chanhvinh.com

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.

chanhvinh.com

chanhvinh.com

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.

chanhvinh.com

RENESAS

User's Manual

μPD789800 Subseries

8-Bit Single-Chip Microcontrollers

μPD789800
μPD78F9801

Document No. U12978EJ3V3UD00 (3rd edition)
Date Published August 2005 N CP (K)

© NEC Electronics Corporation 1998, 2003
Printed in Japan

chanhvinh.com

[MEMO]

chanhvinh.com

chanhvinh.com

NOTES FOR CMOS DEVICES

① VOLTAGE APPLICATION WAVEFORM AT INPUT PIN

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

② HANDLING OF UNUSED INPUT PINS

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

③ PRECAUTION AGAINST ESD

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

④ STATUS BEFORE INITIALIZATION

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

⑤ POWER ON/OFF SEQUENCE

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

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

⑥ INPUT OF SIGNAL DURING POWER OFF STATE

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

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

chanhvinh.com

FIP and EEPROM are trademarks of NEC Electronics Corporation.

Windows and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

PC/AT is a trademark of International Business Machines Corporation.

HP9000 Series 700 and HP-UX are trademarks of Hewlett-Packard Company.

SPARCstation is a trademark of SPARC International, Inc.

Solaris and SunOS are trademarks of Sun Microsystems, Inc.

chanhvinh.com

These commodities, technology or software, must be exported in accordance with the export administration regulations of the exporting country.
Diversion contrary to the law of that country is prohibited.

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

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

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

M8E 02. 11-1

chanhvinh.com

Regional Information

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

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

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

[GLOBAL SUPPORT]
http://www.necel.com/en/support/support.html

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

NEC Electronics (Europe) GmbH
Duesseldorf, Germany
Tel: 0211-65030

- **Sucursal en España**
  Madrid, Spain
  Tel: 091-504 27 87
- **Succursale Française**
  Vélizy-Villacoublay, France
  Tel: 01-30-67 58 00
- **Filiale Italiana**
  Milano, Italy
  Tel: 02-66 75 41
- **Branch The Netherlands**
  Eindhoven, The Netherlands
  Tel: 040-265 40 10
- **Tyskland Filial**
  Taeby, Sweden
  Tel: 08-63 87 200
- **United Kingdom Branch**
  Milton Keynes, UK
  Tel: 01908-691-133

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-558-3737

NEC Electronics Shanghai Ltd.
Shanghai, P.R. China
Tel: 021-5888-5400

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

NEC Electronics Singapore Pte. Ltd.
Novena Square, Singapore
Tel: 6253-8311

J05.6

chanhvinh.com

Major Revisions in This Edition (1/2)

Page	Contents
Throughout	Deletion of CU-type and GB-3BS type packages
	Deletion of indication “under development” for μPD78F9801
p. 21	Modification of operating ambient temperature when flash memory is written in **1.1 Features**
p. 27	Addition of outline of timer in **1.7 Functions**
pp. 29, 31 to 33	Modification of handling of REGC and V_{PP} pins
pp. 35, 36	Correction of address values in **Figure 3-1 Memory Map (μPD789800)** and **Figure 3-2 Memory Map (μPD78F9801)**
p. 75	Modification of **Figure 5-3 External Circuit of System Clock Oscillator (b) External clock**
pp. 98, 103, 105, 106, 108 to 112, 115 to 117, 120, 125, 127 to 130	**CHAPTER 8 USB FUNCTION** • Modification of chapter composition • Standardization of buffer name indications as receive token bank, receive data bank, and transmit data banks 0 and 1 • Addition of image diagrams for reception and transmission • Addition of register value for SETUP reception • Modification of description on data handshake packet receive mode register (URXMOD) • Addition of description on packet receive status register (RXSTAT) and modification of read-only bit • Addition of **Note** for token packet receive result store register (TRXRSL) • Addition of **Caution** for data packet transmit reservation register (DTXRSV) • Modification of description of bit 1 (DNAEN) of handshake packet transmit reservation register (HTXRSV) • Change of contents of **8.5.2 Remote wakeup control operation** • Addition of **Table 8-4 List of Sources of Interrupts from USB Function** • Correction of incorrect flag name in **8.6 Interrupt Request from USB Function** • Addition of description on USB reset/Resume detection interrupt (INTUSBRE) • Addition of **8.7 USB Function Control**
p. 162	Modification of **Figure 10-1 Block Diagram of Regulator and USB Driver/Receiver** and **Cautions**
p. 164	Addition of **Remark** in **Table 11-1 Interrupt Source List**
p. 167	Addition of **Caution 3** on watchdog timer interrupt to **Figure 11-2 Format of Interrupt Request Flag Register**
p. 184	Addition of **12.2.2 STOP mode (3) Cautions on STOP instruction execution**
pp. 191 to 199	Revision of contents of flash memory programming as **14.1 Flash Memory Characteristics**
pp. 210 to 218	Addition of **CHAPTER 16 ELECTRICAL SPECIFICATIONS**
p. 219	Addition of **CHAPTER 17 PACKAGE DRAWING**
p. 220	Addition of **CHAPTER 18 RECOMMENDED SOLDERING CONDITIONS**
pp. 221 to 228	Revision of **APPENDIX A DEVELOPMENT TOOLS** Deletion of embedded software and addition of notes on target system design
pp. 233, 234	Addition of the revision contents in 3rd edition in **APPENDIX C REVISION HISTORY**
Major revisions in modification version (U12978EJ3V2UD00)	
p. 88	Modification of **Figure 6-8. Timing of External Event Counter Operation (with Rising Edge Specified)**
p. 213	Modification of conditions of V_{IL2} and V_{OL2} in **CHAPTER 16 ELECTRICAL SPECIFICATIONS**

The mark ★ shows major revised points.

chanhvinh.com

Major Revisions in This Edition (2/2)

Page	Contents
Major revisions in modification version (U12978EJ3V3UD00)	
pp. 22, 23	Addition of lead-free products in **CHAPTER 1 GENERAL**
p. 221	Addition of soldering conditions of lead-free products in **Table 18-1 Surface Mounting Type Soldering Conditions**

The mark ★ shows major revised points.

chanhvinh.com

INTRODUCTION

Readers

This manual is intended for users who wish to understand the functions of the μPD789800 Subseries and who design and develop its application systems and programs.

Target products:

- μPD789800 Subseries: μPD789800 and μPD78F9801

Purpose

This manual is intended to give users an understanding of the functions described in the **Organization** below.

Organization

Two manuals are available for the μPD789800 Subseries:

This manual and the Instruction Manual (common to the 78K/0S Series).

μPD789800 Subseries User's Manual	78K/0S Series User's Manual Instruction
• Pin functions • Internal block functions • Interrupts • Other internal peripheral functions • Electrical specifications	• CPU function • Instruction set • Instruction description

How to Read This Manual

It is assumed that the readers of this manual have general knowledge in the fields of electrical engineering, logic circuits, and microcontrollers.

- To understand the overall functions of the μPD789800 Subseries
  → Read this manual in the order of the **CONTENTS**.
- How to read register formats
  → The name of a bit whose number is enclosed in angle brackets (< >) is reserved in the assembler and is defined in the C compiler by the header file sfrbit.h.
- To learn the detailed functions of a register whose register name is known
  → See **APPENDIX B REGISTER INDEX**.
- To learn details of the instruction functions of the 78K/0S Series
  → Refer to **78K/0S Series Instruction User's Manual (U11047E)** separately available.
- To know the electrical specifications of the μPD789800 Subseries
  → Refer to **CHAPTER 16 ELECTRICAL SPECIFICATIONS**.

Conventions

Data significance:	Higher digits on the left and lower digits on the right
Active low representation:	$\overline{xxx}$ (overscore over pin or signal name)
Note:	Footnote for item marked with **Note** in the text
Caution:	Information requiring particular attention
Remark:	Supplementary information
Numerical representation:	Binary ... xxxx or xxxxB
	Decimal ... xxxx
	Hexadecimal ... xxxxH

chanhvinh.com

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

Documents Related to Devices

Document Name	Document No.
μPD789800 Subseries User's Manual	This manual
78K/0S Series Instructions User's Manual	U11047E

Documents Related to Development Tools (Software) (User's Manuals)

Document Name		Document No.
RA78K0S Assembler Package	Operation	U14876E
	Language	U14877E
	Structured Assembly Language	U11623E
CC78K0S C Compiler	Operation	U14871E
	Language	U14872E
SM78K Series System Simulator Ver. 2.30 or Later	Operation (Windows™ Based)	U15373E
	External Part User Open Interface Specifications	U15802E
ID78K Series Integrated Debugger Ver. 2.30 or Later	Operation (Windows Based)	U15185E
Project Manager Ver. 3.12 or Later (Windows Based)		U14610E

Documents Related to Development Tools (Hardware) (User's Manuals)

Document Name	Document No.
IE-78K0S-NS In-Circuit Emulator	U13549E
IE-78K0S-NS-A In-Circuit Emulator	U15207E
IE-789801-NS-EM1 Emulation Board	U13390E

Documents Related to Flash Memory Writing

Document Name	Document No.
PG-FP3 Flash Memory Programmer User's Manual	U13502E
PG-FP4 Flash Memory Programmer User's Manual	U15260E

Other Related Documents

Document Name	Document No.
SEMICONDUCTOR SELECTION GUIDE - Products and Packages - (CD-ROM)	X13769X
Semiconductor Device Mounting Technology Manual	C10535E
Quality Grades on NEC Semiconductor Devices	C11531E
NEC Semiconductor Device Reliability/Quality Control System	C10983E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E

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

chanhvinh.com

TABLE OF CONTENTS

CHAPTER 1 GENERAL 22

1.1 Features 22
1.2 Applications 22
1.3 Ordering Information 22
1.4 Pin Configuration (Top View) 23
1.5 78K/0S Series Lineup 24
1.6 Block Diagram 27
1.7 Functions 28

CHAPTER 2 PIN FUNCTIONS 29

2.1 List of Pin Functions 29
2.2 Pin Functions 31
2.2.1 P00 to P07 (Port 0) 31
2.2.2 P10 to P17 (Port 1) 31
2.2.3 P20 to P26 (Port 2) 31
2.2.4 P40 to P47 (Port 4) 32
2.2.5 $\overline{\text{RESET}}$ 32
2.2.6 X1, X2 32
2.2.7 REGC 32
2.2.8 USBDM 32
2.2.9 USBDP 32
2.2.10 V_{DD0}, V_{DD1} 32
2.2.11 V_{SS0}, V_{SS1} 32
2.2.12 V_{PP} (μPD78F9801 only) 33
2.2.13 IC (mask ROM version only) 33
2.3 Pin I/O Circuits and Recommended Connection of Unused Pins 34

CHAPTER 3 CPU ARCHITECTURE 36

3.1 Memory Space 36
3.1.1 Internal program memory space 38
3.1.2 Internal data memory (internal high-speed RAM) space 38
3.1.3 Special function register (SFR) area 38
3.1.4 Data memory addressing 39
3.2 Processor Registers 41
3.2.1 Control registers 41
3.2.2 General-purpose registers 44
3.2.3 Special function registers (SFRs) 45
3.3 Instruction Address Addressing 49
3.3.1 Relative addressing 49
3.3.2 Immediate addressing 50
3.3.3 Table indirect addressing 51

chanhvinh.com

3.3.4 Register addressing 51
3.4 Operand Address Addressing 52
3.4.1 Direct addressing 52
3.4.2 Short direct addressing 53
3.4.3 Special function register (SFR) addressing 54
3.4.4 Register addressing 55
3.4.5 Register indirect addressing 56
3.4.6 Based addressing 57
3.4.7 Stack addressing 57

CHAPTER 4 PORT FUNCTIONS 58

4.1 Port Functions 58
4.2 Port Configuration 60
4.2.1 Port 0 61
4.2.2 Port 1 62
4.2.3 Port 2 63
4.2.4 Port 4 69
4.3 Registers Controlling Port Function 70
4.4 Port Function Operation 73
4.4.1 Writing to I/O port 73
4.4.2 Reading from I/O port 73
4.4.3 Arithmetic operation of I/O port 73

CHAPTER 5 CLOCK GENERATOR 74

5.1 Clock Generator Functions 74
5.2 Clock Generator Configuration 74
5.3 Register Controlling Clock Generator 75
5.4 System Clock Oscillators 76
5.4.1 System clock oscillator 76
★ 5.4.2 Examples of incorrect resonator connection 77
5.4.3 Frequency divider 78
5.5 Clock Generator Operation 78
5.6 Changing Setting of CPU Clock 79
5.6.1 Time required for switching CPU clock 79
5.6.2 Switching CPU clock 79

CHAPTER 6 8-BIT TIMER/EVENT COUNTERS 00 AND 01 80

6.1 Functions of 8-Bit Timer/Event Counters 00 and 01 80
6.2 Configuration of 8-Bit Timer/Event Counters 00 and 01 81
6.3 Registers Controlling 8-Bit Timer/Event Counters 00 and 01 83
6.4 Operation of 8-Bit Timer/Event Counters 00 and 01 86
6.4.1 Operation as interval timer 86
6.4.2 Operation as external event counter (timer 01 only) 88
6.4.3 Operation as square-wave output (timer 01 only) 89
6.5 Notes on Using 8-Bit Timer/Event Counters 00 and 01 91

chanhvinh.com

CHAPTER 7 WATCHDOG TIMER 92

7.1 Watchdog Timer Functions 92
7.2 Watchdog Timer Configuration 93
7.3 Registers Controlling Watchdog Timer 94
7.4 Watchdog Timer Operation 96
7.4.1 Operation as watchdog timer 96
7.4.2 Operation as interval timer 97

CHAPTER 8 USB FUNCTION 98

8.1 USB Overview 98
8.2 USB Function Features 99
8.3 USB Function Configuration 99
8.4 Registers Controlling USB Function 110
8.5 USB Function Operation 123
8.5.1 USB timer operation 123
8.5.2 Remote wakeup control operation 126
8.6 Interrupt Request from USB Function 128
8.6.1 Interrupt sources 128
8.6.2 Cautions when using interrupts 130
8.7 USB Function Control 131
8.7.1 Relationship between packets and operation modes 131
8.7.2 Interrupt servicing flow 137
8.8 USB Function Internal Circuit Operations 141
8.8.1 Operation of transmit/receive pointer 141
8.8.2 Receive bank switching ID detection buffer operation 148
8.8.3 Sync detection/USBCLK detector operation 149
8.8.4 NRZI encoder operation 151
8.8.5 Bit stuffing/strip controller operation 152

CHAPTER 9 SERIAL INTERFACE 10 155

9.1 Functions of Serial Interface 10 155
9.2 Configuration of Serial Interface 10 156
9.3 Register Controlling Serial Interface 10 158
9.4 Operation of Serial Interface 10 160
9.4.1 Operation stop mode 160
9.4.2 3-wire serial I/O mode 161

CHAPTER 10 REGULATOR 163

CHAPTER 11 INTERRUPT FUNCTIONS 164

11.1 Interrupt Function Types 164
11.2 Interrupt Sources and Configuration 164
11.3 Registers Controlling Interrupt Function 167
11.4 Interrupt Servicing Operation 172
11.4.1 Non-maskable interrupt acknowledgment operation 172

chanhvinh.com

11.4.2 Maskable interrupt acknowledgment operation 174
11.4.3 Multiplexed interrupt servicing 176
11.4.4 Interrupt request hold 178

CHAPTER 12 STANDBY FUNCTION 179

12.1 Standby Function and Configuration 179
12.1.1 Standby function 179
12.1.2 Register controlling standby function 180
12.2 Standby Function Operation 181
12.2.1 HALT mode 181
12.2.2 STOP mode 184

CHAPTER 13 RESET FUNCTION 187

CHAPTER 14 μPD78F9801 191

14.1 Flash Memory Characteristics 192
14.1.1 Programming environment 192
14.1.2 Communication mode 193
14.1.3 On-board pin processing 196
14.1.4 Connection of adapter for flash writing 199

CHAPTER 15 INSTRUCTION SET 201

15.1 Operation 201
15.1.1 Operand identifiers and description methods 201
15.1.2 Description of “operation” column 202
15.1.3 Description of “flag operation” column 202
15.2 Operation List 203
15.3 Instructions Listed by Addressing Type 208

★ **CHAPTER 16 ELECTRICAL SPECIFICATIONS 211**

★ **CHAPTER 17 PACKAGE DRAWINGS 220**

★ **CHAPTER 18 RECOMMENDED SOLDERING CONDITIONS 221**

APPENDIX A DEVELOPMENT TOOLS 222

★ **A.1 Software Package 224**
A.2 Language Processing Software 224
★ **A.3 Control Software 225**
A.4 Flash Memory Writing Tools 225
A.5 Debugging Tools (Hardware) 226
A.6 Debugging Tools (Software) 227
A.7 Notes on Target System Design 228

APPENDIX B REGISTER INDEX 230

chanhvinh.com

B.1 Register Index (Alphabetic Order of Register Name) 230
B.2 Register Index (Alphabetic Order of Register Symbol) 232

APPENDIX C REVISION HISTORY 234

chanhvinh.com

LIST OF FIGURES (1/4)

Figure No.	Title	Page
2-1	Pin I/O Circuits	35
3-1	Memory Map (μPD789800)	36
3-2	Memory Map (μPD78F9801)	37
3-3	Data Memory Addressing (μPD789800)	39
3-4	Data Memory Addressing (μPD78F9801)	40
3-5	Configuration of Program Counter	41
3-6	Configuration of Program Status Word	41
3-7	Configuration of Stack Pointer	43
3-8	Data to Be Saved to Stack Memory	43
3-9	Data to Be Restored from Stack Memory	43
3-10	Configuration of General-Purpose Registers	44
4-1	Port Types	58
4-2	Block Diagram of P00 to P07	61
4-3	Block Diagram of P10 to P17	62
4-4	Block Diagram of P20	63
4-5	Block Diagram of P21	64
4-6	Block Diagram of P22	65
4-7	Block Diagram of P23 and P24	66
4-8	Block Diagram of P25	67
4-9	Block Diagram of P26	68
4-10	Block Diagram of P40 to P47	69
4-11	Format of Port Mode Register	70
4-12	Format of Pull-up Resistor Option Register 0	71
4-13	Format of Port Output Mode Register 0	72
4-14	Format of Port Output Mode Register 1	72
5-1	Block Diagram of Clock Generator	74
5-2	Format of Processor Clock Control Register	75
5-3	External Circuit of System Clock Oscillator	76
5-4	Examples of Incorrect Resonator Connection	77
5-5	Switching of CPU Clock	79
6-1	Block Diagram of 8-Bit Timer 00	81
6-2	Block Diagram of 8-Bit Timer/Event Counter 01	82
6-3	Format of 8-Bit Timer Mode Control Register 00	83
6-4	Format of 8-Bit Timer Mode Control Register 01	84
6-5	Format of Port Mode Register 2	85
6-6	Interval Timer Operation Timing of 8-Bit Timer 00	87

chanhvinh.com

LIST OF FIGURES (2/4)

Figure No.	Title	Page
6-7	Interval Timer Operation Timing of 8-Bit Timer/Event Counter 01	87
6-8	Timing of External Event Counter Operation (with Rising Edge Specified)	88
6-9	Timing of Square-Wave Output	90
6-10	Start Timing of 8-Bit Timer Counter	91
6-11	Timing of External Event Counter Operation	91
7-1	Block Diagram of Watchdog Timer	93
7-2	Format of Timer Clock Select Register 2	94
7-3	Format of Watchdog Timer Mode Register	95
8-1	USB Bus Topology (Desktop Type PC)	98
8-2	Block Diagram of USB Function	100
8-3	Block Diagram of USB Timer	101
8-4	Configuration of Receive Token Bank	103
8-5	Configuration of Receive Data Bank	104
8-6	Configuration of Transmit Data Bank 0 (Buffer 0)	105
8-7	Configuration of Transmit Data Bank 1 (Buffer 1)	106
8-8	Configuration of TIDCMP and ADRCMP	108
8-9	Configuration of DIDCMP	109
8-10	Format of USB Receiver Enable Register	110
8-11	Format of Data/Handshake Packet Receive Mode Register	111
8-12	Format of Packet Receive Status Register	113
8-13	Format of Data/Handshake Packet Receive Result Store Register	114
8-14	Format of Token Packet Receive Result Store Register	115
8-15	Format of Data Packet Transmit Reservation Register	116
8-16	Format of Handshake Packet Transmit Reservation Register	117
8-17	Configuration of Handshake Packet Transmit Reservation Register	120
8-18	Format of USB Timer Start Reservation Control Register	121
8-19	Format of Remote Wakeup Control Register	122
8-20	Flowchart of USB Timer Operation	124
8-21	Flow Chart of Remote Wakeup Control Operation	126
8-22	Configuration of Remote Wakeup Control	127
8-23	Timing of Data/Handshake Packet Receive Interrupt Request Generation	128
8-24	Timing of INTUSBRE Generation	129
8-25	Flowchart of Transmit/Receive Pointer Operation	141
8-26	Flowchart of Receive Bank Switching ID Detection Buffer Operation	148
8-27	Timing of Sync Detection/USBCLK Detector Operation	149
8-28	Timing of Sync Detection/USBCLK Generation Operation	149
8-29	Flowchart of Sync Detection/USBCLK Detector Operation	150
8-30	Timing of NRZI Encoder Operation	151

chanhvinh.com

LIST OF FIGURES (3/4)

Figure No.	Title	Page
8-31	Flow Chart of NRZI Encoder Operation	151
8-32	Timing of Bit Stuffing/Strip Controller Operation	152
8-33	Flow Chart of Bit Stuffing Control Operation	153
8-34	Flow Chart of Bit Strip Control Operation	154
9-1	Block Diagram of Serial Interface 10	157
9-2	Format of Serial Operation Mode Register 10	158
9-3	3-Wire Serial I/O Mode Timing	162
10-1	Block Diagram of Regulator and USB Driver/Receiver	163
11-1	Basic Configuration of Interrupt Function	166
11-2	Format of Interrupt Request Flag Register	168
11-3	Format of Interrupt Mask Flag Register	169
11-4	Format of External Interrupt Mode Register 0	169
11-5	Configuration of Program Status Word	170
11-6	Format of Key Return Mode Register 00	171
11-7	Block Diagram of Falling Edge Detector	171
11-8	Flowchart of Non-Maskable Interrupt Request Acknowledgment	173
11-9	Timing of Non-Maskable Interrupt Request Acknowledgment	173
11-10	Acknowledging Non-Maskable Interrupt Request	173
11-11	Interrupt Acknowledgment Program Algorithm	174
11-12	Timing of Interrupt Request Acknowledgment (Example of MOV A,r)	175
11-13	Timing of Interrupt Request Acknowledgment (When Interrupt Request Flag Is Generated at Last Clock of Instruction Execution)	175
11-14	Example of Multiplexed Interrupt Servicing	177
12-1	Format of Oscillation Stabilization Time Select Register	180
12-2	Releasing HALT Mode by Interrupt	182
12-3	Releasing HALT Mode by $\overline{\text{RESET}}$ Input	183
12-4	Releasing STOP Mode by Interrupt	185
12-5	Releasing STOP Mode by $\overline{\text{RESET}}$ Input	186
13-1	Block Diagram of Reset Function	187
13-2	Reset Timing by $\overline{\text{RESET}}$ Input	188
13-3	Reset Timing by Overflow in Watchdog Timer	188
13-4	Reset Timing by $\overline{\text{RESET}}$ Input in STOP Mode	188
14-1	Environment for Writing Program to Flash Memory	192
14-2	Communication Mode Selection Format	193

chanhvinh.com

LIST OF FIGURES (4/4)

Figure No.	Title	Page
14-3	Example of Connection with Dedicated Flash Programmer	194
14-4	V_{PP} Pin Connection Example	196
14-5	Signal Conflict (Input Pin of Serial Interface)	197
14-6	Abnormal Operation of Other Device	197
14-7	Signal Conflict ($\overline{\text{RESET}}$ Pin)	198
14-8	Wiring Example for Flash Writing Adapter with 3-Wire Serial I/O	199
14-9	Wiring Example for Flash Writing Adapter with Pseudo-3-Wire Method	200
A-1	Development Tools	223
A-2	Distance Between In-Circuit Emulator and Conversion Adapter	228
A-3	Connection Condition of Target System (NP-H44GB-TQ)	229

chanhvinh.com

LIST OF TABLES (1/2)

Table No.	Title	Page
2-1	Type of Pin I/O Circuit Recommended Connection of Unused Pins	34
3-1	Vector Table	38
3-2	Special Function Register List	46
4-1	Functions of Ports	59
4-2	Configuration of Port	60
4-3	Port Mode Register and Output Latch Settings When Using Alternate Functions	71
5-1	Configuration of Clock Generator	74
5-2	Maximum Time Required for Switching CPU Clock	79
6-1	Interval Time of 8-Bit Timer 00	80
6-2	Interval Time of 8-Bit Timer/Event Counter 01	80
6-3	Square Wave Output Range of 8-Bit Timer/Event Counter 01	81
6-4	Configuration of 8-Bit Timer/Event Counters 00 and 01	81
6-5	Interval Time of 8-Bit Timer 00	86
6-6	Interval Time of 8-Bit Timer/Event Counter 01	86
6-7	Square-Wave Output Range of 8-Bit Timer/Event Counter 01	89
7-1	Inadvertent Loop Detection Time of Watchdog Timer	92
7-2	Interval Time	92
7-3	Configuration of Watchdog Timer	93
7-4	Inadvertent Loop Detection Time of Watchdog Timer	96
7-5	Interval Time of Interval Timer	97
8-1	Configuration of USB Function	99
8-2	Flag of RXSTAT After Reception of USB Reset Signal and Resume Signal	114
8-3	Conditions in Transmit Reservation	118
8-4	List of Sources of Interrupts from USB Function	128
9-1	Configuration of Serial Interface 10	156
9-2	Operating Mode Settings of Serial Interface 10	159
11-1	Interrupt Source List	165
11-2	Flags Corresponding to Interrupt Request Signals	167
11-3	Time from Generation of Maskable Interrupt Request to Servicing	174
12-1	HALT Mode Operation Status	181
12-2	Operation After Release of HALT Mode	183

chanhvinh.com

LIST OF TABLES (2/2)

Table No.	Title	Page
12-3	STOP Mode Operation Status	184
12-4	Operation After Release of STOP Mode	186
13-1	Hardware Status After Reset	189
14-1	Differences Between μPD78F9801 and Mask ROM Versions	191
14-2	Communication Mode List	193
14-3	Pin Connection List	195
15-1	Operand Identifiers and Description Methods	201
18-1	Surface Mounting Type Soldering Conditions	221

chanhvinh.com

CHAPTER 1 GENERAL

1.1 Features

- On-chip USB functions
  - Implements a USB (Universal Serial Bus) by connecting to Hub and Host.
  - Transfer speed: 1.5 Mbps (at 6.0 MHz operation with system clock)
- On-chip regulator
  - Controls the USB port voltage by using a bus power supply (V_{REG} = 3.3 ±0.3 V) dedicated to the USB driver/receiver.
- On-chip ROM and RAM
  - Internal ROM: 8 KB
    Flash memory (for μPD78F9801 only): 16 KB
  - Internal high-speed RAM: 256 bytes
- Variable minimum instruction execution time: From high-speed (0.33 μs) to low speed (1.33 μs) with the system clock operating at 6.0 MHz
- 31 I/O ports
- Two serial interface channels
  - USB function
  - 3-wire serial I/O mode
- Three timers:
  - 8-bit timer
  - 8-bit timer/event counter
  - Watchdog timer
- On-chip key return signal detector
- 12 vectored interrupt sources
- Power supply voltage: V_{DD} = 4.0 to 5.5 V
- Operating ambient temperature: T_A = –40 to +85°C (when the USB is not operating)
  T_A = 0 to +70°C (when the USB is operating)
★ T_A = 10 to 40°C (when the flash memory is written)

1.2 Applications

USB keyboards, etc.

★ 1.3 Ordering Information

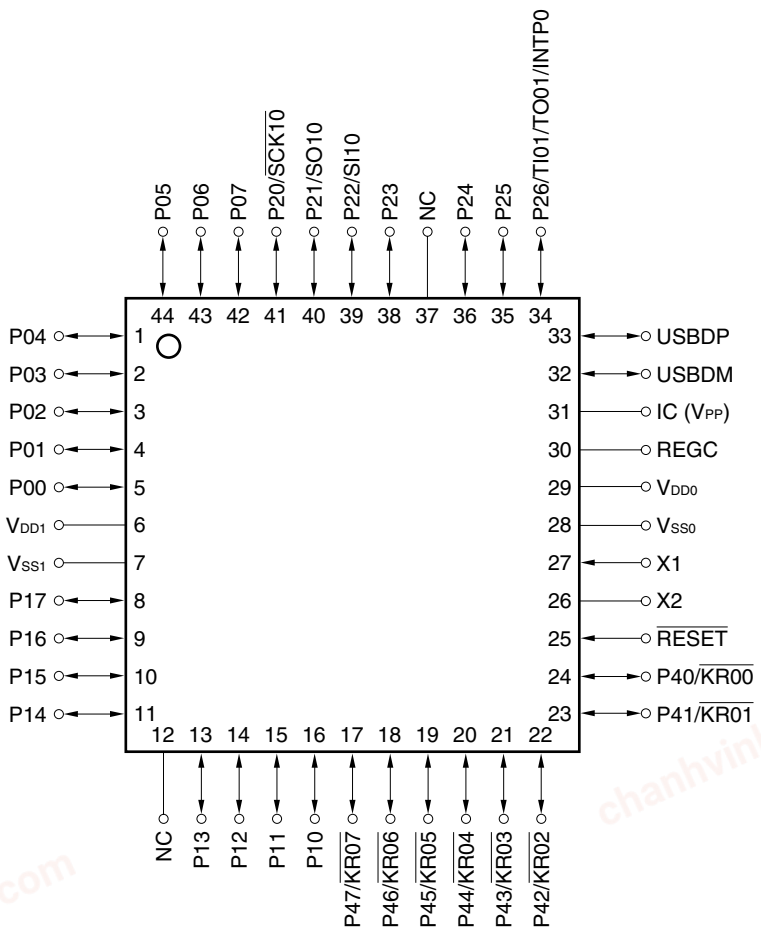
Part Number	Package	Internal ROM
μPD789800GB-×××-8ES	44-pin plastic LQFP (10 × 10)	Mask ROM
μPD78F9801GB-8ES	44-pin plastic LQFP (10 × 10)	Flash memory
μPD789800GB-×××-8ES-A	44-pin plastic LQFP (10 × 10)	Mask ROM
μPD78F9801GB-8ES-A	44-pin plastic LQFP (10 × 10)	Flash memory

Remarks 1. Products that have the part numbers suffixed by "-A" are lead-free products.
2. ××× indicates ROM code suffix.

chanhvinh.com

1.4 Pin Configuration (Top View)

★ • 44-pin plastic LQFP (10 × 10)
μPD789800GB-×××-8ES μPD78F9801GB-8ES
★ μPD789800GB-×××-8ES-A μPD78F9801GB-8ES-A



Cautions 1. Connect the IC pin directly to the V_{SS0} pin.
2. Directly connect the V_{PP} pin to the V_{SS0} pin in the normal operation mode.

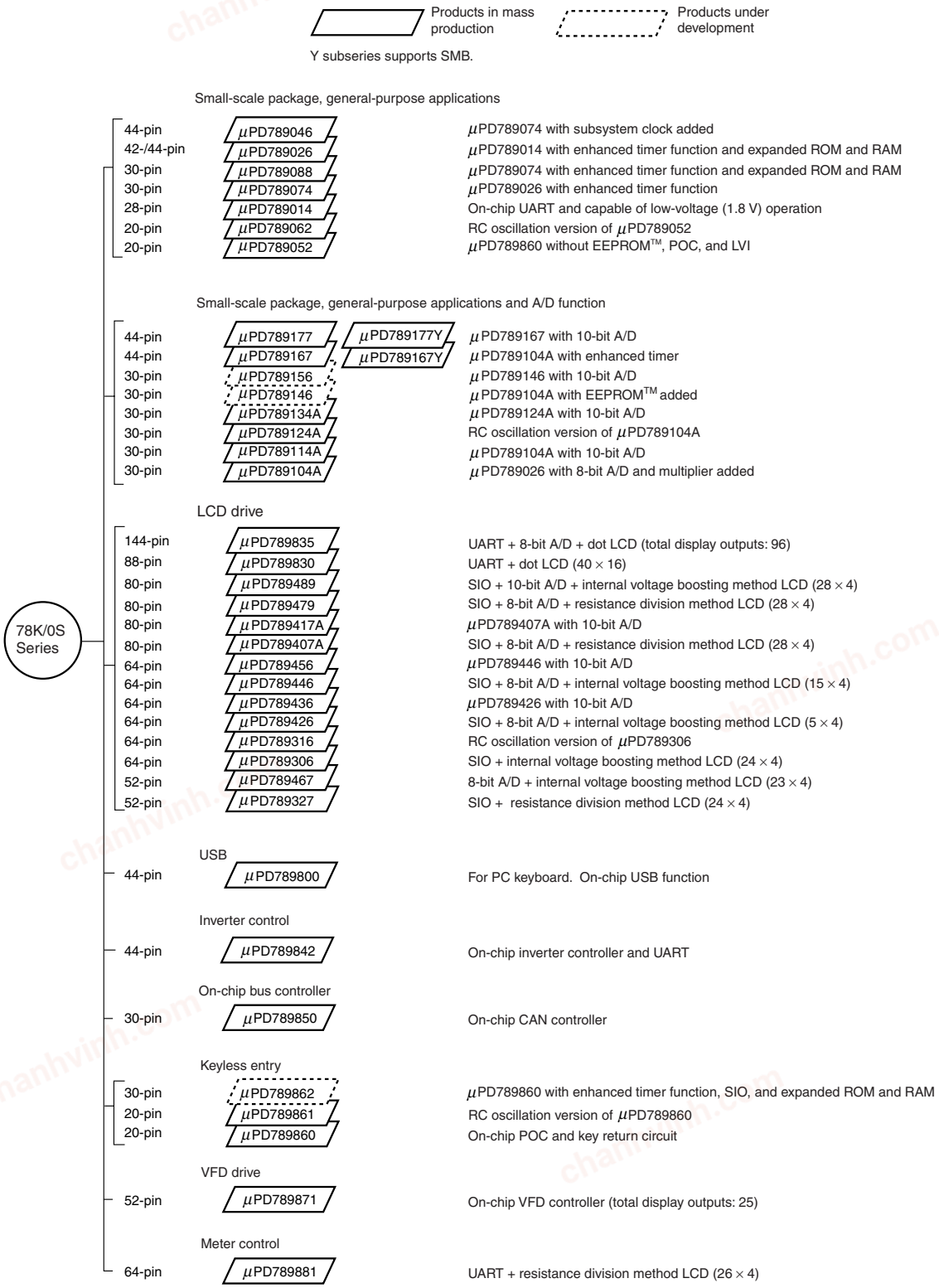
Remark The parenthesized values apply to the μPD78F9801.

IC:	Internally connected	SI10:	Serial data input
INTP0:	Interrupt from peripherals	SO10:	Serial data output
$\overline{KR00}$ to $\overline{KR07}$:	Key return	TI01:	Timer input
NC:	No connection	TO01:	Timer output
P00 to P07:	Port 0	USBDM, USBDP:	Universal serial bus data
P10 to P17:	Port 1	V_{DD0}:	Port power supply
P20 to P26:	Port 2	V_{DD1}:	Power supply
P40 to P47:	Port 4	V_{PP}:	Programming power supply
$\overline{RESET}$:	Reset	V_{SS0}:	Port ground
REGC:	Voltage regulator for USB function	V_{SS1}:	Ground
$\overline{SCK10}$:	Serial clock input/output	X1, X2:	Crystal

chanhvinh.com

★ 1.5 78K/0S Series Lineup

The products in the 78K/0S Series are listed below. The names enclosed in boxes are subseries names.



Remark VFD (Vacuum Fluorescent Display) is referred to as FIP™ (Fluorescent Indicator Panel) in some documents, but the functions of the two are same.

chanhvinh.com

The major differences between subseries are shown below.

Series for General-Purpose and LCD Drive

Subseries \ Function		ROM Capacity (Bytes)	Timer				8-Bit A/D	10-Bit A/D	Serial Interface	I/O	V_{DD} MIN.Value	Remarks
			8-Bit	16-Bit	Watch	WDT						
Small-scale package, general-purpose applications	μPD789046	16 K	1 ch	1 ch	1 ch	1 ch	–	–	1 ch (UART: 1ch)	34	1.8 V	–
	μPD789026	4 K to 16 K			–							
	μPD789088	16 K to 32 K	3 ch							24		
	μPD789074	2 K to 8 K	1 ch									
	μPD789014	2 K to 4 K	2 ch	–						22		
	μPD789062	4 K							–	14		RC-oscillation version
	μPD789052											–
Small-scale package, general-purpose applications + A/D converter	μPD789177	16 K to 24 K	3 ch	1 ch	1 ch	1ch	–	8 ch	1 ch (UART: 1ch)	31	1.8 V	–
	μPD789167						8 ch	–				
	μPD789156	8 K to 16 K	1 ch		–		–	4 ch		20		On-chip EEPROM
	μPD789146						4 ch	–				
	μPD789134A	2 K to 8 K					–	4 ch				RC-oscillation version
	μPD789124A						4 ch	–				
	μPD789114A						–	4 ch				–
	μPD789104A						4 ch	–				
LCD drive	μPD789835	24 K to 60 K	6 ch	–	1 ch	1 ch	3 ch	–	1 ch (UART: 1ch)	37	1.8 V[Note]	Dot LCD supported
	μPD789830	24 K	1 ch	1 ch			–			30	2.7 V	
	μPD789489	32 K to 48 K	3 ch					8 ch	2 ch (UART: 1ch)	45	1.8 V	–
	μPD789479	24 K to 48 K					8 ch	–				
	μPD789417A	12 K to 24 K					–	7 ch	1 ch (UART: 1ch)	43		
	μPD789407A						7 ch	–				
	μPD789456	12 K to 16 K	2 ch				–	6 ch		30		
	μPD789446						6 ch	–				
	μPD789436						–	6 ch		40		
	μPD789426						6 ch	–				
	μPD789316	8 K to 16 K					–		2 ch (UART: 1ch)	23		RC-oscillation version
	μPD789306											–
	μPD789467	4 K to 24 K		–			1 ch		–	18		
	μPD789327						–		1 ch	21		

Note Flash memory version: 3.0 V

chanhvinh.com

Series for ASSP

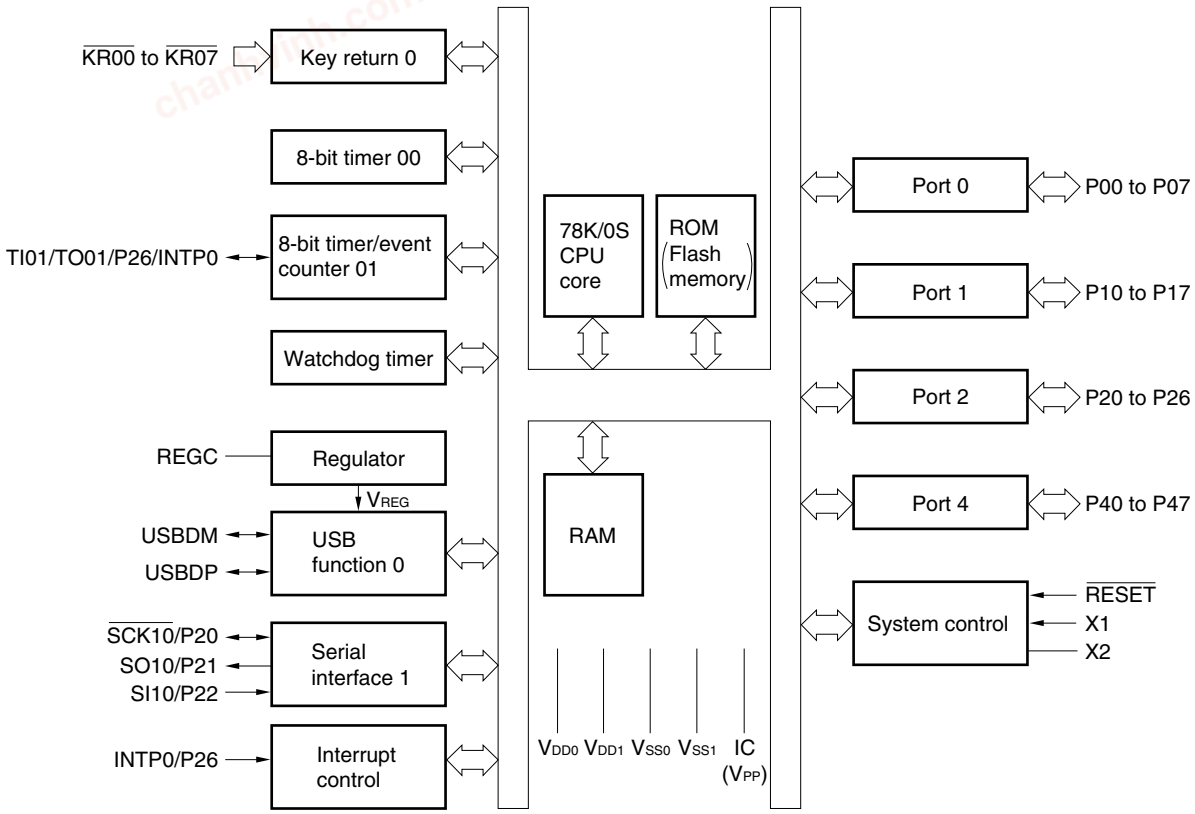
Subseries \ Function		ROM Capacity (Bytes)	Timer				8-Bit A/D	10-Bit A/D	Serial Interface	I/O	V_{DD} MIN.Value	Remarks
			8-Bit	16-Bit	Watch	WDT						
USB	μPD789800	8 K	2 ch	–	–	1 ch	–	–	2 ch (USB: 1ch)	31	4.0 V	–
Inverter control	μPD789842	8 K to 16 K	3 ch	**Note 1**	1 ch	1 ch	8 ch	–	1 ch (UART: 1ch)	30	4.0 V	–
On-chip bus controller	μPD789850	16 K	1 ch	1 ch	–	1 ch	4 ch	–	2 ch (UART: 1ch)	18	4.0 V	–
Keyless entry	μPD789861	4 K	2 ch	–	–	1 ch	–	–	–	14	1.8 V	RC-oscillation version, on-chip EEPROM
	μPD789860											On-chip EEPROM
	μPD789862	16 K	1 ch	2 ch					1 ch (UART: 1ch)	22		
VFD drive	μPD789871	4 K to 8 K	3 ch	–	1 ch	1 ch	–	–	1 ch	33	2.7 V	–
Meter control	μPD789881	16 K	2 ch	1 ch	–	1 ch	–	–	1 ch (UART: 1 ch)	28	2.7 V[Note 2]	–

Notes 1. 10-bit timer: 1 channel

2. Flash memory version: 3.0 V

chanhvinh.com

1.6 Block Diagram



Remark The parenthesized values apply to the μPD78F9801.

chanhvinh.com

1.7 Functions

Item \ Product		μPD789800	μPD78F9801
Internal memory	ROM	Mask ROM 8 KB	Flash memory 16 KB
	High-speed RAM	256 bytes	
Minimum instruction execution time		0.33 μs/1.33 μs (at 6.0 MHz operation with system clock)	
Instruction set		• 16-bit operation • Bit manipulation (set, reset, and test) etc.	
I/O ports		CMOS I/O 31 (Of the above COMS I/O ports, 18 ports can be switched to N-ch open-drain I/O ports.)	
Serial interface		• USB (Universal Serial Bus) function: 1 channel • Three-wired serial I/O mode: 1 channel	
Timer		• 8-bit timer: 1 channel • 8-bit timer/event counter: 1 channel • Watchdog timer: 1 channel	
Regulator		Incorporated (V_{REG} = 3.3 ±0.3 V)	
Vector interrupt sources	Maskable	Internal: 9, external: 2	
	Non-maskable	Internal: 1	
Power supply voltage		V_{DD} = 4.0 to 5.5 V	
Operating ambient temperature		• T_A = –40 to +85°C (when the USB is not operating) • T_A = 0 to +70°C (when the USB is operating) • T_A = 10 to 40°C (when a flash memory is written)	
Package		44-pin plastic LQFP (10 × 10)	

★ An outline of the timer is shown below.

		8-Bit Timer 00	8-Bit Timer/ Event Counter 01	Watchdog Timer
Operation mode	Interval timer	1 channel	1 channel	1 channel[Note]
	External event counter	–	1 channel	–
Function	Timer outputs	–	1 output	–
	Square-wave outputs	–	1 output	–
	Capture	–	–	–
	Interrupt sources	1	1	2

Note The watchdog timer has watchdog timer and interval timer functions. However, use the watchdog timer by selecting either the watchdog timer function or interval timer function.

chanhvinh.com

CHAPTER 2 PIN FUNCTIONS

2.1 List of Pin Functions

(1) Port pins

Pin Name	I/O	Function	After Reset	Alternate Function
P00 to P07	I/O	Port 0 8-bit I/O port Input/output can be specified in 1-bit units. When used as an input port, use of on-chip pull-up resistors can be specified by pull-up resistor option register 0 (PU0). When used as an output port, CMOS output or N-ch open-drain output can be specified in 8-bit units by port output mode register 0 (POM0).	Input	—
P10 to P17	I/O	Port 1 8-bit I/O port Input/output can be specified in 1-bit units. When used as an input port, use of on-chip pull-up resistors can be specified by pull-up resistor option register 0 (PU0). When used as an output port, CMOS output or N-ch open-drain output can be specified in 8-bit units by port output mode register 0 (POM0).	Input	—
P20	I/O	Port 2 7-bit I/O port Input/output can be specified in 1-bit units. When used as an input port, use of on-chip pull-up resistors can be specified by pull-up resistor option register 0 (PU0). When P25 or P26 is used as an output port, CMOS output or N-ch open-drain output can be specified in 1-bit units by port output mode register 1 (POM1).	Input	$\overline{SCK10}$
P21				SO10
P22				SI10
P23 to P25				—
P26				INTP0/TI01/TO01
P40 to P47	I/O	Port 4 8-bit I/O port Input/output can be specified in 1-bit units. When used as an input port, use of on-chip pull-up resistors can be specified by pull-up resistor option register 0 (PU0).	Input	$\overline{KR00}$ to $\overline{KR07}$

chanhvinh.com

(2) Non-port pins

Pin Name	I/O	Function	After Reset	Alternate Function
INTP0	Input	External interrupt request input for which valid edge (rising and/or falling edge) can be specified	Input	P26/TI01/TO01
$\overline{KR00}$ to $\overline{KR07}$	Input	Input for detecting key return signals	Input	P40 to P47
NC	—	No connection. Can be left open.	—	—
★ REGC	—	Internally generated power supply for driving USB driver/receiver. Connect this pin to V_{SS} via a 22 μF capacitor.	—	—
$\overline{RESET}$	Input	System reset input	Input	—
$\overline{SCK10}$	I/O	Serial clock input/output for serial interface	Input	P20
SI10	Input	Serial data input for serial interface	Input	P22
SO10	Output	Serial data output for serial interface	Input	P21
TI01	Input	External count clock input to 8-bit timer TM01	Input	P26/INTP0/TO01
TO01	Output	Output from 8-bit timer TM01	Input	P26/INTP0/TI01
USBDM	I/O	Serial data input/output (negative side) for USB function. The pull-up resistor (1.5 kΩ) for the USBDM pin must be connected to the REGC pin.	Input	—
USBDP	I/O	Serial data input/output (positive side) for USB function	Input	—
V_{DD0}	—	Positive power supply for ports	—	—
V_{DD1}	—	Positive power supply for circuits other than ports	—	—
V_{SS0}	—	Ground potential for ports	—	—
V_{SS1}	—	Ground potential for circuits other than ports	—	—
X1	Input	Crystal resonator connection to for system clock oscillator	Input	—
X2	—		—	—
IC	—	Internally connected directly to V_{SS0}	—	—
V_{PP}	—	Sets flash memory programming mode. Apply high voltage when a program is written or verified.	—	—

chanhvinh.com

2.2 Pin Functions

2.2.1 P00 to P07 (Port 0)

These pins constitute an 8-bit I/O port and can be set to the input or output port mode in 1-bit units by using port mode register 0 (PM0). When these pins are used as an input port, an on-chip pull-up resistor can be used by setting pull-up resistor option register 0 (PU0). When these pins are used as an output port, CMOS output or N-ch open-drain output can be specified in 8-bit units by setting port output mode register 0 (POM0).

2.2.2 P10 to P17 (Port 1)

These pins constitute an 8-bit I/O port. Port 1 can be set to the input or output mode in 1-bit units by using port mode register 1 (PM1). When the port is used as an input port, an on-chip pull-up resistor can be used by setting pull-up resistor option register 0 (PU0). When these pins are used as an output port, CMOS output or N-ch open-drain output can be specified in 8-bit units by setting port output mode register 0 (POM0).

2.2.3 P20 to P26 (Port 2)

These pins constitute a 7-bit I/O port. In addition, these pins function as data I/O, and clock I/O to and from the serial interface, external interrupt input, and timer I/O.

Port 2 can be specified in the following operation modes in 1-bit units.

(1) Port mode

In the port mode, P20 to P26 function as a 7-bit I/O port. Port 2 can be set to the input or output mode in 1-bit units by using port mode register 2 (PM2). When the port is used as an input port, an on-chip pull-up resistor can be used by setting pull-up resistor option register 0 (PU0). When P25 or P26 is used as an output port, CMOS output or N-ch open-drain output can be specified by setting in 1-bit units port output mode register 1 (POM1).

(2) Control mode

In this mode, P20 to P26 function as the data I/O and the clock I/O to and from the serial interface.

(a) SI10, SO10

These are the serial data I/O pins of the serial interface.

(b) $\overline{\text{SCK10}}$

This is the serial clock I/O pin of the serial interface.

(c) TI01

This is the external clock input pin for the 8-bit timer/event counter.

(d) TO01

This is the output pin of the 8-bit timer.

(e) INTP0

This is an external interrupt input pin for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

Caution When using P20 to P26 as serial interface pins, the I/O mode and output latch must be set according to the functions to be used. For setting details, see Table 9-2.

chanhvinh.com

2.2.4 P40 to P47 (Port 4)

These pins constitute an 8-bit I/O port. In addition, they also function as key return signal detection pins.
The following operation modes can be specified in 1-bit units.

(1) Port mode

In this mode, port 4 functions as an 8-bit I/O port. Port 4 can be set to the input or output mode in 1-bit units by using port mode register 4 (PM4). When used as an input port an on-chip pull-up resistor can be used by setting pull-up resistor option register 0 (PU0).

(2) Control mode

In this mode, the pins function as key return signal detection pins ($\overline{KR00}$ to $\overline{KR07}$).

2.2.5 $\overline{RESET}$

This pin inputs an active-low system reset signal.

2.2.6 X1, X2

These pins are used to connect a crystal resonator for system clock oscillation.
To supply an external clock, input the clock to X1 and input the inverted signal to X2.

2.2.7 REGC

★ This pin is a power supply pin for driving a USB driver/receiver generated internally. Connect this pin to the V_{SS} pin via 22 μF capacitor.

2.2.8 USBDM

This pin (negative side) inputs or outputs serial data for the USB function.

2.2.9 USBDP

This pin (positive side) inputs or outputs serial data for the USB function.

2.2.10 V_{DD0}, V_{DD1}

These pins are positive power supply pins.

2.2.11 V_{SS0}, V_{SS1}

These pins are ground pins.

chanhvinh.com

2.2.12 V_{PP} (μPD78F9801 only)

A high voltage should be applied to this pin when the flash memory programming mode is set and when the program is written or verified.

★ Handle this pin in either of the following ways.

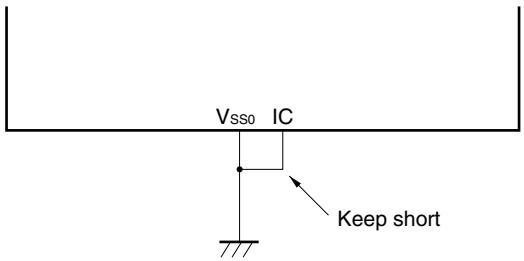
- Independently connect a 10 kΩ pull-down resistor.
- Switch this pin to be directly connected to the dedicated flash programmer in programming mode or to V_{SS0} in normal operation mode using a jumper on the board.

2.2.13 IC (mask ROM version only)

The IC (Internally Connected) pin is used to set the μPD789800 Subseries in the test mode before shipment. In the normal operation mode, directly connect this pin to the V_{SS0} pin with as short a wiring length as possible.

If a potential difference is generated between the IC pin and V_{SS0} pin due to a long wiring length between the IC pin and V_{SS0} pin or an external noise superimposed on the IC pin, a user program may not run correctly.

- Directly connect the IC pin to the V_{SS0} pin.



chanhvinh.com

2.3 Pin I/O Circuits and Recommended Connection of Unused Pins

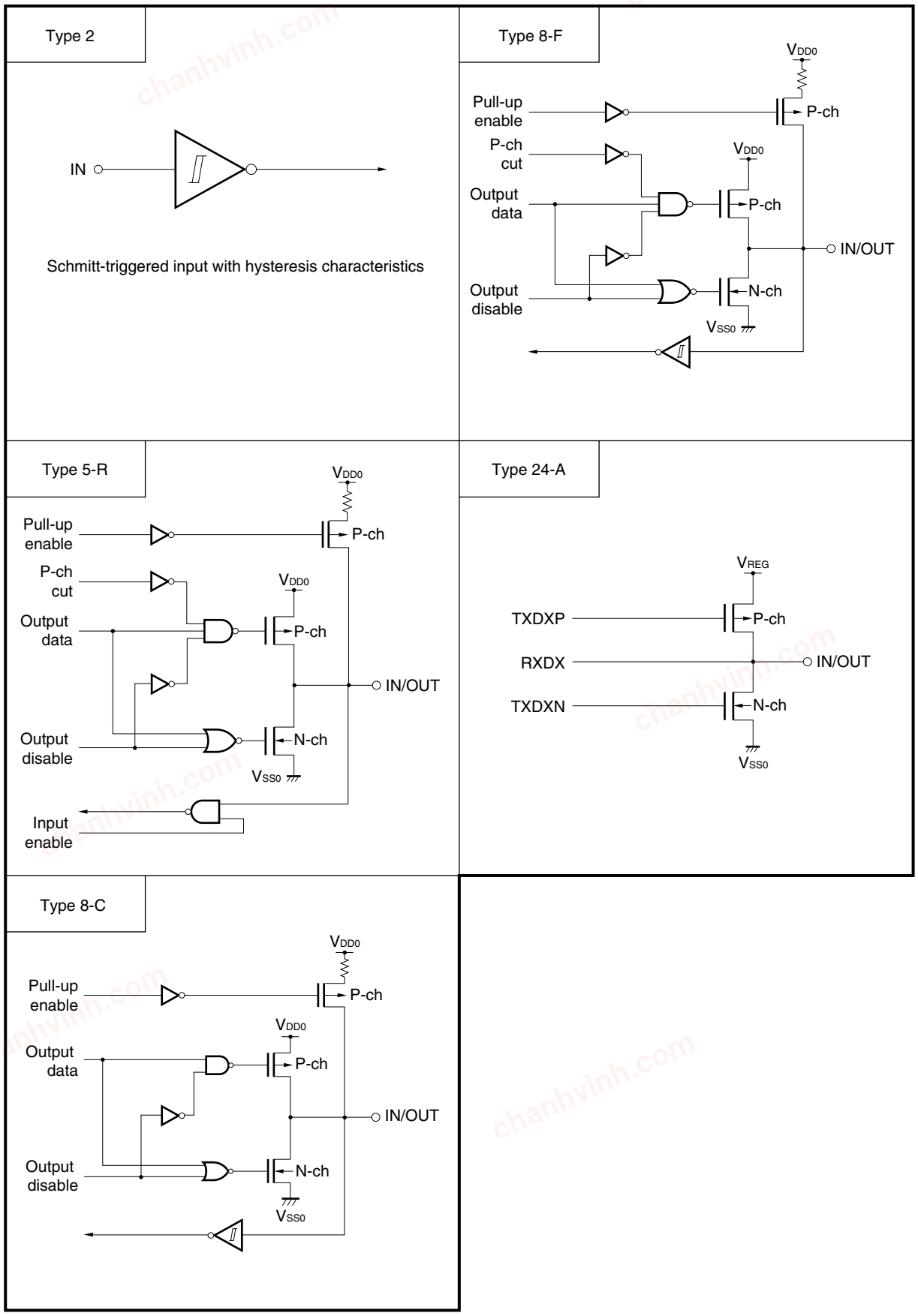
Table 2-1 lists the types of I/O circuits for each pin and explains how unused pins are handled.
Figure 2-1 shows the configuration of each type of I/O circuit.

Table 2-1. Type of Pin I/O Circuit Recommended Connection of Unused Pins

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00 to P07	5-R	I/O	Input: Independently connect to V_{DD0}, V_{DD1}, V_{SS0}, or V_{SS1} via a resistor. Output: Leave open.
P10 to P17			
P20/$\overline{SCK10}$	8-C		
P21/SO10			
P22/SI10			
P23, P24			
P25	8-F		
P26/INTP0/TI01/TO01			
P40/$\overline{KR00}$ to P47/$\overline{KR07}$	8-C		
USBDM	24-A		Connect to the REGC pin.
USBDP			Independently connect to V_{SS0} or V_{SS1} via a resistor.
$\overline{RESET}$	2	Input	—
NC	—	—	Leave open.
REGC	—	—	Connect to USBDM pin.
IC	—	—	Connect directly to V_{SS0}.
★ V_{PP}			Independently connect a 10 kΩ pull-down resistor, or connect directly to V_{SS0}.

chanhvinh.com

Figure 2-1. Pin I/O Circuits



chanhvinh.com

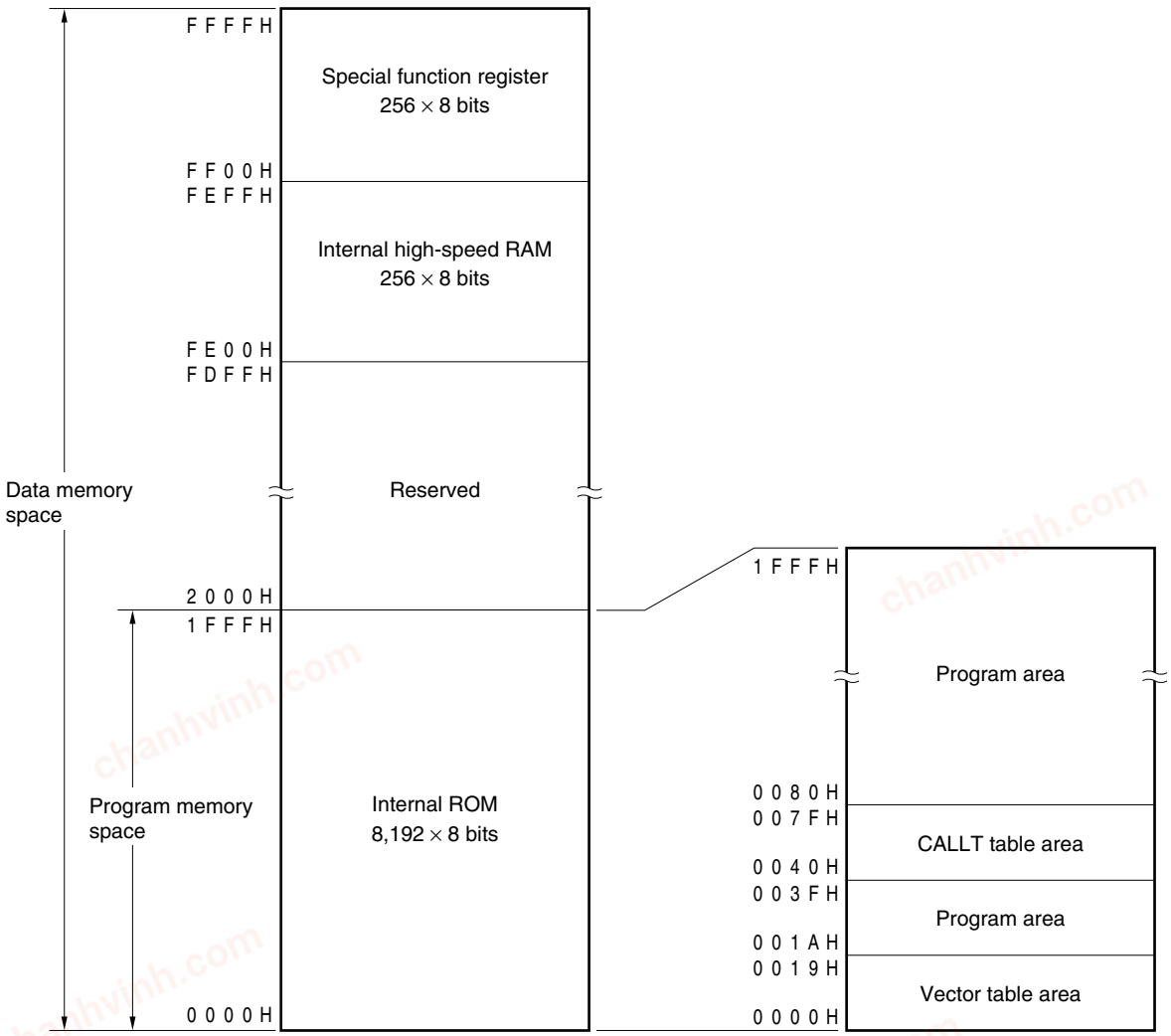
CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

The μPD789800 Subseries can access 64 KB of memory space.

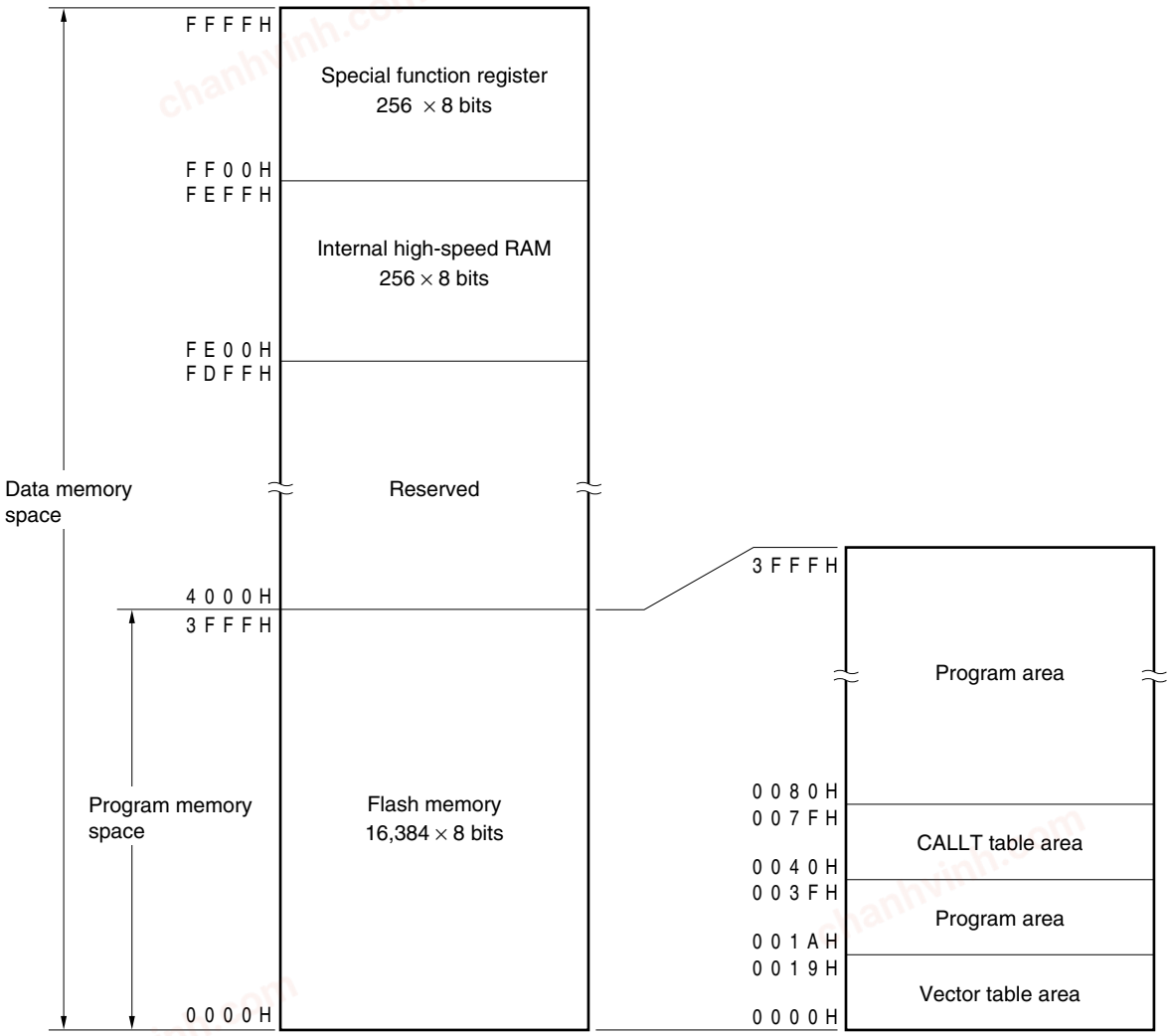
Figures 3-1 and 3-2 show the memory maps.

Figure 3-1. Memory Map (μPD789800)



chanhvinh.com

Figure 3-2. Memory Map (μPD78F9801)



★

chanhvinh.com

3.1.1 Internal program memory space

The internal program memory space stores programs and table data. This space is usually addressed by the program counter (PC).

The following areas are allocated to the internal program memory space.

(1) Vector table area

A 26-byte area of addresses 0000H to 0019H is reserved as a vector table area. This area stores program start addresses to be used when branching by $\overline{\text{RESET}}$ input or interrupt request generation. Of a 16-bit program address, the lower 8 bits are stored in an even address, and the higher 8 bits are stored in an odd address.

Table 3-1. Vector Table

Vector Table Address	Interrupt Request	Vector Table Address	Interrupt Request
0000H	$\overline{\text{RESET}}$ input	000EH	INTUSBRE
0004H	INTWDT	0010H	INTP0
0006H	INTUSBTM	0012H	INTCSI10
0008H	INTUSBRT	0014H	INTTM00
000AH	INTUSBRD	0016H	INTTM01
000CH	INTUSBST	0018H	INTKR00

(2) CALLT instruction table area

The subroutine entry address of a 1-byte call instruction (CALLT) can be stored in a 64-byte area of addresses 0040H to 007FH.

3.1.2 Internal data memory (internal high-speed RAM) space

An internal high-speed RAM is incorporated in the area between FE00H and FEFFH.

The internal high-speed RAM is also used as a stack.

3.1.3 Special function register (SFR) area

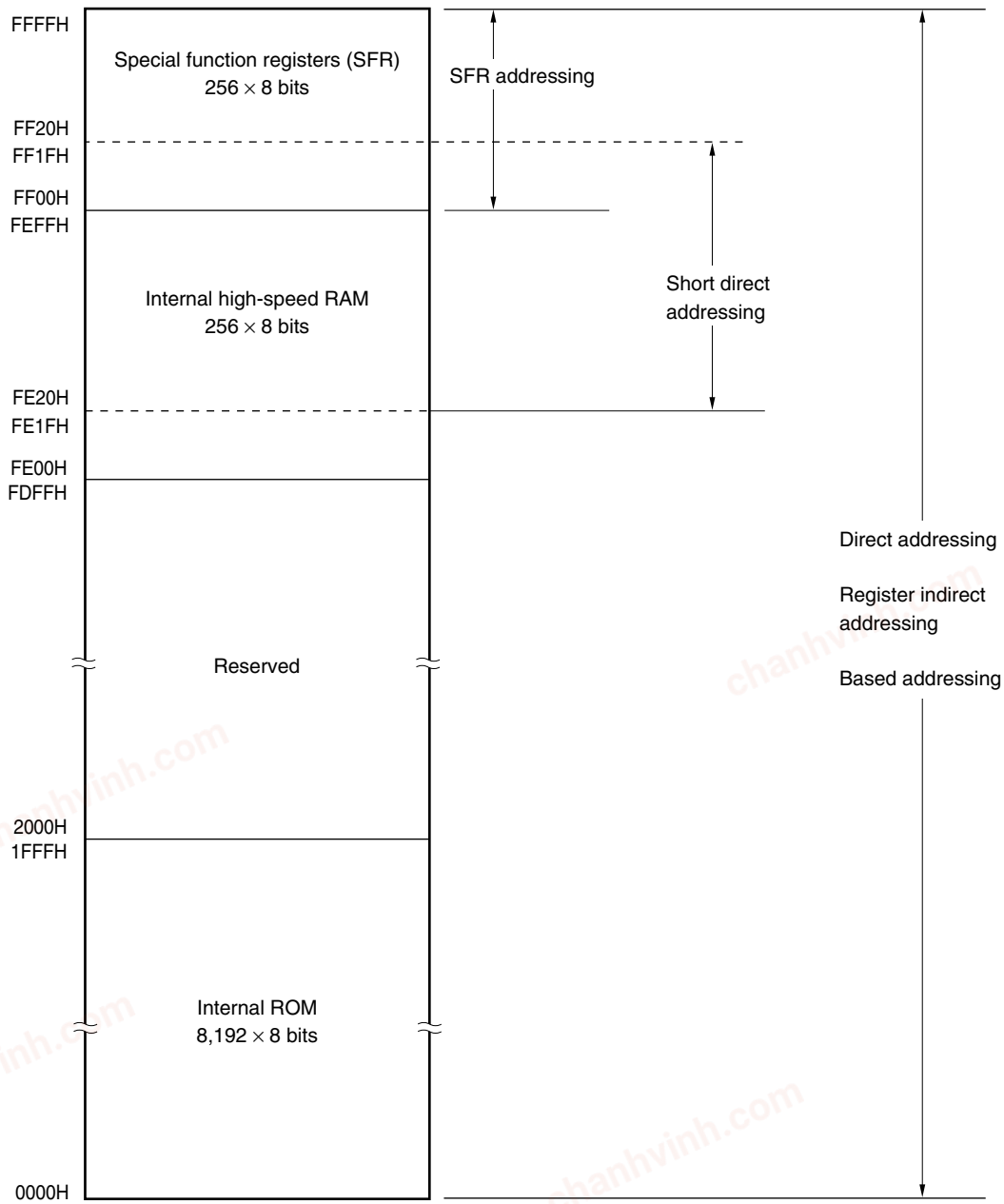
Special function registers (SFRs) of on-chip peripheral hardware are allocated to an area of FF00H to FFFFH (see **Table 3-2**).

chanhvinh.com

3.1.4 Data memory addressing

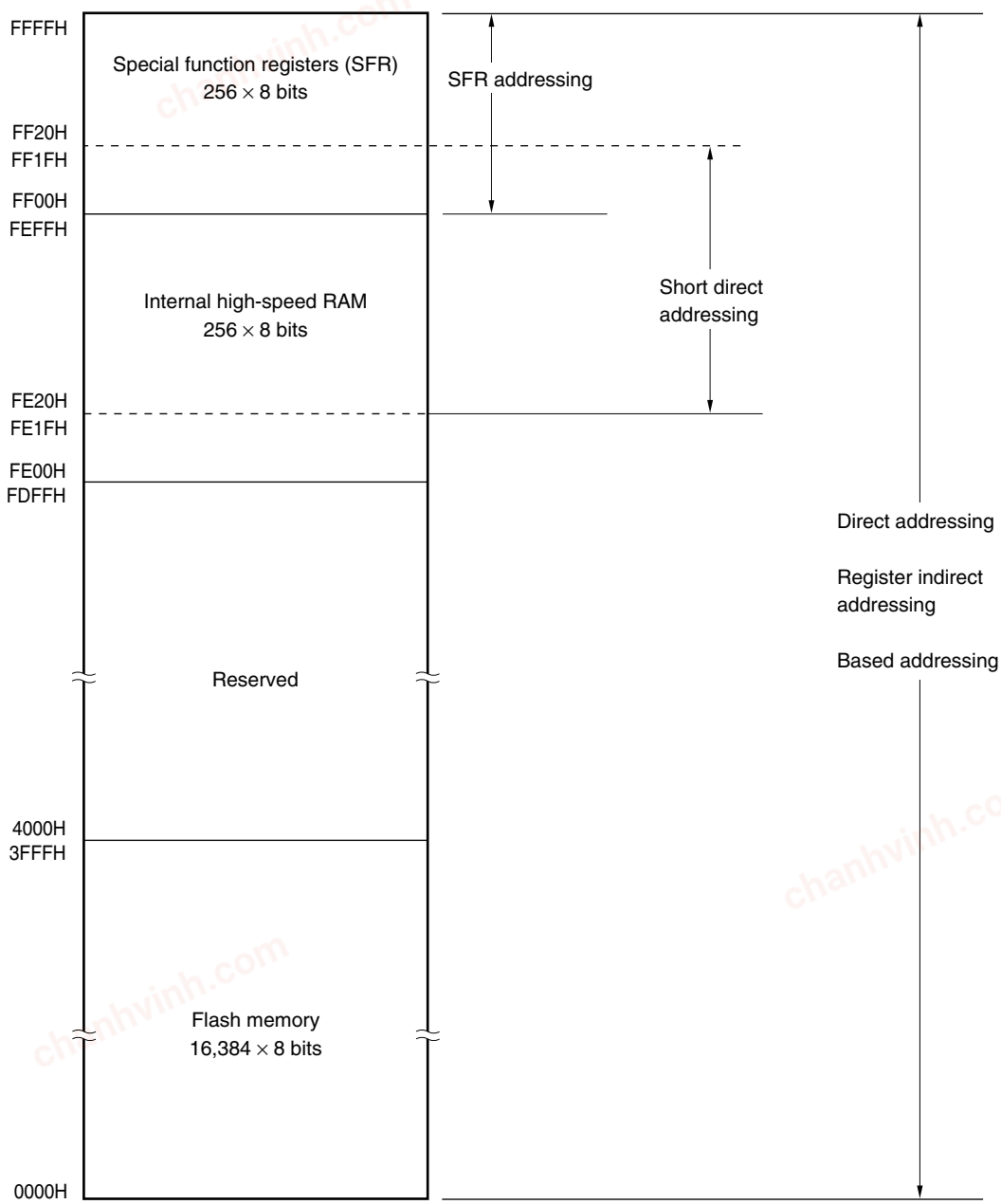
The μPD789800 Subseries provides a variety of addressing modes which take account of memory manipulability, etc. Especially at addresses corresponding to data memory area (FE00H to FFFFH), particular addressing modes are possible to meet the functions of the special function registers (SFR) and general-purpose registers. Figures 3-3 and 3-4 show the data memory addressing modes.

Figure 3-3. Data Memory Addressing (μPD789800)



chanhvinh.com

Figure 3-4. Data Memory Addressing (μPD78F9801)



chanhvinh.com

3.2 Processor Registers

The μPD789800 Subseries provides the following on-chip processor registers.

3.2.1 Control registers

The control registers contain special functions to control the program sequence, statuses and stack memory. A program counter, a program status word, and a stack pointer are the control registers.

(1) Program counter (PC)

The program counter is a 16-bit register which holds the address information of the next program to be executed. In normal operation, the PC is automatically incremented according to the number of bytes of the instruction to be fetched. When a branch instruction is executed, immediate data or register contents are set.

$\overline{\text{RESET}}$ input sets the program counter to the reset vector table values at addresses 0000H and 0001H.

Figure 3-5. Configuration of Program Counter

	15															0
PC	PC15	PC14	PC13	PC12	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0

(2) Program status word (PSW)

The program status word is an 8-bit register consisting of various flags to be set/reset by instruction execution. Program status word contents are automatically stacked upon interrupt request generation or PUSH PSW instruction execution and are automatically restored upon execution of the RETI and POP PSW instructions.

$\overline{\text{RESET}}$ input sets the PSW to 02H.

Figure 3-6. Configuration of Program Status Word

7							0
IE	Z	0	AC	0	0	1	CY

chanhvinh.com

(a) Interrupt enable flag (IE)

This flag controls interrupt request acknowledgment operations of the CPU.

When 0, the IE flag is set to the interrupt disabled status (DI), and interrupt requests other than non-maskable interrupts are all disabled.

When 1, the IE flag is set to the interrupt enabled status (EI). Interrupt request acknowledgment enable is controlled by the interrupt mask flag corresponding to each interrupt source.

The IE flag is reset (0) upon DI instruction execution or interrupt acknowledgment and is set (1) upon EI instruction execution.

(b) Zero flag (Z)

When the operation result is zero, this flag is set (1). It is reset (0) in all other cases.

(c) Auxiliary carry flag (AC)

If the operation result has a carry from bit 3 or a borrow at bit 3, this flag is set (1). It is reset (0) in all other cases.

(d) Carry flag (CY)

This flag stores overflow and underflow upon add/subtract instruction execution. It stores the shift-out value upon rotate instruction execution and functions as a bit accumulator during bit manipulation instruction execution.

chanhvinh.com

(3) Stack pointer (SP)

This is a 16-bit register that holds the start address of the memory stack area. Only the internal high-speed RAM area can be set as the stack area.

Figure 3-7. Configuration of Stack Pointer

	15															0
SP	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0

The SP is decremented ahead of write (save) to the stack memory and is incremented after read (restore) from the stack memory.

Each stack operation saves/restores data as shown in Figures 3-8 and 3-9.

Caution Since $\overline{\text{RESET}}$ input makes the SP contents undefined, be sure to initialize the SP before instruction execution.

Figure 3-8. Data to Be Saved to Stack Memory

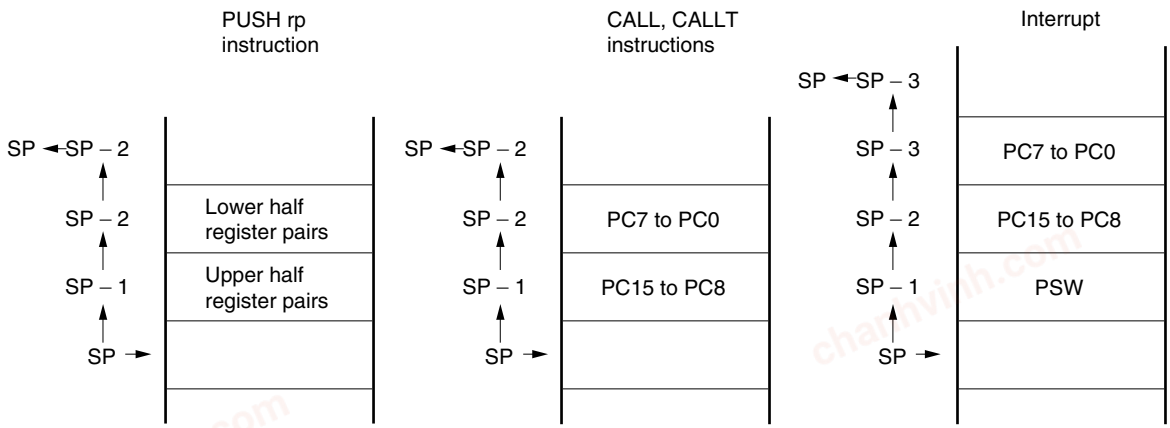
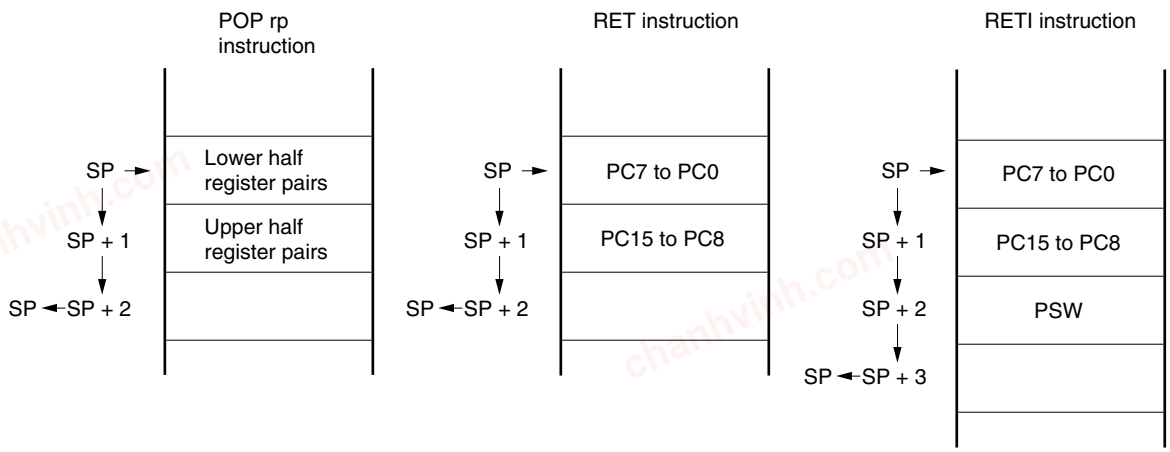


Figure 3-9. Data to Be Restored from Stack Memory



chanhvinh.com

3.2.2 General-purpose registers

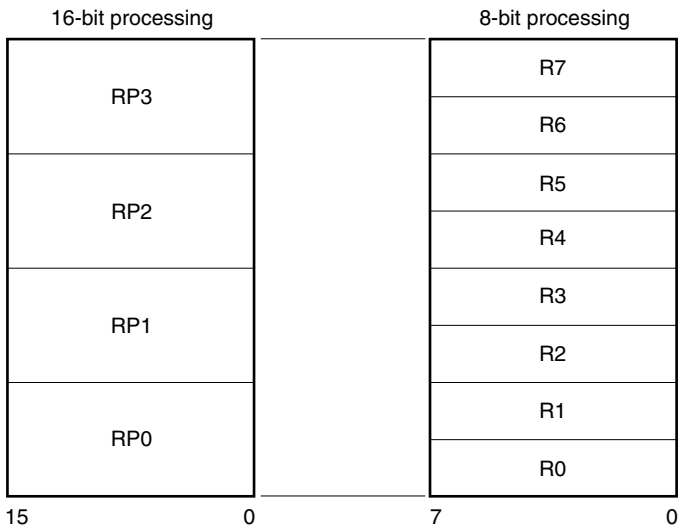
The general-purpose registers consist of eight 8-bit registers (X, A, C, B, E, D, L, and H).

In addition that each register can be used as an 8-bit register, two 8-bit registers in pairs can be used as a 16-bit register (AX, BC, DE, and HL).

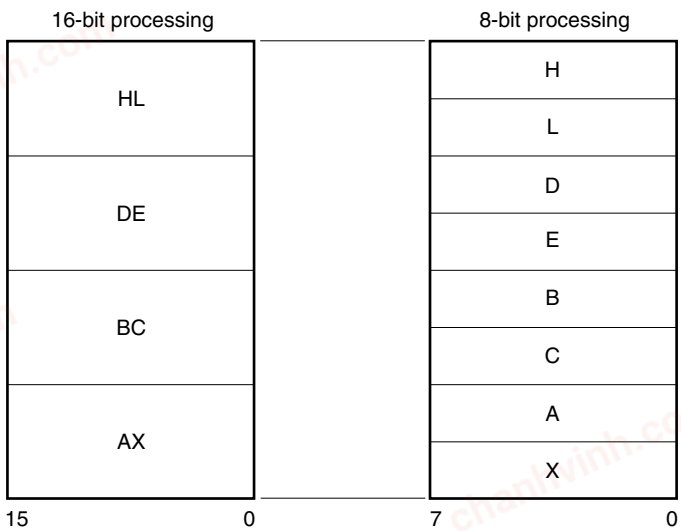
They can be described in terms of functional names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL) and absolute names (R0 to R7 and RP0 to RP3).

Figure 3-10. Configuration of General-Purpose Registers

(a) Absolute names



(b) Functional names



chanhvinh.com

3.2.3 Special function registers (SFRs)

Unlike general-purpose registers, each special function register has a special function.

The special function registers are allocated in the 256-byte area FF00H to FFFFH.

The special function registers can be manipulated, like the general-purpose registers, with operation, transfer, and bit manipulation instructions. Manipulatable bit units (1, 8, and 16) differ depending on the special function register type.

Each manipulation bit can be specified as follows.

- 1-bit manipulation
  Describe the symbol reserved by the assembler for the 1-bit manipulation instruction operand (sfr.bit). This manipulation can also be specified using address.

- 8-bit manipulation
  Describe the symbol reserved by the assembler for the 8-bit manipulation instruction operand (sfr). This manipulation can also be specified using an address.

- 16-bit manipulation
  Describe the symbol reserved by the assembler for the 16-bit manipulation instruction operand. When specifying an address, describe an even address.

Table 3-2 lists the special function registers. The meanings of the symbols in this table are as follows.

- Symbol
  Indicates the address of the implemented special function register. The symbols shown in this column are reserved words in the assembler, and have already been defined in the header file “sfrbit.h” in the C compiler. Therefore, these symbols can be used as instruction operands if an assembler or integrated debugger is used.

- R/W
  Indicates whether the special function register in question can be read or written.
  R/W: Read/write
  R: Read only
  W: Write only

- Bit units for manipulation
  Indicates the bit units (1, 8, 16) in which the special function register in question can be manipulated.

- After reset
  Indicates the status of the special function register when the $\overline{\text{RESET}}$ signal is input.

chanhvinh.com

Table 3-2. Special Function Register List (1/3)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1 Bit	8 Bits	16 Bits	
FF00H	Port 0	P0		R/W	√	√	—	00H
FF01H	Port 1	P1			√	√	—	
FF02H	Port 2	P2			√	√	—	
FF04H	Port 4	P4			√	√	—	
FF07H	Receive data PID	USBRD		R	—	√	—	
FF08H	Receive data address 0	USBR0	USBR10		—	√	√	Note Undefined
FF09H	Receive data address 1	USBR1			—	√		
FF0AH	Receive data address 2	USBR2	USBR32		—	√	√	Note
FF0BH	Receive data address 3	USBR3			—	√		
FF0CH	Receive data address 4	USBR4	USBR54		—	√	√	Note
FF0DH	Receive data address 5	USBR5			—	√		
FF0EH	Receive data address 6	USBR6	USBR76		—	√	√	Note
FF0FH	Receive data address 7	USBR7			—	√		
FF10H	Transmit/receive shift register 10	SIO10		R/W	—	√	—	
FF14H	Handshake packet transmit reservation register	HTX RSV	USB CON		√	√	√	Note 00H
FF15H	Data packet transmit reservation register	DTX RSV			√	√		
FF20H	Port mode register 0	PM0			√	√	—	FFH
FF21H	Port mode register 1	PM1			√	√	—	
FF22H	Port mode register 2	PM2			√	√	—	
FF24H	Port mode register 4	PM4			√	√	—	
FF30H	Port output mode register 0	POM0			√	√	—	00H
FF31H	Port output mode register 1	POM1			√	√	—	
FF42H	Timer clock select register 2	TCL2			—	√	—	
FF50H	8-bit compare register 00	CR00		W	—	√	—	Undefined
FF51H	8-bit timer counter 00	TM00		R	—	√	—	00H
FF53H	8-bit timer mode control register 00	TMC00		R/W	√	√	—	
FF54H	8-bit compare register 01	CR01		W	—	√	—	Undefined
FF55H	8-bit timer counter 01	TM01		R	—	√	—	00H
FF57H	8-bit timer mode control register 01	TMC01		R/W	√	√	—	
FF60H	Token PID compare register	TIDCMP		W	—	√	—	
FF61H	Token address compare register	ADRCMP			—	√	—	

Note 16-bit access is possible only in short direct addressing.

chanhvinh.com

Table 3-2. Special Function Register List (2/3)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FF62H	Token packet receive result store register	TRXRSL	R/W	√	√	—	00H
FF63H	Data/handshake PID compare register	DIDCMP	W	—	√	—	C3H
FF64H	Data/handshake packet receive byte number counter	DRXCON		—	√	—	18H
FF65H	Data/handshake packet receive result store register	DRXRSL	R/W	√	√	—	00H
FF66H	Data/handshake packet receive mode register	URXMOD		√	√	—	
FF67H	Packet receive status register	RXSTAT		—	√	—	
FF68H	Data packet transmit byte number counter 0	DTXCO0	W	—	√	—	20H
FF69H	Data packet transmit byte number counter 1	DTXCO1		—	√	—	30H
FF6AH	Remote wakeup control register	REMWUP	R/W	√	√	—	08H
FF6BH	Transmit/receive pointer	USBPOW	R	—	√	—	00H
FF6CH	USB timer start reservation control register	USBTCL	R/W	√	√	—	01H
FF6DH	USB receiver enable register	USBMOD		√	√	—	00H
FF72H	Serial operation mode register 10	CSIM10		√	√	—	
FFA1H	Transmit data PID bank 0	USBTD0	W	—	√	—	Undefined
FFA2H	Transmit data bank 0 address 00	USBT00		—	√	—	
FFA3H	Transmit data bank 0 address 01	USBT01		—	√	—	
FFA4H	Transmit data bank 0 address 02	USBT02		—	√	—	
FFA5H	Transmit data bank 0 address 03	USBT03		—	√	—	
FFA6H	Transmit data bank 0 address 04	USBT04		—	√	—	
FFA7H	Transmit data bank 0 address 05	USBT05		—	√	—	
FFA8H	Transmit data bank 0 address 06	USBT06		—	√	—	
FFA9H	Transmit data bank 0 address 07	USBT07		—	√	—	
FFABH	Transmit data PID bank 1	USBTD1		—	√	—	
FFACH	Transmit data bank 1 address 10	USBT10		—	√	—	
FFADH	Transmit data bank 1 address 11	USBT11		—	√	—	
FFAEH	Transmit data bank 1 address 12	USBT12		—	√	—	
FFAFH	Transmit data bank 1 address 13	USBT13		—	√	—	
FFB0H	Transmit data bank 1 address 14	USBT14		—	√	—	
FFB1H	Transmit data bank 1 address 15	USBT15		—	√	—	
FFB2H	Transmit data bank 1 address 16	USBT16		—	√	—	
FFB3H	Transmit data bank 1 address 17	USBT17		—	√	—	
FFB5H	Receive token PID	USBRTP	R	—	√	—	00H
FFB6H	Receive token address L	USBRAL		—	√	—	
FFB7H	Receive token address H	USBRAH		—	√	—	
FFE0H	Interrupt request flag register 0	IF0	R/W	√	√	—	
FFE1H	Interrupt request flag register 1	IF1		√	√	—	
FFE4H	Interrupt mask flag register 0	MK0		√	√	—	FFH
FFE5H	Interrupt mask flag register 1	MK1		√	√	—	

chanhvinh.com

Table 3-2. Special Function Register List (3/3)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FFECH	External interrupt mode register 0	INTM0	R/W	—	√	—	00H
FFF5H	Key return mode register 00	KRM00		√	√	—	
FFF7H	Pull-up resistor option register 0	PU0		√	√	—	
FFF9H	Watchdog timer mode register	WDTM		√	√	—	
FFFAH	Oscillation stabilization time select register	OSTS		—	√	—	04H
FFFBH	Processor clock control register	PCC		√	√	—	02H

chanhvinh.com

3.3 Instruction Address Addressing

An instruction address is determined by program counter (PC) contents. PC contents are normally incremented (+1 for each byte) automatically according to the number of bytes of an instruction to be fetched each time another instruction is executed. When a branch instruction is executed, the branch destination information is set to the PC and branched by the following addressing (for details of each instruction, refer to **78K/0S Series Instruction User's Manual (U11047E)**).

3.3.1 Relative addressing

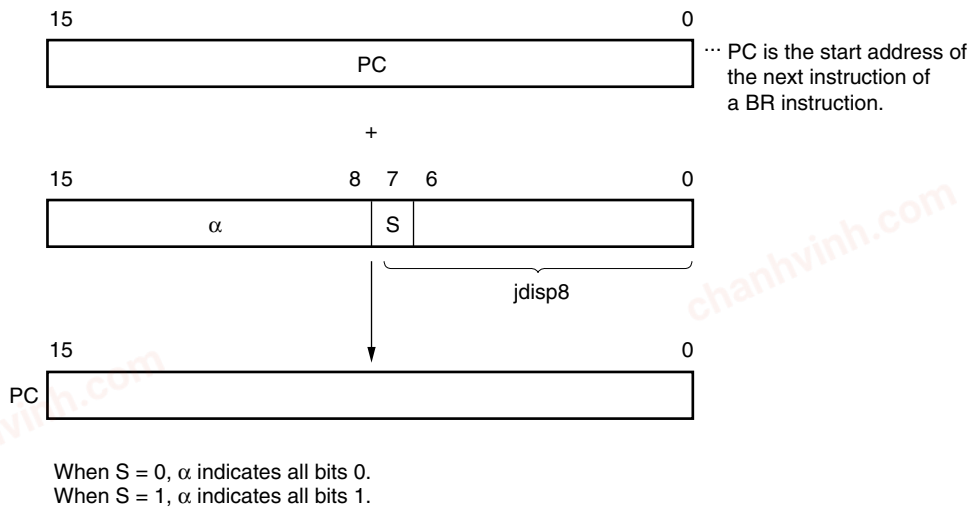
[Function]

The value obtained by adding 8-bit immediate data (displacement value: jdisp8) of an instruction code to the start address of the following instruction is transferred to the program counter (PC) and branched. The displacement value is treated as signed two's complement data (–128 to +127) and bit 7 becomes a sign bit.

This means that information is relatively branched to a location between –128 and +127, from the start address of the next instruction when relative addressing is used.

This function is carried out when the BR $addr16 instruction or a conditional branch instruction is executed.

[Illustration]



When S = 0, α indicates all bits 0.
When S = 1, α indicates all bits 1.

chanhvinh.com

3.3.2 Immediate addressing

[Function]

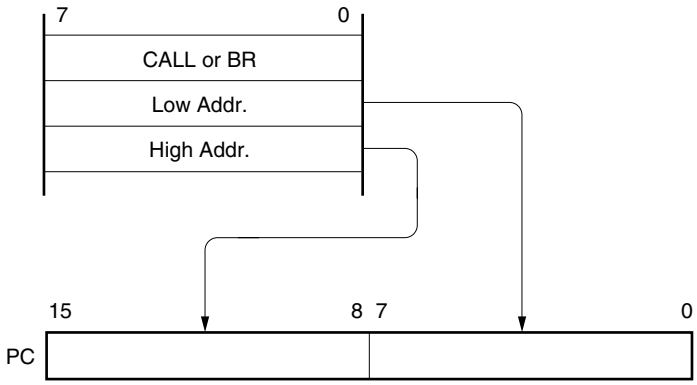
Immediate data in the instruction word is transferred to the program counter (PC) and branched.

This function is carried out when the CALL !addr16 or BR !addr16 instruction is executed.

The CALL !addr16 and BR !addr16 instructions can be branched to any location in the memory space.

[Illustration]

In case of CALL !addr16 and BR !addr16 instructions



chanhvinh.com

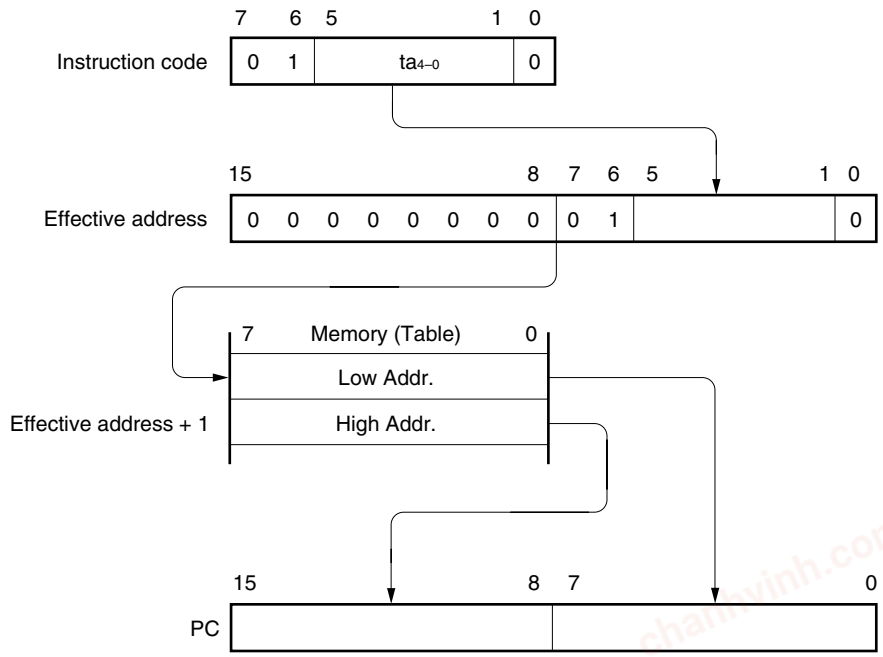
3.3.3 Table indirect addressing

[Function]

Table contents (branch destination address) of the particular location to be addressed by the lower-5-bit immediate data of an instruction code from bit 1 to bit 5 are transferred to the program counter (PC) and branched.

This function is carried out when the CALLT [addr5] instruction is executed. The instruction enables a branch to any location in the memory space by referring to the addresses stored in the memory table at 40H to 7FH.

[Illustration]



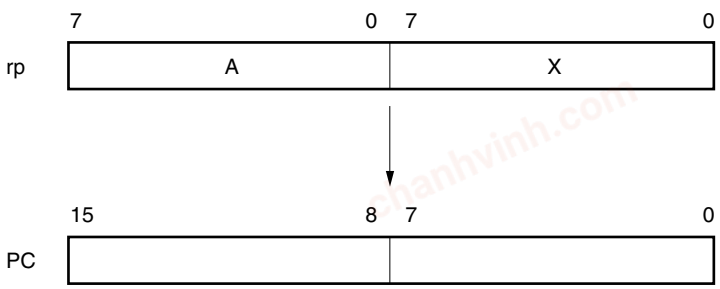
3.3.4 Register addressing

[Function]

Register pair (AX) contents to be specified with an instruction word are transferred to the program counter (PC) and branched.

This function is carried out when the BR AX instruction is executed.

[Illustration]



chanhvinh.com

3.4 Operand Address Addressing

The following methods are available to specify the register and memory (addressing) to undergo manipulation during instruction execution.

3.4.1 Direct addressing

[Function]

The memory indicated by immediate data in an instruction word is directly addressed.

[Operand format]

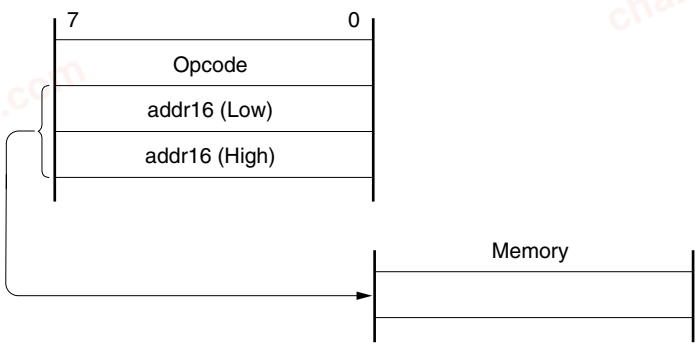
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

MOV A, !FE00H; When setting !addr16 to FE00H

Instruction code	0	0	1	0	1	0	0	1	Opcode
	0	0	0	0	0	0	0	0	00H
	1	1	1	1	1	1	1	0	FEH

[Illustration]



chanhvinh.com

3.4.2 Short direct addressing

[Function]

The memory to be manipulated in the fixed space is directly addressed with 8-bit data in an instruction word.

The fixed space is the 256-byte space FE20H to FF1FH where the addressing is applied. An internal high-speed RAM and special function registers (SFR) are mapped at FE20H to FEFFH and FF00H to FF1FH, respectively.

The SFR area (FF00H to FF1FH), where short direct addressing is applied, is a part of the whole SFR area. In this area, ports which are frequently accessed in a program and a compare register of the timer/event counter are mapped, and these SFRs can be manipulated with a small number of bytes and clocks.

When 8-bit immediate data is at 20H to FFH, bit 8 of an effective address is set to 0. When it is at 00H to 1FH, bit 8 is set to 1. See **[Illustration]** below.

[Operand format]

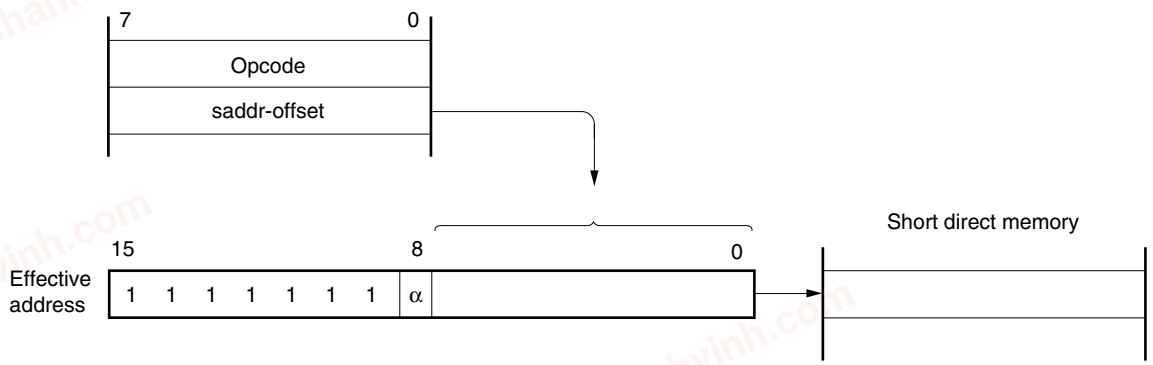
Identifier	Description
saddr	Label or FE20H to FF1FH immediate data
saddrp	Label or FE20H to FF1FH immediate data (even address only)

[Description example]

MOV FE30H, #50H; When setting saddr to FE30H and the immediate data to 50H

Instruction code	1	1	1	1	0	1	0	1	Opcode
	0	0	1	1	0	0	0	0	30H (saddr-offset)
	0	1	0	1	0	0	0	0	50H (Immediate data)

[Illustration]



When 8-bit immediate data is 20H to FFH, α = 0.
When 8-bit immediate data is 00H to 1FH, α = 1.

chanhvinh.com

3.4.3 Special function register (SFR) addressing

[Function]

The memory-mapped special function register (SFR) is addressed with 8-bit immediate data in an instruction word.

This addressing is applied to the 240-byte spaces FF00H to FFCFH and FFE0H to FFFFH. However, the SFR mapped at FF00H to FF1FH can be accessed with short direct addressing.

[Operand format]

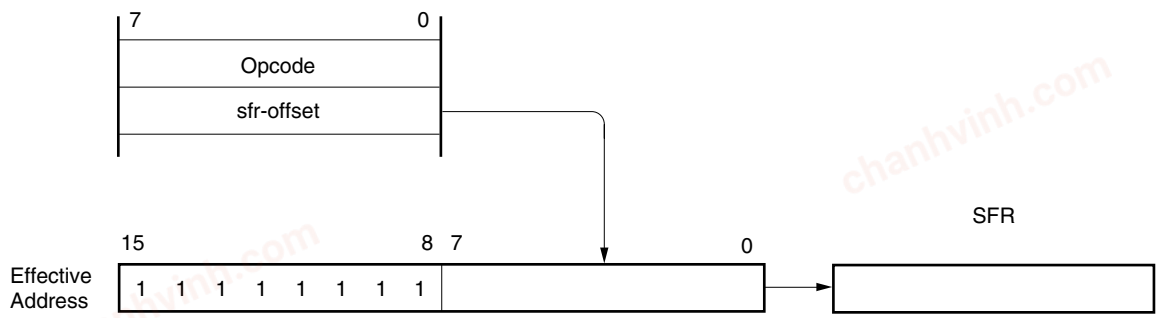
Identifier	Description
sfr	Special function register name

[Description example]

MOV PM0, A; When selecting PM0 for sfr

Instruction code	1	1	1	0	0	1	1	1
	0	0	1	0	0	0	0	0

[Illustration]



chanhvinh.com

3.4.4 Register addressing

[Function]

In the register addressing mode, general-purpose registers are accessed as operands. The general-purpose register to be accessed is specified by the register specification code or function name in the instruction code. Register addressing is carried out when an instruction with the following operand format is executed. When an 8-bit register is specified, one of the eight registers is specified with 3 bits in the instruction code.

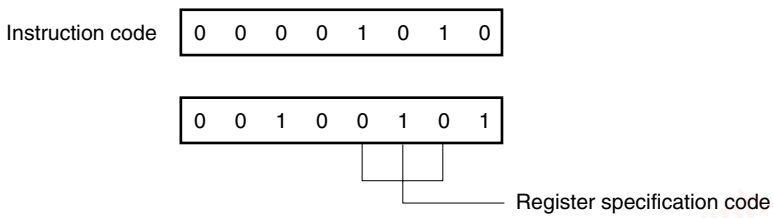
[Operand format]

Identifier	Description
r	X, A, C, B, E, D, L, H
rp	AX, BC, DE, HL

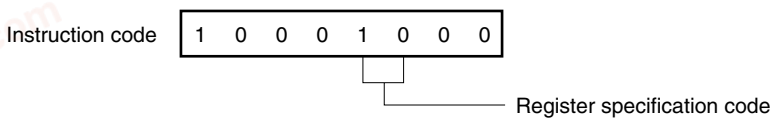
r and rp can be described by absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[Description example]

MOV A, C; When selecting the C register for r



INCW DE; When selecting the DE register pair for rp



chanhvinh.com

3.4.5 Register indirect addressing

[Function]

In the register indirect addressing mode, memory is manipulated according to the contents of a register pair specified as an operand. The register pair to be accessed is specified by the register pair specification code in an instruction code.

This addressing can be carried out for all the memory spaces.

[Operand format]

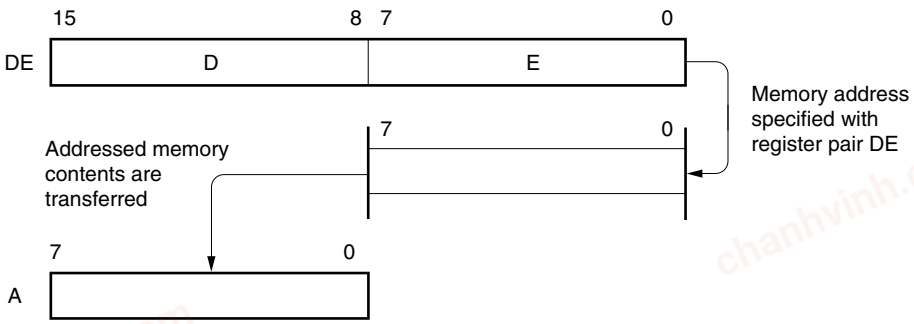
Identifier	Description
—	[DE], [HL]

[Description example]

MOV A, [DE]; When selecting register pair [DE]

Instruction code	0	0	1	0	1	0	1	1

[Illustration]



chanhvinh.com

3.4.6 Based addressing

[Function]

8-bit immediate data is added to the contents of the base register, that is, the HL register pair, and the sum is used to address the memory. Addition is performed by expanding the offset data as a positive number to 16 bits. A carry from the 16th bit is ignored. This addressing can be carried out for all the memory spaces.

[Operand format]

Identifier	Description
—	[HL+byte]

[Description example]

MOV A, [HL+10H]; When setting byte to 10H

Instruction code	0	0	1	0	1	1	0	1
	0	0	0	1	0	0	0	0

3.4.7 Stack addressing

[Function]

The stack area is indirectly addressed with the stack pointer (SP) contents.

This addressing method is automatically employed when the PUSH, POP, subroutine call, and RETURN instructions are executed or the register is saved/reset upon generation of an interrupt request.

Stack addressing can only be used to address the internal high-speed RAM area.

[Description example]

In the case of PUSH DE

Instruction code	1	0	1	0	1	0	1	0

chanhvinh.com

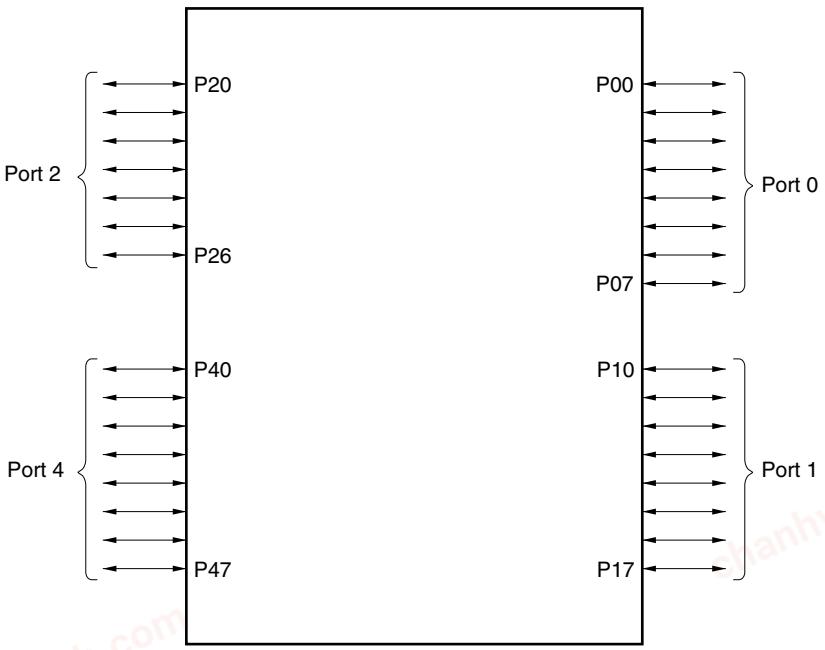
CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The μPD789800 Subseries provides the ports shown in Figure 4-1, enabling various methods of control.

Numerous other functions are provided that can be used in addition to the digital I/O port functions. For more information on these additional functions, see **CHAPTER 2 PIN FUNCTIONS**.

Figure 4-1. Port Types



chanhvinh.com

Table 4-1. Functions of Ports

Pin Name	I/O	Function	After Reset	Alternate Function
P00 to P07	I/O	Port 0 8-bit I/O port Input/output can be specified in 1-bit units. When used as an input port, use of on-chip pull-up resistors can be specified by pull-up resistor option register 0 (PU0). When used as an output port, CMOS output or N-ch open-drain output can be specified in 8-bit units by port output mode register 0 (POM0).	Input	—
P10 to P17	I/O	Port 1 8-bit I/O port Input/output can be specified in 1-bit units. When used as an input port, use of on-chip pull-up resistors can be specified by pull-up resistor option register 0 (PU0). When used as an output port, CMOS output or N-ch open-drain output can be specified in 8-bit units by port output mode register 0 (POM0).	Input	—
P20	I/O	Port 2 7-bit I/O port Input/output can be specified in 1-bit units. When used as an input port, use of on-chip pull-up resistors can be specified by pull-up resistor option register 0 (PU0). When P25 or P26 is used as an output port, CMOS output or N-ch open-drain output can be specified in 1-bit units by port output mode register 1 (POM1).	Input	$\overline{\text{SCK10}}$
P21				SO10
P22				SI10
P23				—
P24				—
P25				—
P26				INTP0/TI01/TO01
P40 to P47	I/O	Port 4 8-bit I/O port Input/output can be specified in 1-bit units. When used as an input port, use of on-chip pull-up resistors can be specified by pull-up resistor option register 0 (PU0).	Input	$\overline{\text{KR00}}$ to $\overline{\text{KR07}}$

chanhvinh.com

4.2 Port Configuration

Ports consists the following hardware.

Table 4-2. Configuration of Port

Parameter	Configuration
Control registers	Port mode register (PMm: m = 0 to 2, 4)
	Pull-up resistor option register (PU0)
	Port output mode register (POMm: m = 0, 1)
Ports	Total: 31 (N-ch open-drain output is specifiable for 18 ports.)
Pull-up resistors	Software control: 31

chanhvinh.com

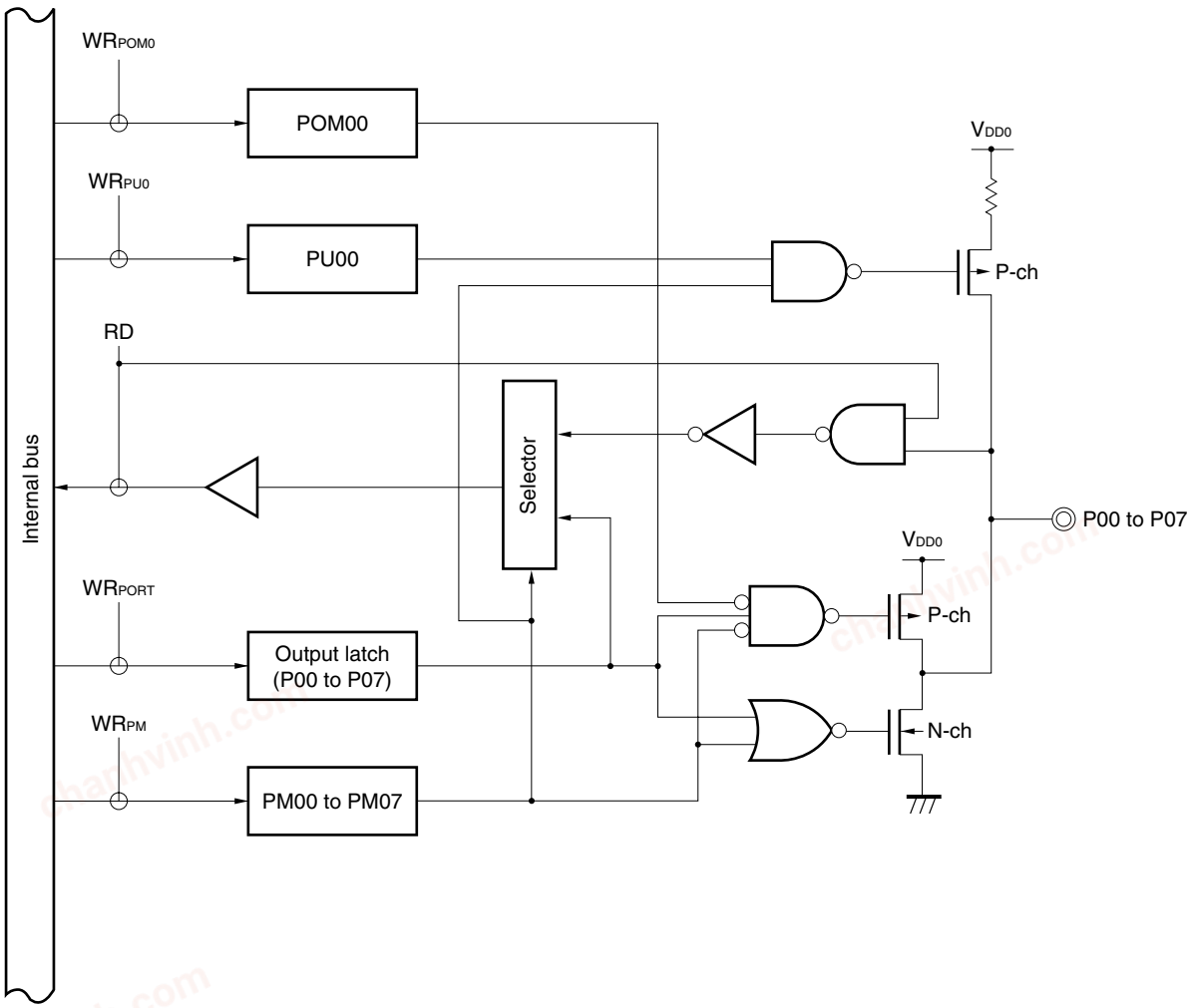
4.2.1 Port 0

This is an 8-bit I/O port with an output latch. Port 0 can be specified in the input or output mode in 1-bit units by using port mode register 0 (PM0). When the P00 to P07 pins are used as input port pins, on-chip pull-up resistors can be connected in 8-bit units by using pull-up resistor option register 0 (PU0). CMOS output or N-ch open-drain output can also be specified in 8-bit unit by using port output mode register 0 (POM0).

Port 0 is set in the input mode when the $\overline{\text{RESET}}$ signal is input.

Figure 4-2 shows a block diagram of port 0.

Figure 4-2. Block Diagram of P00 to P07



POM0: Port output mode register 0
PU0: Pull-up resistor option register 0
PM: Port mode register
RD: Port 0 read signal
WR: Port 0 write signal

chanhvinh.com

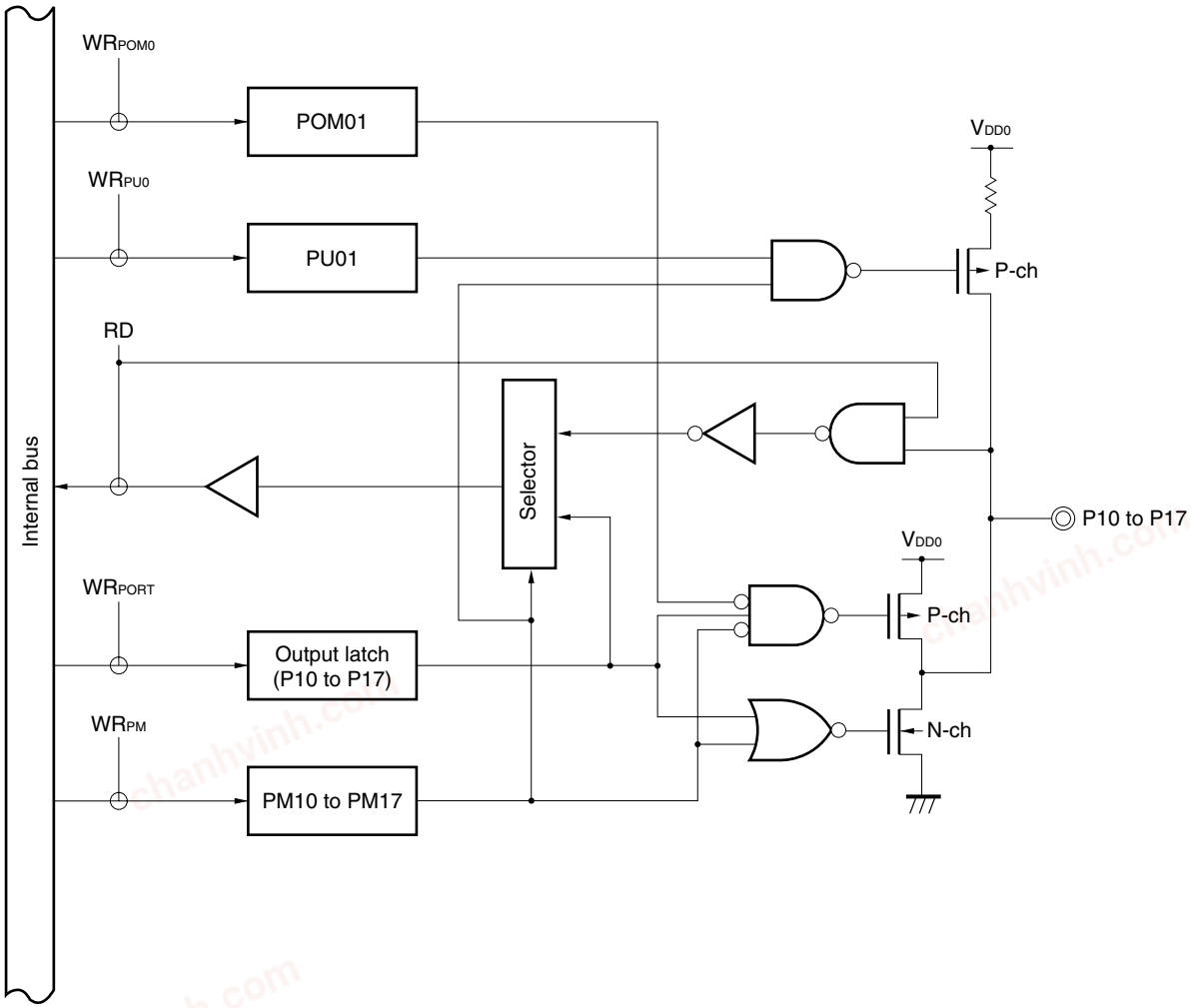
4.2.2 Port 1

This is an 8-bit I/O port with an output latch. Port 1 can be specified in the input or output mode in 1-bit units by using port mode register 1 (PM1). When the P10 to P17 pins are used as input port pins, on-chip pull-up resistors can be connected in 8-bit units by using pull-up resistor option register 0 (PU0). CMOS output or N-ch open-drain output can also be specified in 8-bit unit by using port output mode register 0 (POM0).

Port 0 is set in the input mode when the $\overline{\text{RESET}}$ signal is input.

Figure 4-3 shows a block diagram of port 1.

Figure 4-3. Block Diagram of P10 to P17



POM0: Port output mode register 0
PU0: Pull-up resistor option register 0
PM: Port mode register
RD: Port 1 read signal
WR: Port 1 write signal

chanhvinh.com

4.2.3 Port 2

This is a 7-bit I/O port with an output latch. Port 2 can be specified in the input or output mode in 1-bit units by using port mode register 2 (PM2). When using the P20 to P26 pins as input port pins, on-chip pull-up resistors can be connected in 7-bit units by using pull-up resistor option register 0 (PU0).

When P25 or P26 is used, CMOS output or N-ch open-drain output can be specified in 1-bit units by using port output mode register 1 (POM1).

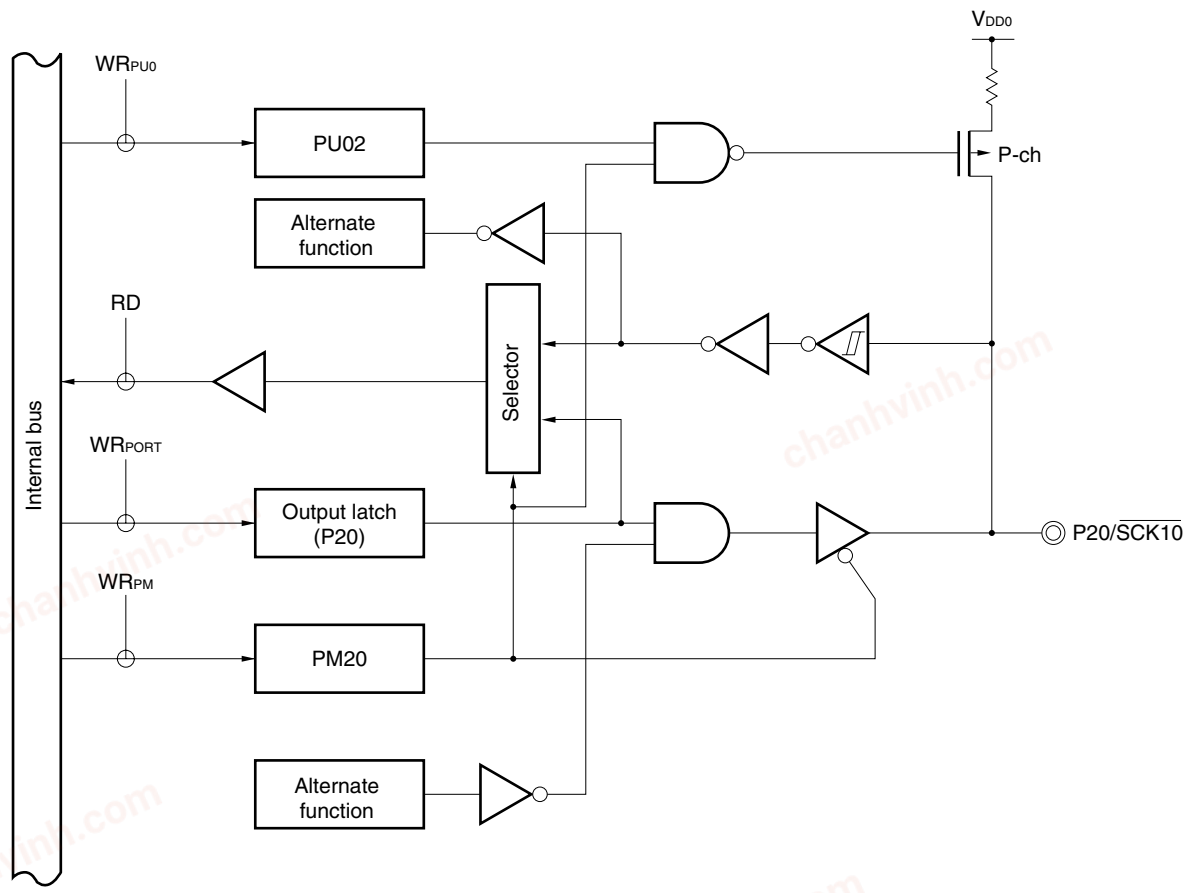
The port is also used as a data I/O and clock I/O to and from the serial interface, timer I/O, and external interrupt.

This port to set to the input mode when the $\overline{\text{RESET}}$ signal is input.

Figures 4-4 through 4-9 show block diagrams of port 2.

Caution When using the pins of port 2 as the serial interface, the I/O or output latch must be set according to the function to be used. For how to set the latches, see Table 9-2.

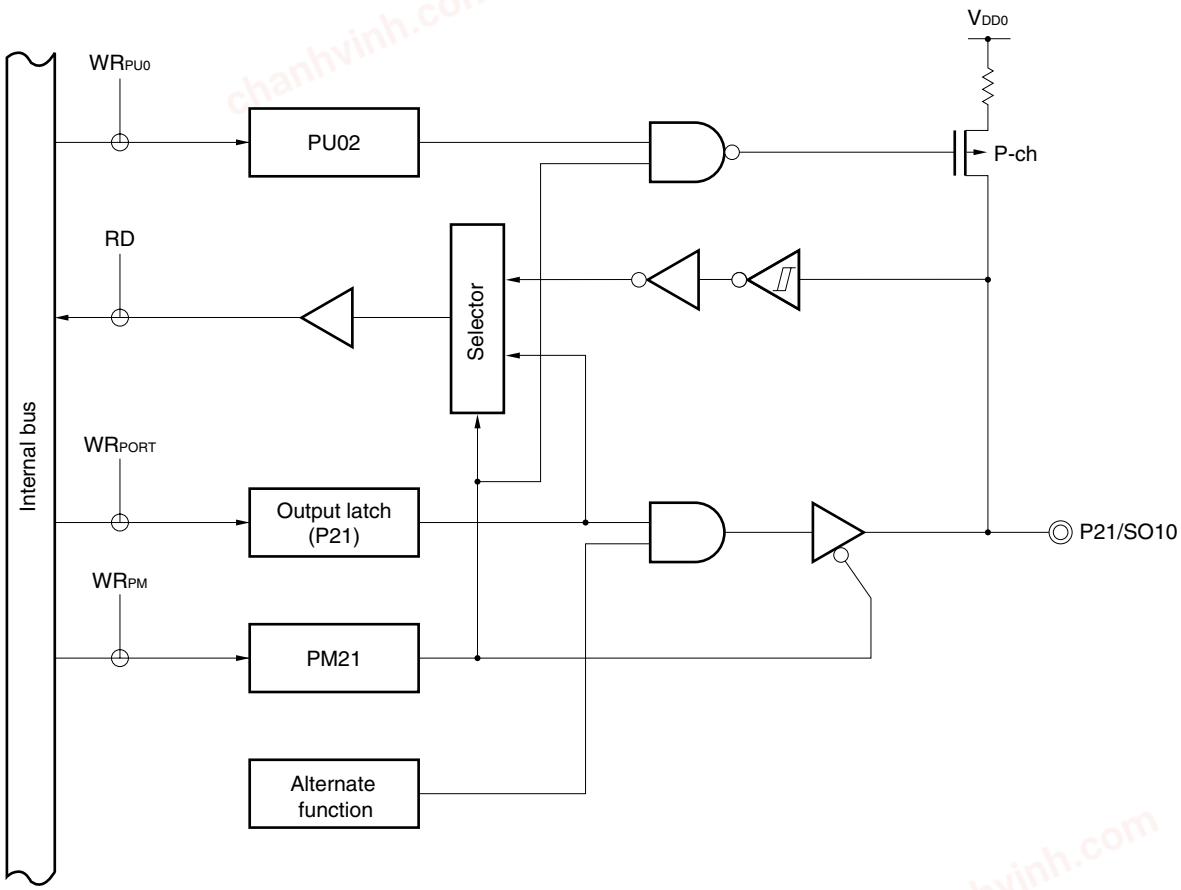
Figure 4-4. Block Diagram of P20



PU0: Pull-up resistor option register 0
PM: Port mode register
RD: Port 2 read signal
WR: Port 2 write signal

chanhvinh.com

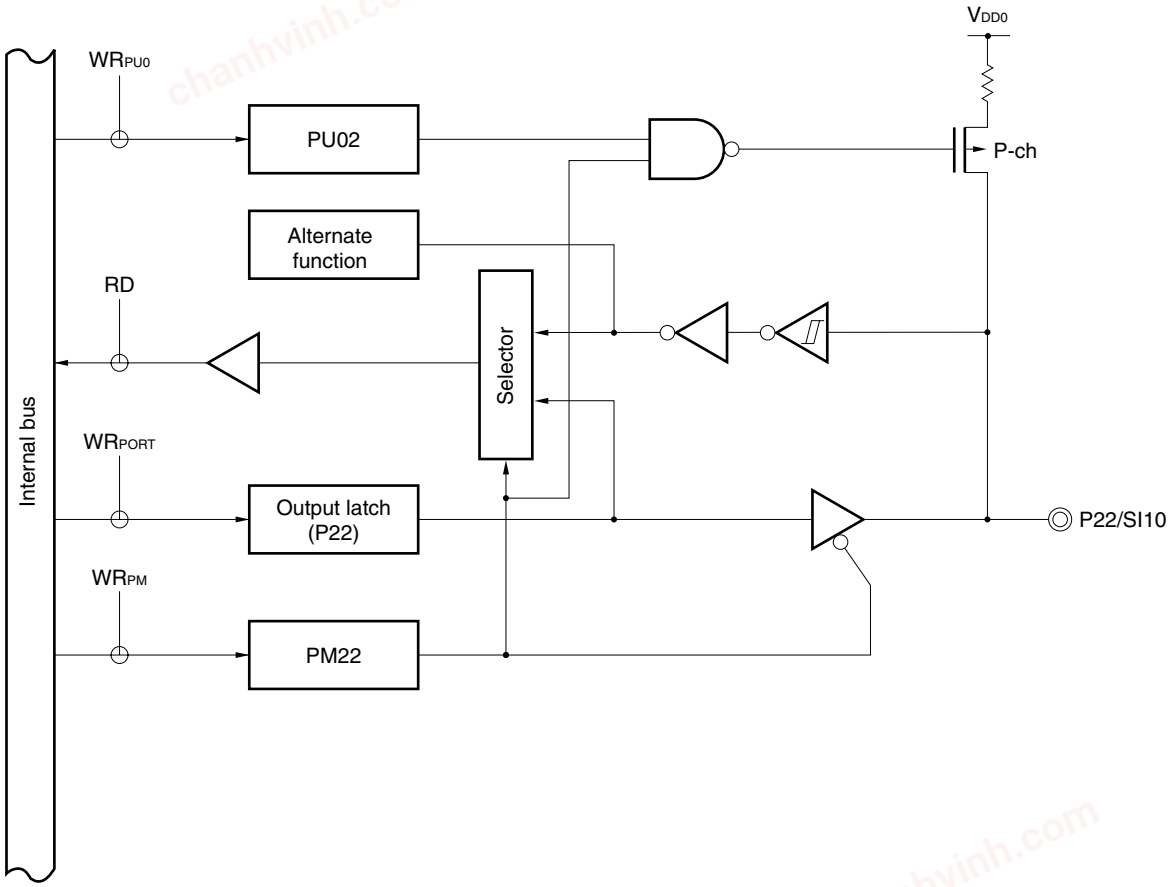
Figure 4-5. Block Diagram of P21



PU0: Pull-up resistor option register 0
PM: Port mode register
RD: Port 2 read signal
WR: Port 2 write signal

chanhvinh.com

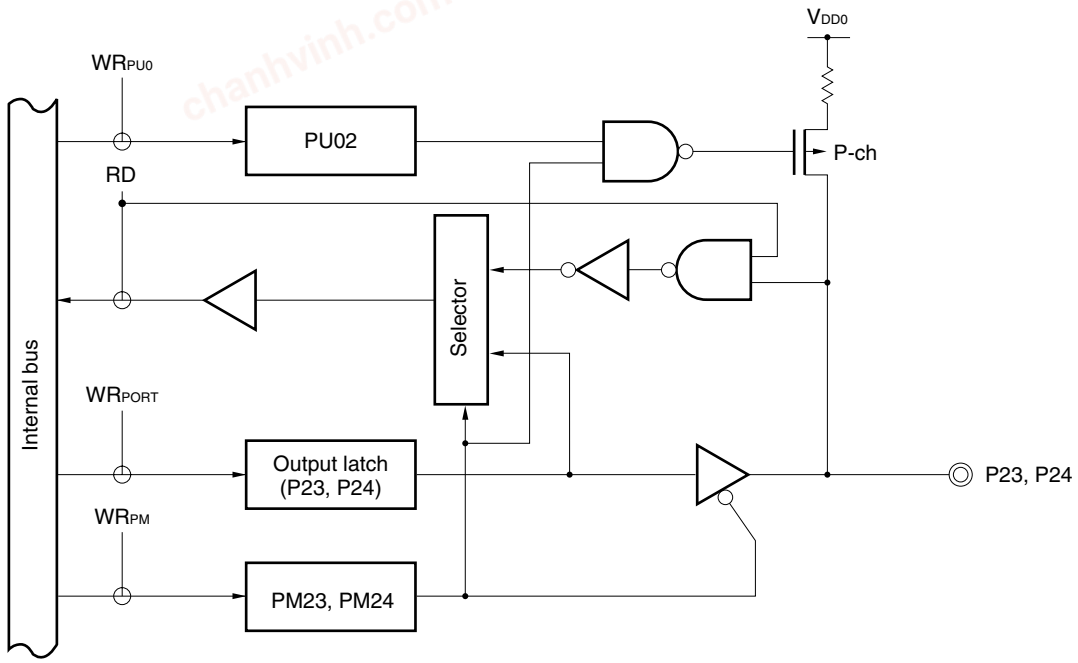
Figure 4-6. Block Diagram of P22



PU0: Pull-up resistor option register 0
PM: Port mode register
RD: Port 2 read signal
WR: Port 2 write signal

chanhvinh.com

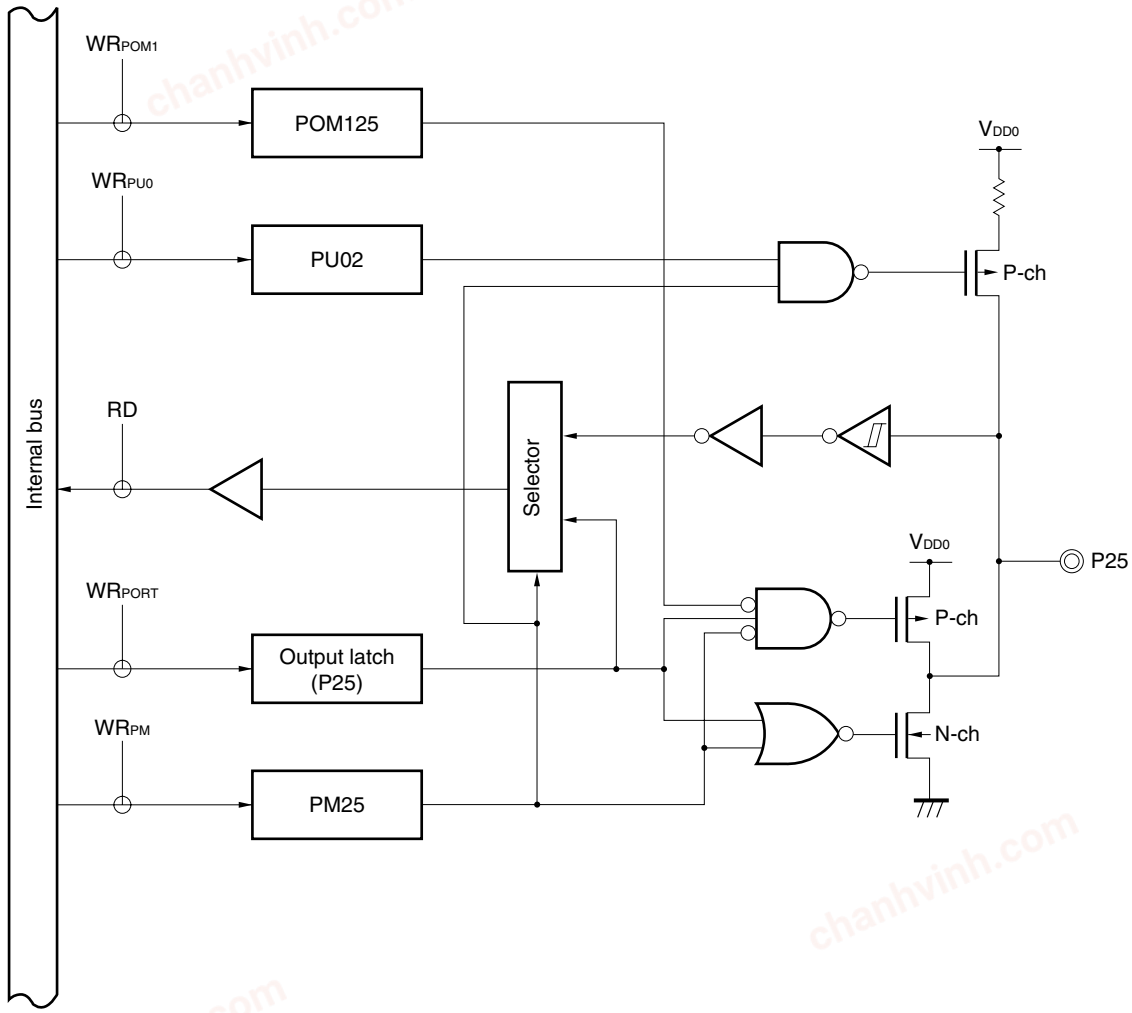
Figure 4-7. Block Diagram of P23 and P24



PU0: Pull-up resistor option register 0
PM: Port mode register
RD: Port 2 read signal
WR: Port 2 write signal

chanhvinh.com

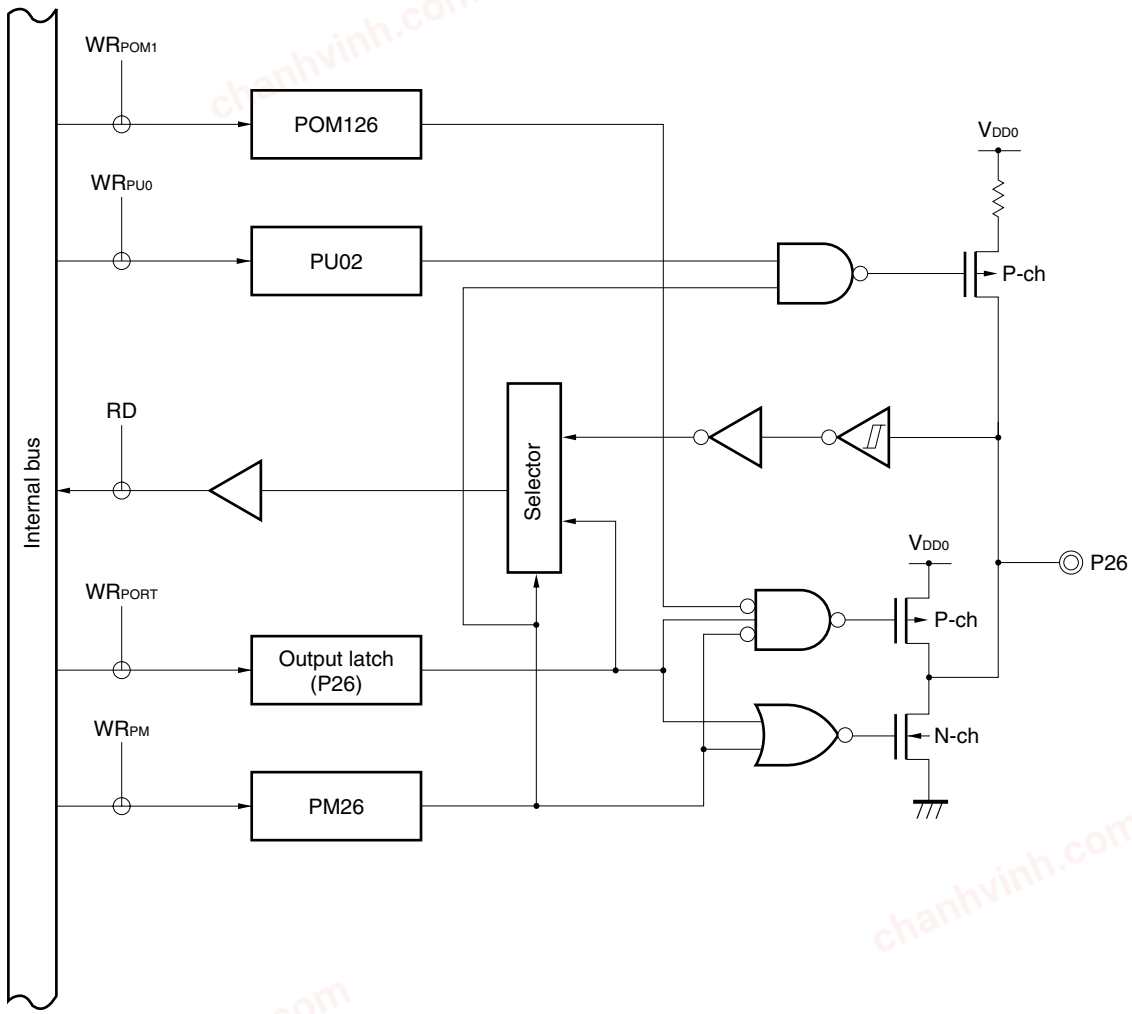
Figure 4-8. Block Diagram of P25



POM1: Port output mode register 1
PU0: Pull-up resistor option register 0
PM: Port mode register
RD: Port 2 read signal
WR: Port 2 write signal

chanhvinh.com

Figure 4-9. Block Diagram of P26



POM1: Port output mode register 1
PU0: Pull-up resistor option register 0
PM: Port mode register
RD: Port 2 read signal
WR: Port 2 write signal

chanhvinh.com

4.2.4 Port 4

This is an 8-bit I/O port with an output latch. Port 4 can be specified in the input or output mode in 1-bit units by using port mode register 4 (PM4). When using P40 to P47 pins as input port pins, on-chip pull-up resistors can be connected in 8-bit units by using pull-up resistor option register 0 (PU0).

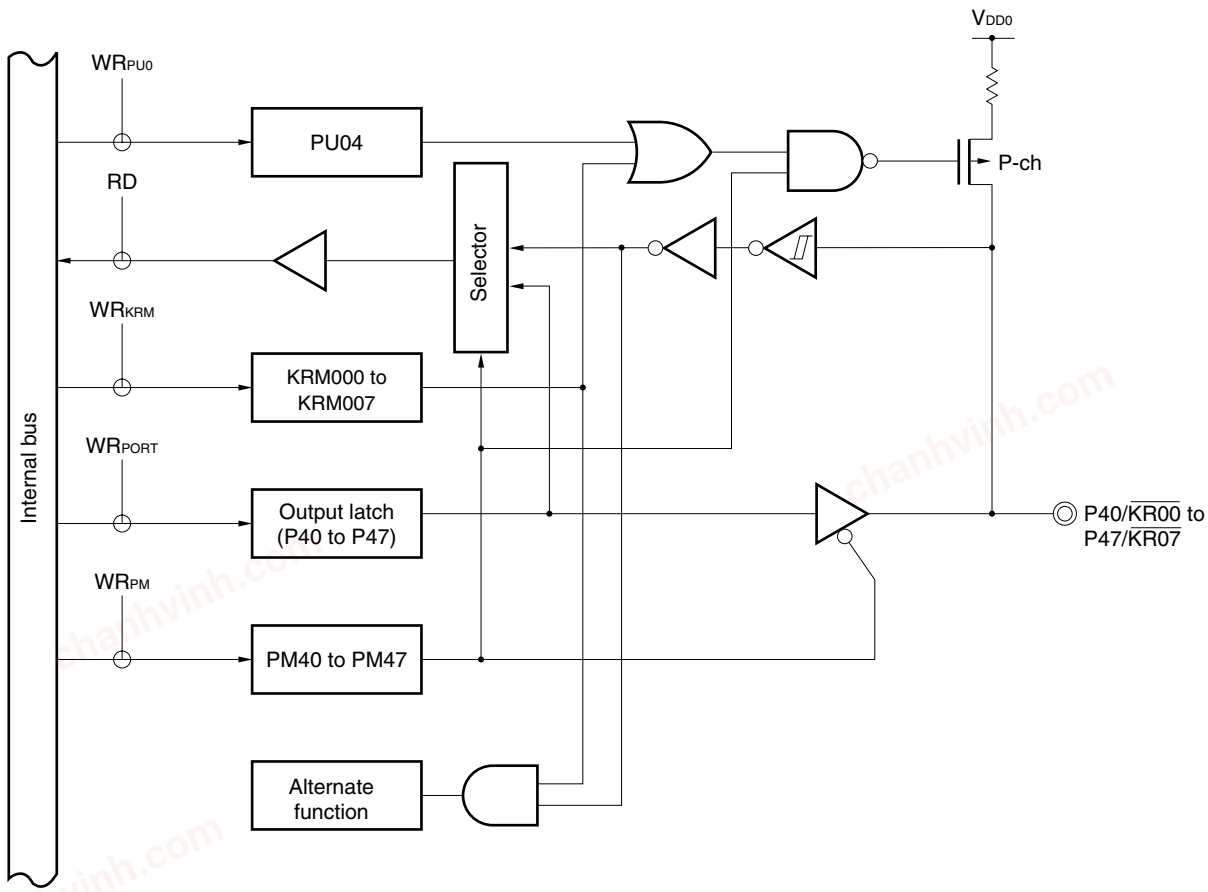
The port is also used as a key return input.

This port is set in the input mode when the $\overline{\text{RESET}}$ signal is input.

Figure 4-10 shows a block diagram of port 4.

Caution When using the pins of port 4 as the key return, key return mode register 00 (KRM00) must be set according to the function to be used. For how to set the register, see Section 11.3 (5) Key return mode register 00 (KRM00).

Figure 4-10. Block Diagram of P40 to P47



KRM00: Key return mode register 00
PU0: Pull-up resistor option register 0
PM: Port mode register
RD: Port 4 read signal
WR: Port 4 write signal

chanhvinh.com

4.3 Registers Controlling Port Function

The following three types of registers control the ports.

- Port mode registers (PM0, PM1, PM2, PM4)
- Pull-up resistor option register (PU0)
- Port output mode registers (POM0, POM1)

(1) Port mode registers (PM0, PM1, PM2, PM4)

These registers are used to set port input/output in 1-bit units.

The port mode registers are independently set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets registers to FFH.

When port pins are used as alternate-function pins, set the port mode register and output latch according to Table 4-3.

Caution As P26 can be used as an external interrupt input, when the port function output mode is specified and the output level is changed, the interrupt request flag is set. When the output mode is used, therefore, the interrupt mask flag should be set to 1 beforehand.

Figure 4-11. Format of Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FF21H	FFH	R/W
PM2	1	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40	FF24H	FFH	R/W

PMmn	Pmn pin input/output mode selection
0	Output mode (output buffer on)
1	Input mode (output buffer off)

chanhvinh.com

Table 4-3. Port Mode Register and Output Latch Settings When Using Alternate Functions

Pin Name	Secondary Function		PMxx	Pxx
	Name	Input/Output		
P26	TO01	Output	0	0
	TI01	Input	1	×
	INTP0	Input	1	×
P40 to P47[Note]	$\overline{KR00}$ to $\overline{KR07}$	Input	1	×

Note Set key return mode register 00 (KRM00) to 1 when using the alternate function (see Section **11.3 (5) Key return mode register 00 (KRM00)**).

Caution When Port 2 is used as a serial interface pin, the I/O latch or output latch must be set according to its function. For the setting method, see Table 9-2 Settings of Serial interface 10 Operating Mode.

Remark x: Don't care
PMxx: Port mode register
Pxx: Port output latch

(2) Pull-up resistor option register 0 (PU0)

The pull-up resistor option register (PU0) sets whether an on-chip pull-up resistor on each port is used or not.
On the port which is specified to use the on-chip pull-up resistor in the PU0, the pull-up resistor can be internally used only for the bits set to the input mode. No on-chip pull-up resistors can be used for the bits set in the output mode regardless of the setting PU0. This applies to the case when using the output pins for alternate functions.
PU0 is set with a 1-bit or 8-bit memory manipulation instruction.
$\overline{RESET}$ input sets PU0 to 00H.

Figure 4-12. Format of Pull-up Resistor Option Register 0

Symbol	7	6	5	<4>	3	<2>	<1>	<0>	Address	After reset	R/W
PU0	0	0	0	PU04	0	PU02	PU01	PU00	FFF7H	00H	R/W

PU0m	Pm on-chip pull-up resistor selection (m = 0 to 2, 4)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

Caution Bits 3 and 5 to 7 must be set to 0.

chanhvinh.com

(3) Port output mode registers (POM0 and POM1)

The port output mode registers (POM0 and POM1) are used to switch from CMOS output to N-ch open-drain output for port 0, port 1, pin P25, and pin P26.

Set POM0 and POM1 with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets P0M0 and POM1 to 00H.

Figure 4-13. Format of Port Output Mode Register 0

Symbol	7	6	5	4	3	2	<1>	<0>	Address	After reset	R/W
POM0	0	0	0	0	0	0	POM01	POM00	FF30H	00H	R/W

POM0m	Pm output mode selection[Note] (m = 0, 1)
0	CMOS output
1	N-ch open-drain output

Note POM0 selects the output mode for a port in 8-bit units.

Caution Bits 2 to 7 must be set to 0.

Figure 4-14. Format of Port Output Mode Register 1

Symbol	7	<6>	<5>	4	3	2	1	0	Address	After reset	R/W
POM1	0	POM126	POM125	0	0	0	0	0	FF31H	00H	R/W

POM12n	Output mode selection for bit n of port 2[Note] (n = 5, 6)
0	CMOS output
1	N-ch open-drain output

Note POM1 selects the output mode for P25 or P26 in 1-bit units.

Caution Bits 0 to 4 and 7 must be set to 0.

chanhvinh.com

4.4 Port Function Operation

The operation of a port differs depending on whether the port is set to the input or output mode, as described below.

4.4.1 Writing to I/O port

(1) In output mode

A value can be written to the output latch of a port by using a transfer instruction. The contents of the output latch can be output from the pins of the port.

The data once written to the output latch is retained until new data is written to the output latch.

(2) In input mode

A value can be written to the output latch by using a transfer instruction. However, the status of the port pin is not changed because the output buffer is off.

The data once written to the output latch is retained until new data is written to the output latch.

Caution A 1-bit memory manipulation instruction is executed to manipulate 1 bit of a port. However, this instruction accesses the port in 8-bit units. When this instruction is executed to manipulate a bit of an input/output port, therefore, the contents of the output latch of the pin that is set to the input mode and not subject to manipulation become undefined.

4.4.2 Reading from I/O port

(1) In output mode

The status of a pin can be read by using a transfer instruction. The contents of the output latch are not changed.

(2) In input mode

The status of a pin can be read by using a transfer instruction. The contents of the output latch are not changed.

4.4.3 Arithmetic operation of I/O port

(1) In output mode

An arithmetic operation can be performed on the contents of the output latch. The result of the operation is written to the output latch. The contents of the output latch are output from the port pins.

The data once written to the output latch is retained until new data is written to the output latch.

(2) In input mode

The contents of the output latch become undefined. However, the status of the pin is not changed because the output buffer is off.

Caution A 1-bit memory manipulation instruction is executed to manipulate 1 bit of a port. However, this instruction accesses the port in 8-bit units. When this instruction is executed to manipulate a bit of an input/output port, therefore, the contents of the output latch of the pin that is set to the input mode and not subject to manipulation become undefined.

chanhvinh.com

CHAPTER 5 CLOCK GENERATOR

5.1 Clock Generator Functions

The clock generator generates the clock to be supplied to the CPU and peripheral hardware. The following type of system clock oscillator is used.

- **System clock oscillator**
  This circuit oscillates at 6.0 MHz. Oscillation can be stopped by executing the STOP instruction.

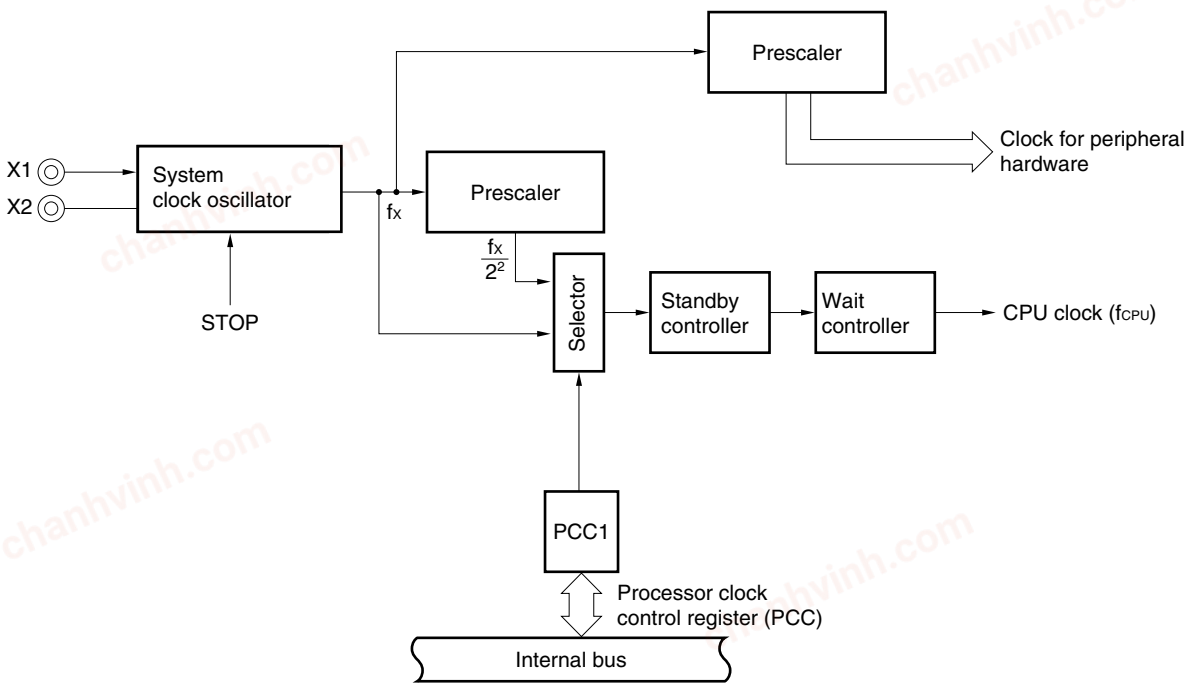
5.2 Clock Generator Configuration

The clock generator consists of the following hardware.

Table 5-1. Configuration of Clock Generator

Item	Configuration
Control register	Processor clock control register (PCC)
Oscillator	System clock oscillator

Figure 5-1. Block Diagram of Clock Generator



chanhvinh.com

5.3 Register Controlling Clock Generator

The clock generator is controlled by the following register.

- Processor clock control register (PCC)

(1) Processor clock control register (PCC)

PCC selects the CPU clock and sets the of division ratio.

PCC is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the PCC to 02H.

Figure 5-2. Format of Processor Clock Control Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PCC	0	0	0	0	0	0	PCC1	0	FFFBH	02H	R/W

PCC1	CPU clock (f_{CPU}) selection	Minimum instruction execution time: $2/f_{CPU}$
		f_X = 6.0 MHz operation
0	f_X	0.33 μs
1	$f_X/2^2$	1.33 μs

Caution Bits 0 and 2 to 7 must be set to 0.

Remark f_X: system clock oscillation frequency

chanhvinh.com

5.4 System Clock Oscillators

5.4.1 System clock oscillator

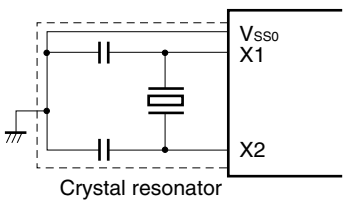
The system clock oscillator is oscillated by the crystal resonator (6.0 MHz TYP.) connected across the X1 and X2 pins.

An external clock can also be input to the circuit. In this case, input the clock signal to the X1 pin, and leave the X2 pin open.

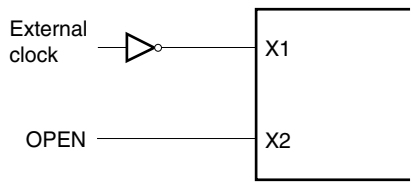
Figure 5-3 shows the external circuit of the system clock oscillator.

★ **Figure 5-3. External Circuit of System Clock Oscillator**

(a) Crystal oscillation



(b) External clock



Caution When using the system clock oscillator, wire as follows in the area enclosed by the broken lines in Figure 5-3 to avoid an adverse effect from wiring capacitance.

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS0}. Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

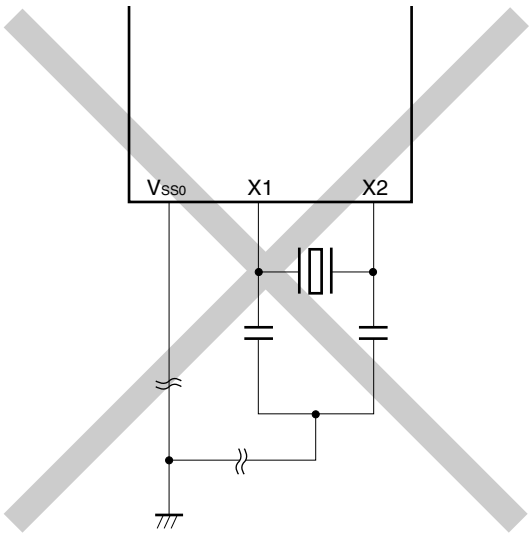
chanhvinh.com

5.4.2 Examples of incorrect resonator connection

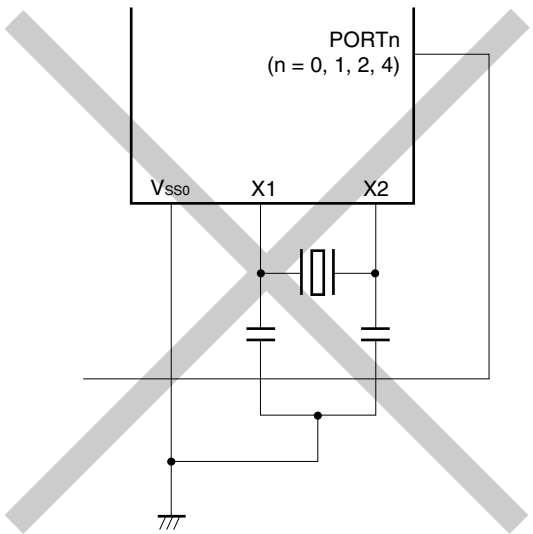
Figure 5-4 shows examples of incorrect resonator connection.

Figure 5-4. Examples of Incorrect Resonator Connection (1/2)

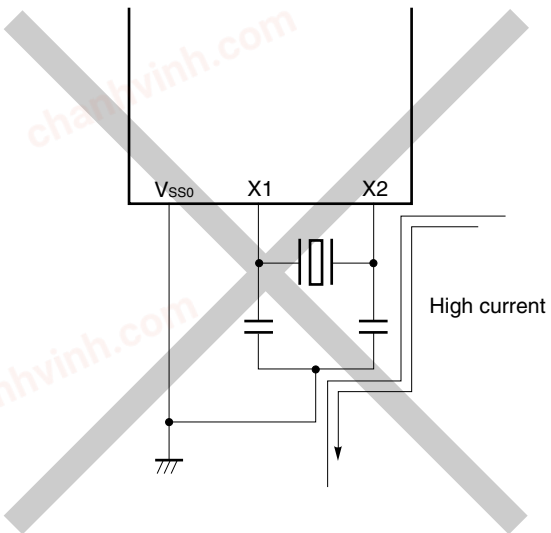
(a) Too long wiring



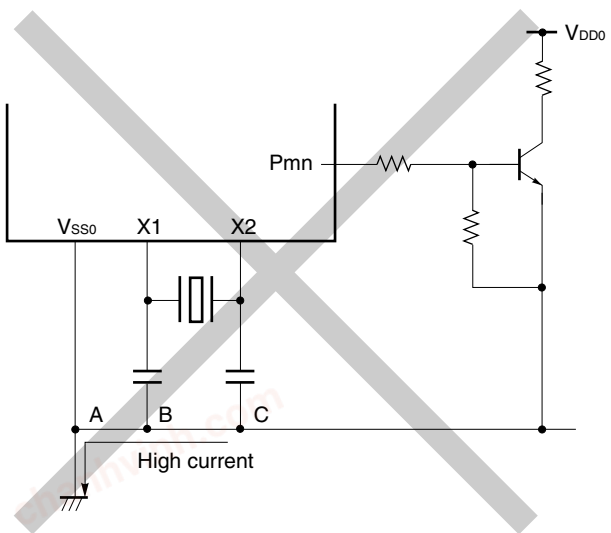
(b) Crossed signal line



(c) Wiring near high fluctuating current



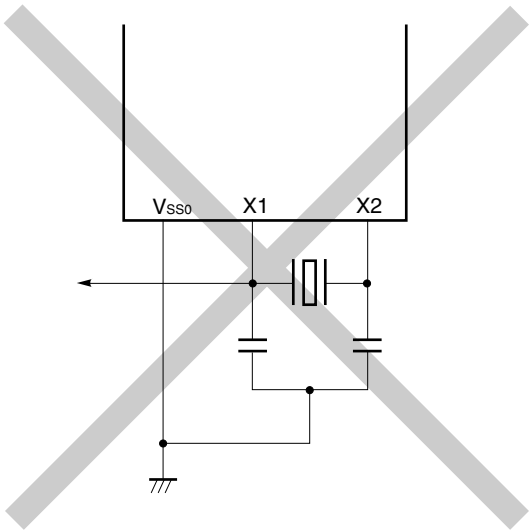
(d) Current flowing through ground line of oscillator (potential at points A, B, and C fluctuates)



chanhvinh.com

Figure 5-4. Examples of Incorrect Resonator Connection (2/2)

(e) Signals are fetched



5.4.3 Frequency divider

The frequency divider divides the output of the system clock oscillator (fx) to generate various clocks.

5.5 Clock Generator Operation

The clock generator generates the following clocks and controls the operation modes of the CPU, such as the standby mode.

- System clock fx
- CPU clock f_{CPU}
- Clock to peripheral hardware

The operation of the clock generator is determined by the processor clock control register (PCC), as follows.

(a) The slow mode (1.33 μs: at 6.0 MHz operation) of the system clock is selected when the $\overline{\text{RESET}}$ signal is generated (PCC = 02H). While a low level is being input to the $\overline{\text{RESET}}$ pin, oscillation of the system clock is stopped.

(b) Two types of minimum instruction execution time (0.33 μs and 1.33 μs: at 6.0 MHz operation) can be selected by the PCC setting.

(c) Two standby modes, STOP and HALT, can be used.

(d) The clock pulse for the peripheral hardware is generated by dividing the frequency of the system clock. So, the other hardware stops when the system clock stops (except for external clock pulses).

chanhvinh.com

5.6 Changing Setting of CPU Clock

5.6.1 Time required for switching CPU clock

The CPU clock can be selected by using bit 1 (PCC1) of the processor clock control register (PCC).

Actually, the specified clock is not selected immediately after the setting of PCC has been changed; the old clock is used for the duration of several instructions after that (see **Table 5-2**).

Table 5-2. Maximum Time Required for Switching CPU Clock

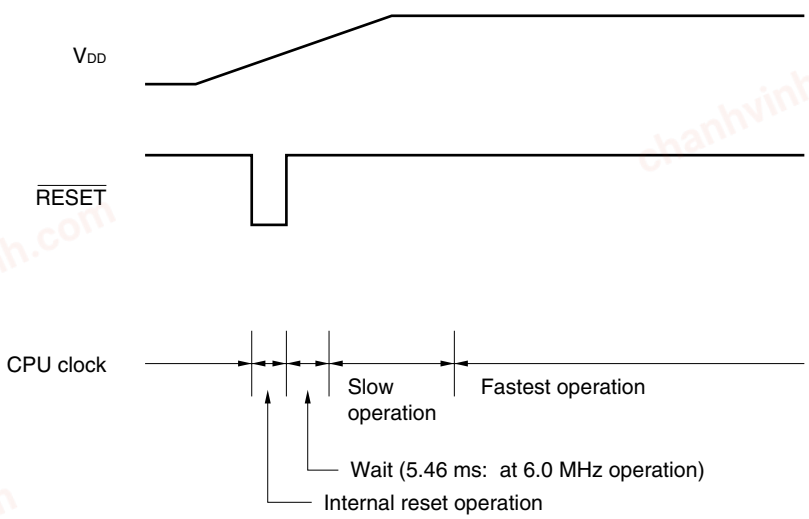
Set Value Before Switching	Set Value After Switching	
PCC1	PCC1	PCC1
	0	1
0		4 clocks
1	2 clocks	

Remark Before switching, the minimum instruction execution time of the CPU clock is two clocks.

5.6.2 Switching CPU clock

The following figure illustrates how the CPU clock switches.

Figure 5-5. Switching of CPU Clock



<1> The CPU is reset when the $\overline{\text{RESET}}$ pin is made low on power application. The effect of resetting is released when the $\overline{\text{RESET}}$ pin is later made high, and the system clock starts oscillating. At this time, the time during which oscillation stabilization (2^{15}/fx) is automatically secured.

After that, the CPU starts instruction execution at the slow speed of the system clock (1.33 μs: at 6.0 MHz operation).

<2> After the time required for the VDD voltage to rise to the level at which the CPU can operate at the highest speed has elapsed, the processor clock control register (PCC) is rewritten so that the highest speed operation can be selected.

chanhvinh.com

CHAPTER 6 8-BIT TIMER/EVENT COUNTERS 00 AND 01

6.1 Functions of 8-Bit Timer/Event Counters 00 and 01

The 8-bit timer/event counters (TM00 and TM01) have the following functions.

- Interval timer (TM00 and TM01)
- External event counter (TM01 only)
- Square wave output (TM01 only)

The μPD789800 Subseries is provided with a 1-channel (TM01) 8-bit timer/event counter and a 1-channel (TM00) 8-bit timer. When reading the description of TM00, “timer/event counter” should be read as “ timer”.

(1) 8-bit interval timer

When the 8-bit timer/event counter is used as an interval timer, it generates an interrupt at any preset time interval.

Table 6-1. Interval Time of 8-Bit Timer 00

Minimum Interval Time	Maximum Interval Time	Resolution
2^6/fx (10.7 μs)	2^{14}/fx (2.73 ms)	2^6/fx (10.7 μs)
2^9/fx (85.3 μs)	2^{17}/fx (21.8 ms)	2^9/fx (85.3 μs)

Remarks 1. fx: System clock oscillation frequency

2. The parenthesized values apply to operation at fx = 6.0 MHz.

Table 6-2. Interval Time of 8-Bit Timer/Event Counter 01

Minimum Interval Time	Maximum Interval Time	Resolution
2^4/fx (2.67 μs)	2^{12}/fx (682.7 μs)	2^4/fx (2.67 μs)
2^8/fx (42.7 μs)	2^{16}/fx (10.9 ms)	2^8/fx (42.7 μs)

Remarks 1. fx: System clock oscillation frequency

2. The parenthesized values apply to operation at fx = 6.0 MHz.

(2) External event counter

The number of pulses of an externally input signal can be measured.

chanhvinh.com

(3) Square wave output

A square wave of arbitrary frequency can be output.

Table 6-3. Square Wave Output Range of 8-Bit Timer/Event Counter 01

Minimum Pulse Width	Maximum Pulse Width	Resolution
2^4/fx (2.67 μs)	2^{12}/fx (682.7 μs)	2^4/fx (2.67 μs)
2^8/fx (42.7 μs)	2^{16}/fx (10.9 ms)	2^8/fx (42.7 μs)

Remarks 1. fx: System clock oscillation frequency

2. The parenthesized values apply to operation at fx = 6.0 MHz

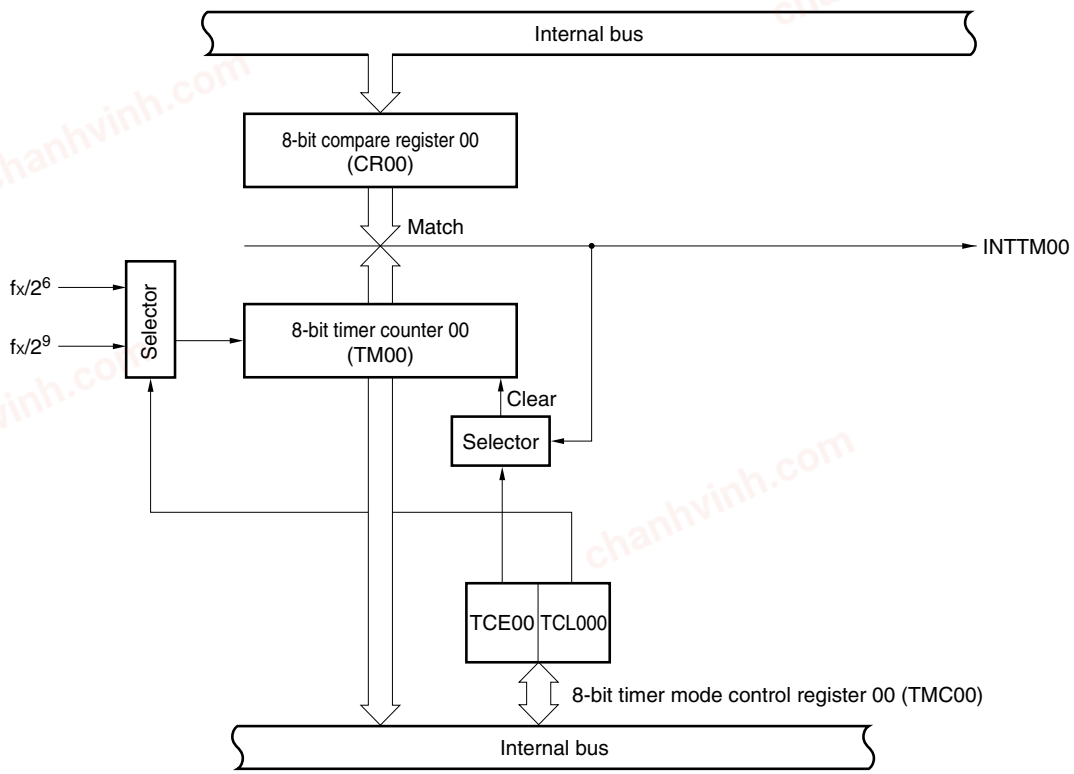
6.2 Configuration of 8-Bit Timer/Event Counters 00 and 01

8-bit timer/event counters 00 and 01 consist of the following hardware.

Table 6-4. Configuration of 8-Bit Timer/Event Counters 00 and 01

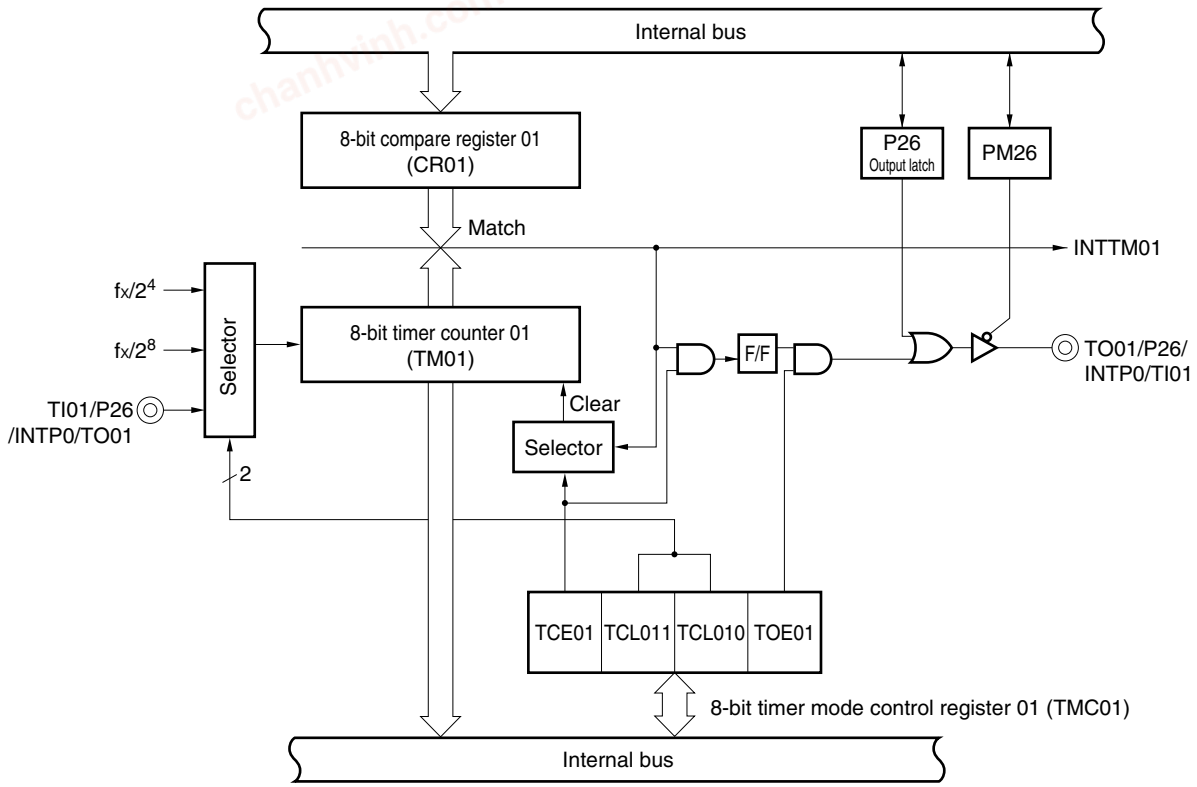
Item	Configuration
Timer counter	8 bits × 2 (TM00 and TM01)
Register	Compare register: 8 bits × 2 (CR00 and CR01)
Timer outputs	1 (TO01)
Control registers	8-bit timer mode control registers 00 and 01 (TMC00 and TMC01) Port mode register 2 (PM2)

Figure 6-1. Block Diagram of 8-Bit Timer 00



chanhvinh.com

Figure 6-2. Block Diagram of 8-Bit Timer/Event Counter 01



(1) 8-bit compare register 0n (CR0n)

This is an 8-bit register used to compare the value set to CR0n with the 8-bit timer counter 0n (TM0n) count value, and if they match, generate used an interrupt request (INTTM0n).

CR0n is set with an 8-bit memory manipulation instruction. Values from 00H to FFH can be set.

$\overline{\text{RESET}}$ input sets CR0n undefined.

Caution Be sure to set CR0n after the timer operation is stopped.

Remark n = 0 or 1

(2) 8-bit timer counter 0n (TM0n)

This is an 8-bit register used to count pulses.

TM0n is read with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TMn to 00H.

Remark n = 0 or 1

chanhvinh.com

6.3 Registers Controlling 8-Bit Timer/Event Counters 00 and 01

The following two types of registers are used to control 8-bit timer/event counters 00 and 01.

- 8-bit timer mode control registers 00 and 01 (TMC00 and TMC01)
- Port mode register 2 (PM2)

(1) 8-bit timer mode control register 00 (TMC00)

This register enables/stops operation of 8-bit timer counter 00 (TM00) and sets the counter clock of 8-bit timer 00.

TMC00 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TMC00 to 00H.

Figure 6-3. Format of 8-Bit Timer Mode Control Register 00

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
TMC00	TCE00	0	0	0	0	0	TCL000	0	FF53H	00H	R/W

TCE00	8-bit timer counter 00 operation control
0	Operation disabled (TM00 cleared to 0)
1	Operation enabled

TCL000	8-bit timer 00 count clock selection
0	$f_X/2^6$ (93.8 kHz)
0	$f_X/2^9$ (11.7 kHz)

Caution Be sure to set the count clock after the timer operation is stopped (TCE00 = 0). Refer to 6.4 Operation of 8-Bit Timer/Event Counters 00 and 01 for details.

Remarks 1. f_X: System clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 6.0 MHz.

chanhvinh.com

(2) 8-bit timer mode control register 01 (TMC01)

TMC01 determines whether to enable or disable 8-bit timer counter 01 (TM01), specifies the count clock for the 8-bit timer/event counter, and controls the operation of the output controller.

TMC01 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TMC01 to 00H.

Figure 6-4. Format of 8-Bit Timer Mode Control Register 01

Symbol	<7>	6	5	4	3	2	1	<0>	Address	After reset	R/W
TMC01	TCE01	0	0	0	0	TCL011	TCL010	TOE01	FF57H	00H	R/W

TCE01	8-bit timer counter 01 operation control
0	Operation disabled (TM01 is cleared to 0.)
1	Operation enabled

TCL011	TCL010	8-bit timer/event counter 01 count clock selection
0	0	$f_X/2^4$ (375 kHz)
0	1	$f_X/2^8$ (23.4 kHz)
1	0	Rising edge of TI01[Note]
1	1	Falling edge of TI01[Note]

TOE01	8-bit timer/event counter 01 output control
0	Output disabled (port mode)
1	Output enabled

Note When inputting a clock signal externally, timer output cannot be used.

Caution Be sure to set the count clock after the timer operation is stopped (TCE01 = 0). Refer to 6.4 Operation of 8-Bit Timer/Event Counters 00 and 01 for details.

Remarks 1. f_X: System clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 6.0 MHz.

chanhvinh.com

(3) Port mode register 2 (PM2)

This register sets port 2 input/output in 1-bit units.

When using the P26/TO01/INTP0/TI01 pin for timer output, set P26 and the output latch of P26 to 0.

When P26/TO01/INTP0/TI01 pin is used as a timer input, set PM26 to 1.

PM2 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM2 to FFH.

Figure 6-5. Format of Port Mode Register 2

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM2	1	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W

PM26	P26 pin input/output mode selection
0	Output mode (output buffer on)
1	Input mode (output buffer off)

chanhvinh.com

6.4 Operation of 8-Bit Timer/Event Counters 00 and 01

6.4.1 Operation as interval timer

Interval timer repeatedly generates an interrupt at time intervals specified by the count value set to 8-bit compare registers 00 and 01 (CR00 and CR01) in advance.

To operate the 8-bit timer/event counter as an interval timer, the following settings are required.

<1> Disable operation of the 8-bit timer counter 0n (TM0n) by setting TCE0n (bit 7 of 8-bit timer mode control register 0n (TMC0n)) to 0.
<2> Set the count clock of the 8-bit timer/event counter (see **Tables 6-5** and **6-6**).
<3> Set count values to CR0n.
<4> Enable operation of TM0n by setting TCE0n to 1.

When the count value of 8-bit timer counter 0n (TM0n) matches the value set to CR0n, the value of TMn is cleared to 0 and TM0n continues counting. At the same time, an interrupt request signal (INTTM0n) is generated.

Tables 6-5 and 6-6 show the interval time, and Figures 6-6 and 6-7 show the timing of interval timer operation.

Caution When the TMC0n count clock is set and the operation of TM0n is enabled simultaneously by an 8-bit memory manipulation instruction, an error of more than 1 clock may occur in 1 cycle after the timer has been started. Therefore, be sure to follow the settings above when the 8-bit timer/event counter is operating as an interval timer.

Remark n = 0 or 1

Table 6-5. Interval Time of 8-Bit Timer 00

TCL000	Minimum Interval Time	Maximum Interval Time	Resolution
0	2^6/fx (10.7 μs)	2^{14}/fx (2.73 μs)	2^6/fx (10.7 μs)
1	2^9/fx (85.3 μs)	2^{17}/fx (21.8 ms)	2^9/fx (85.3 μs)

Remarks 1. fx: System clock oscillation frequency
2. The parenthesized values apply to operation at fx = 6.0 MHz.

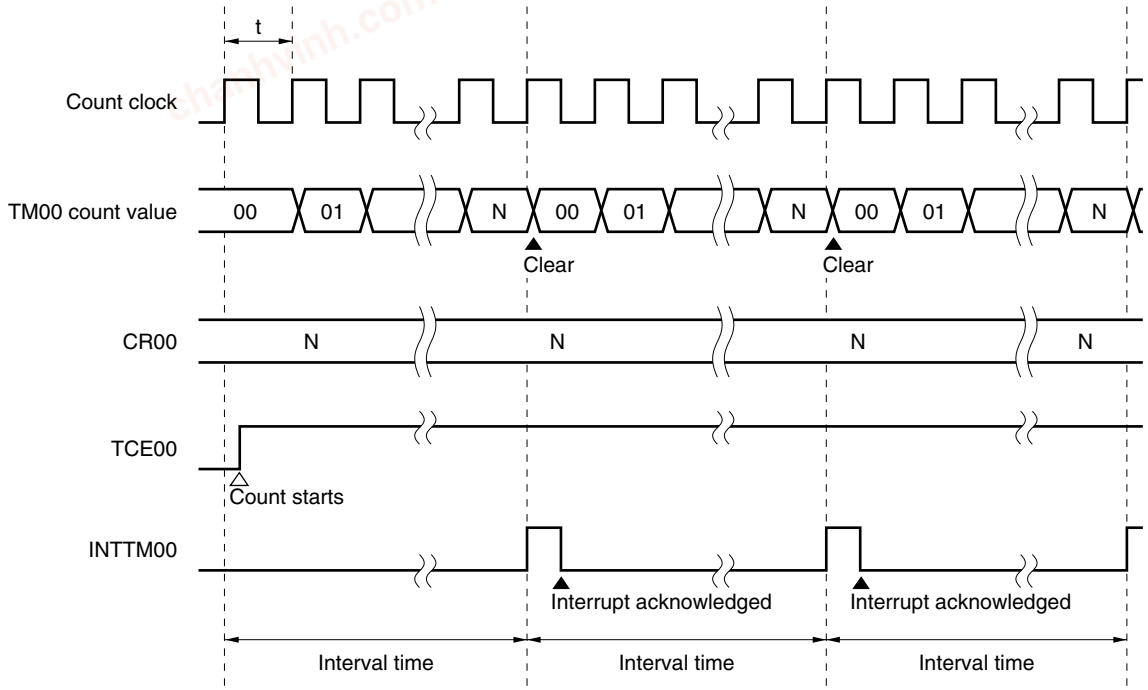
Table 6-6. Interval Time of 8-Bit Timer/Event Counter 01

TCL011	TCL010	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	2^4/fx (2.67 μs)	2^{12}/fx (682.7 μs)	2^4/fx (2.67 μs)
0	1	2^8/fx (42.7 μs)	2^{16}/fx (10.9 ms)	2^8/fx (42.7 μs)
1	0	TI01 input cycle	$2^8 \times$ TI01 input cycle	TI01 input edge cycle
1	1	TI01 input cycle	$2^8 \times$ TI01 input cycle	TI01 input edge cycle

Remarks 1. fx: System clock oscillation frequency
2. The parenthesized values apply to operation at fx = 6.0 MHz.

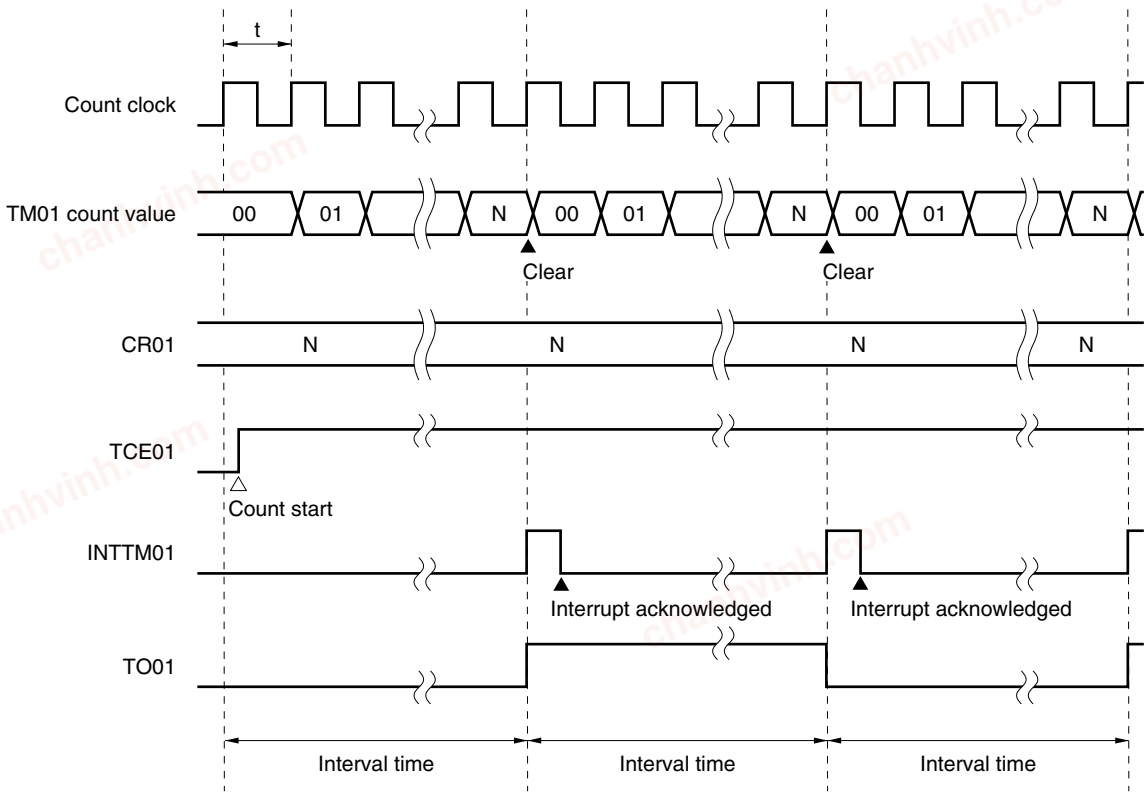
chanhvinh.com

Figure 6-6. Interval Timer Operation Timing of 8-Bit Timer 00



Remark Interval time = (N + 1) × t where N = 00H to FFH

Figure 6-7. Interval Timer Operation Timing of 8-Bit Timer/Event Counter 01



Remark Interval time = (N + 1) × t, where N = 00H to FFH

chanhvinh.com

6.4.2 Operation as external event counter (timer 01 only)

The external event counter counts the number of external clock pulses input to the TI01/P26/INTP0/TO01 pin by using timer counter 01 (TM01).

To operate the 8-bit timer/event counter as an external event counter, the following settings are required.

<1> Disable operation of 8-bit timer counter 01 (TM01) by setting TCE01 (bit 7 of 8-bit timer mode control register 01 (TMC01)) to 0.

<2> Specify the rising/falling edge of TI01 (see **Table 6-6**), and set TO01 to output-disabled (TOE01 (bit 0 of TMC01) = 0).

<3> Set count values to CR01.

<4> Enable operation of TM01 by setting TCE01 to 1.

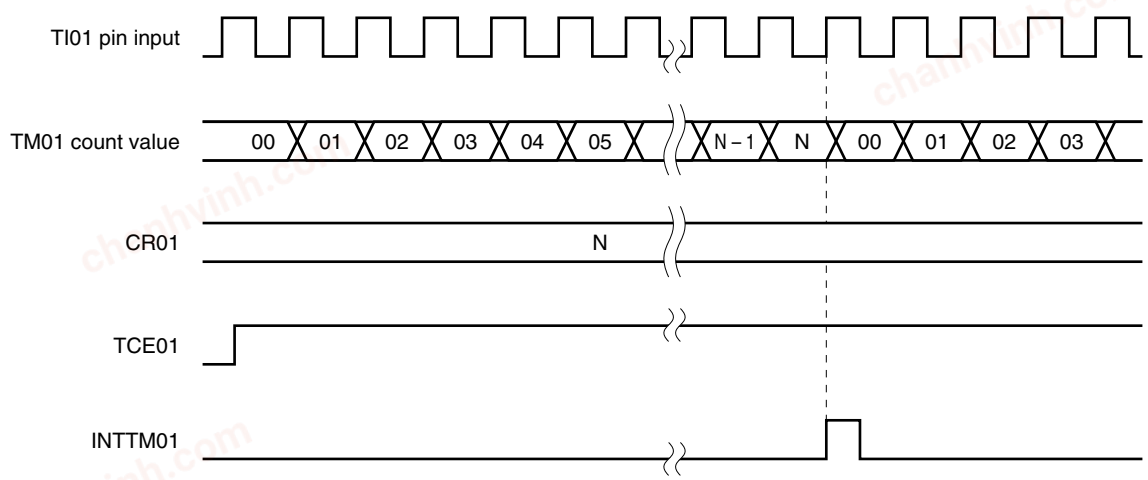
Each time the valid edge specified by bit 1 or 2 (TCL011 or TCL010) of TMC01 is input, the value of 8-bit timer counter 01 (TM01) is incremented.

When the count value of TM01 matches the value set to CR01, the value of TM01 is cleared to 0 and TM01 continues counting. At the same time, an interrupt request signal (INTTM01) is generated.

Figure 6-8 shows the timing of external event counter operation (with rising edge specified).

Caution When the TMC01 count clock is set and the operation of TM01 is enabled simultaneously by an 8-bit memory manipulation instruction, an error of more than 1 clock may occur in 1 cycle after the timer has been started. Therefore, be sure to follow the settings above when the 8-bit timer/event counter is operating as an external event counter.

★ **Figure 6-8. Timing of External Event Counter Operation (with Rising Edge Specified)**



Remark N = 00H to FFH

chanhvinh.com

6.4.3 Operation as square-wave output (timer 01 only)

The 8-bit timer/event counter can generate output square waves of arbitrary frequency at intervals specified by the count value set to 8-bit compare register 01 (CR01) in advance.

To operate 8-bit timer/event counter 01 as square wave output, the following settings are required.

<1> Set P26 to output mode (PM26 = 0) and the output latch of P26 to 0.
<2> Disable operation of 8-bit timer counter 01 (TM01) by setting TCE01 (bit 7 of 8-bit timer mode control register 01 (TMC01)) to 0.
<3> Set the count clock of 8-bit timer/event counter 01 (see **Table 6-7**) and enable output of TO01 by setting TOE01 (bit 0 of TMC01) to 1
<4> Set count values to CR01.
<5> Enable operation of TM01 by setting TCE01 to 1.

When the count value of 8-bit timer counter 01 (TM01) matches the value set to CR01, the TO01/P26/INTP0/TI01 pin output will be inverted. Through application of this mechanism, square waves of any frequency can be output. As soon as a match occurs, the TM01 value is cleared to 0, TM01 resumes counting, and an interrupt request signal (INTTM01 is generated).

Setting bit 7 of TMC01 (TCE01) to 0 clears the square-wave output to 0.

Table 6-7 lists the square wave output range, and Figure 6-9 shows timing of square wave output.

Caution When the TMC01 count clock is set and the operation of TM01 is enabled simultaneously by an 8-bit memory manipulation instruction, an error of more than 1 clock may occur in 1 cycle after the timer has been started. Therefore, be sure to follow the settings above when the 8-bit timer/event counter is operating as square-wave output.

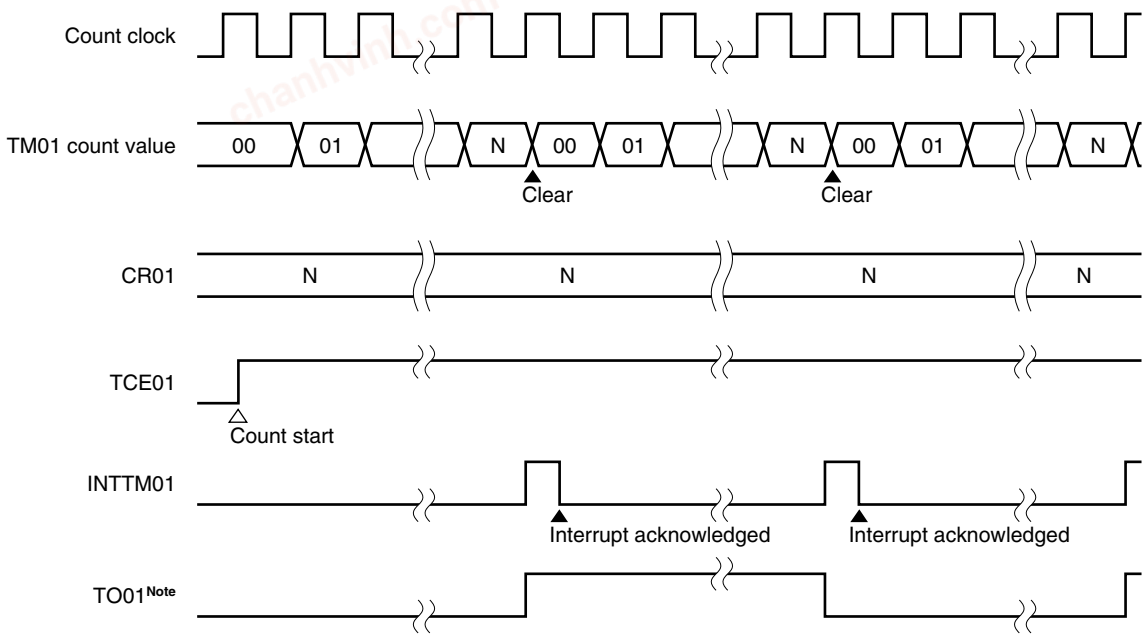
Table 6-7. Square-Wave Output Range of 8-Bit Timer/Event Counter 01

TCL011	TCL010	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	0	2^4/fx (2.67 μs)	2^{12}/fx (682.7 μs)	2^4/fx (2.67 μs)
0	1	2^8/fx (42.7 μs)	2^{16}/fx (10.9 ms)	2^8/fx (42.7 μs)

Remarks 1. fx: System clock oscillation frequency
2. The parenthesized values apply to operation at fx = 6.0 MHz.

chanhvinh.com

Figure 6-9. Timing of Square-Wave Output



Note The initial value of TO01 when output is enabled (TOE01 = 1) becomes low level.

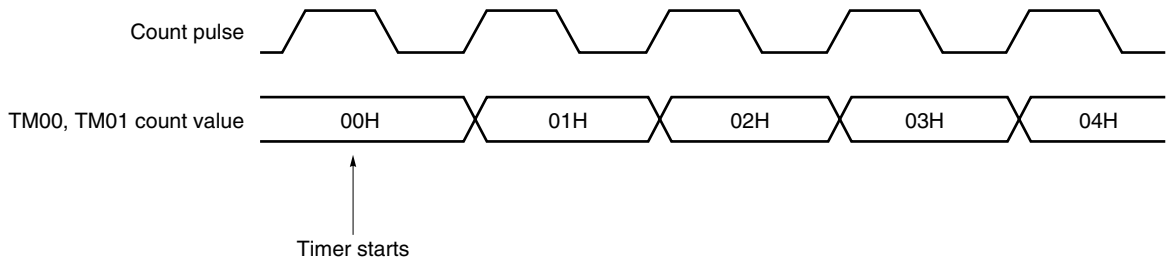
chanhvinh.com

6.5 Notes on Using 8-Bit Timer/Event Counters 00 and 01

(1) Error on starting timer

An error of up to 1 clock occurs after the timer is started until a match signal is generated. This is because 8-bit timer counters 00 and 01 (TM00 and TM01) are started asynchronously to the count pulse.

Figure 6-10. Start Timing of 8-Bit Timer Counter

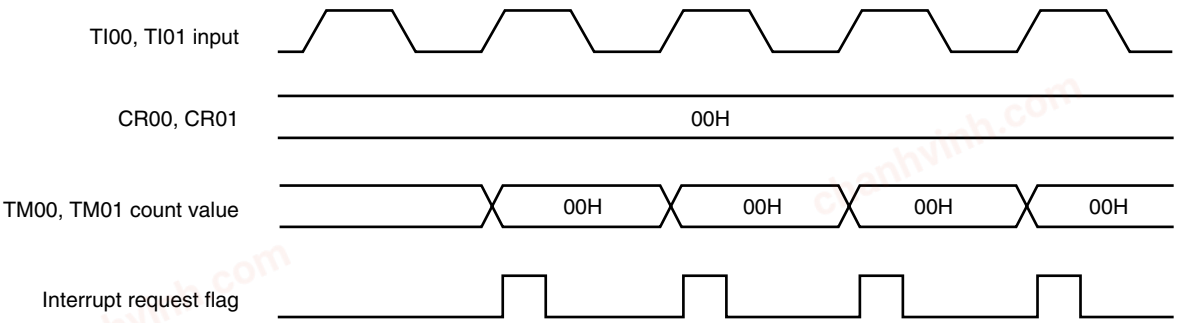


(2) Setting of 8-bit compare register

8-bit compare registers 00 and 01 (CR00 and CR01) can be set to 00H.

Therefore, one pulse can be counted when the 8-bit timer/event counter operates as an event counter.

Figure 6-11. Timing of External Event Counter Operation



chanhvinh.com

CHAPTER 7 WATCHDOG TIMER

7.1 Watchdog Timer Functions

The watchdog timer has the following functions.

- Watchdog timer
- Interval timer

Caution Select the watchdog timer mode or interval timer mode by using the watchdog timer mode register (WDTM).

(1) Watchdog timer

The watchdog timer is used to detect inadvertent program loops. When an inadvertent loop is detected, a non-maskable interrupt or the $\overline{\text{RESET}}$ signal can be generated.

Table 7-1. Inadvertent Loop Detection Time of Watchdog Timer

Inadvertent Loop Detection Time	Operation at fx = 6.0 MHz
$2^{11} \times 1/fx$	341 μs
$2^{13} \times 1/fx$	1.37 ms
$2^{15} \times 1/fx$	5.46 ms
$2^{17} \times 1/fx$	21.8 ms

fx: System clock oscillation frequency

(2) Interval timer

The interval timer generates an interrupt at arbitrary intervals set in advance.

Table 7-2. Interval Time

Interval Time	Operation at fx = 6.0 MHz
$2^{11} \times 1/fx$	341 μs
$2^{13} \times 1/fx$	1.37 ms
$2^{15} \times 1/fx$	5.46 ms
$2^{17} \times 1/fx$	21.8 ms

fx: System clock oscillation frequency

chanhvinh.com

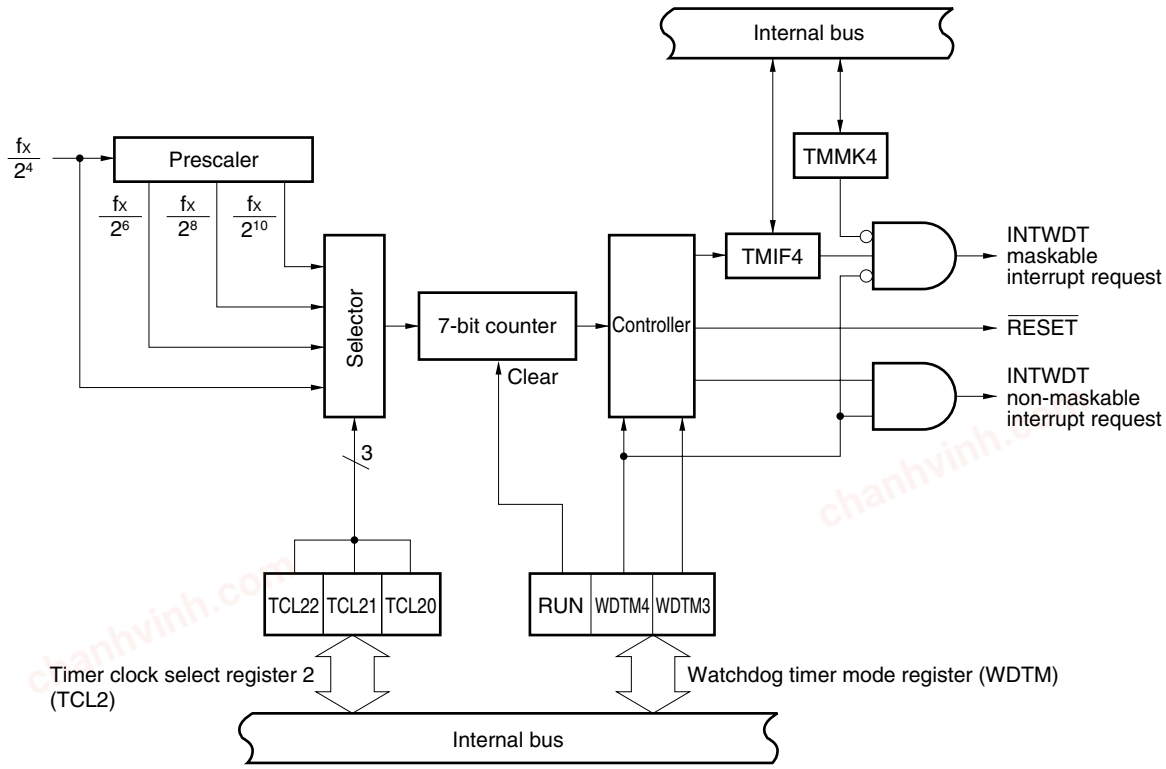
7.2 Watchdog Timer Configuration

The watchdog timer consists of the following hardware.

Table 7-3. Configuration of Watchdog Timer

Item	Configuration
Control register	Timer clock select register 2 (TCL2) Watchdog timer mode register (WDTM)

Figure 7-1. Block Diagram of Watchdog Timer



chanhvinh.com

7.3 Registers Controlling Watchdog Timer

The following two registers are used to control the watchdog timer.

- Timer clock select register 2 (TCL2)
- Watchdog timer mode register (WDTM)

(1) Timer clock select register 2 (TCL2)

This register sets the watchdog timer count clock.

TCL2 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TCL2 to 00H.

Figure 7-2. Format of Timer Clock Select Register 2

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TCL2	0	0	0	0	0	TCL22	TCL21	TCL20	FF42H	00H	R/W

TCL22	TCL21	TCL20	Watchdog timer count clock selection	Interval time
0	0	0	$f_X/2^4$ (375 kHz)	$2^{11}/f_X$ (341 μs)
0	1	0	$f_X/2^6$ (93.8 kHz)	$2^{13}/f_X$ (1.37 ms)
1	0	0	$f_X/2^8$ (23.4 kHz)	$2^{15}/f_X$ (5.46 ms)
1	1	0	$f_X/2^{10}$ (5.86 kHz)	$2^{17}/f_X$ (21.8 ms)
Other than above			Settings prohibited	

Remarks 1. f_X: System clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 6.0 MHz.

chanhvinh.com

(2) Watchdog timer mode register (WDTM)

This register sets the operation mode of the watchdog timer, and enables/disables counting of the watchdog timer.

The WDTM is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the WDTM to 00H.

Figure 7-3. Format of Watchdog Timer Mode Register

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
WDTM	RUN	0	0	WDTM4	WDTM3	0	0	0	FFF9H	00H	R/W

RUN	Selection of operation of watchdog timer[Note 1]
0	Stop counting
1	Clear counter and start counting

WDTM4	WDTM3	Selection of operation mode of watchdog timer[Note 2]
0	0	Operation disabled
0	1	Interval timer mode (overflow and maskable interrupt occur)[Note 3]
1	0	Watchdog timer mode 1 (overflow and non-maskable interrupt occur)
1	1	Watchdog timer mode 2 (overflow occurs and reset operation started)

Notes 1. Once RUN has been set (1), it cannot be cleared (0) by software. Therefore, when counting is started, it cannot be stopped by any means other than $\overline{\text{RESET}}$ input.

2. Once WDTM3 and WDTM4 have been set (1), they cannot be cleared (0) by software.

3. The watchdog timer starts operations as an interval timer when RUN is set to 1.

Cautions 1. When the watchdog timer is cleared by setting RUN to 1, the actual overflow time is up to 0.8% shorter than the time set by timer clock select register 2 (TCL2).

2. In watchdog timer mode 1 or 2, set WDTM4 to 1 after confirming TMIF4 (bit 0 of the interrupt request flag register 0 (IF0)) is set to 0. While TMIF4 is 1, a non-maskable interrupt is generated upon write completion if watchdog timer mode 1 or 2 is selected.

chanhvinh.com

7.4 Watchdog Timer Operation

7.4.1 Operation as watchdog timer

The watchdog timer detects an inadvertent program loop when bit 4 (WDTM4) of the watchdog timer mode register (WDTM) is set to 1.

The count clock (inadvertent loop detection time interval) of the watchdog timer can be selected by bits 0 to 2 (TCL20 to TCL22) of timer clock select register 2 (TCL2). By setting bit 7 (RUN) of WDTM to 1, the watchdog timer is started. Set RUN to 1 within the set inadvertent loop detection time interval after the watchdog timer has been started. By setting RUN to 1, the watchdog timer can be cleared and start counting. If RUN is not set to 1, and the inadvertent loop detection time is exceeded, the system is reset or a non-maskable interrupt is generated by the value of bit 3 (WDTM3) of WDTM.

The watchdog timer continues operation in the HALT mode, but stops in the STOP mode. Therefore, set RUN to 1 before entering the STOP mode to clear the watchdog timer, and then execute the STOP instruction.

Caution The actual inadvertent loop detection time may be up to 0.8% shorter than the set time.

Table 7-4. Inadvertent Loop Detection Time of Watchdog Timer

TCL22	TCL21	TCL20	Inadvertent Loop Detection Time	Operation at fx = 6.0 MHz
0	0	0	$2^{11} \times 1/fx$	341 μs
0	1	0	$2^{13} \times 1/fx$	1.37 ms
1	0	0	$2^{15} \times 1/fx$	5.46 ms
1	1	0	$2^{17} \times 1/fx$	21.8 ms

fx: System clock oscillation frequency

chanhvinh.com

7.4.2 Operation as interval timer

When bit 4 (WDTM4) and bit 3 (WDTM3) of the watchdog timer mode register (WDTM) are set to 0 and 1, respectively, the watchdog timer also operates as an interval timer that repeatedly generates an interrupt at time intervals specified by a count value set in advance.

Select the count clock (or interval time) by setting bits 0 to 2 (TCL20 to TCL22) of timer clock select register 2 (TCL2). The watchdog timer starts operation as an interval timer when the RUN bit (bit 7 of WDTM) is set to 1.

In the interval timer mode, the interrupt mask flag (TMMK4) is valid, and a maskable interrupt (INTWDT) can be generated. The priority of INTWDT is set as the highest of all the maskable interrupts.

The interval timer continues operation in the HALT mode, but stops in the STOP mode. Therefore, set RUN to 1 before entering the STOP mode to clear the interval timer, and then execute the STOP instruction.

Cautions

1. **Once bit 4 (WDTM4) of WDTM is set to 1 (when the watchdog timer mode is selected), the interval timer mode is not set, unless the $\overline{\text{RESET}}$ signal is input.**
2. **The interval time immediately after the setting by WDTM may be up to 0.8% shorter than the set time.**

Table 7-5. Interval Time of Interval Timer

TCL22	TCL21	TCL20	Interval Time	Operation at fx = 6.0 MHz
0	0	0	$2^{11} \times 1/fx$	341 μs
0	1	0	$2^{13} \times 1/fx$	1.37 ms
1	0	0	$2^{15} \times 1/fx$	5.46 ms
1	1	0	$2^{17} \times 1/fx$	21.8 ms

fx: System clock oscillation frequency

chanhvinh.com

CHAPTER 8 USB FUNCTION

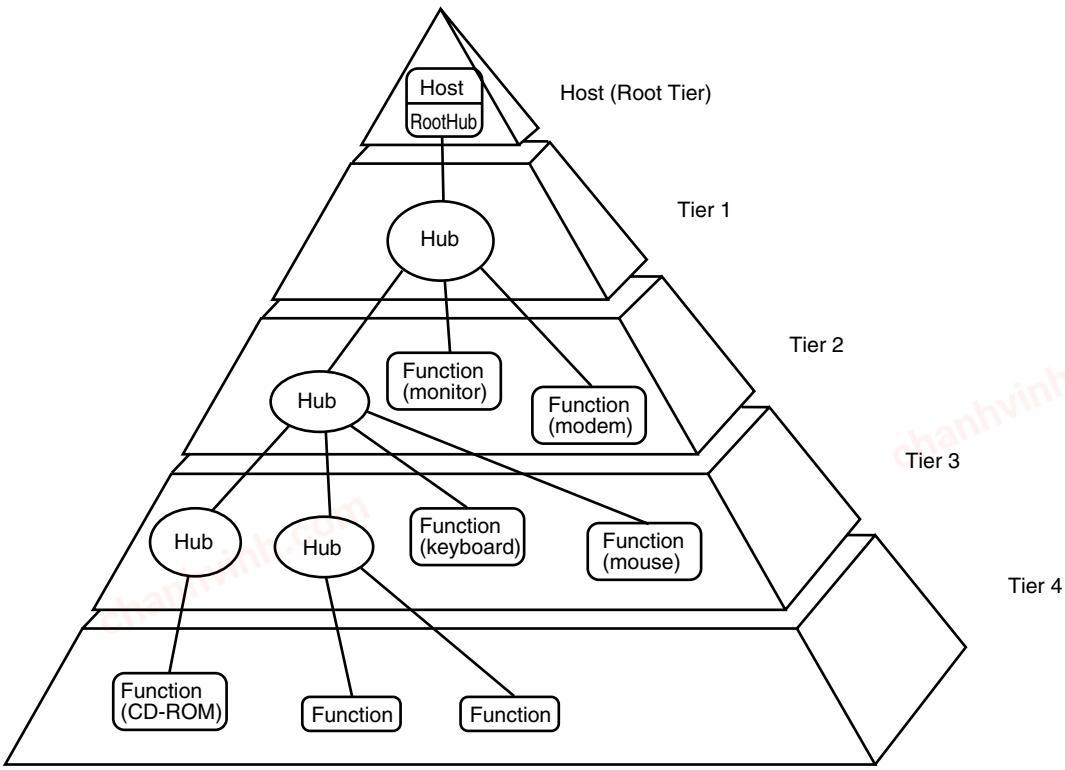
8.1 USB Overview

The USB (Universal Serial Bus) is suitable for connecting personal computers and external devices such as audio equipment, keyboards, pointing devices, and telephones. Two data transfer rates, 12 Mbps and 1.5 Mbps, are provided.

Plug & Play can also be realized.

Figure 8-1 shows an example of USB connection to a desktop PC. The USB consists of the host controller installed in the PC, hubs installed for port expansion and connection, and functions installed at bus ends. These functions are called endpoints and are the data transfer destinations or data transfer sources in the USB.

Figure 8-1. USB Bus Topology (Desktop Type PC)



chanhvinh.com

8.2 USB Function Features

The features of the on-chip USB function provided for the μPD789800 Subseries are described below.

(1) Video display devices and human interface devices are assumed to be the target applications. For this reason, only Endpoint 0 for control transfer and Endpoint 1 for interrupt transfer are supported.
(2) 1.5 Mbps (low speed) data transfer using a 6.0 MHz system clock is supported.
(3) The following buffers are provided on-chip.
★ • Receive token bank: 1 bank (3 bytes)
★ • Receive data bank: 1 bank (9 bytes)
★ • Transmit data bank: 2 banks (9 bytes × 2)
(4) NRZI (Non Return to Zero Invert) decode/encode function specified by the USB communication protocol, bit stuffing function, and on-chip CRC (Cyclic Redundancy Check) function are also provided and automatically executed.

8.3 USB Function Configuration

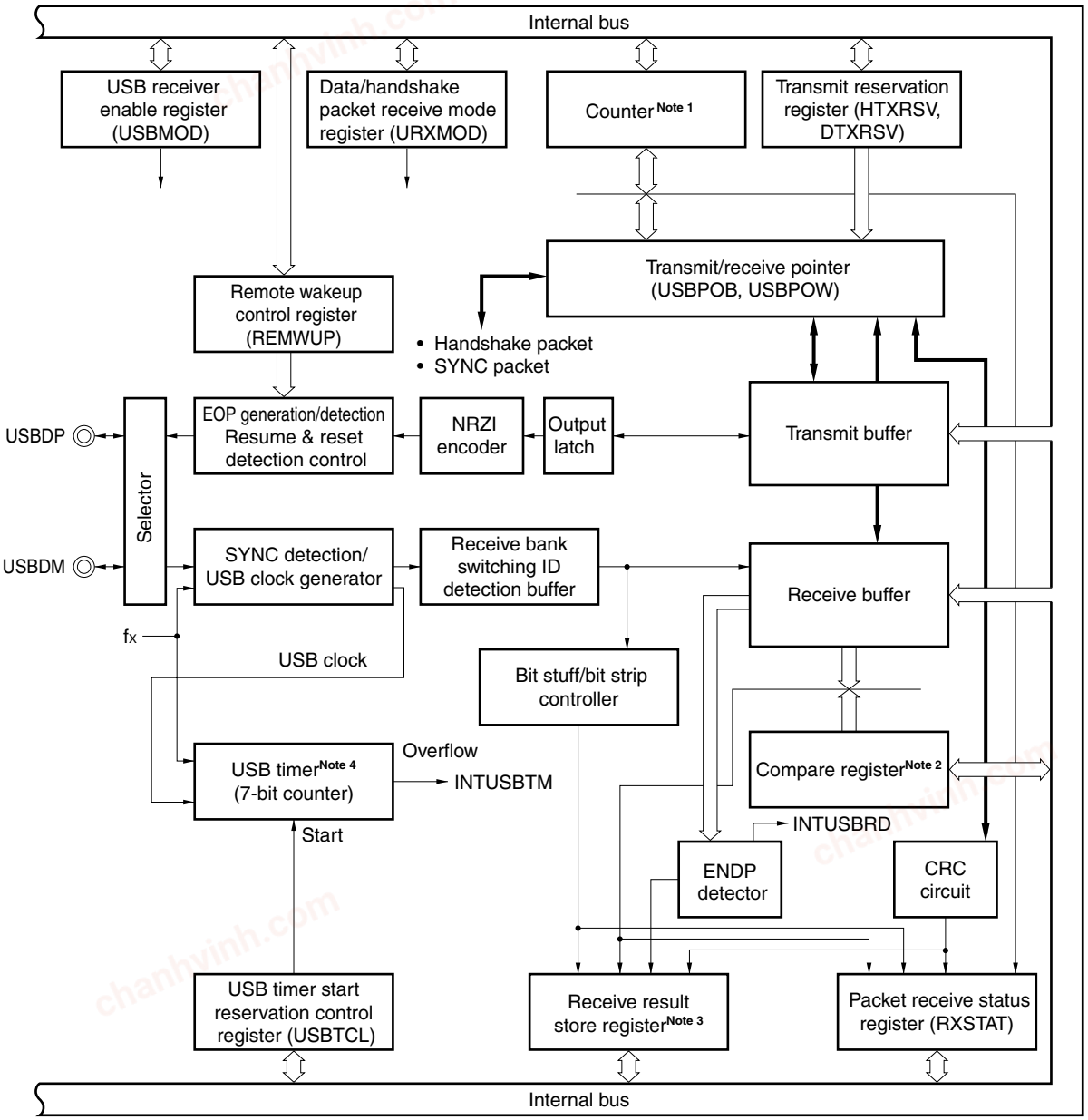
The USB function consists of the following hardware.

Table 8-1. Configuration of USB Function

Item	Configuration
Buffer	Receive bank switching ID detection buffer (internal buffer)
Registers	Transmit/receive pointer (USBPOW) Receive token PID (USBRTP) } Receive token bank Receive token address L, H (USBRAL, USBRAH) } Receive data PID (USBRD) } Receive data bank Receive data address (USBR0 to USBR7) } Transmit data PID bank 0 (USBTD0) } Transmit data bank 0 Transmit data bank 0 address (USBT00 to USBT07) } Transmit data PID bank 1 (USBTD1) } Transmit data bank 1 Transmit data bank 1 address (USBT10 to USBT17) } Data/handshake packet receive byte number counter (DRXCON) Data packet transmit byte number counter 0, 1 (DTXCO0, DTXCO1) Token PID compare register (TIDCMP) Token address compare register (ADRCMP) Data/handshake PID compare register (DIDCMP)
Control registers	USB receiver enable register (USBMOD) Data/handshake packet receive mode register (URXMOD) Packet receive status register (RXSTAT) Data/handshake packet receive result store register (DRXRSL) Token packet receive result store register (TRXRSL) Data packet transmit reservation register (DTXRSV) Handshake packet transmit reservation register (HTXRSV) USB timer start reservation control register (USBTCL) Remote wakeup control register (REMWUP)

chanhvinh.com

Figure 8-2. Block Diagram of USB Function



Notes 1. Data/handshake packet receive byte number counter (DRXCON), data packet transmit byte number counter 0, 1 (DTXCO0, DTXCO1)

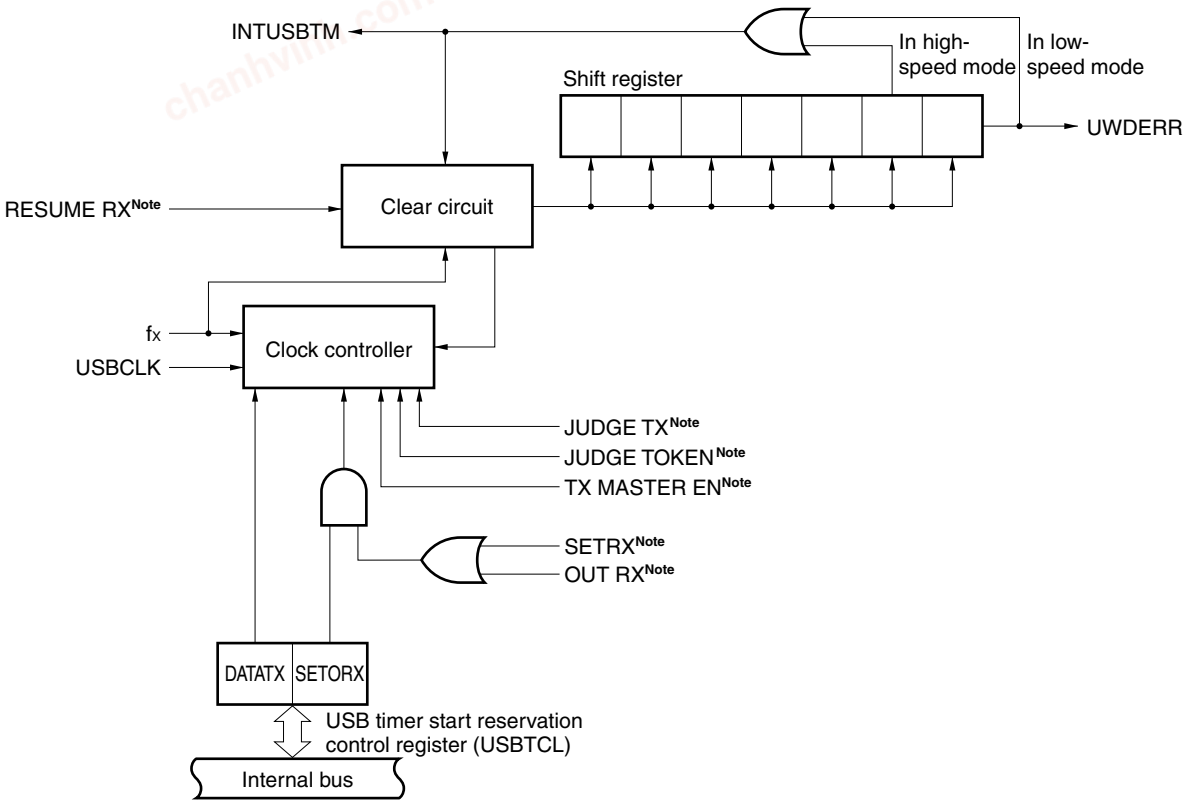
2. Token address compare register (ADRCMP), token PID compare register (TIDCMP), data/handshake PID compare register (DIDCMP)

3. Token packet receive result store register (TRXRSL), data/handshake packet receive result store register (DRXRSL)

4. See **Figure 8-3** for USB timer configuration.

chanhvinh.com

Figure 8-3. Block Diagram of USB Timer



Note As these signals are used internally, confirmation by software is not possible.

Remark fx: System clock oscillation frequency
UWDERR: Bit 7 of packet receive status register (RXSTAT)

chanhvinh.com

(1) Receive bank switching ID detection buffer (internal buffer)

This is an internal 2-bit buffer placed before a receive buffer. It detects the lower 2 bits below the packet ID during packet reception and determines the store bank of a packet.

The following controls are performed depending on the stored 2-bit data. For details, see Section **8.5.3 Receive bank switching ID detection buffer operation**.

<1> If the first 2 bits of the stored bits (lower 2 bits in ID area) are 01B, TOSTAT (bit 0 of the packet receive status register (RXSTAT)) indicating token packet reception is set and a signal specifying packet store at the receive token address is output to the transmit/receive pointer.

<2> If the first 2 bits of the stored bits are 11B, DASTAT (bit 1 of RXSTAT) indicating data packet reception is set and a signal specifying packet store at the receive data address is sent to the transmit/receive pointer.

<3> If the first 2 bits of the stored bits are 10B, HSSTAT (bit 2 of RXSTAT) indicating handshake packet reception is set and a signal specifying packet store at the receive data address is output to the transmit/receive pointer.

(2) Transmit/receive pointer (USBPOB and USBPOW)

The USBPOB is a pointer on the bit side in the transmit/receive buffer and the USBPOW is a pointer on the word side of the transmit/receive buffer. USBPOB and USBPOW output a control signal to the CRC circuit, etc.

They are reset and started by the packet ID detection signal from the receive bank switching ID detection buffer.

USBPOB is incremented by the USB clock. USBPOW is incremented by USBPOB overflow.

USBPOW is read with an 8-bit memory manipulation instruction. As USBPOB is an internal pointer, control with software is not possible.

$\overline{\text{RESET}}$ input sets these pointers to 00H.

The value of USBPOW is changed as follows depending on the receive/transmit byte length match signal or transmit reservation. Moreover, control signals are also output. For details, see Section **8.8.1 Operation of transmit/receive pointer**.

- If the token packet receive signal is detected by the receive bank switching ID detection buffer, the pointers are set to 00H.
- If the data/handshake packet receive signal is detected by the receive bank switching ID detection buffer, the pointers are set to 10H.
- If USBPOW is set to 01H, a signal specifying CRC5 (CRC5 bit mode) execution start is output.
- If USBPOW is set to 11H, 21H, or 31H, a signal specifying CRC16 (CRC16 bit mode) execution start is output.
- If USBPOB is set to 02H after USBPOW is set to 02H, a signal specifying CRC5 comparison start is output and USBPOW is set to 70H.
- If the value of USBPOW matches that of the data/handshake packet receive byte number counter (DRXCON), a signal specifying CRC16 comparison start is output when USBPOB overflows, and USBPOW is set to 70H.
- If a signal specifying transmit start is received from the transmit controller, USBPOW is set to 7FH. After that, USBPOW is set to 20H, 30H, 40H, 50H, or 60H depending on the error between the present transmit reservation and previous receive data, when USBPOB overflows.
- If the value of USBPOW matches that of data packet transmit byte number counter 0 (DTXCO0) or data packet transmit byte number counter 1 (DTXCO1), USBPOW is set to 70H when USBPOB overflows (CRC redundant bits are appended).
- USBPOW is set to 71H, then a signal specifying EOP transmission is output when USBPOB overflows.
- When USBPOW is set to 40H, 50H, or 60H, a signal specifying EOP transmission is output if USBPOB overflows.

chanhvinh.com

(3) Receive token bank

(a) Receive token PID (USBRTP)

This is the receive token packet ID area. The data input to the token PID compare register (TIDCMP) is stored here.

USBRTP is read with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets USBRTP to 00H.

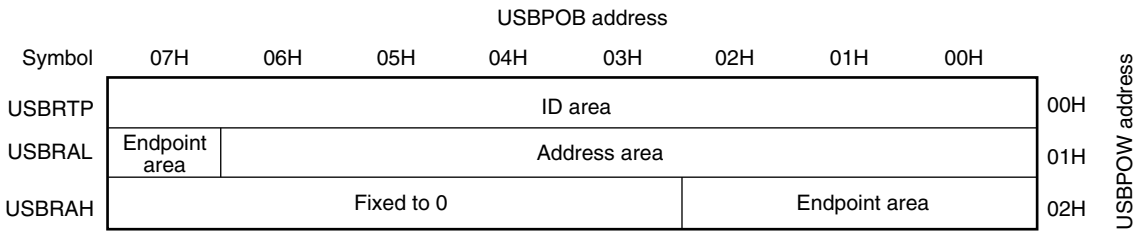
(b) Receive token address L and H (USBRAL and USBRAH)

This stores the token packet to be transferred from the host. USBRAL and USBRAH consist of 16 bits. Bits 0 to 6 of USBRAL store the data input to the token address compare register (ADRCMP).

Both USBRAL and USBRAH are read with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets these registers to 00H.

Figure 8-4. Configuration of Receive Token Bank



chanhvinh.com

(4) Receive data bank

(a) Receive data PID (USBRD)

This is the receive data packet ID area. The data input to the data/handshake PID compare register (DIDCMP) is stored here.

USBRD is read with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets USBRD to 00H.

(b) Receive data address (USBR0 to USBR7)

This is an 8-byte register that stores the data/handshake packet transferred from the host.

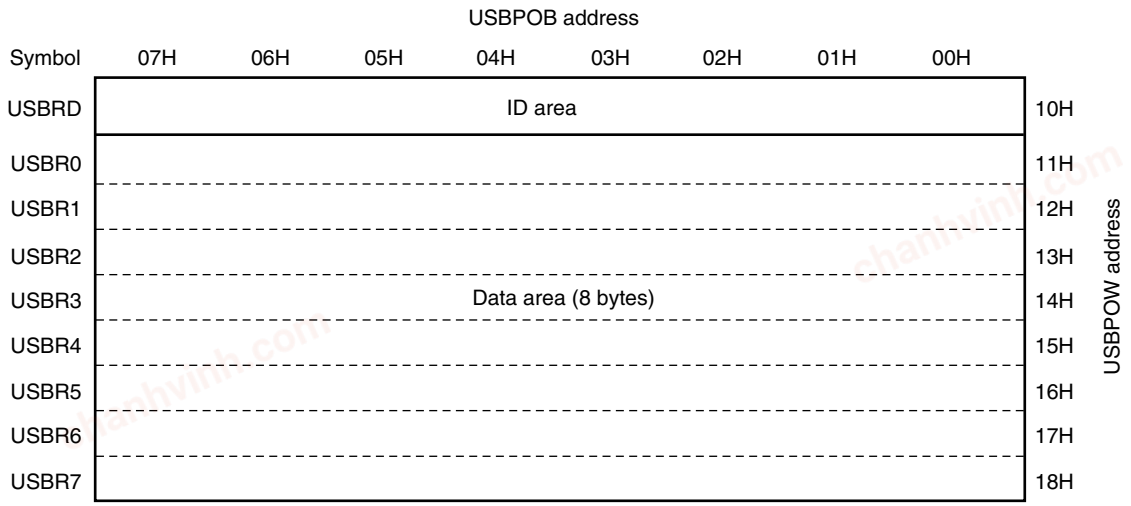
USBR0 to USBR7 are read with an 8-bit memory manipulation instruction.

If the following combinations are used, they are read with a 16-bit memory manipulation instruction.

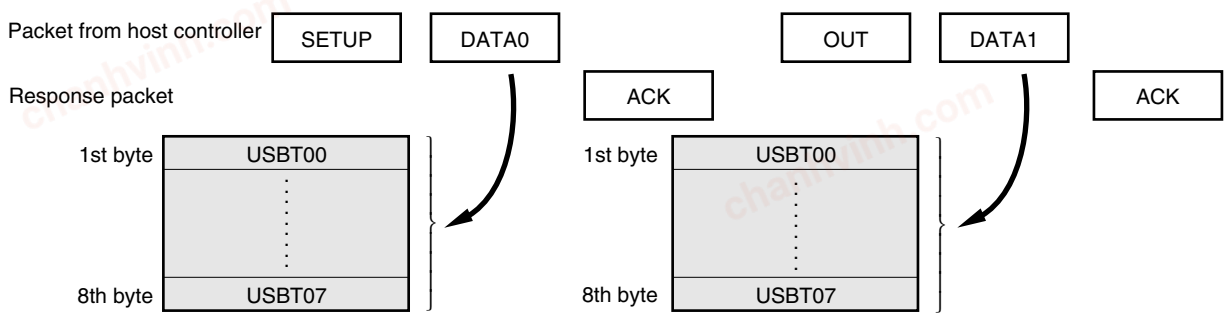
- USBR10: USBR0 and USBR1
- USBR32: USBR2 and USBR3
- USBR54: USBR4 and USBR5
- USBR76: USBR6 and USBR7

$\overline{\text{RESET}}$ input makes USBR0 to USBR7 undefined.

Figure 8-5. Configuration of Receive Data Bank



★ The operation during reception appears as follows.



The data packet from the host controller is stored in the USBT00 to USBT07 registers.

chanhvinh.com

(5) Transmit data banks 0 and 1

(a) Transmit data PID banks 0 and 1 (USBTD0 and USBTD1)

USBTD0 and USBTD1 correspond to the transmit buffer 0 ID area and transmit buffer 1 ID area, respectively. USBTD0 and USBTD1 store DATA0 (C3H) or DATA1 (4BH).

USBTD0 and USBTD1 are set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes both USBTD0 and USBTD1 undefined.

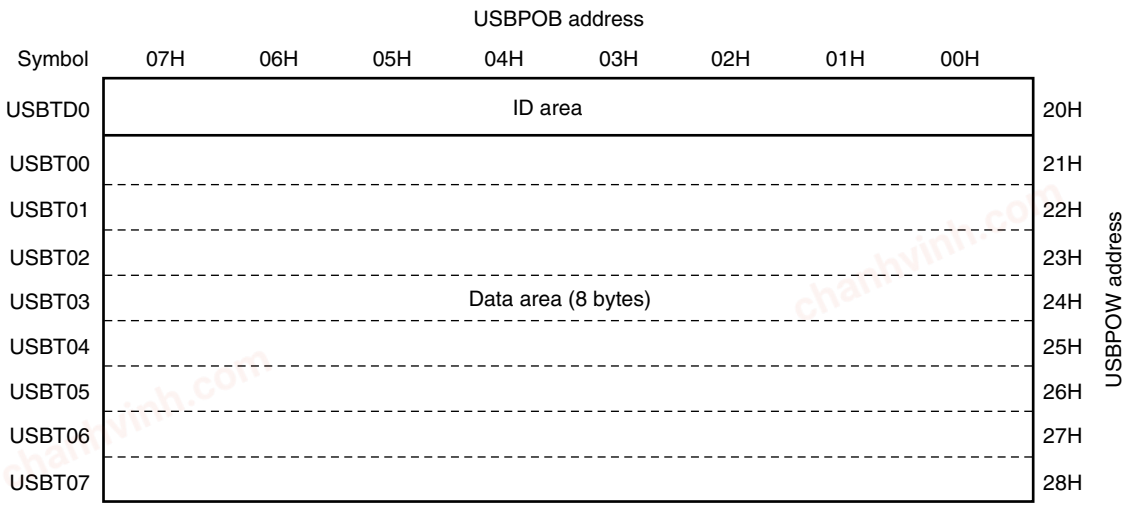
(b) Transmit data bank 0 address (USBT00 to USBT07) and transmit data bank 1 address (USBT10 to USBT17)

These are 8-byte registers that store the data to be transferred to the host. USBT00 to USBT07 and USBT10 to USBT17 correspond to transmit buffer 0 of the data area and transmit buffer 1 of the data area, respectively. Because CRC redundant bits (16 bits) are always appended to packets sent from these registers, these registers cannot be used for transmitting handshake packets.

USBT00 to USBT07 and USBT10 to USBT17 are set with an 8-bit memory manipulation instruction.

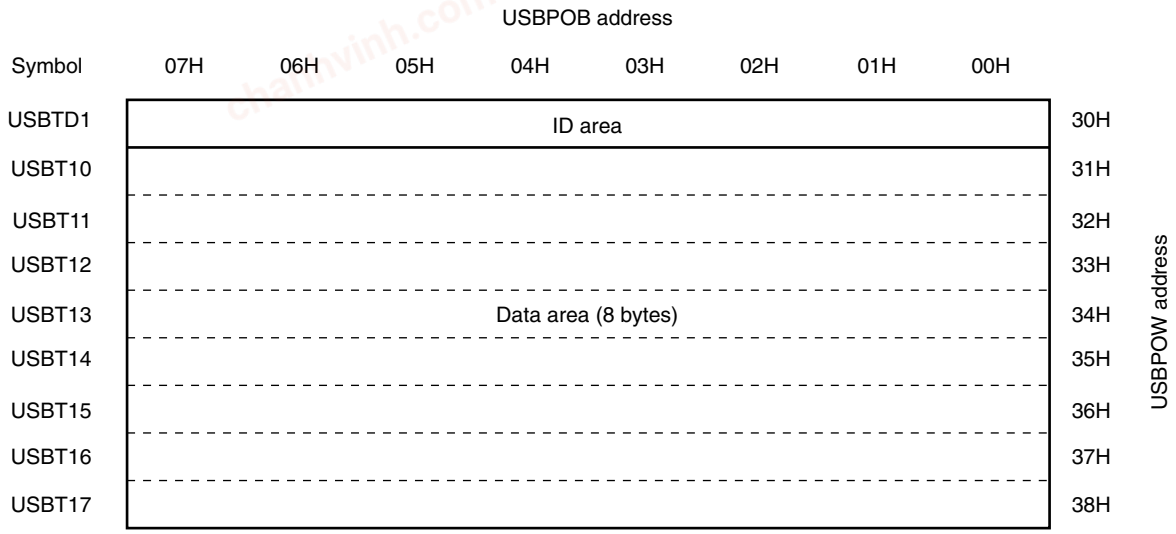
$\overline{\text{RESET}}$ input makes this area undefined.

Figure 8-6. Configuration of Transmit Data Bank 0 (Buffer 0)

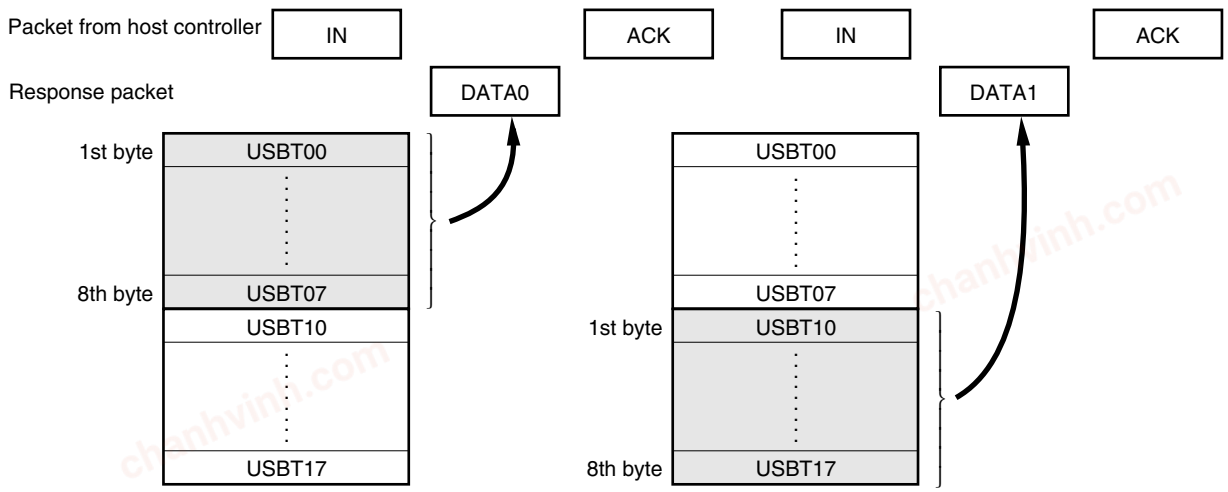


chanhvinh.com

Figure 8-7. Configuration of Transmit Data Bank 1 (Buffer 1)



★ The operation during transmission appears as follows.



Data is read according to the data sequence in the control read data stage and is transmitted to the host. In the DATA0 sequence, the values saved in USBT00 to USBT07 are transmitted in sequence to the host. In the DATA1 sequence, the values saved in USBT10 to USBT17 are transmitted in sequence to the host.

chanhvinh.com

(6) Data/handshake packet receive byte number counter (DRXCON)

This register sets the number of data of the data/handshake packet to be received. During data/handshake packet reception, if this register value and the transmit/receive pointer (USBPOW) value match, a match signal is output from the comparator.

During data packet reception, set the USBPOW address at which the last byte before the appended CRC redundant bits is stored to DRXCON. When a handshake packet is received, set DRXCON to 10H.

DRXCON is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets DRXCON to 18H.

★ SETUP reception[Note] also sets DRXCON to 18H.

Note SETUP reception implies the satisfaction of all the following three conditions.

- Matching of address
- Endpoint 0 received
- No error in reception

(7) Data packet transmit byte number counters 0 and 1 (DTXCO0 and DTXCO1)

DTXCO0 sets the data packet data number of transmit data bank 0 and DTXCO1 sets the transmit data number of transmit data bank 1. During data packet transmission, if these register values and the transmit/receive pointer (USBPOW) value match, a match signal is output from the comparator.

The value to be set to these registers is the USBPOW address (buffer 0: 20H to 28H, buffer 1: 30H to 38H) at which the last byte before the appended CRC redundant bits is stored.

DTXCO0 and DTXCO1 are set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets DTXCO0 to 20H and DTXCO1 to 30H.

(8) Token PID compare register (TIDCMP)

This register sets the token packet ID to be received. If this register value and the value of the receive token PID (USBRTP) match during token packet reception match, TIDRST (bit 1 of the token packet receive result store register (TRXRSL)) is set.

TIDCMP is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TIDCMP to 00H.

chanhvinh.com

(9) Token address compare register (ADRCMP)

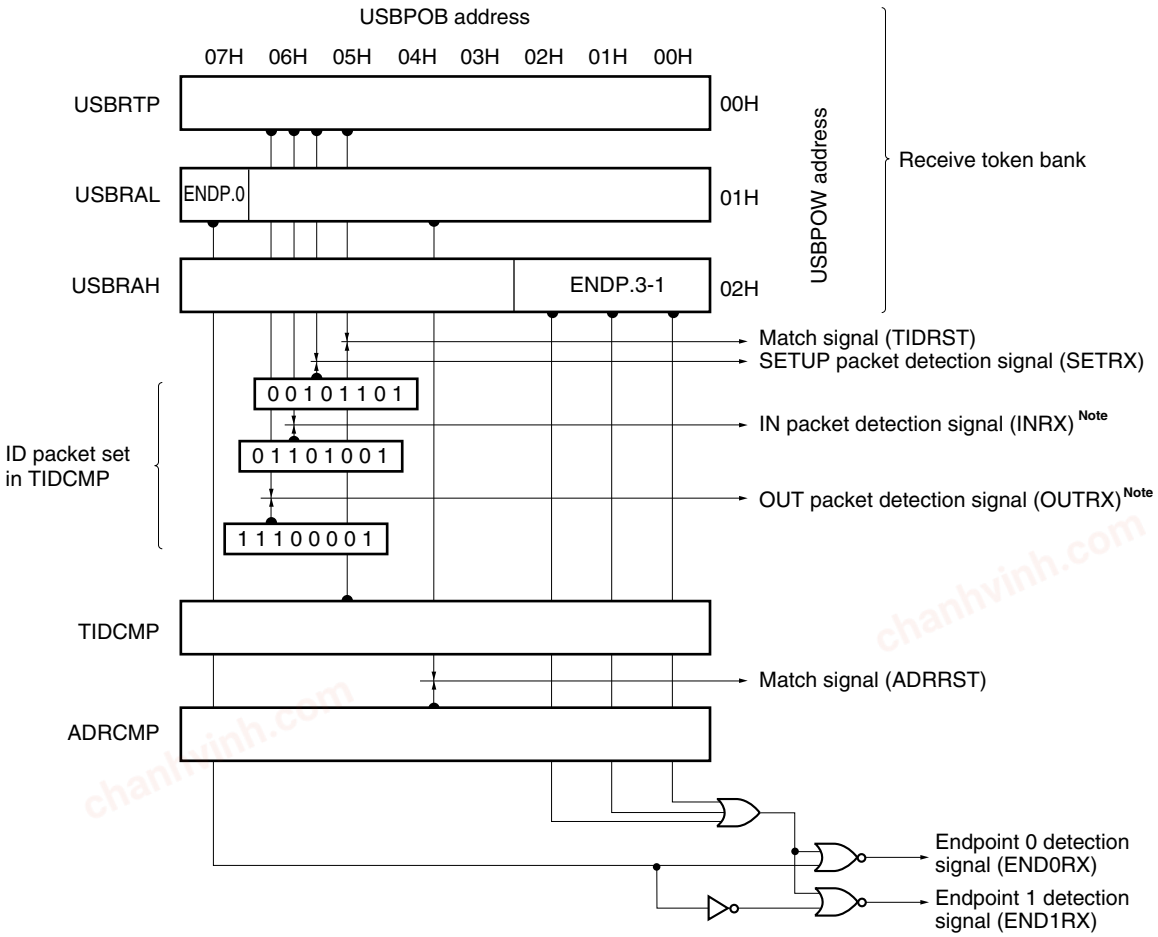
This register sets the address specified from the host during control transfer. If this register value and the address area of the receive token bank (bits 0 to 6 of receive token address L (USBRAL)) match during token packet reception coincide, ADRRST (bit 2 of the token packet receive result store register (TRXRSL)) is set.

00H must be set by software when an USB reset is received.

ADRCMP is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ADRCMP to 00H.

Figure 8-8. Configuration of TIDCMP and ADRCMP



Note Because these signals are used internally, confirmation by software is not possible.

chanhvinh.com

(10) Data/handshake PID compare register (DIDCMP)

This register sets the data/handshake packet ID to be received. If this register value and the value of the receive data PID (USBRD) match during data/handshake packet reception coincide, the DIDRST (bit 1 of the data/handshake packet receive result store register (DRXRSL)) is set.

DIDCMP is set with an 8-bit memory manipulation instruction.

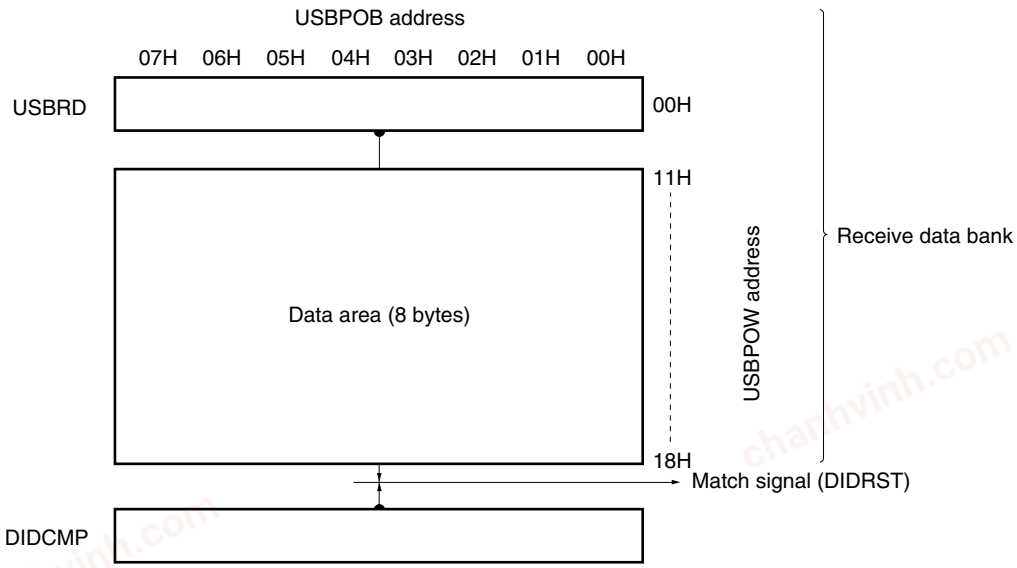
$\overline{\text{RESET}}$ input sets DIDCMP to C3H.

★ SETUP reception[Note] also sets DIDCMP to C3H.

Note SETUP reception implies the satisfaction of all the following three conditions.

- Matching of address
- Endpoint 0 received
- No error in reception

Figure 8-9. Configuration of DIDCMP



chanhvinh.com

8.4 Registers Controlling USB Function

The following nine registers are used to control the USB function.

- USB receiver enable register (USBMOD)
- Data/handshake packet receive mode register (URXMOD)
- Packet receive status register (RXSTAT)
- Data/handshake packet receive result store register (DRXRSL)
- Token packet receive result store register (TRXRSL)
- Data packet transmit reservation register (DTXRSV)
- Handshake packet transmit reservation register (HTXRSV)
- USB timer start reservation control register (USBTCL)
- Remote wakeup control register (REMWUP)

(1) USB receiver enable register (USBMOD)

This register controls USB receiver operation and halt.

Because a single-ended receiver does not have an enable flag, regular operation is possible. Thus reception of the USB reset signal, Resume signal, and EOP signal for bus idle can be performed any time.

USBMOD is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets USBMOD to 00H.

Figure 8-10. Format of USB Receiver Enable Register

Symbol	7	6	5	4	3	2	1	<0>	Address	After reset	R/W
USBMOD	0	0	0	0	0	0	0	RXEN	FF6DH	00H	R/W

RXEN	USB receiver operation control
0	USB receiver operation is stopped.
1	USB receiver operation is possible.

(2) Data/handshake packet receive mode register (URXMOD)

This register sets the data/handshake packet receive mode.

Bit 0 (DWRMSK) is set while saving a data packet and prevents an address greater than 11H in the receive data address from being overwritten when the next packet is received.

★ Bit 1 (DINTEN) is a flag used to set the receive status synchronous interrupt (to generate the INTUSBRD interrupt request to perform data packet reception and data save simultaneously).

Bit 2 (RESMOD) is a flag used to switch the detection mode of the USB reset signal between bus idle mode and bus suspend mode.

URXMOD is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets URXMOD to 00H.

★ SETUP reception[Note] also sets the URXMOD to 00H.

Note SETUP reception implies the satisfaction of all the following three conditions.

- Matching of address
- Endpoint 0 received
- No error in reception

chanhvinh.com

Figure 8-11. Format of Data/Handshake Packet Receive Mode Register

Symbol	7	6	5	4	3	<2>	<1>	<0>	Address	After reset	R/W
URXMOD	0	0	0	0	0	RESMOD	DINTEN	DWRMSK	FF66H	00H	R/W

RESMOD	USB reset signal detection mode setting[Note 1]
0	Reject USB reset signal less than 3.0 μs SE0 (Single-ended 0) period.
1	Detect transition from J state to SE0 as USB reset signal.[Note 2]

★

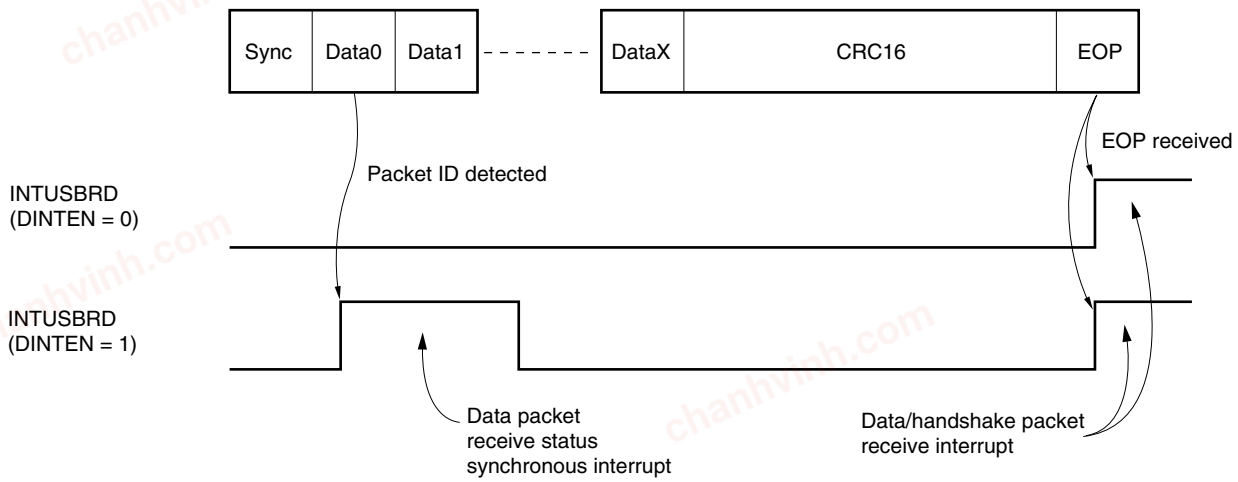
DINTEN	Data packet receive status synchronous interrupt enable flag
0	Do not generate data packet receive status synchronous interrupt.
1	Generate data packet receive status synchronous interrupt [Note 3].

DWRMSK	Data/handshake packet write disable setting
0	Enable write operation to all addresses in data/handshake packet receive buffer.
1	Disable write operation to addresses greater than 11H in data/handshake packet receive buffer.

★ **Notes 1.** Because this is the flag used to detect a USB reset in bus suspend mode, do not set data in bus idle mode. And do not set data immediately before entering bus suspend mode. Clear immediately when returning from the bus suspend mode.

2. If the bus is disturbed by noise, the noise is detected as a USB reset signal. Confirm whether the USB reset signal has been input by checking the URESRX flag (bit 4 of the USB receive status register (RXSTAT)) more than once by software.

★ **3.** The receive status synchronous interrupt occurs at the following timing during data/handshake packet receive interrupt (INTUSBRD) signal input.



chanhvinh.com

(3) Packet receive status register (RXSTAT)

This register indicates the receive status of each packet.

Bits 0 to 2 (TOSTAT, DASTAT, and HSSTAT) are flags that indicate that a token packet, data packet, or handshake packet is currently being received. These flags are set upon detection of a packet ID by an ID detection buffer, and cleared upon reception of EOP.

★ Bits 3 to 6 (EOPRX, URESRX, SE0RX, RESMRX) are flags that detect bus status transition. These flags are set immediately after each bus transition is detected. These flags are cleared by software but cannot be set to 1 by software.

★ Bit 7 (UWDERR) is set if an inadvertent program loop is detected by the USB timer. The flags are cleared by software. UWDERR cannot be set by software. An inadvertent program loop of the USB timer means a status in which the USB clock does not stop when EOP cannot be detected in a packet received from the host, or when noise on the bus was detected as a bus status transition.

RXSTAT is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets RXSTAT to 00H.

chanhvinh.com

Figure 8-12. Format of Packet Receive Status Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
RXSTAT	UWDERR	RESMRX	SE0RX	URESRX	EOPRX	HSSTAT	DASTAT	TOSTAT	FF67H	00H	R/W Note

UWDERR	USB timer inadvertent program loop detection
0	No USB timer inadvertent program loop was detected.
1	USB timer inadvertent program loop (USB clock operation faster than 85.3 μs (at 6.0 MHz)) was detected, forcibly terminating USB clock. This flag is also set when USB reset or Resume signal was received.

RESMRX	Resume signal receive status
0	No Resume signal was received.
1	Received Resume signal (level detection)

SE0RX	Single-ended 0 signal detection status
0	No Single-ended 0 (SE0) signal was detected.
1	Detected SE0 signal one or more times

URESRX	USB reset signal detection status
0	No USB reset signal was detected.
1	Detected USB reset signal one or more times

EOPRX	EOP detection status
0	No EOP signal was detected.
1	Detected EOP one or more times

HSSTAT	Handshake packet receive status
0	No handshake packet was received.
1	Receiving handshake packet

DASTAT	Data packet receive status
0	No data packet was received.
1	Receiving data packet

TOSTAT	Token packet receive status
0	No token packet was received.
1	Receiving token packet

★ **Note** Bits 0 to 2: read only

chanhvinh.com

Table 8-2 shows the state of each flag after receiving the USB reset signal and the Resume signal during the bus idle state and bus suspend state.

Table 8-2. Flag of RXSTAT After Reception of USB Reset Signal and Resume Signal

Bus State	Device State	Received Signal	RESMRX	SE0RX	URESRX
Idle	Main system clock operation mode	USB reset	0	1	1
		Resume	1	1	0
Suspend	STOP mode	USB reset	0	1	1
		Resume	1	1	0

(4) Data/handshake packet receive result store register (DRXRSL)

This register stores the data/handshake packet reception result.

Register contents are updated upon reception of data/handshake EOP.

DRXRSL is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets DRXRSL to 00H.

Figure 8-13. Format of Data/Handshake Packet Receive Result Store Register

Symbol	<7>	<6>	<5>	4	3	2	<1>	0	Address	After reset	R/W
DRXRSL	CR16ER	DBITER	DBYER	0	0	0	DIDRST	0	FF65H	00H	R/W

CR16ER	CRC error detection (16-bit mode)
0	CRC error did not occur in received data packet.
1	CRC error occurred in received data packet.

DBITER	Bit stuffing error detection
0	Bit stuffing error did not occur in received data/handshake packet.
1	Bit stuffing error occurred in received data/handshake packet.

DBYER	Received data/handshake packet length error detection
0	Packet length of received data/handshake packet is normal.
1	Packet length of received data/handshake packet is abnormal.

DIDRST	Data/handshake packet ID comparison result
0	Received data/handshake packet ID and value of data/handshake PID compare register (DIDCMP) do not match.
1	Received data/handshake packet ID and value of DIDCMP match.

(5) Token packet receive result store register (TRXRSL)

This register stores the token packet reception status.

Register contents are updated upon reception of token packet EOP.

TRXRSL is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TRXRSL to 00H.

chanhvinh.com

Figure 8-14. Format of Token Packet Receive Result Store Register

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
TRXRSL	CRC5ER	TBITER	TBYER	END1RX	END0RX	ADRRST	TIDRST	SETRX	FF62H	00H	R/W

CRC5ER	CRC error detection (5-bit mode)
0	CRC error did not occur in received token packet.
1	CRC error occurred in received token packet.

TBITER	Bit stuffing error detection
0	Bit stuffing error did not occur in received token packet.
1	Bit stuffing error occurred in received token packet.

TBYER	Received token packet length error detection
0	Packet length of received token packet is normal.
1	Packet length of received token packet is abnormal.

END1RX	Endpoint 1 reception detection
0	No token packet corresponding to Endpoint 1 is received.
1	Token packet corresponding to Endpoint 1 was received.

END0RX	Endpoint 0 reception detection
0	No token packet corresponding to Endpoint 0 is received.
1	Token packet corresponding to Endpoint 0 was received.

ADRRST	Token packet address compare result
0	Received token packet address and value of token address compare register (ADRCMP) do not match.
1	Received token packet address and value of ADRCMP match.

TIDRST	Token packet ID compare result detection
0	Received token packet ID and value of token PID compare register (TIDCMP) do not match.
1	Received token packet ID and value of TIDCMP match.

SETRX	Setup token packet reception detection
0	Received token packet ID is other than Setup packet.
1	Received token packet ID is Setup packet [Note].

★ **Note** If SETRX is set to 1, the following occurs.

- All reservations are cleared
- DNAEN (bit 1 of the handshake packet transmit reservation register (HTXRSV)) is cleared
- DWRMSK (bit 0 of the data/handshake packet receive mode register (URXMOD)) is cleared

chanhvinh.com

(6) Data packet transmit reservation register (DTXRSV)

This register sets the bank where the data packet to be transmitted is stored. By setting each flag of this register, the stored data is transmitted following normal reception of the IN token packet.

DTXRSV is set with a 1-bit or 8-bit memory manipulation instruction. When DTXRSV is used in combination with the handshake packet transmit reservation register (HTXRSV) as the 16-bit register USBCON, DTXRSV is set with a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets DTXRSV to 00H.

★ SETUP reception[Note] also sets DTXRSV to 00H.

Note SETUP reception implies the satisfaction of all the following three conditions.

- Matching of address
- Endpoint 0 received
- No error in reception

Figure 8-15. Format of Data Packet Transmit Reservation Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
DTXRSV	0	0	0	0	DT11EN	DT10EN	DT01EN	DT00EN	FF15H	00H	R/W

DT11EN	Transmit reservation flag for transmit bank 1 (Endpoint 1)
0	No data is transmitted.
1	Stored data is transmitted when all the following conditions are satisfied in EOP during IN packet reception. Setting is disabled during control read transmission. INRX (Internal signal) = 1, ADRRST = 1, END1RX = 1, TBYER = 0, TBITER = 0, CRC5ER = 0

DT10EN	Transmit reservation flag for transmit bank 1 (Endpoint 0)
0	No data is transmitted.
1	Stored data is transmitted when all the following conditions are satisfied in EOP during IN packet reception. INRX (internal signal) = 1, TIDRST = 1, ADRRST = 1, END0RX = 1, TBYER = 0, TBITER = 0, CRC5ER = 0

DT01EN	Transmit reservation flag for transmit buffer 0 (Endpoint 1)
0	No data is transmitted.
1	Stored data is transmitted when all the following conditions are satisfied in EOP during IN packet reception. Setting is disabled during control read transmission. INRX (internal signal) = 1, ADRRST = 1, END1RX = 1, TBYER = 0, TBITER = 0, CRC5ER = 0

DT00EN	Transmit reservation flag for transmit buffer 0 (Endpoint 0)
0	No data is transmitted.
1	Stored data is transmitted when all the following conditions are satisfied in EOP during IN packet reception. INRX (internal signal) = 1, TIDRST = 1, ADRRST = 1, END0RX = 1, TBYER = 0, TBITER = 0, CRC5ER = 0

★ **Caution Setting bits 1 and 3 (DT01EN, DT11EN) is prohibited during control transfer.**

chanhvinh.com

(7) Handshake packet transmit reservation register (HTXRSV)

This register sets the handshake packet to be transmitted. By setting each flag of this register, a handshake packet is transmitted following normal reception of an IN packet, or normal or abnormal reception of a data packet.

Bit 0 corresponds to the ACK packet transmit reservation flag, bits 1 to 3 correspond to the NAK packet transmit reservation flag, and bits 4 to 7 correspond to the STALL packet transmit reservation flag.

HTXRSV is set with a 1-bit or 8-bit memory manipulation instruction. When HTXRSV is used in combination with the data packet transmit reservation register (DTXRSV) as the 16-bit register USBCON, HTXRSV is set with a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets HTXRSV to 00H.

★ SETUP reception[Note] also sets HTXRSV to 00H.

Note SETUP reception implies the satisfaction of all the following three conditions.

- Matching of address
- Endpoint 0 received
- No error in reception

Figure 8-16. Format of Handshake Packet Transmit Reservation Register (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
HTXRSV	E1STEN	E0STEN	DSTEN	STALEN	E1NAEN	E0NAEN	DNAEN	ACKEN	FF14H	00H	R/W

E1STEN	STALL packet transmit reservation flag for Endpoint 1 after IN packet
0	No data is transmitted.
1	STALL handshake is transmitted when all the following conditions are satisfied in EOP during IN packet reception. INRX (internal signal) = 1, ADRRST = 1, END1RX = 1, TBYER = 0, TBITER = 0, CRC5ER = 0

E0STEN	STALL packet transmit reservation flag for Endpoint 0 after IN packet
0	No data is transmitted.
1	STALL handshake is transmitted when all the following conditions are satisfied in EOP during IN packet reception. INRX (internal signal) = 1, TIDRST = 1, ADRRST = 1, END0RX = 1, TBYER = 0, TBITER = 0, CRC5ER = 0

DSTEN	STALL packet transmitted reservation flag for data packet receive byte length error
0	No data is transmitted.
1	STALL handshake is transmitted when all the following conditions are satisfied in EOP during data packet reception. Set this flag to transmit STALL handshake when byte length error has occurred in one data packet during control write transfer. DIDRST = 1, DBYER = 1, DBITER = 0

STALEN	STALL packet transmit reservation flag after data packet
0	No data is transmitted.
1	STALL handshake is transmitted when all the following conditions are satisfied in EOP during data packet reception. Set this flag when length error of transfer occurs in control write transfer. DIDRST = 0, DBITER = 0

chanhvinh.com

Figure 8-16. Format of Handshake Packet Transmit Reservation Register (2/2)

E1NAEN	NAK packet transmit reservation flag for Endpoint 1 after IN packet
0	No data is transmitted.
1	NAK handshake is transmitted when all the following conditions are satisfied in EOP during IN packet reception. INRX (internal signal) = 1, ADRRST = 1, END1RX = 1, TBYER = 0, TBITER = 0, CRC5ER = 0

E0NAEN	NAK packet transmit reservation flag for Endpoint 0 after IN packet
0	No data is transmitted.
1	NAK handshake is transmitted when all the following conditions are satisfied in EOP during IN packet reception. INRX (internal signal) = 1, TIDRST = 1, ADRRST = 1, END0RX = 0, TBYER = 0, TBITER = 0, CRC5ER = 0

DNAEN	NAK packet transmit reservation flag after data packet reception
0	No data is transmitted.
1	If all the following conditions are met, NAK handshake is transmitted in EOP during data packet reception. Set this flag when saving data from reception data addresses (USBR0 to USBR7). OUTRX (internal signal) = 1, DIDRST = 1, DBYER = 0, DBITER = 0, CR16ER = 0, UWDERR = 0

ACKEN	ACK packet transmit reservation flag after data packet reception
0	No data is transmitted.
1	ACK handshake is transmitted when all the following conditions are satisfied in EOP during data packet reception. DIDRST = 1, DBYER = 0, DBITER = 0, CR16ER = 0

During transmit reservation, all the conditions listed in Table 8-3 below must be satisfied.

Table 8-3. Conditions in Transmit Reservation (1/2)

(a) Transmit reservation for Endpoint 0 and IN token packet

Type of Reservation	DT00EN	DT10EN	E0STEN	E0NAEN
Transmit reservation of data in transmit buffer 0	1	0	0	0
Transmit reservation of data in transmit buffer 1	0	1	0	0
Endpoint 0 STALL transmit reservation (occurrence of length error, or halt status)	0	0	1	0
Endpoint 0 NAK transmit reservation (data creation incomplete)	0	0	0	1
Two or more reservations above	Setting prohibited			

chanhvinh.com

Table 8-3. Conditions in Transmit Reservation (2/2)

(b) Transmit reservation for Endpoint1 and IN token packet

Type of Reservation	DT01EN	DT11EN	E1STEN	E1NAEN
Transmit reservation of data in transmit buffer 0	1	0	0	0
Transmit reservation of data in transmit buffer 1	0	1	0	0
Endpoint 1 STALL transmit reservation (halt status)	0	0	1	0
Endpoint 1 NAK transmit reservation (no transmit data)	0	0	0	1
Two or more reservations above	Setting prohibited			

(c) Handshake transmit reservation for data packet

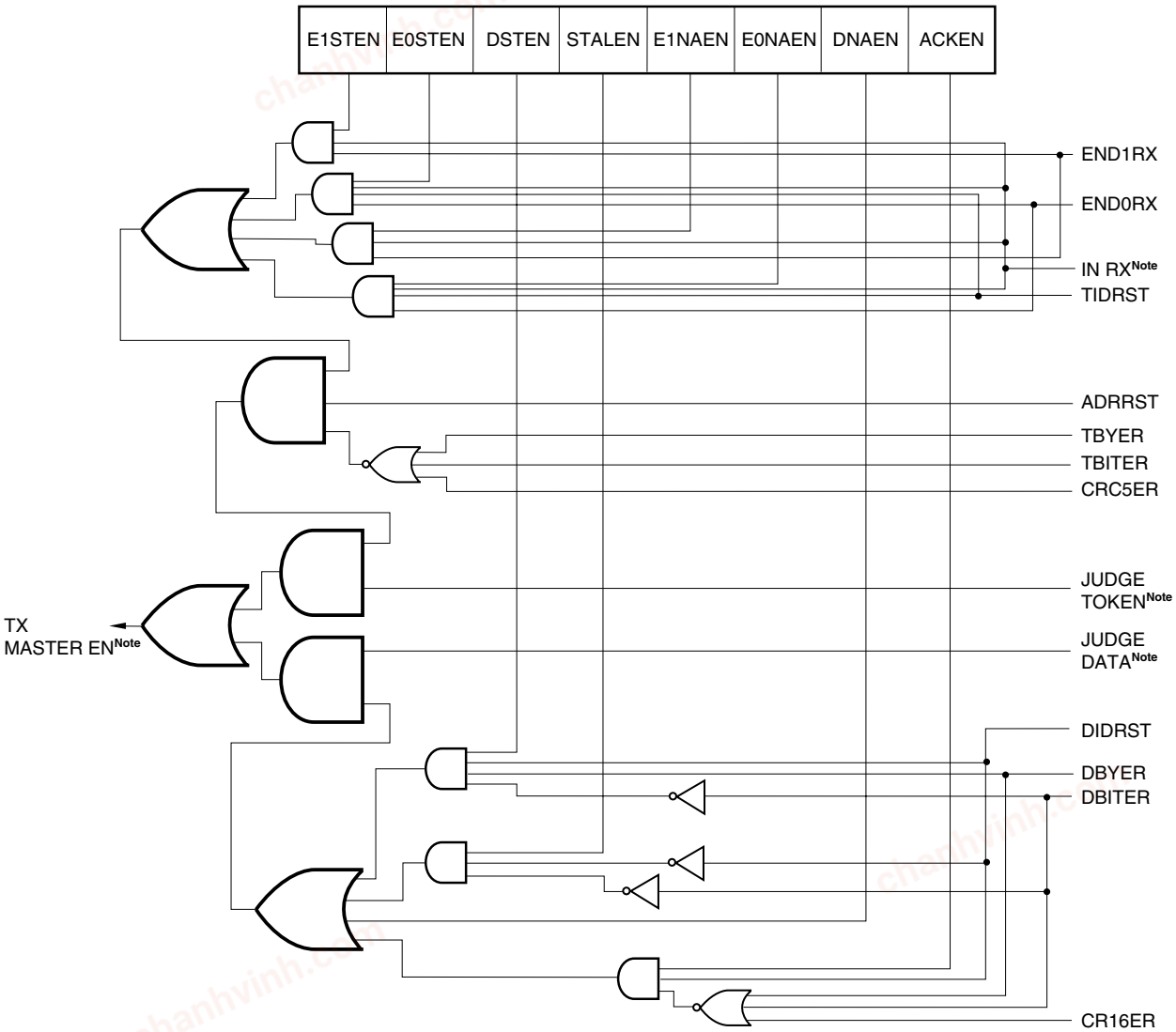
Type of Reservation	STALEN	DNAEN
STALL transmit reservation during occurrence of length error	1	0
NAK transmit reservation during previous receive data save[Note]	0	1
Two or more reservations above	Setting prohibited	

Type of Reservation	ACKEN	DNAEN
ACK transmit reservation during normal data packet reception	1	0
NAK transmit reservation during previous receive data save[Note]	0	1
Two or more reservations above	Setting prohibited	

Note When saving the receive data packet, set DWRMSK to 1 (bit 0 of the data/handshake packet receive mode register (URXMOD)) at the same time.

chanhvinh.com

Figure 8-17. Configuration of Handshake Packet Transmit Reservation Register



Note Because these signals are used internally, confirmation by software is not possible.

chanhvinh.com

(8) USB timer start reservation control register (USBTCL)

This register reserves USB timer start after reception of a SETUP/OUT packet or transmission of a data packet.

USBTCL is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets USBTCL to 01H.

★ SETUP reception[Note] also sets USBTCL to 01H.

Note SETUP reception implies the satisfaction of all the following three conditions.

- Matching of address
- Endpoint 0 received
- No error in reception

Figure 8-18. Format of USB Timer Start Reservation Control Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
USBTCL	0	0	0	0	0	0	DATATX	SETORX	FF6CH	01H	R/W

DATATX	USB timer start reservation after data packet transmission
0	Do not start USB timer.
1	USB timer starts when all the following conditions are satisfied in EOP during data packet transmission. DIDRST = 1, DBYER = 0, DBITER = 0, CR16ER = 0

SETORX	USB timer start reservation after SETUP/OUT token packet reception
0	Do not start USB timer.
1	USB timer starts when all the following conditions are satisfied in EOP during SETUP/OUT token packet reception. OUTRX (internal signal) = 1 or SETRX = 1, ADRRST = 1, END0RX = 1, TBYER = 0, TBITER = 0, CRC5ER = 0

chanhvinh.com

(9) Remote wake-up control register (REMWUP)

This register transmits the Resume signal to perform remote wakeup.

Remote wakeup must be performed after confirming that bus idle has continued longer than 5 ms.

REMWUP is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets REMWUP to 08H.

Figure 8-19. Format of Remote Wakeup Control Register

Symbol	7	6	5	4	<3>	<2>	<1>	<0>	Address	After reset	R/W
REMWUP	0	0	0	0	PULLDM	PULLDP	PULLEN	WAKEUP	FF6AH	08H	R/W

PULLDM	D- lead low/high fixed output setting
0	D- (TXDM) is fixed to low output.
1	D- (TXDM) is fixed to high output.

PULLDP	D+ lead low/high fixed output setting
0	D+ (TXDP) is fixed to low output.
1	D+ (TXDP) is fixed to high output.

PULLEN	D+/D- lead fixed output enable
0	Output from transmit buffer is input to USB driver.
1	Level set to PULLDP or PULLDM is input to USB driver.

WAKEUP	Wakeup signal output
0	No data is sent.
1	Pin status set to PULLDP or PULLDM is output.

chanhvinh.com

8.5 USB Function Operation

8.5.1 USB timer operation

The USB timer is a 7-bit counter that performs time management during packet transmission and reception and inadvertent program loop detection of the USB clock.

The USB timer has two modes: high-speed mode (source clock = fx) and low-speed mode (source clock = USB clock: fx = 1.5 MHz during 6.0 MHz operation). In the high-speed mode, the USB functions as a 6-bit counter.

High-speed mode is used for time management during packet transmission and reception. The timer starts after EOP reception or data packet EOP transmission. The start condition is set by the USB timer start reservation control register (USBTCL).

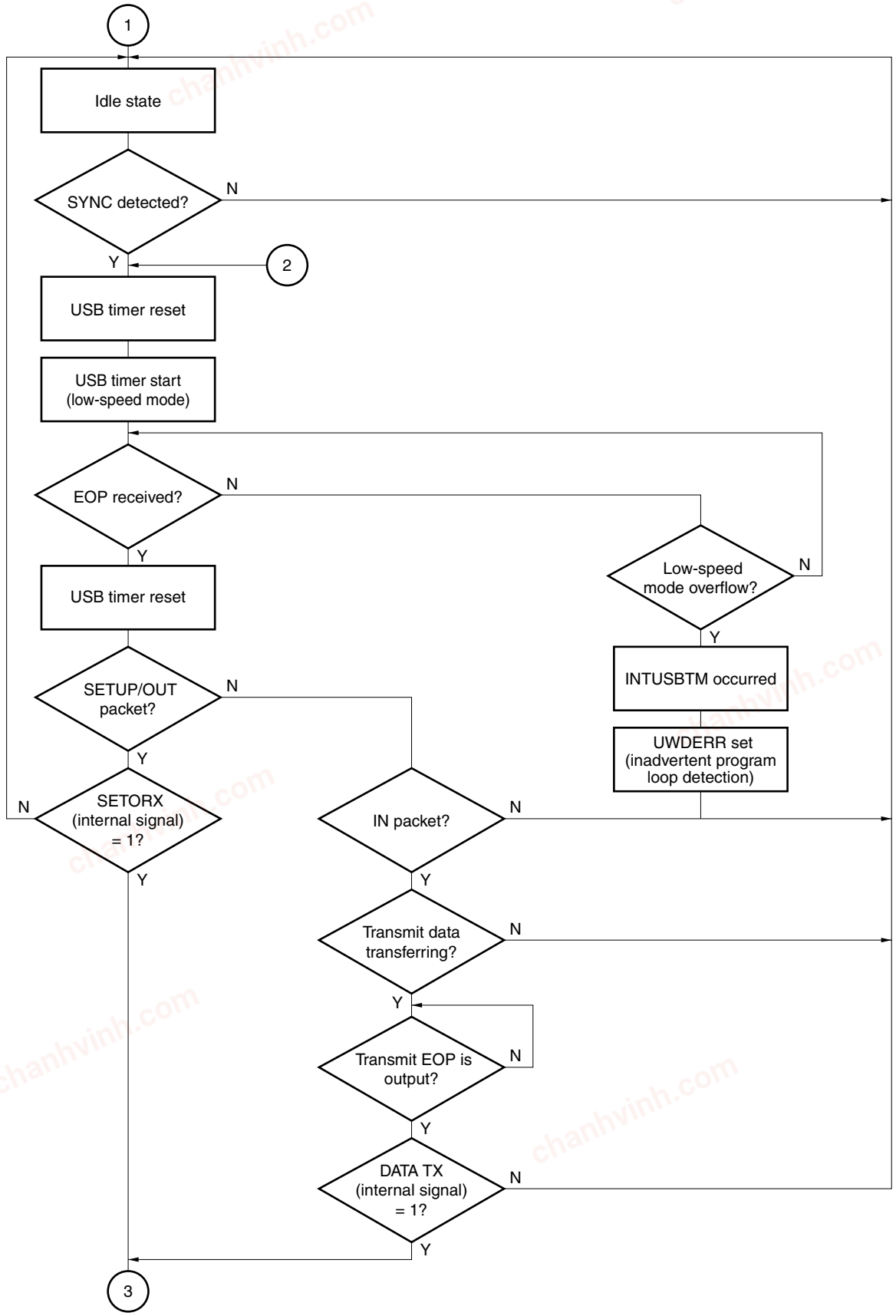
Low-speed mode is used for detecting inadvertent program loops of the USB clock. The timer starts upon detection of the SYNC signal of a receive packet, or upon reception of the USB reset or Resume signals.

When the USB timer overflows, an interrupt request signal (INTUSBTM) is generated, regardless of whether the current mode is the high-speed or low-speed mode. When overflow of the USB timer occurs in the low-speed mode, UWDERR (bit 7 of the packet receive status register (RXSTAT)) is set, allowing detection of an inadvertent loop of the USB timer. At this time, the USB clock is forcibly stopped.

Figure 8-20 shows the operation flowchart of the USB timer.

chanhvinh.com

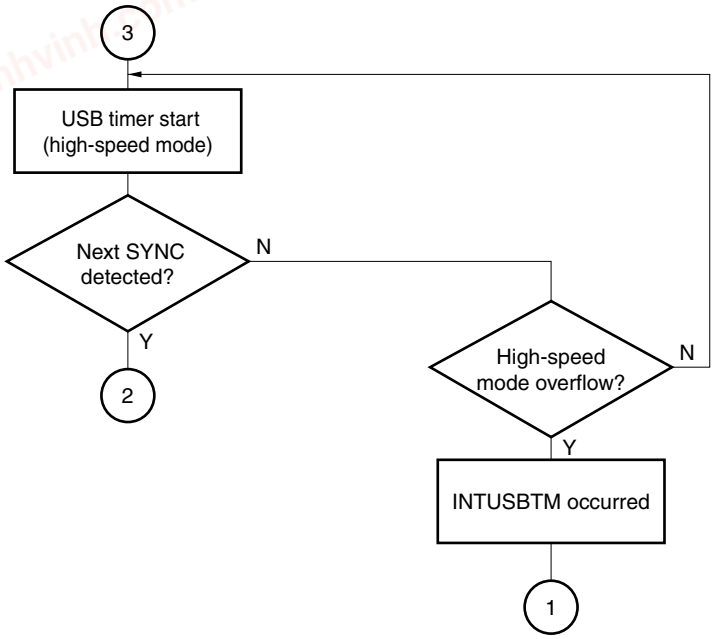
Figure 8-20. Flowchart of USB Timer Operation (1/2)



UWDERR: Bit 7 of packet receive status register (RXSTAT)

chanhvinh.com

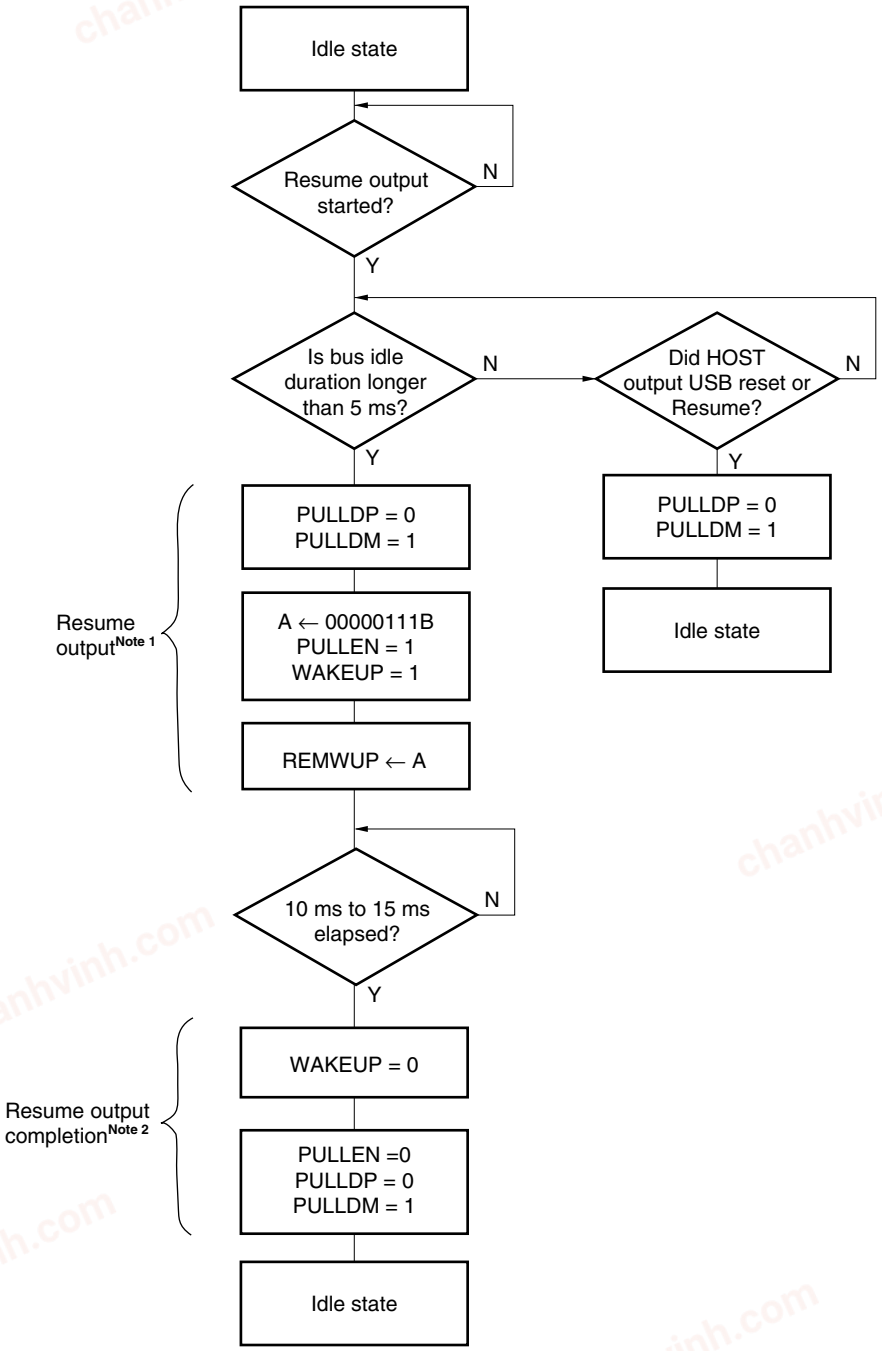
Figure 8-20. Flowchart of USB Timer Operation (2/2)



chanhvinh.com

★ 8.5.2 Remote wakeup control operation

Figure 8-21. Flow Chart of Remote Wakeup Control Operation



PULLDP: Bit 2 of remote wakeup control register (REMWUP)
PULLDM: Bit 3 of remote wakeup control register (REMWUP)
PULLEN: Bit 1 of remote wakeup control register (REMWUP)
WAKEUP: Bit 0 of remote wakeup control register (REMWUP)

chanhvinh.com

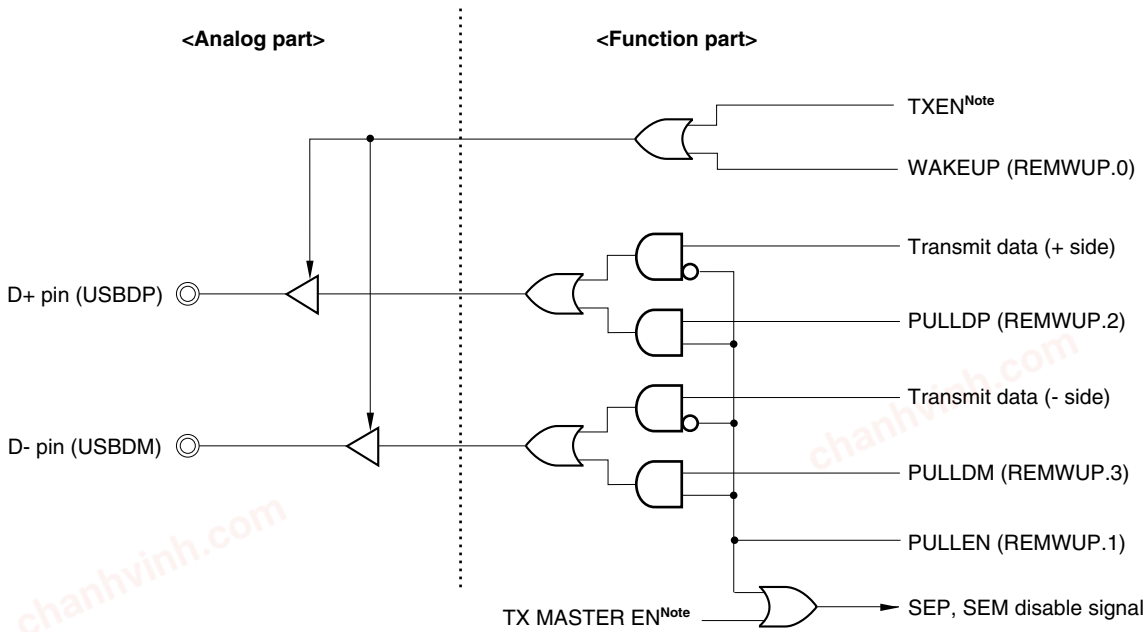
Notes 1. Be sure to follow the exact instruction sequence when the Resume signal (“K” state) is output.

```
SET1    REMWUP.3      ; (PULLDM ← 1)   } “J” state generation
CLR1    REMWUP.2      ; (PULLDP ← 0)   }
MOV     A, #00000111B ; (A ← 00000111B)
SET1    REMWUP.1      ; (PULLEN ← 1)
SET1    REMWUP.0      ; (WAKEUP ← 1),    “J” state output
MOV     REMWUP, A     ; (REMWUP ← A),    “K” state output
```

2. Be sure to follow the exact instruction sequence to append EOP when terminating Resume output.

```
CLR1    REMWUP.0      ; (WAKEUP ← 0)     , Resume output end
CLR1    REMWUP.1      ; (PULLEN ← 0)
CLR1    REMWUP.2      ; (PULLDP ← 0)   } “J” state generation
SET1    REMWUP.3      ; (PULLDM ← 1)   }
```

Figure 8-22. Configuration of Remote Wakeup Control



Note Because this signal is used internally, confirmation by software is not possible.

chanhvinh.com

8.6 Interrupt Request from USB Function

8.6.1 Interrupt sources

Interrupt request sources generated by the USB function fall into the following five categories.

★ **Table 8-4. List of Sources of Interrupts from USB Function**

Type of Interrupt	Priority[Note]	Interrupt Source		Vector Table Address
		Name	Trigger	
Maskable	1	INTUSBTM	USB timer overflow	0006H
	2	INTUSBRT	EOP detection when a USB token packet is received	0008H
	3	INTUSBRD	EOP detection when a USB data/handshake packet is received	000AH
	4	INTUSBST	EOP detection when a USB data/handshake packet is transmitted	000CH
	5	INTUSBRE	Detection of transition from J state to K state or SE0 on the USB bus	000EH

Note The priority is the order of priority when multiple maskable interrupts are generated simultaneously.

(1) Token packet receive interrupt (INTUSBRT)

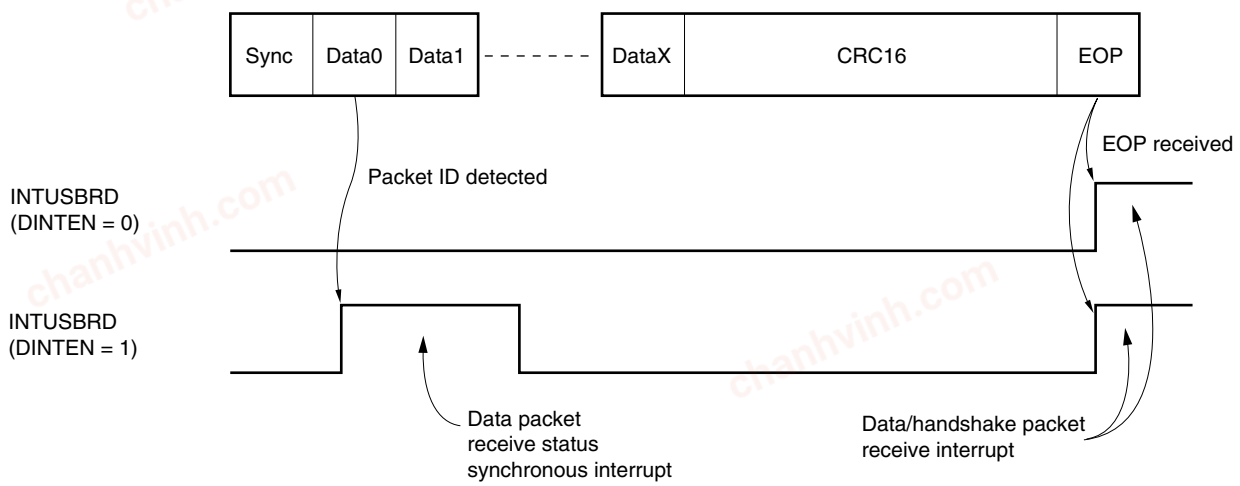
Upon EOP detection during token packet reception, an interrupt request signal is generated and an interrupt request flag (USBRTIF) is set. If ADRRST (bit 2 of the token packet receive result store register (TRXRSL)) is 0, no interrupt request is generated because a token packet of another device exists on the bus.

(2) Data/handshake packet receive interrupt (INTUSBRD)

Upon EOP detection during data/handshake packet reception, an interrupt request signal is generated and an interrupt request flag (USBRDIF) is set regardless of error at reception.

If DINTEN (bit 1 of the data/handshake packet receive mode register (URXMOD)) is set to 1, an interrupt request (receive status synchronous interrupt) signal is generated when 11B is detected by the ID detection buffer.

★ **Figure 8-23. Timing of Data/Handshake Packet Receive Interrupt Request Generation**



DINTEN: Bit 1 of data/handshake packet receive mode register (URXMOD)

chanhvinh.com

(3) Data/handshake packet transmit interrupt (INTUSBST)

Upon EOP detection during data/handshake packet transmission, an interrupt request signal is generated and an ★ interrupt request flag (USBSTIF) is set.

(4) USB timer overflow interrupt (INTUSBTM)

If the USB timer overflows, an interrupt request signal is generated and an interrupt request flag (USBTMIF) is set.

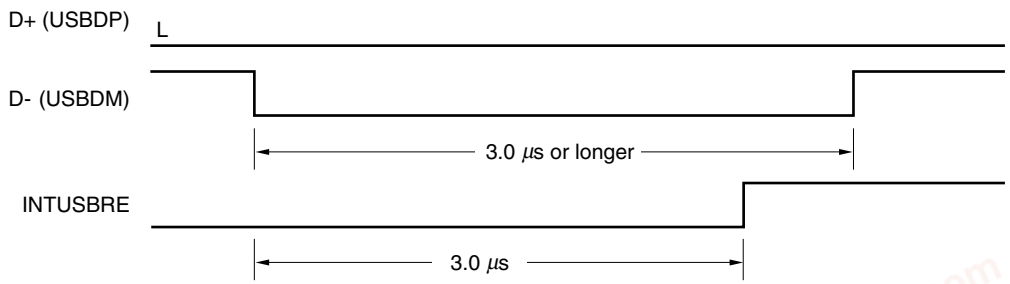
(5) USB reset/Resume detection interrupt (INTUSBRE)

★ This is an interrupt to release the STOP mode.

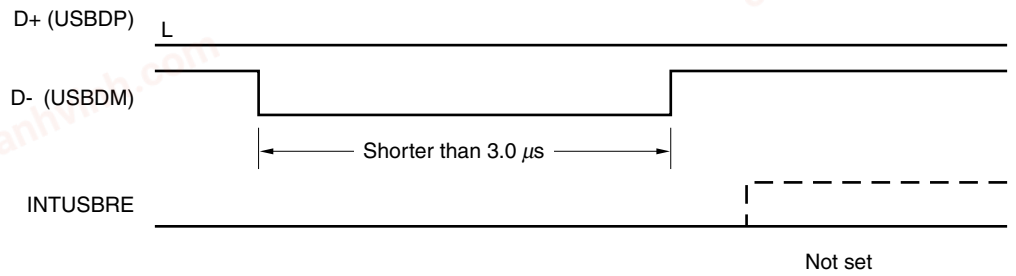
If transition from J state (logic 0) to K state (logic 1) is detected on the bus, or transition to SE0 is detected, an interrupt request signal is generated and an interrupt request flag (USBREIF) is set.

Figure 8-24. Timing of INTUSBRE Generation

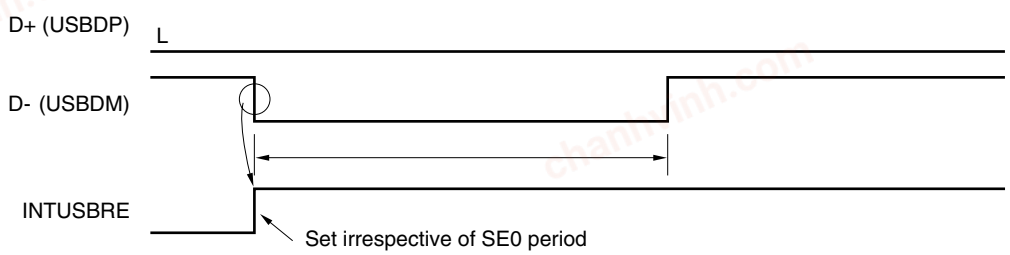
(a) Period of RESMOD = 0 and Single-ended 0 (SE0) is 3.0 μs or longer (microcontroller operation)



(b) Period of RESMOD = 0 and SE0 is shorter than 3.0 μs (microcontroller operation)



(c) RESMOD = 1[Note]



Note Do not set RESMOD to 1 during bus idle. Set RESMOD immediately before the STOP instruction.

Remark RESMOD: Bit 2 of data/handshake packet receive mode register (URXMOD)

chanhvinh.com

8.6.2 Cautions when using interrupts

Pay attention to the following when using an interrupt request generated by the USB function.

(1) Because USBREIF is set by transition from the J state to the K state on the bus, it is also set during sync detection or packet reception. Thus, disable the generation of interrupt requests by setting the interrupt mask flag (USBREMK) during bus idle or control transfer.

★

(2) Because USBREIF is set by transition from the J state to the SE0 state, or by transition from the K state to the SE0 state on the bus, it is also set during EOP reception or bus idle retention EOP reception. Thus, disable the generation of interrupt requests by setting the interrupt mask flag (USBREMK) during bus idle or control transfer.

(3) When clearing the interrupt mask flag (USBREMK), do so immediately before the transition to bus suspend mode (immediately before the STOP instruction execution).

(4) When the USB reset signal or the Resume signal is received, the USB clock starts operating. As a consequence, USBTMIF and UWDERR are set after the elapse of a certain period (85.3 μs: fx = 6.0 MHz operation) of time from reception start. Therefore, when shifting to bus suspend mode (execution of stop instruction), set the interrupt mask flag (USBTMMK) and disable the generation of interrupt requests.

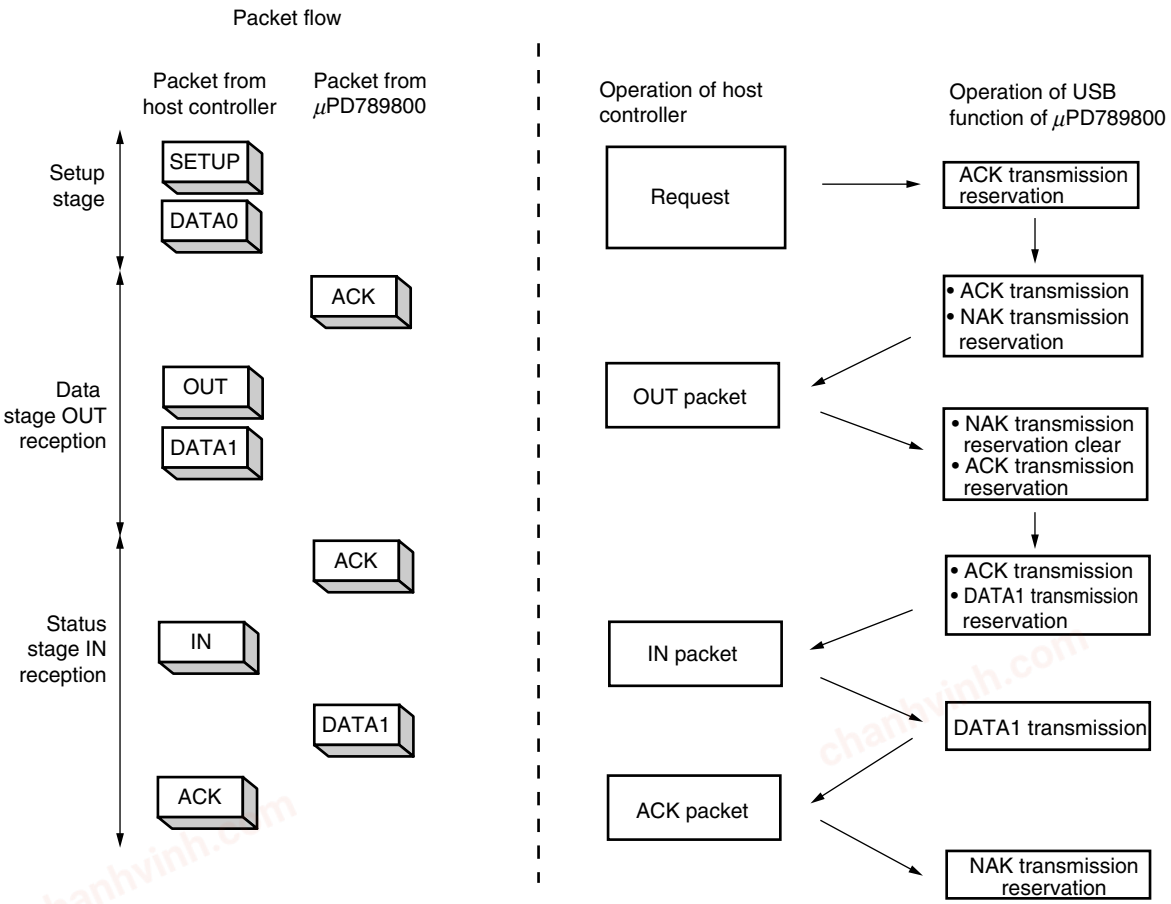
chanhvinh.com

★ 8.7 USB Function Control

8.7.1 Relationship between packets and operation modes

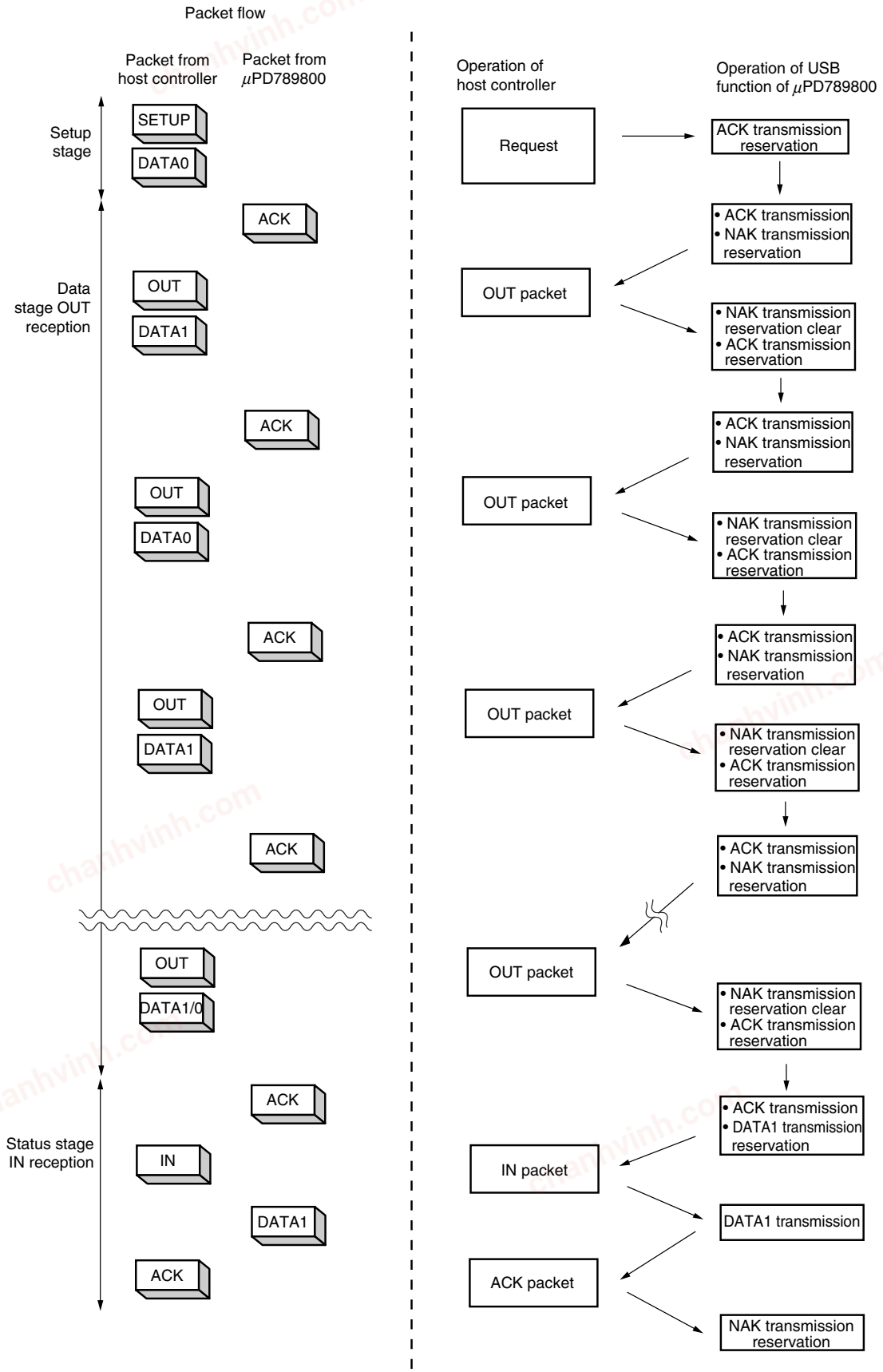
The relationship between packets and operation modes in the USB function is as follows.

(1) Control transfer (OUT) (Transfer byte count: 8 bytes or less)



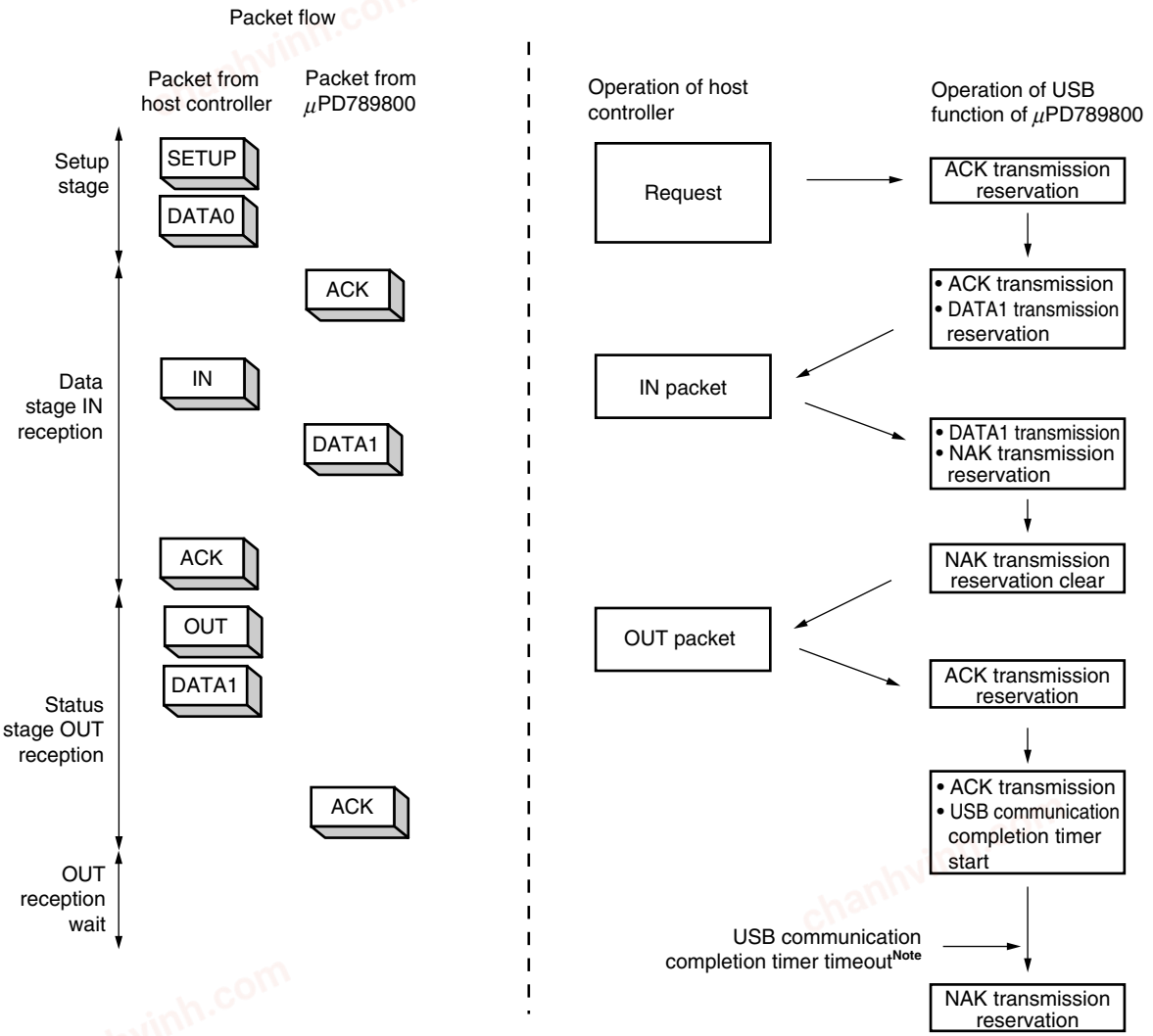
chanhvinh.com

(2) Control transfer (OUT) (Transfer byte count: 9 bytes or more)



chanhvinh.com

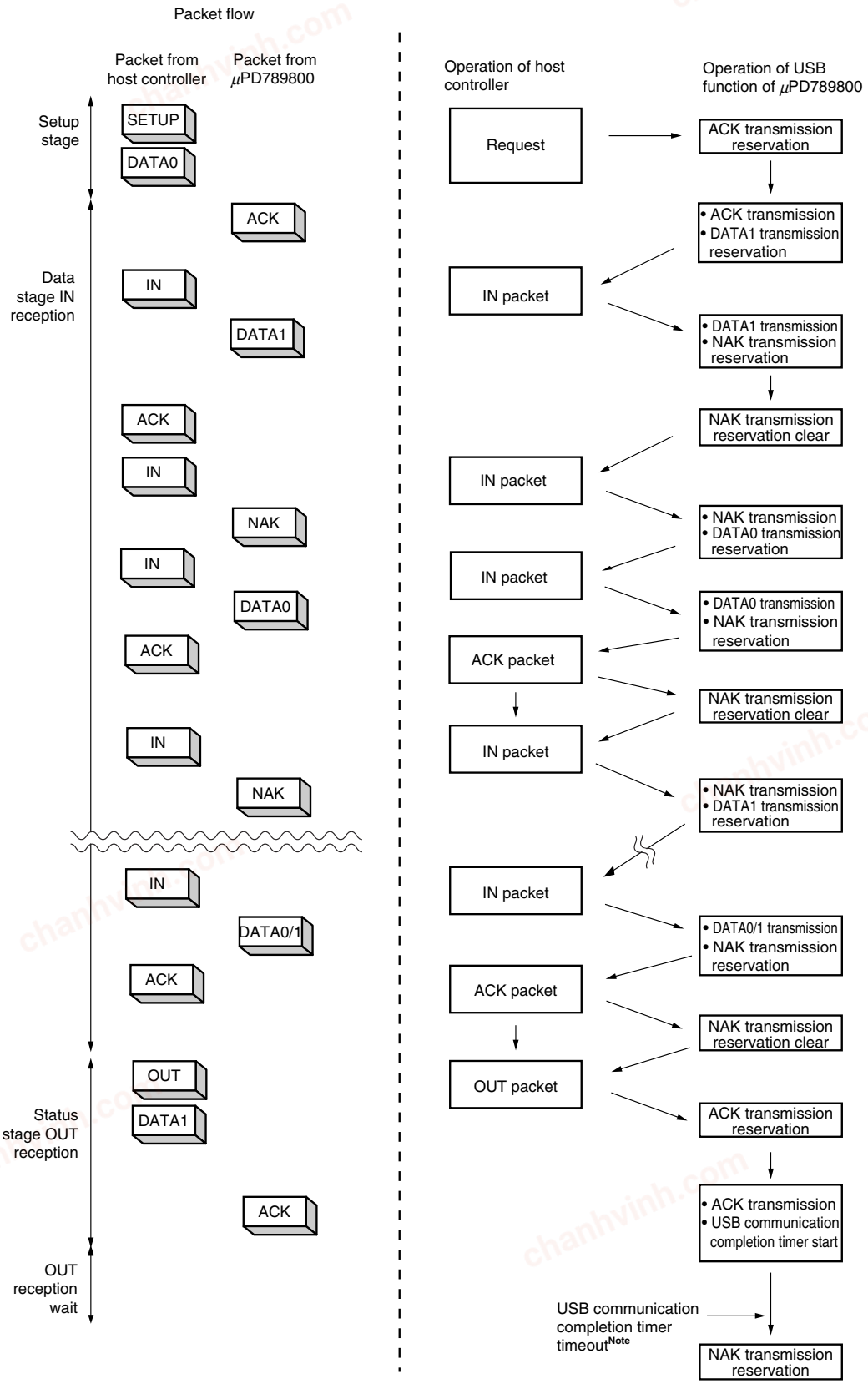
(3) Control transfer (IN) (Transfer byte count: 8 bytes or less)



Note If the ACK from the device cannot be received normally, the host transmits OUT again. Therefore, set the OUT receive wait state for a period so that the OUT can be received. Use a normal 8-bit timer for counting during this period.

chanhvinh.com

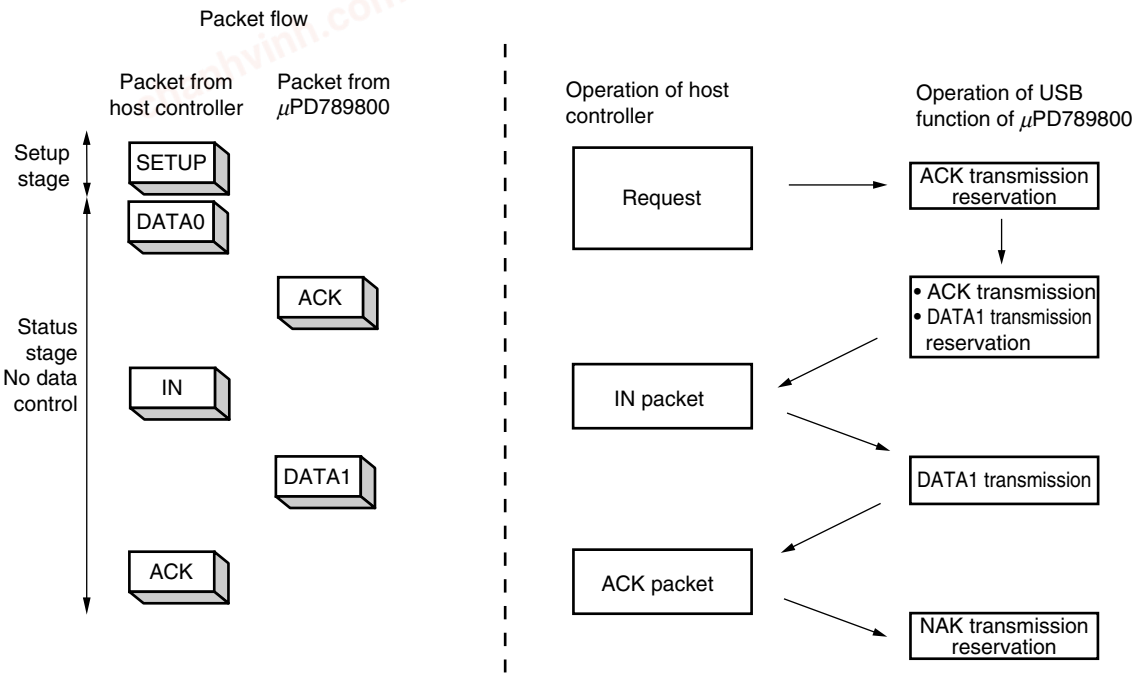
(4) Control transfer (IN) (Transfer byte count: 9 bytes or more)



Note If the ACK from the device cannot be received normally, the host transmits OUT again. Therefore, set the OUT receive wait state for a period so that the OUT can be received. Use a normal 8-bit timer for counting during this period.

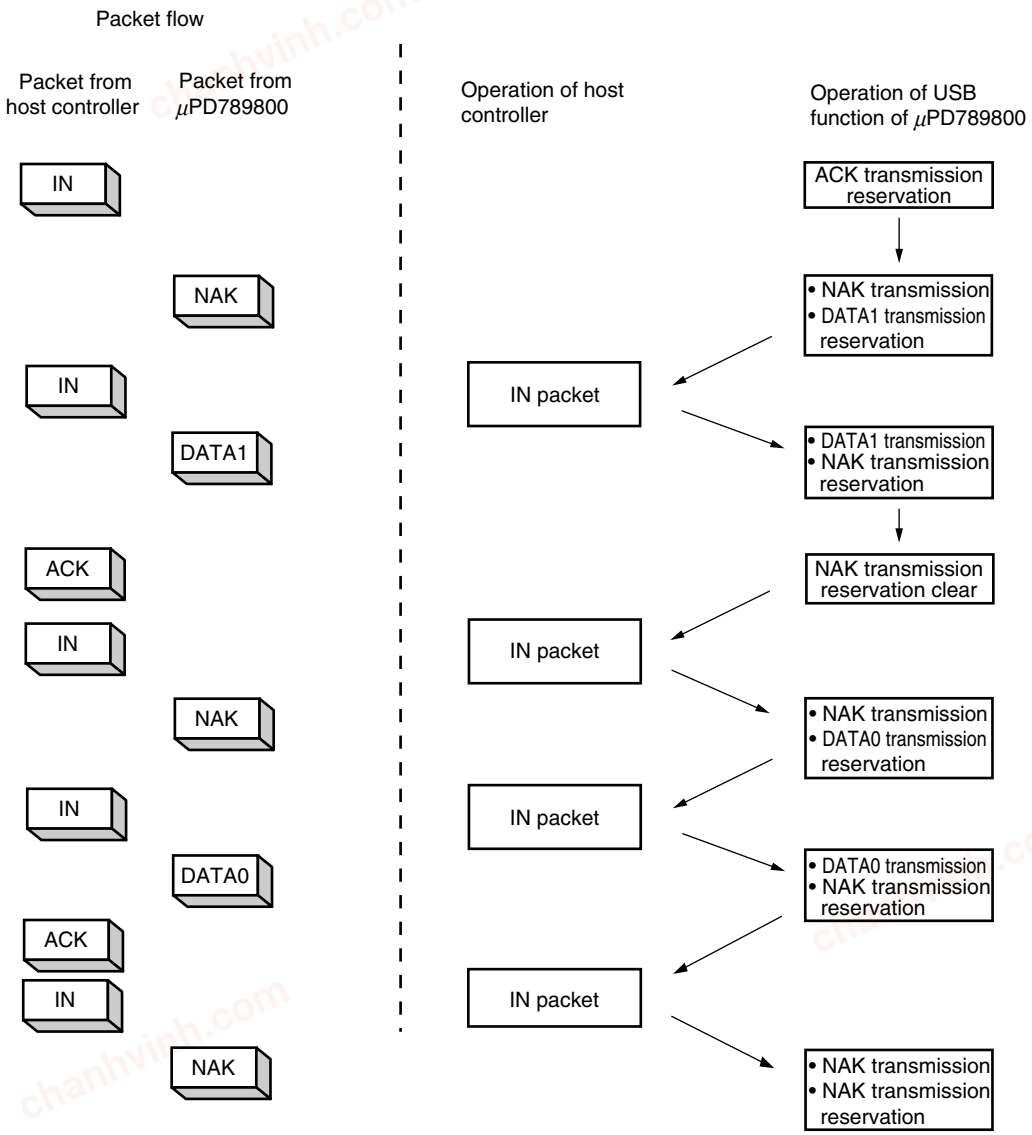
chanhvinh.com

(5) No data control



chanhvinh.com

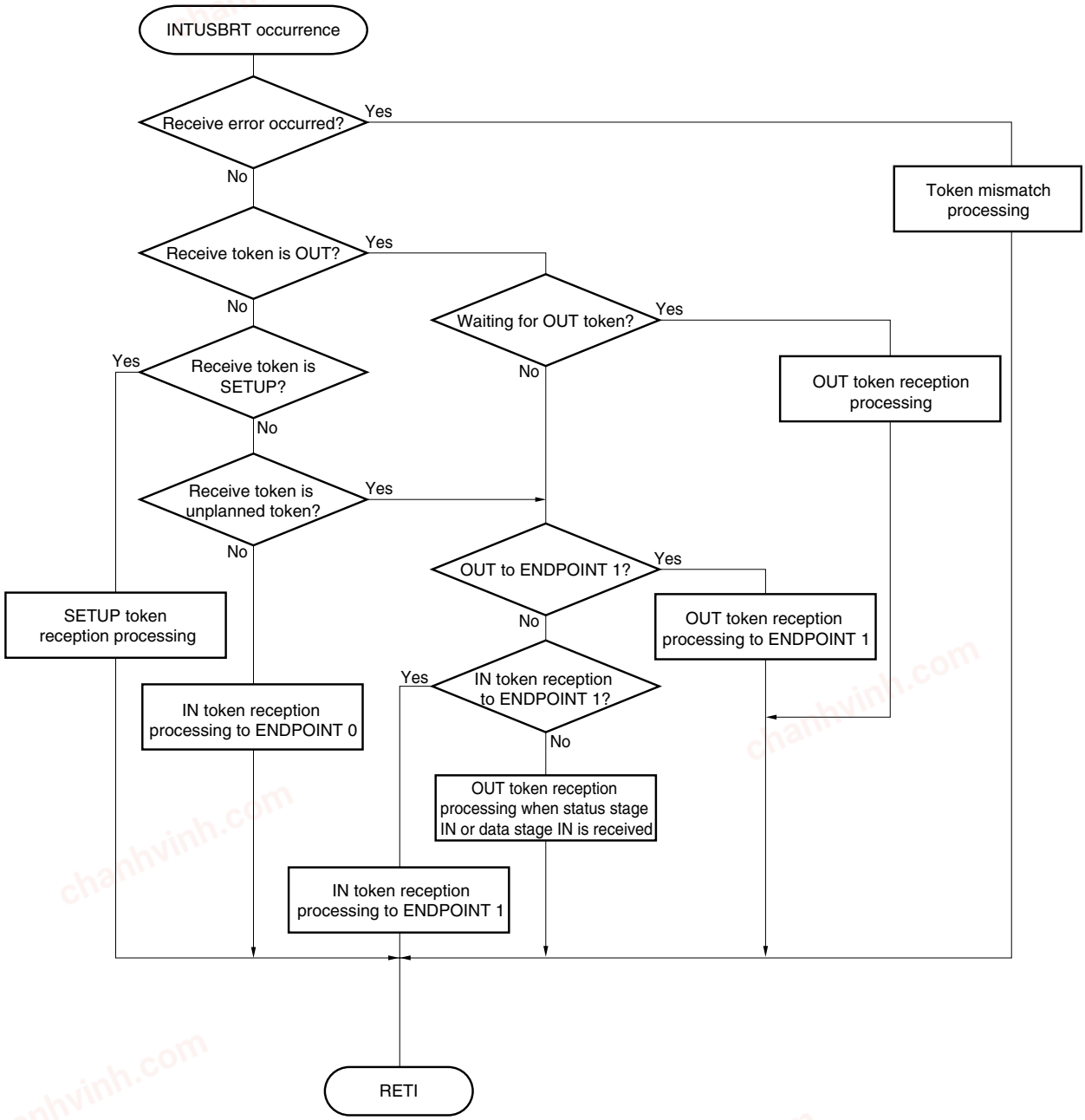
(6) Interrupt transfer



chanhvinh.com

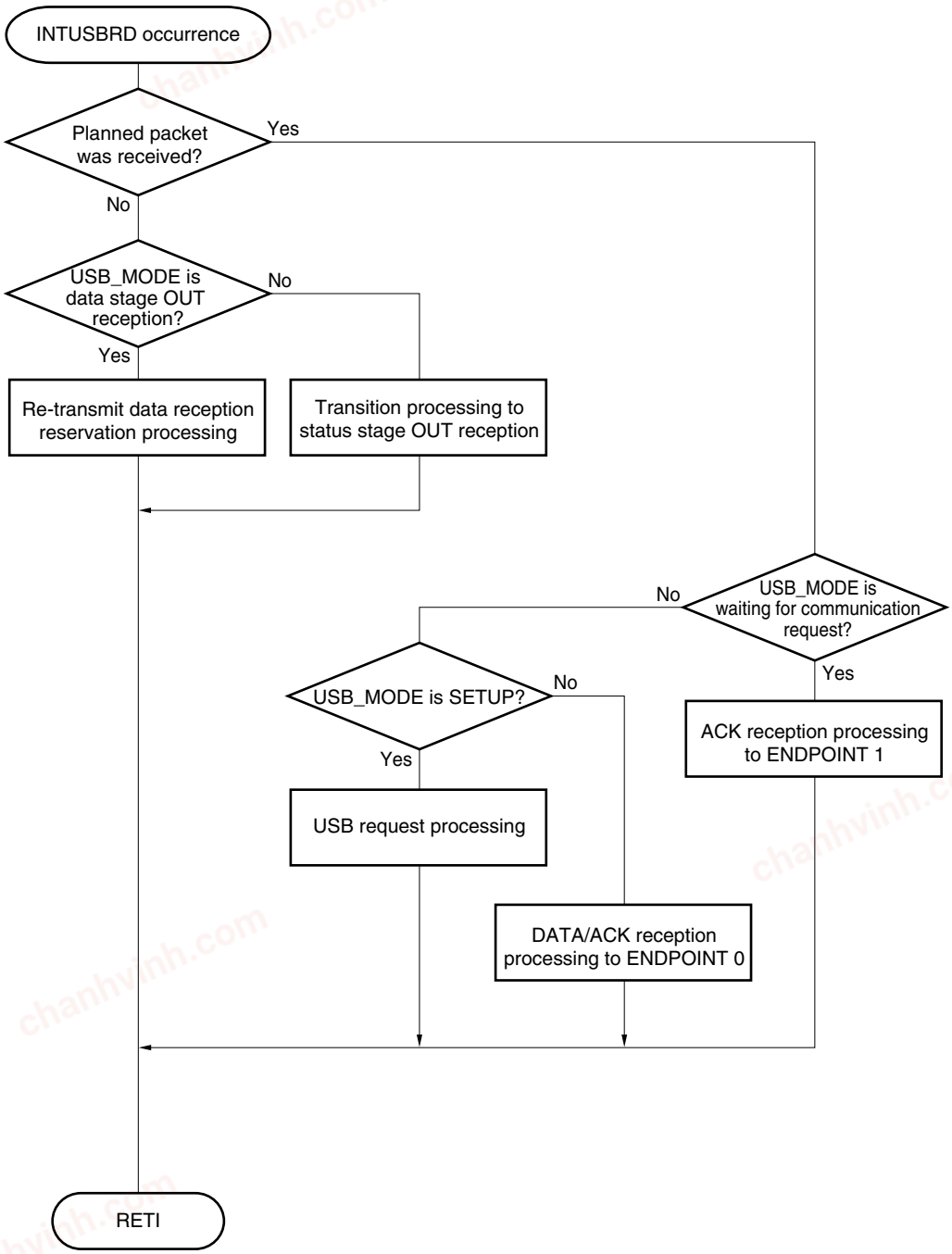
8.7.2 Interrupt servicing flow

(1) USB token packet reception interrupt servicing



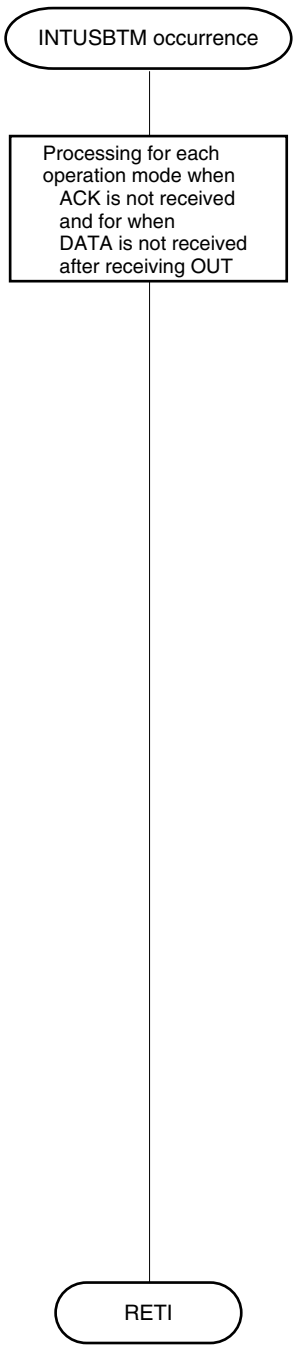
chanhvinh.com

(2) Data/handshake packet reception interrupt servicing



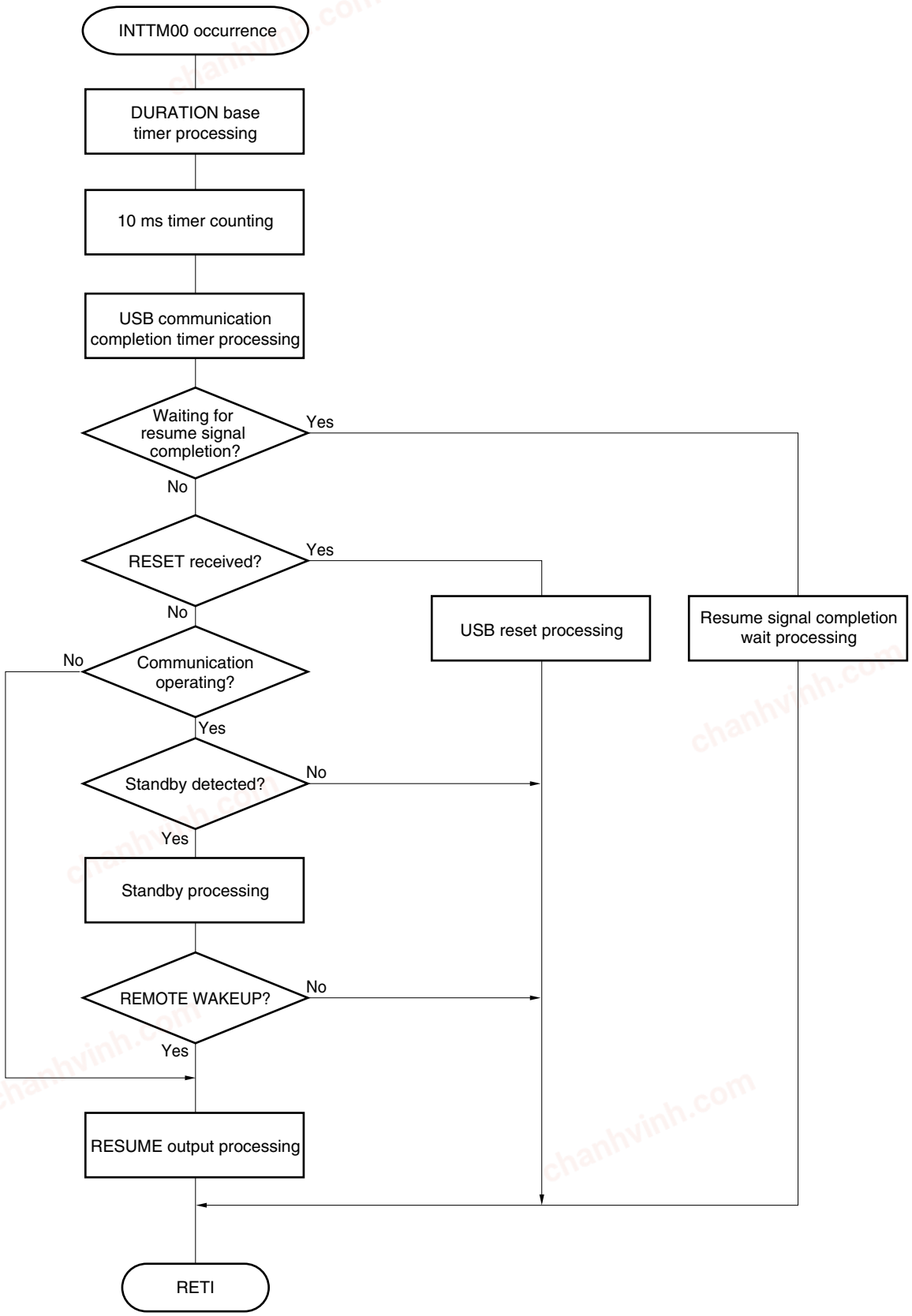
chanhvinh.com

(3) USB timer inadvertent program loop detection interrupt servicing



chanhvinh.com

(4) 1 ms timer interrupt servicing



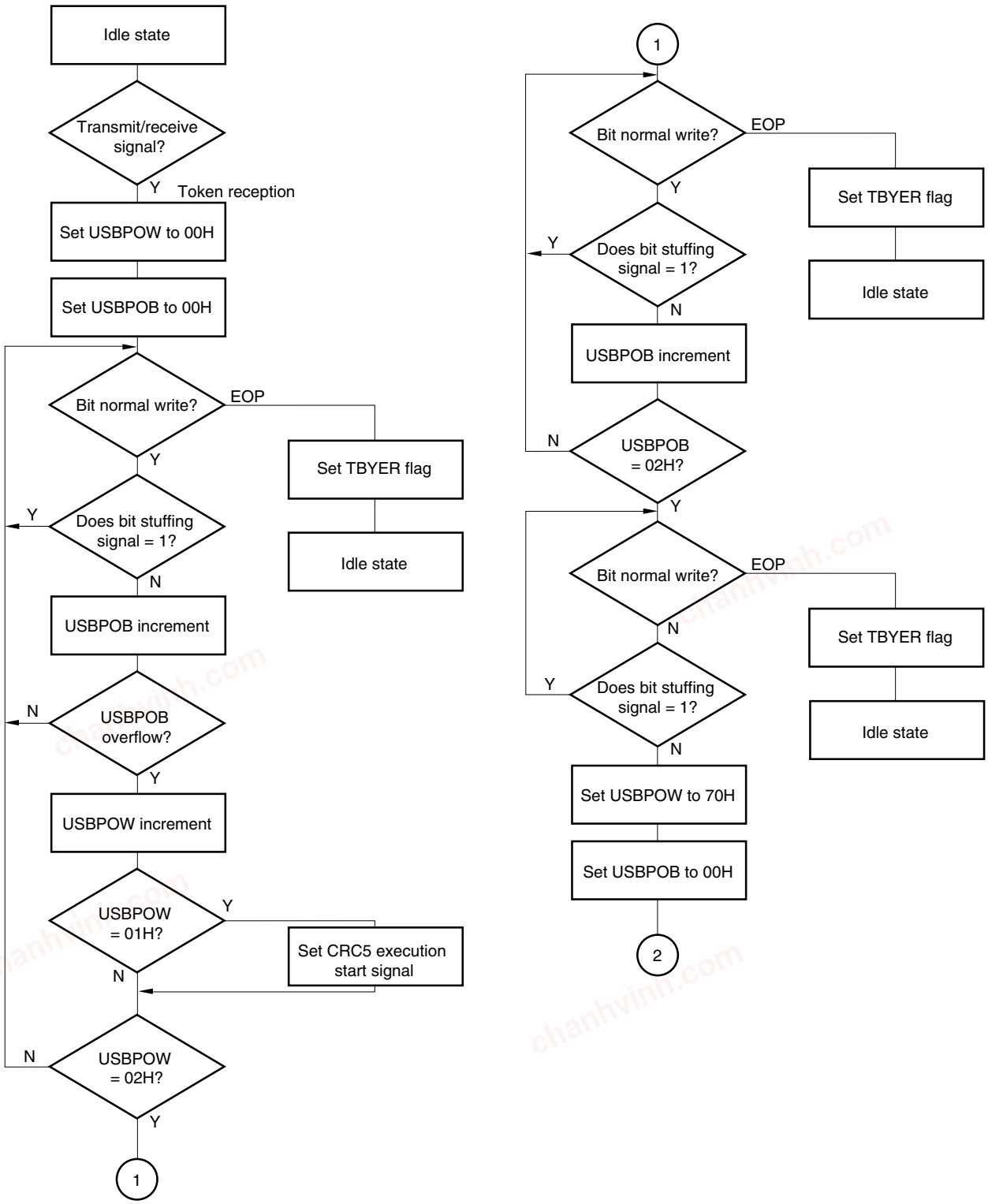
chanhvinh.com

8.8 USB Function Internal Circuit Operations

8.8.1 Operation of transmit/receive pointer

Figure 8-25. Flowchart of Transmit/Receive Pointer Operation (1/7)

(1) Token packet reception (1/2)

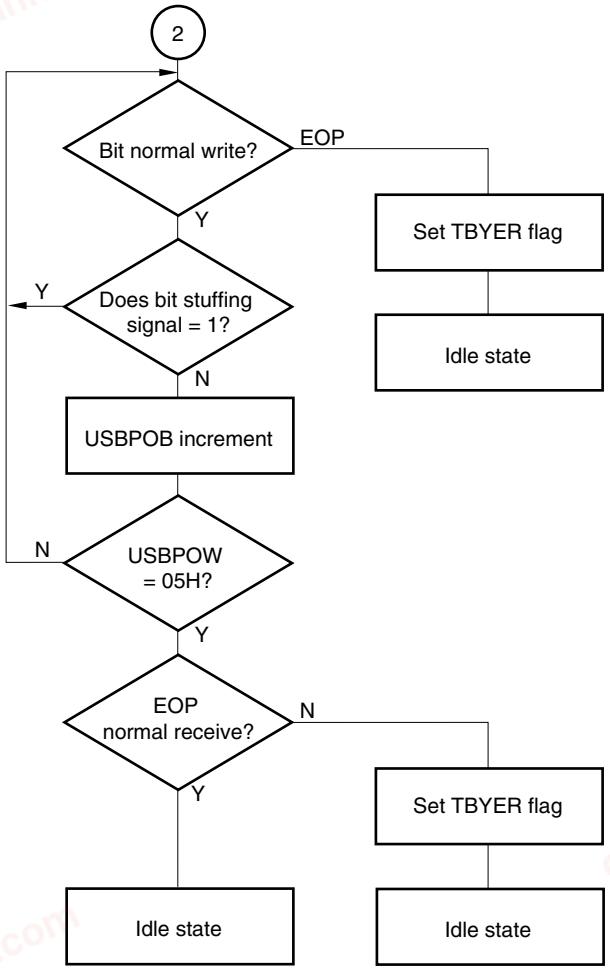


TBYER: Bit 5 of token packet receive result store register (TRXRSL)

chanhvinh.com

Figure 8-25. Flowchart of Transmit/Receive Pointer Operation (2/7)

(1) Token packet reception (2/2)

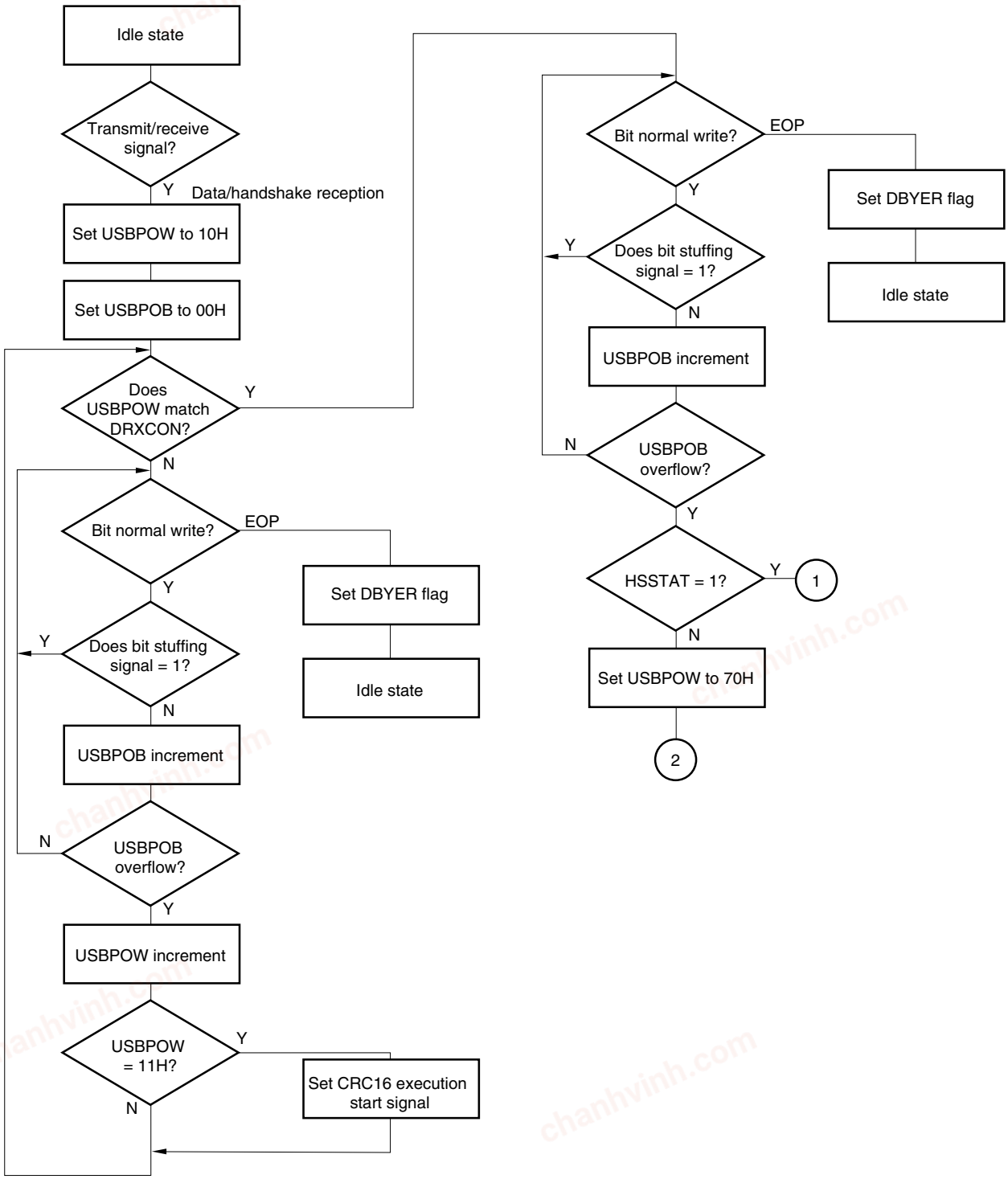


TBYER: Bit 5 of token packet receive result store register (TRXRSL)

chanhvinh.com

Figure 8-25. Flowchart of Transmit/Receive Pointer Operation (3/7)

(2) Data/Handshake packet reception (1/2)



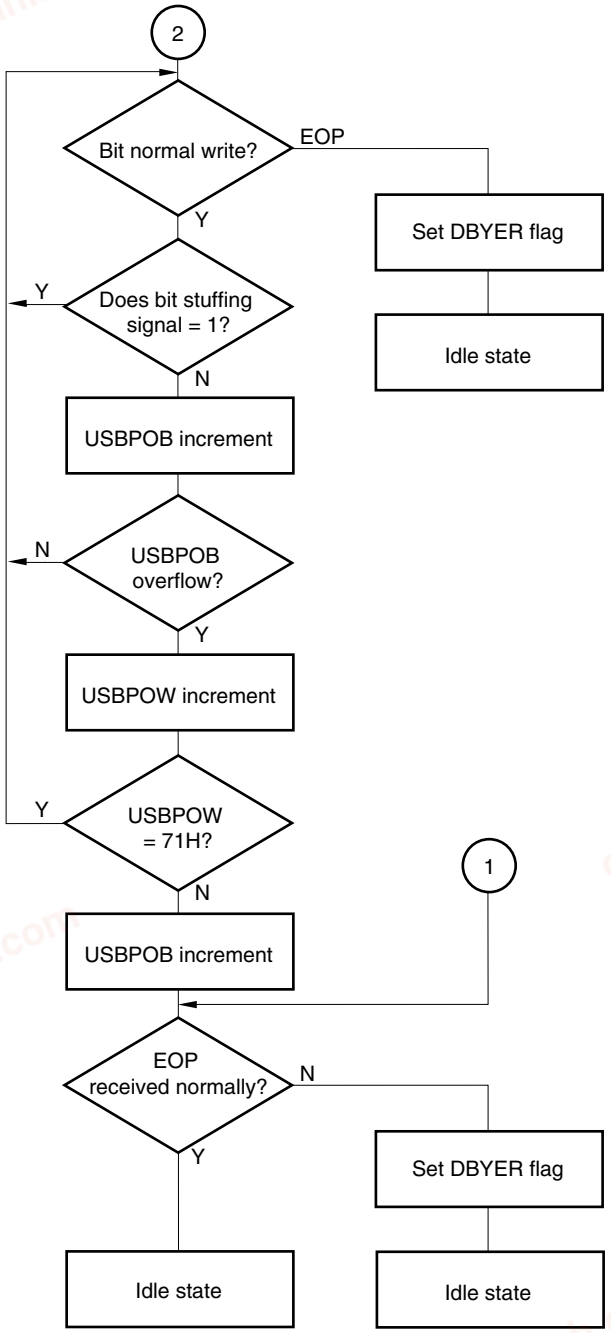
DBYER: Bit 5 of data/handshake packet receive result store register (DRXRSL)

DRXCON: Data/handshake packet receive byte number counter

chanhvinh.com

Figure 8-25. Flowchart of Transmit/Receive Pointer Operation (4/7)

(2) Data/Handshake packet reception (2/2)

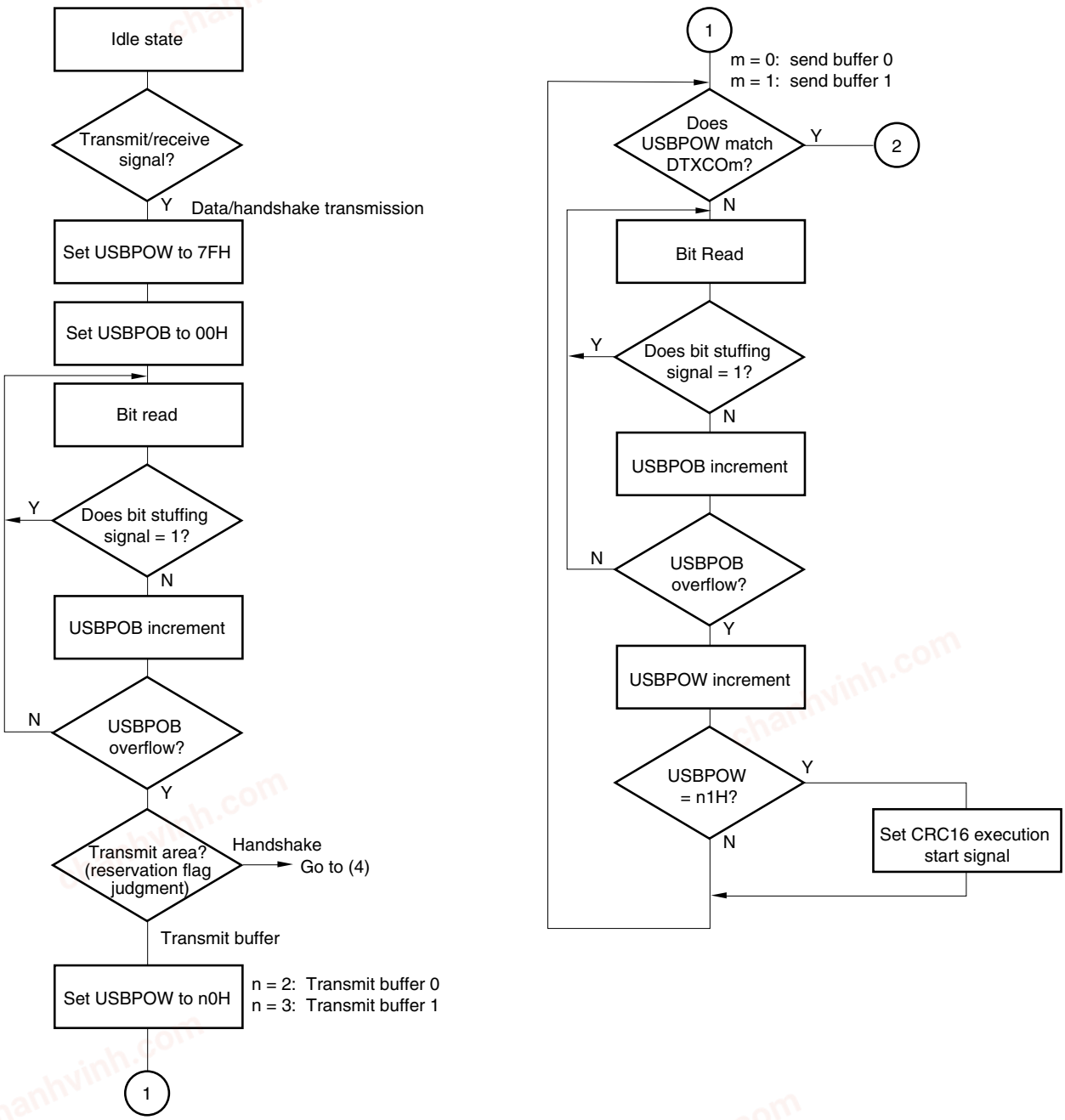


DBYER: Bit 5 of data/handshake packet receive result store register (DRXRSL)

chanhvinh.com

Figure 8-25. Flowchart of Transmit/Receive Pointer Operation (5/7)

(3) Data packet transmit (1/2)

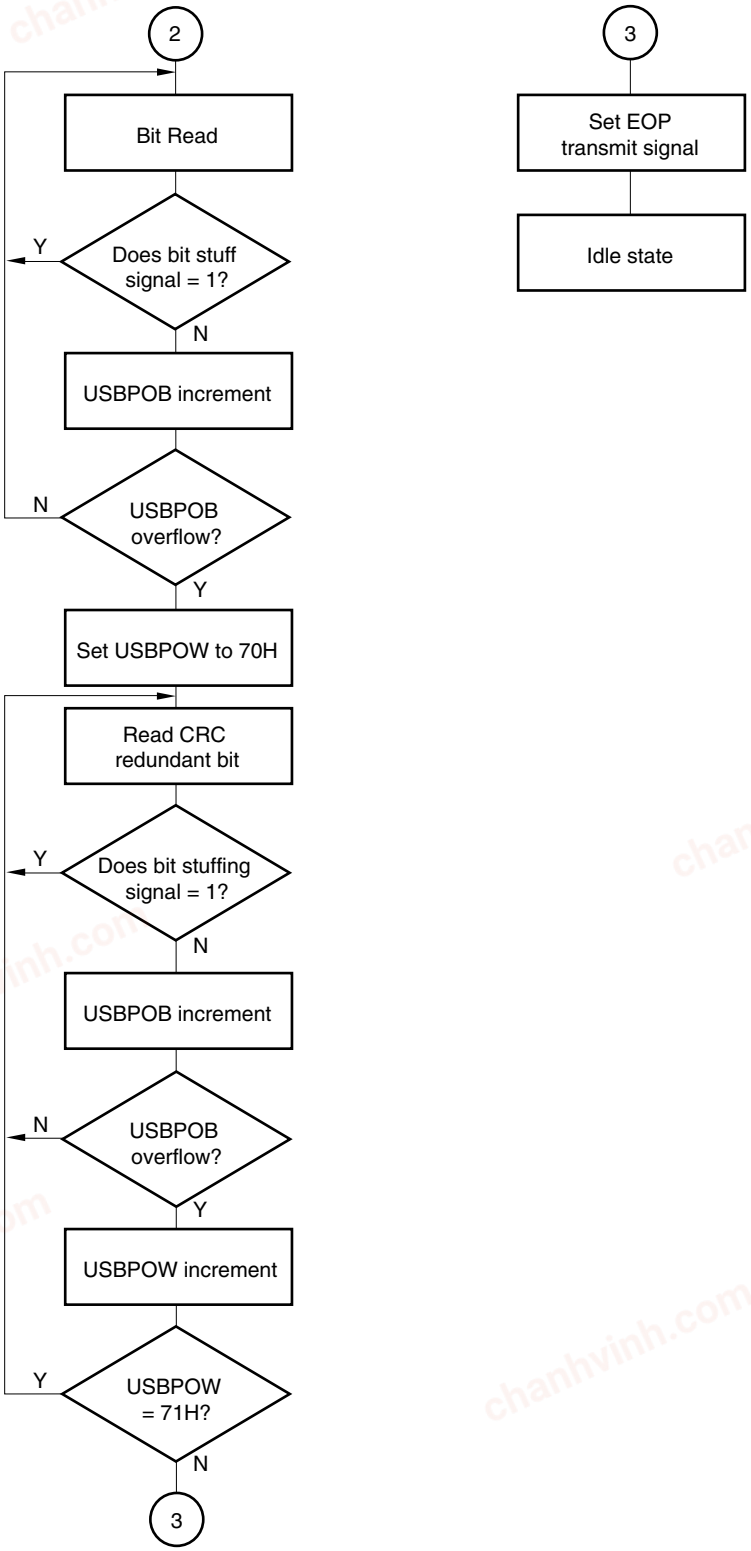


DTXCO0, DTXCO1: Data packet transmit byte number counter

chanhvinh.com

Figure 8-25. Flowchart of Transmit/Receive Pointer Operation (6/7)

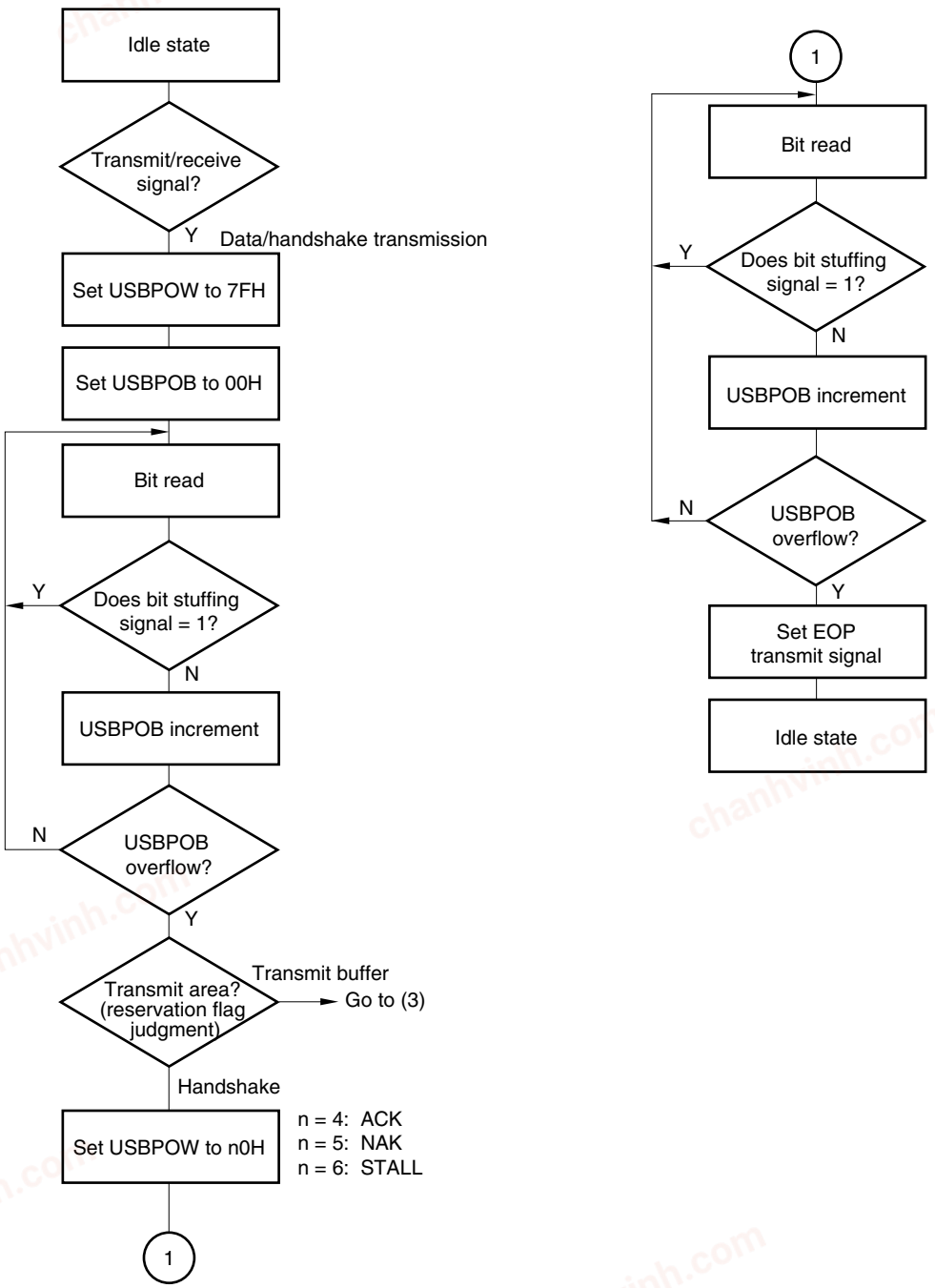
(3) Data packet transmit (2/2)



chanhvinh.com

Figure 8-25. Flowchart of Transmit/Receive Pointer Operation (7/7)

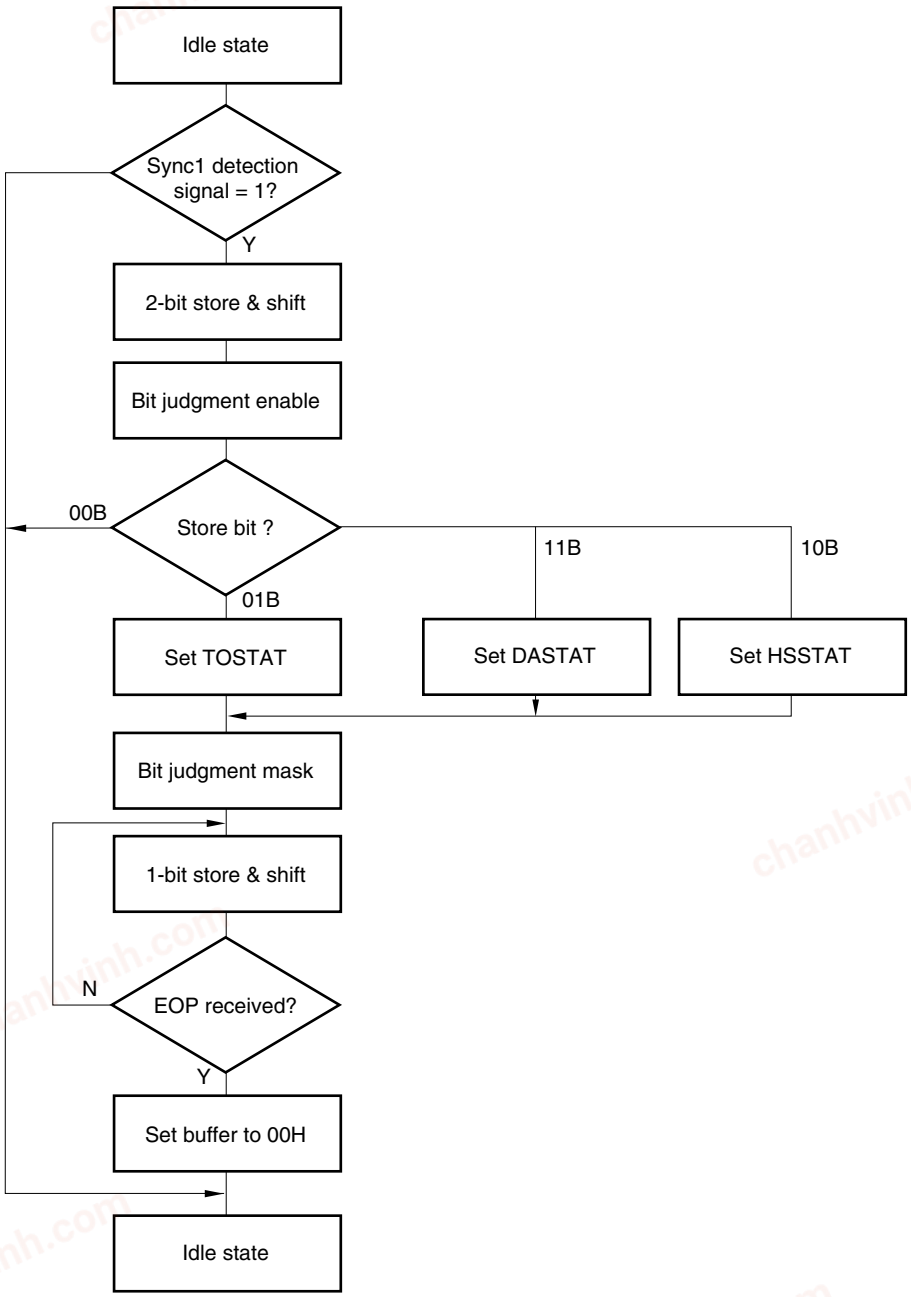
(4) Handshake packet transmission



chanhvinh.com

8.8.2 Receive bank switching ID detection buffer operation

Figure 8-26. Flowchart of Receive Bank Switching ID Detection Buffer Operation



TOSTAT: Bit 0 of packet receive status register (RXSTAT)
DASTAT: Bit 1 of packet receive status register (RXSTAT)
HSSTAT: Bit 2 of packet receive status register (RXSTAT)

chanhvinh.com

8.8.3 Sync detection/USBCLK detector operation

This circuit generates the USBCLK signal (1.5 MHz) upon detecting the sync part of the receive packet. In addition, it contains an NRZI decoder that decodes receive packets and detects the last bit of the sync part.

When the last sync bit is detected, a signal that specifies start of storing in the ID detection buffer is output.

Figure 8-27. Timing of Sync Detection/USBCLK Detector Operation

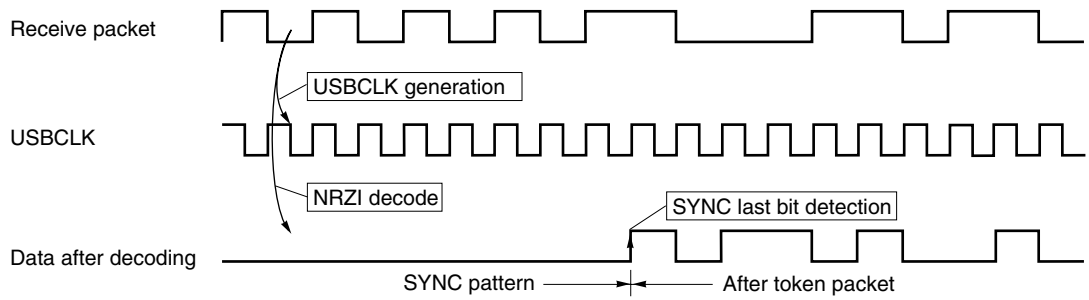
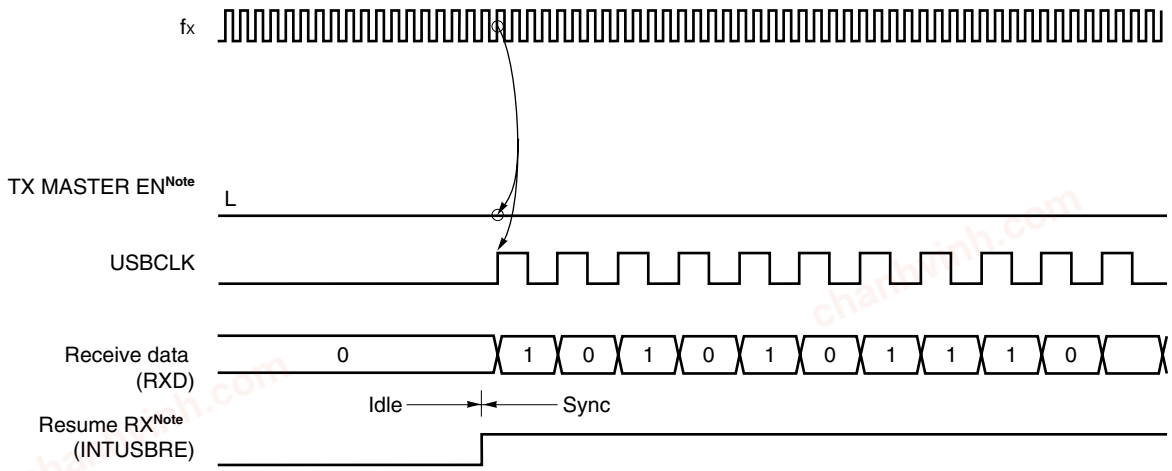


Figure 8-28. Timing of Sync Detection/USBCLK Generation Operation

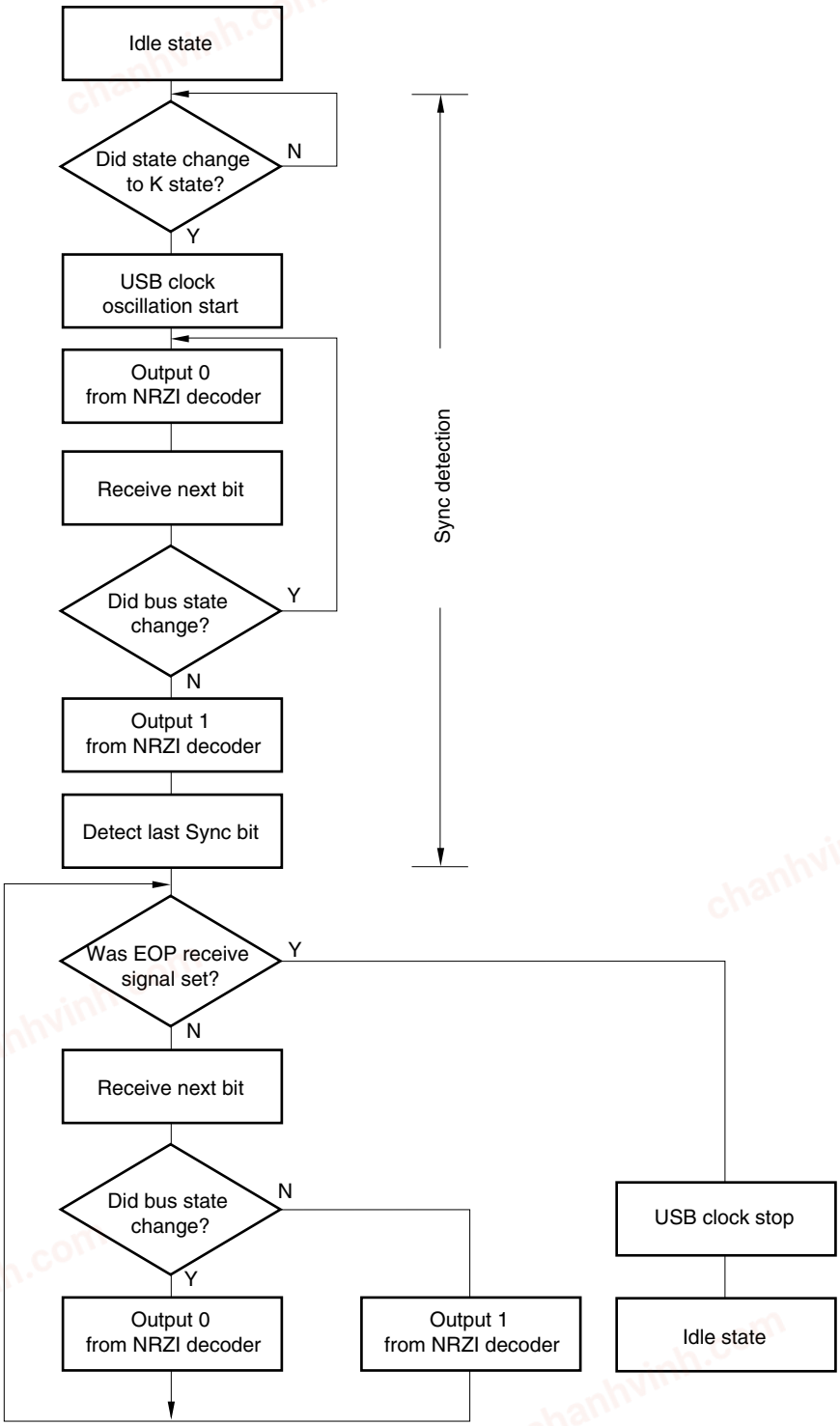


Note Because these signals are used internally, confirmation by software is not possible.

Remark The USB clock starts operating at the falling edge of fx after transition from the J state to the K state of the bus. However, this control is masked if TX MASTER EN = 1.

chanhvinh.com

Figure 8-29. Flowchart of Sync Detection/USBCLK Detector Operation



chanhvinh.com

8.8.4 NRZI encoder operation

This circuit performs NRZI encoding of data to be transmitted.

Figure 8-30. Timing of NRZI Encoder Operation

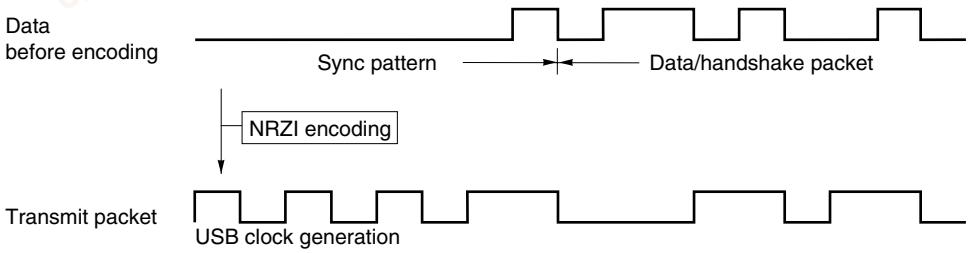
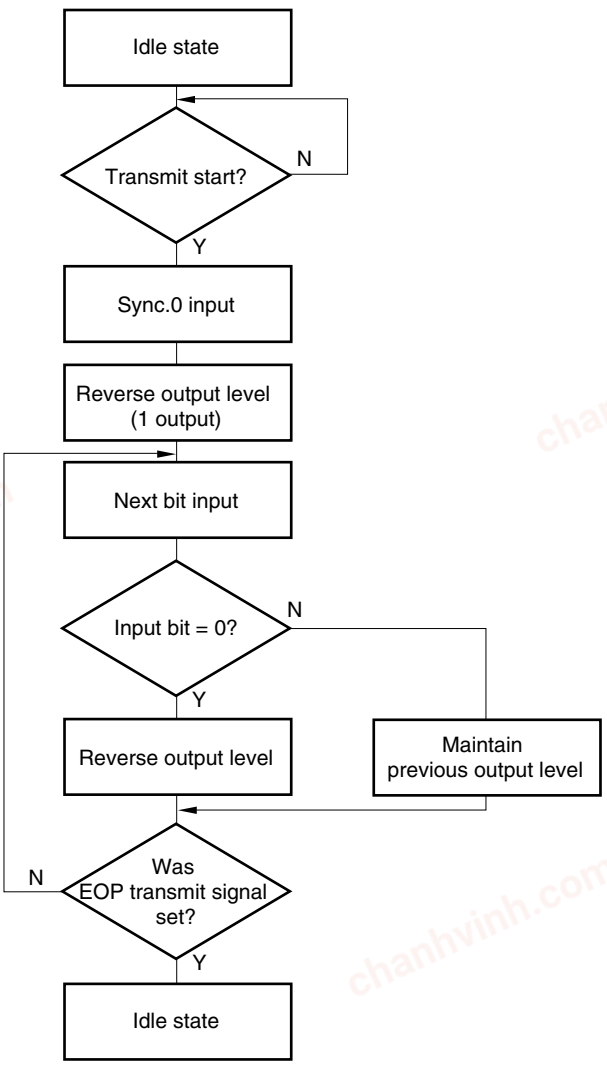


Figure 8-31. Flow Chart of NRZI Encoder Operation



chanhvinh.com

8.8.5 Bit stuffing/strip controller operation

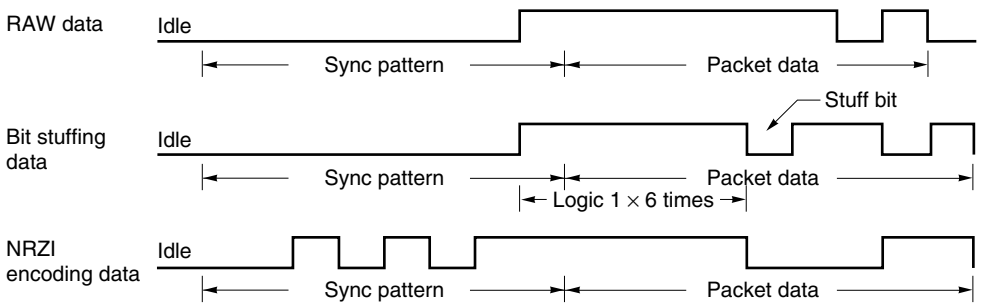
This circuit counts the number of “logic 1” of transmit/receive packets. If six successive logic 1s are detected, it outputs an increment disable signal to the transmit/receive pointer (USBPOB). During packet transmission, it inserts “logic 0” simultaneously with the increment disable signal.

Moreover, during bit stripping, if the bit that should be deleted was a “logic 1,” this is detected as a bit stuffing error.

Figure 8-32. Timing of Bit Stuffing/Strip Controller Operation

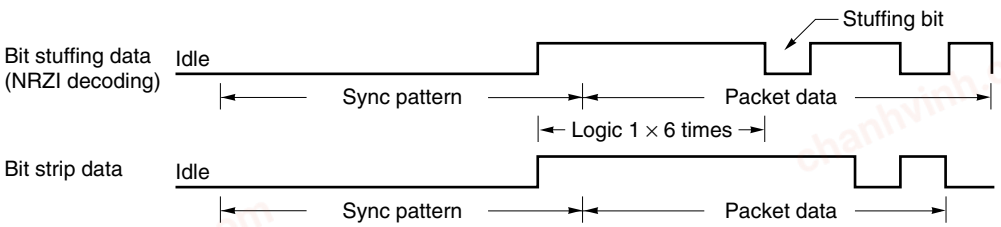
(1) Bit stuffing

If “1” occurs six successive times, a “0” is inserted forcibly to shift the level.



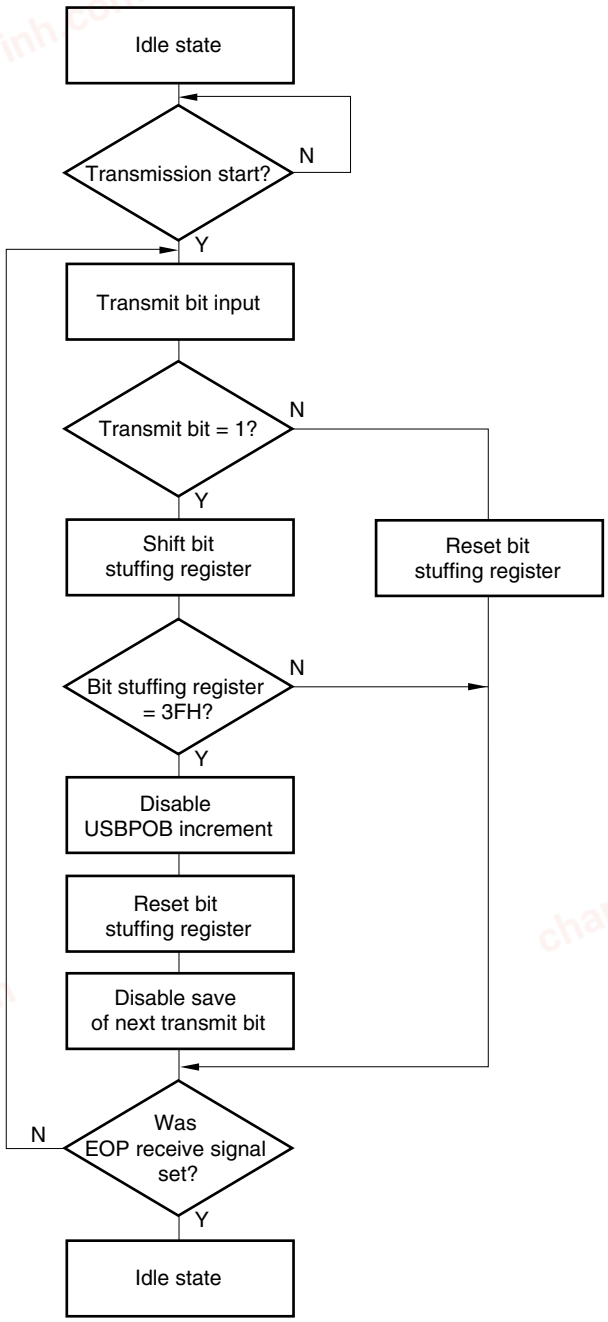
(2) Bit striping

If “1” occurs six successive times, the next bit is deleted as a stuffing bit.



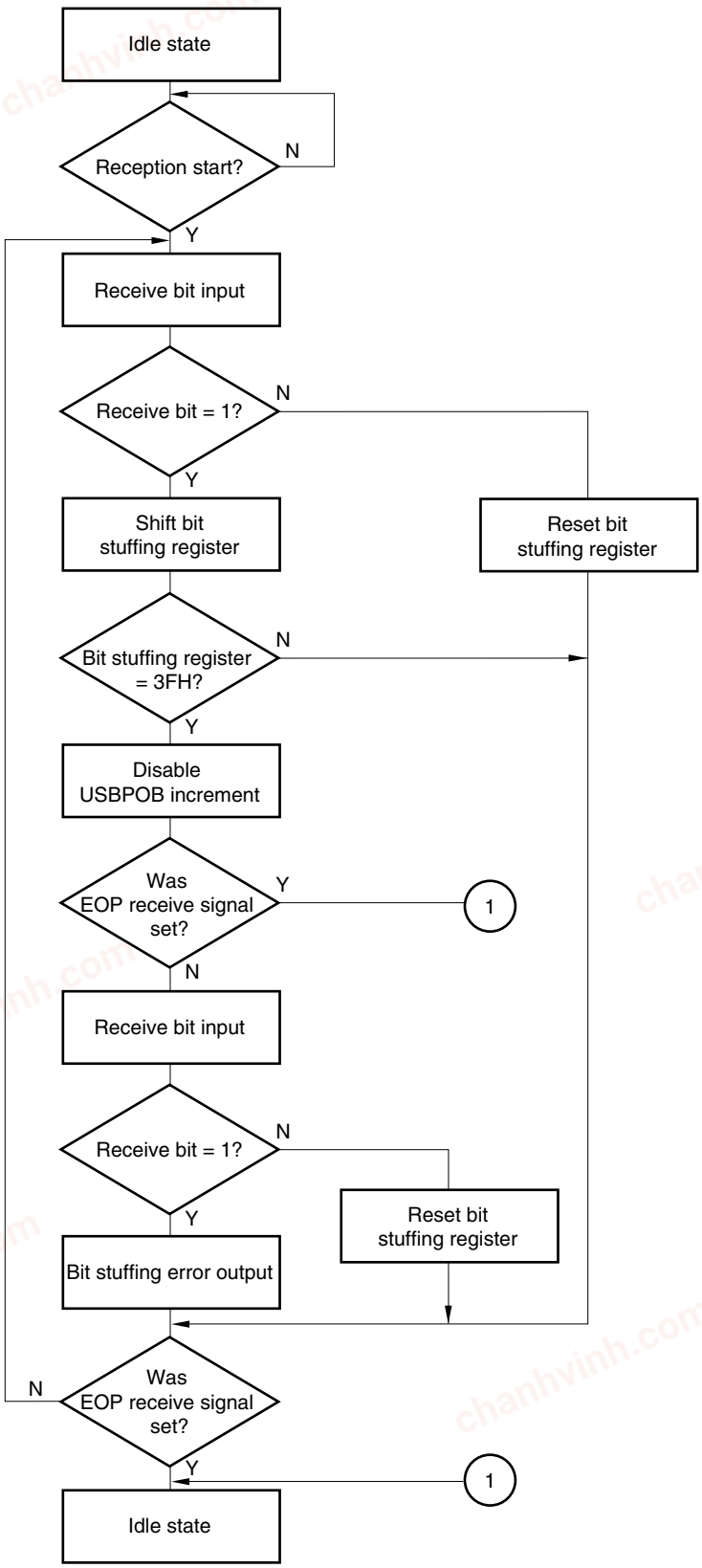
chanhvinh.com

Figure 8-33. Flow Chart of Bit Stuffing Control Operation



chanhvinh.com

Figure 8-34. Flow Chart of Bit Strip Control Operation



chanhvinh.com

CHAPTER 9 SERIAL INTERFACE 10

9.1 Functions of Serial Interface 10

Serial interface 10 has the following two modes.

- Operation stop mode
- 3-wire serial I/O mode

(1) Operation stop mode

This mode is used when serial transfer is not carried out. It enables a reduction in power consumption.

(2) 3-wire serial I/O mode (MSB/LSB start bit switchable)

In this mode, 8-bit data transfer is carried out with three lines: one for the serial clock ($\overline{\text{SCK10}}$) and two for serial data (SI10 and SO10).

The 3-wire serial I/O mode supports simultaneous transmit and receive operations, reducing data transfer processing time.

It is possible to switch the start bit of 8-bit data to be transmitted between the MSB and the LSB, thus allowing connection to devices with either start bit.

The 3-wire serial I/O mode is effective for connecting display controllers and peripheral I/Os such as the 75XL Series, 78K Series, and 17K Series that have an internal conventional clocked serial interface.

chanhvinh.com

9.2 Configuration of Serial Interface 10

Serial interface 10 consists of the following hardware.

Table 9-1. Configuration of Serial Interface 10

Item	Configuration
Register	Transmit/receive shift register 10 (SIO10)
Control register	Serial operating mode register 10 (CSIM10)

(1) Transmit/receive shift register 10 (SIO10)

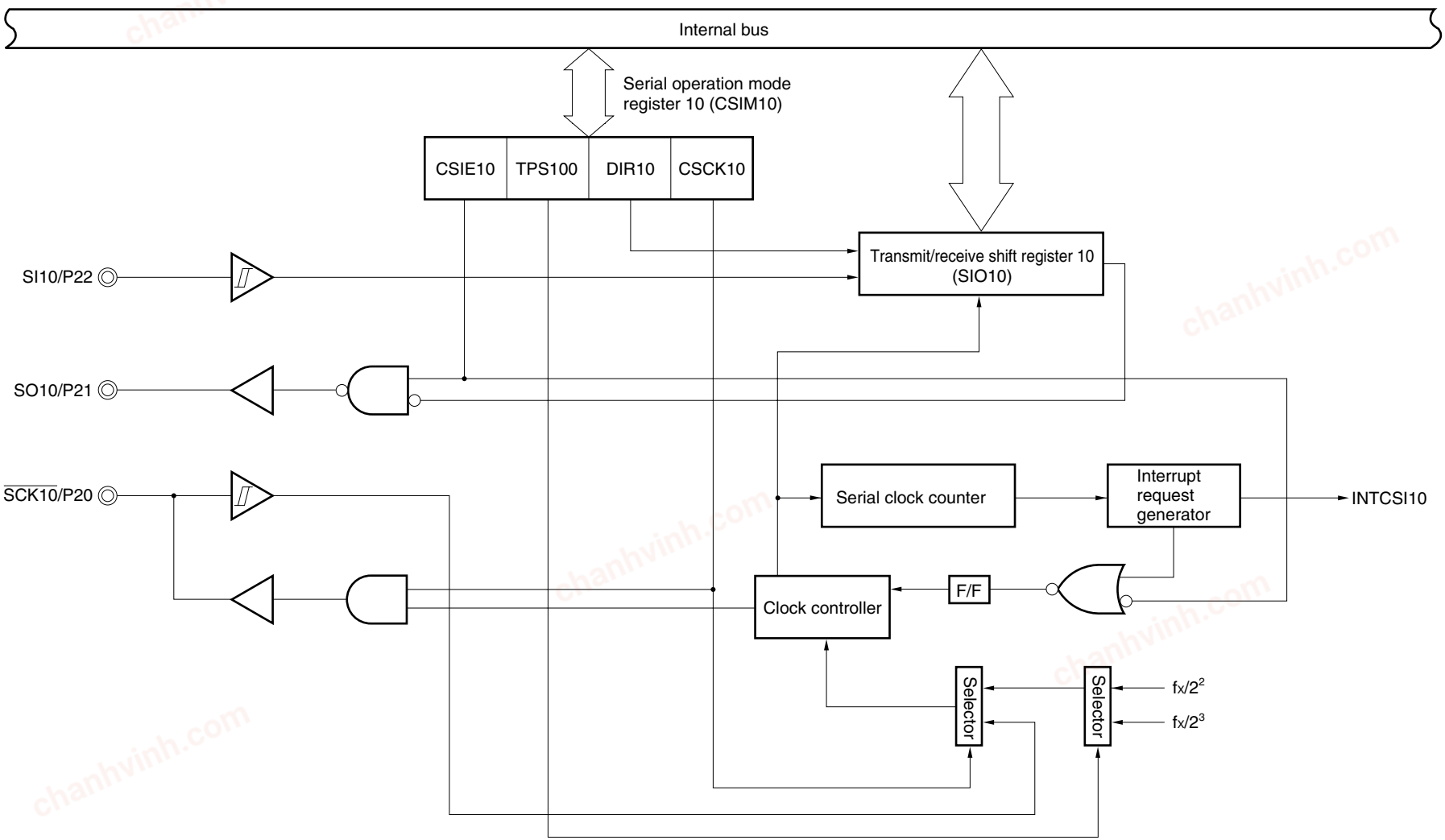
This is an 8-bit register used for parallel-to-serial conversion and to perform serial data transmission/reception in synchronization with the serial clock.

SIO10 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes SIO10 undefined.

chanhvinh.com

Figure 9-1. Block Diagram of Serial Interface 10



chanhvinh.com

9.3 Register Controlling Serial Interface 10

The following register is used to control serial interface 10.

- Serial operation mode register 10 (CSIM10)

(1) Serial operation mode register 10 (CSIM10)

This register is used to control serial interface 10 and set the serial clock and start bit.

CSIM10 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM10 to 00H.

Figure 9-2. Format of Serial Operation Mode Register 10

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM10	CSIE10	0	0	TPS100	0	DIR10	CSCK10	0	FF72H	00H	R/W

CSIE10	Operation control in 3-wire serial I/O mode
0	Operation disabled
1	Operation enabled

TPS100	Count clock selection when operation enabled in 3-wire serial I/O mode
0	$f_X/2^2$
1	$f_X/2^3$

DIR10	Start bit specification
0	MSB
1	LSB

CSCK10	Clock selection in 3-wire serial I/O mode
0	Clock input to $\overline{\text{SCK10}}$ pin from external
1	Internal clock selected by TPS100

Caution Bits 0, 3, 5, and 6 must be set to 0.

chanhvinh.com

Table 9-2. Operating Mode Settings of Serial Interface 10

(1) Operation stop mode

CSIM10			PM22	P22	PM21	P21	PM20	P20	Start Bit	Shift Clock	P22/SI10 Pin Function	P21/SO10 Pin Function	P20/$\overline{\text{SCK10}}$ Pin Function
CSIE10	DIR10	CSCK10											
0	×	×	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	—	—	P22	P21	P20
Other than above									Setting prohibited				

(2) 3-wired serial I/O mode

CSIM10			PM22	P22	PM21	P21	PM20	P20	Start Bit	Shift Clock	P22/SI10 Pin Function	P21/SO10 Pin Function	P20/$\overline{\text{SCK10}}$ Pin Function
CSIE10	DIR10	CSCK10											
1	0	0	1[Note 2]	×[Note 2]	0	1	1	×	MSB	External clock	SI10[Note 2]	SO10 (CMOS output)	$\overline{\text{SCK10}}$ input
		1					0	1		Internal clock			$\overline{\text{SCK10}}$ output
1	1	0					1	×	LSB	External clock			$\overline{\text{SCK10}}$ input
		1					0	1		Internal clock			$\overline{\text{SCK10}}$ output
Other than above									Setting prohibited				

Notes 1. Can be used as port function.

2. If used only for transmission, can be used as P22 (CMOS input/output).

Remark ×: don't care

chanhvinh.com

9.4 Operation of Serial Interface 10

Serial interface 10 provides the following two modes.

- Operation stop mode
- 3-wire serial I/O mode

9.4.1 Operation stop mode

In the operation stop mode, serial transfer is not executed; therefore, the power consumption can be reduced.

The P20/$\overline{SCK10}$, P21/SO10, and P22/SI10 pins can be used as normal I/O ports.

(1) Register setting

Operation stop mode is set by serial operation mode register 10 (CSIM10).

Serial operation mode register 10 (CSIM10)

CSIM10 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{RESET}$ input sets CSIM10 to 00H.

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM10	CSIE10	0	0	TPS100	0	DIR10	CSCK10	0	FF72H	00H	R/W

CSIE10	Operation control in 3-wire serial I/O mode
0	Operation disabled
1	Operation enabled

Caution Bits 0, 3, 5, and 6 must be set to 0.

chanhvinh.com

9.4.2 3-wire serial I/O mode

The 3-wire serial I/O mode is useful for connection of peripheral I/Os and display controllers, etc., which incorporate a conventional synchronous serial interface, such as the 75XL Series, 78K Series, 17K Series, etc.

Communication is performed using three lines: the serial clock ($\overline{SCK10}$), serial output (SO10), and serial input (SI10).

(1) Register setting

3-wire serial I/O mode settings are performed using serial operating mode register 10 (CSIM10).

(a) Serial operation mode register 10 (CSIM10)

CSIM10 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{RESET}$ input sets CSIM10 to 00H.

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM10	CSIE10	0	0	TPS100	0	DIR10	CSCK10	0	FF72H	00H	R/W

CSIE10	Operation control in 3-wire serial I/O mode
0	Operation disabled
1	Operation enabled

TPS100	Count clock selection when operation enabled in 3-wire serial I/O mode
0	$f_X/2^2$
1	$f_X/2^3$

DIR10	Start bit specification
0	MSB
1	LSB

CSCK10	Clock selection in 3-wire serial I/O mode
0	Clock input to $\overline{SCK10}$ pin from external
1	Internal clock selected by TPS100

Caution Bits 0, 3, 5, and 6 must be set to 0.

chanhvinh.com

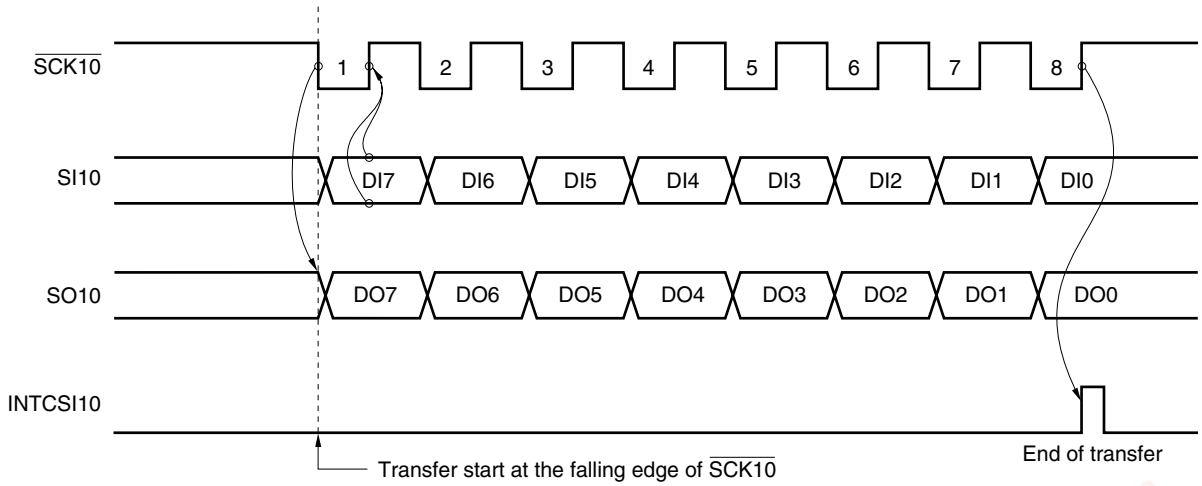
(2) Communication operation

In the 3-wire serial I/O mode, data transmission/reception is performed in 8-bit units. Data is transmitted/received bit by bit in synchronization with the serial clock.

Transmit/receive shift register 10 (SIO10) shift operations are performed in synchronization with the fall of the serial clock ($\overline{SCK10}$). Then transmit data is held in the SO10 latch and output from the SO10 pin. Also, receive data input to the SI10 pin is latched in the input bits of SIO10 on the rise of $\overline{SCK10}$.

At the end of an 8-bit transfer, the operation of SIO10 stops automatically, and the interrupt request signal (INTCSI10) is generated.

Figure 9-3. 3-Wire Serial I/O Mode Timing



Cautions

1. **When data is written to SIO10 in the serial operation disabled status (CSIE10 = 0), the data cannot be transmitted or received.**
2. **When data is written to SIO10 in the serial operation disabled status (CSIE10 = 0) and then serial operation is enabled (CSIE10 = 1), the data cannot be transmitted or received.**
3. **Once data has been written to SIO10 with the serial clock selected (CSCK10 = 0), overwriting the data does not update the contents of SIO10.**
4. **When CSIM10 is operated during data transmission/reception, data cannot be transmitted or received normally.**
5. **When SIO10 is operated during data transmission/reception, the data cannot be transmitted or received normally.**

(3) Transfer start

Serial transfer is started by setting transfer data to transmit/receive shift register 10 (SIO10) when the following two conditions are satisfied.

- Bit 7 (CSIE10) of serial operation mode register 10 (CSIM10) = 1
- Internal serial clock is stopped or $\overline{SCK10}$ is high level after 8-bit serial transfer.

Termination of 8-bit transfer stops the serial transfer automatically and generates the interrupt request signal (INTCSI10).

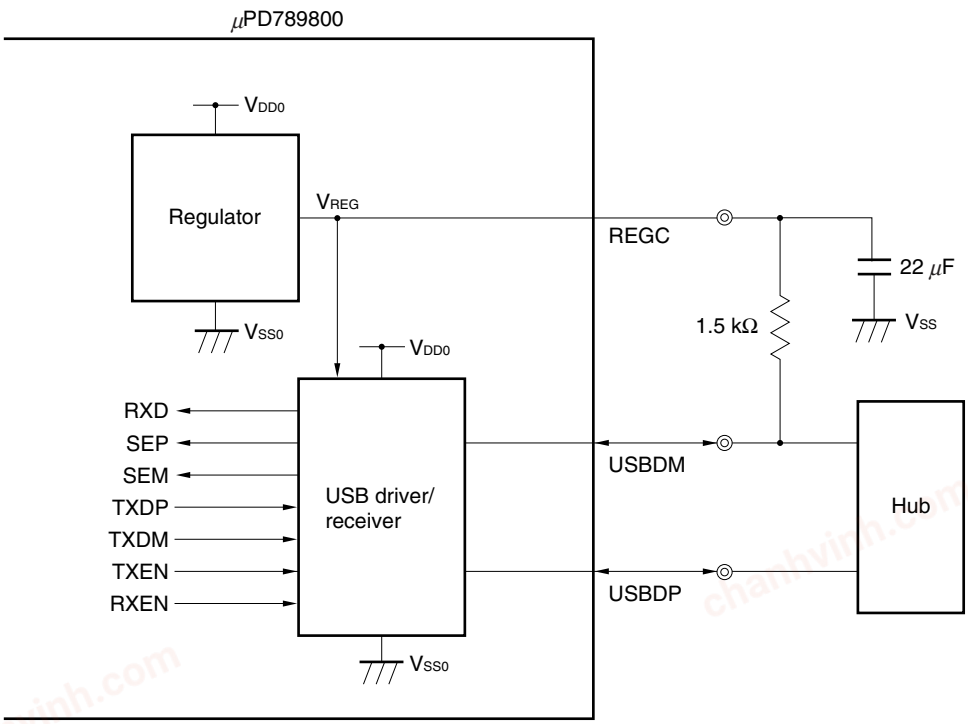
chanhvinh.com

CHAPTER 10 REGULATOR

The μPD789800 incorporates a regulator that powers the USB driver/receiver. The features are as follows.

- Generates V_{REG} (3.3 ±0.3 V) from V_{DD0}, V_{DD1} (4.0 to 5.5 V) and outputs it to the REGC pin.
- Supports power-saving mode, reducing power consumption in mode.

★ **Figure 10-1. Block Diagram of Regulator and USB Driver/Receiver**



Cautions 1. To stabilize the V_{REG} voltage, connect the REGC pin to V_{SS} via 22 μF capacitor.

2. Connect the pull-up resistor (1.5 kΩ) for the USBDM pin to the REGC pin.

chanhvinh.com

CHAPTER 11 INTERRUPT FUNCTIONS

11.1 Interrupt Function Types

The following two types of interrupt functions are used.

(1) Non-maskable interrupt

This interrupt is acknowledged unconditionally. It does no undergo interrupt priority control and is given top priority over all other interrupt requests.
A standby release signal is generated.
One interrupt source from the watchdog timer is incorporated as a non-maskable interrupt.

(2) Maskable interrupt

These interrupts undergo mask control. If two or more interrupts with the same priority are simultaneously generated, each interrupt has a predetermined priority as shown in Table 11-1.
A standby release signal is generated.
Two external and nine internal interrupt sources are incorporated as maskable interrupts.

11.2 Interrupt Sources and Configuration

A total of 12 non-maskable and maskable interrupts are incorporated as interrupt sources (see **Table 11-1**).

chanhvinh.com

Table 11-1. Interrupt Source List

Type of Interrupt	Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Nonmaskable	-	INTWDT	Watchdog timer overflow (when watchdog timer mode 1 is selected)	Internal	0004H	(A)
Maskable	0	INTWDT	Watchdog timer overflow (when interval timer mode is selected)			(B)
	1	INTUSBTM	USB timer overflow		0006H	(B)
	2	INTUSBRT	EOP detection when a USB token packet is received		0008H	
	3	INTUSBRD	EOP detection when a USB data/handshake packet is received		000AH	
	4	INTUSBST	EOP detection when a USB data/ handshake packet is transmitted		000CH	
	5	INTUSBRE	Detection of transition from J state to K state or SE0 on the USB bus		000EH	
	6	INTP0	Detection of a pin input edge	External	0010H	(C)
	7	INTCSI10	End of three-wire SIO bus interface transmission and reception	Internal	0012H	(B)
	8	INTTM00	Generation of the 8-bit timer 00 match signal		0014H	
	9	INTTM01	Generation of the 8-bit timer/event counter 01 match signal		0016H	
	10	INTKR00	Detection of the key return signal	External	0018H	(C)

Notes 1. The priority is the order of priority when multiple maskable interrupts are generated simultaneously. 0 is the highest priority and 10 is the lowest.

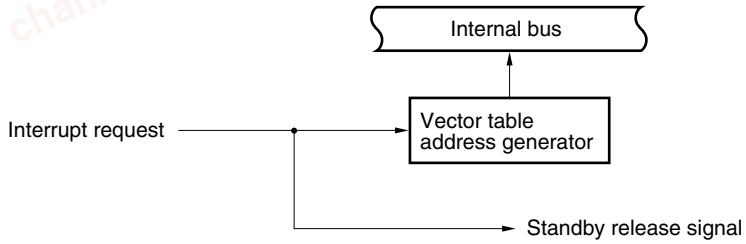
2. Types (A) to (C) in the basic configuration correspond to (A) to (C) in Figure 11-1, respectively.

★ **Remark** Only one of the two watchdog timer interrupt (INTWDT) sources, non-maskable or maskable (internal), can be selected.

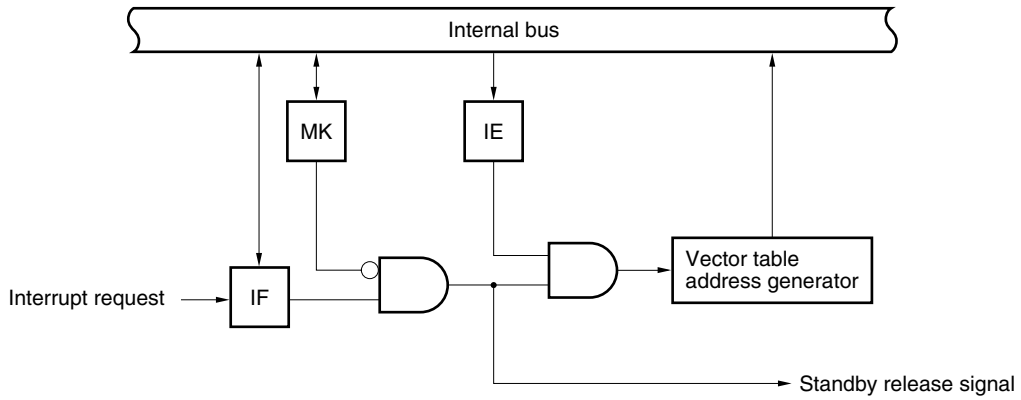
chanhvinh.com

Figure 11-1. Basic Configuration of Interrupt Function

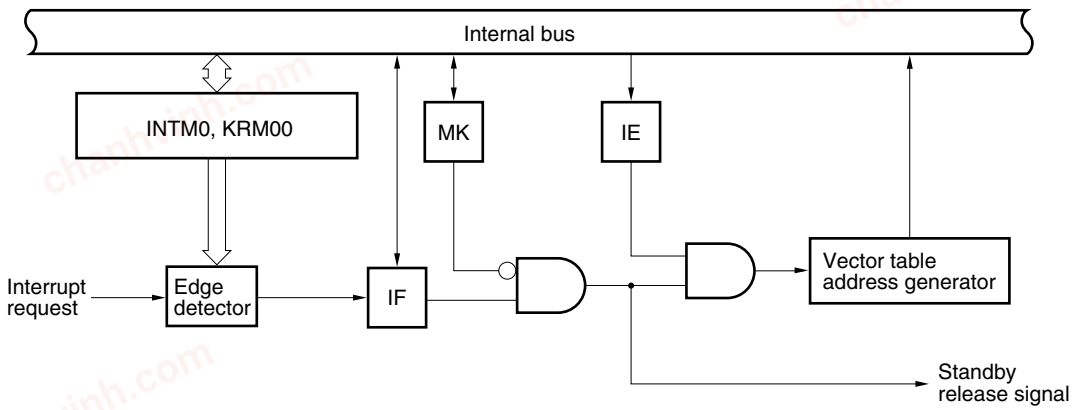
(A) Internal non-maskable interrupt



(B) Internal maskable interrupt



(C) External maskable interrupt



IF: Interrupt request flag
IE: Interrupt enable flag
MK: Interrupt mask flag
INTM0: External interrupt mode register 0
KRM00: Key return mode register 00

chanhvinh.com

11.3 Registers Controlling Interrupt Function

The following five registers are used to control the interrupt functions.

- Interrupt request flag registers 0 and 1 (IF0 and IF1)
- Interrupt mask flag registers 0 and 1 (MK0 and MK1)
- External interrupt mode register 0 (INTM0)
- Program status word (PSW)
- Key return mode register 00 (KRM00)

Table 11-2 gives a listing of interrupt request flag and interrupt mask flag names corresponding to interrupt requests.

Table 11-2. Flags Corresponding to Interrupt Request Signals

Interrupt Request Signal Name	Interrupt Request Flag	Interrupt Mask Flag
INTWDT	TMIF4	TMMK4
INTUSBTM	USBTMIF	USBTMMK
INTUSBRT	USBRTIF	USBRTMK
INTUSBRD	USBRDIF	USBRDMK
INTUSBST	USBSTIF	USBSTMK
INTUSBRE	USBREIF	USBREMK
INTP0	PIF0	PMK0
INTCSI10	CSIIF10	CSIMK10
INTTM00	TMIF00	TMMK00
INTTM01	TMIF01	TMMK01
INTKR00	KRIF00	KRMK00

chanhvinh.com

(1) Interrupt request flag registers (IF0 and IF1)

The interrupt request flag is set to 1 when the corresponding interrupt request is generated or an instruction is executed. It is cleared to 0 when an instruction is executed upon acknowledgement of an interrupt request or upon $\overline{\text{RESET}}$ input.

IF0 and IF1 are set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IF0 and IF1 to 00H.

Figure 11-2. Format of Interrupt Request Flag Register

Symbol	<7>	<6>	<5>	<4>	3	2	<1>	<0>	Address	After reset	R/W
IF0	TMIF01	TMIF00	CSIIF10	KRIF00	0	0	PIF0	TMIF4	FFE0H	00H	R/W

	7	<6>	<5>	<4>	<3>	<2>	1	0			
IF1	0	USBTMIF	USBRTIF	USBRDIF	USBSTIF	USBREIF	0	0	FFE1H	00H	R/W

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request signal is generated; interrupt request state

Cautions

1. **The TMIF4 flag is R/W enabled only when the watchdog timer is used as an interval timer. If watchdog timer mode 1 or 2 is used, set the TMIF4 flag to 0.**
2. **Because port 2 has an alternate function as an external interrupt input, when the output level is changed by specifying the output mode of the port function, an interrupt request flag is set. Therefore, the interrupt mask flag should be set to 1 before using the output mode.**
3. ★ **If an interrupt is acknowledged, an interrupt request flag is automatically cleared and then the interrupt routine is entered.**

chanhvinh.com

(2) Interrupt mask flag registers (MK0 and MK1)

The interrupt mask flag is used to enable/disable the corresponding maskable interrupt servicing.

MK0 and MK1 are set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets MK0 and MK1 to FFH.

Figure 11-3. Format of Interrupt Mask Flag Register

Symbol	<7>	<6>	<5>	<4>	3	2	<1>	<0>	Address	After reset	R/W
MK0	TMMK01	TMMK00	CSIMK10	KRMK00	1	1	PMK0	TMMK4	FFE4H	FFH	R/W

	7	<6>	<5>	<4>	<3>	<2>	1	0			
MK1	0	USBTMMK	USBRTMK	USBRDMK	USBSTMK	USBREIF	1	1	FFE5H	FFH	R/W

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Cautions

1. **If the TMMK4 flag is read when the watchdog timer is used in watchdog timer mode 1 or 2, its value becomes undefined.**
2. **Because port 2 has an alternate function as an external interrupt input, when the output level is changed by specifying the output mode of the port function, an interrupt request flag is set. Therefore, the interrupt mask flag should be set to 1 before using the output mode.**

(3) External interrupt mode register 0 (INTM0)

This register is used to set the valid edge of INTP0.

INTM0 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets INTM0 to 00H.

Figure 11-4. Format of External Interrupt Mode Register 0

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
INTM0	0	0	0	0	ES01	ES00	0	0	FFECH	00H	R/W

ES01	ES00	INTP0 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

Cautions

1. **Bits 0, 1 and 4 to 7 must be set to 0.**
2. **Before setting the INTM0 register, be sure to set xxMKx of the relevant interrupt mask flag to 1 to disable interrupts. After that clear the interrupt mask flag (xxMKx = 0) to enable interrupts after clearing the interrupt request flag(xxIFx = 0).**

chanhvinh.com

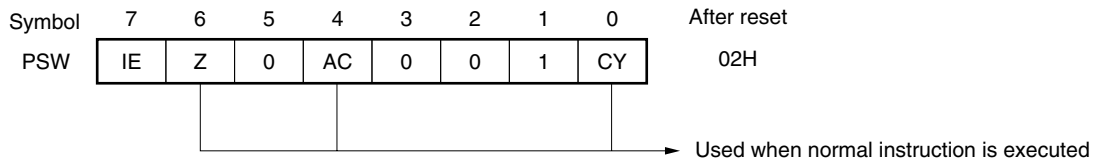
(4) Program status word (PSW)

The program status word is a register used to hold the instruction execution result and the current status for interrupt requests. The IE flag to set maskable interrupt enable/disable is mapped here.

Besides 8-bit unit read/write, this register can carry out operations with bit manipulation instructions and dedicated instructions (EI, DI). When a vectored interrupt is acknowledged, the PSW is automatically saved into a stack, and the IE flag is reset to 0.

$\overline{\text{RESET}}$ input sets the PSW to 02H.

Figure 11-5. Configuration of Program Status Word



IE	Interrupt acknowledgment enabled/disabled
0	Disabled
1	Enabled

chanhvinh.com

(5) Key return mode register 00 (KRM00)

This register sets the pin that detects a key return signal (rising edge of port 4).

KRM00 is set with a 1-bit an 8-bit memory manipulation instruction.

Bit 0 (KRM000) is set in 4-bit units for the $\overline{KR00}$/P40 to $\overline{KR03}$/P43 pins. Bits 4 to 7 (KRM004 to KRM007) are set in 1-bit units for the $\overline{KR04}$/P44 to $\overline{KR07}$/P47 pins, respectively.

$\overline{RESET}$ input sets KRM00 to 00H.

Figures 11-6 and 11-7 show the format of key return mode register 00 and the block diagram of the falling edge detector, respectively.

Figure 11-6. Format of Key Return Mode Register 00

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
KRM00	KRM007	KRM006	KRM005	KRM004	0	0	0	KRM000	FFF5H	00H	R/W

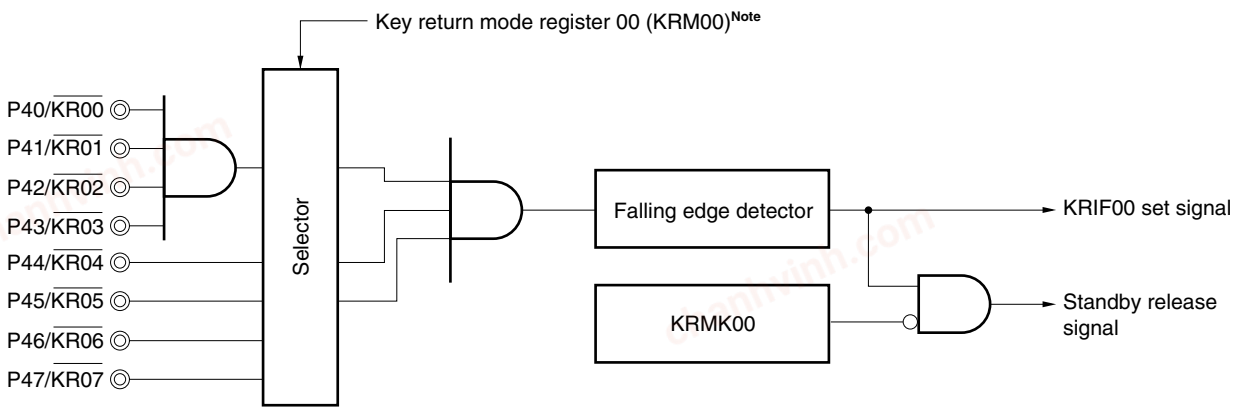
KRM00n	Key return signal detection selection for P4n/$\overline{KR0n}$ pin (n = 4 to 7)
0	No detection
1	Detection (detecting falling edge of P4n/$\overline{KR0n}$)

KRM000	Key return signal detection selection for P40/$\overline{KR00}$ to P43/$\overline{KR03}$ pins
0	No detection
1	Detection (detecting falling edge of P40/$\overline{KR00}$ to P43/$\overline{KR03}$)

Cautions

1. **Bits 1 to 3 must be set to 0.**
2. **When the KRM00 register is set to 1, a pull-up resistor is connected automatically. However, the pull-up resistor is cut for pins in output mode.**
3. **Before setting KRM00, always set bit 4 of MK0 (KRMK00 = 1) to disable interrupts in advance. After setting KRM00, clear bit 4 of MK0 (KRMK00 = 0) after clearing bit 4 of IF0 (KRIF00 = 0) to enable interrupts.**

Figure 11-7. Block Diagram of Falling Edge Detector



Note Register that selects the pin used for falling edge input.

chanhvinh.com

11.4 Interrupt Servicing Operation

11.4.1 Non-maskable interrupt acknowledgment operation

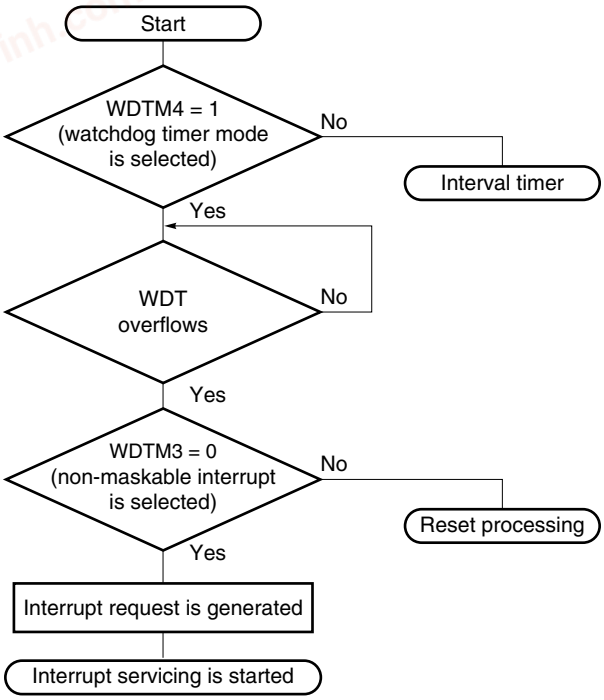
The non-maskable interrupt is unconditionally acknowledged even when interrupts are disabled. It is not subject to interrupt priority control and takes precedence over all other interrupts.

When the non-maskable interrupt request is acknowledged, the PSW and PC are saved to the stack in that order, the IE flag is reset to 0, the contents of the vector table are loaded to the PC, and then program execution branches.

Caution During non-maskable interrupt service program execution, do not input another non-maskable interrupt request; if it is input, the service program will be interrupted and the new interrupt request will be acknowledged.

chanhvinh.com

Figure 11-8. Flowchart of Non-Maskable Interrupt Request Acknowledgment



WDTM: Watchdog timer mode register
WDT: Watchdog timer

Figure 11-9. Timing of Non-Maskable Interrupt Request Acknowledgment

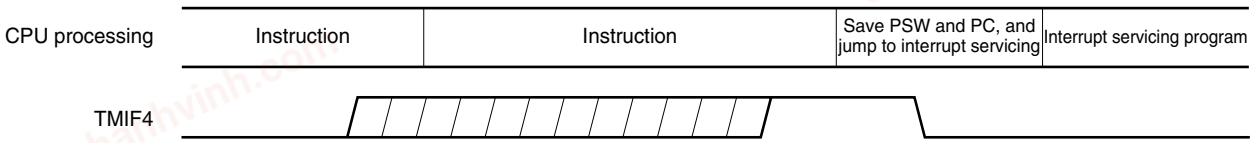
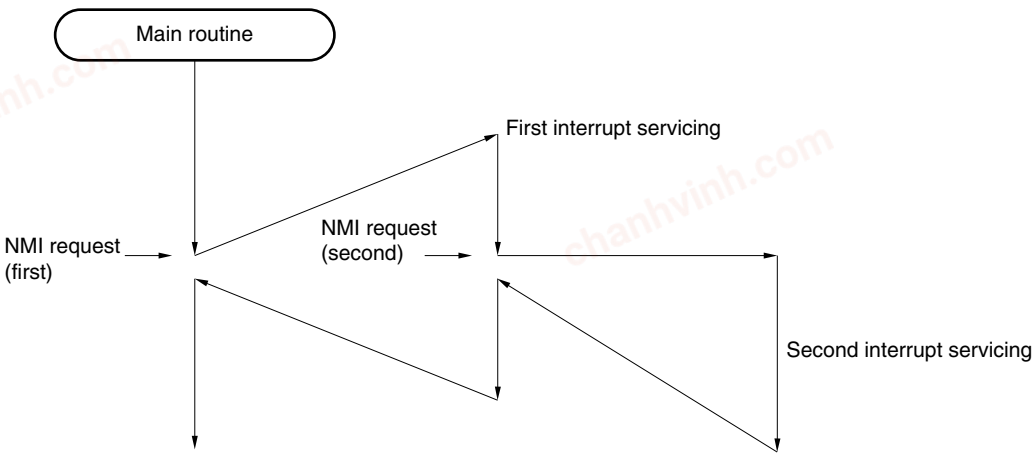


Figure 11-10. Acknowledging Non-Maskable Interrupt Request



chanhvinh.com

11.4.2 Maskable interrupt acknowledgment operation

A maskable interrupt can be acknowledged when the interrupt request flag is set to 1 and the corresponding interrupt mask flag is cleared to 0. A vectored interrupt is acknowledged in the interrupt enabled status (when the IE flag is set to 1).

The time required to start the interrupt servicing after a maskable interrupt request has been generated is as follows.

Table 11-3. Time from Generation of Maskable Interrupt Request to Servicing

Minimum Time	Maximum Time[Note]
8 clocks	19 clocks

Note The wait time is maximum when an interrupt request is generated immediately before a BT or BF instruction.

Remark 1 clock: $\frac{1}{f_{CPU}}$ (f_{CPU}: CPU clock)

When two or more maskable interrupt requests are generated at the same time, they are acknowledged starting from the one assigned the highest priority.

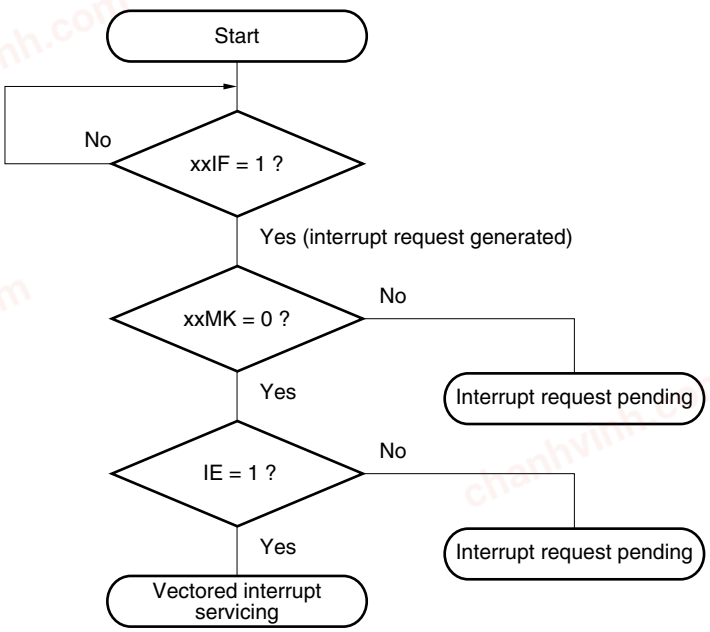
A pending interrupt is acknowledged when the status in which it can be acknowledged is set.

Figure 11-11 shows the algorithm of acknowledging interrupts.

When a maskable interrupt request is acknowledged, the PSW and PC are saved to the stack in that order, the IE flag is reset to 0, the data in the vector table determined for each interrupt request is loaded to the PC, and execution branches.

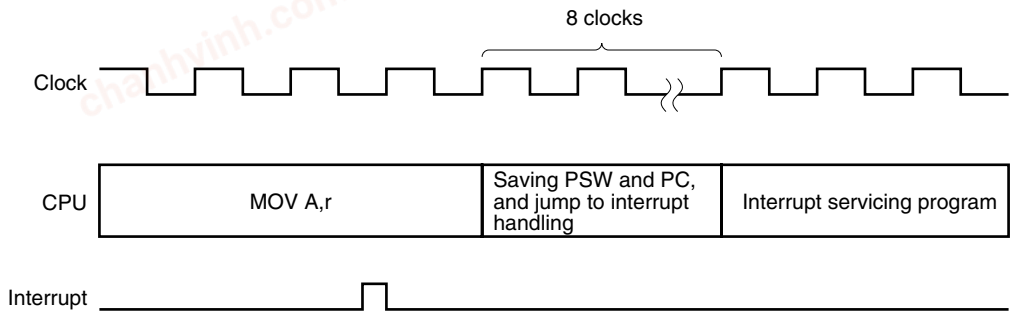
To return from interrupt servicing, use the RETI instruction.

Figure 11-11. Interrupt Acknowledgment Program Algorithm



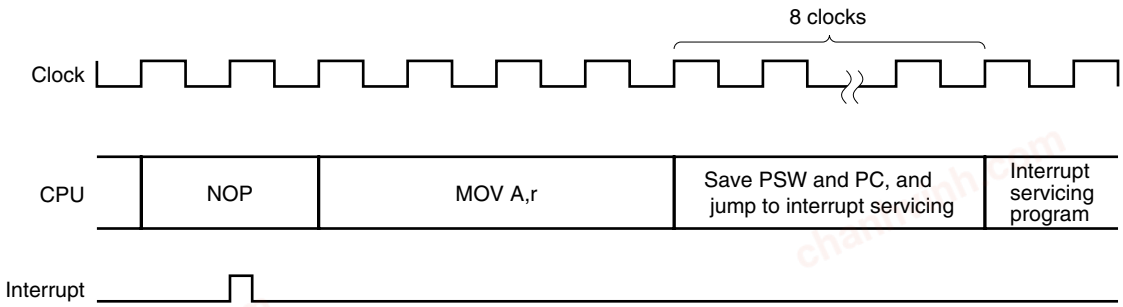
chanhvinh.com

Figure 11-12. Timing of Interrupt Request Acknowledgment (Example of MOV A,r)



When an interrupt request flag (xxIF) is generated before clock n (n = 4 to 10) of the instruction being executed turns to n - 1, the interrupt is acknowledged after the instruction has been executed. Figure 11-12 shows an example for 8-bit data transfer instruction MOV A,r. It takes 4 clocks for this instruction to be executed. So, when an interrupt occurs within 3 clocks after the instruction execution started, the interrupt is acknowledged after MOV A,r has been executed.

Figure 11-13. Timing of Interrupt Request Acknowledgment (When Interrupt Request Flag Is Generated at Last Clock of Instruction Execution)



When an interrupt request flag (xxIF) is generated at the last clock for instruction execution, after the next instruction has been executed, the interrupt acknowledgment servicing starts.

Figure 11-13 shows an example when an interrupt request flag is generated at the second clock for NOP (2-clock instruction). In this case, the interrupt is acknowledged after MOV A,r has been executed after the NOP instruction execution.

Caution Interrupt requests are held pending during access to interrupt request flag registers 0, 1 (IF0, IF1) or interrupt mask flag registers 0, 1 (MK0, MK1).

chanhvinh.com

11.4.3 Multiplexed interrupt servicing

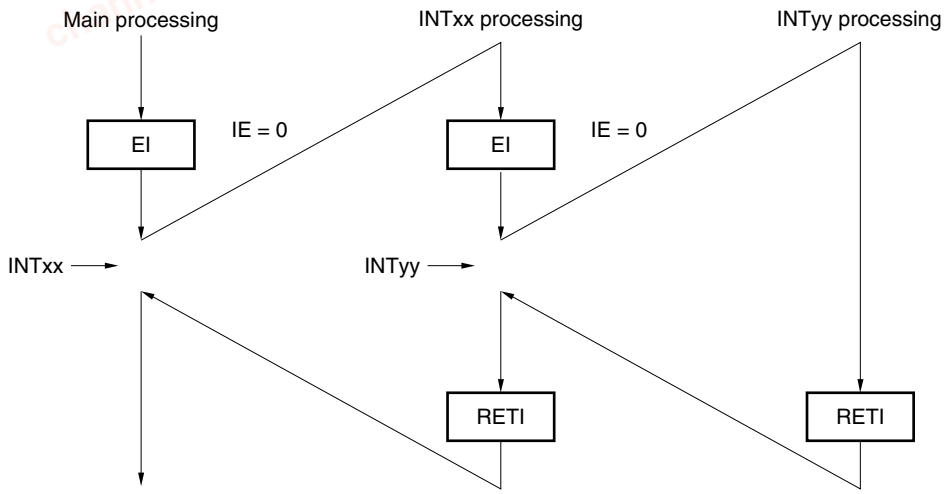
Servicing in which another interrupt is acknowledged while an interrupt is being processed is called multiplexed interrupt servicing.

Multiplexed interrupt is not performed unless interrupt requests are enabled (IE = 1) (except the non-maskable interrupt request). Other interrupt requests are disabled (IE = 0) as soon as an interrupt request is acknowledged. Therefore, it is necessary to set (1) the IE flag to realize the interrupt enable state using the EI instruction during interrupt request servicing in order to enable multiplexed interrupt servicing.

chanhvinh.com

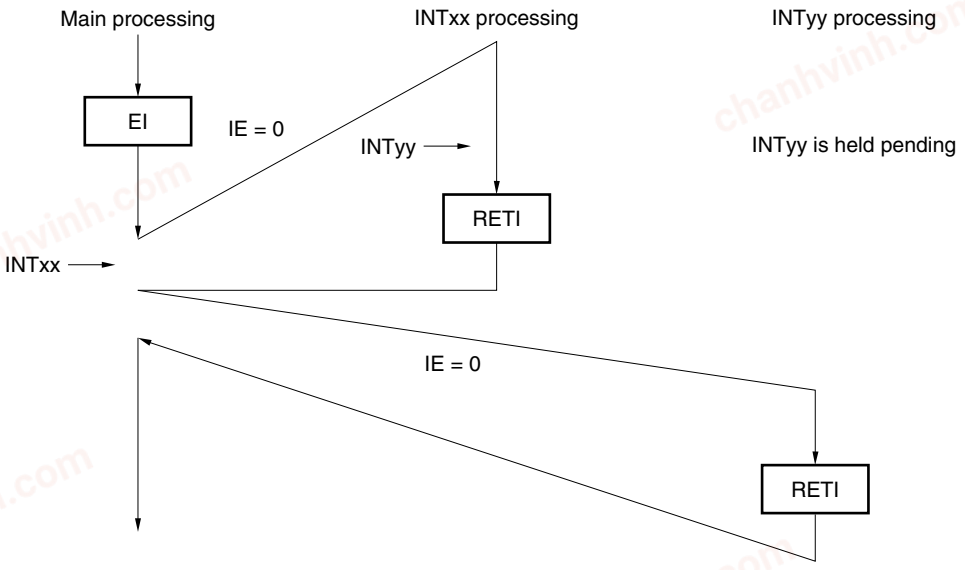
Figure 11-14. Example of Multiplexed Interrupt Servicing

Example 1. Acknowledging multiplexed interrupts



The interrupt request INTyy is acknowledged and multiplexed interrupt servicing is performed during the interrupt INTxx servicing. Before each interrupt is acknowledged, the IE instruction is issued and interrupt requests are enabled.

Example 2. Multiplexed interrupts are not performed because interrupts are disabled.



Interrupt requests are disabled (EI instruction is not issued) in the interrupt INTxx servicing. The interrupt request INTyy is not acknowledged and multiplexed interrupt servicing is not performed. INTyy is held pending and is acknowledged after INTxx servicing is completed.

IE = 0: Interrupt request acknowledgment disabled.

chanhvinh.com

11.4.4 Interrupt request hold

If an interrupt (such as a maskable, non-maskable, or external interrupt) is requested when a certain type of instruction is being executed, the interrupt request will not be acknowledged until the instruction is completed. Such instructions include:

- Instructions that manipulate interrupt request flag registers 0 and 1 (IF0 and IF1)
- Instructions that manipulate interrupt mask flag registers 0 and 1 (MK0 and MK1)

chanhvinh.com

CHAPTER 12 STANDBY FUNCTION

12.1 Standby Function and Configuration

12.1.1 Standby function

The standby function is used to reduce the power consumption of the system and can be effected in the following two modes.

(1) HALT mode

This mode is set when the HALT instruction is executed. The HALT mode stops the operation clock of the CPU. The system clock oscillator continues oscillating. This mode does not reduce the power consumption as much as the STOP mode, but is useful for resuming processing immediately when an interrupt request is generated, or for intermittent operations.

(2) STOP mode

This mode is set when the STOP instruction is executed. The STOP mode stops the system clock oscillator and stops the entire system. The power consumption of the CPU can be substantially reduced in this mode.

The STOP mode can be released by an interrupt request, so that this mode can be used for intermittent operations. However, some time is required until the system clock oscillator stabilizes after the STOP mode has been released. If processing must be resumed immediately by using an interrupt request, therefore, use the HALT mode.

In both modes, the previous contents of the registers, flags, and data memory before setting the standby mode are all retained. In addition, the statuses of the output latches of the I/O ports and output buffers are also retained.

Caution To set the STOP mode, be sure to stop the operations of the peripheral hardware, and then execute the STOP instruction.

chanhvinh.com

12.1.2 Register controlling standby function

The wait time after the STOP mode is released upon interrupt request until oscillation stabilizes is controlled with the oscillation stabilization time select register (OSTS).

OSTS is set with an 8-bit memory manipulation instruction.

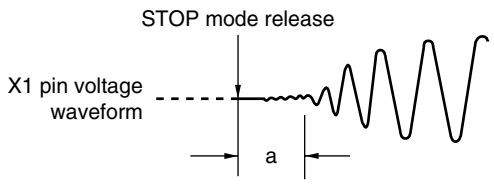
$\overline{\text{RESET}}$ input sets OSTS to 04H. However, it takes 2^{15}/fx until oscillation stabilizes after $\overline{\text{RESET}}$ input.

Figure 12-1. Format of Oscillation Stabilization Time Select Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0	FFFAH	04H	R/W

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection
0	0	0	2^{12}/fx (683 μs)
0	1	0	2^{15}/fx (5.46 ms)
1	0	0	2^{17}/fx (21.8 ms)
Other than above			Setting prohibited

Caution The wait time after the STOP mode is released does not include the time from STOP mode release to clock oscillation start ("a" in the figure below), regardless of release by $\overline{\text{RESET}}$ input or by interrupt generation.



Remarks 1. fx: System clock oscillation frequency

2. The parenthesized values apply to operation at fx = 6.0 MHz.

chanhvinh.com

12.2 Standby Function Operation

12.2.1 HALT mode

(1) HALT mode

The HALT mode is set by executing the HALT instruction.

The operation status in the HALT mode is shown in the following table.

Table 12-1. HALT Mode Operation Status

Item	HALT Mode Operation Status
Clock generator	Oscillation enabled
CPU	Operation disabled
Port (output latch)	Remains in the state before the selection of HALT mode.
8-bit timer 00 (TM00)	Operation enabled
8-bit timer/event counter 01 (TM01)	Operation enabled
Watchdog timer	Operation enabled
USB function	Operation enabled
Serial interface	Operation enabled
Key return	Operation enabled[Note 1]
External interrupt	Operation enabled[Note 2]

Notes 1. Operation is enabled only for pins set in key return mode register 00 (KRM00).

2. Maskable interrupt that is not masked

chanhvinh.com

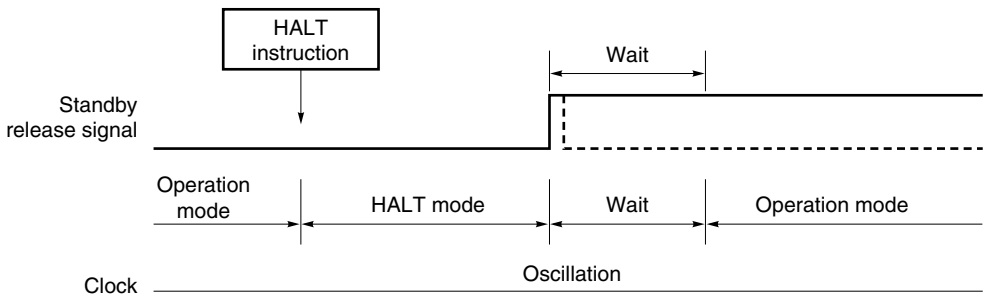
(2) Releasing HALT mode

The HALT mode can be released by the following three sources.

(a) Releasing by unmasked interrupt request

The HALT mode is released by an unmasked interrupt request. In this case, if interrupts are enabled to be acknowledged, vectored interrupt servicing is performed. If interrupts are disabled, the instruction at the next address is executed.

Figure 12-2. Releasing HALT Mode by Interrupt



Remarks 1. The broken lines indicate the case where the interrupt request that has released the standby mode is acknowledged.

2. The wait time is as follows.

- When vectored interrupt servicing is performed: 9 to 10 clocks
- When vectored interrupt servicing is not performed: 1 to 2 clocks

(b) Releasing by non-maskable interrupt request

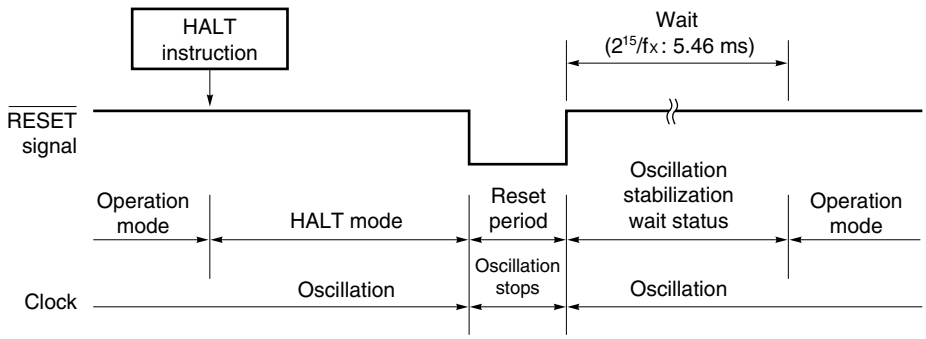
The HALT mode is released regardless of whether interrupts are enabled or disabled, and vectored interrupt servicing is performed.

chanhvinh.com

(c) Releasing by $\overline{\text{RESET}}$ input

When the HALT mode is released by the $\overline{\text{RESET}}$ signal, execution branches to the reset vector address in the same manner as the ordinary reset operation, and program execution is started.

Figure 12-3. Releasing HALT Mode by $\overline{\text{RESET}}$ Input



Remarks 1. fx: System clock oscillation frequency

2. The parenthesized values apply to operation at fx = 6.0 MHz.

Table 12-2. Operation After Release of HALT Mode

Releasing Source	MKxx	IE	Operation
Maskable interrupt request	0	0	Executes next address instruction
	0	1	Executes interrupt servicing
	1	×	Retains HALT mode
Non-maskable interrupt request	—	×	Executes interrupt servicing
$\overline{\text{RESET}}$ input	—	—	Reset processing

×: don't care

chanhvinh.com

12.2.2 STOP mode

(1) Setting and operation status of STOP mode

The STOP mode is set by executing the STOP instruction.

Cautions 1. When the STOP mode is set, the X2 pin is internally pulled up to V_{DD} to suppress the current leakage of the crystal oscillator block. Therefore, do not use the STOP mode in a system where an external clock is used as the system clock.

2. Because the standby mode can be released by an interrupt request signal, the standby mode is released as soon as it is set if there is an interrupt source whose interrupt request flag is set and interrupt mask flag is reset. When the STOP mode is set, therefore, the HALT mode is set immediately after the STOP instruction has been executed, the wait time set by the oscillation stabilization time select register (OSTS) elapses, and then the operation mode is set.

The operation status in the STOP mode is shown in the following table.

Table 12-3. STOP Mode Operation Status

Item	STOP Mode Operation Status
Clock generator	Oscillation disabled
CPU	Operation disabled
Port (output latch)	Remains in the state before the selection of STOP mode.
8-bit timer 00 (TM00)	Operation disabled
8-bit timer/event counter 01 (TM01)	Operation enabled[Note 1]
Watchdog timer	Operation disabled
USB function	Operation enabled[Note 2]
Serial interface 10	Operation enabled[Note 3]
Key return	Operation enabled[Note 4]
External interrupt	Operation enabled[Note 5]

Notes 1. Operation is enabled only when TI01 is selected as the count clock.

2. Operation is enabled only when the USB reset or Resume signal is received.

3. Operation is enabled only when an external clock is selected.

4. Operation is enabled only for pins set in key return mode register 00 (KRM00).

5. Maskable interrupt that is not masked

chanhvinh.com

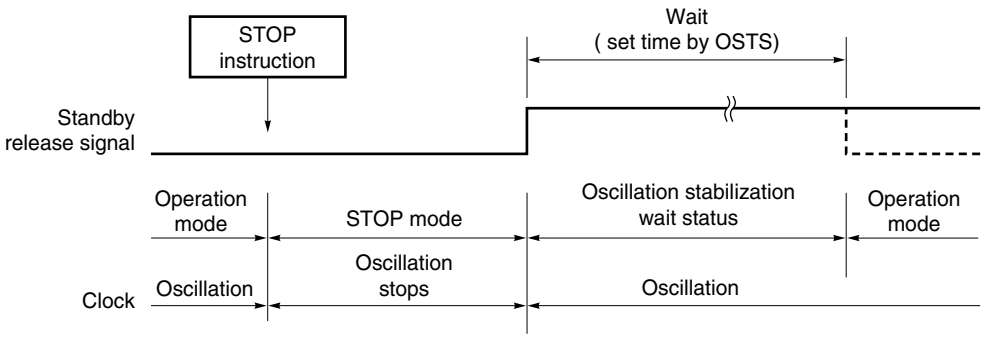
(2) Releasing STOP mode

The STOP mode can be released by the following two sources.

(a) Releasing by unmasked interrupt request

The STOP mode can be released by an unmasked interrupt request. In this case, if interrupts are enabled to be acknowledged, vectored interrupt servicing is performed, after the oscillation stabilization time has elapsed. If interrupt acknowledgement is disabled, the instruction at the next address is executed.

Figure 12-4. Releasing STOP Mode by Interrupt



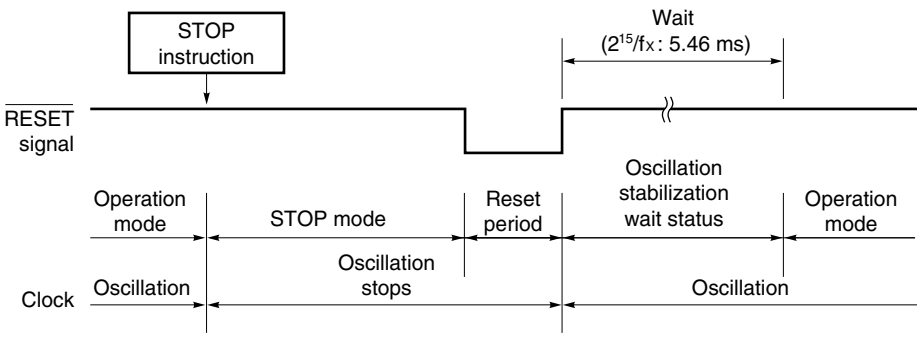
Remark The broken lines indicate the case where the interrupt request that has released the standby mode is acknowledged.

chanhvinh.com

(b) Releasing by $\overline{\text{RESET}}$ input

When the STOP mode is released by the $\overline{\text{RESET}}$ signal, the reset operation is performed after the oscillation stabilization time has elapsed.

Figure 12-5. Releasing STOP Mode by $\overline{\text{RESET}}$ Input



Remarks 1. fx: System clock oscillation frequency

2. The parenthesized values apply to operation at 6.0 MHz.

Table 12-4. Operation After Release of STOP Mode

Releasing Source	MKxx	IE	Operation
Maskable interrupt request	0	0	Executes next address instruction
	0	1	Executes interrupt servicing
	1	×	Retains STOP mode
$\overline{\text{RESET}}$ input	—	—	Reset processing

×: don't care

★ **(3) Cautions an executing STOP instruction**

After the STOP instruction is executed in the SE0 state (USBDM = 0, USBDP = 0), the STOP mode cannot be released by a USB reset/Resume detection interrupt (INTUSBRE).

Therefore, the following control should be performed.

<Control method>

Do not execute the STOP instruction in the SE0 state.

In addition, when executing the STOP instruction in the suspend mode, execute both the following two software countermeasures.

- Do not clear the USB reset/Resume detection interrupt request flag (USBREIF) between when the suspend state is detected and when the STOP instruction is executed.
- Clear the USB reset/Resume detection interrupt request flag (USBREIF) when the 8-bit timer that is used as the 3 ms timer for suspend state detection is reset.

chanhvinh.com

CHAPTER 13 RESET FUNCTION

The following two operations are available to generate reset signals.

(1) External reset input via $\overline{\text{RESET}}$ pin
(2) Internal reset by inadvertent program loop time detected by watchdog timer

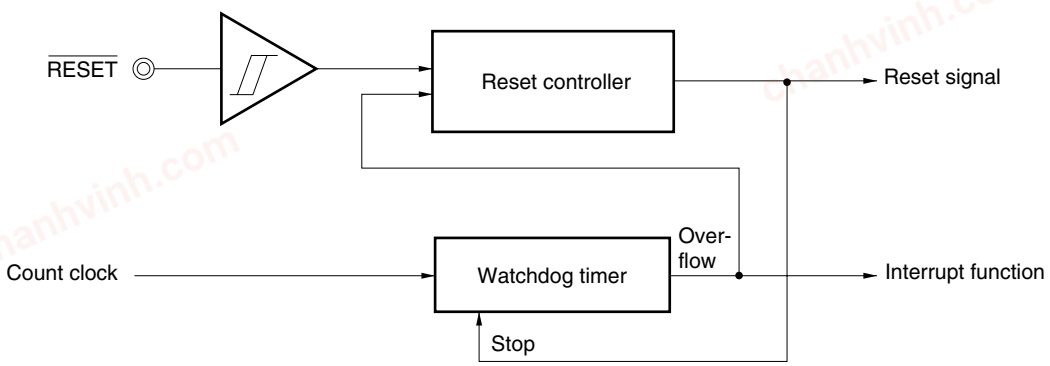
External and internal reset have no functional differences. In both cases, program execution starts at the address at 0000H and 0001H by $\overline{\text{RESET}}$ input.

When a low level is input to the $\overline{\text{RESET}}$ pin or the watchdog timer overflows, a reset is applied and each of the hardware is set to the status shown in Table 13-1. Each pin is high impedance during reset input or during the oscillation stabilization time just after reset release.

When a high level is input to the $\overline{\text{RESET}}$ pin, the reset is released and program execution is started after the oscillation stabilization time (2^{15}/fx) has elapsed. The reset applied by watchdog timer overflow is automatically released after reset, and program execution is started after the oscillation stabilization time (2^{15}/fx) has elapsed (see **Figures 13-2** through **13-4**.)

Cautions
1. **For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.**
2. **When the STOP mode is released by reset, the STOP mode contents are held during reset input. However, the port pins become high impedance.**

Figure 13-1. Block Diagram of Reset Function



chanhvinh.com

Figure 13-2. Reset Timing by $\overline{\text{RESET}}$ Input

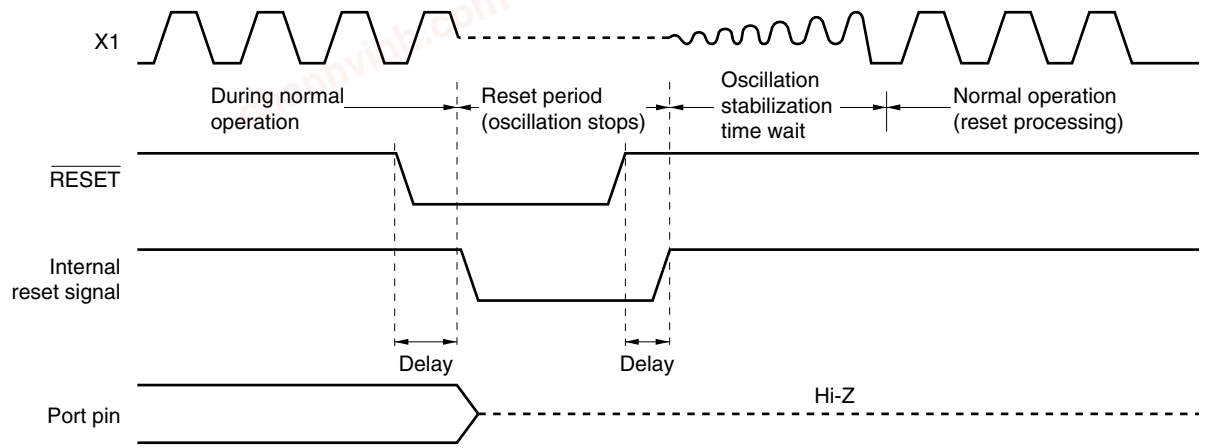


Figure 13-3. Reset Timing by Overflow in Watchdog Timer

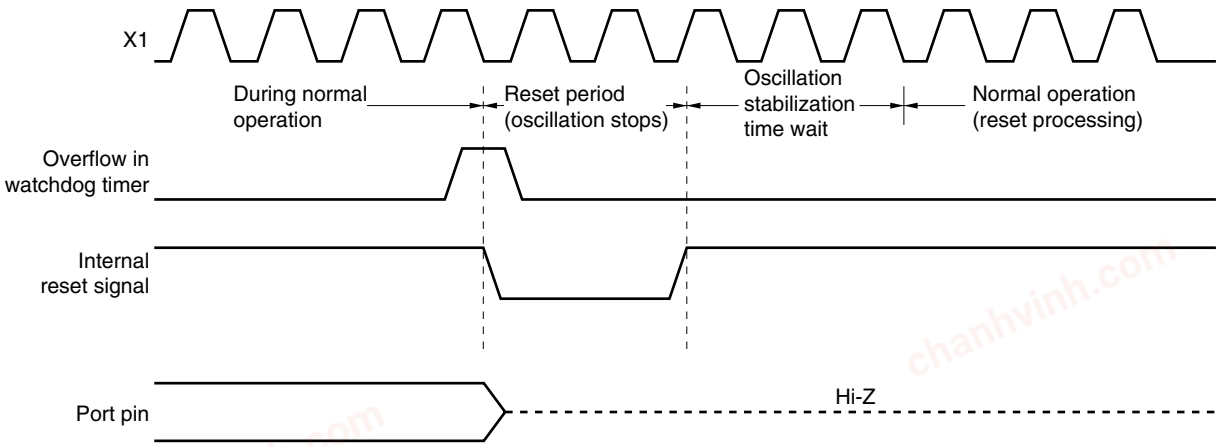
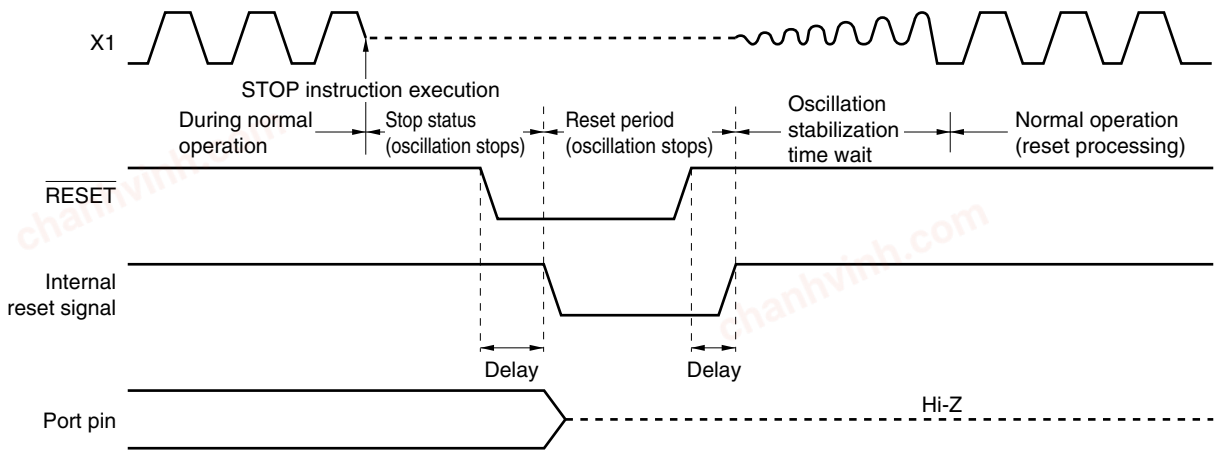


Figure 13-4. Reset Timing by $\overline{\text{RESET}}$ Input in STOP Mode



chanhvinh.com

Table 13-1. Hardware Status After Reset (1/2)

Hardware		Status After Reset
Program counter (PC)[Note 1]		The contents of reset vector tables (0000H and 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined[Note 2]
	General-purpose register	Undefined[Note 2]
Port (P0 to P2, P4) output latch		00H
Port mode register (PM0 to PM2, PM4)		FFH
Pull-up resistor option register (PU0)		00H
Port output mode register (POM0, POM1)		00H
Processor clock control register (PCC)		02H
Oscillation stabilization time select register (OSTS)		04H
8-bit timer/event counter	Timer counter (TM00, TM01)	00H
	Compare register (CR00, CR01)	Undefined
	Mode control register (TMC00, TMC01)	00H
Watchdog timer	Timer clock select register (TCL2)	00H
	Mode register (WDTM)	00H
USB function	Transmit receive pointer (USBPOW)	00H
	Receive token PID (USBRTP)	00H
	Receive token address (USBRAL, USBRAH)	00H
	Receive data PID (USBRD)	00H
	Receive data address (USBR0 to USBR7)	Undefined
	Transmit data PID bank (USBTD0, USBTD1)	Undefined
	Transmit data bank address (USBT00 to USBT07, USBT10 to USBT17)	Undefined
	Data/handshake packet receive byte number counter (DRXCON)	18H

Notes 1. During reset input and oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined.
All other hardware remains unchanged after reset.

2. The post-reset values are retained in the standby mode.

chanhvinh.com

Table 13-1. Hardware Status After Reset (2/2)

Hardware		Status After Reset
USB function	Data packet transmit byte number counter 0 (DTXCO0)	20H
	Data packet transmit byte number counter 1 (DTXCO1)	30H
	Token PID compare register (TIDCMP)	00H
	Token address compare register (ADRCMP)	00H
	USB receiver enable register (USBMOD)	00H
	Packet receive status register (RXSTAT)	00H
	Data/handshake packet receive result store register (DRXRSL)	00H
	Token packet receive result store register (TRXRSL)	00H
	Data/handshake PID compare register (DIDCMP)	C3H
	Data packet transmit reservation register (DTXRSV)	00H
	Handshake packet transmit reservation register (HTXRSV)	00H
	USB timer start reservation control register (USBTCL)	01H
	Remote wakeup control register (REMWUP)	08H
Serial interface	Mode register (CSIM10)	00H
	Transmit/receive shift register (SIO10)	Undefined
Interrupt	Request flag register (IF0, IF1)	00H
	Mask flag register (MK0, MK1)	FFH
	External interrupt mode register (INTM0)	00H
	Key return mode register (KRM00)	00H

chanhvinh.com

CHAPTER 14 μPD78F9801

The μPD78F9801 is a product that substitutes flash memory for the internal ROM of the mask ROM version. The differences between the μPD78F9801 and the mask ROM versions are shown in Table 14-1.

Table 14-1. Differences Between μPD78F9801 and Mask ROM Versions

Item		Flash Memory Version	Mask ROM Version
		μPD78F9801	μPD789800
Internal memory	ROM	16 KB (Flash memory)	8 KB
	High-speed RAM	256 bytes	
IC pin		Not provided	Provided
V_{PP} pin		Provided	Not provided
Electrical specifications		Refer to **CHAPTER 16 ELECTRICAL SPECIFICATIONS**.	

Caution There are differences in noise immunity and noise radiation between the flash memory and mask ROM versions. When pre-producing an application set with the flash memory version and then mass-producing it with the mask ROM version, be sure to conduct sufficient evaluations for the commercial samples (not engineering samples) of the mask ROM version.

chanhvinh.com

★ 14.1 Flash Memory Characteristics

Flash memory programming is performed by connecting a dedicated flash programmer (Flashpro III (part no. FL-PR3, PG-FP3)/Flashpro IV (part no. FL-PR4, PG-FP4)) to the target system with the μPD78F9801 mounted on the target system (on-board). A flash memory program adapter (FA adapter), which is a target board used exclusively for programming, is also provided.

Remark FL-PR3, FL-PR4, and the program adapter are products made by Naito Densei Machida Mfg. Co., Ltd. (TEL +81-45-475-4191).

Programming using flash memory has the following advantages.

- Software can be modified after the microcontroller is solder-mounted on the target system.
- Distinguishing software facilities small-quantity, varied model production
- Easy data adjustment when starting mass production

14.1.1 Programming environment

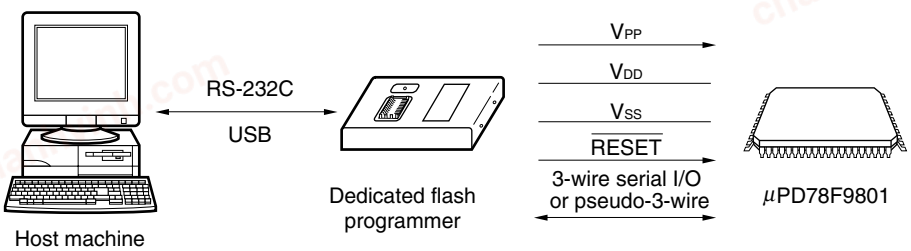
The following shows the environment required for μPD78F9801 flash memory programming.

When Flashpro III (part no. FL-PR3, PG-FP3) or Flashpro IV (part no. FL-PR4, PG-FP4) is used as a dedicated flash programmer, a host machine is required to control the dedicated flash programmer. Communication between the host machine and flash programmer is performed via RS-232C/USB (Rev. 1.1).

For details, refer the manuals for Flashpro III/Flashpro IV.

Remark USB is supported by Flashpro IV only.

Figure 14-1. Environment for Writing Program to Flash Memory



chanhvinh.com

14.1.2 Communication mode

Use the communication mode shown in Table 14-2 to perform communication between the dedicated flash programmer and μPD78F9801.

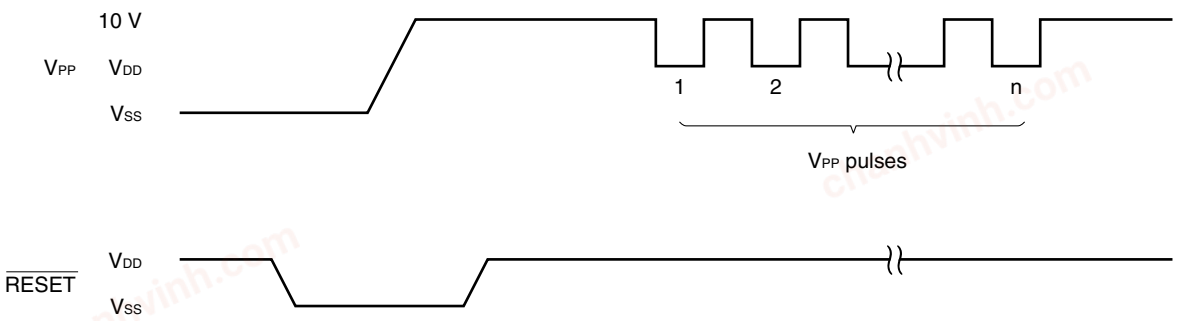
Table 14-2. Communication Mode List

Communication Mode	TYPE Setting[Note 1]					Pins Used	Number of V_{PP} Pulses
	COMM PORT	SIO Clock	CPU Clock	Flash Clock	Multiple Rate		
3-wire serial I/O	SIO ch-0 (3-wire, sync.)	100 Hz to 1.25 MHz[Note 2]	Optional	1 to 5 MHz[Note 2]	1.0	SI10/P22 SO10/P21 $\overline{SCK10}$/P20	0
Pseudo-3-wire	Port A (pseudo-3-wire)	100 Hz to 1 kHz	Optional	1 to 5 MHz[Note 2]	1.0	P10 (serial clock input) P11 (serial data output) P12 (serial data input)	12

Notes 1. Selection items for TYPE settings on the dedicated flash programmer (Flashpro III (part no. FL-PR3, PG-FP3)/Flashpro IV (part no. FL-PR4, PG-FP4)).

2. The possible setting range differs depending on the voltage. For details, refer to **CHAPTER 16 ELECTRICAL SPECIFICATIONS**.

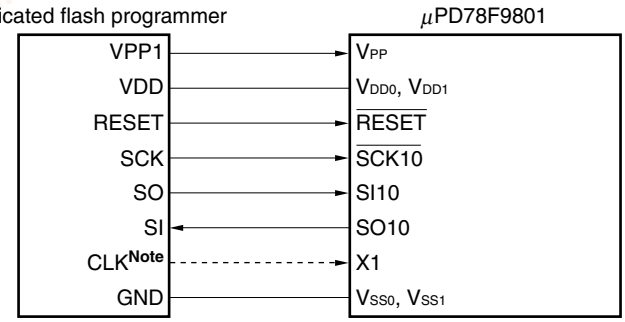
Figure 14-2. Communication Mode Selection Format



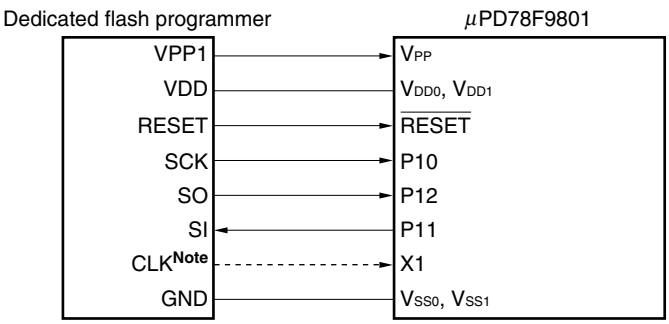
chanhvinh.com

Figure 14-3. Example of Connection with Dedicated Flash Programmer

(a) 3-wire serial I/O



(b) Pseudo-3-wire



Note Connect this pin when the system clock is supplied from the dedicated flash programmer. If a resonator is already connected to the X1 pin, the CLK pin does not need to be connected.

Caution The V_{DD} pin, if already connected to the power supply, must be connected to the VDD pin of the dedicated flash programmer. When using the power supply connected to the V_{DD} pin, supply voltage before starting programming.

chanhvinh.com

If Flashpro III (part no. FL-PR3, PG-FP3)/Flashpro IV is used as a dedicated flash programmer, the following signals are generated for the μPD78F9801. For details, refer to the manual of Flashpro III/Flashpro IV.

Table 14-3. Pin Connection List

Signal Name	I/O	Pin Function	Pin Name	3-Wire Serial I/O	Pseudo-3-Wire
VPP1	Output	Write voltage	V_{PP}	◎	◎
VPP2	–	–	–	×	×
VDD	I/O	V_{DD} voltage generation/voltage monitoring	V_{DD0}, V_{DD1}	◎Note	◎Note
GND	–	Ground	V_{SS0}, V_{SS1}	◎	◎
CLK	Output	Clock output	X1	○	○
RESET	Output	Reset signal	$\overline{\text{RESET}}$	◎	◎
SI	Input	Receive signal	SO10/P11	◎	◎
SO	Output	Transmit signal	SI10/P12	◎	◎
SCK	Output	Transfer clock	$\overline{\text{SCK10}}$/P10	◎	◎
HS	Input	Handshake signal	–	×	×

Note V_{DD} voltage must be supplied before programming is started.

Remark ◎: Pin must be connected.
○: If the signal is supplied on the target board, pin does not need to be connected.
×: Pin does not need to be connected.

chanhvinh.com

14.1.3 On-board pin processing

When performing programming on the target system, provide a connector on the target system to connect the dedicated flash programmer.

An on-board function that allows switching between normal operation mode and flash memory programming mode may be required in some cases.

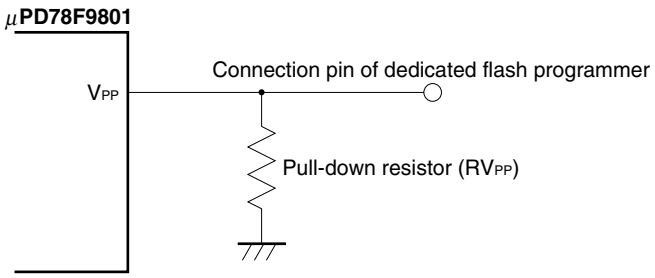
<V_{PP} pin>

In normal operation mode, input 0 V to the V_{PP} pin. In flash memory programming mode, a write voltage of 10.0 V (TYP.) is supplied to the V_{PP} pin, so perform either of the following.

(1) Connect a pull-down resistor (RV_{PP} = 10 kΩ) to the V_{PP} pin.
(2) Use the jumper on the board to switch the V_{PP} pin input to either the writer or directly to GND.

A V_{PP} pin connection example is shown below.

Figure 14-4. V_{PP} Pin Connection Example



<Serial interface pin>

The following shows the pins used by the serial interface.

Serial Interface	Pins Used
3-wire serial I/O	SI10, SO10, $\overline{SCK10}$
Pseudo-3-wire	P12, P11, P10

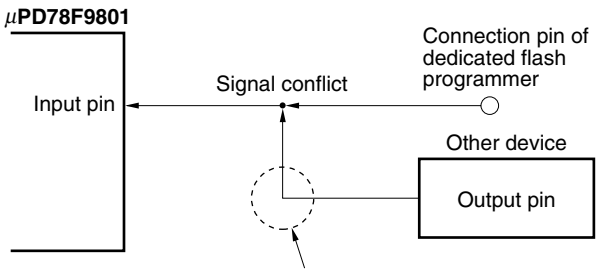
When connecting the dedicated flash programmer to a serial interface pin that is connected to another device on-board, signal conflict or abnormal operation of the other device may occur. Care must therefore be taken with such connections.

chanhvinh.com

(1) Signal conflict

If the dedicated flash programmer (output) is connected to a serial interface pin (input) that is connected to another device (output), a signal conflict occurs. To prevent this, isolate the connection with the other device or set the other device to the output high impedance status.

Figure 14-5. Signal Conflict (Input Pin of Serial Interface)

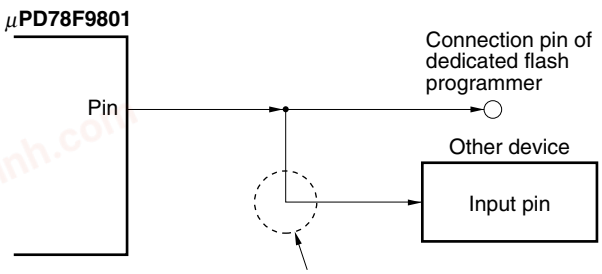


In the flash memory programming mode, the signal output by another device and the signal sent by the dedicated flash programmer conflict; therefore, isolate the signal of the other device.

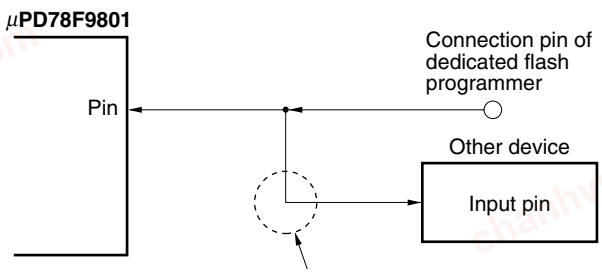
(2) Abnormal operation of other device

If the dedicated flash programmer (output or input) is connected to a serial interface pin (input or output) that is connected to another device (input), a signal is output to the device, and this may cause an abnormal operation. To prevent this abnormal operation, isolate the connection with the other device or set so that the input signals to the other device are ignored.

Figure 14-6. Abnormal Operation of Other Device



If the signal output by the μPD78F9801 affects another device in the flash memory programming mode, isolate the signals of the other device.



If the signal output by the dedicated flash programmer affects another device in the flash memory programming mode, isolate the signals of the other device.

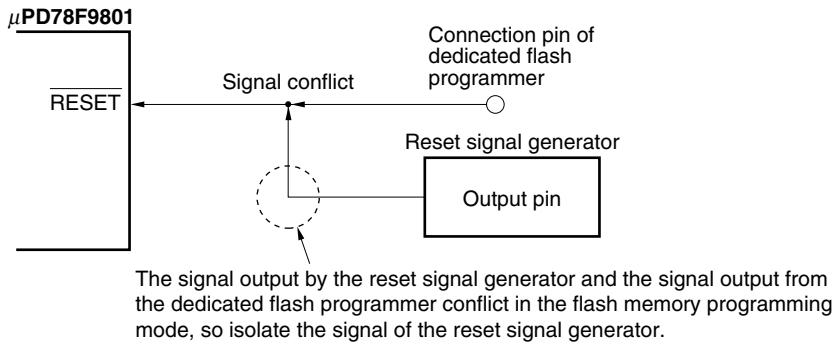
chanhvinh.com

<RESET pin>

If the reset signal of the dedicated flash programmer is connected to the RESET pin connected to the reset signal generator on-board, a signal conflict occurs. To prevent this, isolate the connection with the reset signal generator.

If the reset signal is input from the user system in the flash memory programming mode, a normal programming operation cannot be performed. Therefore, do not input reset signals from other than the dedicated flash programmer.

Figure 14-7. Signal Conflict (RESET Pin)



<Port pins>

When the μPD78F9801 enters the flash memory programming mode, all the pins other than those that communicate with flash programmer are in the same status as immediately after reset.

If the external device does not recognize initial statuses such as the output high impedance status, therefore, connect the external device to V_{DD0}, V_{DD1}, V_{SS0}, or V_{SS1} via a resistor.

<Resonator>

When using the on-board clock, connect X1 and X2 as required in the normal operation mode.

When using the clock output of the flash programmer, connect it directly to X1, disconnecting the main resonator on-board, and leave the X2 pin open.

<Power supply>

To use the power output from the flash programmer, connect the V_{DD0} and V_{DD1} pins to VDD of the flash programmer, and the V_{SS0} and V_{SS1} pins to GND of the flash programmer.

To use the on-board power supply, make connections in accordance with the normal operation mode. However, because the voltage is monitored by the flash programmer, be sure to connect VDD of the flash programmer.

<Other pins>

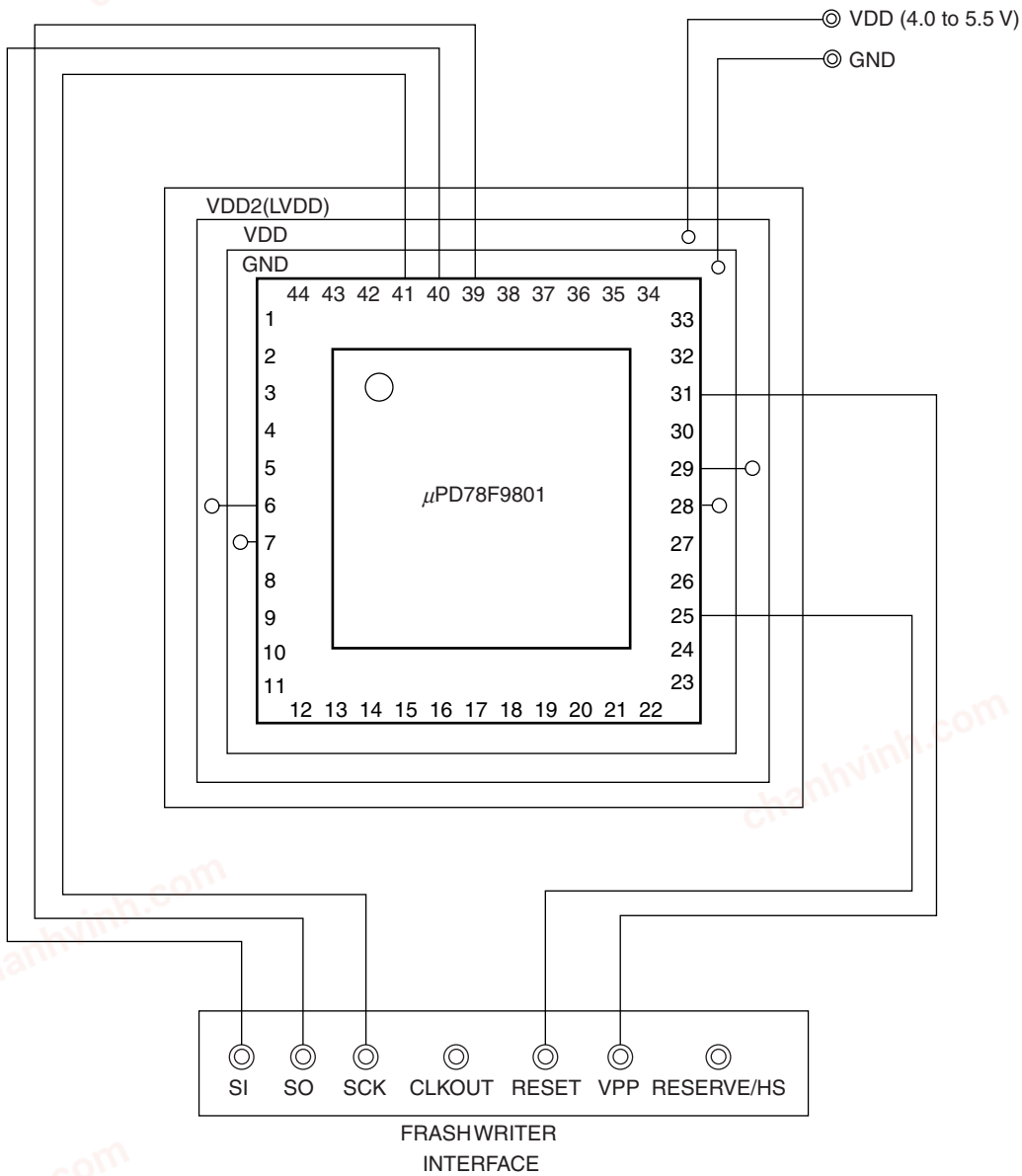
Process the other pins (USBDP, USBDM, REGC) in the same manner as in the normal operation mode.

chanhvinh.com

14.1.4 Connection of adapter for flash writing

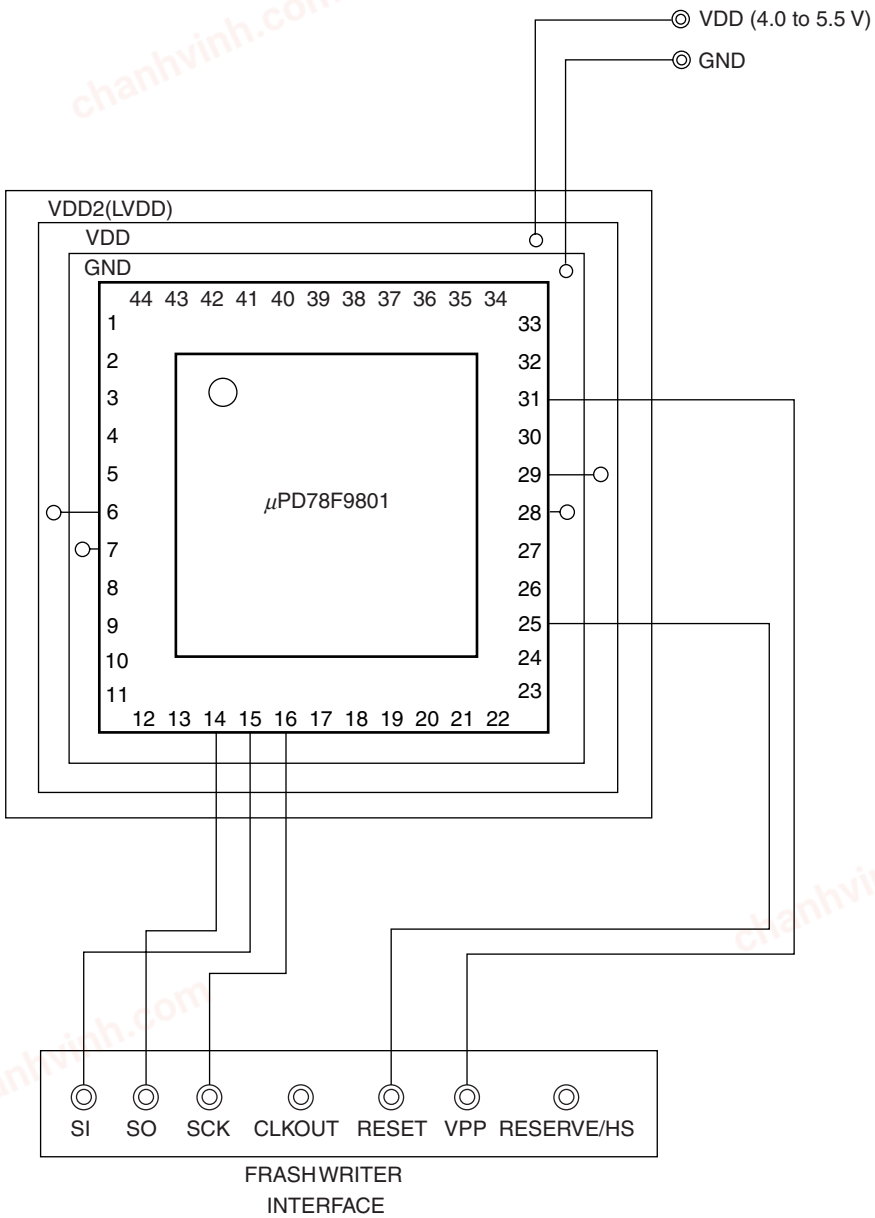
The following figure shows an example of recommended connection when the adapter for flash writing is used.

Figure 14-8. Wiring Example for Flash Writing Adapter with 3-Wire Serial I/O



chanhvinh.com

Figure 14-9. Wiring Example for Flash Writing Adapter with Pseudo-3-Wire Method



chanhvinh.com

CHAPTER 15 INSTRUCTION SET

This chapter lists the instruction set of the μPD789800 Subseries. For details of the operation and machine language (instruction code) of each instruction, refer to **78K/0S Series Instruction User's Manual (U11047E)**.

15.1 Operation

15.1.1 Operand identifiers and description methods

Operands are described in the "Operands" column of each instruction in accordance with the description method of the instruction operand identifier (see the assembler specifications for details). When there are two or more description methods, select one of them. Uppercase letters and the symbols #, !, $, and [ ] are keywords and are described as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: Absolute address specification
- $: Relative address specification
- [ ]: Indirect address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to describe the #, !, $ and [ ] symbols.

For operand register identifiers, r and rp, either function names (X, A, C, etc.) or absolute names (names in parentheses in the table below: R0, R1, R2, etc.) can be used for description.

Table 15-1. Operand Identifiers and Description Methods

Identifier	Description Method
r rp sfr	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7) AX (RP0), BC (RP1), DE (RP2), HL (RP3) Special-function register symbol
saddr saddrp	FE20H to FF1FH Immediate data or label FE20H to FF1FH Immediate data or label (even addresses only)
addr16 addr5	0000H to FFFFH Immediate data or label (only even addresses for 16-bit data transfer instructions) 0040H to 007FH Immediate data or label (even addresses only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label

Remark See **Table 3-2 Special Function Register List** for symbols of special function registers.

chanhvinh.com

15.1.2 Description of “operation” column

A:	A register; 8-bit accumulator
X:	X register
B:	B register
C:	C register
D:	D register
E:	E register
H:	H register
L:	L register
AX:	AX register pair; 16-bit accumulator
BC:	BC register pair
DE:	DE register pair
HL:	HL register pair
PC:	Program counter
SP:	Stack pointer
PSW:	Program status word
CY:	Carry flag
AC:	Auxiliary carry flag
Z:	Zero flag
IE:	Interrupt request enable flag
NMIS:	Flag indicating non-maskable interrupt servicing in progress
( ):	Memory contents indicated by address or register contents in parenthesis
x_H, x_L:	Higher 8 bits and lower 8 bits of 16-bit register
∧:	Logical product (AND)
∨:	Logical sum (OR)
$\veebar$:	Exclusive logical sum (exclusive OR)
¯:	Inverted data
addr16:	16-bit immediate data or label
jdisp8:	Signed 8-bit data (displacement value)

15.1.3 Description of “flag operation” column

(Blank):	Unchanged
0:	Cleared to 0
1:	Set to 1
x:	Set/cleared according to the result
R:	Previously saved value is restored

chanhvinh.com

15.2 Operation List

Mnemonic	Operands	Bytes	Clocks	Operation	Flag
					Z AC CY
MOV	r,#byte	3	6	r ← byte	
	saddr,#byte	3	6	(saddr) ← byte	
	sfr,#byte	3	6	sfr ← byte	
	A,r[Note 1]	2	4	A ← r	
	r,A[Note 1]	2	4	r ← A	
	A,saddr	2	4	A ← (saddr)	
	saddr,A	2	4	(saddr) ← A	
	A,sfr	2	4	A ← sfr	
	sfr,A	2	4	sfr ← A	
	A,!addr16	3	8	A ← (addr16)	
	!addr16,A	3	8	(addr16) ← A	
	PSW,#byte	3	6	PSW ← byte	× × ×
	A,PSW	2	4	A ← PSW	
	PSW,A	2	4	PSW ← A	× × ×
	A,[DE]	1	6	A ← (DE)	
	[DE],A	1	6	(DE) ← A	
	A,[HL]	1	6	A ← (HL)	
	[HL],A	1	6	(HL) ← A	
	A,[HL+byte]	2	6	A ← (HL+byte)	
	[HL+byte],A	2	6	(HL+byte) ← A	
XCH	A,X	1	4	A ↔ X	
	A,r[Note 2]	2	6	A ↔ r	
	A,saddr	2	6	A ↔ (saddr)	
	A,sfr	2	6	A ↔ sfr	
	A,[DE]	1	8	A ↔ (DE)	
	A,[HL]	1	8	A ↔ (HL)	
	A,[HL+byte]	2	8	A ↔ (HL+byte)	

Notes 1. Except r = A.

2. Except r = A, X.

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

chanhvinh.com

Mnemonic	Operands	Bytes	Clocks	Operation	Flag
					Z AC CY
MOVW	rp,#word	3	6	rp ← word	
	AX,saddrp	2	6	AX ← (saddrp)	
	saddrp,AX	2	8	(saddrp) ← AX	
	AX,rp[Note]	1	4	AX ← rp	
	rp,AX[Note]	1	4	rp ← AX	
XCHW	AX,rp[Note]	1	8	AX ↔ rp	
ADD	A,#byte	2	4	A,CY ← A+byte	× × ×
	saddr,#byte	3	6	(saddr),CY ← (saddr) + byte	× × ×
	A,r	2	4	A,CY ← A+r	× × ×
	A,saddr	2	4	A,CY ← A+(saddr)	× × ×
	A,!addr16	3	8	A,CY ← A+(addr16)	× × ×
	A,[HL]	1	6	A,CY ← A+(HL)	× × ×
	A,[HL+byte]	2	6	A,CY ← A+(HL+byte)	× × ×
ADDC	A,#byte	2	4	A,CY ← A+byte+CY	× × ×
	saddr,#byte	3	6	(saddr),CY ← (saddr)+byte+CY	× × ×
	A,r	2	4	A,CY ← A+r+CY	× × ×
	A,saddr	2	4	A,CY ← A+(saddr)+CY	× × ×
	A,!addr16	3	8	A,CY ← A+(addr16)+CY	× × ×
	A,[HL]	1	6	A,CY ← A+(HL)+CY	× × ×
	A,[HL+byte]	2	6	A,CY ← A+(HL+byte)+CY	× × ×
SUB	A,#byte	2	4	A,CY ← A–byte	× × ×
	saddr,#byte	3	6	(saddr), CY ← (saddr)– byte	× × ×
	A,r	2	4	A,CY ← A–r	× × ×
	A,saddr	2	4	A,CY ← A–(saddr)	× × ×
	A,!addr16	3	8	A,CY ← A–(addr16)	× × ×
	A,[HL]	1	6	A,CY ← A–(HL)	× × ×
	A,[HL+byte]	2	6	A,CY ← A–(HL+byte)	× × ×

Note Only when rp = BC, DE, or HL.

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

chanhvinh.com

Mnemonic	Operands	Bytes	Clocks	Operation	Flag
					Z AC CY
SUBC	A,#byte	2	4	A,CY ← A–byte–CY	× × ×
	saddr,#byte	3	6	(saddr),CY ← (saddr)–byte–CY	× × ×
	A,r	2	4	A,CY ← A–r–CY	× × ×
	A,saddr	2	4	A,CY ← A–(saddr)–CY	× × ×
	A,!addr16	3	8	A,CY ← A–(addr16)–CY	× × ×
	A,[HL]	1	6	A,CY ← A–(HL)–CY	× × ×
	A,[HL+byte]	2	6	A,CY ← A–(HL+byte)–CY	× × ×
AND	A,#byte	2	4	A ← A∧byte	×
	saddr,#byte	3	6	(saddr) ← (saddr)∧byte	×
	A,r	2	4	A ← A∧r	×
	A,saddr	2	4	A ← A∧(saddr)	×
	A,!addr16	3	8	A ← A∧(addr16)	×
	A,[HL]	1	6	A ← A∧(HL)	×
	A,[HL+byte]	2	6	A ← A∧(HL+byte)	×
OR	A,#byte	2	4	A ← A∨byte	×
	saddr,#byte	3	6	(saddr) ← (saddr)∨byte	×
	A,r	2	4	A ← A∨r	×
	A,saddr	2	4	A ← A∨(saddr)	×
	A,!addr16	3	8	A ← A∨(addr16)	×
	A,[HL]	1	6	A ← A∨(HL)	×
	A,[HL+byte]	2	6	A ← A∨(HL+byte)	×
XOR	A,#byte	2	4	A ← A∨̄byte	×
	saddr,#byte	3	6	(saddr) ← (saddr)∨̄byte	×
	A,r	2	4	A ← A∨̄r	×
	A,saddr	2	4	A ← A∨̄(saddr)	×
	A,!addr16	3	8	A ← A∨̄(addr16)	×
	A,[HL]	1	6	A ← A∨̄(HL)	×
	A,[HL+byte]	2	6	A ← A∨̄(HL+byte)	×

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

chanhvinh.com

Mnemonic	Operands	Bytes	Clocks	Operation	Flag		
					Z	AC	CY
CMP	A,#byte	2	4	A–byte	×	×	×
	saddr,#byte	3	6	(saddr)–byte	×	×	×
	A,r	2	4	A–r	×	×	×
	A,saddr	2	4	A–(saddr)	×	×	×
	A,!addr16	3	8	A–(addr16)	×	×	×
	A,[HL]	1	6	A–(HL)	×	×	×
	A,[HL+byte]	2	6	A–(HL+byte)	×	×	×
ADDW	AX,#word	3	6	AX,CY ← AX+word	×	×	×
SUBW	AX,#word	3	6	AX,CY ← AX–word	×	×	×
CMPW	AX,#word	3	6	AX–word	×	×	×
INC	r	2	4	r ← r+1	×	×	
	saddr	2	4	(saddr) ← (saddr)+1	×	×	
DEC	r	2	4	r ← r–1	×	×	
	saddr	2	4	(saddr) ← (saddr)–1	×	×	
INCW	rp	1	4	rp ← rp+1			
DECW	rp	1	4	rp ← rp–1			
ROR	A,1	1	2	$(CY, A_7 \leftarrow A_0, A_{m-1} \leftarrow A_m) \times 1$			×
ROL	A,1	1	2	$(CY, A_0 \leftarrow A_7, A_{m+1} \leftarrow A_m) \times 1$			×
RORC	A,1	1	2	$(CY \leftarrow A_0, A_7 \leftarrow CY, A_{m-1} \leftarrow A_m) \times 1$			×
ROLC	A,1	1	2	$(CY \leftarrow A_7, A_0 \leftarrow CY, A_{m+1} \leftarrow A_m) \times 1$			×
SET1	saddr.bit	3	6	(saddr.bit) ← 1			
	sfr.bit	3	6	sfr.bit ← 1			
	A.bit	2	4	A.bit ← 1			
	PSW.bit	3	6	PSW.bit ← 1	×	×	×
	[HL].bit	2	10	(HL).bit ← 1			
CLR1	saddr.bit	3	6	(saddr.bit) ← 0			
	sfr.bit	3	6	sfr.bit ← 0			
	A.bit	2	4	A.bit ← 0			
	PSW.bit	3	6	PSW.bit ← 0	×	×	×
	[HL].bit	2	10	(HL).bit ← 0			
SET1	CY	1	2	CY ← 1			1
CLR1	CY	1	2	CY ← 0			0
NOT1	CY	1	2	CY ← $\overline{CY}$			×

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

chanhvinh.com

Mnemonic	Operands	Bytes	Clocks	Operation	Flag
					Z AC CY
CALL	!addr16	3	6	(SP–1) ← (PC+3)H, (SP–2) ← (PC+3)L, PC ← addr16, SP ← SP–2	
CALLT	[addr5]	1	8	(SP–1) ← (PC+1)H, (SP–2) ← (PC+1)L, PCH ← (00000000, addr5+1), PCL ← (00000000, addr5), SP ← SP–2	
RET		1	6	PCH ← (SP+1), PCL ← (SP), SP ← SP+2	
RETI		1	8	PCH ← (SP+1), PCL ← (SP), PSW ← (SP+2), SP ← SP+3, NMIS ← 0	R R R
PUSH	PSW	1	2	(SP–1) ← PSW, SP ← SP–1	
	rp	1	4	(SP–1) ← rpH, (SP–2) ← rpL, SP ← SP–2	
POP	PSW	1	4	PSW ← (SP), SP ← SP+1	R R R
	rp	1	6	rpH ← (SP+1), rpL ← (SP), SP ← SP+2	
MOVW	SP, AX	2	8	SP ← AX	
	AX, SP	2	6	AX ← SP	
BR	!addr16	3	6	PC ← addr16	
	$addr16	2	6	PC ← PC+2+jdisp8	
	AX	1	6	PCH ← A, PCL ← X	
BC	$saddr16	2	6	PC ← PC+2+jdisp8 if CY=1	
BNC	$saddr16	2	6	PC ← PC+2+jdisp8 if CY=0	
BZ	$saddr16	2	6	PC ← PC+2+jdisp8 if Z=1	
BNZ	$saddr16	2	6	PC ← PC+2+jdisp8 if Z=0	
BT	saddr.bit,$addr16	4	10	PC ← PC+4+jdisp8 if (saddr.bit)=1	
	sfr.bit,$addr16	4	10	PC ← PC+4+jdisp8 if sfr.bit=1	
	A.bit,$addr16	3	8	PC ← PC+3+jdisp8 if A.bit=1	
	PSW.bit,$addr16	4	10	PC ← PC+4+jdisp8 if PSW.bit=1	
BF	saddr.bit,$addr16	4	10	PC ← PC+4+jdisp8 if (saddr.bit)=0	
	sfr.bit,$addr16	4	10	PC ← PC+4+jdisp8 if sfr.bit=0	
	A.bit,$addr16	3	8	PC ← PC+3+jdisp8 if A.bit=0	
	PSW.bit,$addr16	4	10	PC ← PC+4+jdisp8 if PSW.bit=0	
DBNZ	B,$addr16	2	6	B ← B–1, then PC ← PC+2+jdisp8 if B≠0	
	C,$addr16	2	6	C ← C–1, then PC ← PC+2+jdisp8 if C≠0	
	saddr,$addr16	3	8	(saddr) ← (saddr)–1, then PC ← PC+3+jdisp8 if (saddr)≠0	
NOP		1	2	No Operation	
EI		3	6	IE ← 1 (Enable interrupt)	
DI		3	6	IE ← 0 (Disable interrupt)	
HALT		1	2	Set HALT mode	
STOP		1	2	Set STOP mode	

Remark One instruction clock cycle is one CPU clock cycle (fCPU) selected by the processor clock control register (PCC).

chanhvinh.com

15.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, INC, DEC, ROR, ROL, RORC, ROLC, PUSH, POP, DBNZ

1st Operand \ 2nd Operand	#byte	A	r	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL+byte]	$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV[Note] XCH[Note] ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											
[HL+byte]		MOV											

Note Except r = A.

chanhvinh.com

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

1st Operand \ 2nd Operand	#word	AX	rp[Note]	saddrp	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	
rp	MOVW	MOVW[Note]				INCW DECW PUSH POP
saddrp		MOVW				
SP		MOVW				

Note Only when rp = BC, DE, or HL .

(3) Bit manipulation instructions

SET1, CLR1, NOT1, BT, BF

1st Operand \ 2nd Operand	$addr16	None
A.bit	BT BF	SET1 CLR1
sfr.bit	BT BF	SET1 CLR1
saddr.bit	BT BF	SET1 CLR1
PSW.bit	BT BF	SET1 CLR1
[HL].bit		SET1 CLR1
CY		SET1 CLR1 NOT1

chanhvinh.com

(4) Call instructions/branch instructions

CALL, CALLT, BR, BC, BNC, BZ, BNZ, DBNZ

2nd Operand / 1st Operand	AX	!addr16	[addr5]	$addr16
Basic Instructions	BR	CALL BR	CALLT	BR BC BNC BZ BNZ
Compound Instructions				DBNZ

(5) Other instructions

RET, RETI, NOP, EI, DI, HALT, STOP

chanhvinh.com

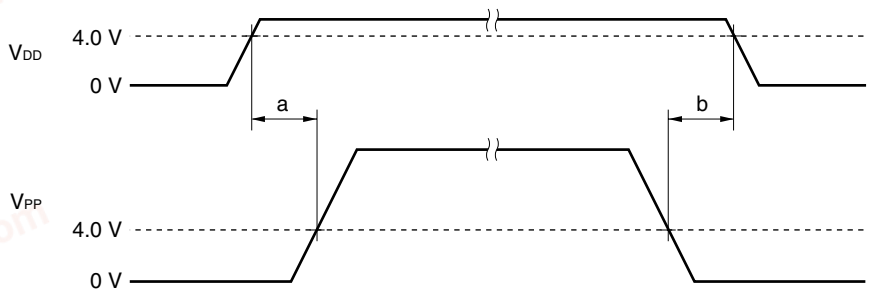
★ # CHAPTER 16 ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings (T_A = 25°C)

Parameter	Symbol	Conditions	Rating	Unit
Supply voltage	V_{DD}		–0.3 to +6.5	V
	V_{PP}	μPD78F9801 only[Note 1]	–0.3 to +10.5	V
Input voltage	V_I		–0.3 to V_{DD} + 0.3[Note 2]	V
Output voltage	V_O		–0.3 to V_{DD} + 0.3[Note 2]	V
Output current, high	I_{OH}	Per pin	–10	mA
		Total for all pins	–30	mA
Output current, low	I_{OL}	Per pin	30	mA
		Total for all pins	160	mA
Operating ambient temperature	T_A	In normal operation mode	–40 to +85	°C
		During flash memory programming	10 to 40	°C
Storage temperature	T_{stg}	Mask ROM version	–65 to +150	°C
		μPD78F9801	–40 to +125	°C

Notes 1. Make sure that the following conditions of the V_{PP} voltage application timing are satisfied when the flash memory is written.

- When supply voltage rises
  V_{PP} must exceed V_{DD} 10 μs or more after V_{DD} has reached the lower-limit value (4.0 V) of the operating voltage range (see a in the figure below).
- When supply voltage drops
  V_{DD} must be lowered 10 μs or more after V_{PP} falls below the lower-limit value (4.0 V) of the operating voltage range of V_{DD} (see b in the figure below).



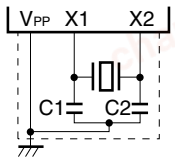
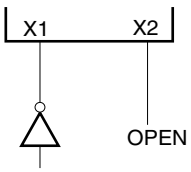
2. Must be 6.5 V or lower

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

chanhvinh.com

System Clock Oscillation Circuit Characteristics (T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Crystal		Oscillator frequency (f_X)Note 1		6.0	6.0	6.0	MHz
		Oscillation stabilization timeNote 2				10	ms
External clock		X1 input frequency (f_X)Note 1		6.0	6.0	6.0	MHz
		X1 input high-low-level width (t_{XH}, t_{XL})		71		83	ns

Notes 1. Indicates only oscillator characteristics. Refer to **AC Characteristics** for the instruction execution time.

2. Time required to oscillation after reset or STOP mode release. Use a resonator that can stabilize oscillation before the oscillation stabilization time elapses.

Caution When using the system clock oscillator, wire as follows in the area enclosed by the broken lines in the above figure to avoid an adverse effect from wiring capacitance.

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with other signal lines.**
- **Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS0}.**
- **Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

★ **Remark** For the resonator selection and oscillator constant, customers are required to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

Flash Memory Write/Erase Characteristics (T_A = 10 to 40°C, V_{DD} = 4.0 to 5.5 V) (μPD78F9801 only)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Write current (V_{DD} pin)	I_{DDW}	When V_{PP} supply voltage = V_{PP1} (in 6.0 MHz operation mode)			18Note	mA
Write current (V_{PP} pin)	I_{PPW}	When V_{PP} supply voltage = V_{PP1}			7.5	mA
Erase current (V_{DD} pin)	I_{DDE}	When V_{PP} supply voltage = V_{PP1} (in 6.0 MHz operation mode)			18Note	mA
Erase current (V_{PP} pin)	I_{PPE}	When V_{PP} supply voltage = V_{PP1}			100	mA
Unit erase time	t_{er}		1	1	1	s
Total erase time	t_{era}				20	s
Write count		Erase/write are regarded as 1 cycle.			1	Times
V_{PP} supply voltage	V_{PP0}	In normal operation	0		0.2V_{DD}	V
	V_{PP1}	During flash memory programming	9.7	10.0	10.3	V

Note The current flowing to the ports (including the current flowing through the on-chip pull-up resistors) is not included.

chanhvinh.com

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V) (1/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high	I_{OH}	Per pin				–1	mA
		Total for all pins				–15	mA
Output current, low	I_{OL}	Per pin				10	mA
		Total for all pins				80	mA
Input voltage, high	V_{IH1}	P00 to P07, P10 to P17		0.7V_{DD}		V_{DD}	V
	V_{IH2}	$\overline{\text{RESET}}$, P20 to P26, P40 to P47		0.8V_{DD}		V_{DD}	V
	V_{IH3}	X1		V_{DD} – 0.1		V_{DD}	V
	V_{IH4}	USBDM, USBDP T_A = 0 to +70°C		2.0		3.6	V
Input voltage, low	V_{IL1}	P00 to P07, P10 to P17		0		0.3V_{DD}	V
★	V_{IL2}	$\overline{\text{RESET}}$, P20 to P26, P40 to P47		0		0.2V_{DD}	V
	V_{IL3}	X1		0		0.1	V
	V_{IL4}	USBDM, USBDP T_A = 0 to +70°C		0		0.8	V
Output voltage, high	V_{OH1}	Pins other than USBDM and USBDP	I_O = –1 mA	V_{DD} – 1.0			V
	V_{OH2}	USBDM, USBDP T_A = 0 to +70°C, RL = 15 kΩ (connected to V_{SS})[Note]		2.8			V
Output voltage, low	V_{OL1}	Pins other than USBDM and USBDP	I_O = 10 mA			1.0	V
★	V_{OL2}	USBDM, USBDP T_A = 0 to +70°C, RL = 1.5 kΩ (connected to V_{DD})[Note]				0.3	V
Input leakage current, high	I_{LIH1}	Pins other than X1, X2, USBDM, and USBDP	$V_I = V_{DD}$			3	μA
	I_{LIH2}	X1, X2	$V_I = V_{DD}$			20	μA
	I_{LIH3}	USBDM, USBDP T_A = 0 to +70°C	0 V ≤ V_I ≤ V_{REG}			10	μA
Input leakage current, low	I_{LIL1}	Pins other than X1, X2, USBDM, and USBDP	V_I = 0 V			–3	μA
	I_{LIL2}	X1, X2	V_I = 0 V			–20	μA
	I_{LIL3}	USBDM, USBDP T_A = 0 to +70°C	0 V ≤ V_I ≤ V_{REG}			–10	μA
Output leakage current, high	I_{LOH}	V_O = 0 V				3	μA
Output leakage current, low	I_{LOL}	V_O = 0 V				–3	μA
Software pull-up resistance	R	V_I = 0 V		50	100	200	kΩ
Regulator output voltage	V_{REG}	I_O = 0 to –3 mA		3.0	3.3	3.6	V

Note RL is the resistance connected to the bus line.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

chanhvinh.com

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V) (2/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Supply current[Note 1] (Mask ROM Version)	I_{DD1}	6.0 MHz crystal oscillation (operating mode)[Note 2]			1.5	3.0	mA
	I_{DD2}	6.0 MHz crystal oscillation (HALT mode)[Note 2]			0.5	1.1	mA
	I_{DD3}	STOP mode	When the USB function is disabled		10	30	μA
			When the USB function is enabled (T_A = 0 to +70°C)		50	100	μA
Supply current[Note 1] (μPD78F9801)	I_{DD1}	6.0 MHz crystal oscillation (operating mode)[Note 2]			5.0	10.0	mA
	I_{DD2}	6.0 MHz crystal oscillation (HALT mode)[Note 2]			1.5	3.5	mA
	I_{DD3}	STOP mode	When the USB function is disabled		10	30	μA
			When the USB function is enabled (T_A = 0 to +70°C)		50	100	μA

Notes 1. The power supply current does not include the current flowing through the on-chip pull-up resistors.

2. During high-speed mode operation (when the processor clock control register (PCC) is cleared to 00H)

chanhvinh.com

AC Characteristics

(1) Basic operations (T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Cycle time (minimum instruction execution time)	T_{CY}	When PCC = 00H (f_X = 6.0 MHz)	0.333	0.333	0.333	μs
		When PCC = 02H (f_X = 6.0 MHz)	1.333	1.333	1.333	μs
TI01 input frequency	f_{TI}		0		4.0	MHz
TI01 input high-/low-level width	t_{TIH}, t_{TIL}		0.1			μs
Interrupt input high-/low-level width	t_{INTH}, t_{INTL}	INTP0	10			μs
$\overline{\text{RESET}}$ input low-level width	t_{RSL}		10			μs

(2) Serial interface

(a) USB function (T_A = 0 to +70°C, V_{DD} = 4.0 to 5.5 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
USBDM and USBDP rise time	t_R	CL = 50 pF[Note]	75			ns
		CL = 350 pF[Note]			300	ns
USBDM and USBDP fall time	t_F	CL = 50 pF[Note]	75			ns
		CL = 350 pF[Note]			300	ns
t_R and t_F matching	t_{RFM}	t_R/t_F	80		120	%
Differential output signal cross-over point	V_{CRS}		1.3		2.0	V
Data transfer rate	t_{DRATE}	When the microcontroller operates at the system clock (f_X) of 6.0 MHz	1.5	1.5	1.5	Mbps
Transmission differential signal jitter	t_{UDJ1}	Upon transferring the next bit	–95	0	95	ns
	t_{UDJ2}	Upon transferring the bit following the next bit	–150	0	150	ns
Transmission EOP width	t_{EOPT1}		1.25	1.33	1.50	μs
Reception EOP width	t_{EOPR1}	EOP width to be eliminated			300	μs
	t_{EOPR2}	EOP width to be detected	675			μs
Reception USB reset width	t_{URES1}	USB reset width to be eliminated			2.5	μs
	t_{URES2}	USB reset width to be detected	5.5			μs

Note CL is the capacitance of the USBDM and USBDP output lines.

chanhvinh.com

(b) 3-wire serial I/O mode (T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V)

(i) $\overline{SCK10}$ ...Internal clock output (when f_X = 6.0 MHz)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCK10}$ cycle time	t_{KCY1}	When TPS100[Note 1] = 0		667	667	667	ns
		When TPS100[Note 1] = 1		1,333	1,333	1,333	ns
$\overline{SCK10}$ high-/low-level width	t_{KH1}, t_{KL1}	When TPS100[Note 1] = 0		283	333		ns
		When TPS100[Note 1] = 1		617	667		ns
SI10 setup time	t_{SIK1}	To $\overline{SCK10}$ ↑		150			ns
SI10 hold time	t_{KSI1}	From $\overline{SCK10}$ ↑	When TPS100[Note 1] = 0	333			ns
			When TPS100[Note 1] = 1	667			ns
SO10 output delay	t_{KSO1}	From $\overline{SCK10}$ ↓, CL = 100 pF[Note 2]		0		200	ns

Notes 1. Bit 4 of serial operation mode register 10 (CSIM10)

2. CL is the capacitance of the SO output line.

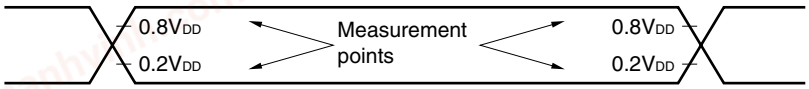
(ii) $\overline{SCK10}$ ...External clock output

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCK10}$ cycle time	t_{KCY2}		667			ns
$\overline{SCK10}$ high-/low-level width	t_{KH2}, t_{KL2}		283			ns
SI10 setup time	t_{SIK2}		100			ns
SI10 hold time	t_{KSI2}		333			ns
SO10 output delay	t_{KSO2}	From $\overline{SCK10}$ ↓, CL = 100 pF[Note]	0		250	ns

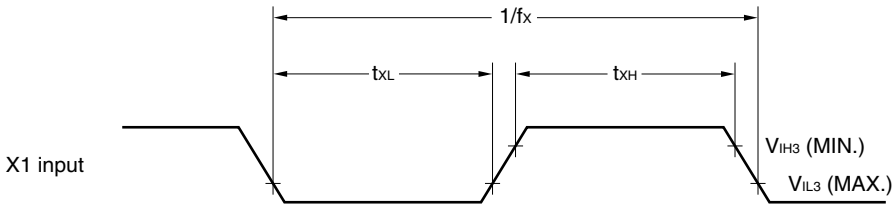
Note CL is the capacitance of the SO output line.

chanhvinh.com

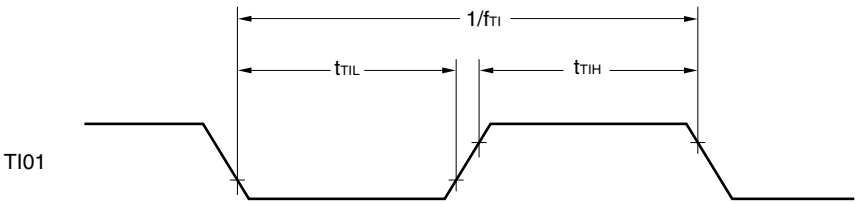
AC Timing Measurement Points (Except X1 Input and USB Function)



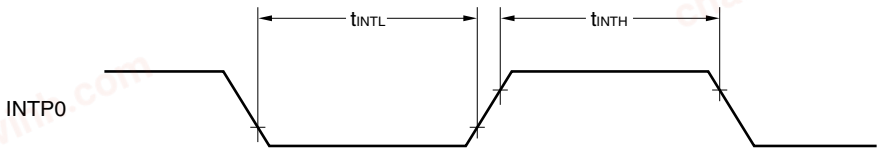
Clock timing



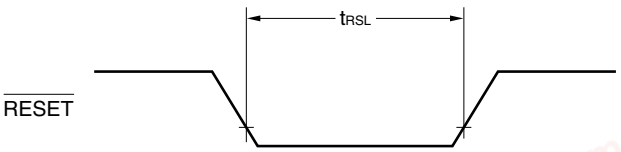
TI Timing



Interrupt Input Timing



$\overline{\text{RESET}}$ Input Timing

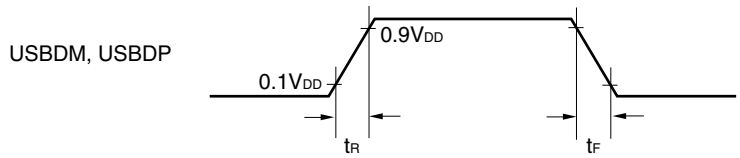


chanhvinh.com

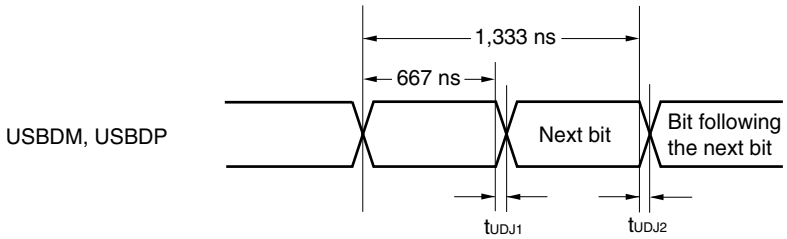
Serial Transfer Timing

USB function:

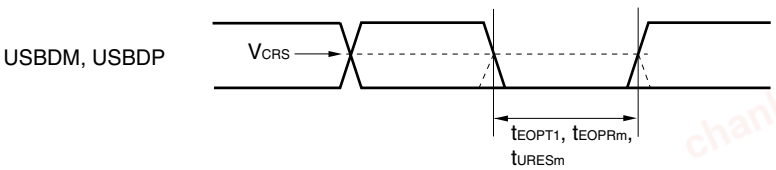
USBDM and USBDP rise/fall time



Transmission differential signal jitter

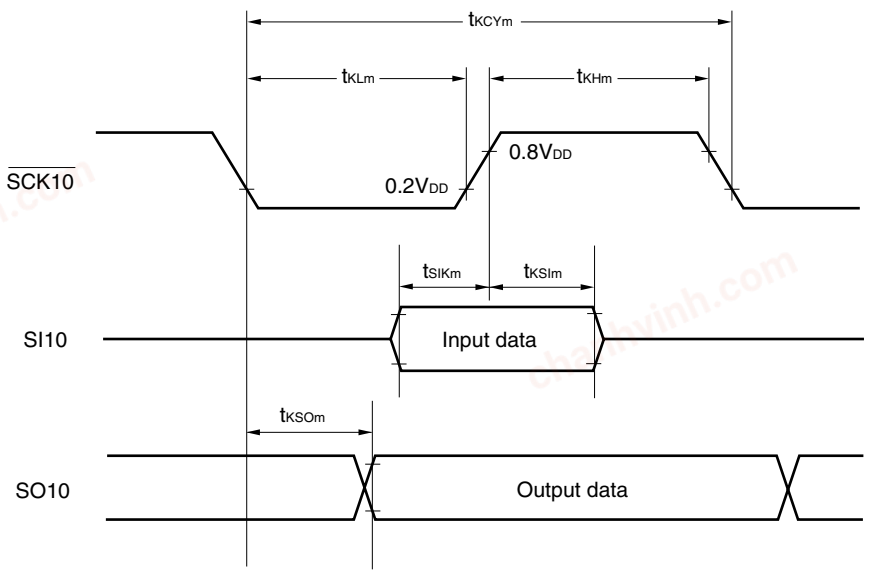


Differential output signal cross-over point, transmission EOP width, reception EOP width, and reception USB reset width



m = 1, 2

3-wire serial I/O mode:



m = 1, 2

chanhvinh.com

Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics (T_A = –40 to +85°C)

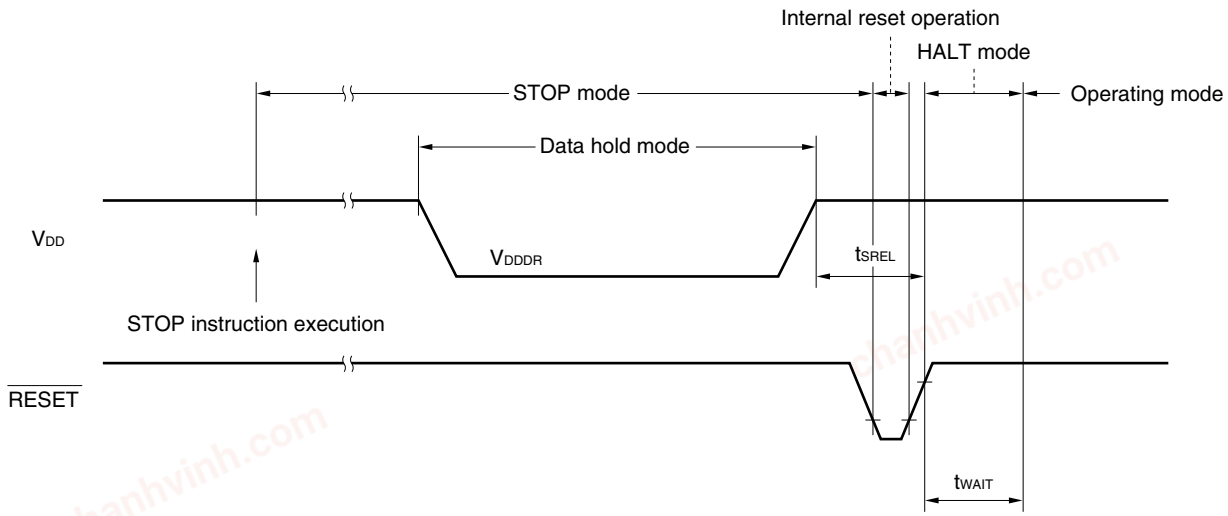
Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data hold supply voltage	V_{DDDR}		4.0		5.5	V
Release signal set time	t_{SREL}		0			μs
Oscillation stabilization time Note 1	t_{WAIT}	Release by $\overline{RESET}$		$2^{15}/f_X$		ms
		Release by interrupt request		**Note 2**		ms

Notes 1. During the oscillation stabilization time, CPU operations are disabled to prevent them from becoming unstable upon the start of oscillation.

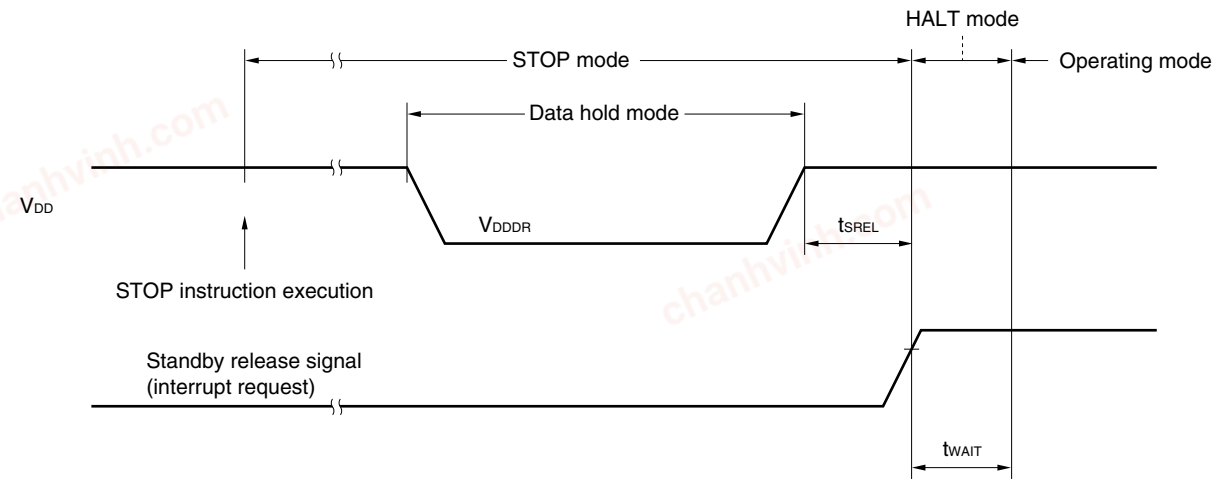
2. $2^{12}/f_X$, $2^{15}/f_X$, or $2^{17}/f_X$ can be selected according to the setting of bits 0 to 2 (OSTS0 to OSTS2) of the oscillation stabilization time selection register.

Remark f_X: System clock oscillation frequency

Data Hold timing (STOP Mode Release by $\overline{RESET}$)

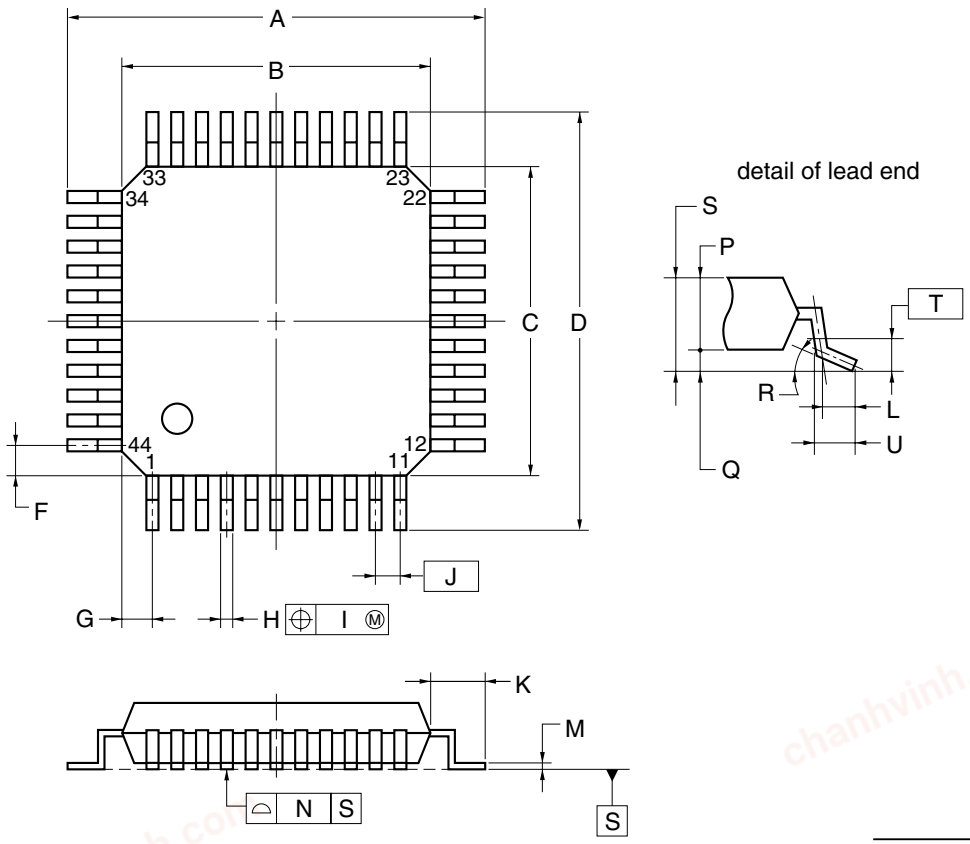


Data Hold Timing (Standby Release Signal: STOP Mode Release by Interrupt Signal)



chanhvinh.com

44 PIN PLASTIC LQFP (10x10)



NOTE

Each lead centerline is located within 0.20 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	12.0±0.2
B	10.0±0.2
C	10.0±0.2
D	12.0±0.2
F	1.0
G	1.0
H	$0.37^{+0.08}_{-0.07}$
I	0.20
J	0.8 (T.P.)
K	1.0±0.2
L	0.5
M	$0.17^{+0.03}_{-0.06}$
N	0.10
P	1.4±0.05
Q	0.1±0.05
R	$3^{\circ}{}^{+4^{\circ}}_{-3^{\circ}}$
S	1.6 MAX.
T	0.25 (T.P.)
U	0.6±0.15

S44GB-80-8ES-2

chanhvinh.com

★ CHAPTER 18 RECOMMENDED SOLDERING CONDITIONS

The μPD789800 Subseries should be soldered and mounted under the following recommended conditions.

For details of the recommended soldering conditions, refer to the document **Semiconductor Device Mounting Technology Manual (C10535E)**.

For soldering methods and conditions other than those recommended below, contact an NEC Electronics sales representative.

Table 18-1. Surface Mounting Type Soldering Conditions

(1) μPD789800GB-×××-8ES: 44-pin plastic LQFP (10x10)
μPD78F9801GB-8ES: 44-pin plastic LQFP (10x10)

Soldering Method	Soldering Conditions	Symbol
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: Twice or less	IR35-00-2
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (at 200 °C or higher), Count: Twice or less	VP15-00-2
Wave soldering	Solder bath temperature: 260°C max., Time: 10 seconds max., Count: Once, Preheating temperature: 120°C max. (package surface temperature)	WS60-00-1
Partial heating method	Pin temperature: 350°C max. Time: 3 seconds max. (per pin row)	–

Caution Do not use different soldering methods together (except for partial heating).

★ (2) μPD789800GB-×××-8ES-A: 44-pin plastic LQFP (10x10)
μPD78F9801GB-8ES-A: 44-pin plastic LQFP (10x10)

Soldering Method	Soldering Conditions	Symbol
Infrared reflow	Package peak temperature: 260°C, Time: 60 seconds max. (at 220°C or higher), Count: Three times or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 20 to 72 hours)	IR60-207-3
Wave soldering	When the pin pitch of the package is 0.65 mm or more, wave soldering can also be performed. For details, contact an NEC Electronics sales representative.	–
Partial heating method	Pin temperature: 350°C max. Time: 3 seconds max. (per pin row)	–

Note After opening the dry pack, store it at 25°C or less and 65% RH or less for the allowable storage period.

Caution Do not use different soldering methods together (except for partial heating).

Remark Products that have the part numbers suffixed by "-A" are lead-free products.

chanhvinh.com

★ APPENDIX A DEVELOPMENT TOOLS

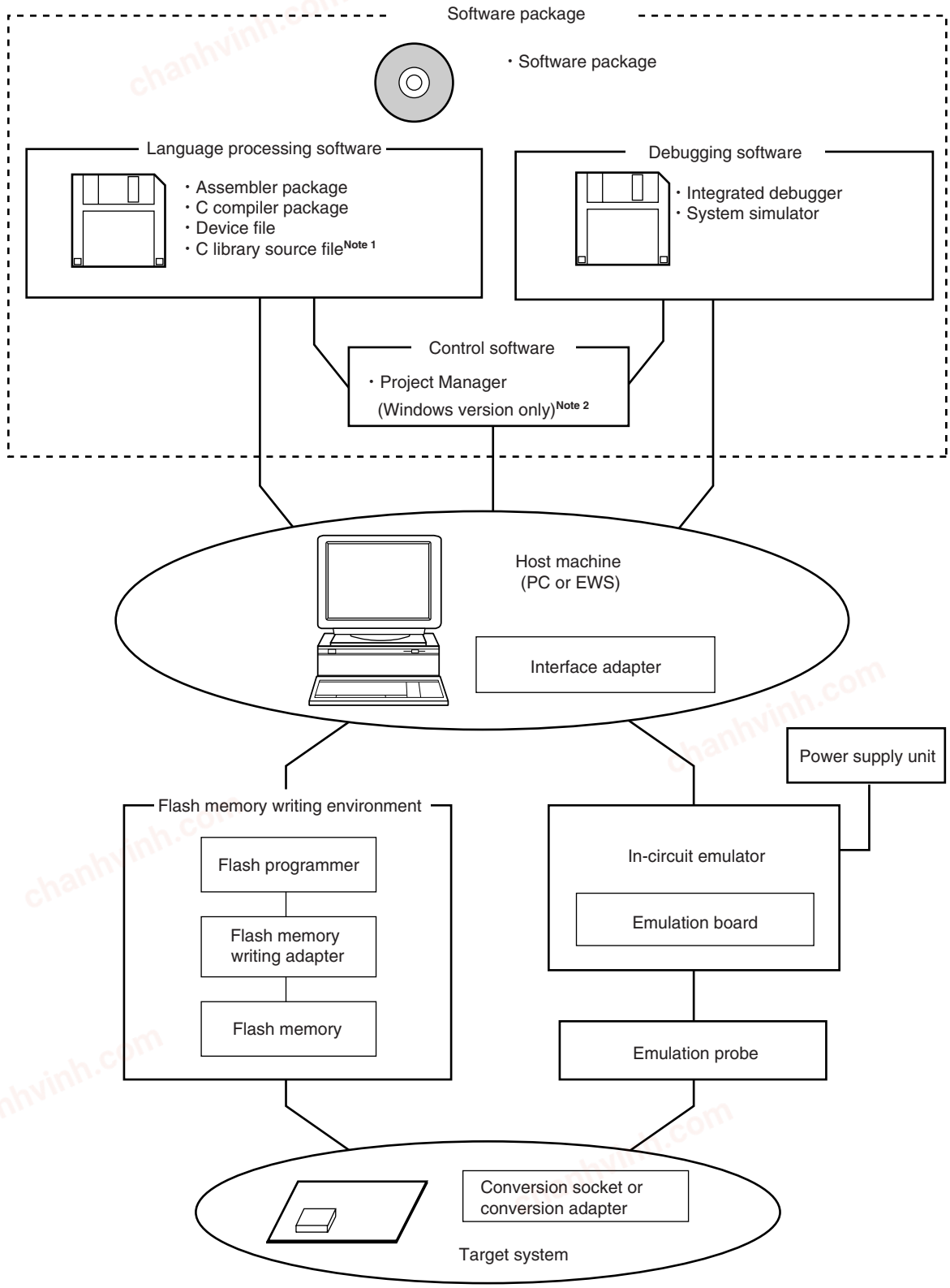
The following development tools are available for development of systems using the μPD789800 Subseries.
Figure A-1 shows the development tools.

- Support of the PC98-NX series
  Unless otherwise stated, the μPD789800 Subseries, which is supported by IBM PC/AT™ and compatibles, can be used for the PC98-NX series. When using the PC98-NX series, refer to the descriptions of IBM PC/AT and compatibles.

- Windows
  Unless otherwise stated, “Windows” indicates the following OSs.
  - Windows 3.1
  - Windows 95, 98, 2000
  - Windows NT™ Ver.4.0

chanhvinh.com

Figure A-1. Development Tools



Notes 1. C library source file is not included in the software package.

2. Project Manager is included in the assembler package.
Project Manager is used only in the Windows environment.

chanhvinh.com

A.1 Software Package

SP78K0S Software package	Software tools for development of the 78K/0S Series are combined in this package. The following tools are included. RA78K0S, CC78K0S, ID78K0S-NS, SM78K0S, and device files
	Part number: μS××××SP78K0S

Remark ×××× in the part number differs depending on the OS used.

μS××××SP78K0S

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Japanese Windows	CD-ROM
BB17		English Windows	

A.2 Language Processing Software

RA78K0S Assembler package	Program that converts a program written in mnemonic into object codes that can be executed by a microcontroller. In addition, automatic functions to generate a symbol table and optimize branch instructions are also provided. Used in combination with a device file (DF789801) (sold separately). **<Caution when used in PC environment>** The assembler package is a DOS-based application but may be used in the Windows environment by using the Project Manager of Windows (included in the assembler package).
	Part number: μS××××RA78K0S
CC78K0S C compiler package	Program that converts a program written in C language into object codes that can be executed by a microcontroller. Used in combination with an assembler package (RA78K0S) and device file (DF789801) (both sold separately). **<Caution when used in PC environment>** The C compiler package is a DOS-based application but may be used in the Windows environment by using the Project Manager of Windows (included in the assembler package).
	Part number: μS××××CC78K0S
DF789801[Note 1] Device file	File containing information inherent to the device. Used in combination with the RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S (all sold separately).
	Part number: μS××××DF789801
CC78K0S-L[Note 2] C library source file	Source file of functions for generating the object library included in the C compiler package. Necessary for changing the object library included in the C compiler package according to the customer's specifications. Since this is a source file, its working environment does not depend on any particular operating system.
	Part number: μS××××CC78K0S-L

Notes 1. DF789801 is a common file that can be used with RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S.

2. CC78K0S-L is not included in the software package (SP78K0S).

chanhvinh.com

Remark ××××in the part number differs depending on the host machine and operating system to be used.

μS××××RA78K0S
μS××××CC78K0S

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Japanese Windows	3.5" 2HD FD
BB13		English Windows	
AB17		Japanese Windows	CD-ROM
BB17		English Windows	
3P17	HP9000 series 700™	HP-UX™ (Rel. 10.10)	
3K17	SPARCstation™	SunOS™ (Rel. 4.1.4), Solaris™ (Rel. 2.5.1)	

μS××××DF789801
μS××××CC78K0S-L

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Japanese Windows	3.5" 2HD FD
BB13		English Windows	
3P16	HP9000 series 700	HP-UX (Rel. 10.10)	DAT
3K13	SPARCstation	SunOS (Rel. 4.1.4), Solaris (Rel. 2.5.1)	3.5" 2HD FD
3K15			1/4-inch CGMT

A.3 Control Software

Project Manager	Control software created for efficient development of the user program in the Windows environment. User program development operations such as editor startup, build, and debugger startup can be performed from the Project Manager. <Caution> The Project Manager is included in the assembler package (RA78K0S). The Project Manager is used only in the Windows environment.

A.4 Flash Memory Writing Tools

Flashpro III (FL-PR3, PG-FP3) Flashpro IV (FL-PR4, PG-FP4) Flash programmer	Dedicated flash programmer for microcontrollers incorporating flash memory
FA-44GB-8ES Flash memory writing adapter	Adapter for writing to flash memory and connected to Flashpro III or Flashpro IV. FA-44GB-8ES: For 44-pin plastic LQFP (GB-8ES type)

Remark The FL-PR3, FL-PR4 and FA-44GB-8ES are products made by Naito Densei Machida Mfg. Co., Ltd. (TEL +81-45-475-4191).

chanhvinh.com

A.5 Debugging Tools (Hardware)

IE-78K0S-NS In-circuit emulator		In-circuit emulator for debugging hardware and software of an application system using the 78K/0S Series. Supports a integrated debugger (ID78K0S-NS). Used in combination with an AC adapter, emulation probe, and interface adapter for connecting the host machine.
IE-78K0S-NS-A In-circuit emulator		The IE-78K0S-NS-A provides a coverage function in addition to the IE-78K0S-NS functions, thus enhancing the debug functions, including the tracer and timer functions.
IE-70000-MC-PS-B AC adapter		Adapter for supplying power from an AC 100 to 240 V outlet.
IE-70000-98-IF-C Interface adapter		Adapter necessary when using PC-9800 series PC (except notebook type) as host machine (C bus supported)
IE-70000-CD-IF-A PC card interface		PC card and interface cable necessary when using notebook PC as host machine (PCMCIA socket supported)
IE-70000-PC-IF-C Interface adapter		Interface adapter necessary when using IBM PC/AT compatible as host machine (ISA bus supported)
IE-70000-PCI-IF-A Interface adapter		Adapter necessary when using personal computer incorporating PCI bus as host machine
IE-789801-NS-EM1 Emulation board		Board for emulating the peripheral hardware inherent to the device. Used in combination with an in-circuit emulator.
NP-44GB-TQ NP-H44GB-TQ Emulation probe		Cable to connect the in-circuit emulator and target system. Used in combination with the TGB-044SAP.
	TGB-044SAP Conversion adapter	Conversion adapter to connect the NP-44GB-TQ or NP-H44GB-TQ and a target system board on which a 44-pin plastic LQFP (GB-8ES type) can be mounted

Remarks 1. The NP-44GB-TQ and NP-H44GB-TQ are products made by Naito Densei Machida Mfg. Co., Ltd. (TEL +81-45-475-4191).

2. The TGB-044SAP is a product made by TOKYO ELETECH CORPORATION.
For further information, contact: Daimaru Kogyo, Ltd.
Tokyo Electronics Department (TEL +81-3-3820-7112)
Osaka Electronics Department (TEL +81-6-6244-6672)

chanhvinh.com

A.6 Debugging Tools (Software)

ID78K0S-NS Integrated debugger	This debugger supports the in-circuit emulators IE-78K0S-NS and IE-78K0S-NS-A for the 78K/0S Series. The ID78K0S-NS is Windows-based software. It has improved C-compatible debugging functions and can display the results of tracing with the source program using an integrating window function that associates the source program, disassemble display, and memory display with the trace result. Used in combination with a device file (DF789801) (sold separately).
	Part number: μS××××ID78K0S-NS
SM78K0S System simulator	This is a system simulator for the 78K/0S Series. The SM78K0S is Windows-based software. It can be used to debug the target system at C source level or assembler level while simulating the operation of the target system on the host machine. Using SM78K0S, the logic and performance of the application can be verified independently of hardware development. Therefore, the development efficiency can be enhanced and the software quality can be improved. Used in combination with a device file (DF789801) (sold separately).
	Part number: μS××××SM78K0S
DF789801[Note] Device file	File containing information inherent to the device. Used in combination with the RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S (sold separately).
	Part number: μS××××DF789801

Note DF789801 is a common file that can be used with the RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S.

Remark ×××× in the part number differs depending on the operating system and supply medium to be used.

μS××××ID78K0S-NS
μS××××SM78K0S

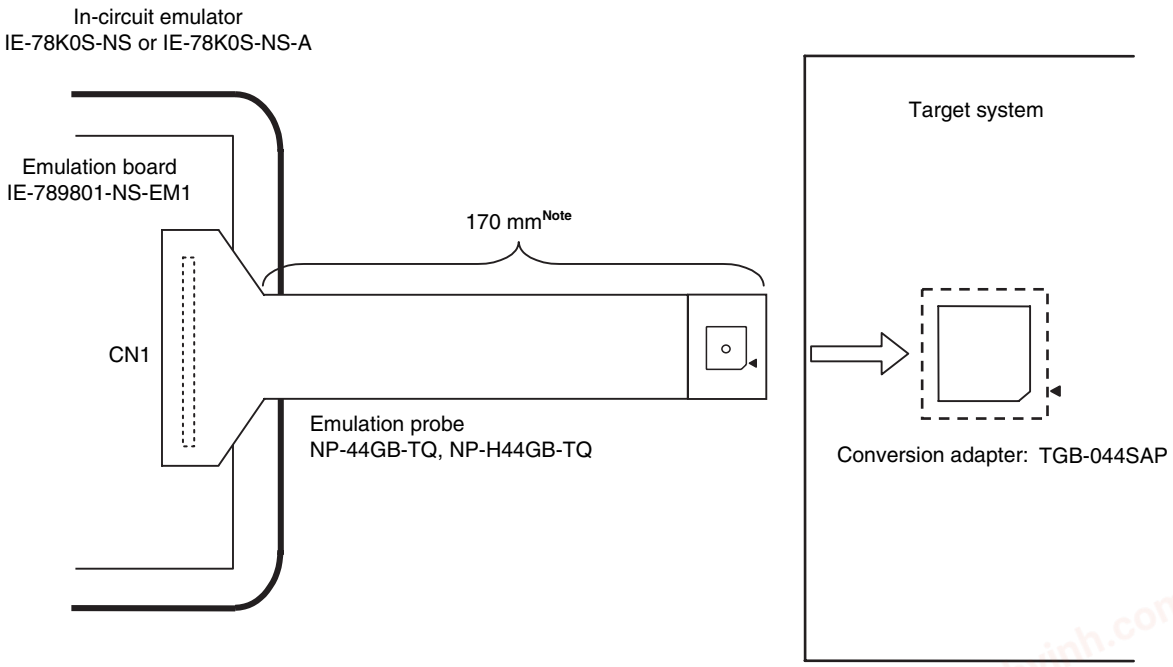
××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series IBM PC/AT compatibles	Japanese Windows	3.5" 2HD FD
BB13		English Windows	
AB17		Japanese Windows	CD-ROM
BB17		English Windows	

chanhvinh.com

★ A.7 Notes on Target System Design

Figures A-2 and A-3 show the conditions when connecting the emulation probe to the conversion adapter. Follow the configuration below and consider the shape of parts to be mounted on the target system when designing a system.

Figure A-2. Distance Between In-Circuit Emulator and Conversion Adapter

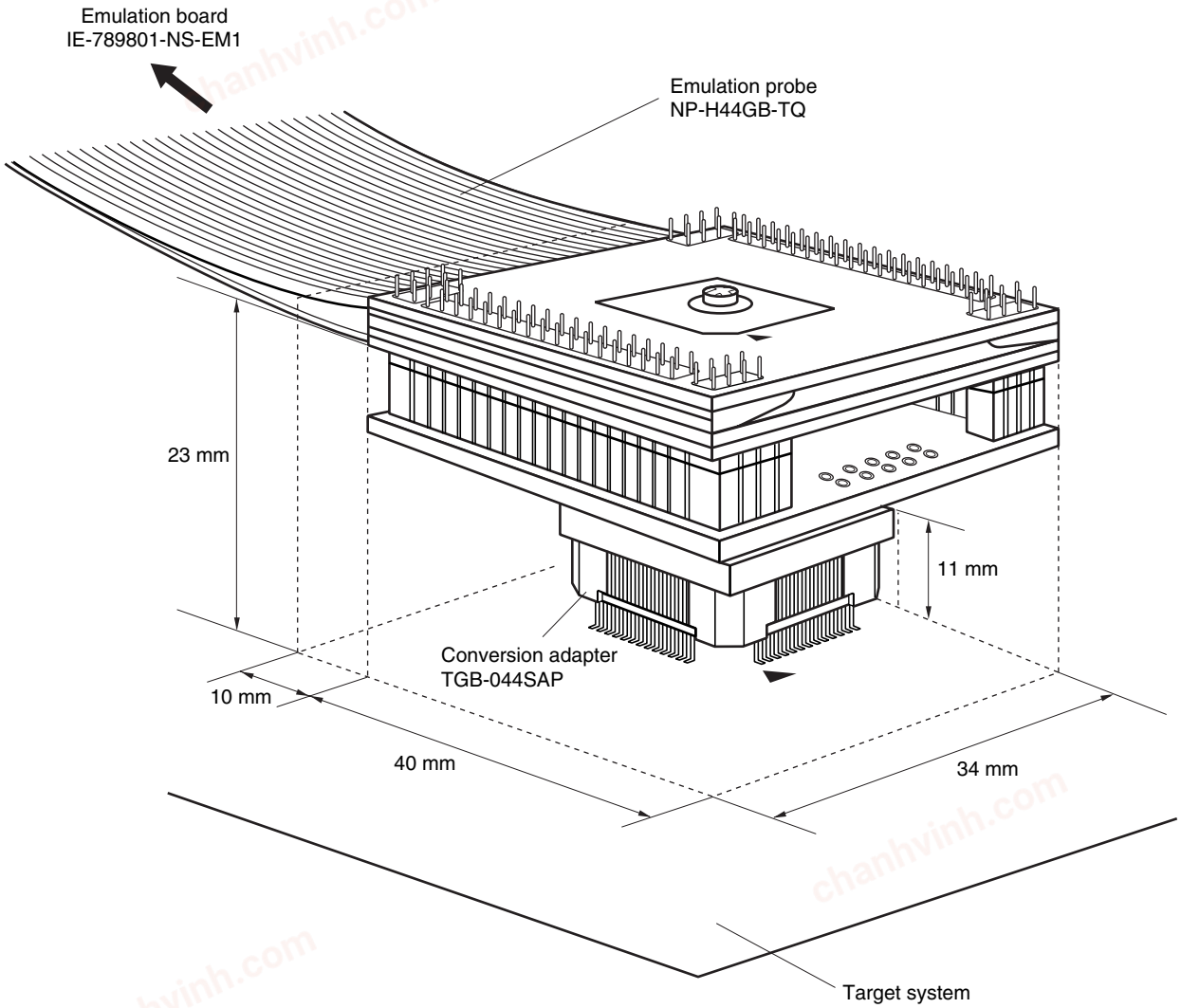


Note Distance when NP-44GB-TQ is used. When NP-H44GB-TQ is used, the distance is 370 mm.

Remarks 1. NP-44GB-TQ and NP-H44GB-TQ are products of Naito Densei Machida Mfg. Co., Ltd.
2. TGB-044SAP is a product of TOKYO ELETECH CORPORATION.

chanhvinh.com

Figure A-3. Connection Condition of Target System (NP-H44GB-TQ)



Remarks 1. NP-H44GB-TQ is a product of Naito Densei Machida Mfg. Co., Ltd.

2. TGB-044SAP is a product of TOKYO ELETECH CORPORATION.

chanhvinh.com

APPENDIX B REGISTER INDEX

B.1 Register Index (Alphabetic Order of Register Name)

8-bit compare register 00 (CR00) 82
8-bit compare register 01 (CR01) 82
8-bit timer mode control register 00 (TMC00) 83
8-bit timer mode control register 01 (TMC01) 84
8-bit timer counter 00 (TM00) 82
8-bit timer counter 01 (TM01) 82

[D]

Data/handshake packet receive result store register (DRXRSL) 114
Data packet transmit reservation register (DTXRSV) 116
Data/handshake packet receive byte number counter (DRXCON) 107
Data/handshake packet receive mode register (URXMOD) 110
Data packet transmit byte number counter (DTXCO0, DTXCO1) 107
Data/handshake PID compare register (DIDCMP) 109

[E]

External interrupt mode register 0 (INTM0) 169

[H]

Handshake packet transmit reservation register (HTXRSV) 117

[I]

Interrupt mask flag register 0 (MK0) 169
Interrupt mask flag register 1 (MK1) 169
Interrupt request flag register 0 (IF0) 168
Interrupt request flag register 1 (IF1) 168

[K]

Key return mode register 00 (KRM00) 171

[O]

Oscillation stabilization time select register (OSTS) 180

[P]

Packet receive status register (RXSTAT) 112
Port 0 (P0) 61
Port 1 (P1) 62
Port 2 (P2) 63
Port 4 (P4) 69
Port mode register 0 (PM0) 70
Port mode register 1 (PM1) 70

chanhvinh.com

Port mode register 2 (PM2) 70
Port mode register 4 (PM4) 70
Port output mode register 0 (POM0) 72
Port output mode register 1 (POM1) 72
Processor clock control register (PCC) 75
Pull-up resistor option register 0 (PU0) 71

[R]

Receive data address (USBR0 to USBR7) 104
Receive data PID (USBRD) 104
Receive token address (USBRAL, USBRAH) 103
Receive token PID (USBRTP) 103
Remote wake-up control register (REMWUP) 122

[S]

Serial operation mode register 10 (CSIM10) 158

[T]

Timer clock select register 2 (TCL2) 94
Token address compare register (ADRCMP) 108
Token packet receive result store register (TRXRSL) 114
Token PID compare register (TIDCMP) 107
Transmit/receive shift register 10 (SIO10) 156
Transmit data bank 0 address (USBT00 to USBT07) 105
Transmit data bank 1 address (USBT10 to USBT17) 105
Transmit data PID bank 0 (USBTD0) 105
Transmit data PID bank 1 (USBTD1) 105
Transmit/receive pointer (USBPOB, USBPOW) 102

[U]

USB receiver enable register (USBMOD) 110
USB timer start reservation control register (USBTCL) 121

[W]

Watchdog timer mode register (WDTM) 95

chanhvinh.com

B.2 Register Index (Alphabetic Order of Register Symbol)

[A]

ADRCMP: Token address compare register 108

[C]

CR00: 8-bit compare register 00 82
CR01: 8-bit compare register 01 82
CSIM10: Serial operation mode register 10 158

[D]

DIDCMP: Data/handshake PID compare register 109
DRXCON: Data/handshake packet receive byte number counter 107
DRXRSL: Data/handshake packet receive result store register 114
DTXCO0, DTXCO1: Data packet transmit byte number counter 107
DTXRSV: Data packet transmit reservation register 116

[H]

HTXRSV: Handshake packet transmit reservation register 117

[I]

IF0: Interrupt request flag register 0 168
IF1: Interrupt request flag register 1 168
INTM0: External interrupt mode register 0 169

[K]

KRM00: Key return mode register 00 171

[M]

MK0: Interrupt mask flag register 0 169
MK1: Interrupt mask flag register 1 169

[O]

OSTS: Oscillation settling time select register 180

[P]

P0: Port 0 61
P1: Port 1 62
P2: Port 2 63
P4: Port 4 69
PCC: Processor clock control register 75
PM0: Port mode register 0 70
PM1: Port mode register 1 70
PM2: Port mode register 2 70
PM4: Port mode register 4 70
POM0: Port output mode register 0 72
POM1: Port output mode register 1 72

chanhvinh.com

PU0: Pull-up resistor option register 0 71

[R]

REMWUP: Remote wake-up control register 122
RXSTAT: Packet receive status register 112

[S]

SIO10: Transmit/receive shift register 10 156

[T]

TCL2: Timer clock select register 2 94
TIDCMP: Token PID compare register 107
TM00: 8-bit timer counter 00 82
TM01: 8-bit timer counter 01 82
TMC00: 8-bit timer mode control register 00 83
TMC01: 8-bit timer mode control register 01 84
TRXRSL: Token packet receive result store register 114

[U]

URXMOD: Data/handshake packet receive mode register 110
USBMOD: USB receiver enable register 110
USBPOB, USBPOW: Transmit/receive pointer 102
USBR0 to USBR7: Receive data address 104
USBRAL, USBRAH: Receive token address 103
USBRD: Receive data PID 104
USBRTP: Receive token PID 103
USBTCL: USB timer start reservation control register 121
USBT00 to USBT07: Transmit data bank 0 address 105
USBT10 to USBT17: Transmit data bank 1 address 105
USBTD0: Transmit data PID bank 0 105
USBTD1: Transmit data PID bank 1 105

[W]

WDTM: Watchdog timer mode register 95

chanhvinh.com

APPENDIX C REVISION HISTORY

The revision history is described below. The “Applied to” column indicates the chapters in each edition.

(1/3)

Edition	Major Revisions from Previous Edition	Applied to:
2nd edition	Deletion of description “under development” for μPD789800, since it has been developed	Throughout
	Addition of GB-8ES type package	
	Modification of recommended connection of unused pins in type of input/output circuit for each pin	CHAPTER 2 PIN FUNCTIONS
	Addition of illustration in direct addressing	CHAPTER 3 CPU ARCHITECTURE
	Addition of caution regarding operation of 8-bit compare register n (CR0n)	CHAPTER 6 8-BIT TIMER/EVENT COUNTER
	Addition to setting method in description of operation as interval timer	
	Addition to setting method in description of operation as external event counter	
	Addition to setting method in description of operation as square wave output	
	Addition of the value of each register when SETUP received	CHAPTER 8 USB FUNCTION
	Deletion of setting RXSTAT with a 1-bit memory manipulation instruction in description of packet receive status register (RXSTAT)	
	Modification of each bit of RXSTAT from reserved words to non-reserved words in packet receive status register format	
	Modification of note in Format of Packet Receive Status Register	
	Modification of flag names in Conditions in Transmit Reservation	
	Modification of contents in block diagram of regulator and USB driver/receiver	CHAPTER 10 REGULATOR
	Addition of caution regarding replacement of flash memory version with mask ROM version	CHAPTER 14 μPD78F9801
	Modification of description from Flashpro II to Flashpro III in μPD78F9801	
	Addition of restrictions to pins used when pseudo 3-wire mode is selected as communication mode	
	Addition of setting example when Flashpro III (PG-FP3) is used	
	Whole appendix revised for supporting IE-78K0S-NS	APPENDIX A DEVELOPMENT TOOLS
	Addition of ordering information of MX78K0S in embedded software	APPENDIX B EMBEDDED SOFTWARE
3rd	Deletion of CU-type and GB-3BS type packages	Throughout
	Deletion of indication “under development” for μPD78F9801	
	Modification of operating ambient temperature when flash memory is written in **1.1 Features**	CHAPTER 1 GENERAL
	Addition of outline of timer in **1.7 Functions**	
	Modification of handling of REGC and V_{PP} pins	CHAPTER 2 PIN FUNCTIONS

chanhvinh.com

(2/3)

Edition	Major Revisions from Previous Edition	Applied to:
3rd	Correction of address values in **Figure 3-1 Memory Map (μPD789800)** and **Figure 3-2 Memory Map (μPD78F9801)**	CHAPTER 3 CPU ARCHITECTURE
	Modification of **Figure 5-3 External Circuit of System Clock Oscillator (b) External clock**	CHAPTER 5 CLOCK GENERATOR
	• Modification of chapter composition • Standardization of buffer name indications as receive token bank, receive data bank, and transmit data banks 0 and 1 • Addition of image diagrams for reception and transmission • Addition of register value for SETUP reception • Modification of description on data handshake packet receive mode register (URXMOD) • Addition of description on packet receive status register (RXSTAT) and modification of read-only bit • Addition of **Note** for token packet receive result store register (TRXRSL) • Addition of **Caution** for data packet transmit reservation register (DTXRSV) • Modification of description of bit 1 (DNAEN) of handshake packet transmit reservation register (HTXRSV) • Change of contents of **8.5.2 Remote wakeup control operation** • Addition of **Table 8-4 List of Sources of Interrupts from USB Function** • Correction of incorrect flag name in **8.6 Interrupt Request from USB Function** • Addition of description on USB reset/Resume detection interrupt (INTUSBRE) • Addition of **8.7 USB Function Control**	CHAPTER 8 USB FUNCTION
	Modification of **Figure 10-1 Block Diagram of Regulator and USB Driver/Receiver** and **Cautions**	CHAPTER 10 REGULATOR
	Addition of **Remark** in **Table 11-1 Interrupt Source List**	CHAPTER 11 INTERRUPT FUNCTIONS
	Addition of **Caution 3** on watchdog timer interrupt to **Figure 11-2 Format of Interrupt Request Flag Register**	
	Addition of **12.2.2 STOP mode (3) Cautions on STOP instruction execution**	CHAPTER 12 STANDBY FUNCTION
	Revision of contents of flash memory programming as **14.1 Flash Memory Characteristics**	CHAPTER 14 μPD78F9801
	Addition of **CHAPTER 16 ELECTRICAL SPECIFICATIONS**	CHAPTER 16 ELECTRICAL SPECIFICATIONS
	Addition of **CHAPTER 17 PACKAGE DRAWING**	CHAPTER 17 PACKAGE DRAWING
	Addition of **CHAPTER 18 RECOMMENDED SOLDERING CONDITIONS**	CHAPTER 18 ECOMMENDED SOLDERING CONDITIONS
	Revision of **APPENDIX A DEVELOPMENT TOOLS** Deletion of embedded software and addition of notes on target system design	APPENDIX A DEVELOPMENT TOOLS
	Addition of the revision contents in 3rd edition in **APPENDIX C REVISION HISTORY**	APPENDIX C REVISION HISTORY
3rd (modification ver.2)	Modification of **Figure 6-8. Timing of External Event Counter Operation (with Rising Edge Specified)**	CHAPTER 6 8-BIT TIMER/EVENT COUNTER
	Modification of conditions of V_{IL2} and V_{OL2}	CHAPTER 16 ELECTRICAL SPECIFICATIONS

chanhvinh.com

(3/3)

Edition	Major Revisions from Previous Edition	Applied to:
3rd (modification ver.3)	Addition of lead-free products	CHAPTER 1 GENERAL
	Addition of soldering conditions of lead-free products in **Table 18-1 Surface Mounting Type Soldering Conditions**	CHAPTER 18 ECOMMENDED SOLDERING CONDITIONS