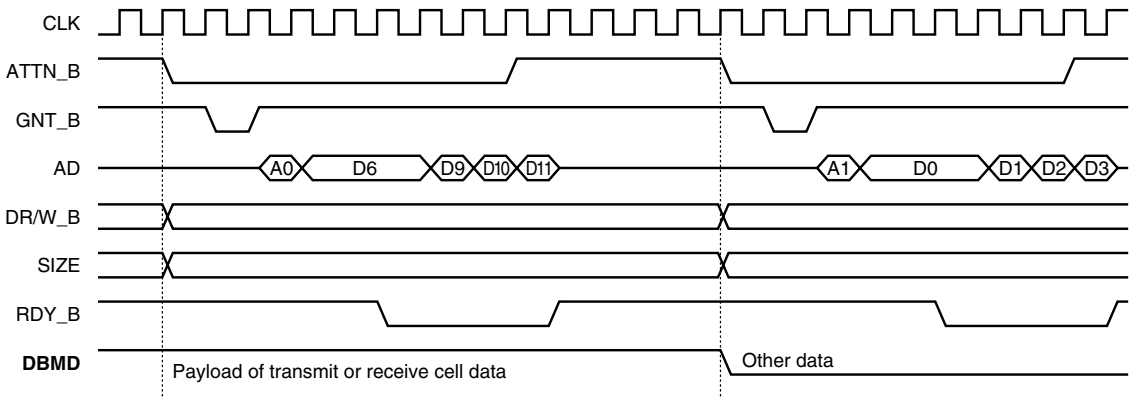
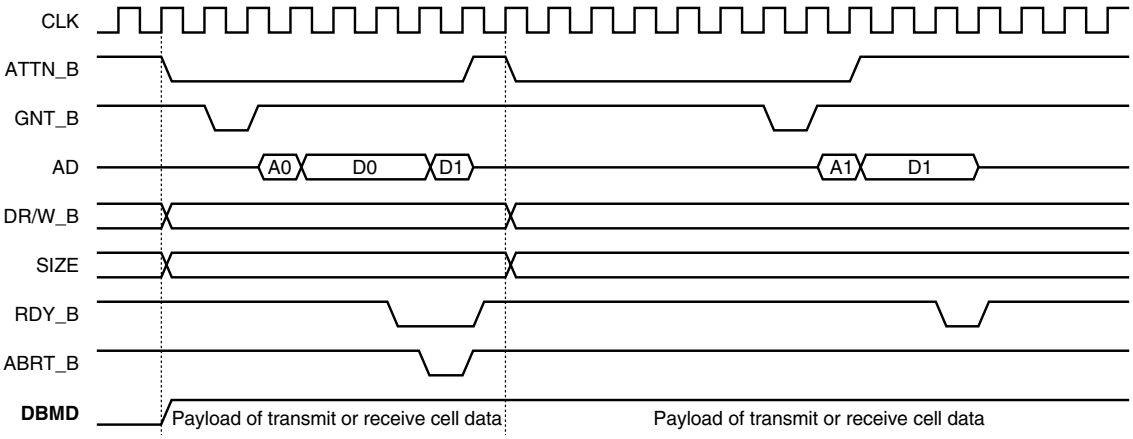


Figure 4-14. DBMD Signal Output Timing Example 2 (if aborted)



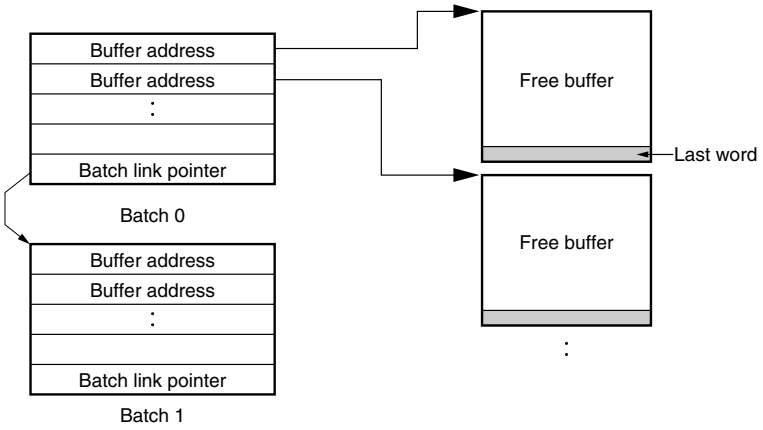
(b) DBML pin

This pin indicates that the data under DMA write is the last cell of a packet or last write to the receive buffer.

High level : Satisfy any of the following three conditions.
Low level : Transfers data that does not satisfy any of the following three conditions.

<1> End of packet
Transfers the last 1 word of the last cell data of a receive AAL-5 packet (cell with PTI field being '001' or '011). Transfers a 32-bit word including the last byte of the receive AAL-5 packet in the case of byte alignment transfer.

<2> End of buffer
When the last one word in the free buffer is written.

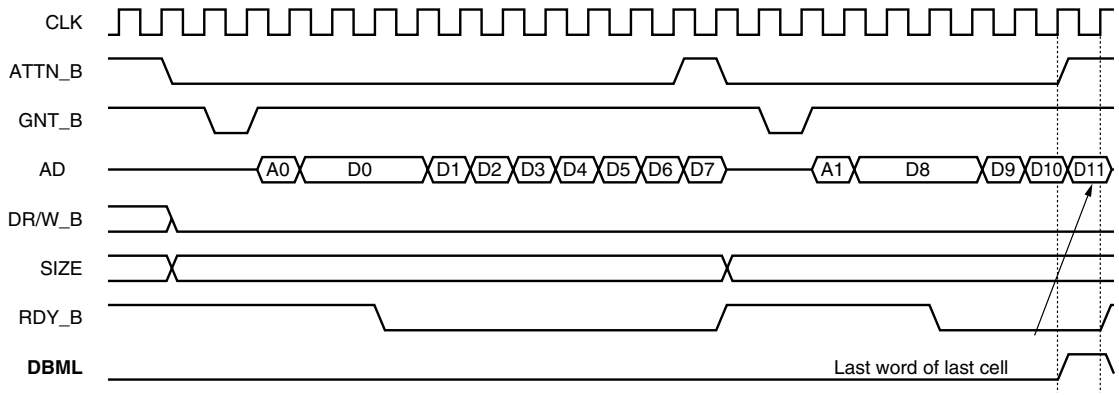


For the free buffer, refer to **(1) Receive pool** in **5.5.2 Structure of receive data**.

<3> End of packet because of MAX. NUMBER OF SEGMENTS violation.
Transfers the last one word of the last cell data of the receive packet that is responsible for the MAX. NUMBER OF SEGMENTS violation.
For MAX. NUMBER OF SEGMENTS violation, refer to **(4) AAL-5 packet reception error detection** in **5.5.5 Reception operation**.
Low level: Transfers other data.

Caution This pin is checked to see if a given condition is satisfied, and is valid only when the data of the AAL-5 packet is to be transferred. It is not valid when Raw cell data is being transferred.

Figure 4-15. DBML Output Timing

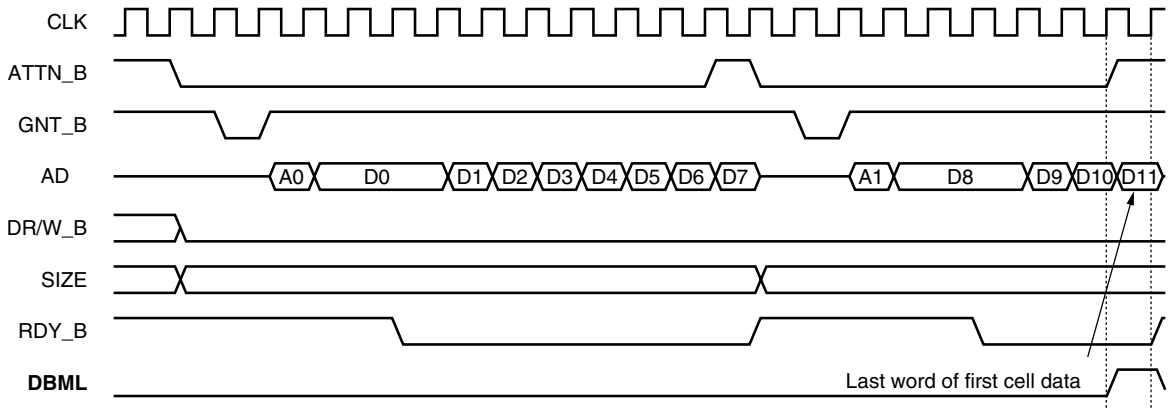


(c) DBMF pin

This pin indicates that the first cell of a receive AAL-5 packet is transferred.

High level : Transfers the last one word of the first cell data of a receive AAL-5 packet.
Low level : Transfers other data

Figure 4-16. DBMF Output Timing



(d) DBMR pin

This pin indicates that the data in the data buffer to be transmitted has run short. The user assigns the number of remaining data units at which to be notified to the RCS register in cell units. The DBMR signal changes its level in synchronization with the falling of ATTN_B when the data to be transmitted reaches a set value.

High level : The number of data remaining in the transmit data buffer falls short of the number of data assigned to the RCS register as a result of the transmit data DMA read this time.
Low level : More data units than the number set in the RCS register are in the data buffer. When the RCS register is set to 0, this pin always outputs a low level.

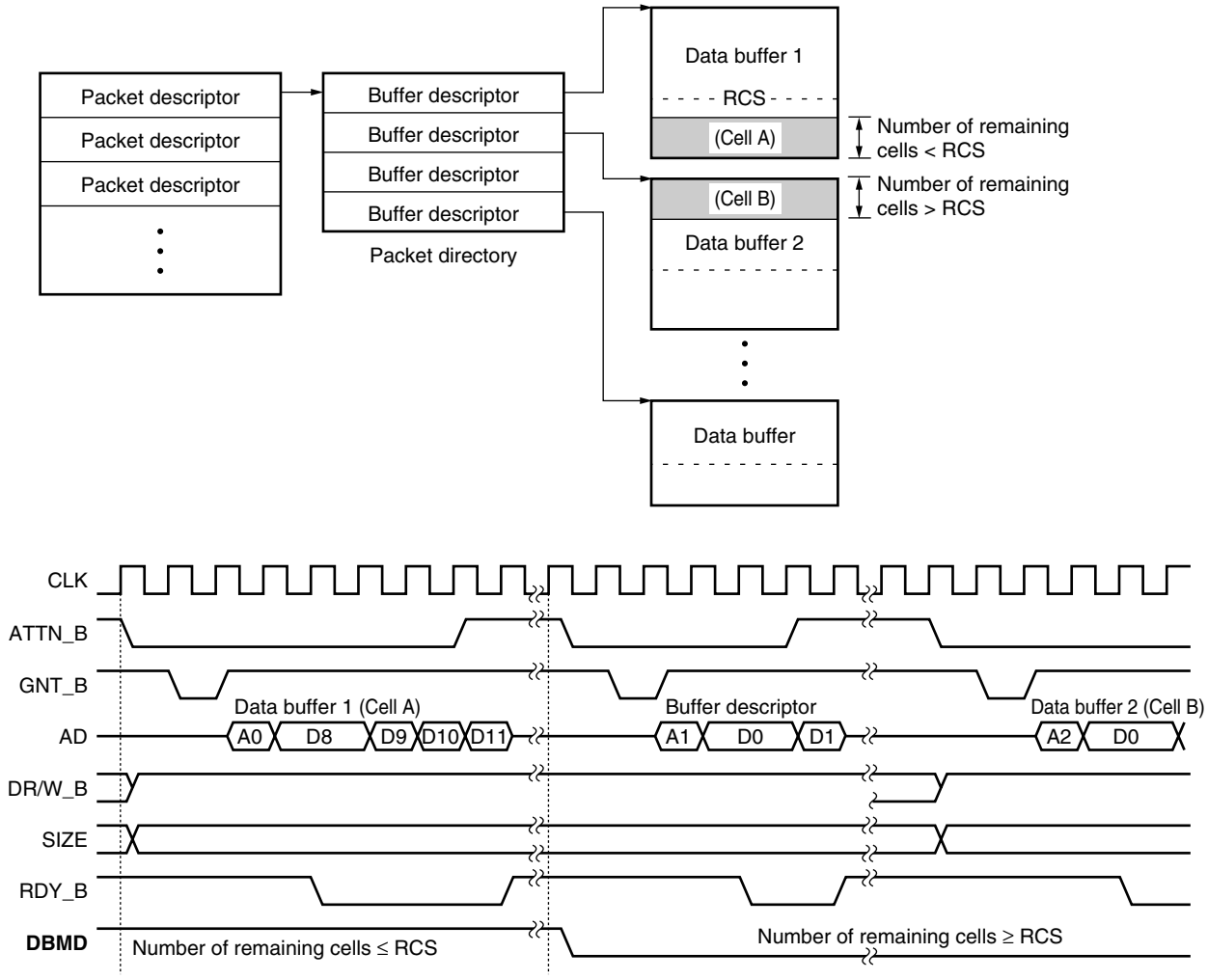
- **Example of DBMR pin function**

Setting condition example: RCS register = 3, transmit data buffer size: 256 bytes (5 cells + 16 bytes)

DMA Read	Number of Data Units Remaining in Transmit Data Buffer	'DBMR' Output at Falling Edge of ATTN_B
DMA read of first cell	5 cells + 16 bytes	Low level
DMA read of second cell	4 cells + 16 bytes	Low level
DMA read of third cell	3 cells + 16 bytes	Low level
DMA read of fourth cell	2 cells + 16 bytes	High level
DMA read of fifth cell	1 cell + 16 bytes	High level
16-byte read of sixth cell	16 bytes	High level

If the transmit data buffer is in the multi-buffer mode and one packet straddles two or more data buffers, the DBMR signal is output for each buffer.

Figure 4-17. DBMR Output Timing Example in Multi-Buffer Mode



For the structure of the transmit data, refer to **5.4.2 Structure of transmit data**.

(e) DBVC pin

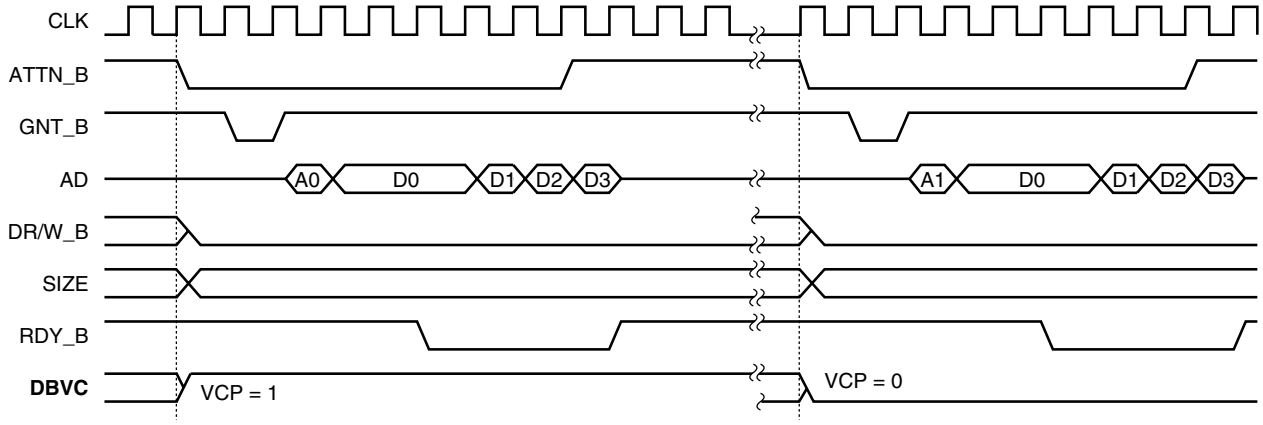
This pin indicates that the receive cell data of a specific VC is being written by means of DMA. When the host sets the VCP bit (word0, bit 22) of the receive VC table to 1, the μPD98401A notifies the host that the cell data of the VC is being transferred. This signal changes its level in synchronization with the falling of the ATTN_B signal, and maintains the level until the next DMA cycle.

High: Transfers the cell data of the AAL-5 packet of the VC for which the VCP bit (word0, bit 22) of the receive VC table is set to 1.

Low : Transfers the cell data of the AAL-5 packet for which the VCP bit of the receive VC table is reset to 0.

For the receive VC table, refer to (2) Setting of receive VC table in 5.5.3 Receive channel.

Figure 4-18. DBVC Output Timing Example



4.1.4 Slave operation

The slave operation is used when the host accesses the direct access register of the μPD98401A, or when it accesses the control memory, indirect address register, or μPD98402A via the direct address register. The slave operation is controlled by the following three pins.

- SR/W_B (slave read/write, input):
  Determines the read/write direction of slave access.
- SEL_B (slave select, input):
  Selects the slave operation of the μPD98401A.
- ASEL_B (slave address select, input):
  Selects the indirect address register of the μPD98401A. When a low level is input to the ASEL_B pin, the μPD98401A samples and loads as an address the low-order 8 bits of AD31 through AD0 at the rising edge of the system clock.

The host makes the slave operation of the μPD98401A active by making the SEL_B signal active (low). It also makes the ASEL_B signal active (low) and outputs the address of the direct address register to be accessed onto the AD bus.

The μPD98401A latches the address from AD31 through AD0 at the rising edge of the clock when a low level is input as the ASEL_B signal. Of AD31 through AD0, the μPD98401A internally decodes only the low-order 8 bits as an address, and ignores the high-order 24 bits.

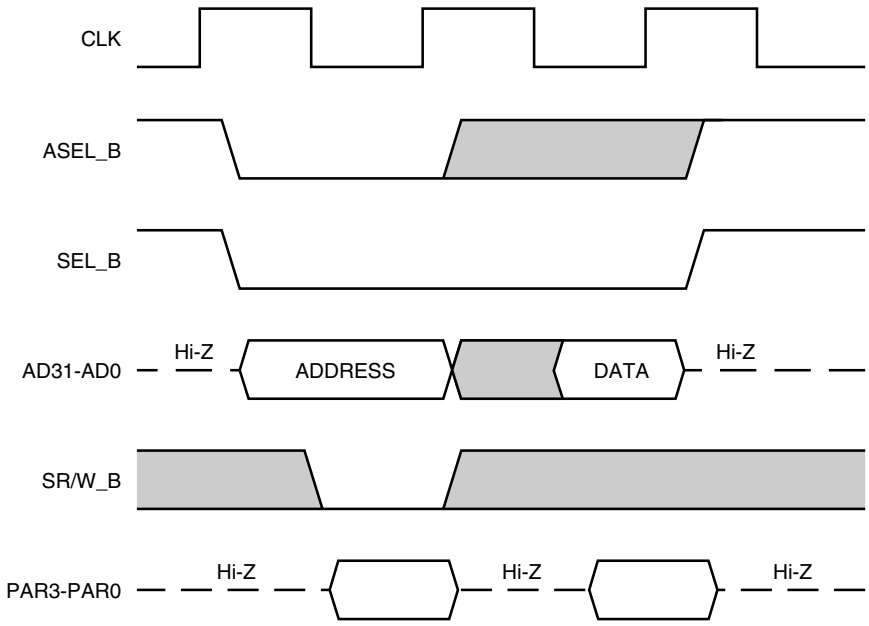
At this time, the μPD98401A also checks SR/W_B to determine the write/read direction.

During slave write, the μPD98401A outputs data to AD31 through AD0 at the rising edge of the clock next to the one at which a low level has been input to the SEL_B signal. The μPD98401A retains the output data on AD31 through AD0 until the SEL_B signal goes high.

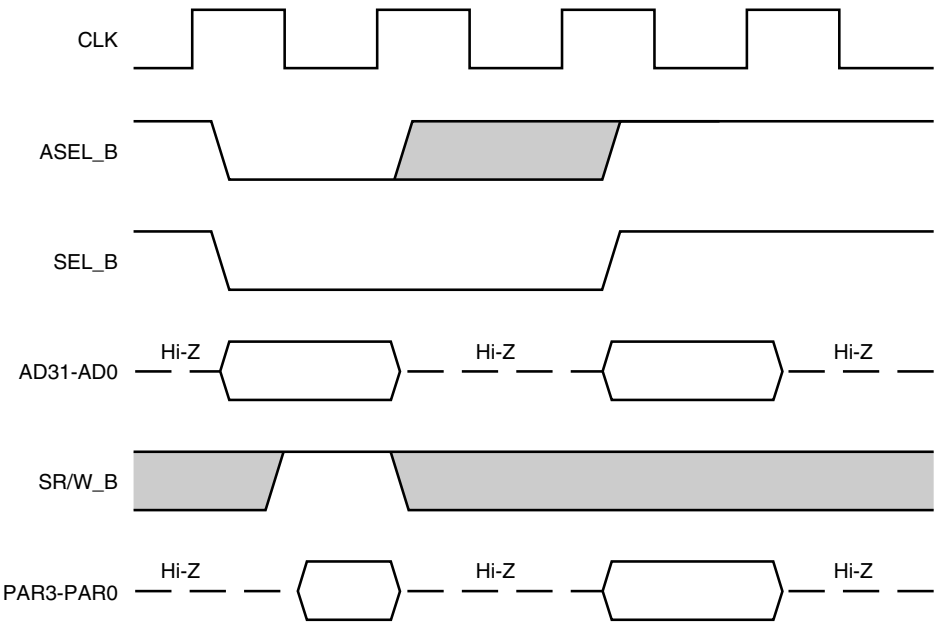
The user can change the timing of data latch and output by extending the low-level width of the SEL_B signal. Figure 4-19 shows the timing of the slave operation.

Figure 4-19. Slave Access Timing

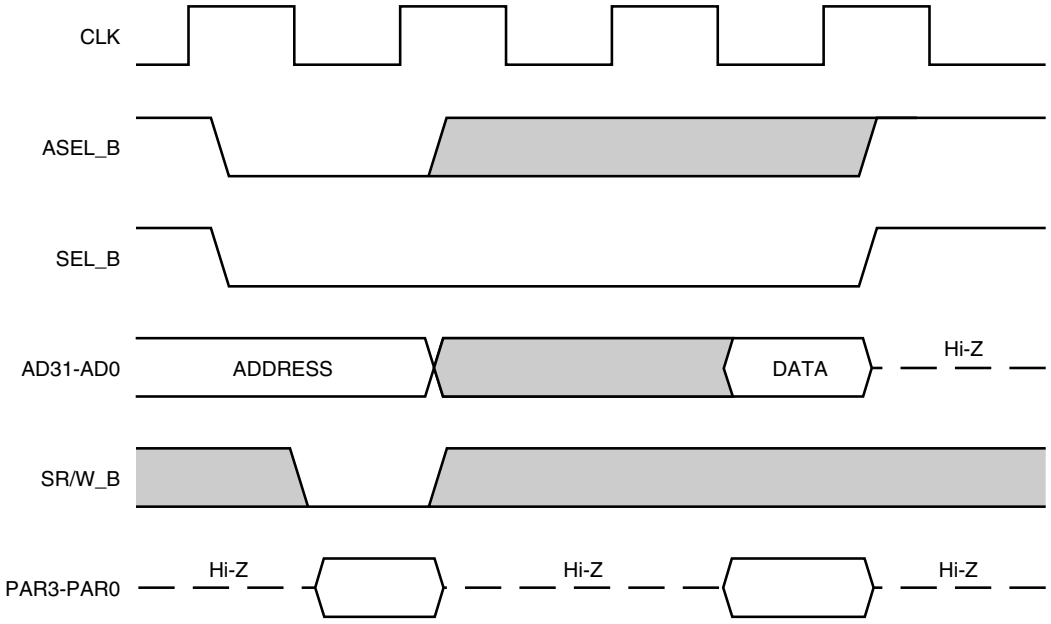
(a) Write timing



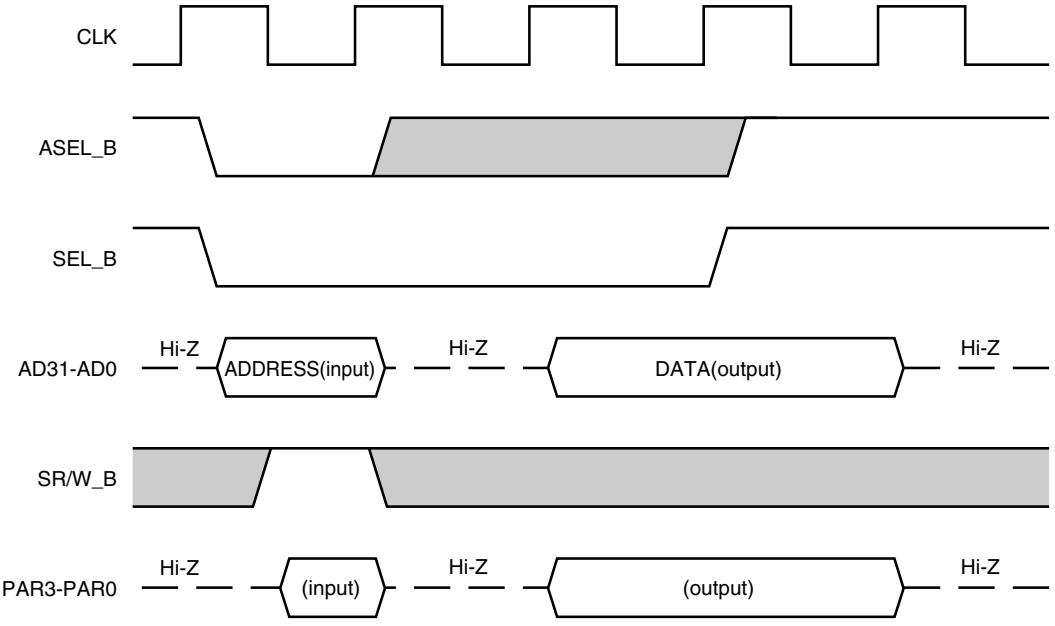
(b) Read timing



(c) Write timing when SEL_B signal is extended



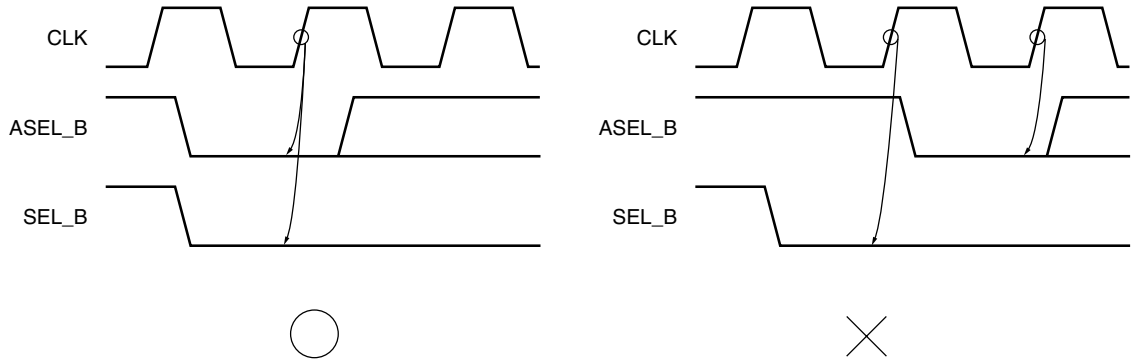
(d) Read timing when SEL_B signal is extended



<Notes on slave operation>

- Slave access cannot be accepted for the duration of 20 cycles of the system clock after reset.
- If a low level is input to the SEL_B or ASEL_B pin while the μPD98401A is performing a master (DMA) operation, the μPD98401A executes the slave operation and samples AD31 through AD0. If the right to use the bus is granted to the μPD98401A by the bus arbiter, the user must prevent the SEL_B and ASEL_B from becoming active, so that the master and slave operations do not contend.
- Do not input a falling edge to the SEL_B signal one clock before the ASEL_B signal. Make sure that the μPD98401A detects the low level of ASEL_B signal at the same time as detection of the low level of the SEL_B signal, or at the rising edge of the clock after that.
  If the low level of the SEL_B signal is detected earlier than the ASEL_B signal, the μPD98401A malfunctions.

Figure 4-20. Input Timing of SEL_B and ASEL_B Signals



- The ASEL_B signal can be raised as soon as or before the SEL_B signal goes high.
- Be sure to insert an inactive period of two system clock cycles or more between when the SEL_B signal has become inactive and when the signal becomes active again.

4.1.5 Little/big endian select function

The bus interface supports both little endian and big endian data formats. These formats can be selected by setting the “BO bit” of the GMR register.

Selects little/big endian		
BO bit (GMR: bit 6)	0	Selects little endian. μPD98401A transfers data in little endian format.
	1	Selects big endian. μPD98401A transfers data in big endian format.
	Default = 0	

In the little endian format, the MSB is stored in the most significant address byte. Bits 31 through 24 of a data word are stored in Byte 3, and Bits 7 through 0 are stored in Byte 0.

Figure 4-21. Format of Little Endian

Halfword1		Halfword0	
Byte3	Byte2	Byte1	Byte0
AD31	AD16	AD15	AD0

In the big endian format, the MSB is stored in the least significant address byte. Bits 31 through 24 of a data word are stored to Byte 0, and Bits 7 through 0 are stored to Byte 3.

Figure 4-22. Format of Big Endian

Halfword0		Halfword1	
Byte0	Byte1	Byte2	Byte3
AD31	AD16	AD15	AD0

Only the byte ordering of the transmit/receive data is changed by setting the little/big endian format. The other information is arranged in the format described in this User’s Manual. You do not have to be aware of the byte ordering in accordance with the setting of the little/big endian format.

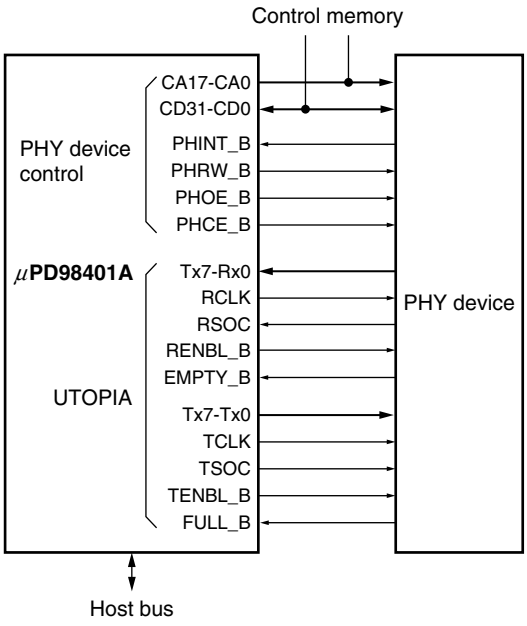
Data handled with format fixed	Receive indication
	Transmit indication
	Packet descriptor
	Buffer descriptor
	Receive batch link pointer
	Buffer address in receive batch
	Indication of Word12 through Word15 of raw cell data
Data whose byte ordering is changed when little/big endian format is set	Receive cell data
	Transmit cell data
	Data of Word0 through Word12 of raw cell data

Caution The raw cell data the μPD98401A transfers to the system memory consists of data and indications. The byte ordering of the 12 words of the data of Word0 through Word11 changes depending on whether the little or big endian format is selected. The 4 words of the indication of Word12 through Word15 are independent of the setting of endianness, and are stored in the format described in this User's Manual. They therefore partially differ from the setting of the little/big endian format.

4.2 PHY Device Interface

This is an interface between the μPD98401A and a PHY device. To interface the PHY device, a UTOPIA interface that transfers cell data between the μPD98401A and the PHY device, and a PHY device control interface that controls the PHY device or obtains statuses, are used.

Figure 4-23. PHY Layer Interface for Data and Control



4.2.1 UTOPIA interface

The μPD98401A 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	

The transmission and reception sides of this interface consist of a clock line through which the clock is supplied by the uμPD98401A, 8 bits of data signal lines, and three control signal lines.

- TCLK (output) : Transmit clock. The system clock input to the CLK 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.

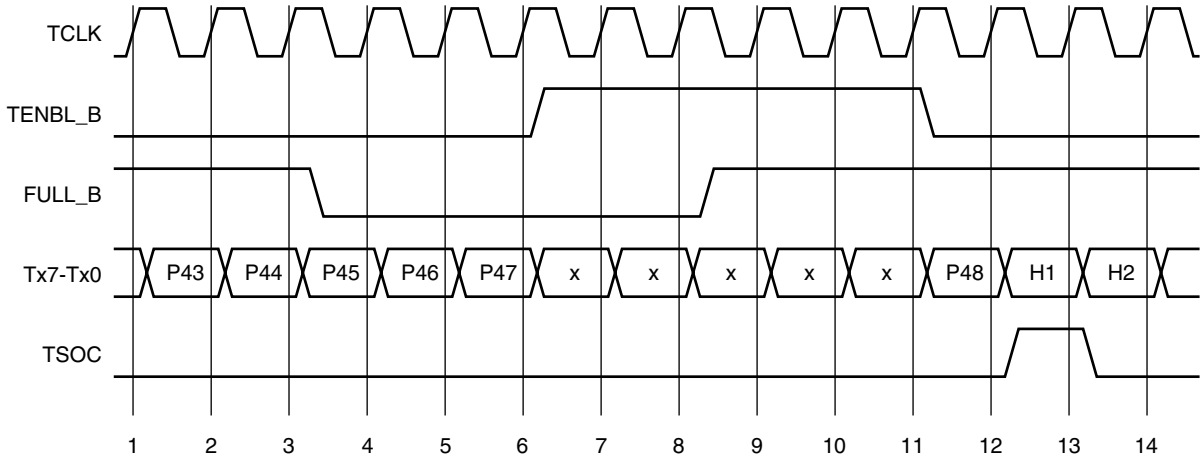
- FULL_B/TxCLAV (input) : This pin differs in meaning depending on whether the octet-level handshake or cell-level handshake mode is used. In the octet-level handshake mode, this pin functions as a FULL_B signal by which the PHY device reports to the μPD98401A that it can receive no more data because the buffer is full. In the cell-level handshake mode, this pin functions as a TxCLAV signal to report to the μPD98401A 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 CLK pin is output as is from this pin.
- RENBL_B (output) : Receive enable signal by which the μPD98401A 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 μPD98401A 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 μPD98401A 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 μPD98401A outputs the system clock input to the CLK 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 data bus 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 μPD98401A. The μPD98401A 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 μPD98401A 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 μPD98401A 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 4-24. Octet-Level Handshake Transmission Timing

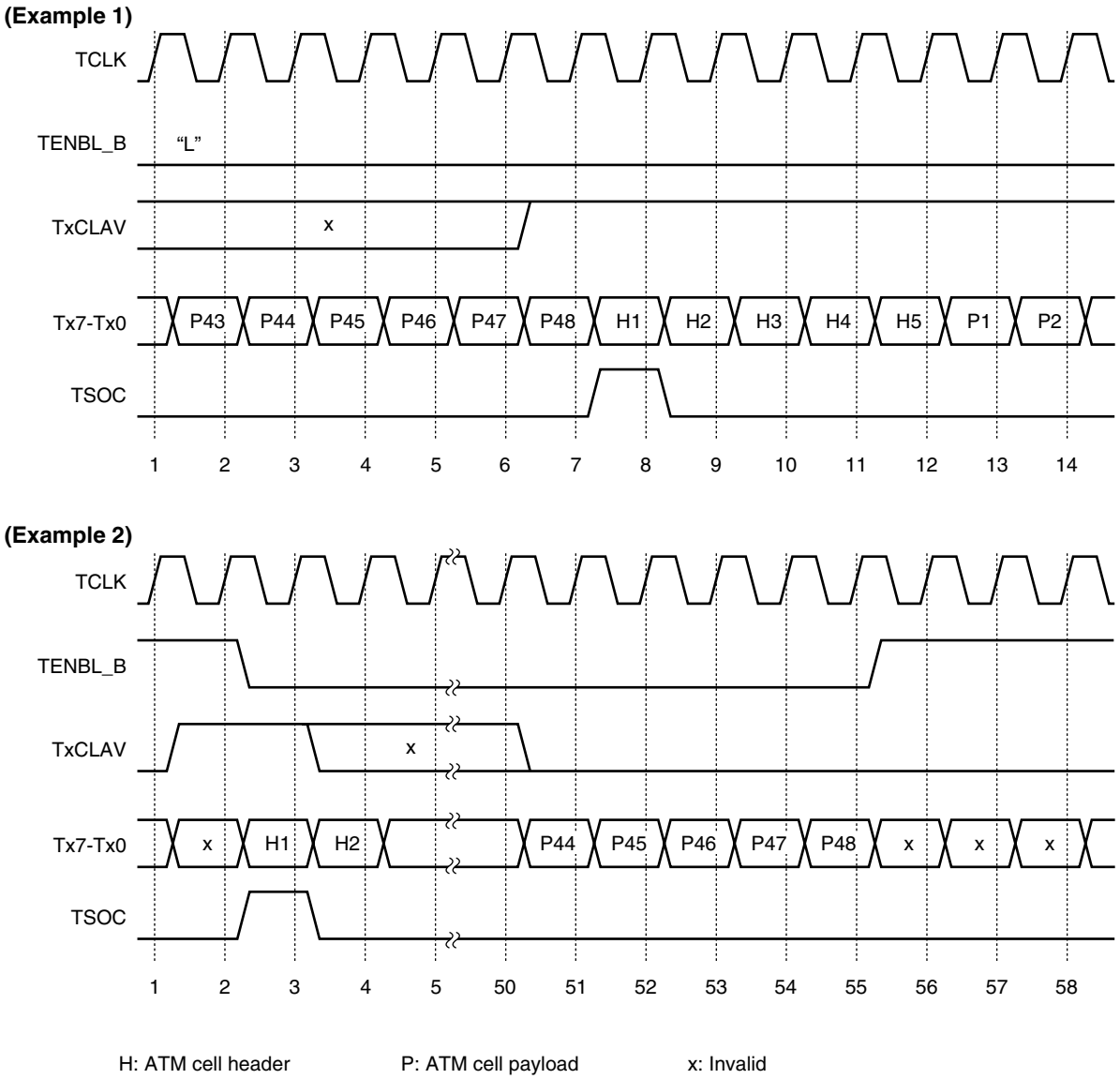


H: ATM cell header P: ATM cell payload x: Invalid

(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 4-25. Cell-Level Handshake Transmission Timing



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

(2) Receive interface

(a) Octet-level handshake

The μPD98401A supplies the system clock input to the CLK 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 bus of Rx7 through Rx0 in synchronization with this RCLK clock.

The μPD98401A 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 μPD98401A, which has a capacity of 23 cells, has become full and therefore, the μPD98401A can receive no more data, the μPD98401A 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 μPD98401A and therefore, the μPD98401A is ready for reception, it makes the RENBL_B signal low again in a cycle one clock before to report to the PHY device.

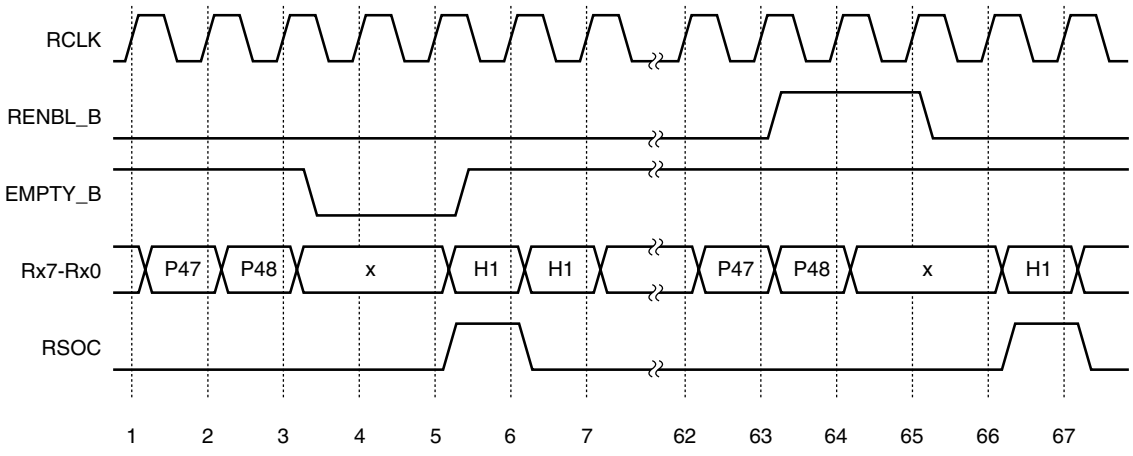
The operation of the RENBL_B signal of the μPD98401A differs depending on whether the DROP mode or No DROP mode is set. These modes can be selected by the DR bit of the GMR register, as follows.

DROP mode/No DROP mode (GMR register: DR bit)	
DROP mode (DR bit = 0)	The μPD98401A 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.
No DROP mode (DR bit = 1)	When the receive FIFO becomes full and therefore, the μPD98401A 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 μPD98401A, the μPD98401A 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 μPD98401A 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 μPD98401A 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 μPD98401A samples the RSOC signal at the rising edge of RCLK. When it detects the high level of the RSOC signal, the μPD98401A starts counting the valid data input at the same rising edge. When the μPD98401A has counted 53 bytes, it assumes that it has received one cell, and executes receive cell processing.

Caution Make sure that the input levels to the Rx7 through Rx0 pins of the UTOPIA receive interface of the μPD98401A do not go into a high-impedance state.

Figure 4-26. Octet-Level Handshake Reception Timing

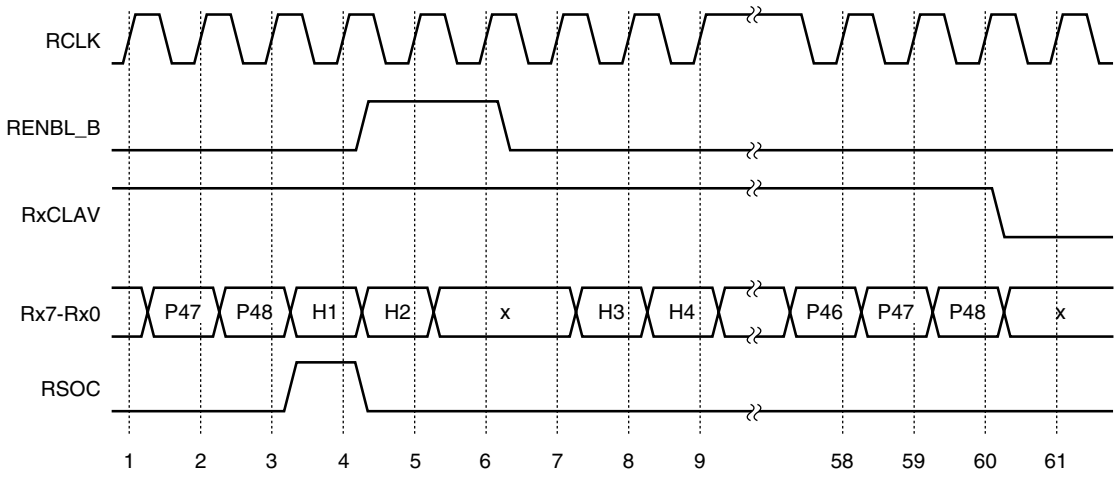


H: ATM cell header P: ATM cell payload x: Invalid

(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 μPD98401A 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 4-27. Cell-Level Handshake Reception Timing



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

4.2.2 PHY device control interface

The μPD98401A 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. Its address bus and data bus are shared with the control memory interface. The control memory interface of the μPD98401A has 32 data bus lines (CD31 through CD0) and 18 address bus lines (CA17 through CA0). The PHY device can be connected to all or some of these 32 data bus lines and 18 address bus lines.

The host uses the Indirect_Access command, one of the commands of the μPD98401A, to access the PHY device. If the Indirect_Access command is issued to the PHY device, the μPD98401A starts a PHY control cycle to manipulate the following control signals.

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

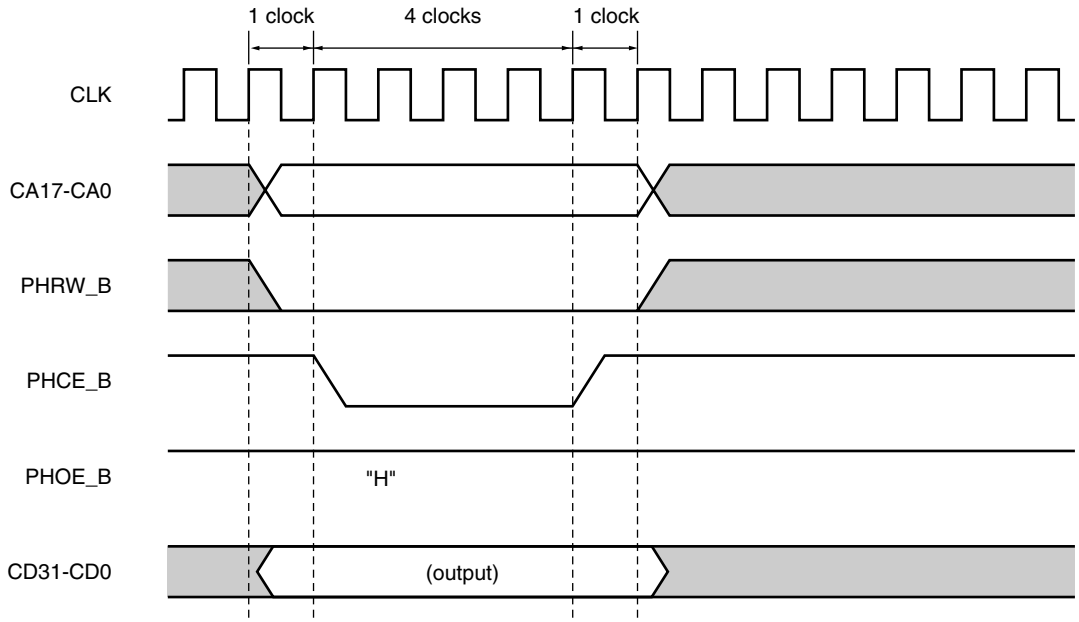
- PHCE_B (output) : Chip enable signal to the PHY device. If the host issues the Indirect_Access command to the PHY device via the μPD98401A, this pin outputs a low level.
- PHRW_B (output) : This pin indicates whether the PHY device is accessed for read or write. It outputs a high level if the device 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 μPD98401A sets the PHY interrupt (PI) bit in the GSR register and generates an interrupt to the host.

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

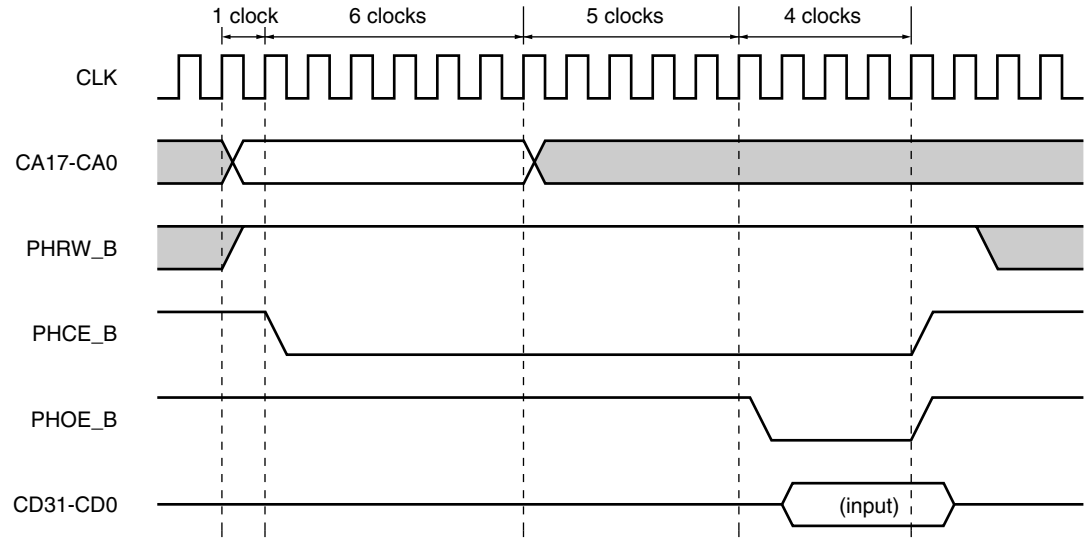
Remark PHINT_B can be used as a general-purpose port pin to detect a level via the μPD98401A, in addition to as a pin to input an interrupt from the PHY device.

Figure 4-28. PHY Device Control Signal Timing

(a) Write operation



(b) Read operation



4.3 Control Memory Interface

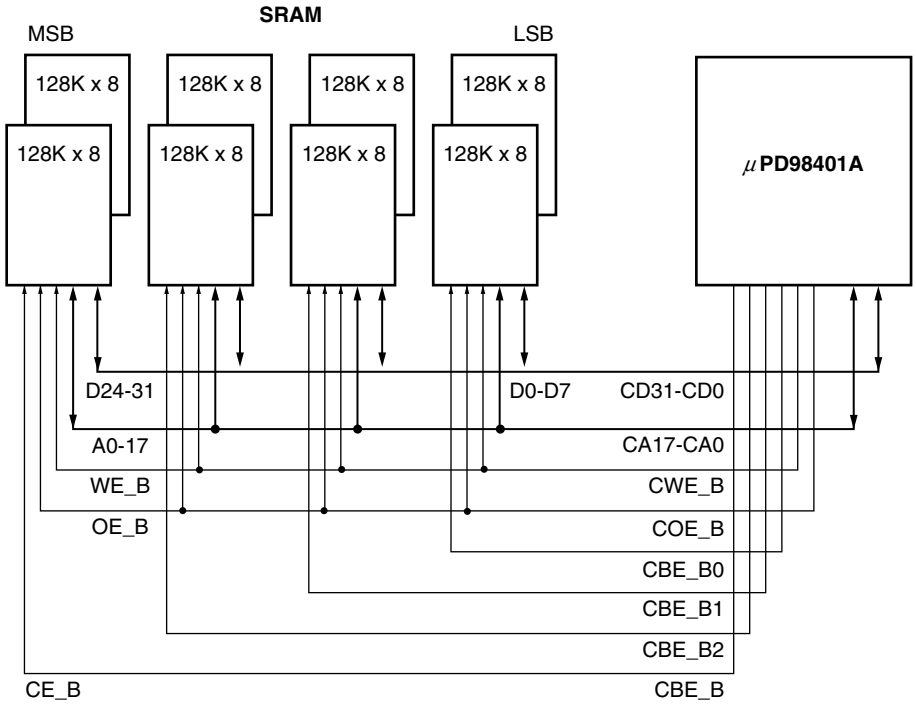
The μPD98401A always requires an external control memory (CM) to execute transmission/reception operation. The control memory interface is used not only to access the control memory, but also as a port to control the PHY device. The control memory interface consists of the following signal lines.

- CD31 through CD0 : 32 data I/O signal lines
- CA17 through CA0 : 18 address signal lines
- CBE_B3 through CBE_B0 : 4 byte enable signal lines
- COE_B : Output enable signal line
- CWE_B : Write enable signal line
- CPAR3 through CPAR0 : Parity bit input/output signal line

The μPD98401A writes/reads the control memory by controlling these signals. The size of the memory to be connected as the control memory depends on the number of channels supported by the μPD98401A, and is 1M bytes of 256K words (1 word = 32 bits) MAX. (how the capacity is determined is explained in detail **in CHAPTER 5 OPERATIONS OF μPD98401A**).

Figure 4-29 shows an example of connecting the control memory.

Figure 4-29. Control Memory Interface Using Standard SRAM



The control memory is basically managed by the μPD98401A. The μPD98401A always toggles the COE_B signal at fixed intervals (low level for 14 clocks in system clock cycle, and high level for 10 clocks) even when it is not performing transmission/reception operation. When the μPD98401A makes COE_B low, it makes CBE_B active and continuously executes the read operation. While the COE_B signal is high, the μPD98401A makes CWE_B and CBE_B active and continuously executes the write operation.

The μPD98401A changes the number of bytes accessed by the control memory, as necessary. The access is made in 1-word (32-bit) units, half-word units (with the low-order or high-order half word accessed at one time), or 1-byte units. The unit in which the access is to be made is selected by the μPD98401A by controlling the CBE_B signal. Therefore, the memory to be connected must be able to be enabled or disabled in 1-byte units.

The control memory interface has functions to append and check a parity bit.

The parity bit is input/output via CPAR3 through CPAR0. CPAR3 through CPAR0 indicate the parity of CD31 through CD0 in 8-bit units, and calculate the parity in even number. The parity bit input from CPAR3 through CPAR0 is checked during read operation. The function to check the input parity bit is enabled or disabled by bit 15 (CPE bit) of the GMR register.

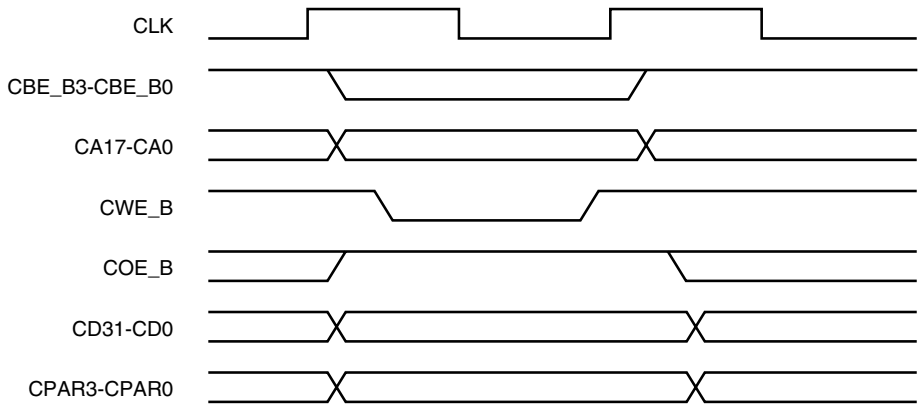
Selects control memory parity check function (GMR register: bit 15)		
CPE bit	0	Disables the control memory parity check function. The μPD98401A does not detect the error of the parity bit input from CPAR3 through CPAR0.
	1	Enables the control memory parity check function. The μPD98401A detects the error of the parity bit input from CPAR3 through CPAR0. If it has detected an error, it sets the CPE bit of the GSR register and reports to the host by using an interrupt (if not masked).
	Default = 0	

During write operation, the parity operation bit for 8 bits each is output from CPAR3 through CPAR0. The parity bit is always output regardless of the setting of the CPE bit of the GMR register. The pins perform a 3-state operation regardless of the setting of the CPE bit. When the CPAR3 through CPAR0 pins are not used, pull them up to prevent malfunctioning due to generation of a through current by an intermediate potential.

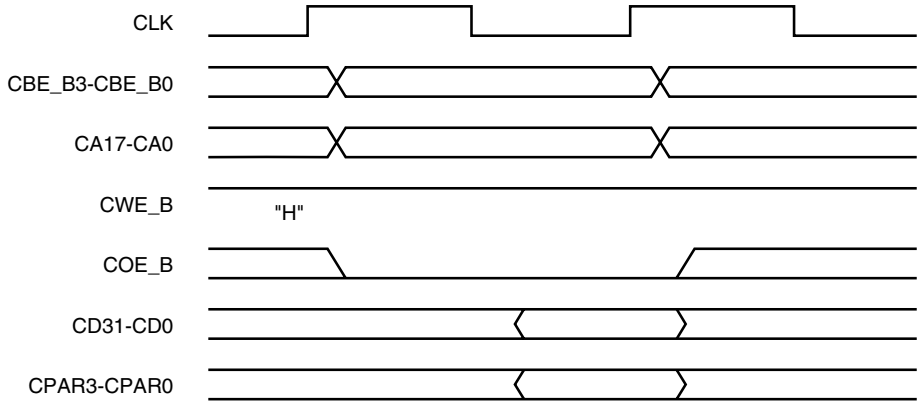
Caution If a parity error is detected on the control memory interface, the operation of the μPD98401A is not guaranteed. Reset the μPD98401A if a parity error has been detected.

Figure 4-30. Control Memory Access Timing

(a) Write



(b) Read



CHAPTER 5 OPERATIONS OF μPD98401A

This chapter explains the functions of the μPD98401A.

5.1 Initialization

(1) Initializing the chip

So that the μPD98401A may function appropriately, the chip must be initialized to set the internal state machine and registers in specific status. Hardware reset is executed if a low level is input to the RST_B pin. The chip can be also initialized by software reset during normal operation. Software reset is executed by performing a write operation on the SWR register. Both hardware reset and software reset have the same effect on the μPD98401A, and there is no difference in status after reset.

Caution The μPD98401A requires a time of 20 clock cycles to initialize the internal circuitry of the chip. Therefore, do not execute slave access to the μPD98401A within 20 clock cycle time after reset.

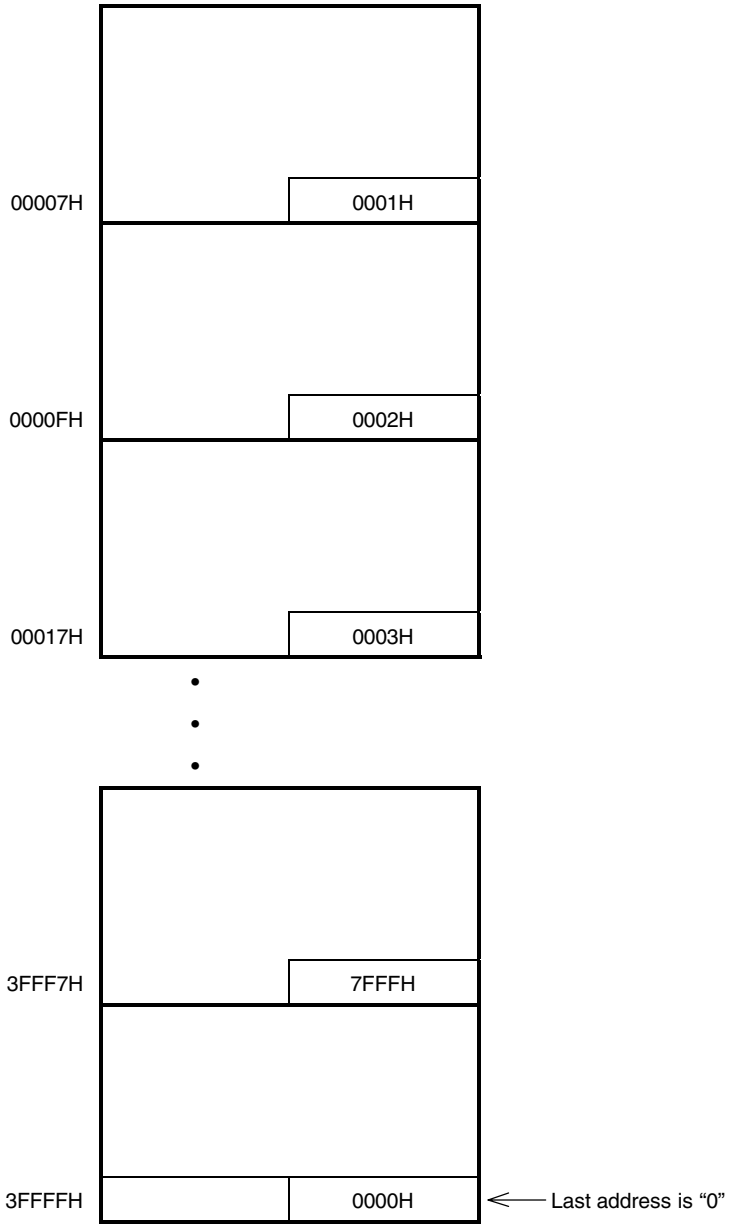
(2) Initializing control memory

The control memory connected to the μPD98401A is initialized after hardware or software reset. The μPD98401A always operates assuming that the maximum of 256K words (1M-byte size) are assigned as the area of the control memory.

When reset, the μPD98401A writes “0000H” to the low-order 16 bits of the address “3FFFFH” of the last word as a block number. It then decrements the address to “3FFF7, 3FFEF, 3FFE7, 3FFDF” and so on, in 8-word units, and repeats writing a block number decremented by one, like “7FFF, 7FFE, 7FFD”, and so on, to the low-order 16 bits, 32K times. After a block number “1H” has been written to address “00007H”, the μPD98401A writes “all 0” to address “3FFFFH” again to complete the initialization.

The written block numbers function as pointers that indicate the first address of the next block, and blocks divided by 8 words are chained on all areas. The information on this chaining is used by the μPD98401A to manage the “free block pool” area of the control memory, and is rewritten as necessary to open/close VC and perform transmission/reception processing.

Figure 5-1. Control Memory after Initialization



This automatic initialization processing can be disabled when the μPD98401A is tested. To disable the initialization, input "1" (high level) to external pin INITD. During normal operation, input "0" (low level) to the INITD pin.

INITD pin input	0	The μPD98401A initializes the control memory after it has been reset.
	1	The μPD98401A does not initialize the control memory after it has been reset.

Any area other than those with a block number is cleared to 0.

After reset, the μPD98401A immediately starts writing block numbers to the control memory. Because one block number is written to one block, it takes 32K clock cycles to complete writing all the block numbers. At this time, access using the control memory interface cannot be made, and only slave access to the direct address registers other than CMR/CMR_L, CER, and CER_L can be made.

The μPD98401A sets the IND bit of the GSR register and generates an interrupt, if not masked, after completing initialization of the control memory.

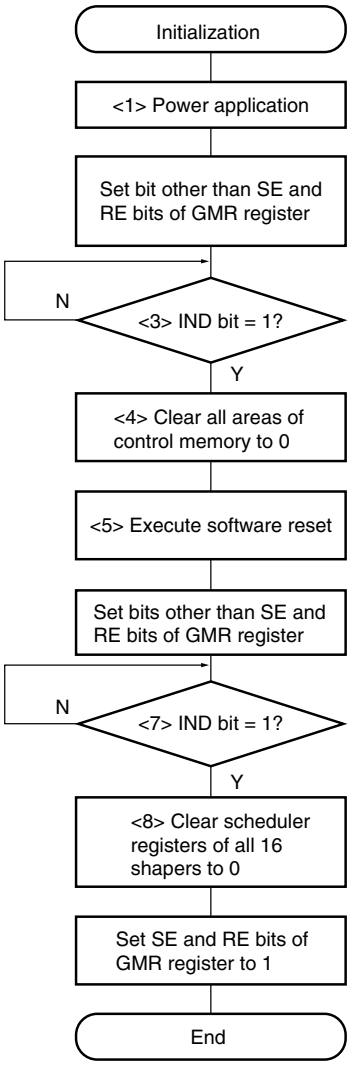
<Notes on initializing control memory>

As described above, the μPD98401A does not clear the contents of the control memory to 0 in the initialization process. However, some of the areas of the control memory do not operate normally unless the initial value is “0” when the control memory is read. Therefore, clear the contents of the control memory to 0 in software after power application.

<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. The μPD98401A will start writing block numbers to the control memory.

<2> The host sets a parity mode to the GMR register. At this time, write 0 to the SE and RE bits.
The parity bit output function of the μPD98401A is always enabled. If the slave register is read before the parity mode is set after power application, the μPD98401A outputs the parity bit in the default mode, causing the host system to detect a parity error. To avoid this, set the parity bit of the GMR register first.
The SE and RE bits are not set because the parameter of the scheduler register must be cleared to 0 before the SE bit is set to 1.

<3> Wait until the IND bit of the GSR register is set. 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).

<4> If the IND bit is set to 1, the Indirect_Access command can be used. The host writes 0 to all the areas of the control memory by using this command.

<5> Execute software reset. The μPD98401A writes block numbers again over the control memory that has been cleared to 0.
<6> Set a parity mode to the GMR register again. At this time, do not set the SE and RE bits yet.
<7> Check to see if the IND bit is 1, and confirm that the write operation of the block numbers is completed.
<8> Clear all the areas of the scheduler registers (5 words x 16) to 0 by using the Indirect_Access command. This is because the variables and flags of the scheduler registers must start operating from 0 at the beginning.
<9> Set the SE and RE bits of the GMR register to 1.

5.2 Setting of Control Memory

The size of the control memory ranges from 0 word (1 word = 32 bits) to 256K words, and is determined by the number of channels supported by the μPD98401A. The control memory is divided into the following four areas ("1 block" in the explanation below means 8 words).

(1) Receive lookup table:

This area stores "enable bits" and "VC Number" in accordance with the pattern of VPI/VCI to be received. It occupies up to 4096 blocks. The number of blocks to be occupied depends on the type of receive VPI/VCI supported. For details, refer to **5.5.4 Setting of receive lookup table**.

(2) Receive free buffer pool pointer:

This area stores "pool descriptors". Its size depends on the number of pools set. Because up to thirty-two 2-word descriptors can be set, up to eight blocks are occupied. For details, refer to **5.5.2 (2) Receive free buffer pool pointer**.

(3) Transmit shaper pointer:

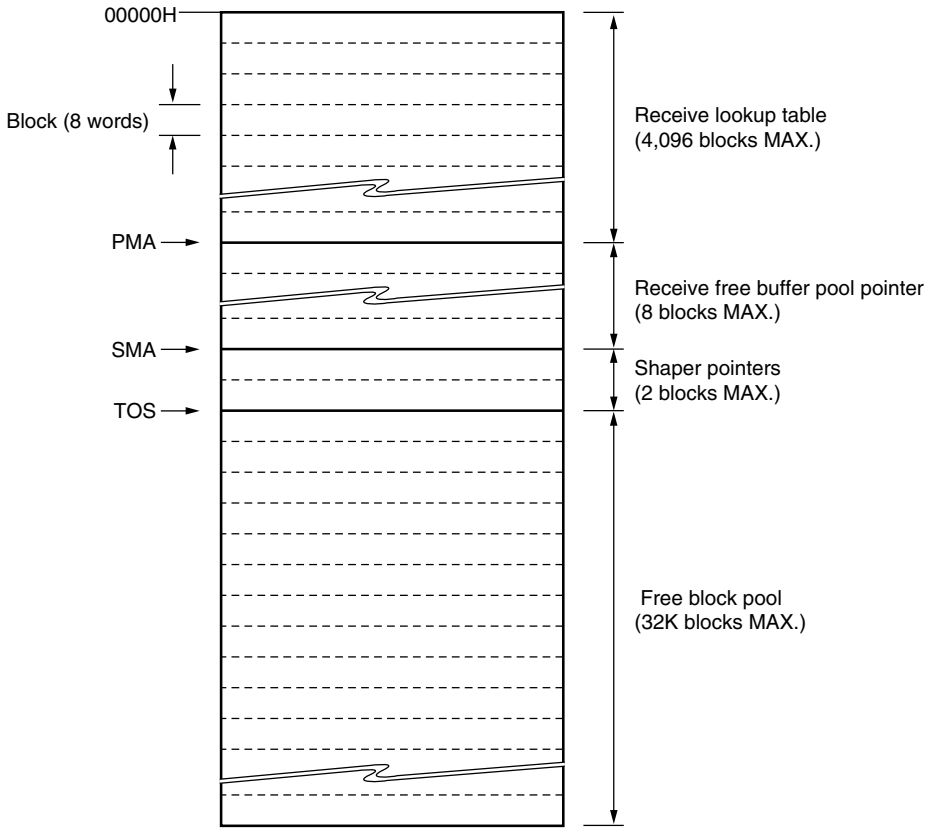
This is an area used by the μPD98401A for transmission processing. Its size depends on the number of shapers used. Because one word is used per shaper, a maximum of two blocks are occupied if all the 16 shapers are used. For details, refer to **5.4.4 (5) Shaper pointer entry**.

(4) Free block pool:

This area stores transmit/receive VC tables. Its size depends on the number of VC tables set, and is in a range of 0 to 32K blocks. For example, because one transmit/receive VC table uses one block (8 words), 32 blocks are occupied when 16 transmit/receive channels are set.

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 control memory is divided into areas before the user starts a transmit/receive operation. The start address of each area is set to the indirect register of the μPD98401A. The following three registers are used to set start addresses.

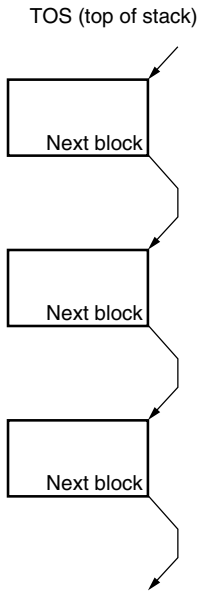
- PMA register: Sets the start address of the receive free buffer pool descriptor area
- SMA register: Sets the start address of the shaper pointer.
- TOS register: Sets the start address of the free block pool

Set these three areas in accordance with the following rules.

<1> The receive lookup table must always start from address 0 and have size of 2^n words.
<2> The receive free buffer pool must always start from an address having its size information in the low-order bits (for example, if the size is 64 words, all the low-order 6 bits of the start address (PMA) must be 0).
<3> The shaper pointer must always start from an address having its size in the low-order bits (for example, if the size is 16 words, all the low-order 4 bits of the start address (SMA) must be 0).

Although an initial value is set to TOS (Top of tack) by the user, the μPD98401A indicates the beginning of the free block pool and changes TOS as the stack pointer as shown below. The μPD98401A sequentially connects 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 μPD98401A returns the address indicated by TOS from the free block pool by using command indication, and allocates blocks. At this time, the μPD98401A 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 μPD98401A returns a block to the free block pool and allocates it to TOS.

Figure 5-4. Stack of Free Block Pool



Remark The μPD98401A 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.

<Example of determining size of control memory>

As an example, the size necessary for the control memory is calculated under the following conditions:

Condition 1: Type of receive VPI/VCI supported = Up to 64K types of VPI/VCI supported
Condition 2: Number of pools set = Pool 0 only
Condition 3: Number of shapers used = Shaper 0 only
Condition 4: Number of transmit/receive channels supported = 16 transmit channels
16 receive channels

(Size)

Receive lookup table	= 32768 words (supports up to 65536 types of VPI/VCI)
Receive free buffer pool pointer	= 2 words (1 pool descriptor consists of 2 words)
Shaper pointer	= 1 word (1 shaper pointer entry consists of 1 word)
Free block pool = Transmit 16 x 8 + Receive 16 x 8	= 256 words (1 VC table consists of 8 words)
	Total: 33027 words

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 μPD98401A. An indication is status information the μPD98401A 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 μPD98401A 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 μPD98401A 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 μPD98401A
- Mailbox tail address (MTA) : Low-order 16 bits of the address updated up to where the host has read

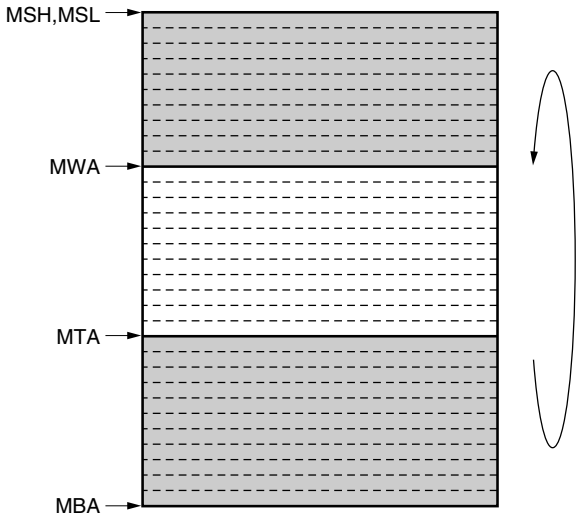
All the pointers of a mailbox are 16 bits long. The μPD98401A concatenates a pointer with the 16 bits of the mailbox start address high to create an actual 32-bit address. The maximum size of one mailbox is 64K bytes. All the pointers of a mailbox are allocated to the direct address register of the μPD98401A. 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 μPD98401A is executing a transmission/reception operation. The μPD98401A 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.

Figure 5-5. Structure of Mailbox



5.3.2 Operation of mailbox

The μPD98401A increments the write pointer (MWA) when it has written an indication. Each time the μPD98401A 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 μPD98401A updates the write pointer (MWA), it jumps MWA to the start address (MSL).

The read pointer (MTA) is used to prevent the μPD98401A 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 μPD98401A.

The μPD98401A compares the write pointer (MWA) and read pointer (MTA) before writing the indication. If these pointers are found to be equal, it assumes the mailbox full (MF) status and sets the MF bit corresponding to the mailbox of the GSR register to issue an interrupt if not masked. The μPD98401A does not issue the next indication to the mailbox in the MF status (MWA = MTA). The host must read indication from a mailbox in the MF status and quickly update MTA.

Caution If the μPD98401A 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 μPD98401A

(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 μPD98401A transmits a cell in several steps.

Figure 5-6. Outline of Transmission Flow

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

	Operation of μPD98401A
(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 μPD98401A 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 μPD98401A 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 μPD98401A 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 μPD98401A to read the transmit queue. The μPD98401A 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 μPD98401A 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 μPD98401A writes transmit indication (12) to the mailbox as status information, and issues an interrupt. The host reads the mailbox (13) and updates the read pointer of the mailbox in the μPD98401A (14).

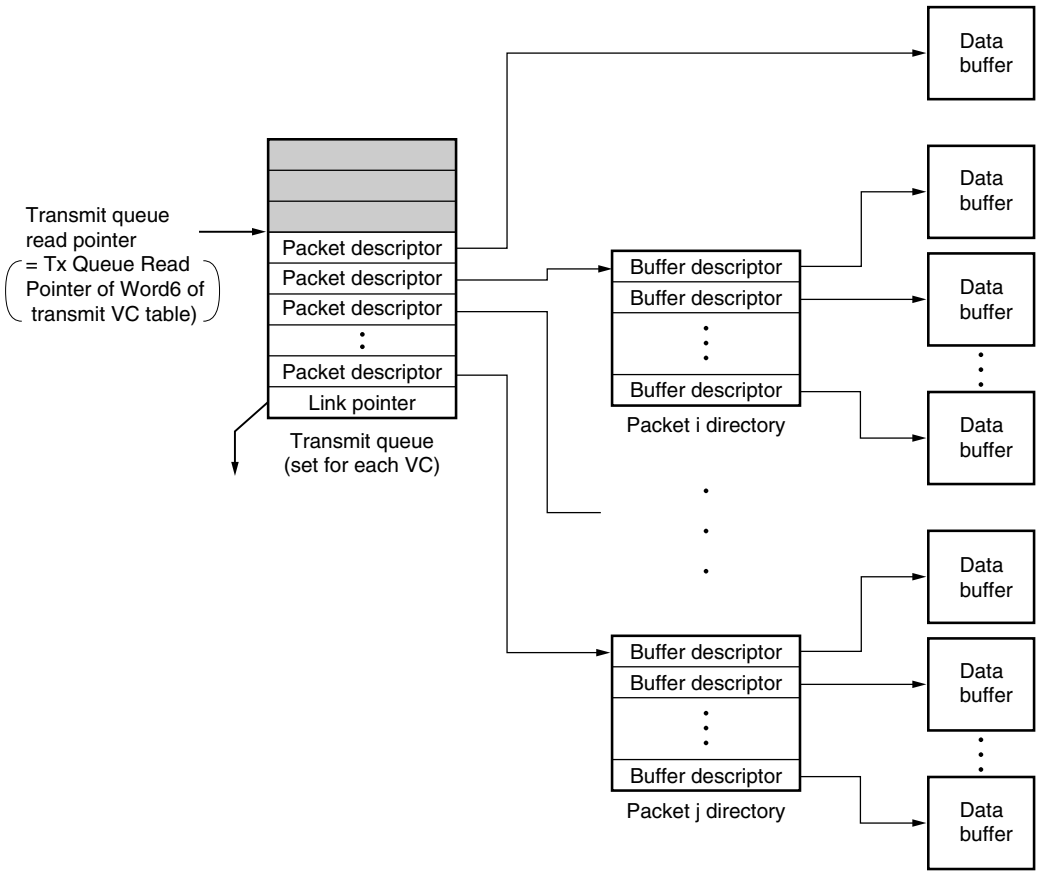
The transmission processing flow in Figure 5-6 is shown only for the purpose of illustration, and actually, the host does not perform its processing in synchronization with the software processing.

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

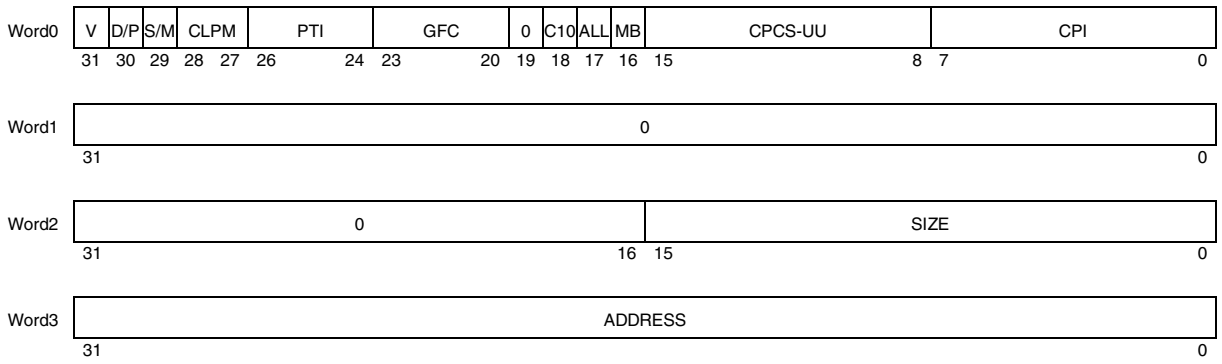
The host creates and manages a transmit queue for each channel (transmit VC). The transmit queue consists of the packet descriptor of the corresponding channel. A packet has one or more data buffer. The read pointer of the transmit queue is initialized to the corresponding transmit VC table of the control memory by the host. After that, the read pointer is corrected and managed by the μPD98401A each time the transmit packet descriptor has been fetched.

While a VC exists, do not release the transmit queue for the VC.

The transmit queue is set for each VC, and the initial value of its read pointer (Tx Queue Read Pointer) is assigned to the transmit VC table by the host. After that, the value of the read pointer is updated each time the μPD98401A has transmitted a packet. The μPD98401A reads the packet descriptor in the system memory in accordance with this read pointer during transmission. Because the host manages the packet descriptor asynchronously with the read access by the μPD98401A to the packet descriptor, if the host releases the transmit queue area in the system memory, the μPD98401A cannot correctly obtain information from the packet descriptor, and may malfunction. To release 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

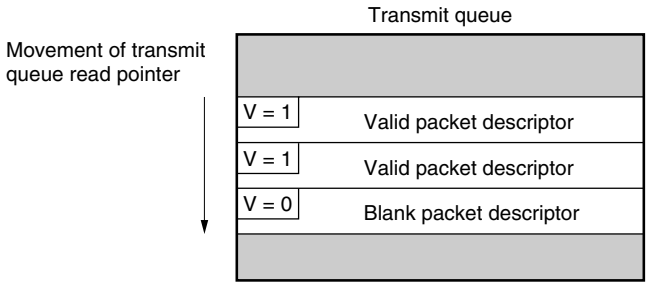
V	D/P	S/M	Function
0	–	–	Blank packet descriptor
1	0	–	Link pointer
1	1	0	Packet descriptor in multi-buffer mode
1	1	1	Packet descriptor in single-buffer mode

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 μPD98401A 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 μPD98401A ignores these areas. Because the μPD98401A reads the packet descriptor in 4-word units, however, the blank packet descriptor must also consist of a 4-word area.

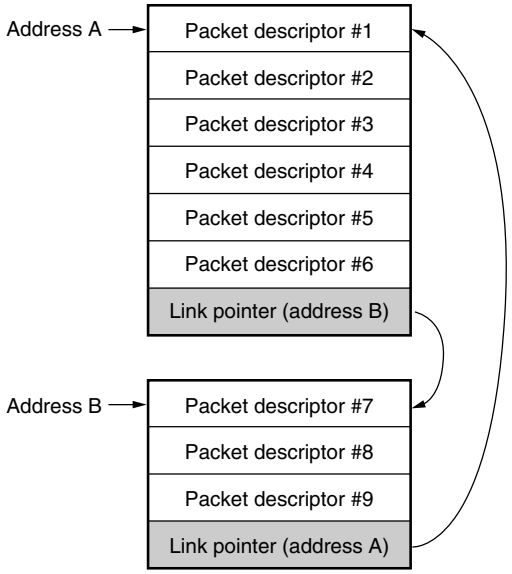
Caution The host must set the V bit to 1 after it has set all the areas other than the V bit of the packet descriptor. If Word1 through Word3 are set after the V bit of Word0 has been set to 1, the μPD98401A completes the processing of the preceding packet descriptor before the host has set all the valid 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 μPD98401A ignores these fields.

Figure 5-10. Example of Arrangement of Transmit Queue

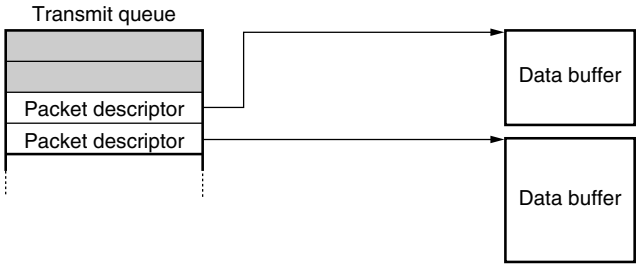


The S/M bit of the packet descriptor indicates whether this transmit packet is constructed in the single-buffer mode or multi-buffer mode.

- **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.

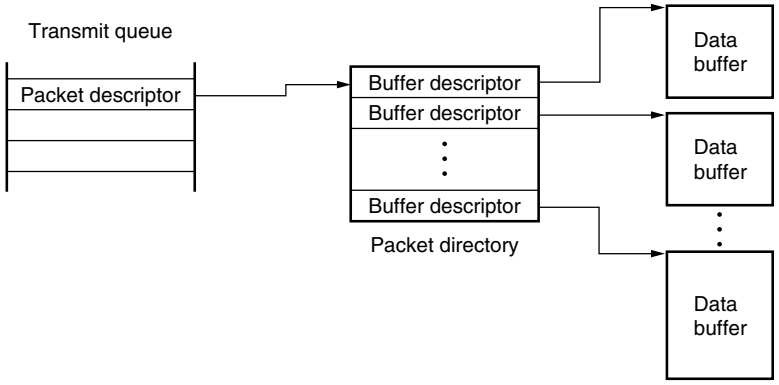
Figure 5-11. Single-Buffer Mode



- **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 μPD98401A 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 μPD98401A 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 μPD98401A 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 μPD98401A 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 μPD98401A 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 μPD98401A changing the low-order 1 bit to "1".
  - **AAL type other than AAL-5 (AAL bit = 0):**

    The μPD98401A inserts the contents of this field as is to all the headers of the transmit packet. The μPD98401A 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 μPD98401A 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 μPD98401A performs 10-bit CRC operation and inserts an error detection code to each cell.
	0	The μPD98401A does not insert CRC-10.

If this bit is set to "1", the μPD98401A 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

Header	Payload	CRC-10
5 bytes	46 bytes + 6 bits	10 bits

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 μPD98401A, the following points must be noted.

<Cautions>

- **The μPD98401A 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 μPD98401A 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 μPD98401A processes this packet as AAL-5-PDU.
	0	The μPD98401A 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 μPD98401A 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 μPD98401A 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 μPD98401A ignores this field.

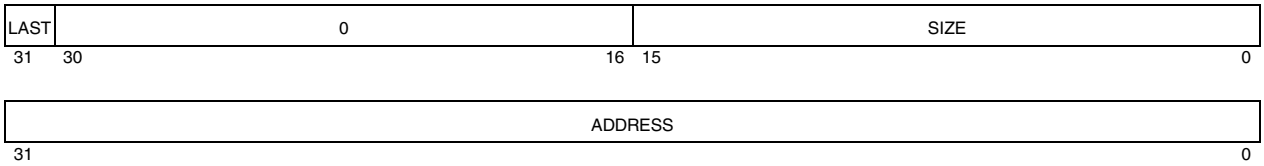
<10> ADDRESS

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.

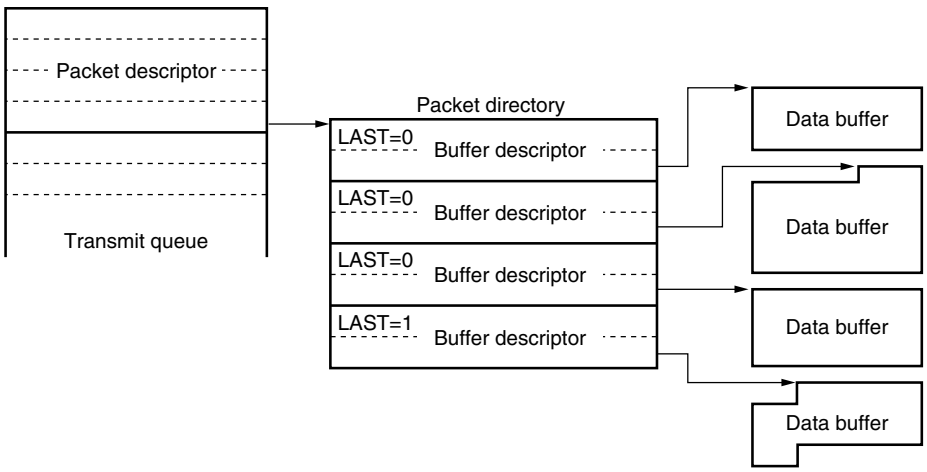
Figure 5-14. Format of Packet Descriptor



<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

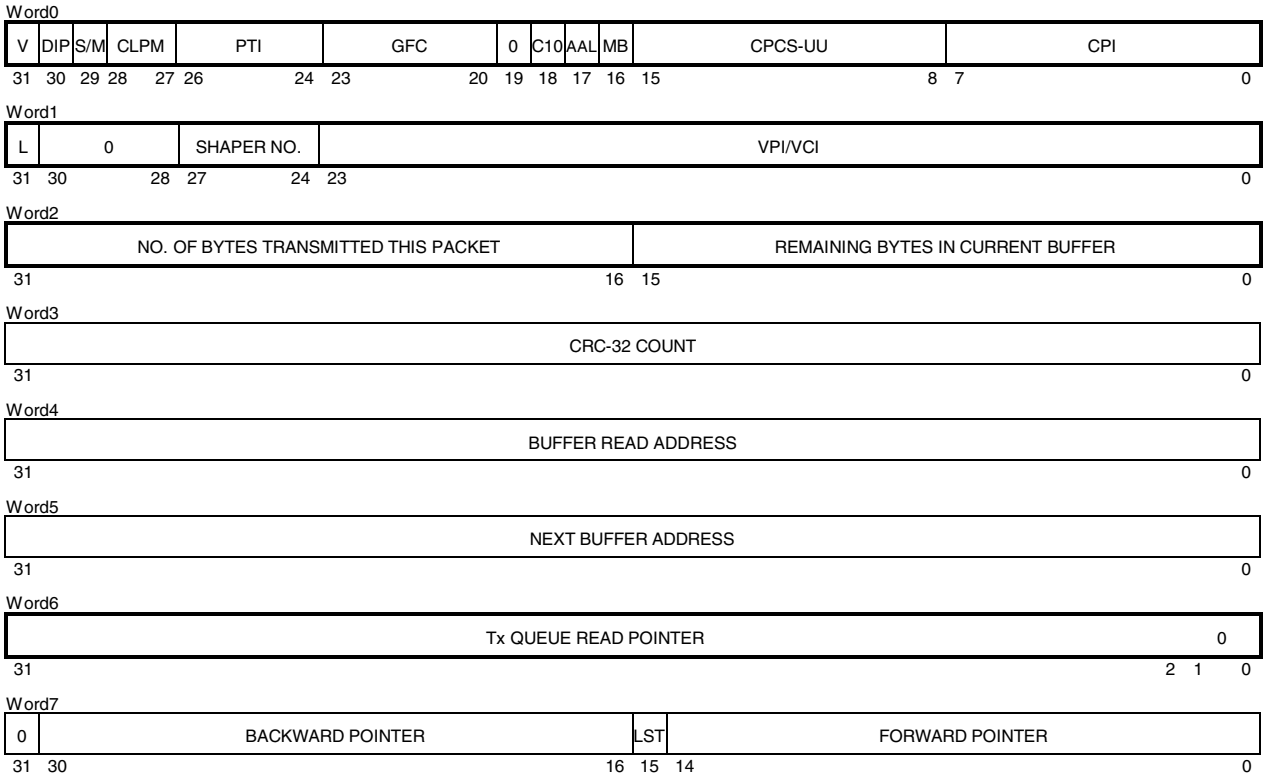
When the host issues the Open_Channel command, the μPD98401A allocates the block indicated by TOS (top of stack) from the free block pool of the control memory, and returns its first address to the host by using command indication. The host sets the allocated block as a transmit VC table.

(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 μPD98401A for transmission. The host can read the table at any time and use it as status information.

Caution When using an allocated block as a transmit VC table for the first time, word 0 must be cleared to all 0. Otherwise, the μPD98401A may not correctly execute its transmission operation. There is no problem if the all the areas of the control memory are cleared to 0 on first resetting the chip.

Figure 5-16. Structure of Transmit VC Table



Word0 ---------- The initial value must be cleared to all 0. Stores the contents of Word0 of the packet descriptor which the μPD98401A has read from the system memory as is.

L ---------- This bit is used by the μPD98401A as a flag. Be sure to set "1" to this bit as the initial value.

SHAPER NO. ---------- Sets the shaper number 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 μPD98401A 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 μPD98401A to manage the VC table. If this word is changed by an external host, the μPD98401A 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 μPD98401A 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 μPD98401A. When the μPD98401A 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 μPD98401A 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 μPD98401A 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 μPD98401A starts transmitting a cell. The μPD98401A also add 4 words so that the transmit queue read pointer of the transmit VC table indicates the next packet descriptor.

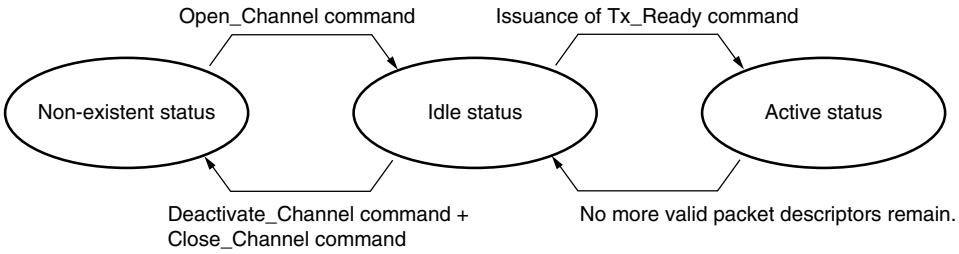
When the μPD98401A completes reading all the data of the packet, it fetches the next packet descriptor. If this packet descriptor is valid, the VC remains in the active status and continues transmitting the packet. While a valid packet descriptor exists in the transmit queue, the channel remains in the active status.

When the μPD98401A has read a blank packet descriptor (V = 0), it recognizes that there no more packets remain. When the μPD98401A 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 μPD98401A 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 μPD98401A 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

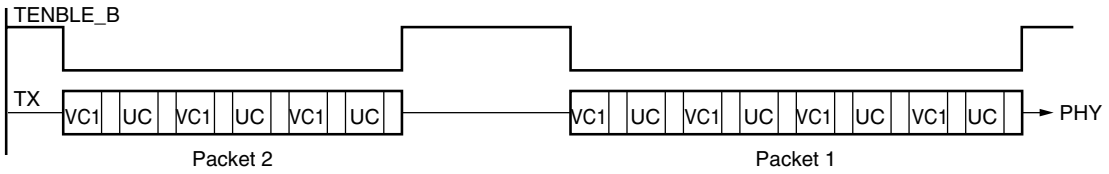


5.4.4 Traffic Control

(1) Transmitting cells

The μPD98401A transfers cells to the PHY device via the UTOPIA interface. Cells are successively transferred if at least one of the transmit VC is active, or if there is a shaper set as an unassigned/idle cell generator; otherwise, with the TENBL_B signal is made inactive and cells are not transferred. The transmission rate of cells is controlled in cell units and up to 16 different rates can be selected.

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



VC: data cell
UC: unassigned cell

(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 16 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

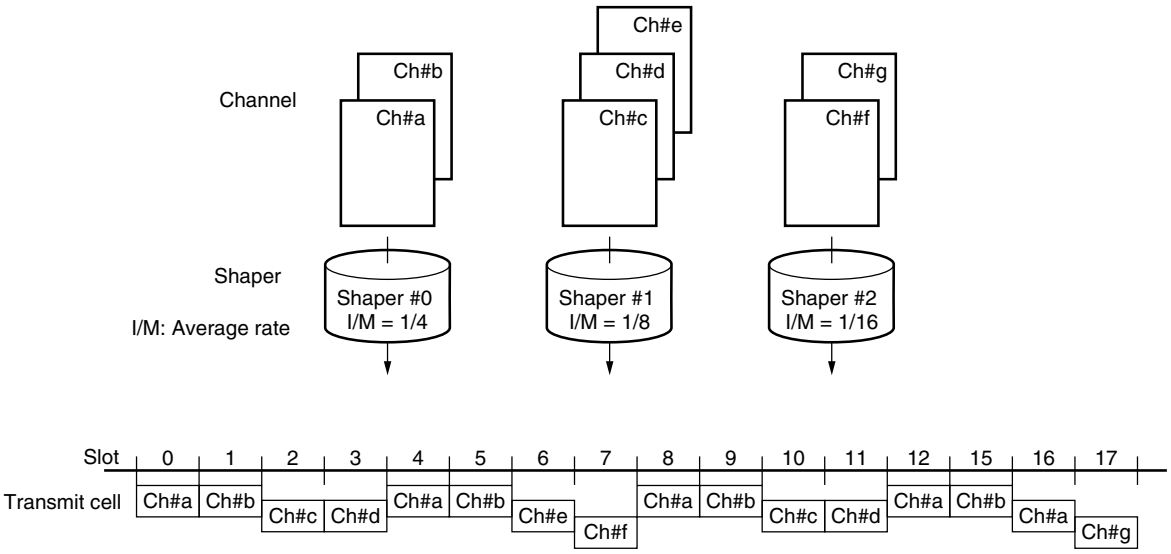
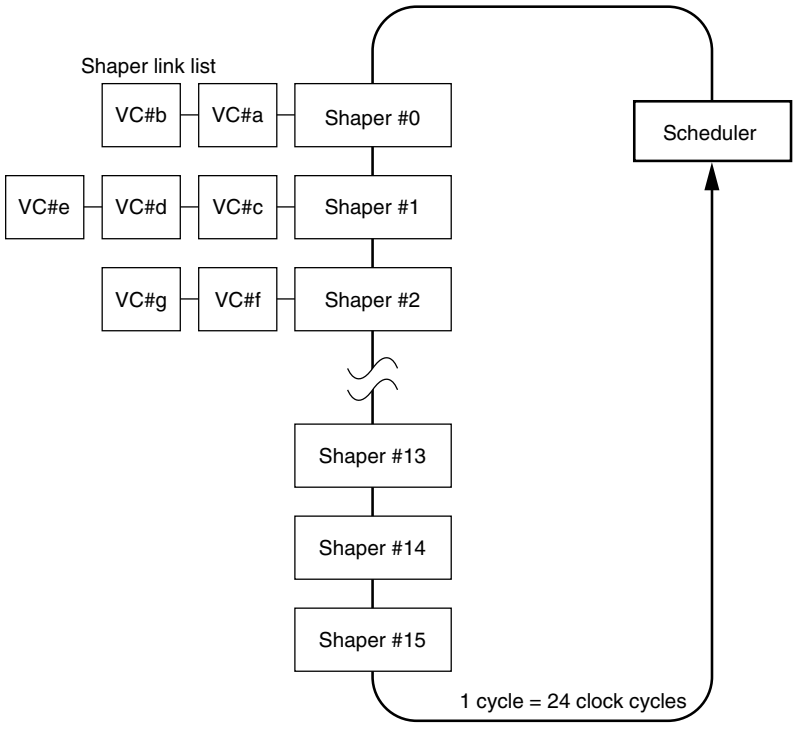


Figure 5-20. Concept of Scheduler Operation



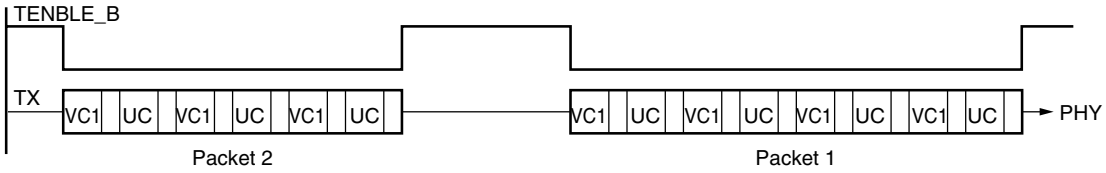
(3) Transmitting vacant cell

The μPD98401A 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 μPD98401A transmits a vacant cell. The user can select the type of vacant cell transmitted by the μPD98401A 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.

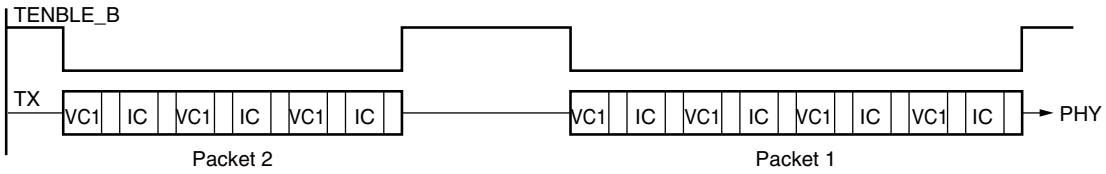
Selects vacant cell		
ICM bit (GMR: bit 28)	0	Inserts unassigned cell
	1	Inserts idle cell
	Default = 0	

<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 16 shapers has a scheduler register that stores the parameters set by the host and the variables managed by the μPD98401A.

The user assigns the following 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.

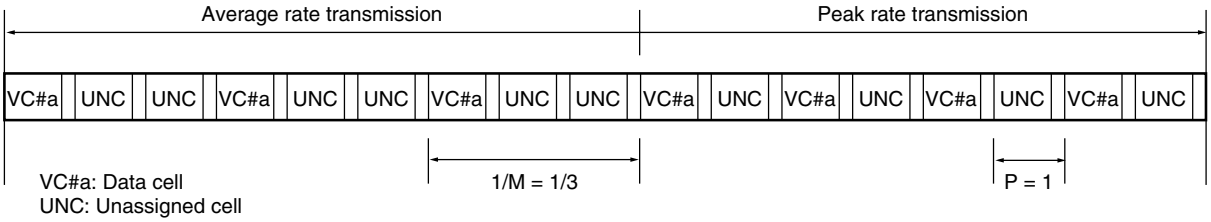
Table 5-3. Description of Bits of Scheduler Register

Bit Name	Description
I (8 bits), M (24 bits)	The quotient of I/M sets the average rate allocated to a shaper (valid cell of I cell is transmitted in M cell time).
P (peak, 8 bits)	These bits set the minimum interval between the cells of the same VC in cell units.
C (credit, 8 bits)	These bits set the maximum number of cells that can be successively transmitted at the peak rate. These bits must always be set to 1 or more.
AGM (aggregate mode, 1 bit)	This bit selects a transmission rate control mode. 0: Normal mode. The set rate parameter is adapted in VC units. 1: Aggregate mode. The set rate parameter is adapted in shaper units. For details, refer to **5.4.4 (7) Cell transmission mode of shaper in scan status**.
PRIORITY (4 bits)	These bits set a priority to be allocated to a shaper. The highest priority is “0000”, and the lowest priority is “1111”. When a priority is set, a shaper transmits cell data only when a shaper with higher priority is not waiting for transmission. **Remark** The user must set priorities to shapers 0 through 15, starting with the shaper with the youngest number (If the priority of shaper 0 is the highest, and i > j, the priority of shaper i must be the same as or lower than the priority of shaper j).
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 (number of cells that can be successively transmitted at peak rate)



The μPD98401A 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 μPD98401A, 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 These variables and flags used by the μPD98401A must start from initial value 0. The default value of all the scheduler registers is "undefined" after reset. Therefore, the host must initialize the variables and flags of all the 16 shapers to 0 before the SE bit of the GMR register is set to "1" after reset.

Because the set rate is changed, the parameters of I, M, P, and C, and all the contents of the scheduler registers cannot be changed while the shaper is active (A = 1) and transmitting a packet. If they are changed, the μPD98401A cannot correctly keep the rate. When changing the values of I, M, P, and C to change the set rate, clear the enable bit E to 0, and clear all the contents of the registers only when the shaper is inactive (A = 0) (except, however, the manipulation of 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 μPD98401A automatically links the transmit VC to the shaper or releases the VC from the link.

The μPD98401A 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 μPD98401A 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

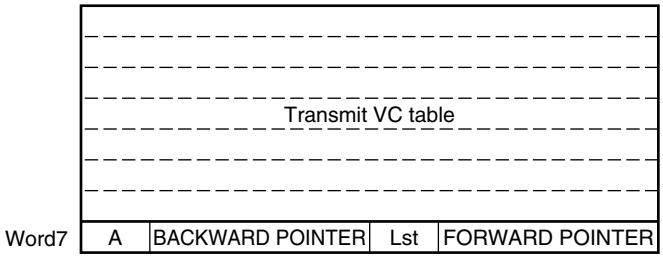
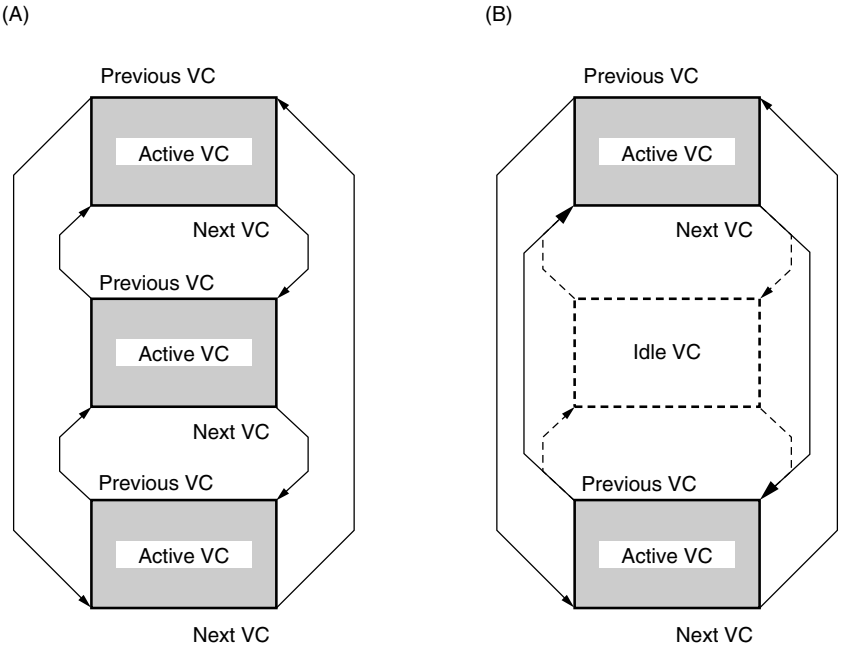


Figure 5-23. Linking/Releasing Transmit VC to/from Traffic Shaper



(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 μPD98401A 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 μPD98401A 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 μPD98401A 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 μPD98401A 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 μPD98401A determines for which VC it should perform processing when it has moved between shapers.

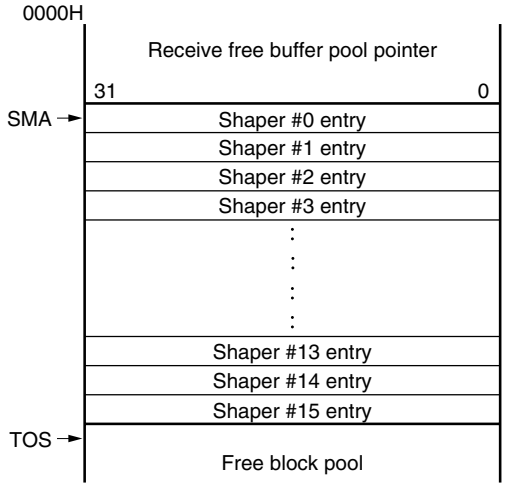
(6) Shaper pointer entry

The shaper pointer entry is a table that stores a record indicating up to which VC the μPD98401A has performed processing when the μPD98401A 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 up to 16 words. The μPD98401A sequentially uses this area as the entry of shaper 0, entry of shaper 1, and so on, starting from the address of SMA. For example, when the user enables and uses shaper 3, the μPD98401A accesses the one word of the address resulting from incrementing the SMA address by 4, as the entry of shaper 3. Therefore, if the user needs only shapers 0 through 3, the size the user must set as the shaper pointer area is only four words from the SMA address.

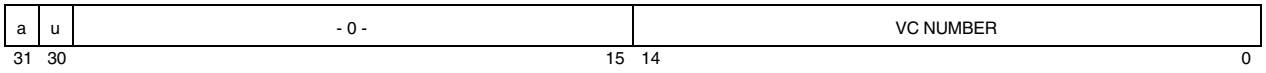
The shaper pointer entry is mainly used and managed by the μPD98401A as a table. The host reads or writes this area by using the Indirect_Access command. However, the host only accesses this area when it clears the area to 0 on initialization, or when it uses the unassigned cell generator function; otherwise, the host only does not access the area.

Remark The shaper pointer entry must be always cleared to 0 when the μPD98401A access this area for the first time to start transmission. Clear all the areas of the control memory to 0 after power application.

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 μPD98401A.
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.

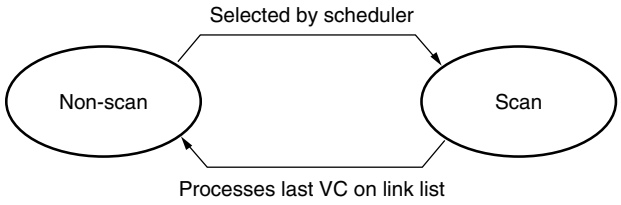
u bit ---------- Unassigned cell generator. This bit is set to 1 by the host only when this shaper is used as an unassigned cell generator (for details, refer **to 5.4.4 (9) Unassigned (idle) cell generator function**).

VC NUMBER -- This field is used as a table to which the μPD98401A 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 bit = 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 μPD98401A.

(iv) c variable > 0 (scheduler register)

The μPD98401A updates the c variable according to the I, M, and C parameters given by the user.

(v) p variable = 0 (scheduler register)

The μPD98401A 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 μPD98401A, 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 → I and M are loaded to x and y, respectively.
- When x > y → I is added to x and M is added to y.
- When x < y → I is added to x.

"c" is incremented where x ≥ 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".

Here is an example:

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	2		
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.

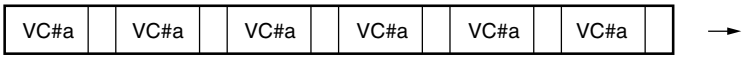
<Example> Setting of shaper (Priority is not considered.)

- 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 all the 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. However, the user must increase the priority from all the youngest of shaper numbers 0 to 15 (if shaper 0 has the highest priority and if i > j, the priority of shaper i must be equal to or lower than the priority of shaper j).

Setting of "Priority"

Correct: Shaper#0, Shaper#1, Shaper#3, Shaper#4 ...

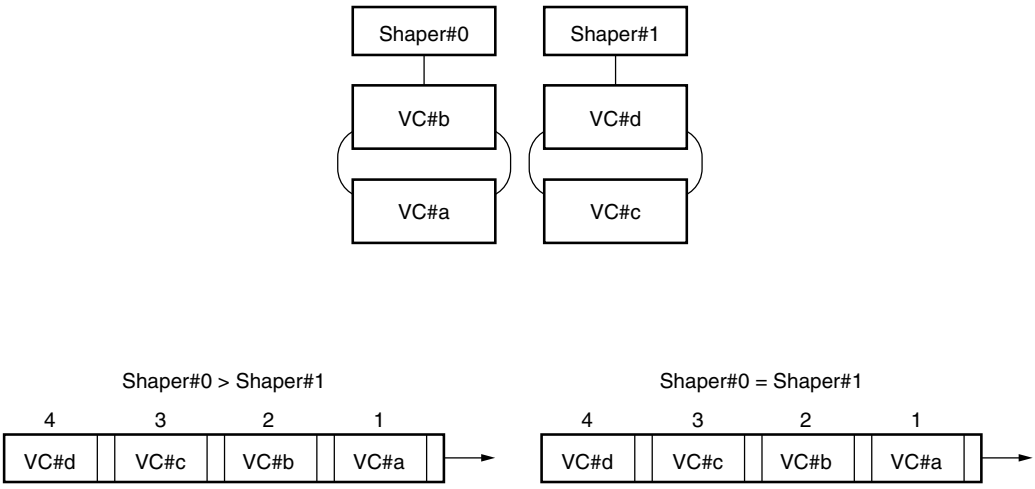
High → Low

Incorrect: Shaper#4, Shaper#2, Shaper#1, Shaper#3 ...

If two or more shapers having the same priority exist, the scheduler selects one shaper at a time by means of round robin. The μPD98401A 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 in the scan status sends linked VC cells. After the cells have been transmitted, the shaper returns to the non-scan status in either of the following two modes. By changing the mode in which the shaper returns to the non-scan status, the rate parameter set for the shaper can be used to select a mode that controls the rate in VC units, or a mode that controls the rate in shaper units.

These modes can be selected for each shaper, by using the AGM bit of the scheduler register.

(a) Normal mode (AGM bit = 0)

Once a shaper has entered the scan status it sequentially transmits all the linked VC cells on a one-by-one basis, and returns to the non-scan status after the cell of the VC (LST bit = 1) linked last. In this mode, the transmit rate set for the shaper is controlled in VC units. However, the bandwidth occupied by one shaper changes with the number of the VCs linked to that shaper.

(b) Aggregate mode (AGM bit = 1)

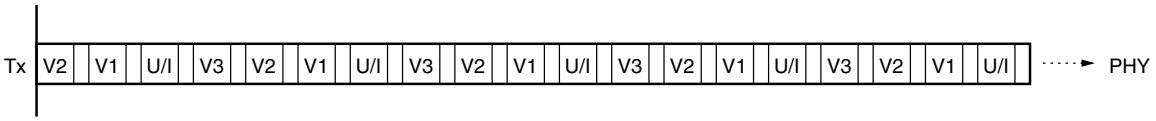
A shaper in the scan status returns to the non-scan status when it has transmitted one cell of 1VC of the linked VCs. When the shaper enters the scan status again, it transmits one cell of the next VC on the link list.

In this mode, the set rate parameter controls the bandwidth of the shaper. The transmission rate of each VC is obtained by dividing the bandwidth given by the shaper by the linked VCs. The bandwidth occupied by the shaper is not affected by the number of VCs.

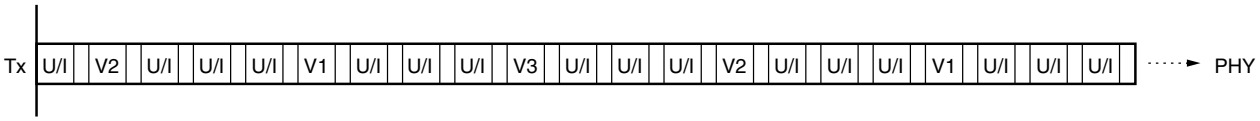
Figure 5-27. Differences in Cell Transmission Procedure by Mode

When 3 VCs (VC1, VC2, and VC3) are linked to shaper of I/M = 1/4

Normal mode



Aggregate mode



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

V# : Cell data of Tx VC, V1 = VC1 data, V2 = VC2 data, V3 = VC3 data

(9) Unassigned cell generator function

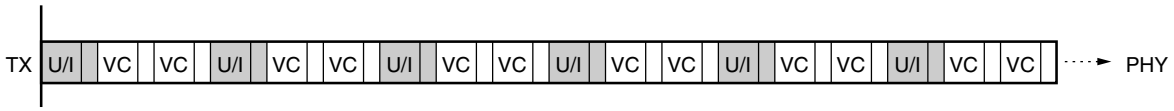
The user can use one or more shaper as an unassigned cell generator to limit the band used by the active VC. The shaper set as an unassigned cell generator functions as if it were always linked to the VC that transmits a cell with the header and data field being all 0. By assigning a priority higher 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.

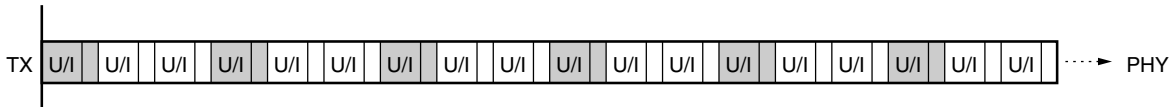
Cautions 1. A normal VC cannot be linked to the shaper set as an unassigned cell generator.
2. The host does not have to set the “a” bit of the shaper pointer entry and “A” bit of the scheduler register to “1” when the unassigned cell generator function is not used. These bits are automatically set to “1” by the μPD98401A for the shaper for data transmission.

<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 Transmission operation

The μPD98401A 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 μPD98401A has a FIFO of 10 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 μPD98401A 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 μPD98401A 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 μPD98401A 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 μPD98401A manages the information on the current transmit buffer by using the VC table.

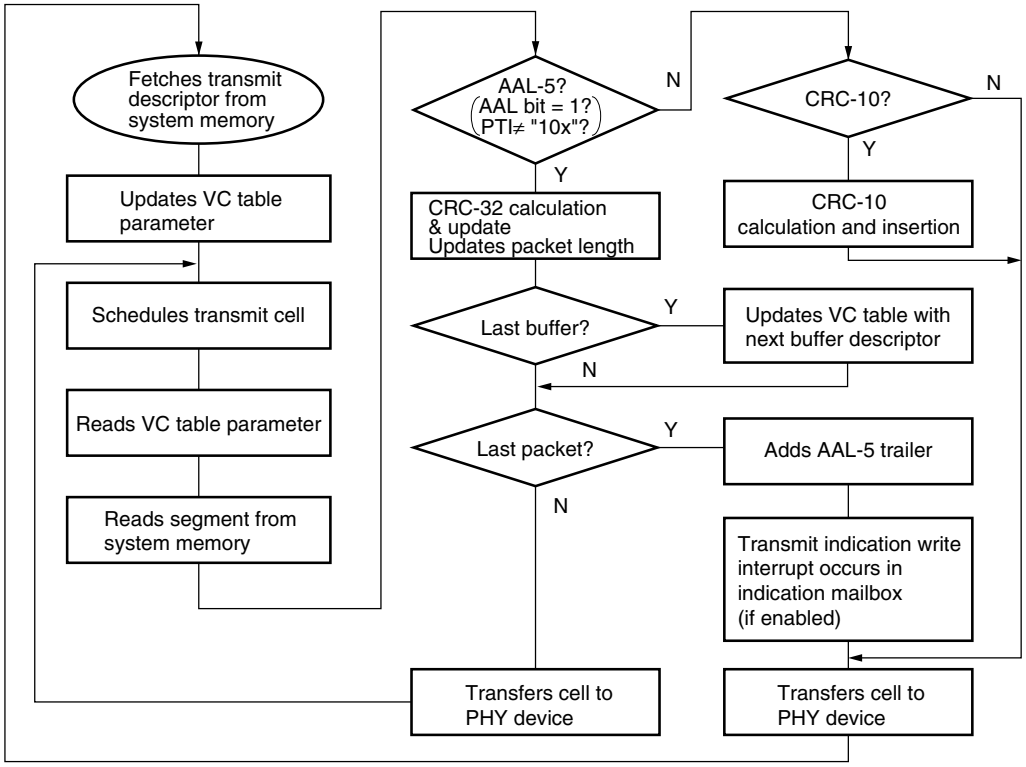
If the field of the VC table is less than 40 bytes when the L flag of the current transmit buffer indicates the last buffer, the current cell is the last cell of an AAL-PDU.

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 μPD98401A 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 5-28. Outline of Transmission Operation



5.4.6 Support of non-AAL-5 traffic

To support non-AAL-5 traffic, the μPD98401A has a function to transmit raw cells without appending an AAL-5 trailer. The μPD98401A executes non-AAL-5 transmission in two cases: (1) to transmit an OAM F5 cell, and (2) to transmit raw cell/RM cell.

(1) Transmitting OAM F5 cell

When the host assigns the pattern of an OAM F5 cell (100, 101) to the PTI field of the packet descriptor, the μPD98401A does not append an AAL-5 trailer. In this case, the μPD98401A 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 set the "SIZE" field using less than 48 bytes. If the transmit data is less than 48 bytes, extend the data length to 48 bytes by appending dummy data (such as "all 0"). For OAM F5 cell transmission, the host sets one packet descriptor per OAM F5 cell.

CRC-10 operation can be inserted when transmitting an OAM F5 cell. If the "C10 bit" of the packet descriptor is set to 1, the μPD98401A executes a CRC-10 operation on the 46 bytes and 6 bits of the 48 bytes of the segment of the OAM F5 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 cell, the following points must be noted.

- The "AAL" bit must be always cleared to 0 when transmitting an OAM F5 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 μPD98401A 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 μPD98401A 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 μPD98401A. When this function has been enabled, the μPD98401A 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 μPD98401A 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 μPD98401A 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 μPD98401A 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 μPD98401A 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 cell, RM cell, and raw cell transmit packets).

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

The μPD98401A 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 μPD98401A 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="XXX" 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 cell	PTI="10X" PTI="110X" 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 μPD98401A receives a cell in several steps.

Figure 5-29. 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)	Issues Open_Channel command
(5)	Sets receive VC table
(6)	Sets look-up table entry in control memory

Operation of μPD98401A (AAL-5 packet reception)	
(7)	Receives first cell of packet
(8)	Assigns batch of pool
(9)	Links VC to T1 timer list (monitors T1 error)
(10)	Transfers data to system memory in segment units
(11)	Assigns new batch as necessary
(12)	Receives last cell of packet
(13)	Checks error from AAL-5 trailer
(14)	Issues receive indication and interrupt

(15)	Reads receive indication
(16)	Updates mailbox read pointer (MTA)
(17)	Reads data from receive pool
(18)	Manages buffer of receive pool (Add_Batches command)

Operation of μPD98401A (raw cell reception)	
(7)*	Receives cell
(8)*	CRC-10 check
(9)*	Transfers data to system memory as raw cell data
(10)*	Sets RCR bit of GSR register and issues interrupt

The μPD98401A supports 32K VCs (receive VCs and transmit VCs can be used in any combination). The number of receive VCs actually supported is determined by bit manipulation of the VPI/VCI field.

The μPD98401A supports 16 bits of the receive VPI/VCI 24 bits. The user determines the method to convert the received VPI/VCI 24 bits into a logic code of 16 bits. This is then assigned to the VRR register (1). Before reception, a receive pool to store receive data must be prepared in the system memory (2). The information in the system memory such as the receive pool size and address is set in the free buffer pointer area of the control memory as a pool descriptor (3).

The host issues the Open_Channel command to each connection to be received and opens a channel (VC). The μPD98401A 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 (4). The host initializes the parameters in the VC table (5). Next, to actually start reception, the host sets the address of the VC table to the look-up table and enables the look-up table (6).

On receiving the first cell of the packet from the PHY device, the μPD98401A checks whether the cell is to be received or dropped, by looking at the look-up table (7). If the look-up table is enabled, the μPD98401A accesses the VC table, and assigns the batch of the pool to the VC (8). It also adds the VC to the list of the T1 timer and monitors time-out (9). Each time the μPD98401A has received a cell, it repeatedly transfers the cell to the system

memory in segment units (10). If the buffer becomes full in the middle, the μPD98401A fetches and assigns a new batch (11). When the last cell of the packet is received (12), the μPD98401A checks for errors from the trailer information contained in the cell (13), issues receive indication to a specified mailbox, and issues an interrupt to the host if not masked (14). The host reads the receive indication (15), changes the read pointer for the mailbox (16), and receives the receive data from the pool (17).

If the area for the number of remaining batches in the pool of the system memory has run short during reception, the host issues the Add_Batch command as necessary (18).

The μPD98401A has a raw cell reception function to transfer received cells to the system memory as is in cell units when it receives traffic other than that of AAL-5 type. The μPD98401A receives a raw cell (7)*, performs CRC-10 check (8)*, and stores the raw cell data to the free buffer of the specified pool (9)*. When receiving raw cells, the processing is completed in cell units, and the receive indication is not stored in the mailbox for AAL-5 packet reception. Each time a cell has been stored, the RCR bit of the GSR register is set, and an interrupt is issued (10)*.

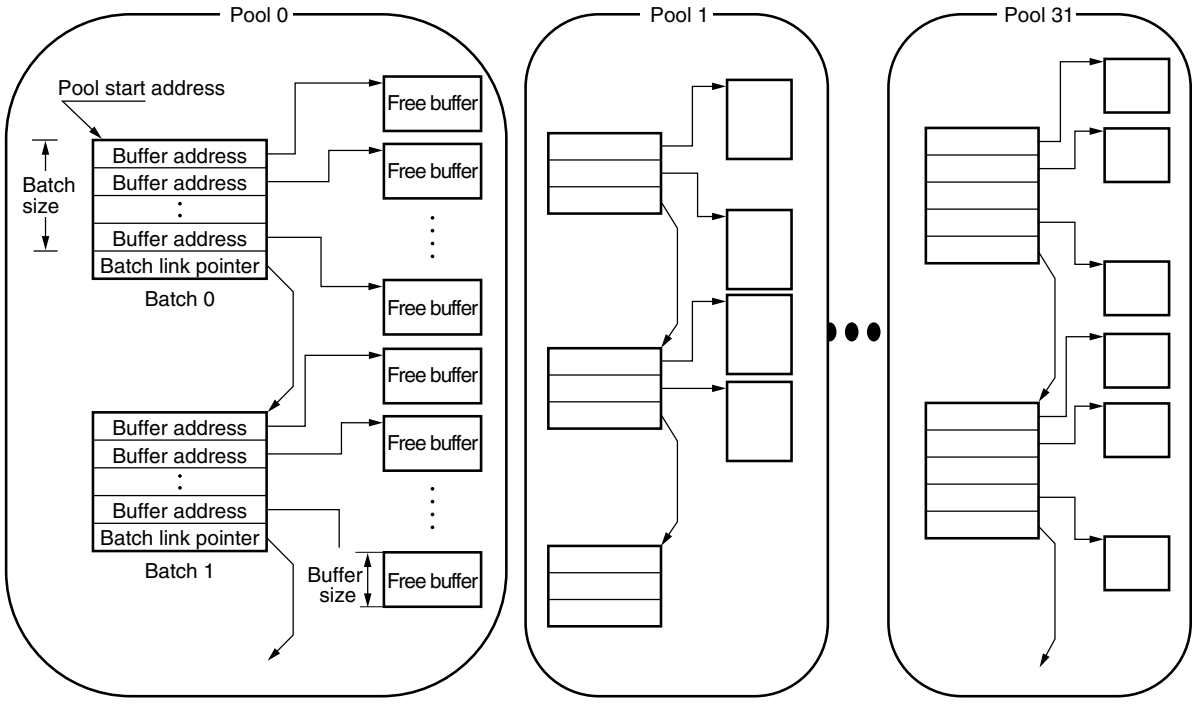
The reception processing flow shown above is for illustration only, and processing is not synchronized with the software processing the host must perform.

5.5.2 Structure of receive data

(1) Receive pool

The μPD98401A stores receive data 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 μPD98401A 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 5-30. 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. If the function to recognize the low-order 2 bits of the DMA address and the SIZE [2:0] signals of the bus interface is not provided, the start address of the free buffer must be always located from a 32-bit boundary, and its low-order 2 bits must be “00”. If byte alignment transfer is supported, the start address can be located at any byte boundary. If the low-order 2 bits of the address are not set to “00”, the μPD98401A executes a byte alignment DMA write. The byte alignment DMA write operation, refer to **(4) Byte alignment transfer** in **4.1.3 Master (DMA) operation** of bus interface.

Link pointer

<1> AAL-5 type reception pool

NEXT BATCH ADDRESS1 : 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”.

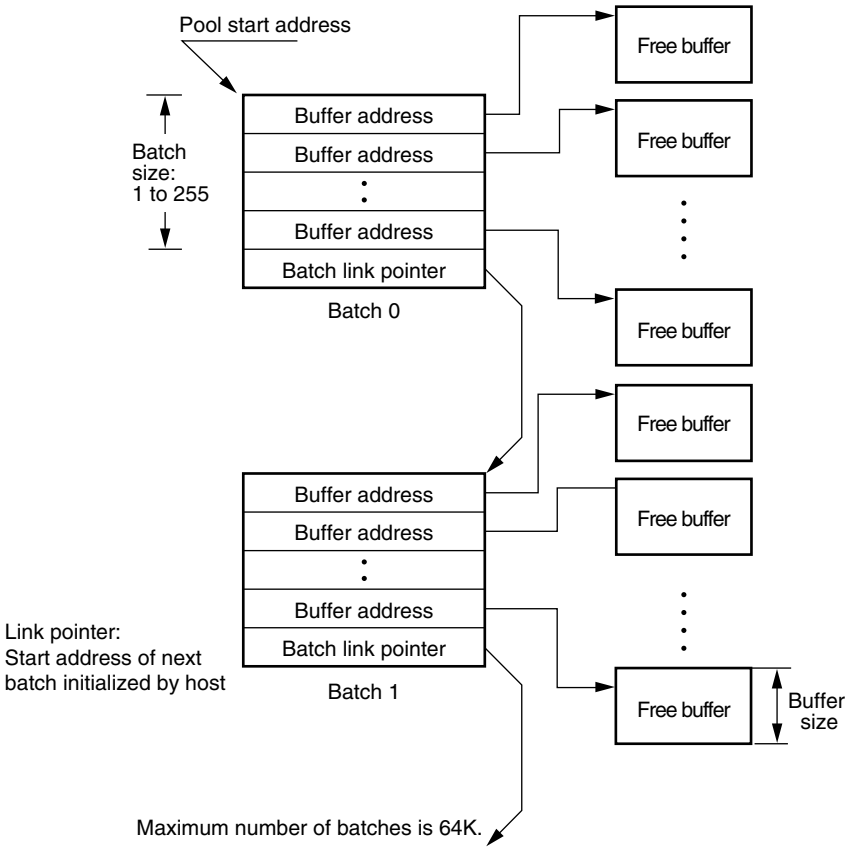
<2> Raw cell reception pool

NEXT BATCH ADDRESS2 : 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 μPD98401A 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 μPD98401A to store the actual received data. The size of one buffer can be set from 64 bytes form 64K bytes. The size of all the free buffers in the same pool must be the same, however.

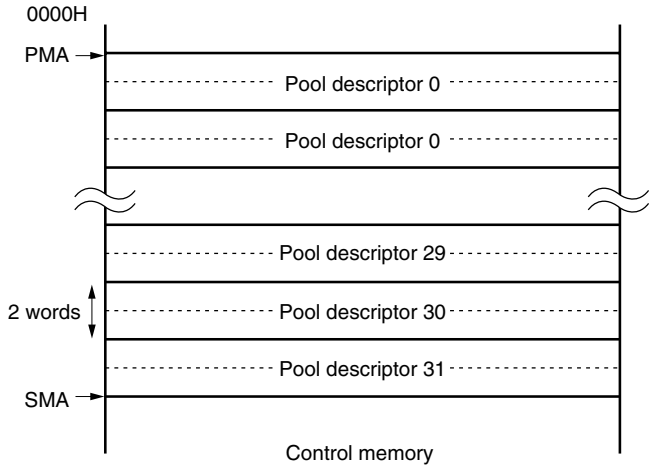
Figure 5-31. Configuration of Pool



(2) Receive free buffer pool pointer

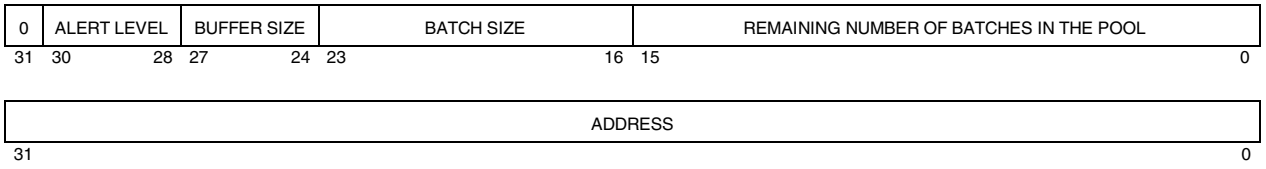
The maximum number of pools prepared by the host in the system memory is 32. These pools are numbered from 0 to 31. The information on each pool such as address and size is set in a “receive free buffer pool pointer” area by a pool descriptor consisting of 2 words. The “receive free buffer pool pointer” area starts with a PMA address of the control memory, and the μPD98401A sequentially recognizes pool 0, pool 1, pool 2, and so on, from the PMA address. For example, when the μPD98401A accesses the descriptor of pool 3, it accesses the address resulting from adding 6 or 7 to the PMA address. The receive free buffer pool pointer area consists of 64 words when all the 32 pools are set. When less than 32 pools are used, the area between the PMA and SMA addresses can be reduced depending on the number of pools used.

Figure 5-32. Location of Pool Descriptor in Receive Free Buffer Pool Pointer



The format of the pool descriptor is as follows.

Figure 5-33. 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 μPD98401A 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 μPD98401A 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 x 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 μPD98401A 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 μPD98401A, and indicates the number of batches remaining in the pool. Up to 65535 batches can be set.

The μPD98401A 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_Batch 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 μPD98401A as a pointer that indicates the next batch.

(3) Pool storing raw cell

The pool numbers that can be stored by the μPD98401A differ depending on the type of the data received. The user can specify any of pools 0 through 31 for each VC when data of ALL-5 type is received. When raw cell data is received, any of pool 0 to 7 can be specified. If an OAM F5 cell is to be received, however, the μPD98401A unconditionally stores the cell to pool 0. If the user makes setting to receive an OAM F5 cell, therefore, pool 0 must always be specified to store the raw cell data of the OAM cell.

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

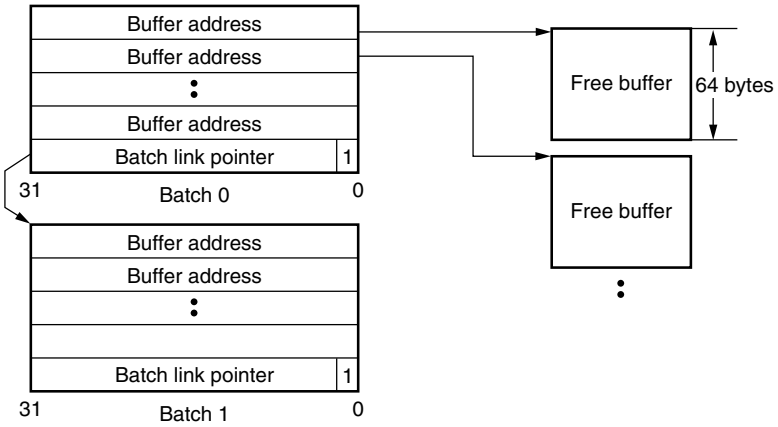
Data Type	Usable Pool
AAL-5 type packet reception	Pool 0 to 31 can be used.
Raw cell data	Pool 0 to 7
OAM F5 cell/raw cell data of RM cell	Pool 0 only

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 μPD98401A monitors the least significant bit of the buffer address read from the batch. If the least significant bit is "1", the μPD98401A 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 μPD98401A ignores the setting of the "ALERT LEVEL" field. The RQU interrupt, which indicates that the number of remaining batches is 0, functions.

The μPD98401A 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 μPD98401A ignores the area.

Figure 5-34. Raw Cell Pool



5.5.3 Receive channel (receive VC)

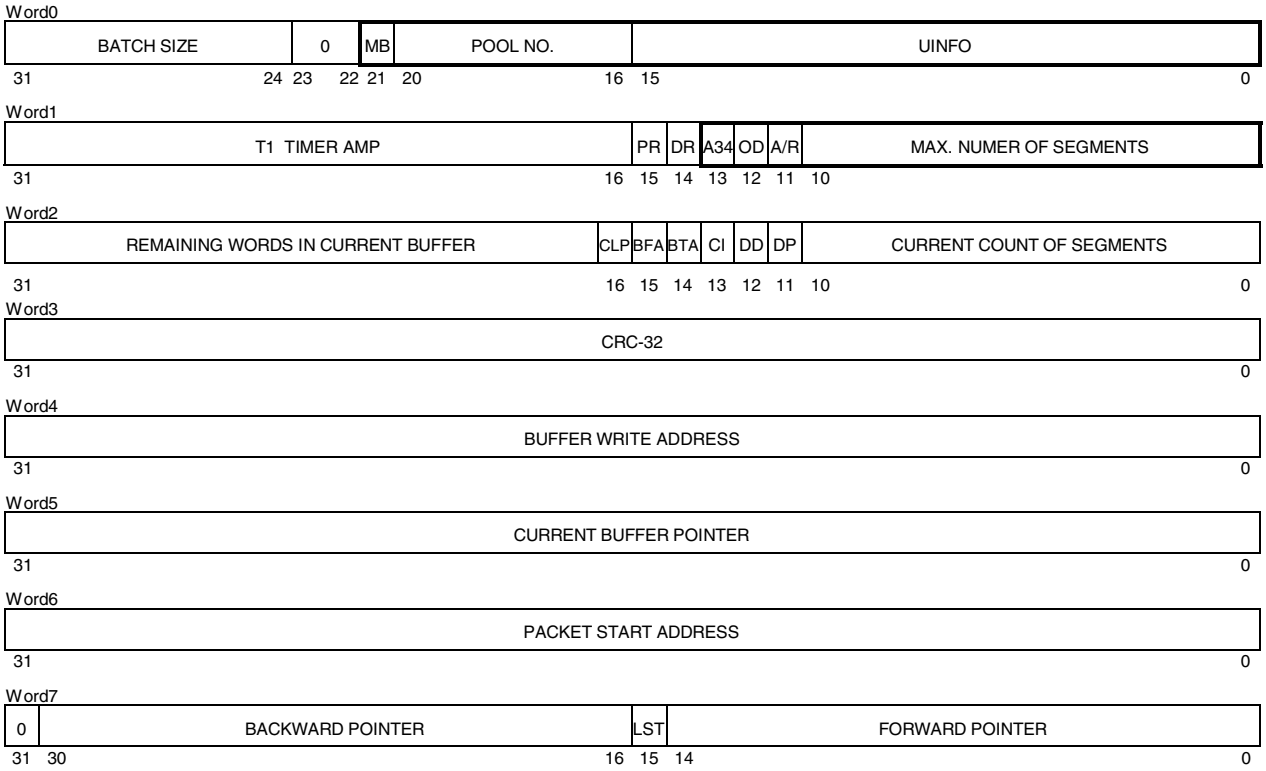
(1) Opening receive channel

The host opens a receive channel by issuing the Open_Channel command. When receiving this command, the μPD98401A 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 5-35 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 5-35. Receive VC Table



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.

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

1 - Mailbox 1

0 - Mailbox 0

POOL NO. ---------- Pool number. This field selects one of the 32 pools for this VC.

UINFO---------- User information. The user can assign any pattern to this field. The pattern assigned to this field is returned by the μPD98401A with receive indication.

T1 TIMESTAMP ---------- This area is used by the μPD98401A 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.

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.

BTA ---------- This bit is set to “1” if there is a batch allocated to this VC.

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.

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 μPD98401A 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.

CURRENT BUFFER POINTER ---------- First address of the free buffer to be allocated next.

PACKET START ADDRESS ---------- Start address of the packet. This is the first address of the batch allocated first.
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

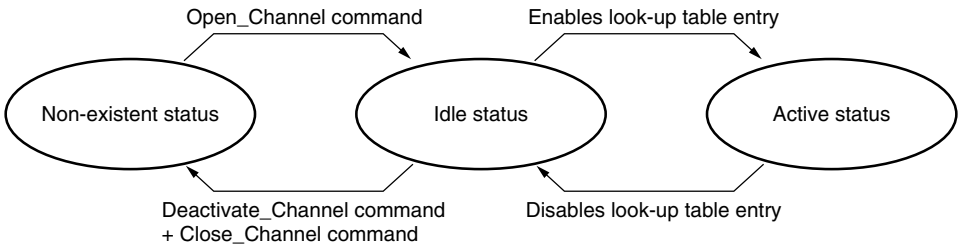
The receive channel may be in three statuses: non-existent, idle, and active. The host opens a channel by issuing the Open_Channel command. On receiving this command, the μPD98401A reports the address of the VC table from the free block pool of the control memory to the host as a command indication. Next, the host writes appropriate parameters (such as pool number used, AAL-5 processing/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, "VC NUMBER" and enable bit must be entered to the look-up table. 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 μPD98401A 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 disables the look-up table 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 disabling the look-up table. 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 starts from address 0000H 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 μPD98401A internally converts the 24 bits of VPI/VCI included in a receive cell into a 16-bit logic code. This conversion is made according to the setting of the "SHIFT" and "MASK" fields of the VRR register made by the user. 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 9 bits or more, 0 is added to the high-order bits.

<2> The 16 bits created in<1>and the contents of the "MASK" field of the VRR register are ANDed.

<3> The high-order 15 bits of the 16 bits resulting from ANDing is a part of the address of the receive look-up table.

<4> The least significant bit ("L" in Figure 5-37) indicates the high-order 16 bits or low-order 16 bits of the word indicated by the address. If the least significant bit is "0", it indicates the high-order 16 bits; if it is "1", the low-order 16 bits are indicated.

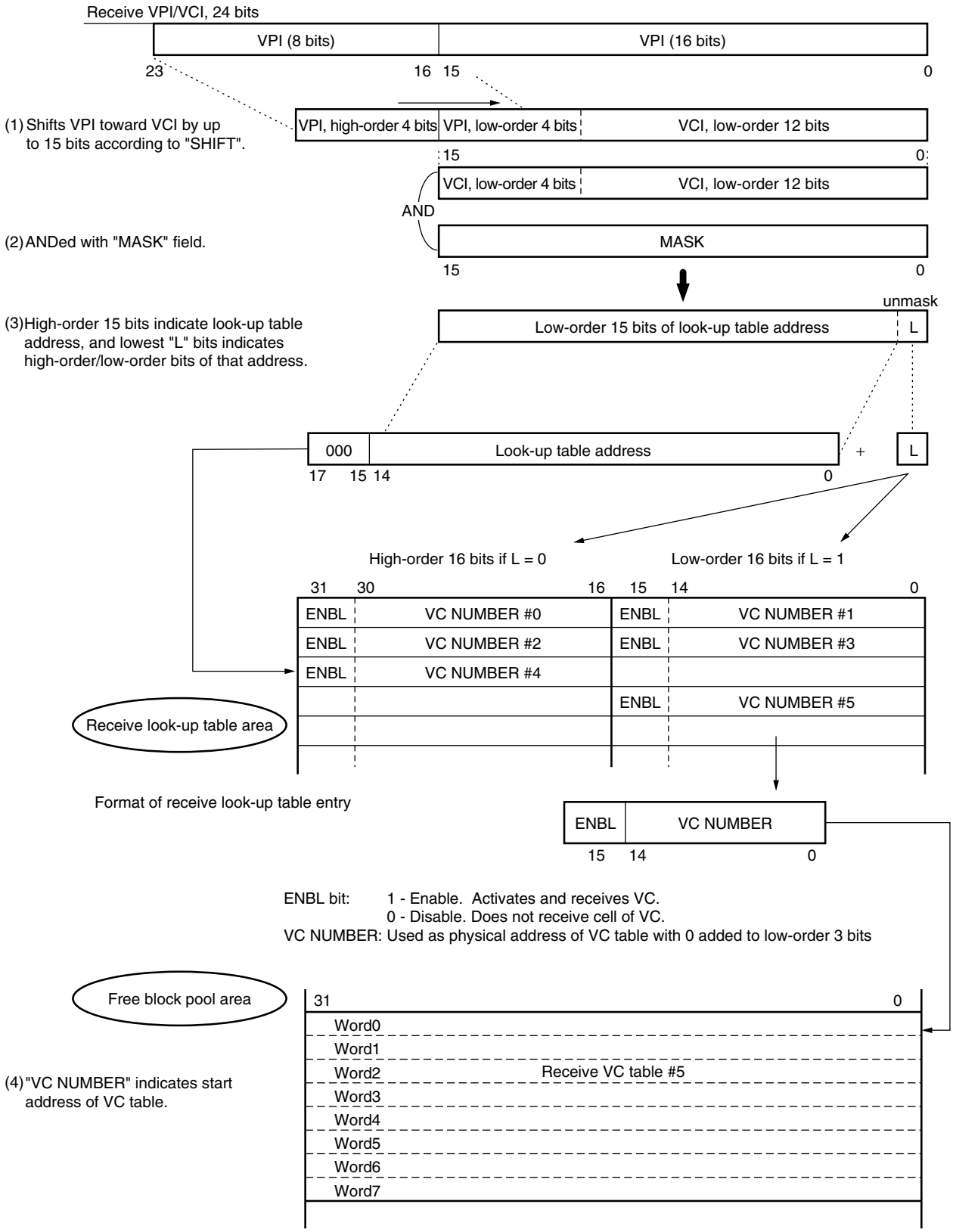
<5> The host stores a 16-bit code combining an enable bit ("ENBL" bit") and the "VC NUMBER" of the VC to the look-up table address created in this way.
When the enable bit is set to "1", the VC enters the active status.

When the μPD98401A receives a cell from the PHY device, it generates an address for the receive look-up table in the same manner as above, according to the VPI/VCI in the header of the receive cell and the setting of the VRR register. The μPD98401A accesses the look-up table entry by using this address, and receives a cell if the enable bit is "1"; otherwise, it drops the cell. When receiving a cell, the address of the corresponding VC table is created from "VC NUMBER" stored to the look-up table, and setting necessary for processing is checked or updated.

Caution Note that "VC NUMBER" is not a physical address indicating the beginning of the VC table in the free block pool, but is a code resulting from shifting the low-order 3 bits of the address.

The size of the control memory occupied by the look-up table is determined by the number of patterns of VPI/VCI. A capacity sufficient for storing the number of look-up table entries specified by VPI/VCI to be received is necessary.

Figure 5-37. Receive Look-up Table (when SHIF = 4)

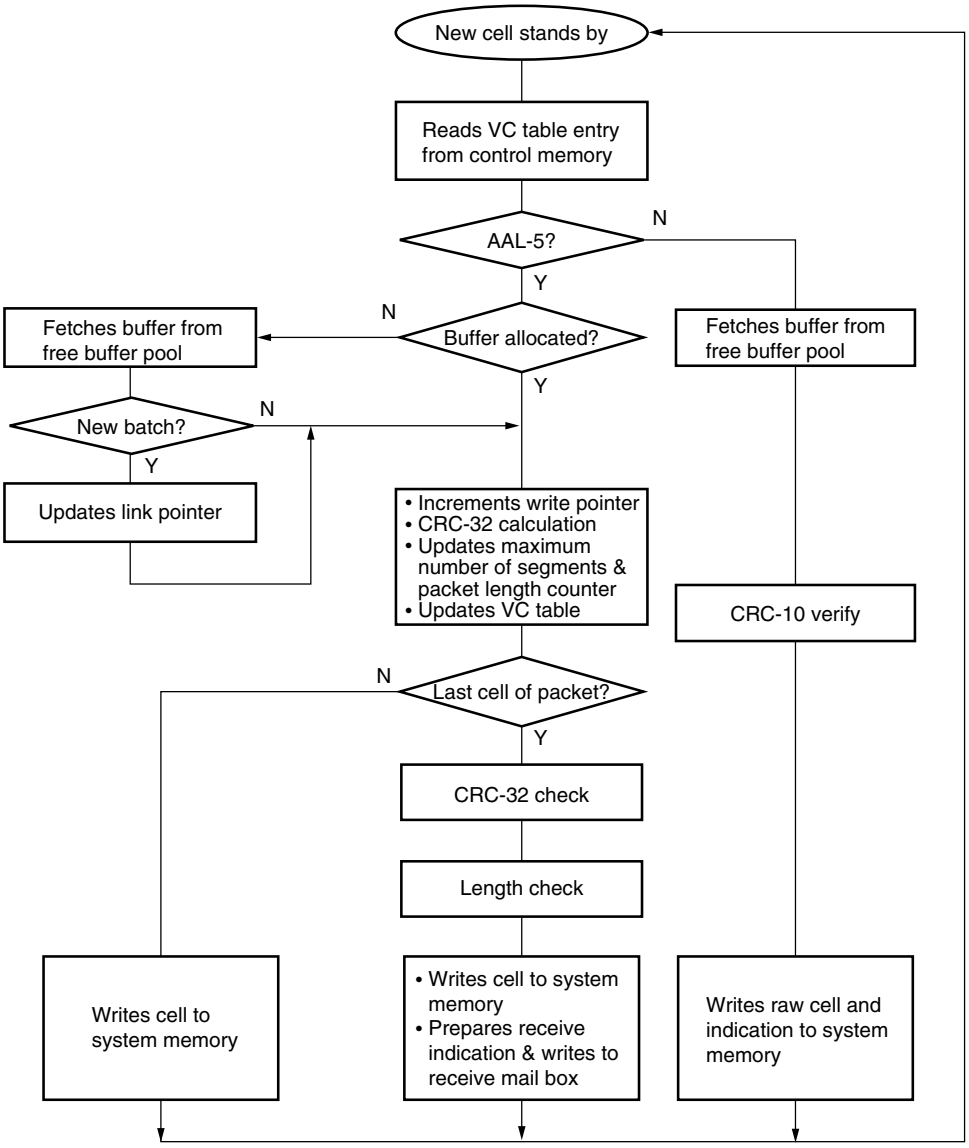


5.5.5 Reception operation

(1) Reception operation

- The μPD98401A 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 μPD98401A 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 μPD98401A reads the VC table of the free block pool from "VC NUMBER" stored in the look-up table, and obtains the pool number of the system memory in which the cell is to be stored, and its address information.
- 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 μPD98401A 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.
- μPD98401A 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 μPD98401A 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 μPD98401A 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 μPD98401A 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 μPD98401A overwrites the link pointer of the batch used, to change the chain.
- If the μPD98401A 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. Reception Operation Flowchart



(2) Storing receive data

This section explains the procedure in which the μPD98401A stores the receive data in the system memory, and the procedure of the host operation. The μPD98401A uses the pool of a packet and the pool for raw cell data storage in different procedures.

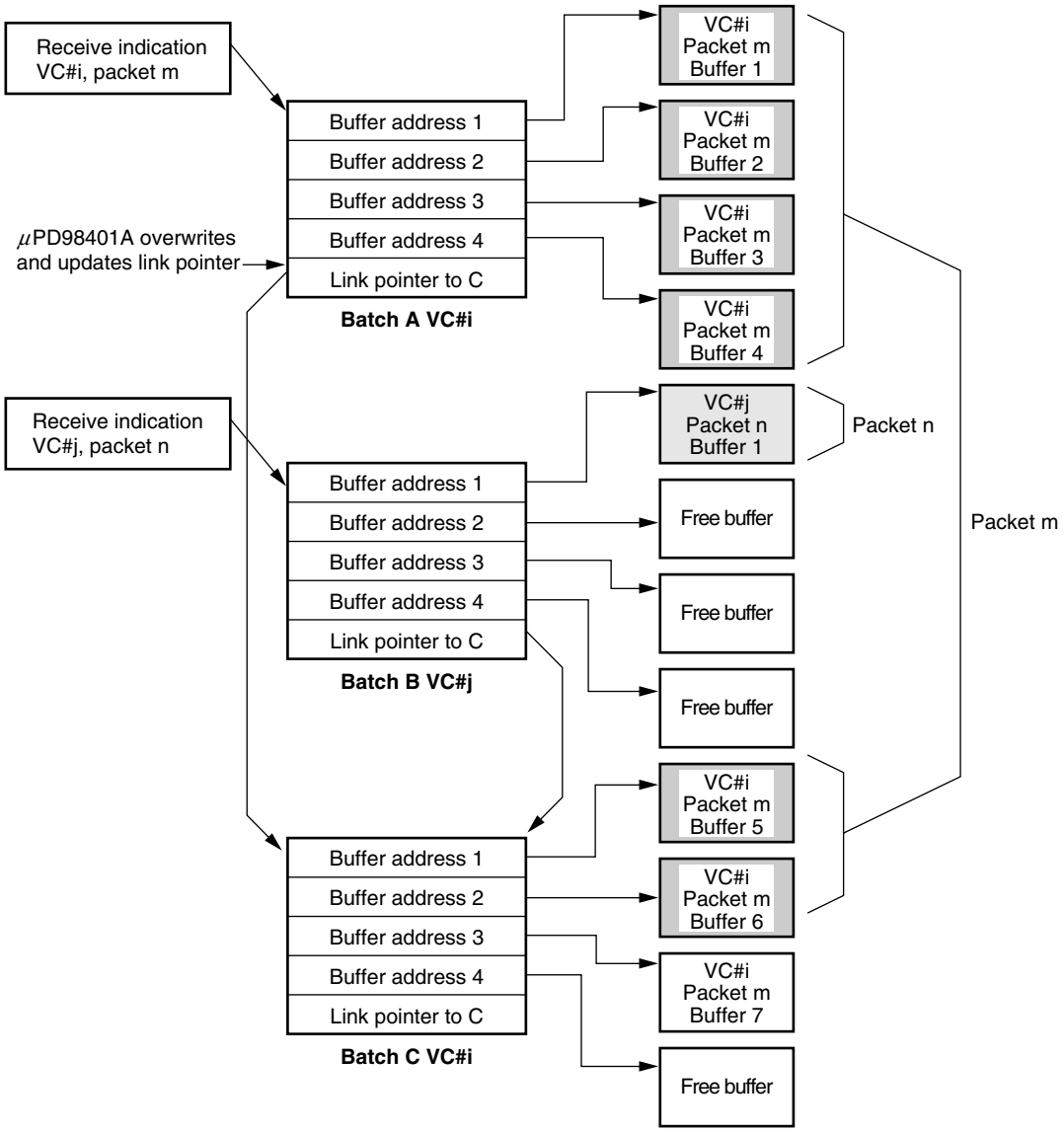
(a) Pool for AAL-5 packet

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 μPD98401A 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 5-39 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 5-39. Configuration Example of Receive Data



- The first cell of packet m of VC#i is received. The μPD98401A 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 μPD98401A 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 μPD98401A 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 μPD98401A 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 μPD98401A 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 μPD98401A. The μPD98401A decrements the "REMAINING NO. 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 μPD98401A 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 NO. 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 μPD98401A 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 μPD98401A 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.

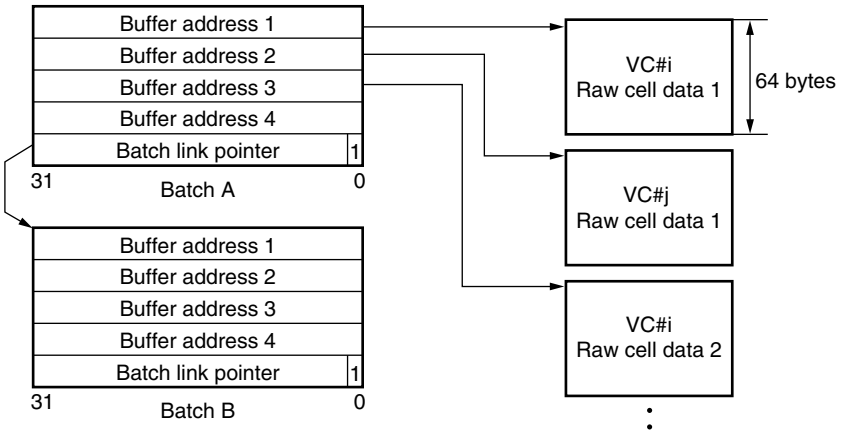
(b) Pool storing raw cell data

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 μPD98401A has received cells, it sequentially stores raw cell data, starting from the first free buffer of the first batch of a pool. The μPD98401A does not issue a receive indication when it has received raw cell data. Instead, the μPD98401A 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 μPD98401A 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 μPD98401A 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 μPD98401A recognizes this address as a link pointer, and proceeds to the next batch.

Figure 5-40. Structure of Raw Cell Data Storage Pool
(if raw cell data of VC#i 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 μPD98401A 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 μPD98401A monitors whether the packet being received exceeds this time. If the time is exceeded, the μPD98401A stops reception of the packet, and reports this to the host as a T1 error by using a receive indication.

The μPD98401A 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 μPD98401A 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, and the low-order 16 bits are all 0. The unit is the cycle of the system clock.

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 μPD98401A operates.

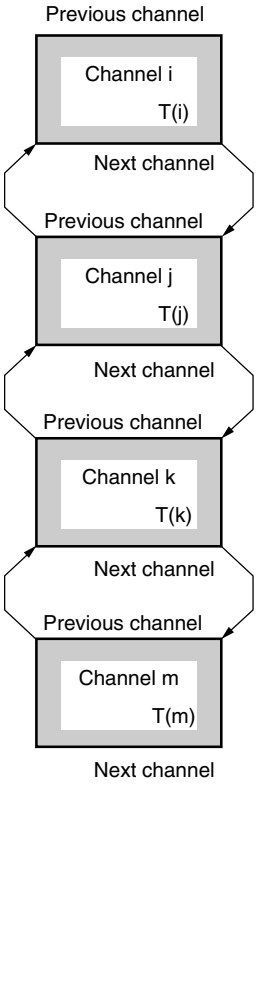
The μPD98401A monitors the permissible time by a link list method which is formed by using the “FORWARD POINTER” and “BACKWARD POINTER” at the seventh word in the VC table. These pointers store the “VC NUMBER” of a VC. Each time the first cell of a new packet has arrived, the μPD98401A writes the current 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 μPD98401A 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 μPD98401A 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 5-41. Reassembly Timer (T1)

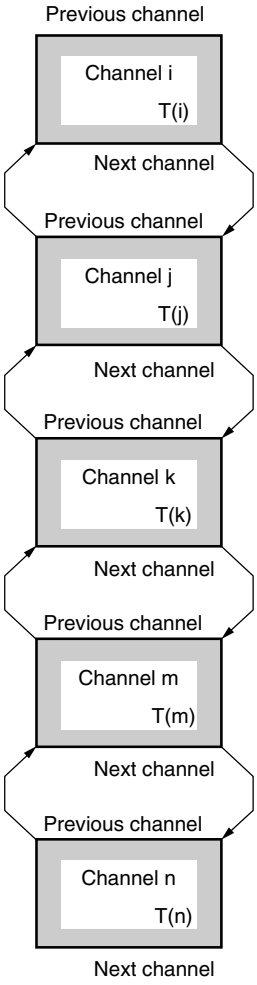
(a) Current status of T1 list



(b) Starts PDU with channel n

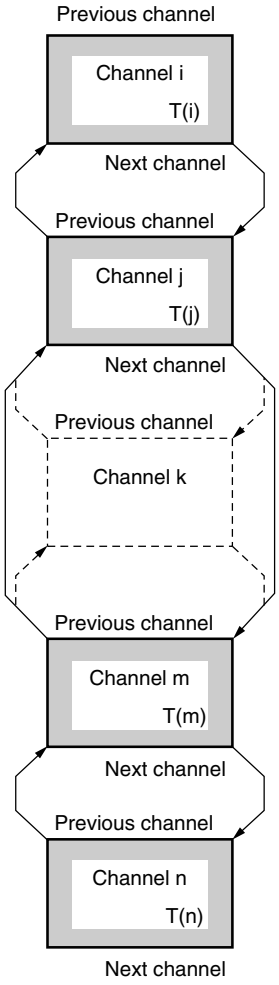
Adds channel n to list

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 μPD98401A 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 μPD98401A 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.

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 underflow

When the μPD98401A 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 μPD98401A tries to transfer the segment of the cell to the system memory, the μPD98401A drops the cell. If the dropped cell is an intermediate or last cell of the packet, the μPD98401A 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 μPD98401A 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 μPD98401A 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 μPD98401A 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 μPD98401A, 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 μPD98401A 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 the μPD98401A receives a packet of AAL-5 traffic, it issues receive indication to the host to report that it has received the packet. As the receive indication, 4-word information is generated for each packet, and is stored in a mailbox. Whether the receive indication is stored in mailbox 0 or 1 is selected by the host by using the "MB" bit of the VC table set for each VC. When the receive indication has been stored, the μPD98401A sets the corresponding MM bit of the GSR register to 1 to issue an interrupt, if not masked. The receive indication includes the start address and size of the batch used by the μPD98401A to store the packet. The host can perform processing of the packet data that has arrived, by reading the receive indication.

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 53-byte 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 μPD98401A 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 μPD98401A 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

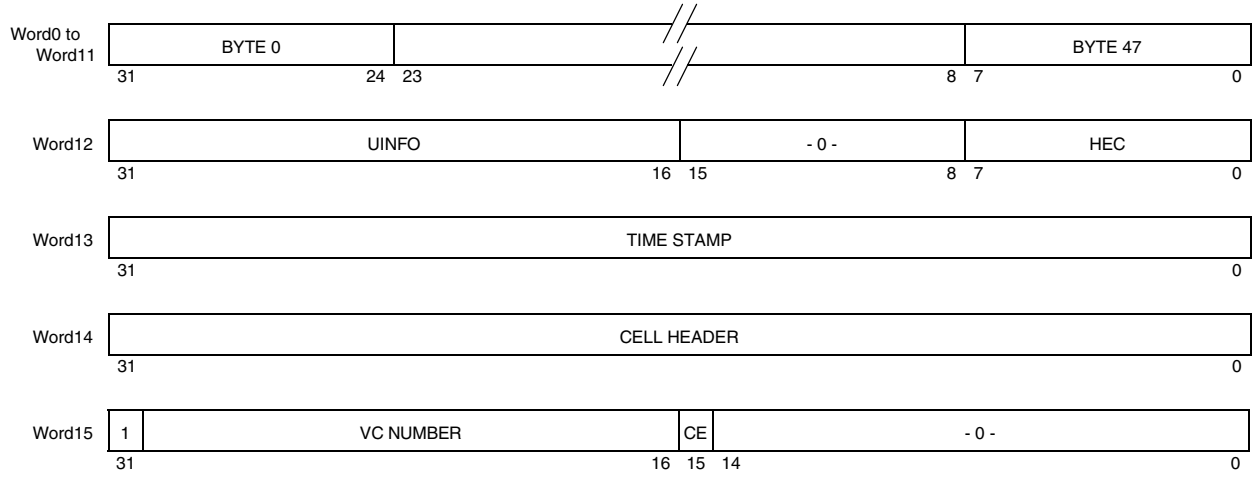
If the A/R bit of the VC table opened by the user is cleared to "0", the μPD98401A transfers a cell of the VC to the system memory as raw cell data, each time it has received such a cell. When a cell is received as raw cell data, the user can specify each of pool 0 to 7 as "POOL NO.". The specified pool 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 μPD98401A 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 5-42. 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.

The above format is in big endian. In little endian, the byte location of only the data of Word0 through Word11 is changed, and the indication of Word12 through Word15 is not changed.

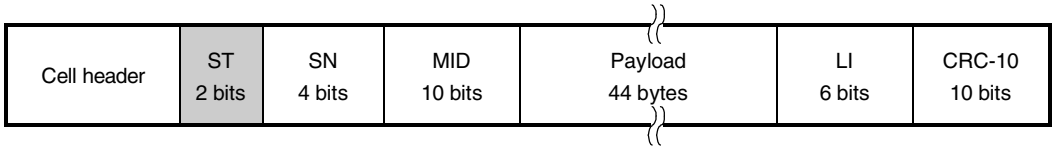
(3) AAL-3/4 packet raw cell reception assist function

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 μPD98401A 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 μPD98401A 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

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 μPD98401A 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.

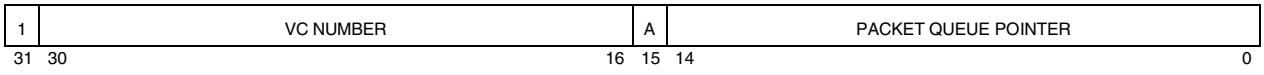
The MSB B of the last word of the transmit/receive indication stored by the μPD98401A is always set to "1" By resetting this bit of an indication that has been read to "0", it can be used as a flag indicating which indications in the mailbox have been processed and which have not been processed. If this 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 format of the transmit indication is as follows (same in both big endian and little endian).



VC NUMBER ---------- "VC NUMBER" used by this VC.

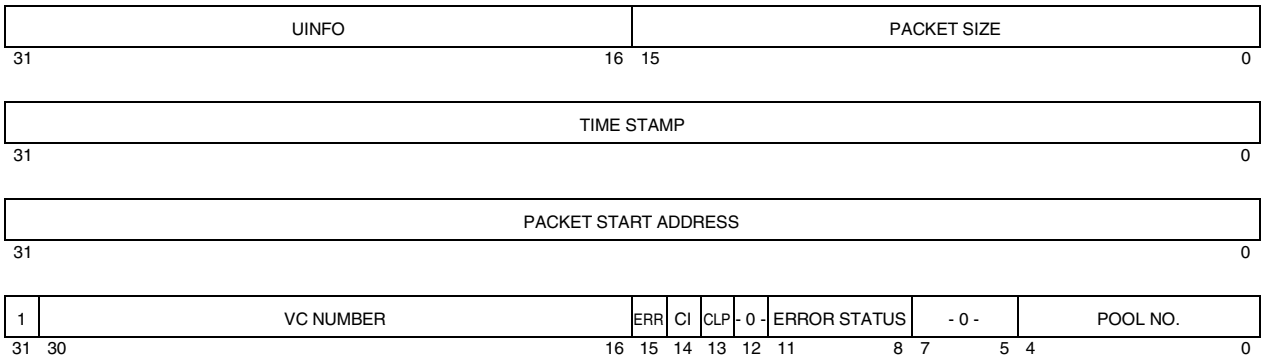
A (Active) ---------- '0' - The VC has entered the idle status because the next packet descriptor is a blank packet descriptor.

'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 issued for a packet of AAL-5 traffic. It is not issued when a raw cell of non-AAL-5 traffic is processed. Its format is as follows (same in big endian and little endian).



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.

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

5.7 Interrupt Function

The μPD98401A 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.

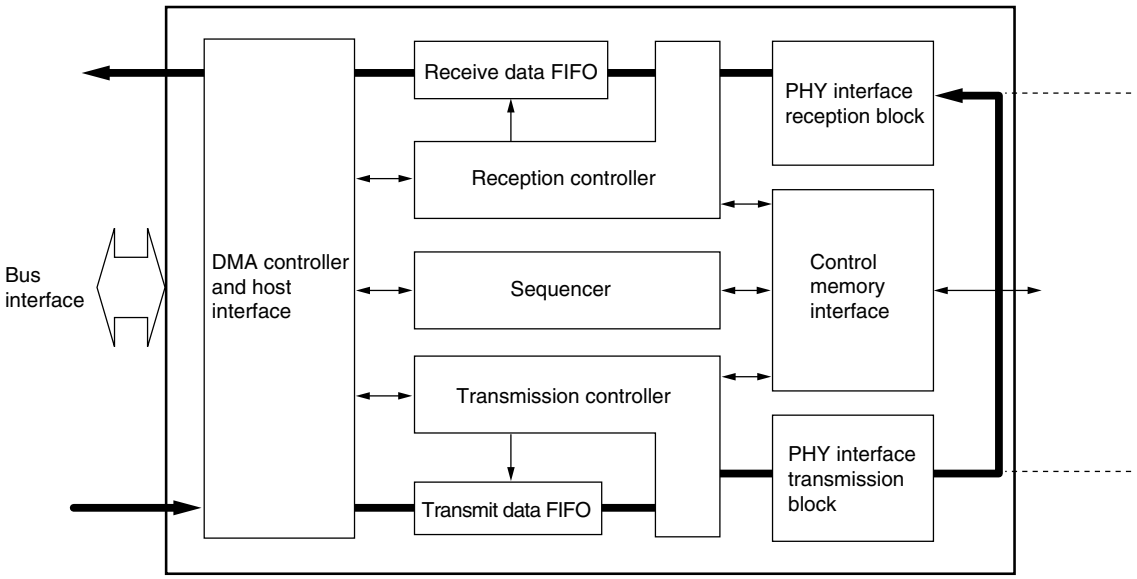
Of the 32 bits of the GSR register, the PI bit for PHY interrupt is used to input an interrupt from an external device such as a PHY device. When it is detected that the PHY interrupt has become active, the PI bit is set, and an interrupt is generated to the host. In response, the host issues the Indirect_Access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

For the details on the GMR register and IMR register, refer to **7.2 (1) GMR** and **(3) IMR**.

5.8 Loopback Function

The μPD98401A 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 μPD98401A 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.

Figure 5-43. Loopback Mode



5.9 Global Shutdown Function

Global shutdown is executed by setting the MSB bit of the VRR register. When the μPD98401A receives this command, it stops processing all the receive VCs under way. After that, the μPD98401A does not receive cells, stopping its reception function completely. The transmission function is not affected and continues operating.

When the μPD98401A 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 μPD98401A cannot be enabled again once this command has been issued to the μPD98401A. To activate the μPD98401A again, execute software reset or hardware reset to the μPD98401A.

CHAPTER 6 COMMANDS

The host issues the seven types of commands listed in Table 6-1 to the μPD98401A. This chapter first explains the procedure to issue a command to the μPD98401A, and then the details of each command.

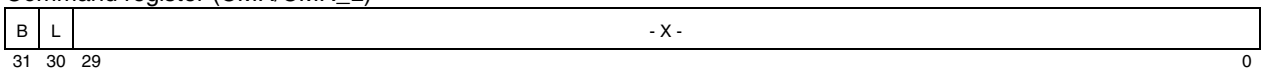
Table 6-1. Command Type

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)

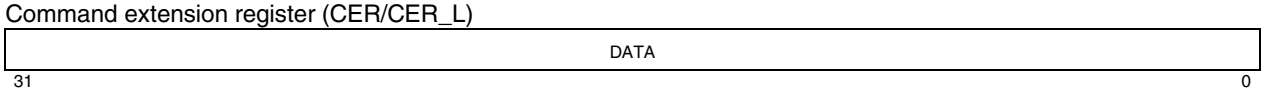
All commands are input to the μPD98401A via the command register and command extension register. Each register has two addresses called CMR and CMR_L, and CER and CER_L, respectively. The addresses of each register are as shown in the table below. The command register has a “busy flag” and a “lock flag” that serve as command status flags.

Command register (CMR/CMR_L)



B bit : Busy flag
L bit : Lock flag
X : Command code, parameter

Command extension register (CER/CER_L)



Name	Register Name	Address	
		Word address mode	Byte address mode
Command register	CMR	08H	20H
	CMR_L	09H	24H
Command extension register	CER	0AH	28H
	CER_L	0BH	2CH

The μPD98401A uses different registers depending on whether it is used in a single-host system where it is controlled by one host CPU or in a multi-host system where it is controlled by multiple CPUs.

<1> In single-host system

When a command is issued to the μPD98401A, only CMR and CER are used as the command register and command extension register. CMR_L and CER_L are not used.

<2> In multi-host system

When a command is issued to the μPD98401A, CMR, CMR_L, CER, and CER_L are used as the command register and command extension register.

6.2 Busy Flag

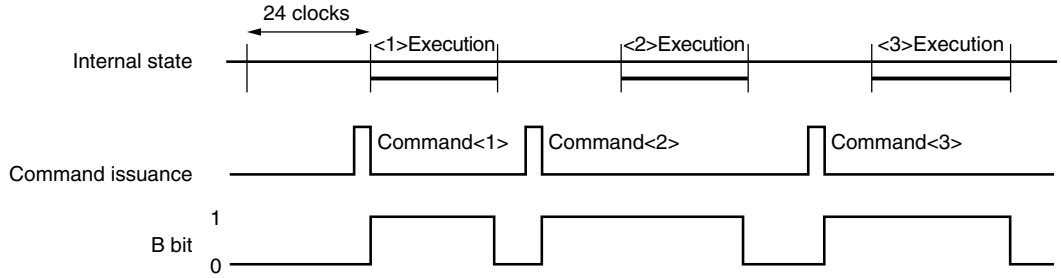
The host writes a command with various parameters to the command register.

Internally, the μPD98401A operates in units called states which are equivalent to the duration of 24 system clocks. The μPD98401A receives one command from the host in one state, and executes and completes it in the next state. Because the μPD98401A 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 μPD98401A 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 μPD98401A ignores and invalidates this command. The host can issue a new command only while the B bit is "0".

The μPD98401A 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 μPD98401A 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 μPD98401A 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 μPD98401A 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 μPD98401A 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 24 clocks.

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

6.5 Support of Multi-Host System (lock flag)

In a multi-host system where the μPD98401A is controlled by multilpe host CPUs, bit 30 (L bit) of the command register is used. The L bit functions as a lock flag to prevent the other hosts from writing another command to the μPD98401A before execution of the command issued by one host has been completed.

For example, suppose the μPD98401A is controlled by two host CPUs. Also suppose the two host CPUs try to write a command to the CMR register. The two hosts first read CMR, and confirm that the B bit is not "1" (not busy). Next each host writes a command to the CMR register. Then one host overwrites the command written by the other.

To prevent this, the "L bit" located at the position of bit 30 of the CMR register functions as a lock flag that indicates that the other host is accessing the CMR register. While the L bit is "1", the μPD98401A is in the "lock status" in which one host is executing a command and therefore, the command from the other host cannot be accepted. Conversely, if the L bit is "0", it indicates that the other host is not executing a command.

When setting or resetting the lock flag "L bit", the two addresses each of the command register and command extension register are used. The μPD98401A is locked or unlocked when a host accesses each register, as shown in Table 6-2.

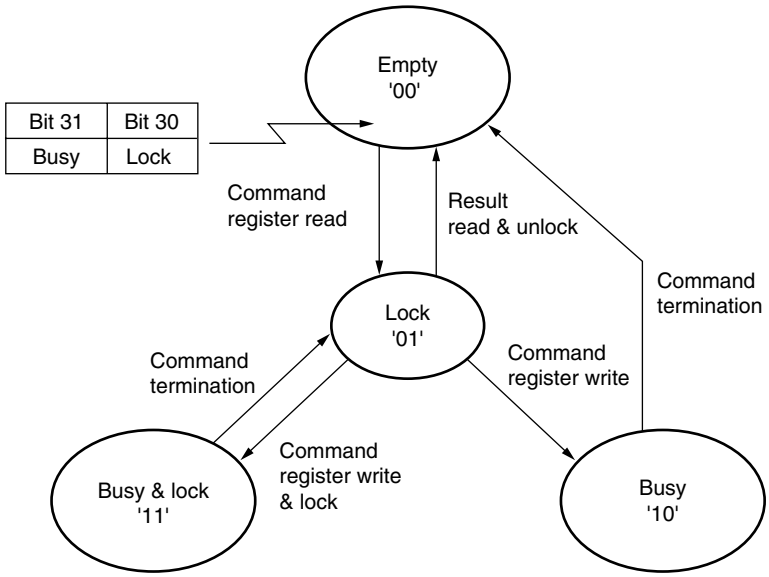
Table 6-2. Operation of L Bit When Register Is Accessed

Operation	L Bit Transition	Read	Write
Lock	0 → 1	CMR CMR_L CER CER_L	CMR_L
Lock	1 → 0	CMR_L CER_L	CMR

All the hosts must read the CMR register before they issue a command. They are permitted to issue a command only when the L bit is “0”. If the L bit is “1”, the host must wait until it is cleared to “0”. A host permitted to issue a command keeps the L bit to “1” and locks the μPD98401A, until execution of the command is completed. The L bit must be cleared to “0” to unlock the μPD98401A on the responsibility of the host that set the bit to “1”.

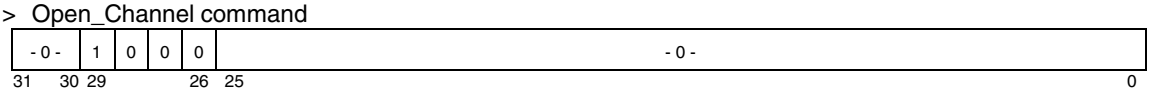
Depending on accessing, the next command can be executed with the L bit kept at “1”. The user can give a priority to execute a command to one host over the others by maintaining the lock status.

Figure 6-2. Status Transition of Command Register

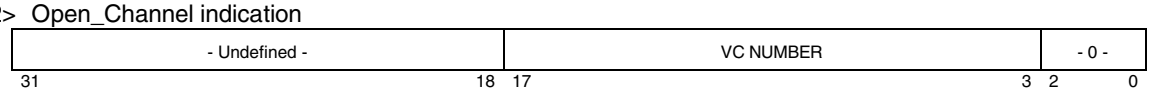


6.6 Commands and Command Indication

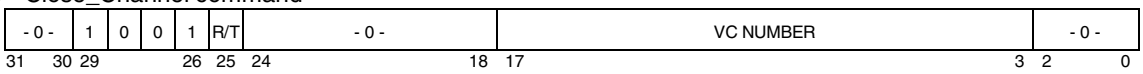
<1> Open_Channel command



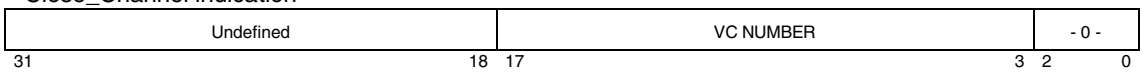
<2> Open_Channel indication



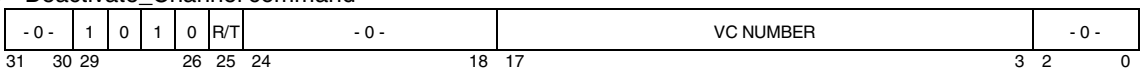
<3> Close_Channel command



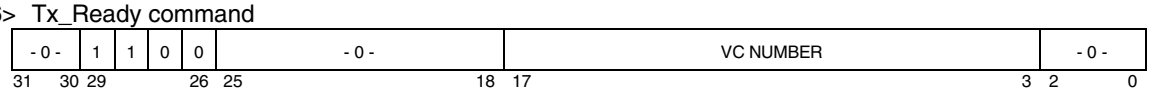
<4> Close_Channel indication



<5> Deactivate_Channel command

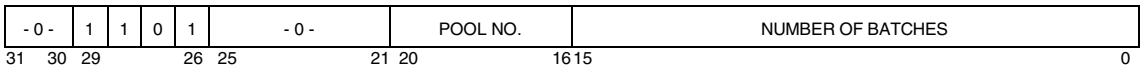


<6> Tx_Ready command

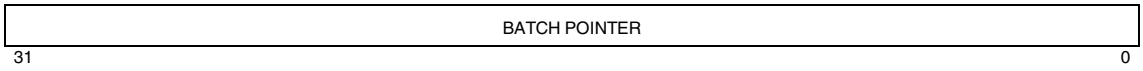


<7> Add_Batches command

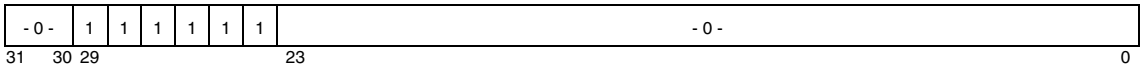
CMR:



CER:

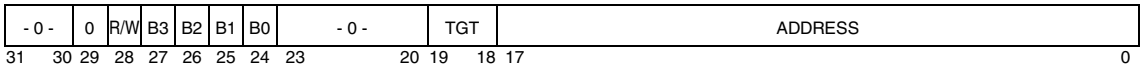


<8> NOP command

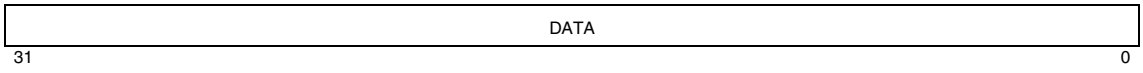


<9> Indirect_Access command

CMR:



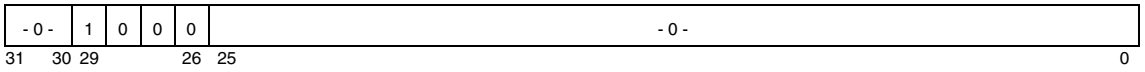
CER:



6.7 Command Details

<1> Open_Channel command

Programming of command register

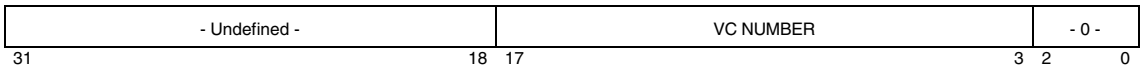


Command description

The Open_Channel command is used to open a new channel used for transmission or reception. When the μPD98401A 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 17): Bits 17 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.

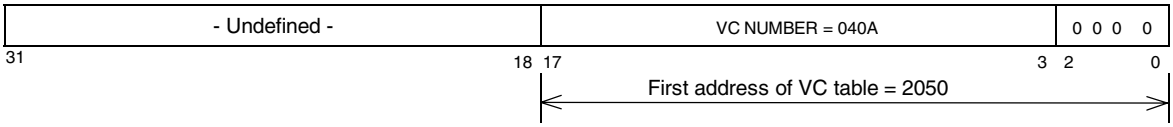
Command description

When the μPD98401A 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 μPD98401A 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.

Example: The address of the VC table in the control memory is as follows when the host has received the following command indication by issuing the Open_Channel command.

Command indication of Open_Channel command

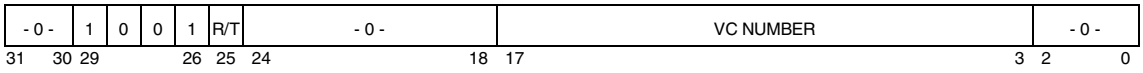


Address	31 0
2050	Word 0
2051	Word 1
2052	Word 2
2053	Word 3
2054	Word 4
2055	Word 5
2056	Word 6
2057	Word 7

The host uses the Indirect_Access command to access the control memory.
These addresses are set to the “ADDRESS” field of the Indirect_Access Command.

<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 17): “VC NUMBER” of the channel to be closed.

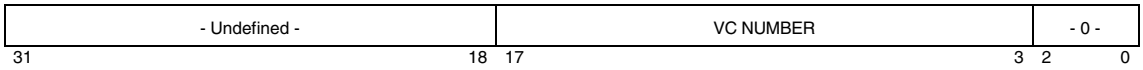
Command description

The Close_Channel command is used to close a receive channel or transmit channel. When the μPD98401A 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 17 through 3): Bits 17 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 μPD98401A rejected the issued Close_Channel command. The host repeatedly issues the Close_Channel command until it obtains a VC NUMBER.

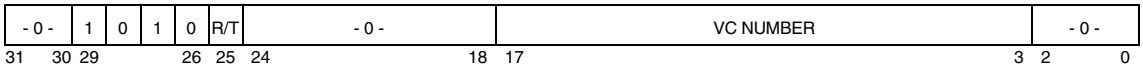
Command description

When the μPD98401A 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 μPD98401A 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 μPD98401A 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 17): "VC NUMBER" of the channel to be deactivated.

Command description

The Deactivate_Channel command is to get the μPD98401A 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 μPD98401A and the μPD98401A has not completed processing of the data, the contents of the VC table used by the channel are cleared and, as a result, the μPD98401A 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.

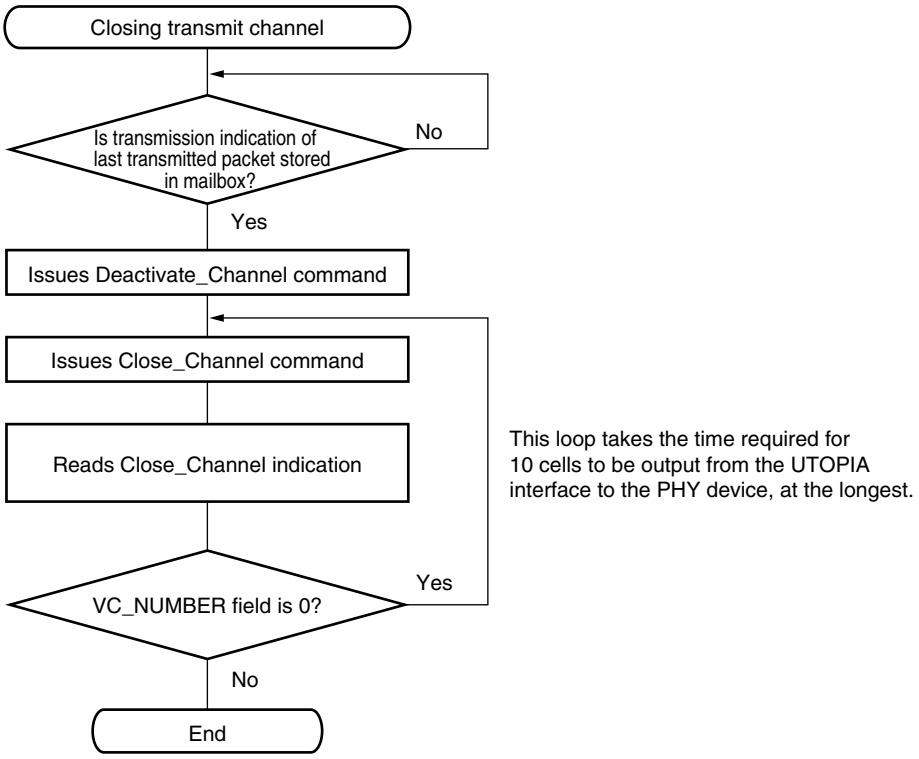
- **Transmit channel**

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

When the Close_Channel command is issued, the μPD98401A 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 μPD98401A 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 μPD98401A 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 6-3 illustrates how a transmit channel is closed.

Figure 6-3. 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 μPD98401A.

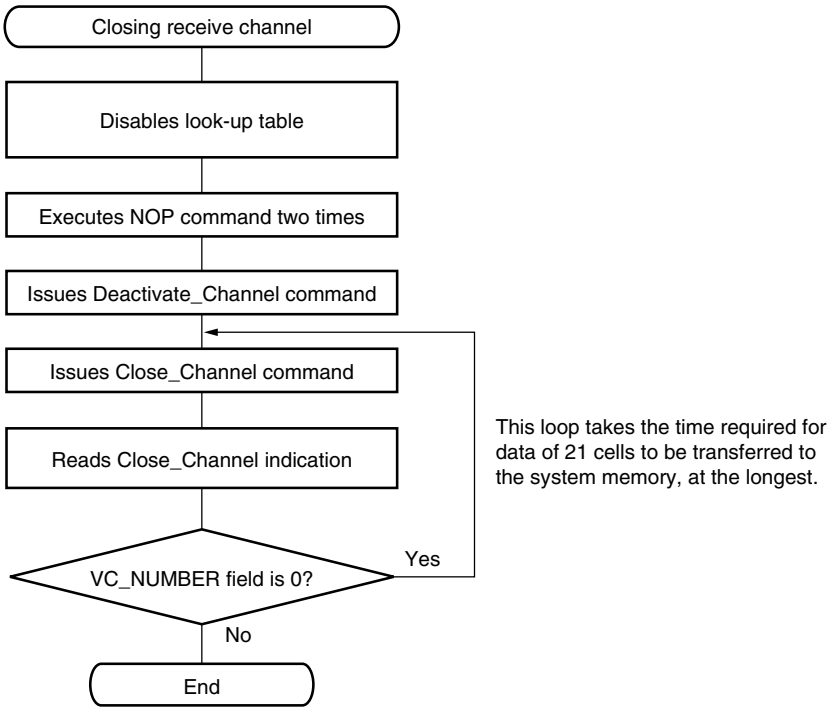
<5> Check to see if the VC NUMBER field is '0'. If so, perform proc essing from <3>.
If the VC NUMBER of the channel to be closed is stored in the VC NUMBER field, closing the transmit channel is completed.

- **Receive channel close**

  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 6-4 illustrates how a receive channel is closed.

Figure 6-4. Closing Receive Channel

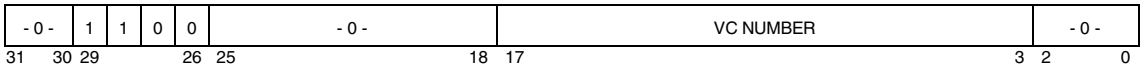


<1> Clear the enable bit of the receive look-up table.
<2> Insert a wait state by issuing the NOP command two times.
<3> After that, issue the Deactivate_Channel command.
<4> Issue the Close_Channel command.
<5> Read the Close_Channel indication.
<6> Check to see if the VC NUMBER field is '0'. If '0', 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.

Remark When the μPD98401A 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.

<6> Tx_Ready command

Programming of command register



Parameter description

VC NUMBER (bits 3 through 17): “VC NUMBER” of the channel to which a transmit packet has been added.

Command description

The Tx_Ready command is used by the host to report to the μPD98401A that a transmit packet has been added to this channel (that a new packet descriptor has been added to the transmit queue). When the μPD98401A receives the Tx_Ready command, it activates the targeted transmit VC table.

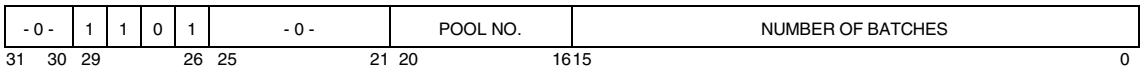
The active status of the channel is retained until a packet descriptor whose V bit is “0” arrives. Therefore, this Tx_Ready command does not have to be issued once to one transmit packet.

For example, if three valid packet descriptors are arranged in the transmit queue in a row and the Tx_Ready command is issued, the μPD98401A transmits the three packets and then enters the idle status. The μPD98401A ignores the Tx_Ready command issued to a channel in the active status, and does not influence the transmission/reception of that channel.

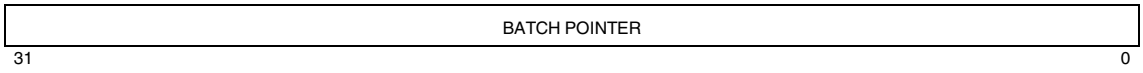
<7> Add_Batches command

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/CER_L). “BATCH POINTER” is written to the command extension register and then to the command register (CMR/CMR_L).

The operation of the μPD98401A when it has received the Add_Batches command differs depending on whether any unused batches remain in the targeted pool or not.

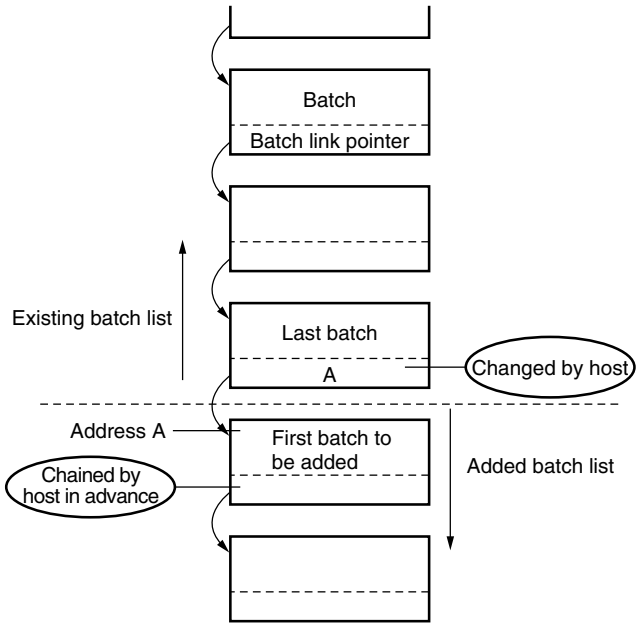
(a) If unused batches remain in pool

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

The μPD98401A 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 6-5**).

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

Figure 6-5. Work of Host before Issuance of Add_Batches Command



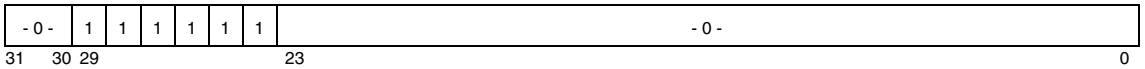
(b) If pool is empty

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

If the μPD98401A 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



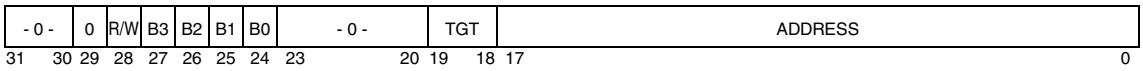
Command description

The host uses the NOP command to delay execution of the other commands (such as the Deactivate_Channel command). When the μPD98401A 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 : Select bytes. These control bits correspond to the CBE_B3 through CBE_B0 signals of the control memory interface. The μPD98401A sets the CBE_B signal corresponding to any of these bits that has been set to "1" during write access to a low level.
During read access, the μPD98401A makes all the CBE_B3 through CBE_B0 signals low; therefore, setting of these bit is invalid.

1 - CBE_B signal is enabled.
0 - CBE_B signal is disabled.

TGT (Target) : Specifies the target device.
'00' - Control memory
'01' - Indirect address register of μPD98401A
'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 μPD98401A.

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 μPD98401A
- Control memory
- PHY device

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

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 μPD98401A 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 μPD98401A 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 of the μPD98401A is executed by the internal bus of the μPD98401A. The cycle to access the control memory is executed by the control memory interface. The cycle to access the PHY device is executed by using the address lines (CA17 through CA0) and data lines (CD31 through CD0) of the control memory interface, and the control signals (PHOE_B, PHCE_B, and PHRW_B) of the PHY device interface.

If the data bus of the PHY device is connected to part of the control memory interface, the user makes the location of the data to be read/written by the command extension register correspond to the connected signals.

For example, if the PHY device has an 8-bit data bus and is connected to CD7 through CD0 of the control memory interface, the host sets the low-order bits 7 through 0 in the command extension register as data. At this time, the high-order 24 bits are ignored during write, and an undefined value is stored in these bits during read.

CHAPTER 7 INTERNAL REGISTERS

The internal registers of the μPD98401A 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.

7.1 Register List

(1) Direct address register

(1/2)

Address W (H)	Address B (H)	Register	Function	Read/Write
00	00	GMR	General mode register	Read/Write
01	04	GSR	General status register	Read
02	08	IMR	Interrupt mask register	Read/Write
03	0C	RQU	Receive queue underrun	Read
04	10	RQA	Receive queue alert	Read
05	14	ADDR	Last burst address	Read
06	18	VER	Version number	Read
07	1C	SWR	Software reset	Write
08	20	CMR	Command register (without lock/unlock)	Read/Write
09	24	CMR_L	Command register (with lock/unlock)	Read/Write
0A	28	CER	Command extension register (without unlock)	Read/Write
0B	2C	CER_L	Command extension register (with unlock)	Read/Write
10	40	MSH0	Mailbox 0 start address, high	Read/Write
11	44	MSH1	Mailbox 1 start address, high	Read/Write
12	48	MSH2	Mailbox 2 start address, high	Read/Write
13	4C	MSH3	Mailbox 3 start address, high	Read/Write
14	50	MSL0	Mailbox 0 start address, low	Read/Write
15	54	MSL1	Mailbox 1 start address, low	Read/Write
16	58	MSL2	Mailbox 2 start address, low	Read/Write
17	5C	MSL3	Mailbox 3 start address, low	Read/Write
18	60	MBA0	Mailbox 0 bottom address	Read/Write
19	64	MBA1	Mailbox 1 bottom address	Read/Write
1A	68	MBA2	Mailbox 2 start address, high	Read/Write
1B	6C	MBA3	Mailbox 3 start address, high	Read/Write
1C	70	MTA0	Mailbox 0 tail address	Read/Write
1D	74	MTA1	Mailbox 1 tail address	Read/Write
1E	78	MTA2	Mailbox 2 tail address	Read/Write
1F	7C	MTA3	Mailbox 3 tail address	Read/Write

(2/2)

Address W (H)	Address B (H)	Register	Function	Read/Write
20	80	MWA0	Mailbox 0 write address	Read/Write
21	84	MWA1	Mailbox 1 write address	Read/Write
22	88	MWA2	Mailbox 2 write address	Read/Write
23	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 μPD98401A.

Scheduler registers

Address (H)	Register	Function
40000 - 4000F	I, M	I and M entries of schedulers 0 through 15
40010 - 4001F	X	X entry of schedulers 0 through 15
40020 - 4002F	Y	Y entry of schedulers 0 through 15
40030 - 4003F	P, C, p, c	P, C, p, and c entries of schedulers 0 through 15
40040 - 4004F	Pri & Status	Priority and status of schedulers 0 through 15

Other registers

Address (H)	Register	Function
40070	RCS	Number of remaining cells setting register for DBMR pin output 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

7.2 Direct Address Registers

The direct address registers are located at addresses in two modes: word address mode and byte address mode. These two modes can be selected by the SLM bit of the GMR register. Select the appropriate mode in accordance with the address decoding method of the system to be configured.

(1) GMR (General Mode Register)

The GMR register is set by the host mainly to select the operation mode of the Time stamp register μPD98401A 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 μPD98401A, do not change its contents after the host has set the SE and RE bits of this register and the μPD98401A has started transmission/reception; otherwise, malfunctioning may occur.

Address : 00H
Access mode : Read/write

x	SLM	0	ICM	PSM	UOC	BME	- 0 -	TBE	CPE	LP	WA	RA	SZ	AD	BO	PM	PC	BPE	DR	SE	RE
31	30	29	28	27	26	25	24	17	16	15	14	13	12	11	8	7	6	5	4	3	2	1	0

x : Don't care.

Field	Function	Value after Reset
SLM	Selects the addresses of the direct address registers from two modes: word and byte. 0 - Word address mode (address W in Register List) 1 - Byte address mode (address B in Register List)	0: Word address mode
ICM	Idle cell mode. 0 - Transmits an unassigned cell to adjust the rate. 1 - Transmits an idle cell to adjust the rate. The cell transmitted by the unassigned cell generator function also serves as an idle cell.	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
BME	Bus monitor pin output enable. 0 - Bus monitor pin disable. DBMD, DBML, DBMF, DBMR, and DBVC pins go into a high-impedance state. 1 - Bus monitor pin enable.	0: Disable
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

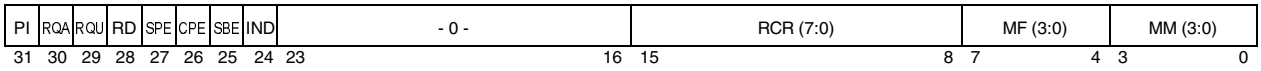
Field	Function	Value after Reset
CPE	Enables or disables the parity check function of the control memory. 0 - Parity disable 1 - Parity enable	0: Control memory parity disable
LP	Sets loopback mode 0 - Normal operation 1 - Loopback mode	0: Normal operation
WA	Sampling timing of the ABRT_B and RDY_B signals for DMA write operation. 0 - Normal mode 1 - Early mode	0: Normal
RA	Sampling timing of the ABRT_B and RDY_B signals for DMA read operation. 0 - Normal mode 1 - Early mode	0: Normal
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 μPD98401A 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 μPD98401A executes DMA transfer. 0 - Enable 1 - Disable	0: Disable
BO	Selects big endian or little endian. 0 - Little endian 1 - Big endian	0: Little endian
PM	Selects a bus parity mode. 0 - Byte parity mode 1 - Word parity mode	0: Byte parity
PC	Select even or odd bus parity. 0 - Even parity 1 - Odd parity	0: Even parity
BPE	Enables or disables bus parity check. 0 - Bus parity disable 1 - Bus parity enable	0: Bus parity 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 μPD98401A does not drop the cells.	0: Drop mode

Field	Function	Value after Reset
SE	Enables or disables the shapers. This bit enables the transmission function of the μPD98401A. 1 - Transmission enable 0 - Transmission disable	0: Transmission disable
RE	Enables or disables the reception function of the μPD98401A. 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 : 01H
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 NO. 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 NO. 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
SPE	Bus parity error detection. '1' indicates that a parity error has been detected on the host bus interface.	0
CPE	Control memory interface parity error detection. '1' indicates that a parity error has been detected on the control memory interface.	0
SBE	Bus error detection. '1' indicates that a low level has been input to the ERR_B input pin.	0

Field	Function	Value after Reset
IND	Control memory initialization complete. '1' indicates that initialization of the control memory performed by the μPD98401A 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 μPD98401A.	0
RCR (7:0)	Raw cell data reception. A bit in this field which is '1' indicates that the μPD98401A 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 μPD98401A 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 μPD98401A 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 : 02H
Access mode : Read/write

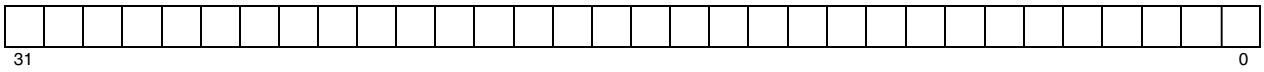
MASK
31 0

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 μPD98401A 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 : 03H
Access mode : Read only



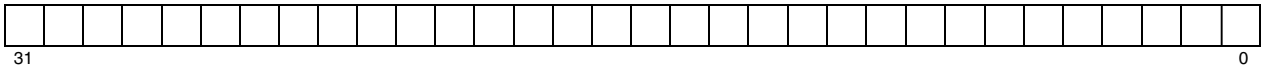
Field	Function	Value after Reset
Receive Queue Underrun	Bit 0 corresponds to pool 0, and bit 31 corresponds to pool 31. '1' indicates that there is no free buffer for the corresponding pool for the bit location.	All 0

(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 NO. 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 : 04H
Access mode : Read only

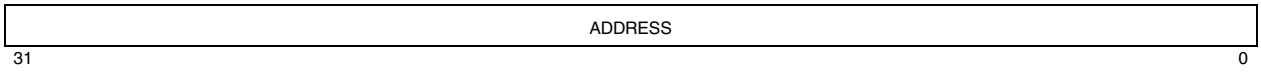


Field	Function	Value after Reset
Receive Queue Alert	Bit 0 corresponds to pool 0, and bit 31 corresponds to pool 31. '1' indicates that the number of remaining batches of the pool corresponding to the bit location has exceeded ALERT LEVEL. This means that the free buffers of the pool are about to be 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 μPD98401A is stored.

Address : 05H
Access mode : Read only

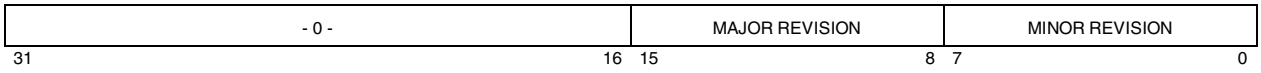


Field	Function	Value after Reset
Address	Last burst address. Transfer address of the last DMA cycle transfer executed up to the time the host reads this register.	All 0

(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 : 06H
Access mode : Read only



Field	Function	Value after Reset
MINOR REV.	Stores a code indicating the version of this chip.	Version number
MAJOR REV.		

(8) SWR (Software Reset virtual register)

Address : 07H
Access mode : Write only

SWR is a virtual register. Any write operation to address 07H causes software reset.

(9) CMR (Command Register)

Address : CMR : 08H
CMR_L: 09H
Access mode : Read/write

The CMR register is used by the host to write a command to the μPD98401A or to receive a command indication for the written command. For an explanation of how to use this register, refer to **CHAPTER 6 COMMANDS**. The CMR_L register is used in a multi-host system where the μPD98401A is controlled by multiple hosts. It is not used in a single-host system where the μPD98401A is controlled by only one host, and only the CMR register is used.

(10) CER (Command Extension Register)

Address : CER : 0AH
CER_L : 0BH
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**. The CER_L register is used in a multi-host system, like the CMR_L register.

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

Addresses : 10H - MSH0: Mailbox 0
11H - MSH1: Mailbox 1
12H - MSH2: Mailbox 2
13H - 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

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

Addresses : 14H - MSL0: Mailbox 0
15H - MSL1: Mailbox 1
16H - MSL2: Mailbox 2
17H - 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

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

Address : 18H - MBA0: Mailbox 0
19H - MBA1: Mailbox 1
1AH - MBA2: Mailbox 2
1BH - 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 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

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

Address : ‘1C’H - MTA0: Mailbox 0
‘1D’H - MTA1: Mailbox 1
‘1E’H - MTA2: Mailbox 2
‘1F’H - 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.

- 0 -	MTA0
31 16	15 0

- 0 -	MTA1
31 16	15 0

- 0 -	MTA2
31 16	15 0

- 0 -	MTA3
31 16	15 0

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

Address : 20H - MWA0: Mailbox 0
21H - MWA1: Mailbox 1
22H - MWA2: Mailbox 2
23H - 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 μPD98401A. Each time the μPD98401A has stored indication, it increments and updates the addresses of these registers. The initial values of these registers are undefined after reset. The host writes the same values as those of the MSL registers to these registers for initialization after reset.

- 0 -	MWA0
31 16	15 0

- 0 -	MWA1
31 16	15 0

- 0 -	MWA2
31 16	15 0

- 0 -	MWA3
31 16	15 0

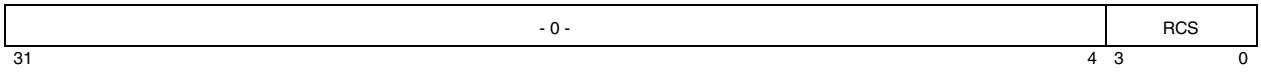
7.3 Indirect Address Registers

(1) RCS (Remaining Cell Set)

This register sets the number of cells remaining in the transmit buffer, which is the condition to make one of the bus monitor pins, DBMR, active, to 1 to 15. If 0 is assigned to this register, the DBMR pin does not become active but is disabled.

Address : 40070H

Access mode : Read/write

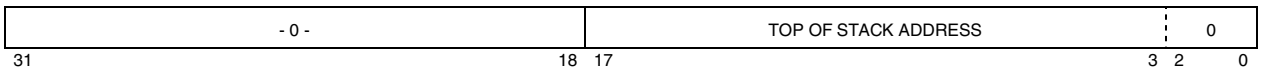


(2) TOS (Top Of Stack)

The host assigns the start address of a free block pool of the control memory to this register. After that, this register is managed by the μPD98401A 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 undefined after reset.

Address : 40100H

Access mode : Read/write

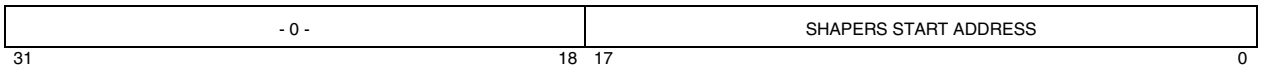


(3) SMA (Shaper Pointer Entry Start Address)

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

Address : 40200H

Access mode : Read/write

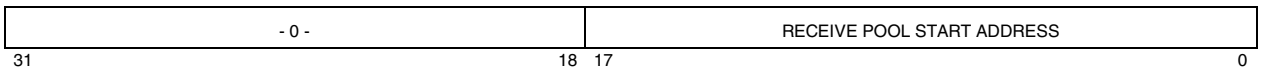


(4) PMA (Receive Free Buffer Pool Pointer Start Address)

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

Address : 40201H

Access mode : Read/write



(5) 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, and the low-order bits are “0000H”. The time is defined in system clock cycle units.

The initial value after reset is “FFFFH”.

Address : 40300H

Access mode : Read/write

- 0 -	T1 VALUE
31 16	15 0

(6) VRR (VPI/VCI Reduction Register)

Address : 40301H

Access mode : Read/write

The VRR register is used to set a 4-bit “SHIFT” parameter and a 16-bit “MASK” parameter used to reduce the 24 patterns of the received VPI/VCI to an internal 16-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 μPD98401A 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 “0000FFFFH’.

SDM	- 0 -	SHIFT	MASK
31	30 20	19 16	15 0

SMD : Global shutdown bit

0 - Normal operation

1 - Executes shutdown

(7) TSR (Time Stamp Register)

Address : 40302H

Access mode : Read/write

The TSR register is a 32-bit counter register that is counted up by the μPD98401A 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 μPD98401A immediately starts counting up after this register has been reset.

COUNT
31 0

(8) Scheduler registers

Five 32-bit registers are assigned to each of the 16 shapers. In all, therefore, eighty 32-bit registers are available. The host sets a parameter that determines the average rate and peak rate of the shaper to be used to these registers. The μPD98401A stores the variables used for calculation of traffic control, and flags indicating the status of a shaper in these registers. The initial values of all these registers are undefined.

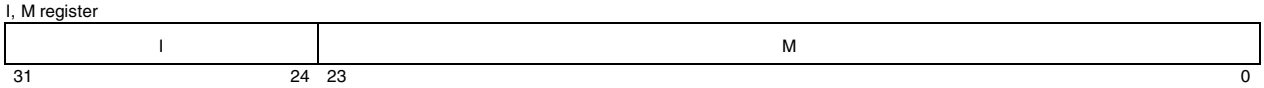
Cautions 1. Clear the enable bit (E) of the scheduler register only when the variable and flag used by the μPD98401A are cleared to 0 and after the parameters (I, M, P, C, and Priority) have been set.

2. The variable and flag of the scheduler register must start operation from all 0.
Because the enable bit of each shaper is undefined after reset, the operation is immediately started from the undefined value if the SE bit of the GMR register is set to 1. The host must clear the registers of all the 16 shapers to 0 before setting the SE bit to 1 after reset.

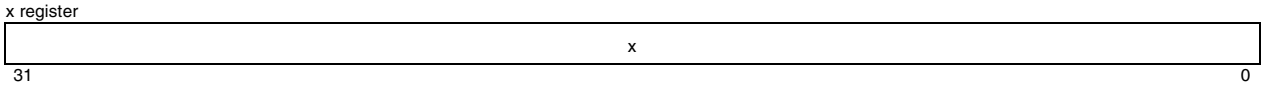
3. The contents of the scheduler registers cannot be changed if the corresponding shaper is active because the A bit = 1. A shaper becomes inactive when no VC table is linked. Before changing parameters I, M, P, and C, confirm that the shaper is inactive (A bit = 0) and clear the enable bit (E) to 0. At this time, clear all the variables and flags used by the μPD98401A to 0.

Address (H)	Register	Access Mode	Function
40000 - 4000F	I, M	Read/write	I and M parameters of schedulers 0 through 15
40010 - 4001F	x	Read/write	x value of schedulers 0 through 15
40020 - 4002F	y	Read/write	y value of schedulers 0 through 15
40030 - 4003F	P, C, p, c	Read/write	P and C parameters, and p and c values of schedulers 0 through 15
40040 - 4004F	Pin & Status	Read/write	Priority and status of schedulers 0 through 15

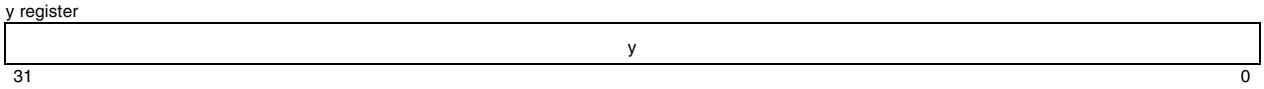
I, M register



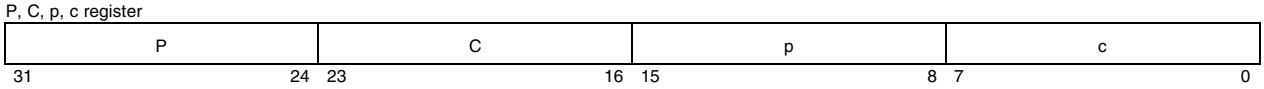
x register



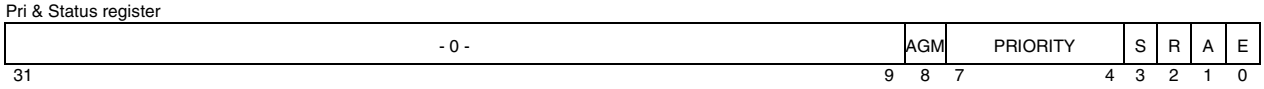
y register



P, C, p, c register



Pri & Status register



Field	Function	Value after Reset
I, M	The host sets an average rate parameter in these fields in cell units. I cells are transmitted for M cells.	Undefined
x, y	Used by the μPD98401A to store temporary parameters.	Undefined
P	Peak cell rate. Minimum time difference between two successive cells in the channel linked to the shaper. Set in cell units.	Undefined
C	Size of packet of shaper. Channel linked by shaper.	Undefined
p	Used by the μPD98401A to store temporary parameters.	Undefined
c	Used by the μPD98401A to store temporary parameters.	Undefined
AGM	Selects transmission mode of cell. 0 - Normal mode Rate is set for each VC. 1 - Aggregate mode Rate is set for each shaper.	Undefined
PRIORITY	Priority. 16 levels of priorities are set in this field by the host. '0000' is the highest priority, and '1111' is the lowest.	Undefined
S	Scan flag. Used by the μPD98401A to manage the shaper status. The user must not change the value of this field.	Undefined
R	Round-robin flag. Used by the μPD98401A to manage the shaper status. The user must not change the value of this field.	Undefined
A	Active flag. Used by the μPD98401A 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

CHAPTER 8 JTAG BOUNDARY SCAN

The μPD98401A has a JTAG boundary scan circuit.

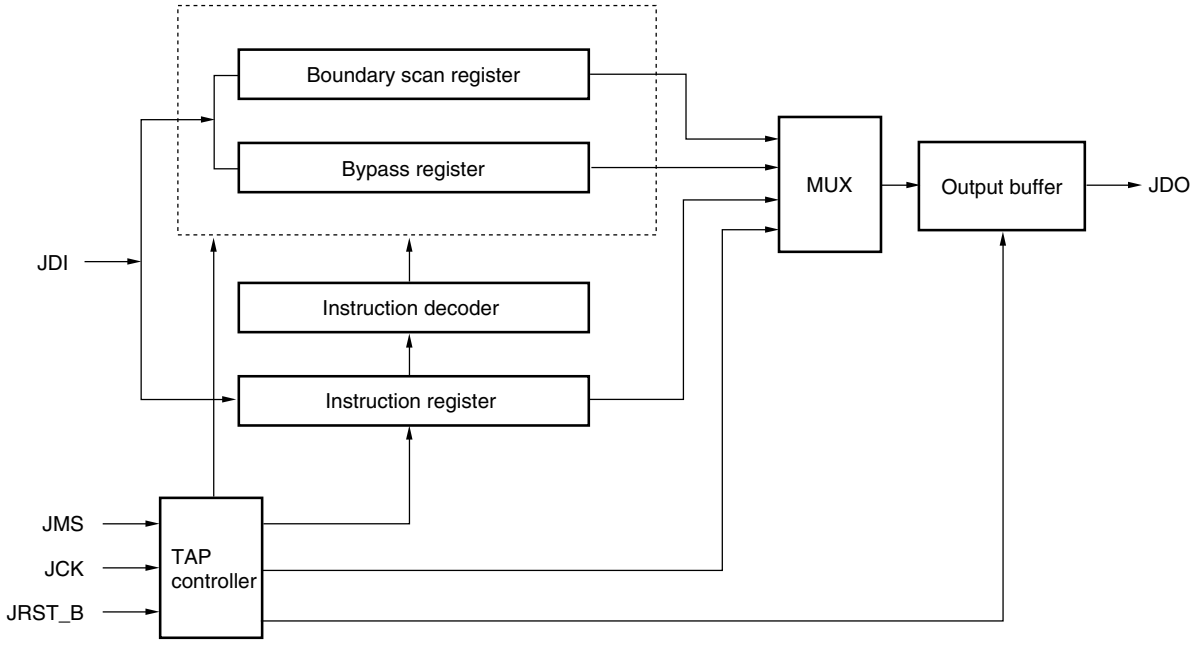
8.1 Features

- 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
- Five pins dedicated to boundary scan (five 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 μPD98401A.

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 μPD98401A.

8.2.4 Boundary scan register

The boundary scan register is located between an external pin of the μPD98401A 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 μPD98401A.

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 μPD98401A in the normal operation mode and the boundary register in non-operation state.

8.4 Operation Description

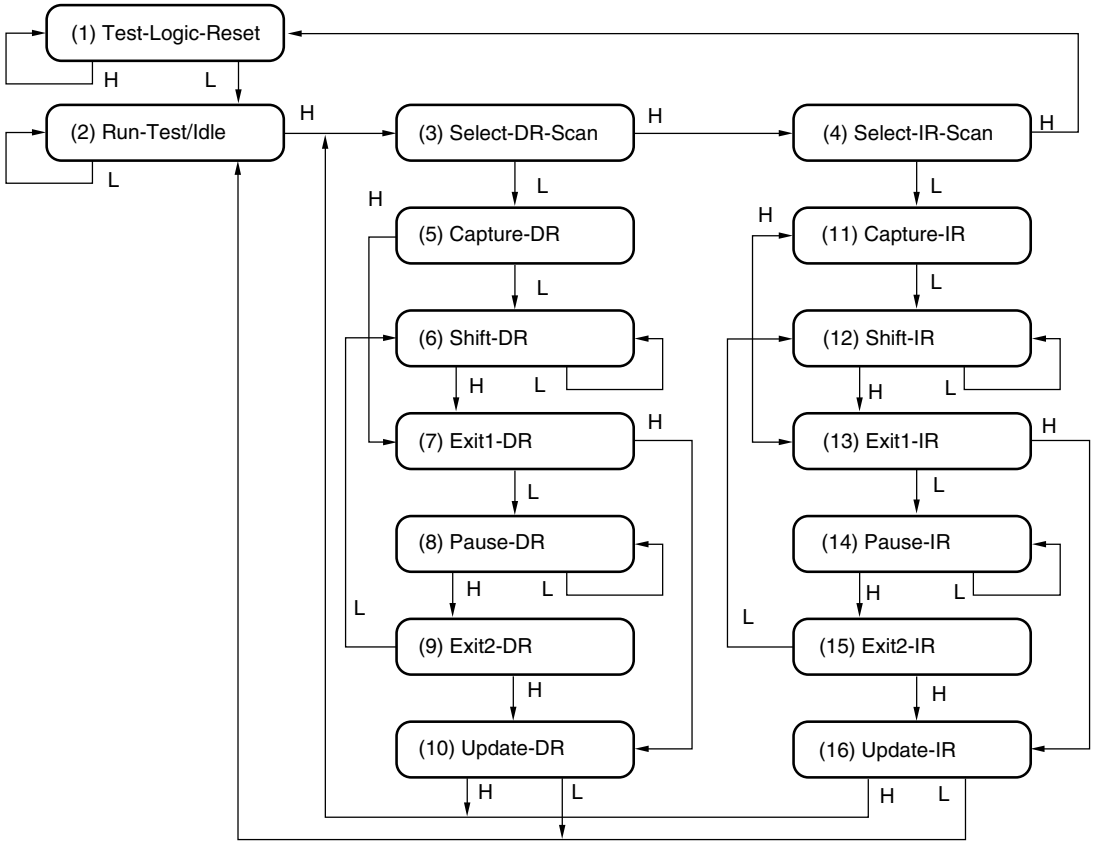
8.4.1 TAP controller

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**).

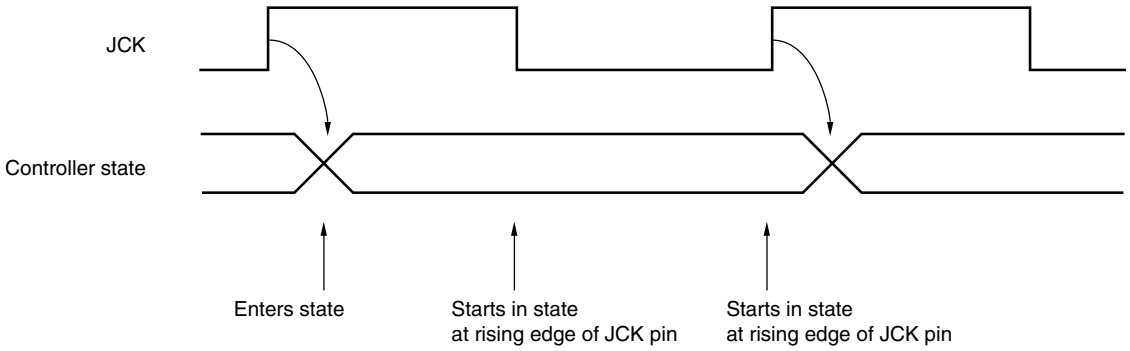
Figure 8-2. State Transition of TAP Controller



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 μPD98401A. Therefore, it does not affect the system logic of the μPD98401A. 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 μPD98401A 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 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 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 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 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 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 “01H” 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

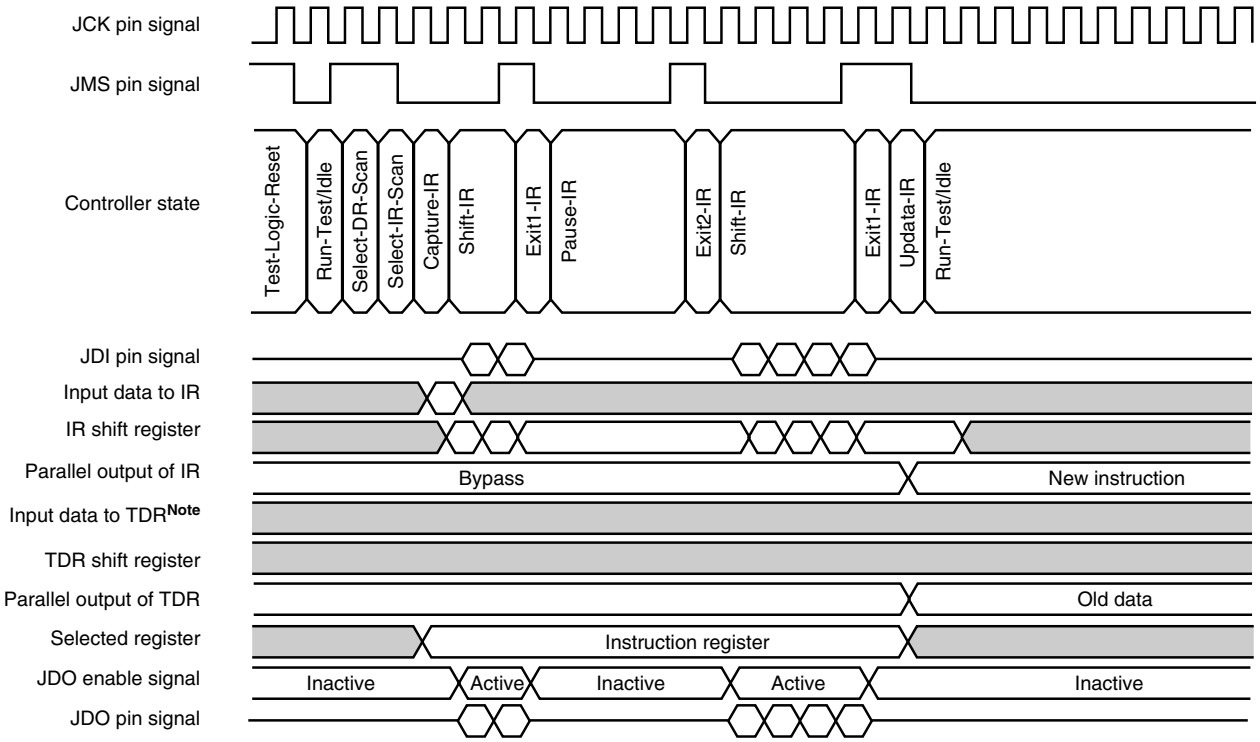
The TAP controller generates signals that control the operations of the bypass register, boundary scan register, and instruction register defined by the IEEE1149.1 JTAG Boundary Scan Standard (refer to **Figures 8-4** and **8-5**).

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		

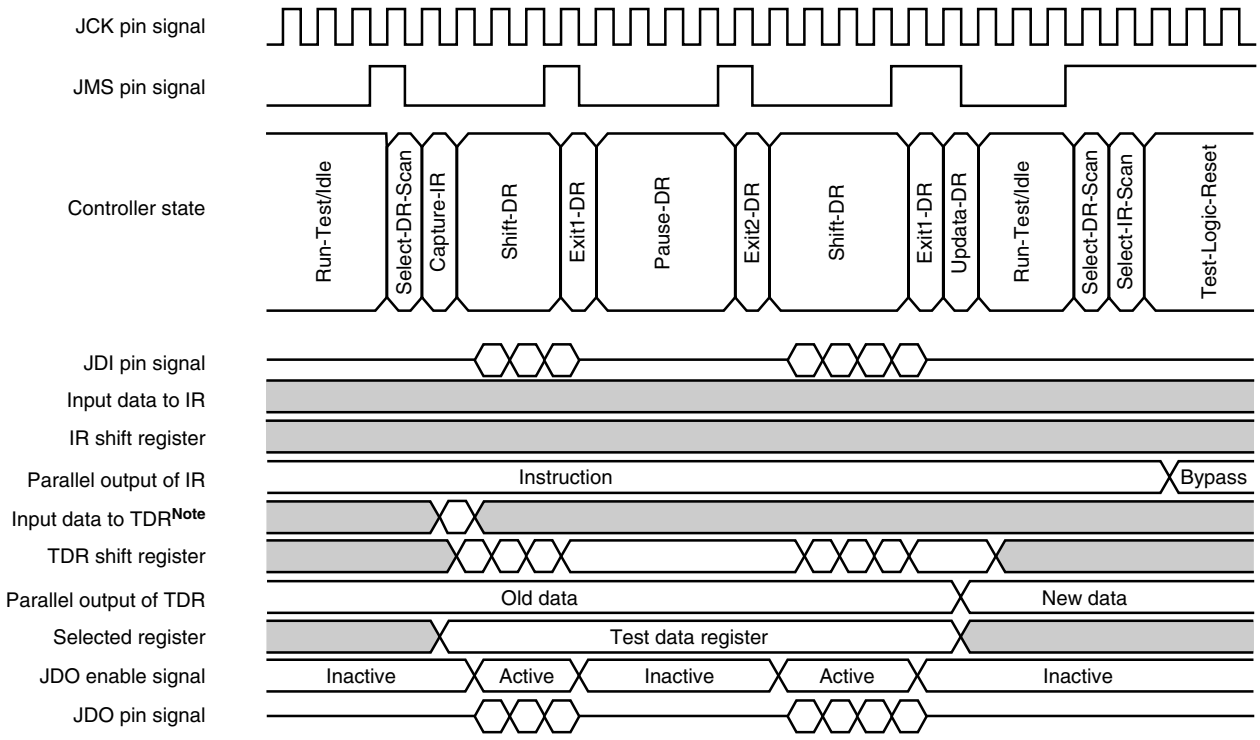
Figure 8-4. Operation of Test Logic (instruction scan)



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 μPD98401A can support only the following two instructions depending on the data set to the instruction register.

- BYPASS instruction
- EXTEST instruction

Instruction Register		Supported Instruction
D1	D0	
0	0	EXTEST instruction
0	1	Unused
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 μPD98401A.

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 Boundary scan data bit definition

Table 8-3 shows the allocated bit functions.

Table 8-2. μPD98401A Boundary Scan Data Bit Definition (1/4)

Bit	Function	Bit	Function
D0	Enable signal, DBMF pin	D37	Data input, CPAR1 pin
D1	Data output, DBMF pin	D38	Enable signal, CPAR2 pin
D2	Enable signal, DBML pin	D39	Data output, CPAR2 pin
D3	Data output, DBML pin	D40	Data input, CPAR2 pin
D4	Enable signal, DBMD pin	D41	Enable signal, CPAR3 pin
D5	Data output, DBMD pin	D42	Data output CPAR3 pin
D6	Data input, TRF_B pin	D43	Data input, CPAR3 pin
D7	Data input, INITD pin	D44	Enable signal, CD0 pin
D8	Data output, COE_B pin	D45	Data output, CD0 pin
D9	Data output, CWE_B pin	D46	Data input, CD0 pin
D10	Data output, CBE_B0 pin	D47	Enable signal, CD1 pin
D11	Data output, CBE_B1 pin	D48	Data output, CD1 pin
D12	Data output, CBE_B2 pin	D49	Data input, CD1 pin
D13	Data output, CBE_B3 pin	D50	Enable signal, CD2 pin
D14	Data output, CA0 pin	D51	Data output, CD2 pin
D15	Data output, CA1 pin	D52	Data input, CD2 pin
D16	Data output, CA2 pin	D53	Enable signal, CD3 pin
D17	Data output, CA3 pin	D54	Data output, CD3 pin
D18	Data output, CA4 pin	D55	Data input, CD3 pin
D19	Data output, CA5 pin	D56	Enable signal, CD4 pin
D20	Data output, CA6 pin	D57	Data output, CD4 pin
D21	Data output, CA7 pin	D58	Data input, CD4 pin
D22	Data output, CA8 pin	D59	Enable signal, CD5 pin
D23	Data output, CA9 pin	D60	Data output, CD5 pin
D24	Data output, CA10 pin	D61	Data input, CD5 pin
D25	Data output, CA11 pin	D62	Enable signal, CD6 pin
D26	Data output, CA12 pin	D63	Data output, CD6 pin
D27	Data output, CA13 pin	D64	Data input, CD6 pin
D28	Data output, CA14 pin	D65	Enable signal, CD7 pin
D29	Data output, CA15 pin	D66	Data output, CD7 pin
D30	Data output, CA16 pin	D67	Data input, CD7 pin
D31	Data output, CA17 pin	D68	Enable signal, CD8 pin
D32	Enable signal, CPAR0 pin	D69	Data output, CD8 pin
D33	Data output, CPAR0 pin	D70	Data input CD8 pin
D34	Data input, CPAR0 pin	D71	Enable signal, CD9 pin
D35	Enable signal, CPAR1 pin	D72	Data output, CD9 pin
D36	Data output, CPAR1 pin	D73	Data input, CD9 pin

Table 8-2. μPD98401A Boundary Scan Data Bit Definition (2/4)

Bit	Function	Bit	Function
D74	Enable signal, CD10 pin	D114	Data output, CD23 pin
D75	Data output, CD10 pin	D115	Data input, CD23 pin
D76	Data input, CD10 pin	D116	Enable signal, CD24 pin
D77	Enable signal, CD11 pin	D117	Data output, CD24 pin
D78	Data output, CD11 pin	D118	Data input, CD24 pin
D79	Data input, CD11 pin	D119	Enable signal, CD25 pin
D80	Enable signal, CD12 pin	D120	Data output, CD25 pin
D81	Data output, CD12 pin	D121	Data input, CD25 pin
D82	Data input, CD12 pin	D122	Enable signal, CD26 pin
D83	Enable signal, CD13 pin	D123	Data output, CD26 pin
D84	Data output, CD13 pin	D124	Data input, CD26 pin
D85	Data input, CD13 pin	D125	Enable signal, CD27 pin
D86	Enable signal, CD14 pin	D126	Data output, CD27 pin
D87	Data output CD14 pin	D127	Data input, CD27 pin
D88	Data input, CD14 Pin	D128	Enable signal, CD28 pin
D89	Enable signal, CD15 pin	D129	Data output, CD28 pin
D90	Data output, CD15 pin	D130	Data input, CD28 pin
D91	Data input, CD15 pin	D131	Enable signal, CD29 pin
D92	Enable signal, CD16 pin	D132	Data output, CD29 pin
D93	Data output, CD16 pin	D133	Data input, CD29 pin
D94	Data input, CD16 pin	D134	Enable signal, CD30 pin
D95	Enable signal, CD17 pin	D135	Data output, CD30 pin
D96	Data output, CD17 pin	D136	Data input, CD30 pin
D97	Data input, CD17 pin	D137	Enable signal, CD31 pin
D98	Enable signal, CD18 pin	D138	Data output, CD31 pin
D99	Data output, CD18 pin	D139	Data input pin, CD31 pin
D100	Data input, CD18 pin	D140	Data output, PHRW_B pin
D101	Enable signal, CD19 pin	D141	Data output, PHOE_B pin
D102	Data output, CD19 pin	D142	Data input, PHINT_B pin
D103	Data input, CD19 pin	D143	Data output, PHCE_B pin
D104	Enable signal, CD20 pin	D144	Data output, Tx0 pin
D105	Data output, CD20 pin	D145	Data output, Tx1 pin
D106	Data input, CD20 pin	D146	Data output, Tx2 pin
D107	Enable signal, CD21 pin	D147	Data output, Tx3 pin
D108	Data output, CD21 pin	D148	Data output, Tx4 pin
D109	Data input, CD21 pin	D149	Data output, Tx5 pin
D110	Enable signal, CD22 pin	D150	Data output, Tx6 pin
D111	Data output, CD22 pin	D151	Data output, Tx7 pin
D112	Data input, CD22 pin	D152	Data output, TCLK pin
D113	Enable signal, CD23 pin	D153	Data output, TENBL_B pin

Table 8-2. μPD98401A Boundary Scan Data Bit Definition (3/4)

Bit	Function	Bit	Function
D154	Data output, TSOC pin	D194	Enable signal, AD0 pin
D155	Data input, FULL_B pin	D195	Data output, AD0 pin
D156	Data input, EMPTY_B pin	D196	Data input, AD0 pin
D157	Data input, RSOC pin	D197	Enable signal, AD1 pin
D158	Data output, RENBL_B pin	D198	Data output, AD1 pin
D159	Data output, RCLK pin	D199	Data input, AD1 pin
D160	Data input, Rx0 pin	D200	Enable signal, AD2 pin
D161	Data input, Rx1 pin	D201	Data output, AD2 pin
D162	Data input, Rx2 pin	D202	Data input, AD2 pin
D163	Data input, Rx3 pin	D203	Enable signal, AD3 pin
D164	Data input, Rx4 pin	D204	Data output, AD3 pin
D165	Data input, Rx5 pin	D205	Data input, AD3 pin
D166	Data input, Rx6 pin	D206	Enable signal, AD4 pin
D167	Data input, Rx7 pin	D207	Data output, AD4 pin
D168	Data output, INTR_B pin	D208	Data input, AD4 pin
D169	Data input, ASEL_B pin	D209	Enable signal, AD5 pin
D170	Data input, SEL_B pin	D210	Data output, AD5 pin
D171	Data input, SR/W_B pin	D211	Data input, AD5 pin
D172	Data input, ERR_B pin	D212	Enable signal, AD6 pin
D173	Data input, ABRT_B pin	D213	Data output, AD6 pin
D174	Data input, RDY_B pin	D214	Data input, AD6 pin
D175	Data input, GNT_B pin	D215	Enable signal, AD7 pin
D176	Data output, ATTN_B pin	D216	Data output, AD7 pin
D177	Data output, DR/W_B pin	D217	Data input, AD7 pin
D178	Data output, SIZE0 pin	D218	Enable signal, AD8 pin
D179	Data output, SIZE1 pin	D219	Data output, AD8 pin
D180	Data output, SIZE2 pin	D220	Data input, AD8 pin
D181	Data input, OE_B pin	D221	Enable signal, AD9 pin
D182	Enable signal, PAR0 pin	D222	Data output, AD9 pin
D183	Data output, PAR0 pin	D223	Data input, AD9 pin
D184	Data input, PAR0 pin	D224	Enable signal, AD10 pin
D185	Enable signal, PAR1 pin	D225	Data output, AD10 pin
D186	Data output, PAR1 pin	D226	Data input, AD10 pin
D187	Data input, PAR1 pin	D227	Enable signal, AD11 pin
D188	Enable signal, PAR2 pin	D228	Data output, AD11 pin
D189	Data output, PAR2 pin	D229	Data input, AD11 pin
D190	Data input, PAR2 pin	D230	Data input, CLK pin
D191	Enable signal, PAR3 pin	D231	Data input, RST_B pin
D192	Data output, PAR3 pin	D232	Enable signal, AD12 pin
D193	Data input, PAR3 pin	D233	Data output, AD12 pin

Table 8-2. μPD98401A Boundary Scan Data Bit Definition (4/4)

Bit	Function	Bit	Function
D234	Data input, AD12 pin	D265	Enable signal, AD23 pin
D235	Enable signal, AD13 pin	D266	Data output, AD23 pin
D236	Data output, AD13 pin	D267	Data input, AD23 pin
D237	Data input, AD13 pin	D268	Enable signal, AD24 pin
D238	Enable signal, AD14 pin	D269	Data output, AD24 pin
D239	Data output, AD14 pin	D270	Data input, AD24 pin
D240	Data input, AD14 pin	D271	Enable signal, AD25 pin
D241	Enable signal, AD15 pin	D272	Data output, AD25 pin
D242	Data output, AD15 pin	D273	Data input, AD25 pin
D243	Data input, AD15 pin	D274	Enable signal, AD26 pin
D244	Enable signal, AD16 pin	D275	Data output, AD26 pin
D245	Data output, AD16 pin	D276	Data input, AD26 pin
D246	Data input, AD16 pin	D277	Enable signal, AD27 pin
D247	Enable signal, AD17 pin	D278	Data output, AD27 pin
D248	Data output, AD17 pin	D279	Data input, AD27 pin
D249	Data input, AD17 pin	D280	Enable signal, AD28 pin
D250	Enable signal, AD18 pin	D281	Data output, AD28 pin
D251	Data output, AD18 pin	D282	Data input, AD28 pin
D252	Data input, AD18 pin	D283	Enable signal, AD29 pin
D253	Enable signal, AD19 pin	D284	Data output, AD29 pin
D254	Data output, AD19 pin	D285	Data input, AD29 pin
D255	Data input, AD19 pin	D286	Enable signal, AD30 pin
D256	Enable signal, AD20 pin	D287	Data output, AD30 pin
D257	Data output, AD20 pin	D288	Data input, AD30 pin
D258	Data input, AD20 pin	D289	Enable signal, AD31 pin
D259	Enable signal, AD21 pin	D290	Data output, AD31 pin
D260	Data output, AD21 pin	D291	Data input, AD31 pin
D261	Data input, AD21 pin	D292	Enable signal, DBVC pin
D262	Enable signal, AD22 pin	D293	Data output, DBVC pin
D263	Data output, AD22 pin	D294	Enable signal, DBMR pin
D264	Data input, AD22 pin	D295	Data output, DBMR pin

APPENDIX A DIFFERENCES FROM μPD98401

A.1 Additional Functions

The μPD98401A is compatible with the μPD98401 in terms of hardware and software. However, the μPD98401A has the following additional functions as compared with the μPD98401. All the additional functions are enabled by the setting of the GMR register.

Additional Function	Refer to:	Title
Addition of DMA 12-word burst cycle	p.46	4.1.3 (2) Burst transfer
Additional byte alignment transfer function of receive data buffer	p.50	4.1.3 (4) Byte alignment transfer
Addition of bus monitor pin	p.52	4.1.3 (5) Byte monitor signal output function
Mode to insert idle cell for transmission rate adjustment	p.94	5.4.4 (3) Transmitting vacant cell
New scheduling function: aggregate mode	p.104	5.4.4 (8) (b) Aggregate mode
Receive packet size indication (cell units/Length mode added)	p.134	5.6 (2) Receive indication
Cell-level support of UTOPIA interface	p.63	4.2.1 UTOPIA interface
Addition of AAL-3/4 traffic assist function	p.132	5.5.7 (3) AAL-3/4 packet raw cell reception assist function
JTAG boundary scan support	p.167	CHAPTER 8 JTAG BOUNDARY SCAN

A.2 Differences

(1) Increased receive FIFO size ... p.122

μPD98401 : 10 cells
μPD98401A : 23 cells

(2) Cell processing of PTI field (1XX) ... p.131

μPD98401 : Receives cells other than those of OAM F5 pattern (101, 100) as user data cells.
μPD98401A : Processes as raw cell of 1XX pattern. Stores in pool 0.

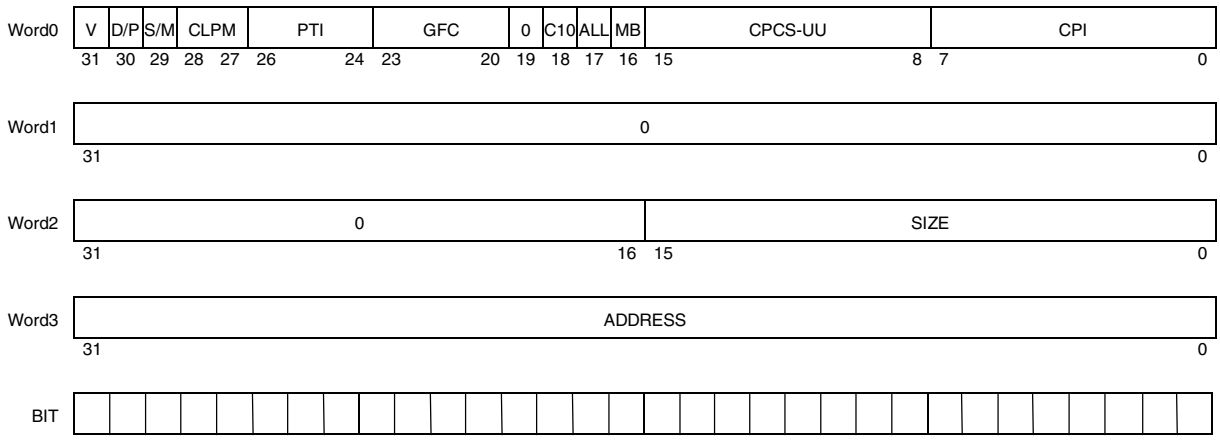
(3) Changing transmission mode of unassigned cell ... p.92

The μPD98401 starts transmitting unassigned cells immediately after power application and continues transmitting the unassigned cells while there is no active transmission VC. It also has a function to stop transmitting unassigned cells while there is not an active VC, by using the UCE bit of the GMR register.

The μPD98401A deletes this UCE bit function, makes the TENBL_B signal inactive on power application and when there is no active VC, and does not transmit unassigned cells. The μPD98401A transmits unassigned cells only when there is an active VC and when the unassigned cell generator function is enabled.

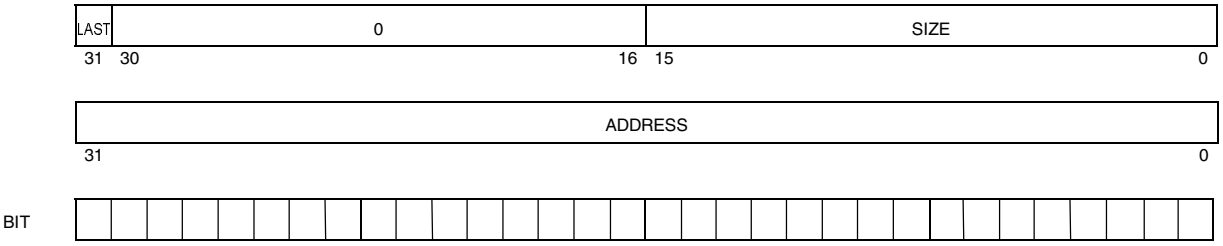
APPENDIX B DESCRIPTOR/INDICATION LIST

(1) Transmit packet descriptor



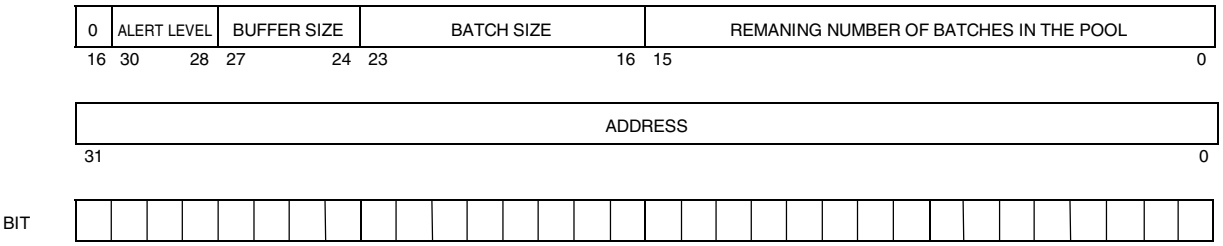
Register Name	Explanation
V D/P S/M	V / D/P / S/M / Function 0 / – / – / Blank packet descriptor 1 / 0 / – / Link pointer 1 / 1 / 0 / Multi-buffer mode 1 / 1 / 1 / Single-buffer mode
CLPM	00: Clears CLP of all cells to 0 11: Sets CLP of all cells to 1 01: Sets CLP bit of cells other than last cell to 1, clears CLP bit of last cell to 0 10: Setting prohibited
PTI	000 or 010: User data, 100 or 101: OAM F5 cell transmission
GFC	Any pattern
C10	1: CRC-10 operation enabled, 0: CRC-10 operation disabled
AAL	1: AAL-5 transmission, 0: Raw cell transmission
MB	1: Mailbox 3, 0: Mailbox 2
CPCS-UU CPI	Any pattern
SIZE	Size in byte units of packet
ADDRESS	Data buffer/packet directory/address of next packet descriptor

(2) Buffer descriptor



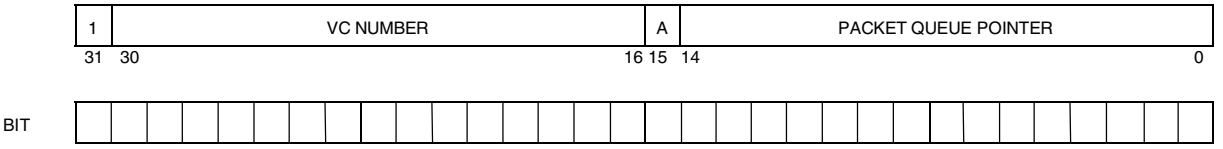
Register Name	Explanation
LAST	1: Last buffer descriptor, 0: Buffer descriptors other than last
SIZE	Size in byte units of packet
ADDRESS	Address of data buffer

(3) Receive pool descriptor



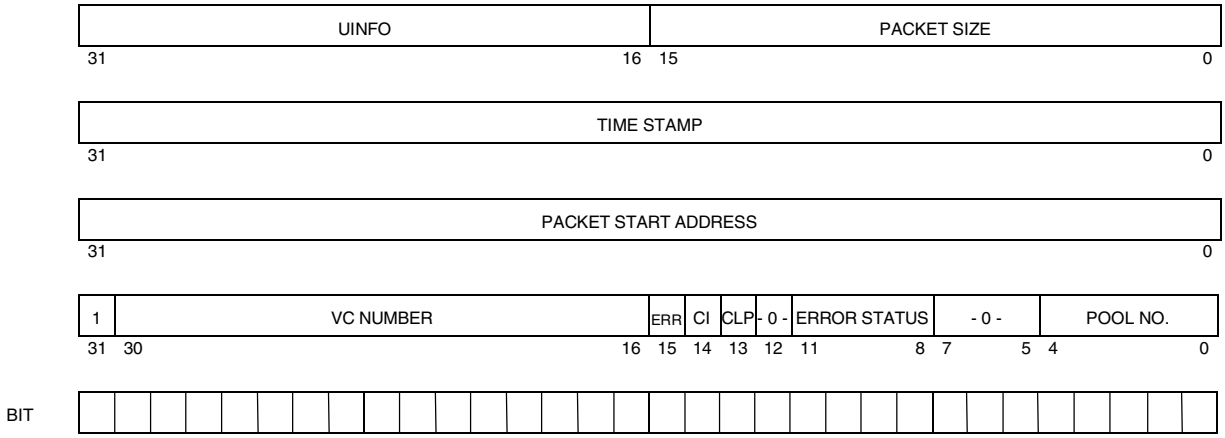
Register Name	Explanation
ALERT LEVEL	Alert of remaining number of batches. n x 4
BUFFER SIZE	Free buffer size. 64 x 2^n bytes
BATCH SIZE	Number of free buffers n bundled by one batch
REMAINING NO. OF BATCHES	Number of batches
ADDRESS	Batch start address

(4) Transmit indication



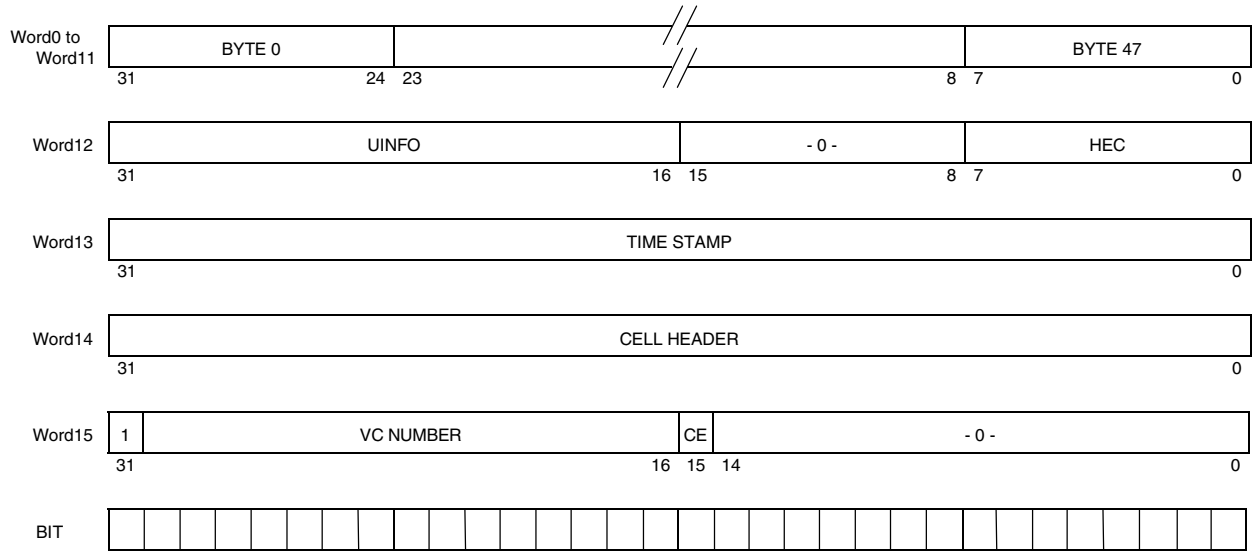
Register Name	Explanation
VC NUMBER	VC NUMBER used by this VC
A	0: Channel becomes inactive at last packet descriptor. 1: Channel remains active.
PACKET QUEUE POINTER	Low-order 15 bits of first address of next packet descriptor

(5) Receive indication



Register Name	Explanation
UNIFO	Any pattern set to UINFO of VC table
PACKET SIZE	Size in cell units of receive packet
TIME STAMP	TSR register value on reception of packet
PACKET START ADDRESS	Start address of first batch used
VC NUMBER	VC NUMBER used by this VC
ERR	1: Error occurs, 0: No error
CI	1: Congested cell received, 0: No congested cell
CLP	1: At least once cell with CLP = 1 is received, 0: CLP of all cells = 0
ERROR STATUS	0000 No error 0001 Free buffer underflow 0010 Receive FIFO overrun 0011 MAX. NO. OF SEGMENTS error 0100 CRC-32 error 0101 User abort 0110 Length error 0111 T1 error 1000 Deactivate_Channel command completion
POOL NO.	Stored pool number

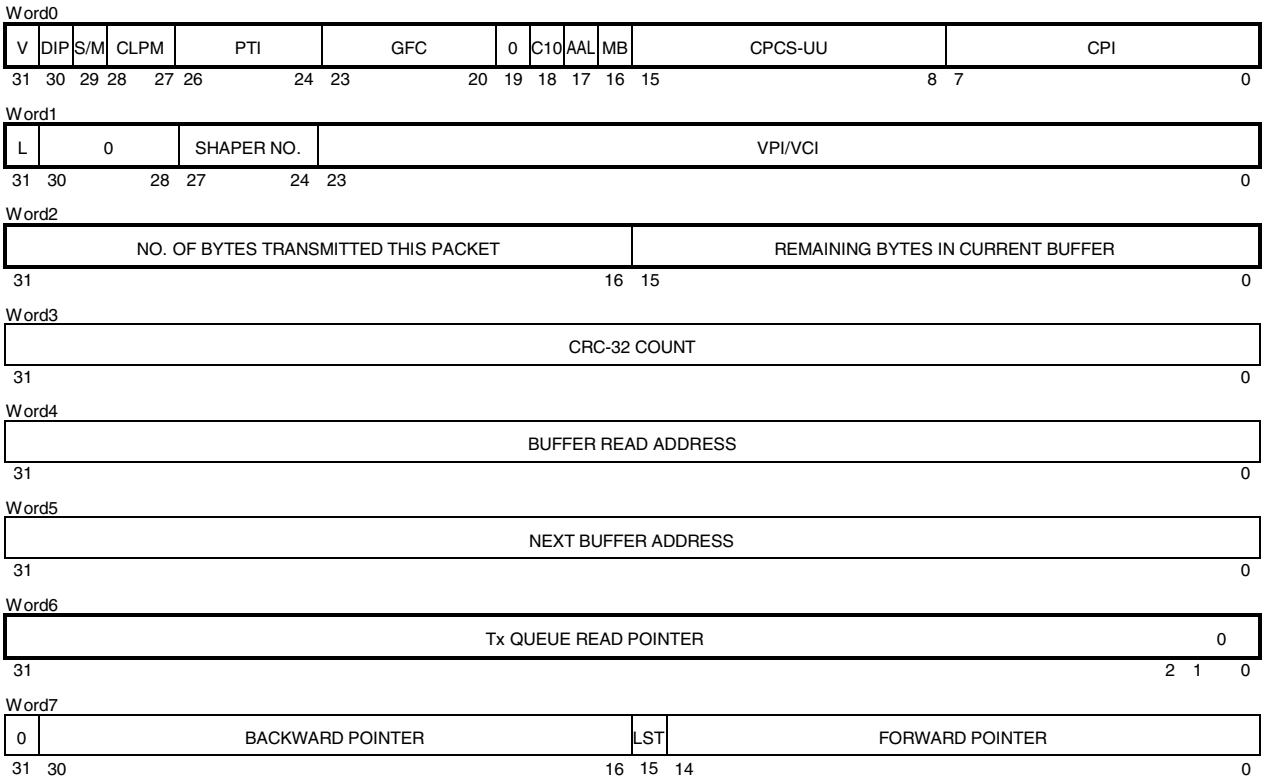
(6) Raw cell data



Register Name	Explanation
BYTE0-BYTE47	Segment data
UINFO	Any pattern set to VC table
HEC	5th byte of cell header
TIME STAMP	Value of TSR register on reception
CELL HEADER	1st through 4th bytes of cell header
VC NUMBER	VC NUMBER used by this VC
CE	1: CRC-10 error occurs, 0: CRC-10 error does not occur

APPENDIX C TRANSMIT/RECEIVE VC TABLE

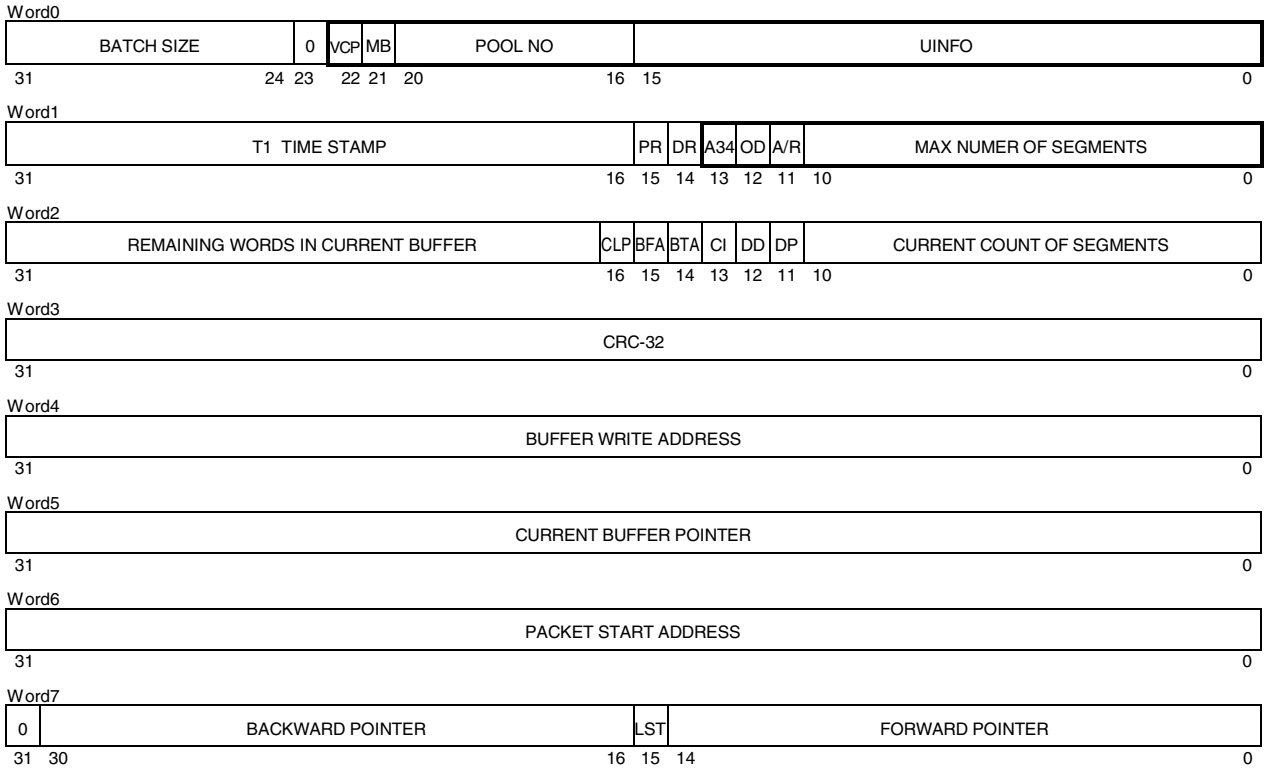
(1) Transmit VC table



The portions indicated by the bold solid line are set by the host. After that, they do not have to be set until the VC is closed.

Register Name	Explanation
Word 0	Cleared to 0 on initialization
L	Cleared to 0 on initialization
SHAPER NO.	Shaper number linked by this VC
VPI/VCI	VPI/VCI stored to header of this cell
NO. OF BYTES TRANSMITTED THIS PACKET	Cleared to 0 on initialization
REMAINING BYTES IN CURRENT BUFFER	
TRANSMIT QUEUE READ POINTER	Start address of first packet descriptor on initialization

(2) Receive VC table

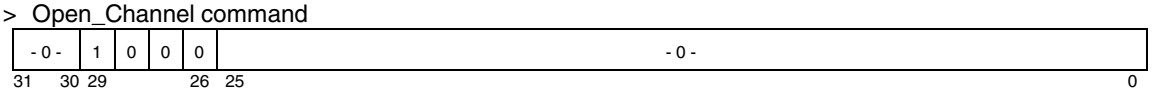


The portions indicated by the bold solid line are set by the host. After that, they do not have to be set until the VC is closed.

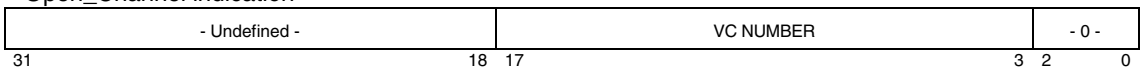
Register Name	Explanation
VCP	1: DBVC is high, 0: DBVC remains low
MB	1: Mailbox 1, 0: Mailbox 0
POOL NO.	Pool number to be stored
UINFO	Any pattern
A34	1: AAL-3/4 cell receive assist function enabled, 0: Disabled
OD	1: OAM F5 cell dropped, 0: OAM F5 cell received
A/R	1: AAL-5 packet received, 0: raw cell received
MAX. NO. OF SEGMENTS	Number of cells permitted for one packet

APPENDIX D COMMAND LIST

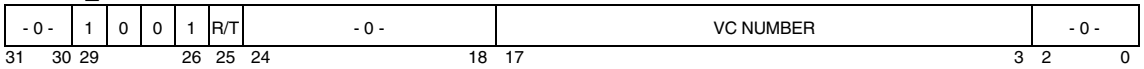
<1> Open_Channel command



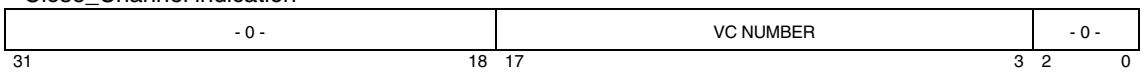
<2> Open_Channel indication



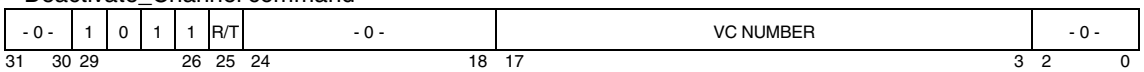
<3> Close_Channel command



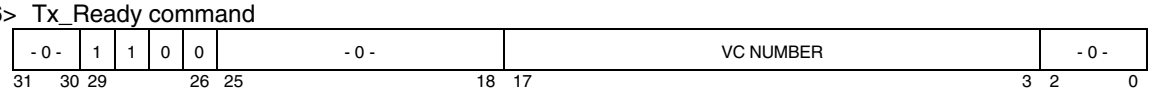
<4> Close_Channel indication



<5> Deactivate_Channel command

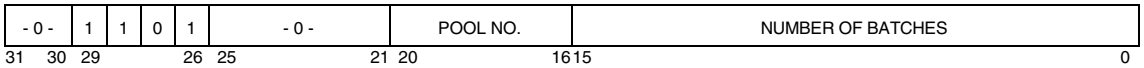


<6> Tx_Ready command

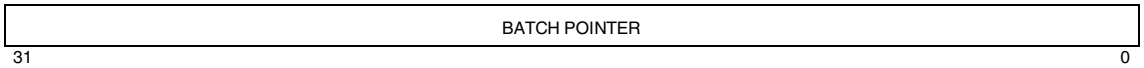


<7> Add_Batches command

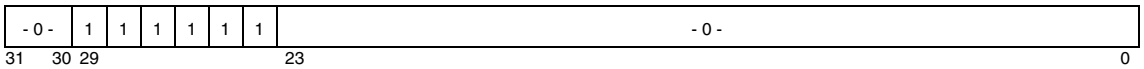
CMR:



CER:

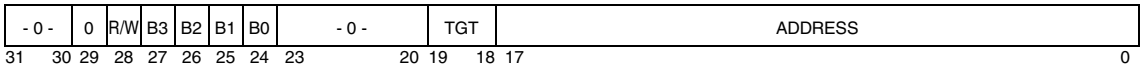


<8> NOP command



<9> Indirect_Access command

CMR:



CER:

Phase-out/Discontinued

NEC

Facsimile Message

Although NEC has taken all possible steps to ensure that the documentation supplied to our customers is complete, bug free and up-to-date, we readily accept that errors may occur. Despite all the care and precautions we've taken, you may encounter problems in the documentation. Please complete this form whenever you'd like to report errors or suggest improvements to us.

From:

Name

Company

Tel. FAX

Address

Thank you for your kind support.

North America
NEC Electronics Inc.
Corporate Communications Dept.
Fax: +1-800-729-9288
+1-408-588-6130

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

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

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

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

P.R. China
NEC Electronics Shanghai, Ltd.
Fax: +86-21-6841-1137

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

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

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

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

Document title: _____

Document number: _____ Page number: _____

If possible, please fax the referenced page or drawing.

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