To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

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

   “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

RENESAS

Application Note

78K/0 Series

8-bit Single-chip Microcontroller

Basic (III)

μPD78054 subseries
μPD78064 subseries
μPD78078 subseries
μPD78083 subseries
μPD780018 subseries
μPD780058 subseries
μPD780308 subseries
μPD78058F subseries
μPD78064B subseries
μPD78075B subseries
μPD78098B subseries
μPD78054Y subseries
μPD78064Y subseries
μPD78078Y subseries
μPD78098 subseries
μPD780018Y subseries
μPD780058Y subseries
μPD780308Y subseries
μPD78058FY subseries
μPD78070A, 78070AY
μPD78075BY subseries

Document No. U10182EJ2V0AN00 (2nd edition)
Date Published October 1997 N

[MEMO]

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note:

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

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note:

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

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note:

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

The export of these products from Japan is regulated by the Japanese government. The export of some or all of these products may be prohibited without governmental license. To export or re-export some or all of these products from a country other than Japan may also be prohibited without a license from that country. Please call an NEC sales representative.

The application circuits and their parameters are for reference only and are not intended for use in actual design-ins.

Regional Information

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

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

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

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

NEC Electronics (Germany) GmbH
Duesseldorf, Germany
Tel: 0211-65 03 02
Fax: 0211-65 03 490

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

NEC Electronics Italiana s.r.1.
Milano, Italy
Tel: 02-66 75 41
Fax: 02-66 75 42 99

NEC Electronics (Germany) GmbH
Benelux Office
Eindhoven, The Netherlands
Tel: 040-2445845
Fax: 040-2444580

NEC Electronics (France) S.A.
Velizy-Villacoublay, France
Tel: 01-30-67 58 00
Fax: 01-30-67 58 99

NEC Electronics (France) S.A.
Spain Office
Madrid, Spain
Tel: 01-504-2787
Fax: 01-504-2860

NEC Electronics (Germany) GmbH
Scandinavia Office
Taeby, Sweden
Tel: 08-63 80 820
Fax: 08-63 80 388

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

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

NEC Electronics Singapore Pte. Ltd.
United Square, Singapore 1130
Tel: 253-8311
Fax: 250-3583

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-719-2377
Fax: 02-719-5951

NEC do Brasil S.A.
Sao Paulo-SP, Brasil
Tel: 011-889-1680
Fax: 011-889-1689

Major Revisions in This Edition

Page	Description
Throughout	Addition of following products as target products: μPD780018, 780018Y, 780058, 780058Y, 780308, 780308Y, 78058F, 78058FY, 78064B, 78075B, 78075BY, 78098B subseries, μPD78070A, 78070AY μPD78052(A), 78053(A), 78054(A) μPD78062(A), 78063(A), 78064(A) μPD78081(A), 78082(A), 78P083(A), 78081(A2) μPD78058F(A), 78058FY(A) μPD78064B(A)
	Deletion of following products as target products: μPD78P054Y, 78P064Y, 78074, 78075, 78075, 78074Y, 78075Y
p.100	Addition of **Note 2** and **Caution 2** to **Figure 4-5 Format of Watchdog Timer Mode Register**
p.113	Addition of Caution to **Figure 5-8 Format of External Interrupt Mode Register 0**
p.196	Addition of **Table 8-2 Items Supported by Each Subseries**
p.197	Addition of **Table 8-3 Registers of Serial Interface**
p.204, p206	Addition of note on using wake-up function and note on changing operation mode to **Figures 8-7** and **8-8 Format of Serial Operating Mode Register 0**
p.218, p.224	Addition of Caution to **Figures 8-16** and **8-17 Format of Automatic Data Transfer/Reception Interval Specification Register**
p.239	Addition of **Figures 8-23** and **8-24 Format of Serial Interface Pin Select Register**
p.240	μPD6252 as maintenance product in **8.1 Interface with EEPROM™ (μPD6252)**
p.250	Addition of **(5) Limitations when using I^2C** bus mode to **8.1.2 Communication in I^2C bus mode**
p.286	Addition of **(f) Limitations when using UART mode** to **8.5 Interface in Asynchronous Serial Interface (UART) Mode**
p.347	Addition of **Figure 11-3 Format of Port Mode Register 12**
p.216, p.217 p.229-p.232 p.352, p.353	Description of following register formats and tables for each subseries: Figures 8-14 and 8-15 Format of Automatic Data Transmission/Reception Control Register Tables 8-4, 8-5, and 8-6 Setting of Operation Modes of Serial Interface Channel 2 Figures 12-1 and 12-2 Format of LCD Display Mode Register
p.387	Addition of **APPENDIX B REVISION HISTORY**

The mark ★ shows major revised points.

INTRODUCTION

Readers

This Application Note is intended for use by engineers who understand the functions of the 78K/0 series and wish to design application programs with the following subseries products:

- **Subseries**

μPD78054 subseries	:	μPD78052, 78053, 78054, 78P054, 78055, 78056, 78058, 78P058, 78052(A), 78053(A), 78054(A)
μPD78054Y subseries	:	μPD78052Y, 78053Y, 78054Y, 78055Y, 78056Y, 78058Y, 78P058Y
μPD78064 subseries	:	μPD78062, 78063, 78064, 78P064, 78062(A), 78063(A), 78064(A)
μPD78064Y subseries	:	μPD78062Y, 78063Y, 78064Y
μPD78078 subseries	:	μPD78076, 78078, 78P078
μPD78078Y subseries	:	μPD78076Y, 78078Y, 78P078Y
μPD78083 subseries	:	μPD78081, 78082, 78P083, 78081(A), 78082(A), 78P83(A), 78081(A2)
μPD78098 subseries	:	μPD78094, 78095, 78096, 78098A[Note 1], 78P098A[Note 1]
μPD780018 subseries	:	μPD780016[Note 2], 780018[Note 2], 78P0018[Note 2]
μPD780018Y subseries	:	μPD780016Y[Note 2], 780018Y[Note 2], 78P0018Y[Note 2]
μPD780058 subseries	:	μPD780053[Note 1], 780054[Note 1], 780055[Note 1], 780056[Note 1], 780058[Note 1], 78F0058[Note 1]
μPD780058Y subseries	:	μPD780053Y[Note 2], 780054Y[Note 2], 780055Y[Note 2], 780056Y[Note 2], 780058Y[Note 2], 78F0058Y[Note 2]
μPD780308 subseries	:	μPD780306[Note 1], 780308[Note 1], 78P0308[Note 1]
μPD780308Y subseries	:	μPD780306Y[Note 1], 780308Y[Note 1], 78P0308Y[Note 1]
μPD78058F subseries	:	μPD78056F, 78058F, 78P058F, 78058F(A)
μPD78058FY subseries	:	μPD78056FY, 78058FY, 78P058FY, 78P058FY(A)
μPD78064B subseries	:	μPD78064B, 78P064B, 78064B(A)
μPD78070A, 78070AY		
μPD78075B subseries	:	μPD78074B, 78075B
μPD78075BY subseries	:	μPD78074BY[Note 1], 78075BY[Note 1]
μPD78098B subseries	:	μPD78095B[Note 2], 78096B[Note 2], 78098B[Note 2], 78P098B[Note 2]

Notes 1. Under development

2. Planned

Remarks 1. The μPD78052(A), 78053(A), and 78054(A) have higher reliability than the μPD78052, 78053, and 78054.

2. The μPD78062(A), 78063(A), and 78064(A) have higher reliability than the μPD78062, 78063, and 78064.

3. The μPD78081(A), 78082(A), 78P083(A), and 78081(A2) have higher reliability than the μPD78081, 78082, and 78P083.

4. The μPD78058F(A) and 78058FY(A) have higher reliability than the μPD78058F and 78058FY.

5. The μPD78064B(A) has higher reliability than the μPD78064B.

Purpose

This Application Note is to deepen your understanding of the basic functions of the 78K/0 series by using program examples.
Note that the programs and hardware configuration shown in this document are only examples and not subject to mass production.

Organization

This Application Note consists of the following contents:

- General
- Software
- Hardware

In addition to this Application Note, the following Application Notes are also available:

Document Name	Document Number		Targeted Subseries	Contents
	Japanese	English		
78K/0 Series Application Note Basic (I)	IEA-715	IEA-1288	μPD78002, 78002Y μPD78014, 78014Y μPD78018F, 78018FY	Explains basic functions of products in 78K/0 series by using program examples
78K/0 Series Application Note Basic (II)	U10121J	U10121E	μPD78044 μPD78044H μPD780208 μPD780228	
78K/0 Series Application Note Basic (III)	U10182J	This document	μPD78054, 78054Y μPD78064, 78064Y μPD78078, 78078Y μPD78083 μPD78098 μPD780018, 780018Y μPD780058, 780058Y μPD780308, 780308Y μPD78058F, 78058FY μPD78064B μPD78070A, 78070AY μPD78075B, 78075BY μPD78098B	
78K/0 Series Application Note Floating-Point Operation Program	IEA-718	IEA-1289	All subseries in 78K/0 series (except μPD78002 and 78002Y subseries)	Explains floating-point operation programs of products in 78K/0 series
μPD78014 Series Application Note Electronic Pocketbook	IEA-744	IEA-1301	μPD78014 (only μPD78014 and 78P014)	Explains how to organize electronic pocketbook by using μPD78014 subseries

Caution The application examples and program lists shown in this Application Note assume that the main system clock operates at 4.19 MHz, not at 5.0 MHz.

How to Read This Manual Although this Application Note explains the functions of the 78K/0 series products, the functions of some products in each subseries differ from those of the others.

(1/2)

Chapter \ Subseries	μPD78054 μPD78054Y	μPD78064 μPD78064Y	μPD78078 μPD78078Y	μPD78083	μPD78098	μPD780018 μPD780018Y	μPD780058 μPD780058Y
CHAPTER 1 GENERAL	○	○	○	○	○	○	○
CHAPTER 2 FUNDAMENTALS OF SOFTWARE	○	○	○	○	○	○	○
CHAPTER 3 APPLICATIONS OF SYSTEM CLOCK SELECTION	○	○	○	○	○	○	○
CHAPTER 4 APPLICATIONS OF WATCHDOG TIMER	○	○	○	○	○	○	○
CHAPTER 5 APPLICATIONS OF 16-BIT TIMER/EVENT COUNTER	○	○	○	–	○	○	○
CHAPTER 6 APPLICATIONS OF 8-BIT TIMER/EVENT COUNTER	○	○	○	–	○	○	○
CHAPTER 7 APPLICATIONS OF WATCH TIMER	○	○	○	–	○	○	○
CHAPTER 8 APPLICATIONS OF SERIAL INTERFACE	○	○	○	○	○	○	○
CHAPTER 9 APPLICATIONS OF A/D CONVERTER	○	○	○	○	○	○	○
CHAPTER 10 APPLICATIONS OF D/A CONVERTER	○	–	○	–	○	–	○
CHAPTER 11 APPLICATION OF REAL-TIME OUTPUT PORT	○	–	○	–	○	–	○
CHAPTER 12 APPLICATIONS OF LCD CONTROLLER/DRIVER	–	○	–	–	–	–	–
CHAPTER 13 APPLICATIONS OF KEY INPUT	○	○	○	–	○	○	○

(2/2)

Chapter \ Subseries	μPD780308 μPD780308Y	μPD78058F μPD78058FY	μPD78064B	μPD78070A μPD78070AY	μPD78075B μPD78075BY	μPD78098B
CHAPTER 1 GENERAL	○	○	○	○	○	○
CHAPTER 2 FUNDAMENTALS OF SOFTWARE	○	○	○	○	○	○
CHAPTER 3 APPLICATIONS OF SYSTEM CLOCK SELECTION	○	○	○	○	○	○
CHAPTER 4 APPLICATIONS OF WATCHDOG TIMER	○	○	○	○	○	○
CHAPTER 5 APPLICATIONS OF 16-BIT TIMER/EVENT COUNTER	○	○	○	○	○	○
CHAPTER 6 APPLICATIONS OF 8-BIT TIMER/EVENT COUNTER	○	○	○	○	○	○
CHAPTER 7 APPLICATIONS OF WATCH TIMER	○	○	○	○	○	○
CHAPTER 8 APPLICATIONS OF SERIAL INTERFACE	○	○	○	○	○	○
CHAPTER 9 APPLICATIONS OF A/D CONVERTER	○	○	○	○	○	○
CHAPTER 10 APPLICATIONS OF D/A CONVERTER	–	○	–	○	○	○
CHAPTER 11 APPLICATION OF REAL-TIME OUTPUT PORT	–	○	–	○	○	○
CHAPTER 12 APPLICATIONS OF LCD CONTROLLER/DRIVER	○	–	○	–	–	–
CHAPTER 13 APPLICATIONS OF KEY INPUT	○	○	○	○	○	○

The (A)-model and standard models differ only in quality grade.
The μPD78081(A2) differs from standard models and (A)-models in terms of supply voltage and operating temperature range. For details, refer to the individual Data Sheet.
In this document, read (A)-models and (A2)-model as follows:

μPD78052	→ μPD78052(A)	μPD78053	→ μPD78053(A)
μPD78054	→ μPD78054(A)	μPD78062	→ μPD78062(A)
μPD78063	→ μPD78063(A)	μPD78064	→ μPD78064(A)
μPD78081	→ μPD78081(A)	μPD78082	→ μPD78082(A)
μPD78P083	→ μPD78P083(A)	μPD78081	→ μPD78081(A2)
μPD78058F	→ μPD78058F(A)	μPD78058FY	→ μPD78058FY(A)
μPD78064B	→ μPD78064B(A)		

Legend

Data significance	:	Left: higher digit, right: lower digit
Low active	:	$\overline{\times\times\times}$ (top bar over pin or signal name)
Note	:	Description of **Note** in the text
Caution	:	Important information
Remark	:	Supplement
Numeric representation	:	Binary ... ×××× or ××××B
		Decimal ... ××××
		Hexadecimal ... ××××H

Quality Grade

- **Standard**

  μPD78052, 78053, 78054, 78055, 78056, 78058, 78P058
  μPD78052Y, 78053Y, 78054Y, 78055Y, 78056Y, 78058Y, 78P058Y
  μPD78062, 78063, 78064, 78P064
  μPD78062Y, 78063Y, 78064Y
  μPD78076, 78078, 78P078
  μPD78076Y, 78078Y, 78P078Y
  μPD78081, 78082, 78P083
  μPD78094, 78095, 78096, 78098A, 78P098A
  μPD780016, 780018, 78P0018
  μPD780016Y, 780018Y, 78P0018Y
  μPD780053, 780054, 780055, 780056, 780058, 78F0058
  μPD780053Y, 780054Y, 780055Y, 780056Y, 780058Y, 78F0058Y
  μPD780306, 780308, 78P0308
  μPD780306Y, 780308Y, 78P0308Y
  μPD78056F, 78058F, 78P058F
  μPD78056FY, 78058FY, 78P058FY
  μPD78064B, 78P064B
  μPD78070A, 78070AY
  μPD78074B, 78075B
  μPD78074BY, 78075BY
  μPD78095B, 78096B, 78098B, 78P098B

- **Special**
  μPD78052(A), 78053(A), 78054(A)
  μPD78062(A), 78063(A), 78064(A)
  μPD78082(A), 78083(A), 78P083(A), 78081(A2)
  μPD78058F(A), 78058FY(A)
  μPD78064B(A)

Please refer to "Quality Grades on NEC Semiconductor Devices" (Document No. C11531E) published by NEC Corporation to know the specification of quality grade on the devices and its recommended applications.

Application Field

- Consumer appliances

Related documents

Some of the related documents listed below are preliminary versions but not so specified here.

- **Common related documents**

Document Name	Document Number	
	Japanese	English
78K/0 Series Application Note - Basic (III)	U10182J	This document
78K/0 Series Application Note - Floating-Point Operation Program	IEA-718	IEA-1289
78K/0 Series User's Manual - Instruction	U12326J	U12326E
78K/0 Series Instruction Set	U10904J	–
78K/0 Series Instruction Table	U10903J	–

- **Documents dedicated to product**

(1) μPD78054 subseries

Document Name	Document Number	
	Japanese	English
μPD78052, 78053, 78054, 78055, 78056, 78058 Data Sheet	U12327J	IC-3403
μPD78P054 Data Sheet	U12346J	U12346E
μPD78P058 Data Sheet	IC-8884	U10417E
μPD78054, μPD78054Y Subseries User's Manual	U11747J	U11747E
μPD78054 Subseries Special Function Register Table	U10102J	–
μPD78052(A), 78053(A), 78054(A) Data Sheet	U12171J	U12171E

(2) μPD78054Y subseries

Document Name	Document Number	
	Japanese	English
μPD78052Y, 78053Y, 78054Y, 78056Y, 78058Y Data Sheet	U10906J	U10906E
μPD78P058Y Data Sheet	U10907J	U10907E
μPD78054, 78054Y Subseries User's Manual	U11747J	U11747E
μPD78054Y Subseries Special Function Register Table	U10087J	–

(3) μPD78064 subseries

Document Name	Document Number	
	Japanese	English
μPD78062, 78063, 78064 Data Sheet	U12238J	U12338E
μPD78P064 Data Sheet	U12589J	U12589E
μPD78062(A), 78063(A), 78064(A) Data Sheet	U10335J	U10335E
μPD78064, 78064Y Subseries User's Manual	U10105J	U10105E
μPD78064 Subseries Special Function Register Table	IEM-5568	–

(4) μPD78064Y subseries

Document Name	Document Number	
	Japanese	English
μPD78062Y, 78063Y, 78064Y Data Sheet	U10330J	U10330E
μPD78064, 78064Y Subseries User's Manual	U10105J	U10105E
μPD78064Y Subseries Special Function Register Table	IEM-5583	–

(5) μPD78078 subseries

Document Name	Document Number	
	Japanese	English
μPD78076, 78078 Data Sheet	U10167J	U10167E
μPD78P078 Data Sheet	U10168J	U10168E
μPD78078 Subseries User's Manual	U10641J	U10641E
μPD78078 Subseries Special Function Register Table	IEM-5607	–

(6) μPD78078Y subseries

Document Name	Document Number	
	Japanese	English
μPD78076Y, 78078Y Data Sheet	U10605J	U10605E
μPD78P078Y Data Sheet	U10606J	U10606E
μPD78078, 78078Y Subseries User's Manual	U10641J	U10641E
μPD78078Y Subseries Special Function Register Table	U10257J	–

(7) μPD78083 subseries

Document Name	Document Number	
	Japanese	English
μPD78081, 78082 Data Sheet	U11415J	U11415E
μPD78P083 Data Sheet	U11006J	U11006E
uPD78081(A), 78082(A) Data Sheet	U12436J	To be released soon
uPD78P083(A) Data Sheet	U12175J	U12175E
μPD78083 Subseries User's Manual	U12176J	U12176E
μPD78083 Subseries Special Function Register Table	IEM-5599	–

(8) μPD78098 subseries

Document Name	Document Number	
	Japanese	English
μPD78094, 78095, 78096, 78098A Data Sheet	U10146J	U10146E
μPD78P098A Data Sheet	U10203J	U10203E
μPD78098 Subseries User's Manual	IEU-854	IEU-1381
μPD78098 Subseries Special Function Register List	IEM-5591	–

(9) μPD780018 subseries

Document Name	Document Number	
	Japanese	English
μPD780016, 780018 Preliminary Product Information	Plan to prepare	Plan to prepare
μPD78P0018 Preliminary Product Information	Plan to prepare	Plan to prepare
μPD780018, 780018Y Subseries User's Manual	Plan to prepare	Plan to prepare

(10) μPD780018Y subseries

Document Name	Document Number	
	Japanese	English
μPD780016Y, 780018Y Preliminary Product Information	U11810J	U11810E
μPD78P0018Y Preliminary Product Information	Plan to prepare	Plan to prepare
μPD780018, 780018Y Subseries User's Manual	Plan to prepare	Plan to prepare

(11) μPD780058 subseries

Document Name	Document Number	
	Japanese	English
μPD780053, 780054, 780055, 780056, 780058 Preliminary Product Information	U12182J	U12182E
μPD78F0058 Preliminary Product Information	U12092J	U12092E
μPD780058, 780058Y Subseries User's Manual	U12013J	U12013E

(12) μPD780058, 780058Y subseries

Document Name	Document Number	
	Japanese	English
μPD780053Y, 780054Y, 780055Y, 780056Y, 780058Y Preliminary Product Information	Plan to prepare	Plan to prepare
μPD78F0058Y Preliminary Product Information	U12324J	U12324E
μPD780058, 780058Y Subseries User's Manual	U12013J	U12013E

(13) μPD780308 subseries

Document Name	Document Number	
	Japanese	English
μPD780306, 780308 Data Sheet	U11105J	U11105E
μPD78P0308 Preliminary Product Information	U11776J	U11776E
μPD780308, 780308Y Subseries User's Manual	U11377J	U11377E

(14) μPD780308Y subseries

Document Name	Document Number	
	Japanese	English
μPD780306Y, 780308Y Data Sheet	U12251J	U12251E
μPD78P0308Y Preliminary Product Information	U11832J	U11832E
μPD780308, 780308Y Subseries User's Manual	U11377J	U11377E

(15) μPD78058F subseries

Document Name	Document Number	
	Japanese	English
μPD78056F, 78058F Data Sheet	U11795J	U11795E
μPD78P058F Data Sheet	U11796J	U11796E
μPD78058F(A) Data Sheet	To be released soon	Plan to prepare
μPD78058F, 78058FY Subseries User's Manual	U12068J	U12068E

(16) μPD78058FY subseries

Document Name	Document Number	
	Japanese	English
μPD78056FY, 78058FY Data Sheet	U12142J	U12142E
μPD78P058FY Data Sheet	U12076J	U12076E
μPD78058F, 78058FY Subseries User's Manual	U12068J	To be released soon

(17) μPD78064B subseries

Document Name	Document Number	
	Japanese	English
μPD78064B Data Sheet	U11590J	U11590E
μPD78064B(A) Data Sheet	U11597J	U11597E
μPD78P064B Data Sheet	U11598J	U11598E
μPD780308, 780308Y User's Manual	U10785J	U10785E

(18) μPD78070A, 78070AY subseries

Document Name	Document Number	
	Japanese	English
μPD78070A Data Sheet	U10326J	U10326E
μPD78070AY Data Sheet	U10542J	U10542E
μPD78070A, 78070AY User's Manual	IEU-907	U10200E
μPD78070A	U10133J	–
μPD78070AY	U10134J	–

(19) μPD78075B subseries

Document Name	Document Number	
	Japanese	English
μPD78074B, 78075B Data Sheet	U12017J	U12017E
μPD78075B, 78075BY Subseries User's Manual	U12560J	To be released soon

(20) μPD78075BY subseries

Document Name	Document Number	
	Japanese	English
μPD78074BY, 78075BY Data Sheet	Plan to prepare	Plan to prepare
μPD78075B, 78075BY Subseries User's Manual	U12560J	To be released soon

(21) μPD78098B subseries

Document Name	Document Number	
	Japanese	English
μPD78095B, 78096B, 78098B Data Sheet	Plan to prepare	Plan to prepare
μPD78P098B Data Sheet	Plan to prepare	Plan to prepare
μPD78098B Subseries User's Manual	To be released soon	Plan to prepare

The contents of the above related documents are subject to change without notice. Be sure to use the latest edition when you design your system.

CONTENTS

LIST OF FIGURES (1/6)

LIST OF FIGURES (2/6)

LIST OF FIGURES (3/6)

LIST OF FIGURES (4/6)

LIST OF FIGURES (5/6)

LIST OF FIGURES (6/6)

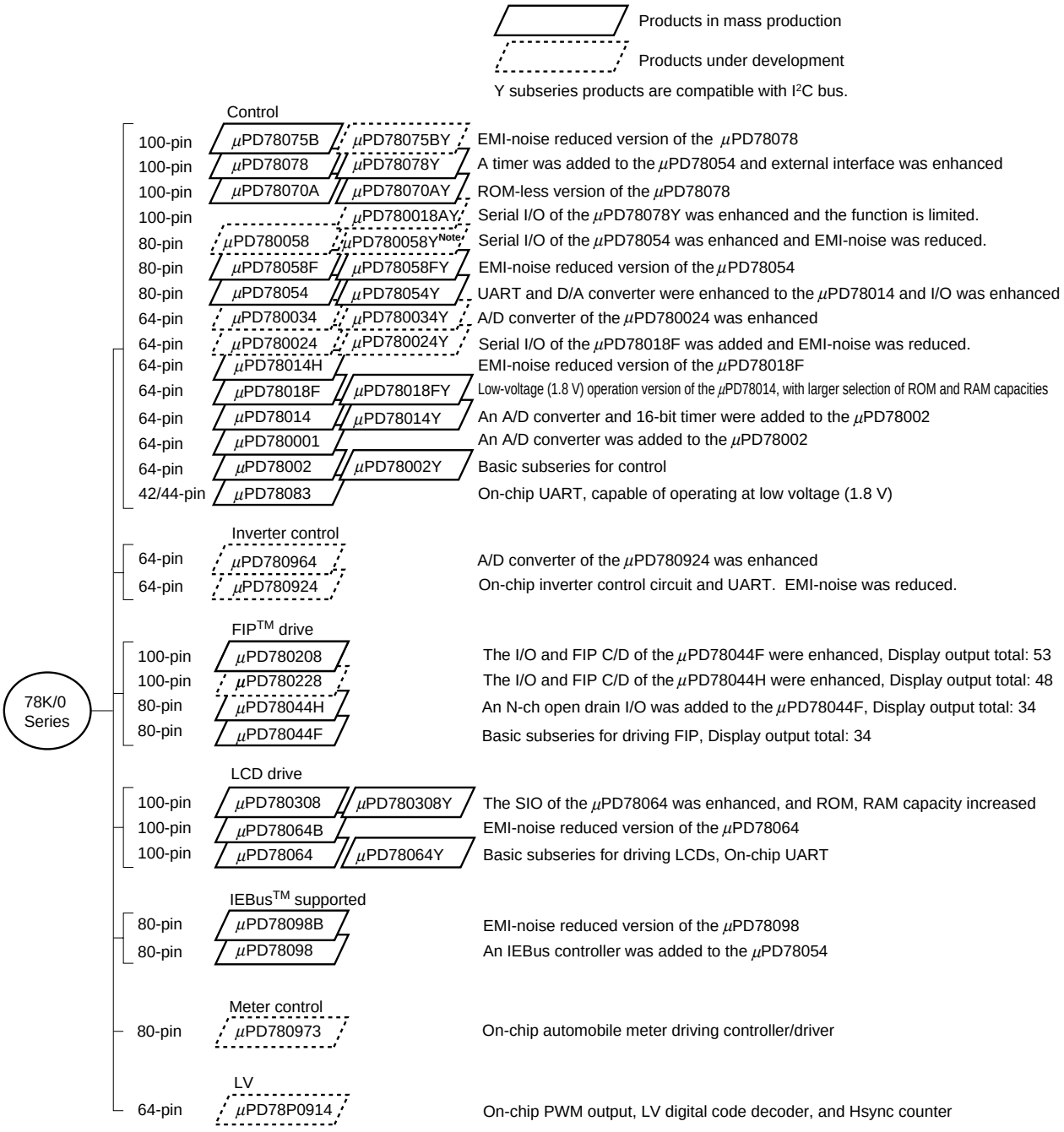
LIST OF TABLES (1/2)

LIST OF TABLES (2/2)

CHAPTER 1 GENERAL

★ 1.1 Product Development of 78K/0 Series

The following shows the products organized according to usage. The names in the parallelograms are subseries names.



Note Under planning

The following lists the main functional differences between subseries products.

Subseries Name \ Function		ROM Capacity	Timer				8-bit A/D	10-bit A/D	8-bit D/A	Serial Interface	I/O	V_{DD} MIN. Value	External Expansion
			8-bit	16-bit	Watch	WDT							
Control	μPD78075B	32K-40K	4ch	1ch	1ch	1ch	8ch	–	2ch	3ch (UART: 1ch)	88	1.8 V	○
	μPD78078	48K-60K											
	μPD78070A	–									61	2.7 V	
	μPD780058	24K-60K	2ch						2ch	3ch (time division UART: 1ch)	68	1.8 V	
	μPD78058F	48K-60K								3ch (UART: 1ch)	69	2.7 V	
	μPD78054	16K-60K										2.0 V	
	μPD780034	8K-32K					–	8ch	–	3ch (UART: 1ch, time division 3-wire: 1ch)	51	1.8 V	
	μPD780024						8ch	–					
	μPD78014H									2ch	53	1.8 V	
	μPD78018F	8K-60K											
	μPD78014	8K-32K										2.7 V	
	μPD780001	8K		–	–					1ch	39		–
	μPD78002	8K-16K			1ch		–				53		○
	μPD78083				–		8ch			1ch (UART: 1ch)	33	1.8 V	–
Inverter control	μPD780964	8K-32K	3ch	**Note**	–	1ch	–	8ch	–	2ch (UART: 2ch)	47	2.7 V	○
	μPD780924						8ch	–					
FIP drive	μPD780208	32K-60K	2ch	1ch	1ch	1ch	8ch	–	–	2ch	74	2.7 V	–
	μPD780228	48K-60K	3ch	–	–					1ch	72	4.5 V	
	μPD78044H	32K-48K	2ch	1ch	1ch						68	2.7 V	
	μPD78044F	16K-40K								2ch			
LCD drive	μPD780308	48K-60K	2ch	1ch	1ch	1ch	8ch	–	–	3ch (time division UART: 1ch)	57	2.0 V	–
	μPD78064B	32K								2ch (UART: 1ch)			
	μPD78064	16K-32K											
IEBus supported	μPD78098	40K-60K	2ch	1ch	1ch	1ch	8ch	–	2ch	3ch (UART: 1ch)	69	2.7 V	○
	μPD78098B	32K-60K											
Meter control	μPD780973	24K-32K	3ch	1ch	1ch	1ch	5ch	–	–	2ch (UART: 1ch)	56	4.5 V	–
LV	μPD78P0914	32K	6ch	–	–	1ch	8ch	–	–	2ch	54	4.5 V	○

Note 10-bit timer: 1 channel

1.2 Features of 78K/0 Series

The 78K/0 series is a collection of 8-bit single-chip microcontrollers ideal for commercial systems.

The μPD78054 and 78054Y subseries are provided with peripheral hardware functions such as an A/D converter, D/A converter, timer, serial interface, real-time output port, and interrupt function.

The μPD78064 and 78064Y subseries are provided with peripheral hardware functions such as an LCD controller/driver, A/D converter, timer, serial interface, and interrupt function.

The μPD78078 and 78078Y subseries are based on the μPD78054 and 78054Y subseries with a timer added and the external interface function reinforced.

The μPD78083 subseries is provided with peripheral hardware functions such as an A/D converter, timer, serial interface, and interrupt function.

The μPD78098 subseries is based on the μPD78054 subseries with an IEBus controller added.

★ The μPD780018 and 780018Y subseries are versions of the μPD78078 and 78078Y subseries (serial interface with time division transfer function) with an improved serial interface and a limited number of functions.

★ The μPD780058 and 780058Y subseries are low-EMI noise versions of the μPD78054 and 78054Y subseries (serial interface with time division transfer function), with an improved serial interface.

★ The μPD780308 and 780308Y subseries are versions of the μPD78064 and 78064Y subseries with increased ROM and RAM with an improved serial interface.

★ The μPD78058F, 78058FY, 78064B, 78075B, 78075BY, and 78098B subseries are low-EMI noise versions of the μPD78054, 78054Y, 78064, 78078, 78078Y, and 78098 subseries.

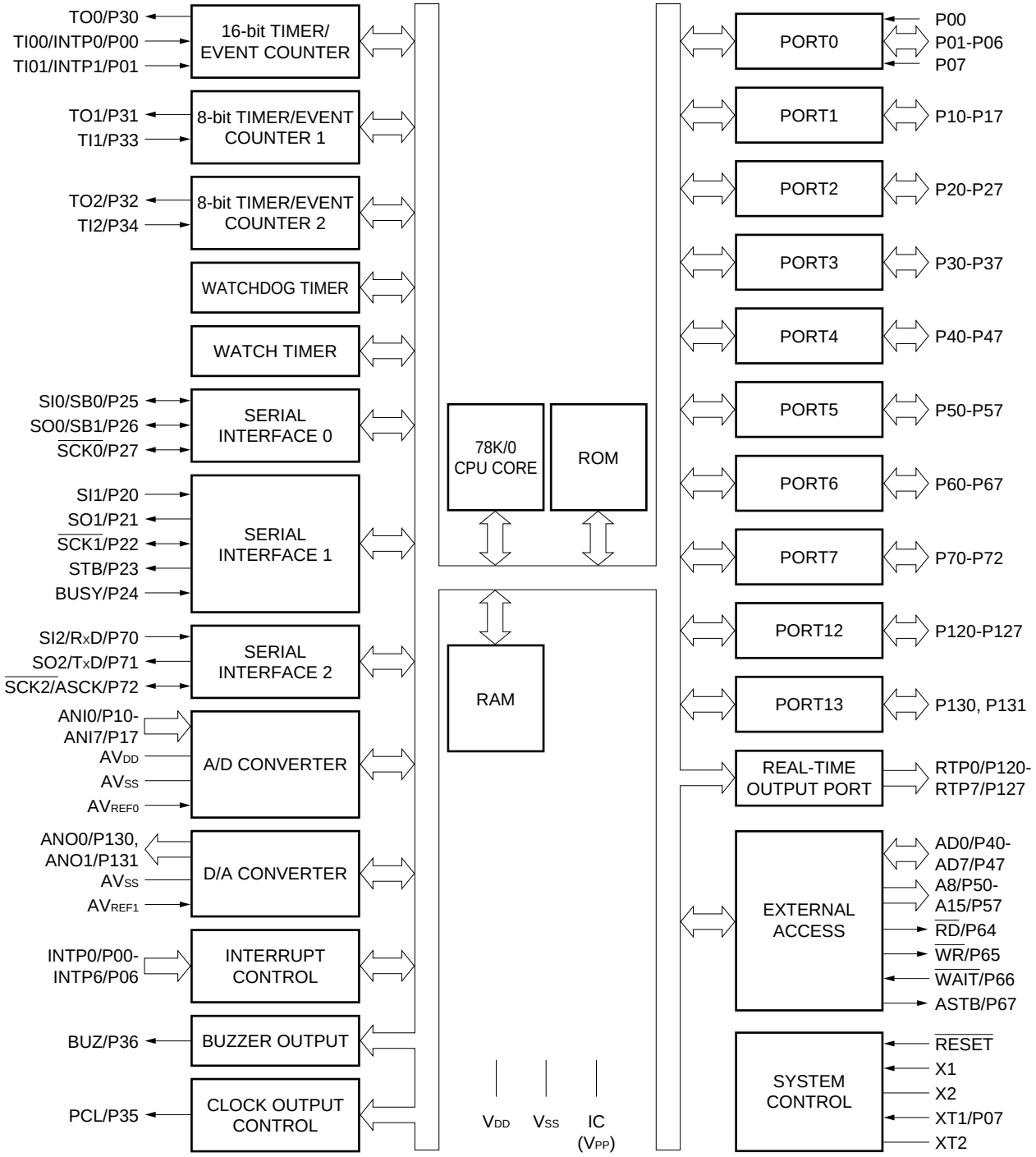
★ The μPD78070A and 78070AY subseries are the ROM-less versions of the μPD78078 and 78078Y subseries.

The μPD78054Y, 78064Y, 78078Y, 780058Y, 780308Y, 78058FY, 78075BY subseries and μPD78070AY are provided with I^2C bus control function instead of the SBI function of the μPD78054, 78064, 78078, 780058, 780308, 78058F, 78075B subseries and μPD78070A.

In addition, one-time PROM, EPROM, or flash-memory models that can operate at the same operating voltage as the mask ROM models and that are ideal for early and small-scale production of the application system are also available.

The block diagram and function outline of each series is shown on the following pages.

Figure 1-1. Block Diagram of μPD78054 Subseries



Remarks 1. The internal ROM and RAM capacities differ depending on the model.

2. ( ): μPD78P054, 78P058

Table 1-1. Functional Outline of μPD78054 Subseries (1/2)

Item / Part Number		μPD78052	μPD78053	μPD78054	μPD78P054 Note 1	μPD78055	μPD78056	μPD78058	μPD78P058 Note 2
Internal memory	ROM	Mask ROM			PROM	Mask ROM			PROM
		16K bytes	24K bytes	32K bytes	32K bytes Note 2	40K bytes	48K bytes	60K bytes	60K bytes Note 3
	High-speed RAM	512 bytes	1024 bytes		1024 bytes Note 3	1024 bytes			1024 bytes Note 3
	Buffer RAM	32 bytes							
	Expansion RAM	None						1024 bytes	1024 bytes Note 4
Memory space		64K bytes							
General-purpose register		8 bits × 8 × 4 banks							
Minimum instruction execution time	With main system clock	0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs/12.8 μs (at 5.0 MHz)							
	With subsystem clock	122 μs (at 32.768 kHz)							
Instruction set		• 16-bit operation • Multiplication/division (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, Boolean operation) • BCD adjustment, etc.							
I/O port		• Total : 69 • CMOS input : 2 • CMOS I/O : 63 • N-ch open-drain I/O : 4							
A/D converter		8-bit resolution × 8 channels							
D/A converter		8-bit resolution × 2 channels							
Serial interface		• 3-wire serial I/O/SBI/2-wire serial I/O mode selectable : 1 channel • 3-wire serial I/O mode (with function to automatically transfer/receive up to 32 bytes) : 1 channel • 3-wire serial I/O/UART mode selectable : 1 channel							
Timer		• 16-bit timer/event counter : 1 channel • 8-bit timer/event counter : 2 channels • Watch timer : 1 channel • Watchdog timer : 1 channel							
Timer output		3 (14-bit PWM output: 1)							
Clock output		19.5 kHz, 39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5.0 MHz (with main system clock of 5.0 MHz), 32.768 kHz (with subsystem clock of 32.768 kHz)							

Notes 1. The μPD78P054 is a PROM model of the μPD78052, 78053, and 78054.

2. The μPD78P058 is a PROM model of the μPD78055, 78056, and 78058.

3. The capacities of the internal PROM and internal high-speed RAM can be changed by using a memory size select register (IMS).

4. The internal expansion RAM capacity can be changed by using an internal expansion RAM size select register (IXS).

Table 1-1. Functional Outline of μPD78054 Subseries (2/2)

Item / Part Number		μPD78052	μPD78053	μPD78054	μPD78P054 Note 1	μPD78055	μPD78056	μPD78058	μPD78P058 Note 2
Buzzer output		1.2 kHz, 2.4 kHz, 4.9 kHz, 9.8 kHz (with main system clock of 5.0 MHz)							
Vectored interrupt source	Maskable	Internal: 13, external: 7							
	Non-maskable	Internal: 1							
	Software	1							
Test input		Internal: 1, external: 1							
Supply voltage		V_{DD} = 2.0 to 6.0 V							
Package		• 80-pin plastic QFP (14 × 14 mm, resin thickness 2.7 mm) • 80-pin plastic QFP (14 × 14 mm, resin thickness 1.4 mm) Note 3 • 80-pin plastic TQFP (fine pitch) (12 × 12 mm) (μPD78052, 78053, 78054, 78P054, 78058 only) • 80-pin ceramic WQFN (14 × 14 mm) (μPD78P054, 78P058 only)							

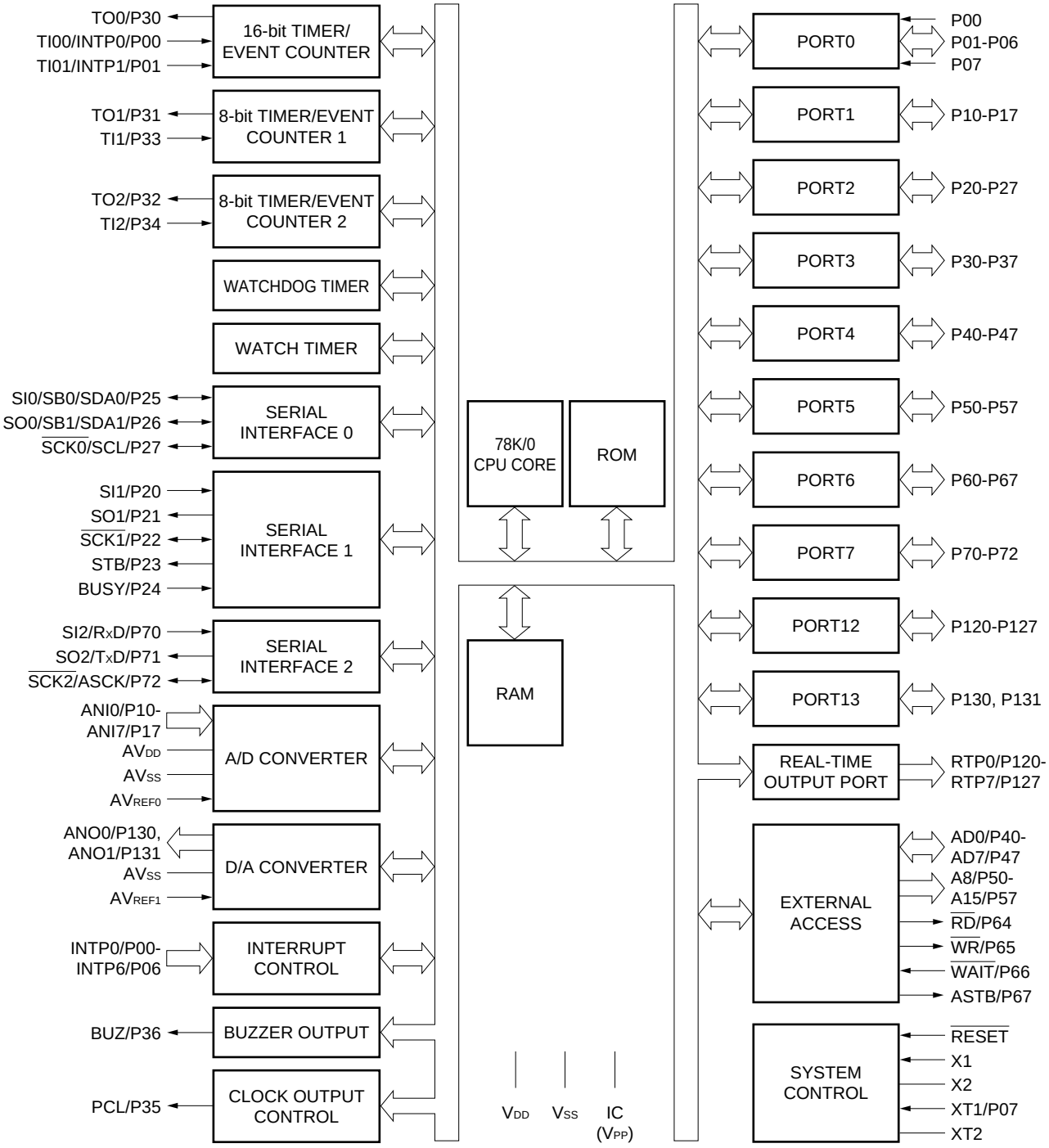
★

Notes 1. The μPD78P054 is a PROM model of the μPD78052, 78053, and 78054.

2. The μPD78P058 is a PROM model of the μPD78055, 78056, and 78058.

3. Under planning

Figure 1-2. Block Diagram of μPD78054Y Subseries



Remarks 1. The capacities of the internal ROM and RAM differ depending on the model.

2. ( ): μPD78P058Y

Table 1-2. Functional Outline of μPD78054Y Subseries (1/2)

Item / Part Number		μPD78052Y	μPD78053Y	μPD78054Y	μPD78055Y	μPD78056Y	μPD78058Y	μPD78P058Y
Internal memory	ROM	Mask ROM						PROM
		16K bytes	24K bytes	32K bytes	40K bytes	48K bytes	60K bytes	60K bytes[Note 1]
	High-speed RAM	512 bytes	1024 bytes					1024 bytes[Note 1]
	Buffer RAM	32 bytes						
	Expansion RAM	None					1024 bytes	1024 bytes[Note 2]
Memory space		64K bytes						
General-purpose register		8 bits × 8 × 4 banks						
Minimum instruction execution time	With main system clock	0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs/12.8 μs (at 5.0 MHz)						
	With subsystem clock	122 μs (at 32.768 kHz)						
Instruction set		• 16-bit operation • Multiplication/division (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, Boolean operation) • BCD adjustment, etc.						
I/O port		• Total : 69 • CMOS input : 2 • CMOS I/O : 63 • N-ch open-drain I/O : 4						
A/D converter		8-bit resolution × 8 channels						
D/A converter		8-bit resolution × 2 channels						
Serial interface		• 3-wire serial I/O/2-wire serial I/O/I^2C bus mode selectable : 1 channel • 3-wire serial I/O mode (with function to automatically transfer/receive up to 32 bytes) : 1 channel • 3-wire serial I/O/UART mode selectable : 1 channel						
Timer		• 16-bit timer/event counter : 1 channel • 8-bit timer/event counter : 2 channels • Watch timer : 1 channel • Watchdog timer : 1 channel						
Timer output		3 (14-bit PWM output: 1)						
Clock output		19.5 kHz, 39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5.0 MHz (with main system clock of 5.0 MHz), 32.768 kHz (with subsystem clock of 32.768 kHz)						

Notes 1. The capacities of the internal PROM and internal high-speed RAM can be changed by using a memory size select register (IMS).

2. The internal expansion RAM capacity can be changed by using an internal expansion RAM size select register (IXS).

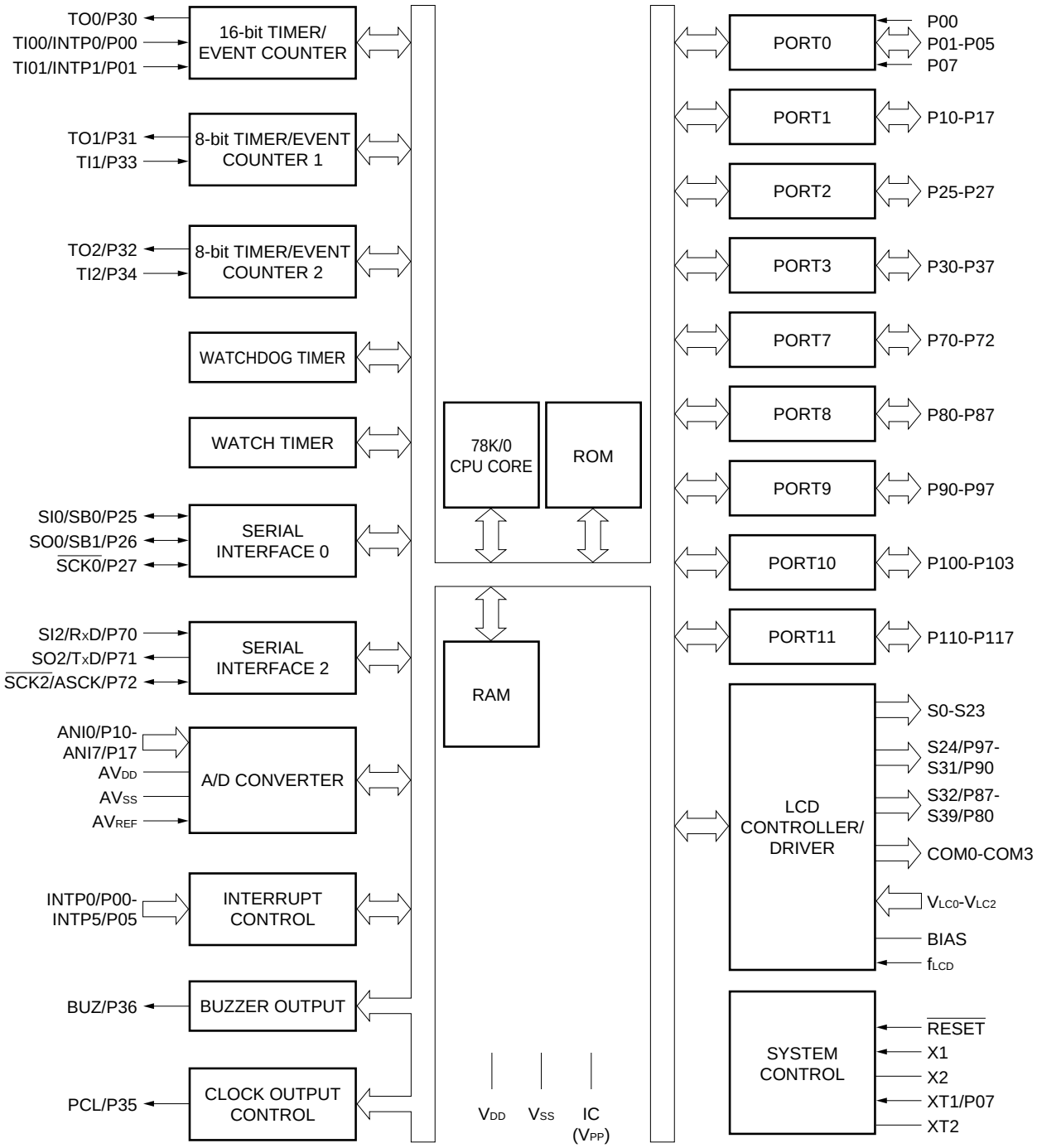
Table 1-2. Functional Outline of μPD78054Y Subseries (2/2)

Item / Part Number		μPD78052Y	μPD78053Y	μPD78054Y	μPD78055Y	μPD78056Y	μPD78058Y	μPD78P058Y
Buzzer output		1.2 kHz, 2.4 kHz, 4.9 kHz, 9.8 kHz (with main system clock of 5.0 MHz)						
Vectored interrupt source	Maskable	Internal: 13, external: 7						
	Non-maskable	Internal: 1						
	Software	1						
Test input		Internal: 1, external: 1						
Supply voltage		V_{DD} = 2.0 to 6.0 V						
Package		• 80-pin plastic QFP (14 × 14 mm, resin thickness 2.7 mm) • 80-pin plastic QFP (14 × 14 mm, resin thickness 1.4 mm)**Note** • 80-pin ceramic WQFN (14 × 14 mm)(μPD78P058Y only)						

★

Note Under planning

Figure 1-3. Block Diagram of μPD78064 Subseries



Remarks 1. The internal ROM and RAM capacities differ depending on the model.

2. ( ): μPD78P064

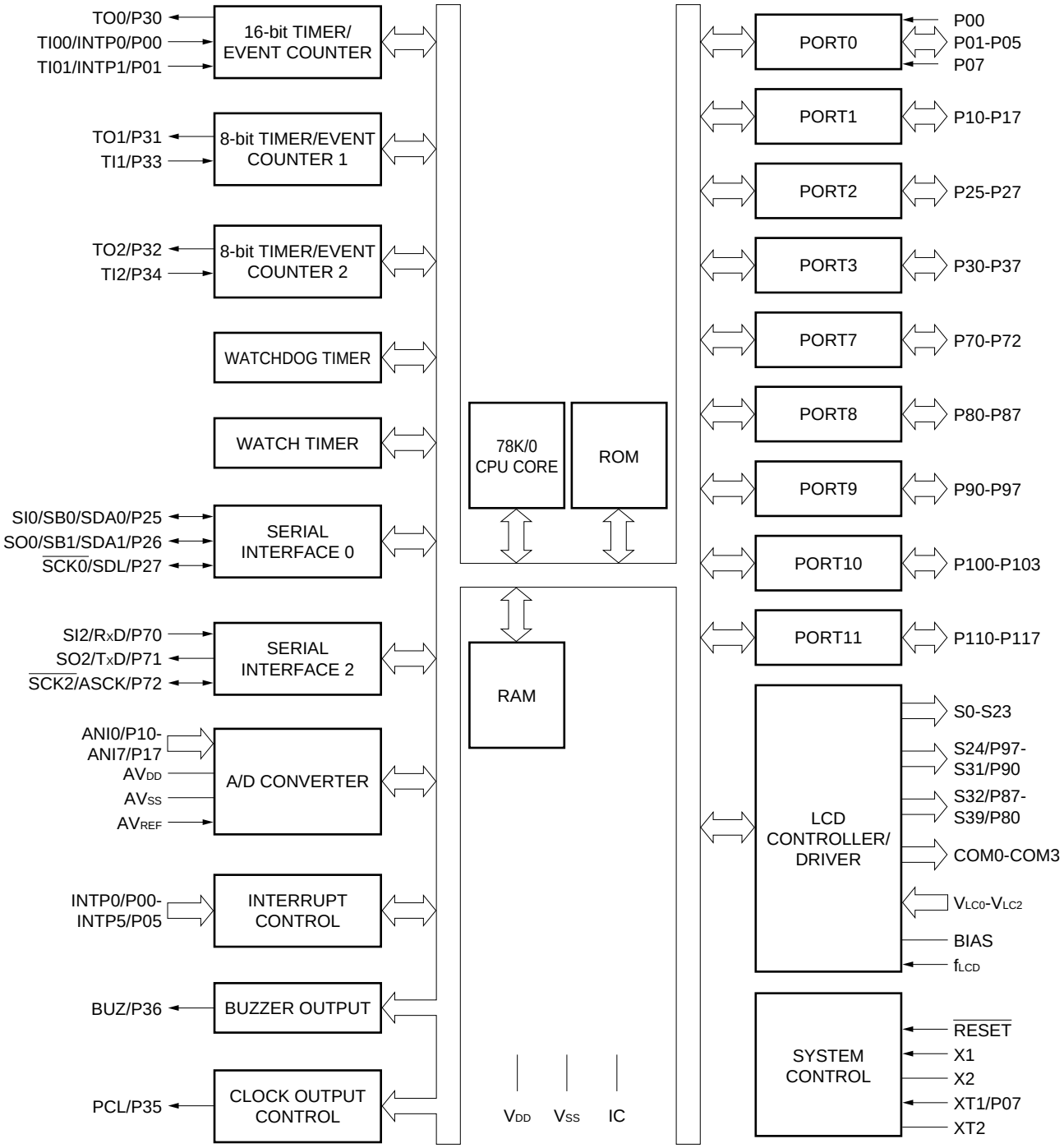
Table 1-3. Functional Outline of μPD78064 Subseries

Item / Part Number		μPD78062	μPD78063	μPD78064	μPD78P064
Internal memory	ROM	Mask ROM			PROM
		16K bytes	24K bytes	32K bytes	32K bytes[Note 1]
	High-speed RAM	512 bytes	1024 bytes		1024 bytes[Note 1]
	LCD display RAM	40 × 4 bits			
Memory space		64K bytes			
General-purpose register		8 bits × 8 × 4 banks			
Minimum instruction execution time	With main system clock	0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs/12.8 μs (at 5.0 MHz)			
	With subsystem clock	122 μs (at 32.768 kHz)			
Instruction set		• 16-bit operation • Multiplication/division (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, Boolean operation) • BCD adjustment, etc.			
I/O port (including pins multiplexed with segment signal output)		• Total : 57 • CMOS input : 2 • CMOS I/O : 55			
A/D converter		8-bit resolution × 8 channels			
LCD controller/driver		• Segment signal output : 40 max. • Common signal output : 4 max. • Bias : 1/2 or 1/3 bias selectable			
Serial interface		• 3-wire serial I/O/SBI/2-wire serial I/O mode selectable : 1 channel • 3-wire serial I/O/UART mode selectable : 1 channel			
Timer		• 16-bit timer/event counter : 1 channel • 8-bit timer/event counter : 2 channels • Watch timer : 1 channel • Watchdog timer : 1 channel			
Timer output		3 (14-bit PWM output: 1)			
Clock output		19.5 kHz, 39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5.0 MHz (with main system clock of 5.0 MHz), 32.768 kHz (with subsystem clock of 32.768 kHz)			
Buzzer output		1.2 kHz, 2.4 kHz, 4.9 kHz, 9.8 kHz (with main system clock of 5.0 MHz)			
Vectored interrupt source	Maskable	Internal: 12, external: 6			
	Non-maskable	Internal: 1			
	Software	1			
Test input		Internal: 1, external: 1			
Supply voltage		V_{DD} = 2.0 to 6.0 V			
★ Package		• 100-pin plastic QFP (fine pitch) (14 × 14 mm, resin thickness 1.45 mm) • 100-pin plastic LQFP (fine pitch) (14 × 14 mm, resin thickness 1.4 mm) • 100-pin plastic QFP (14 × 20 mm) • 100-pin ceramic WQFN (14 × 20 mm)[Note 2] (μPD78P064 only)			

Notes 1. The capacities of the internal PROM and internal high-speed RAM can be changed by using a memory size select register (IMS).

2. Under development

Figure 1-4. Block Diagram of μPD78064Y Subseries

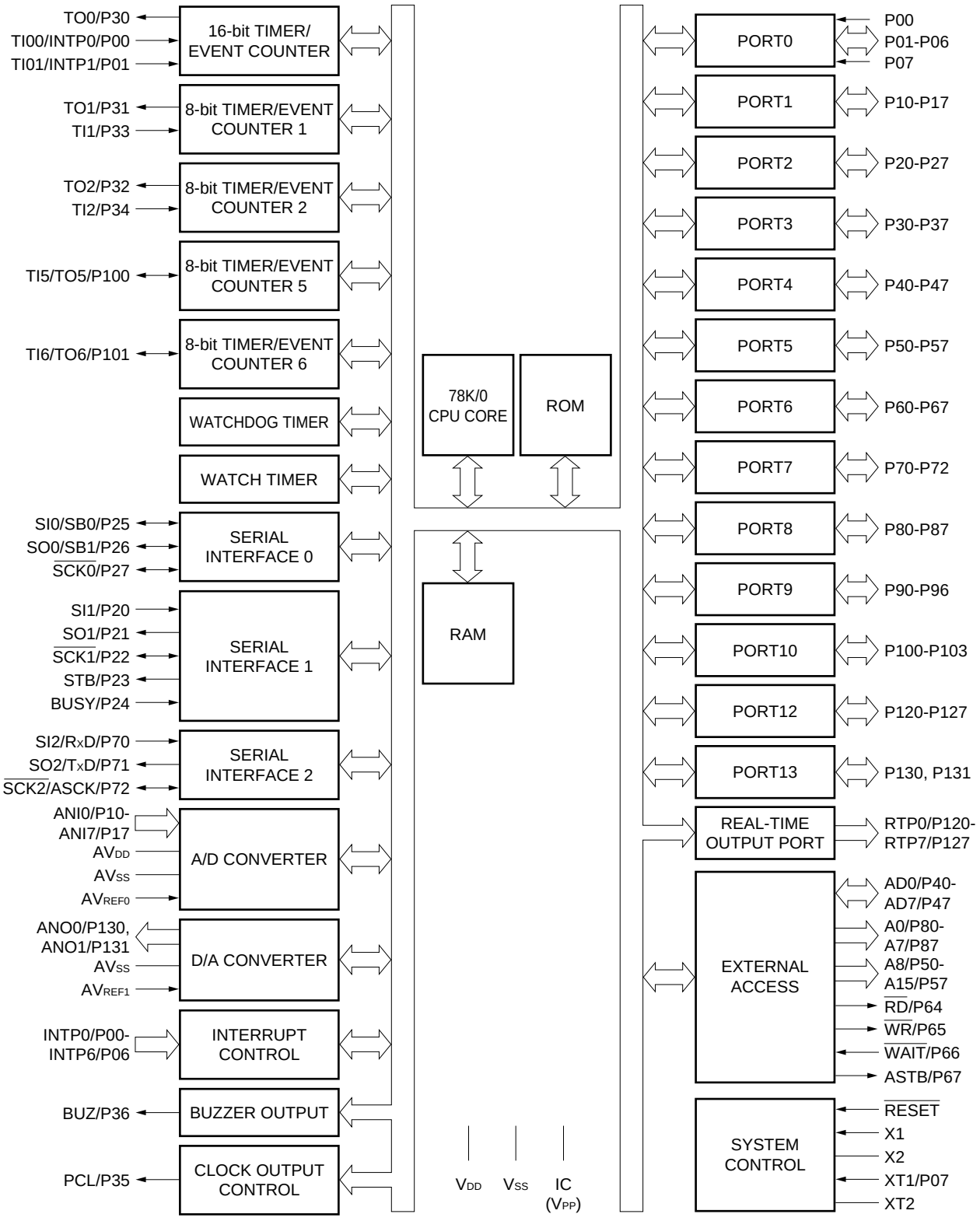


Remark The internal ROM and RAM capacities differ depending on the model.

Table 1-4. Functional Outline of μPD78064Y Subseries

Item / Part Number		μPD78062Y	μPD78063Y	μPD78064Y
Internal memory	ROM	Mask ROM		
		16K bytes	24K bytes	32K bytes
	High-speed RAM	512 bytes	1024 bytes	
	LCD display RAM	40 × 4 bits		
Memory space		64K bytes		
General-purpose register		8 bits × 8 × 4 banks		
Minimum instruction execution time	With main system clock	0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs/12.8 μs (at 5.0 MHz)		
	With subsystem clock	122 μs (at 32.768 kHz)		
Instruction set		• 16-bit operation • Multiplication/division (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, Boolean operation) • BCD adjustment, etc.		
I/O port (including pins multiplexed with segment signal output)		• Total : 57 • CMOS input : 2 • CMOS I/O : 55		
A/D converter		8-bit resolution × 8 channels		
LCD controller/driver		• Segment signal output : 40 max. • Common signal output : 4 max. • Bias : 1/2 or 1/3 bias selectable		
Serial interface		• 3-wire serial I/O/2-wire serial I/O/I^2C bus mode selectable : 1 channel • 3-wire serial I/O/UART mode selectable : 1 channel		
Timer		• 16-bit timer/event counter : 1 channel • 8-bit timer/event counter : 2 channels • Watch timer : 1 channel • Watchdog timer : 1 channel		
Timer output		3 (14-bit PWM output: 1)		
Clock output		19.5 kHz, 39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5.0 MHz (with main system clock of 5.0 MHz), 32.768 kHz (with subsystem clock of 32.768 kHz)		
Buzzer output		1.2 kHz, 2.4 kHz, 4.9 kHz, 9.8 kHz (with main system clock of 5.0 MHz)		
Vectored interrupt source	Maskable	Internal: 12, external: 6		
	Non-maskable	Internal: 1		
	Software	1		
Test input		Internal: 1, external: 1		
Supply voltage		V_{DD} = 2.0 to 6.0 V		
★ Package		• 100-pin plastic QFP (fine pitch) (14 × 14 mm, resin thickness 1.45 mm) • 100-pin plastic LQFP (fine pitch) (14 × 14 mm, resin thickness 1.4 mm) • 100-pin plastic QFP (14 × 20 mm)		

Figure 1-5. Block Diagram of μPD78078 Subseries



Remarks 1. The internal ROM capacitiy differs depending on the model.

2. ( ): μPD78P078

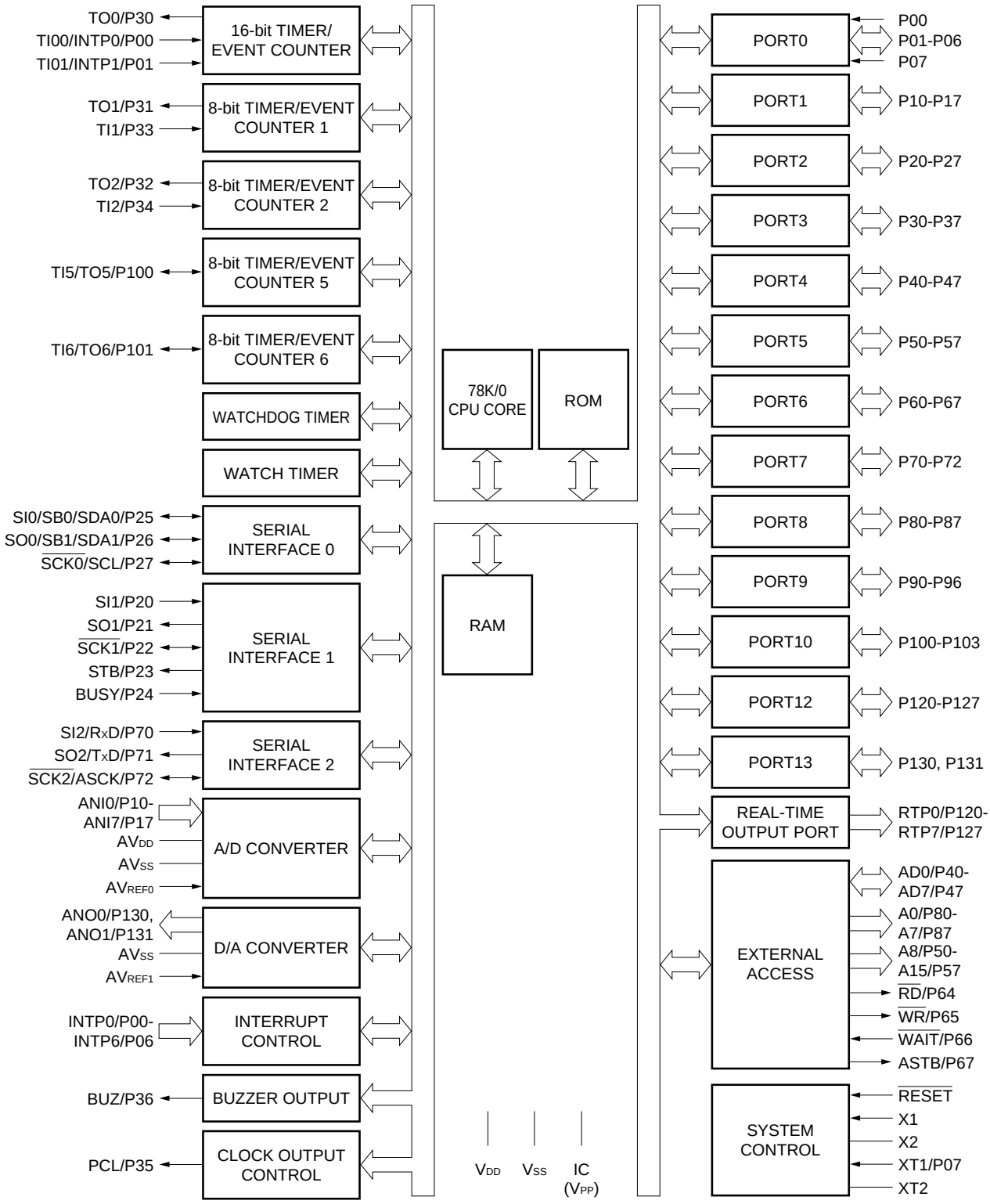
Table 1-5. Functional Outline of μPD78078 Subseries

Item / Part Number		μPD78076	μPD78078	μPD78P078
Internal memory	ROM	Mask ROM		PROM
		48K bytes	60K bytes	60K bytes[Note 1]
	High-speed RAM	1024 bytes		
	Buffer RAM	32 bytes		
	Expansion RAM	1024 bytes		
Memory space		64K bytes		
General-purpose register		8 bits × 8 × 4 banks		
Minimum instruction execution timon	With main system clock	0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs/12.8 μs (at 5.0 MHz)		
	With subsystem clock	122 μs (at 32.768 kHz)		
Instruction set		• 16-bit operation • Multiplication/division (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, Boolean operation) • BCD adjustment, etc.		
I/O port		• Total : 88 • CMOS input : 2 • CMOS I/O : 78 • N-ch open-drain I/O : 8		
A/D converter		8-bit resolution × 8 channels		
D/A converter		8-bit resolution × 2 channels		
Serial interface		• 3-wire serial I/O/SBI/2-wire serial I/O mode selectable : 1 channel • 3-wire serial I/O mode (with function to automatically transfer/receive up to 32 bytes) : 1 channel • 3-wire serial I/O/UART mode selectable : 1 channel		
Timer		• 16-bit timer/event counter : 1 channel • 8-bit timer/event counter : 4 channels • Watch timer : 1 channel • Watchdog timer : 1 channel		
Timer output		5 (14-bit PWM output: 1, 8-bit PWM output: 2)		
Clock output		19.5 kHz, 39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5.0 MHz (with main system clock of 5.0 MHz), 32.768 kHz (with subsystem clock of 32.768 kHz)		
Vectored interrupt source	Maskable	Internal: 15, external: 7		
	Non-maskable	Internal: 1		
	Software	1		
Test input		Internal: 1, external: 1		
Supply voltage		V_{DD} = 1.8 to 5.5 V		
Package		• 100-pin plastic QFP (fine pitch) (14 × 14 mm, resin thickness 1.45 mm) • 100-pin plastic LQFP (fine pitch) (14 × 14 mm, resin thickness 1.4 mm)[Note 2] • 100-pin plastic QFP (14 × 20 mm, resin thickness 2.7 mm) • 100-pin ceramic WQFN (14 × 20 mm) (μPD78P078 only)		

Notes 1. The internal ROM capacity can be changed by using a memory size select register (IMS).

2. Under planning

Figure 1-6. Block Diagram of μPD78078Y Subseries



Remarks 1. The internal ROM capacity differs depending on the model.

2. ( ): μPD78P078Y

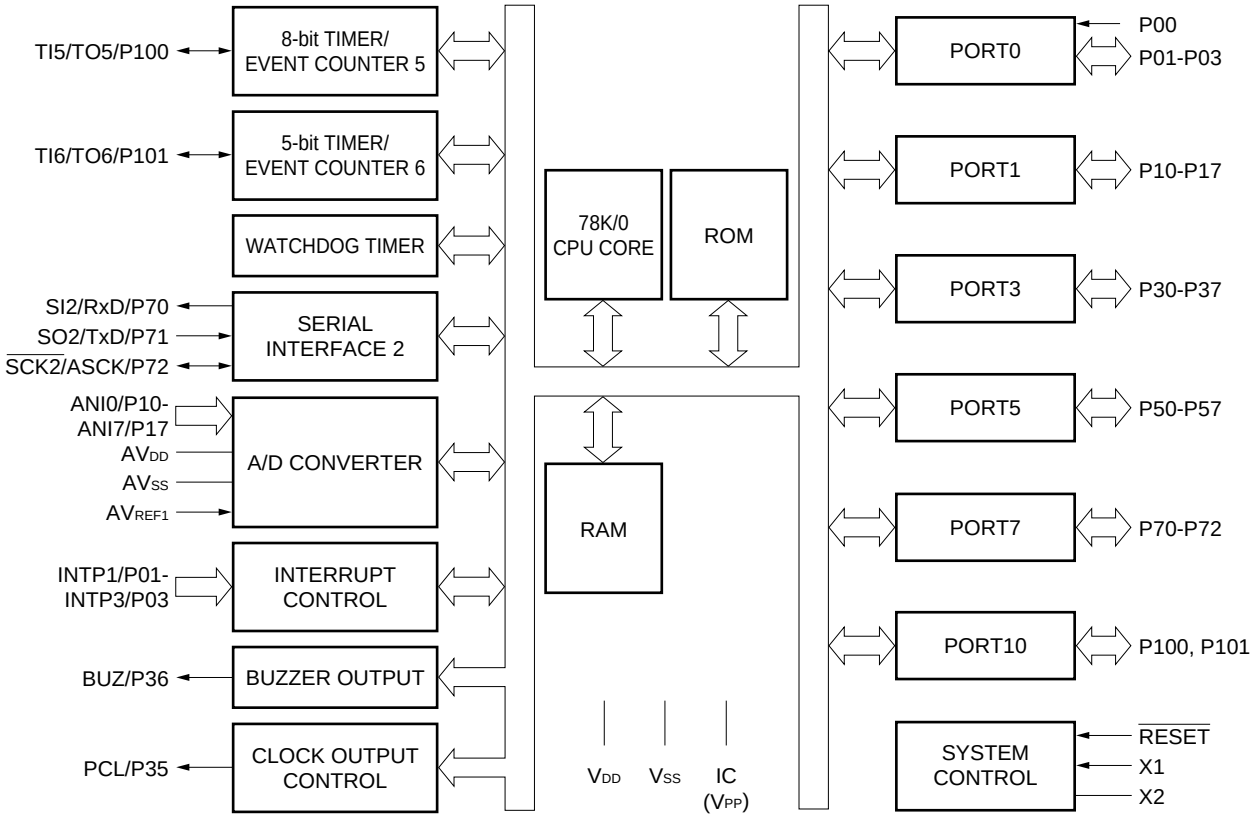
Table 1-6. Functional Outline of μPD78078Y Subseries

Item / Part Number		μPD78076Y	μPD78078Y	μPD78P078Y
Internal memory	ROM	Mask ROM		PROM
		48K bytes	60K bytes	60K bytes[Note 1]
	High-speed RAM	1024 bytes		
	Buffer RAM	32 bytes		
	Expansion RAM	1024 bytes		
Memory space		64K bytes		
General-purpose register		8 bits × 8 × 4 banks		
Minimum instruction execution time	With main system clock	0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs/12.8 μs (at 5.0 MHz)		
	With subsystem clock	122 μs (at 32.768 kHz)		
Instruction set		• 16-bit operation • Multiplication/division (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, Boolean operation) • BCD adjustment, etc.		
I/O port		• Total : 88 • CMOS input : 2 • CMOS I/O : 78 • N-ch open-drain I/O : 8		
A/D converter		8-bit resolution × 8 channels		
D/A converter		8-bit resolution × 2 channels		
Serial interface		• 3-wire serial I/O/2-wire serial I/O/I^2C bus mode selectable : 1 channel • 3-wire serial I/O mode (with function to automatically transfer/receive up to 32 bytes) : 1 channel • 3-wire serial I/O/UART mode selectable : 1 channel		
Timer		• 16-bit timer/event counter : 1 channel • 8-bit timer/event counter : 4 channels • Watch timer : 1 channel • Watchdog timer : 1 channel		
Timer output		5 (14-bit PWM output: 1, 8-bit PWM output: 2)		
Clock output		19.5 kHz, 39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5.0 MHz (with main system clock of 5.0 MHz), 32.768 kHz (with subsystem clock of 32.768 kHz)		
Buzzer output		1.2 kHz, 2.4 kHz, 4.9 kHz, 9.8 kHz (with main system clock of 5.0 MHz)		
Vectored interrupt source	Maskable	Internal: 15, external: 7		
	Non-maskable	Internal: 1		
	Software	1		
Test input		Internal: 1, external: 1		
Supply voltage		V_{DD} = 1.8 to 5.5 V		
Package		• 100-pin plastic QFP (14 × 20 mm, resin thickness 2.7 mm) • 100-pin plastic LQFP (fine pitch) (14 × 14 mm, resin thickness 1.4 mm)[Note 2] • 100-pin ceramic WQFN (14 × 20 mm) (μPD78P078Y only)		

Notes 1. The internal ROM capacity can be changed by using a memory size select register (IMS).

2. Under development

Figure 1-7. Block Diagram of μPD78083 Subseries



Remarks 1. The internal ROM and RAM capacities differ depending on the model.

2. ( ): μPD78P083

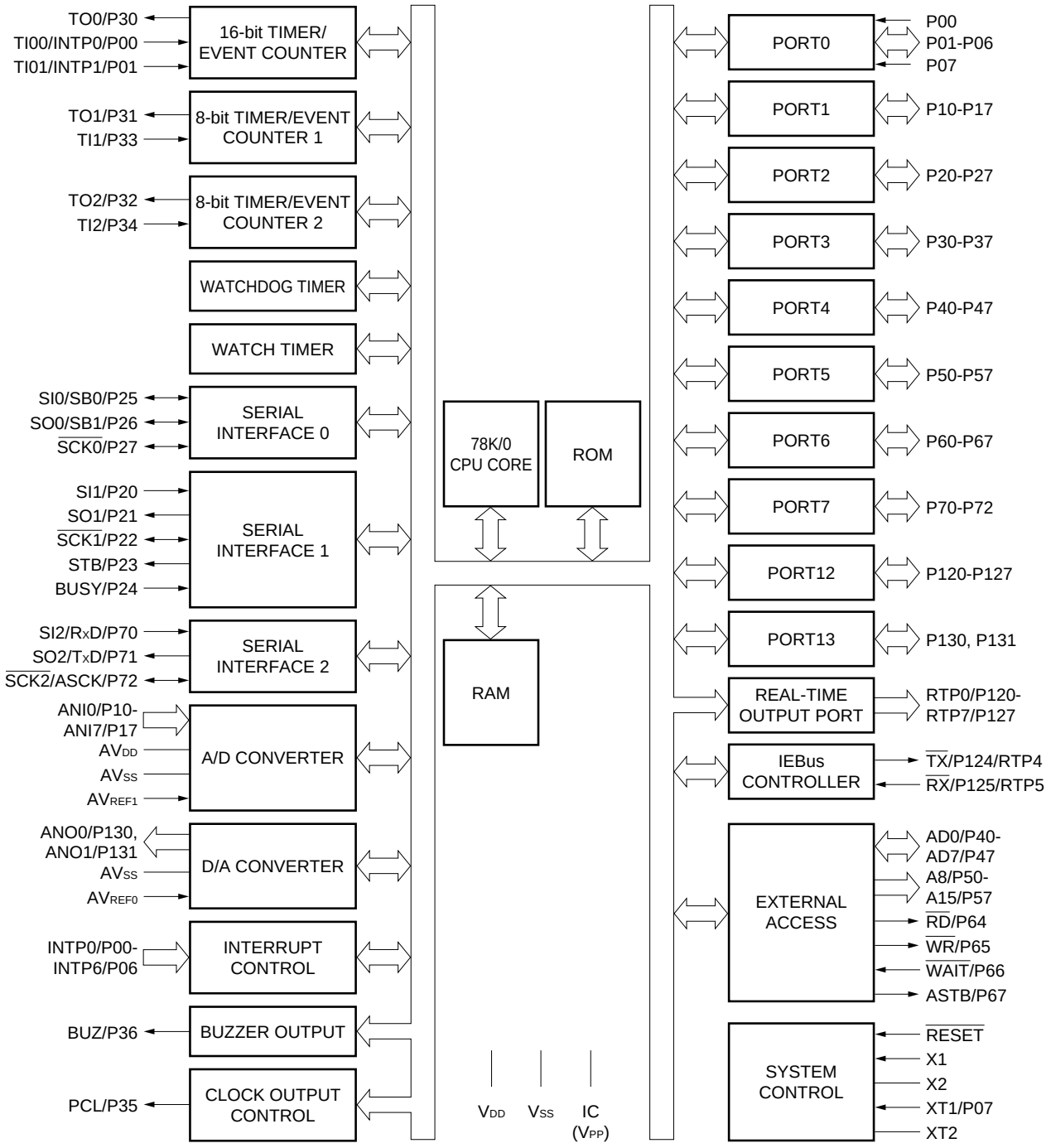
Table 1-7. Functional Outline of μPD78083 Subseries

Item / Part Number		μPD78081	μPD78082	μPD78P083
Internal memory	ROM	Mask ROM		PROM
		8K bytes	16K bytes	24K bytes[Note 1]
	High-speed RAM	256 bytes	384 bytes	512 bytes[Note 1]
Memory space		64K bytes		
General-purpose register		8 bits × 8 × 4 banks		
Minimum instruction execution time		0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs/12.8 μs (with main system clock of 5.0 MHz)		
Instruction set		• 16-bit operation • Multiplication/division (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, Boolean operation) • BCD adjustment, etc.		
I/O port		• Total : 33 • CMOS input : 1 • CMOS I/O : 32		
A/D converter		8-bit resolution × 8 channels		
Serial interface		3-wire serial I/O/UART mode selectable: 1 channel		
Timer		• 8-bit timer/event counter : 2 channels • Watchdog timer : 1 channel		
Timer output		2 (8-bit PWM output)		
Clock output		19.5 kHz, 39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5.0 MHz (with main system clock of 5.0 MHz)		
Buzzer output		1.2 kHz, 2.4 kHz, 4.9 kHz, 9.8 kHz (with main system clock of 5.0 MHz)		
Vectored interrupt source	Maskable	Internal: 8, external: 3		
	Non-maskable	Internal: 1		
	Software	1		
Supply voltage		V_{DD} = 1.8 to 5.5 V[Note 2]		
Package		• 42-pin plastic shrink DIP (600 mil) • 42-pin ceramic shrink DIP (with window) (600 mil) (μPD78P083 only) • 44-pin plastic QFP (10 × 10 mm)		

Notes 1. The capacities of the internal PROM and internal-high-speed RAM can be changed by using a memory size select register. (IMS)

★ **2.** The supply voltage (V_{DD}) of the μPD78081(A2) is 4.5 to 5.5 V.

Figure 1-8. Block Diagram of μPD78098 Subseries



Remarks 1. The internal ROM and RAM capacities differ depending on the model.

2. ( ): μPD78P098A

Table 1-8. Functional Outline of μPD78098 Subseries (1/2)

Item / Part Number		μPD78094	μPD78095	μPD78096	μPD78098A[Note 1]	μPD78P098A[Note 1, 2]
Internal memory	ROM	Mask ROM				PROM
		32K bytes	40K bytes	48K bytes	60K bytes	60K bytes[Note 3]
	High-speed RAM	1024 bytes				
	Buffer RAM	32 bytes				
	Expansion RAM	None			2048 bytes	2048 bytes[Note 4]
Memory space		64K bytes				
General-purpose register		8 bits × 8 × 4 banks				
Minimum instruction execution time	With main system clock	0.5 μs/1.0 μs/2.0 μs/4.0 μs/8.0 μs/16.0 (at 6.0 MHz)				
	With subsystem clock	122 μs (at 32.768 kHz)				
Instruction set		• 16-bit operation • Multiplication/division (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, Boolean operation) • BCD adjustment, etc.				
I/O port		• Total : 69 • CMOS input : 2 • CMOS I/O : 63 • N-ch open-drain I/O: 4				
IEBus controller		Effective transfer rate: 3.9 kbps/17 kbps/26 kbps				
A/D converter		8-bit resolution × 8 channels				
D/A converter		8-bit resolution × 2 channels				
Serial interface		• 3-wire serial I/O/SBI/2-wire serial I/O mode selectable : 1 channel • 3-wire serial I/O mode (with function to automatically transfer/receive up to 32 bytes) : 1 channel • 3-wire serial I/O/UART mode selectable : 1 channel				
Timer		• 16-bit timer/event counter : 1 channel • 8-bit timer/event counter : 2 channels • Watch timer : 1 channel • Watchdog timer : 1 channel				
Timer output		3 (14-bit PWM output: 1)				
Clock output		15.6 kHz, 31.3 kHz, 62.5 kHz, 125 kHz, 250 kHz, 500 kHz, 1.0 MHz, 2.0 MHz, 4.0 MHz (with main system clock of 6.0 MHz), 32.768 kHz (with subsystem clock of 32.768 kHz)				

Notes 1. Under development

2. The μPD78P098A is the PROM model of the μPD78094, 78095, 78096, and 78098A.

3. The internal PROM capacity can be changed by using a memory size select register (IMS).

4. The internal expansion RAM can be changed by using an internal expansion RAM size select register (IXS).

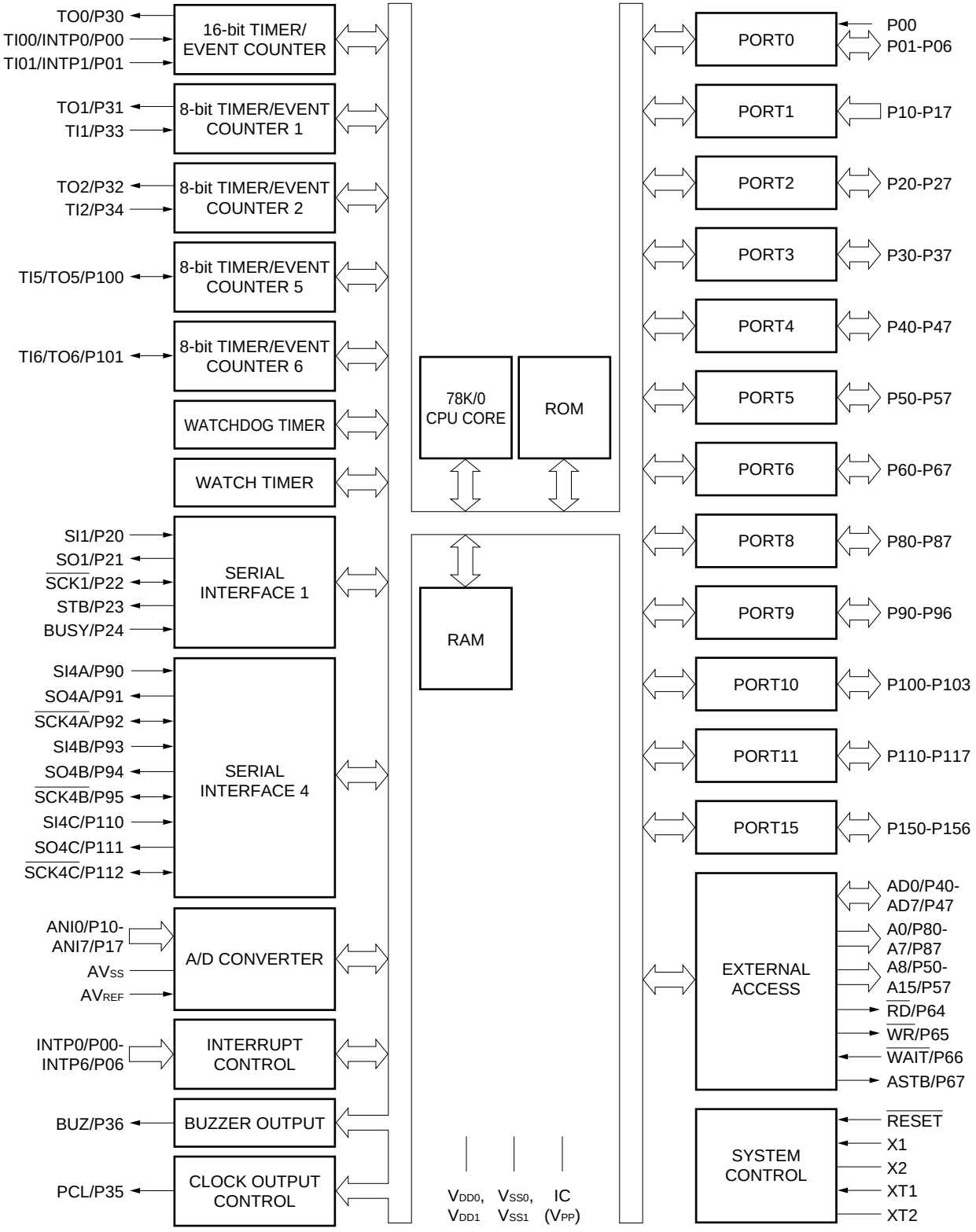
Table 1-8. Functional Outline of μPD78098 Subseries (2/2)

Item / Part Number		μPD78094	μPD78095	μPD78096	μPD78098A[Note 1]	μPD78P098A[Note 1, 2]
Buzzer output		977 Hz, 1.95 kHz, 3.9 kHz, 7.8 kHz (with main system clock of 6.0 MHz)				
Vectored interrupt source	Maskable	Internal: 14, external: 7				
	Non-maskable	Internal: 1				
	Software	1				
Test input		Internal: 1, external: 1				
Supply voltage		V_{DD} = 2.7 to 6.0 V				
Package		• 80-pin plastic QFP (14 × 14 mm) • 80-pin ceramic WQFN (14 × 14 mm)[Note 1] (μPD78P098A only)				

Notes 1. Under development

2. The μPD78P098A is a PROM model of the μPD78094, 78095, 78096, and 78098A.

★ **Figure 1-9. Block Diagram of μPD780018 Subseries**



Remarks 1. The internal ROM capacity differs depending on the model.

2. ( ): μPD78P0018

★ **Table 1-9. Functional Outline of μPD780018 Subseries (1/2)**

Item / Part Number		μPD780016	μPD780018	μPD78P0018
Internal memory	ROM	Mask ROM		PROM
		48K bytes	60K bytes	60K bytes[Note]
	High-speed RAM	1024 bytes		
	Buffer RAM	32 bytes		
	Expansion RAM	1024 bytes		
Memory space		64K bytes		
General-purpose register		8 bits × 8 × 4 banks		
Minimum instruction execution time	With main system clock	0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs (at 5.0 MHz)		
	With subsystem clock	122 μs (at 32.768 kHz)		
Instruction set		• 16-bit operation • Multiplication/division (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, Boolean operation) • BCD adjustment, etc.		
I/O port		• Total : 88 • CMOS input : 9 • CMOS I/O : 79		
A/D converter		8-bit resolution × 8 channels		
Serial interface		• 3-wire serial I/O mode (with automatical transfer/reception function) : 1 channel • 3-wire serial I/O mode selectable (with time-division transfer function) : 1 channel		
Timer		• 16-bit timer/event counter : 1 channel • 8-bit timer/event counter : 4 channels • Watch timer : 1 channel • Watchdog timer : 1 channel		
Timer output		5 (14-bit PWM output: 1, 8-bit PWM output: 2)		
Clock output		39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5.0 MHz (with main system clock of 5.0 MHz), 32.768 kHz (with subsystem clock of 32.768 kHz)		

Note The internal ROM capacity can be changed by using a memory size select register. (IMS)

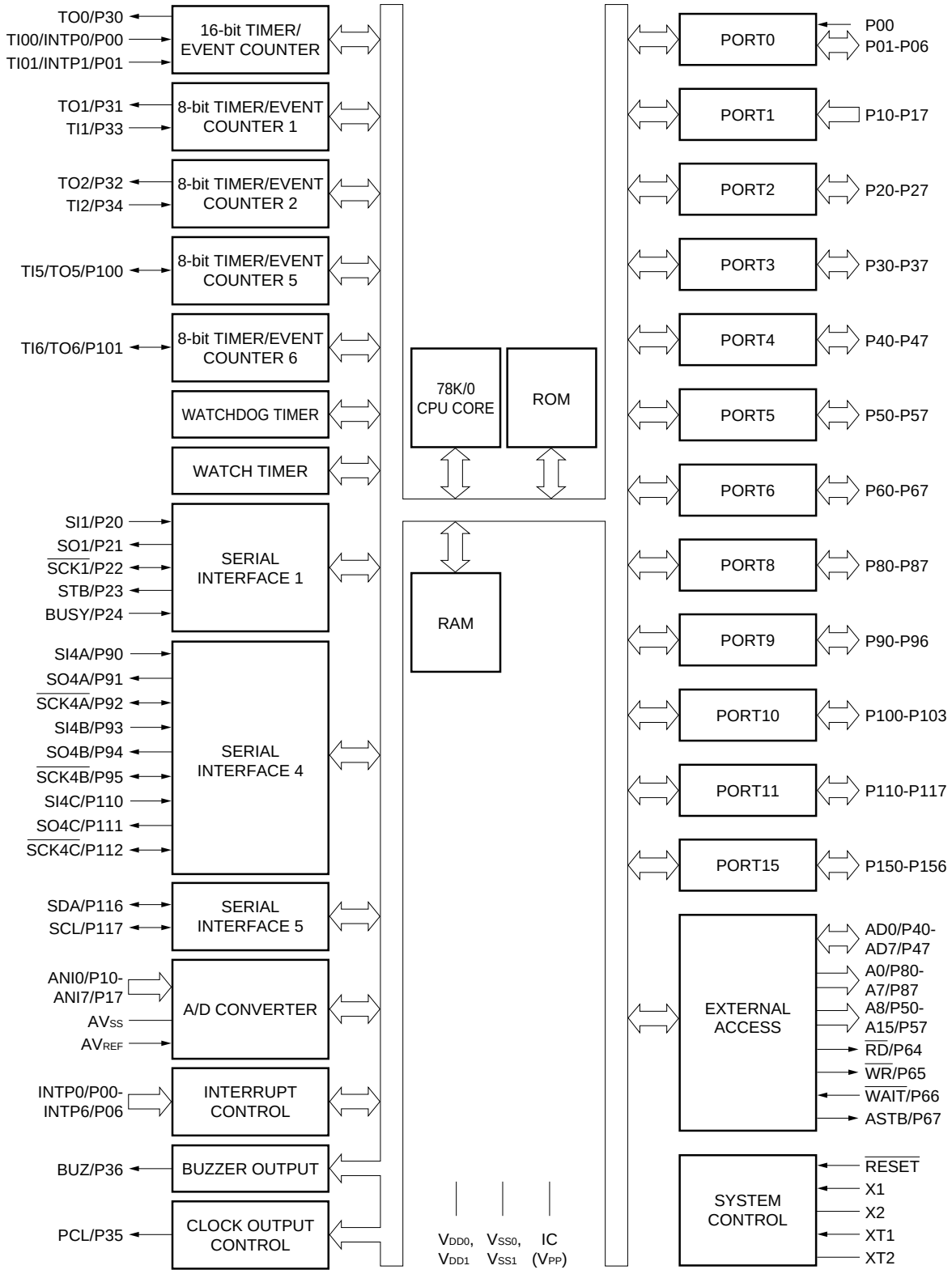
Caution The μPD780018 subseries is under planning.

Table 1-9. Functional Outline of μPD780018 Subseries (2/2)

Item / Part Number		μPD780016	μPD780018	μPD78P0018
Buzzer output		2.4 kHz, 4.9 kHz, 9.8 kHz (with main system clock of 5.0 MHz)		
Vectored interrupt source	Maskable	Internal: 12, external: 7		
	Non-maskable	Internal: 1		
	Software	1		
Test input		Internal: 1, external: 1		
Supply voltage		V_{DD} = 2.7 to 5.5 V		
Operating temperature		T_A = –40 to +85 °C		
Package		• 100-pin plastic QFP (14 × 20 mm) • 100-pin ceramic WQFN (14 × 20 mm) (μPD78P0018 only)		

Caution The μPD780018 subseries is under planning.

★ **Figure 1-10. Block Diagram of μPD780018Y Subseries**



Remarks 1. The internal ROM capacity differs depending on the model.

2. ( ): μPD78P0018Y

★ **Table 1-10. Functional Outline of μPD780018Y Subseries (1/2)**

Item / Part Number		μPD780016Y	μPD780018Y	μPD78P018Y
Internal memory	ROM	Mask ROM		PROM
		48K bytes	60K bytes	60K bytes[Note]
	High-speed RAM	1024 bytes		
	Buffer RAM	32 bytes		
	Expansion RAM	1024 bytes		
Memory space		64K bytes		
General-purpose register		8 bits × 8 × 4 banks		
Minimum instruction execution time	With main system clock	0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs (at 5.0 MHz)		
	With subsystem clock	122 μs (at 32.768 kHz)		
Instruction set		• 16-bit operation • Multiplication/division (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, Boolean operation) • BCD adjustment, etc.		
I/O port		• Total : 88 • CMOS input : 9 • CMOS I/O : 79		
A/D converter		8-bit resolution × 8 channels		
Serial interface		• 3-wire serial I/O mode (with automatical transfer/reception function) : 1 channel • 3-wire serial I/O mode selectable (with time-division transfer function) : 1 channel • I^2C bus mode (multi-master compatible) : 1 channel		
Timer		• 16-bit timer/event counter : 1 channel • 8-bit timer/event counter : 4 channels • Watch timer : 1 channel • Watchdog timer : 1 channel		
Timer output		5 (14-bit PWM output: 1, 8-bit PWM output: 2)		
Clock output		39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5.0 MHz (with main system clock of 5.0 MHz), 32.768 kHz (with subsystem clock of 32.768 kHz)		

Note The internal PROM capacity can be changed by using a memory size select register. (IMS)

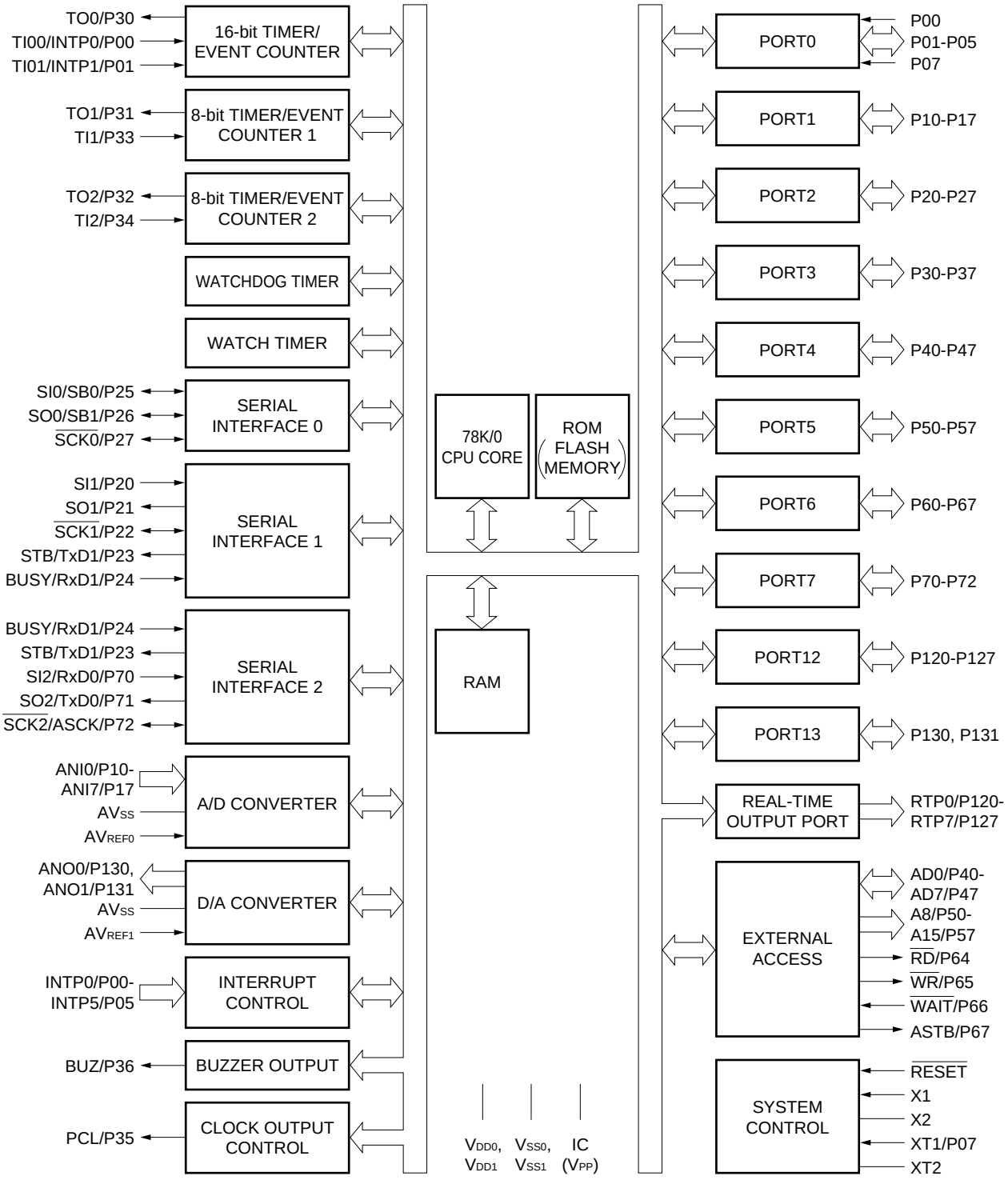
Caution The μPD780018Y subseries is under development.

Table 1-10. Functional Outline of μPD78P0018Y Subseries (2/2)

Item / Part Number		μPD780016Y	μPD780018Y	μPD78P018Y
Buzzer output		2.4 kHz, 4.9 kHz, 9.8 kHz (with main system clock of 5.0 MHz)		
Vectored interrupt source	Maskable	Internal: 12, external: 7		
	Non-maskable	Internal: 1		
	Software	1		
Test input		Internal: 1, external: 1		
Supply voltage		V_{DD} = 2.7 to 5.5 V		
Operating temperature		T_A = –40 to +85 °C		
Package		• 100-pin plastic QFP (14 × 20 mm) • 100-pin ceramic WQFN (14 × 20 mm) (μPD78P0018Y only)		

Caution The μPD780018Y subseries is under development.

★ **Figure 1-11. Block Diagram of μPD780058 Subseries**



Remarks 1. The capacities of the internal ROM and RAM differ depending on the model.

2. ( ): μPD78F0058

★ **Table 1-11. Functional Outline of μPD780058 Subseries (1/2)**

Item / Part Number		μPD780053	μPD780054	μPD780055	μPD780056	μPD780058	μPD78F0058
Internal memory	ROM	Mask ROM					Flash memory
		24K bytes	32K bytes	40K bytes	48K bytes	60K bytes	60K bytes[Note 1]
	High-speed RAM	1024 bytes					
	Buffer RAM	32 bytes					
	Expansion RAM	None				1024 bytes	1024 bytes[Note 2]
Memory space		64K bytes					
General-purpose register		8 bits × 8 × 4 banks					
Minimum instruction execution time	With main system clock	0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs/12.8 μs (at 5.0 MHz)					
	With subsystem clock	122 μs (at 32.768 kHz)					
Instruction set		• 16-bit operation • Multiplication/division (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, Boolean operation) • BCD adjustment, etc.					
I/O port		• Total : 68 • CMOS input : 2 • CMOS I/O : 62 • N-ch open-drain I/O : 4					
A/D converter		8-bit resolution × 8 channels					
D/A converter		8-bit resolution × 2 channels					
Serial interface		• 3-wire serial I/O/SBI/2-wire serial I/O mode selectable : 1 channel • 3-wire serial I/O mode (with function to automatically transfer/receive up to 32 bytes) : 1 channel • 3-wire serial I/O/UART mode selectable (with time-division transfer function) : 1 channel					
Timer		• 16-bit timer/event counter : 1 channel • 8-bit timer/event counter : 2 channels • Watch timer : 1 channel • Watchdog timer : 1 channel					
Timer output		3 (14-bit PWM output: 1)					
Clock output		19.5 kHz, 39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5.0 MHz (with main system clock of 5.0 MHz), 32.768 kHz (with subsystem clock of 32.768 kHz)					

Notes 1. The capacities of the flash memory can be changed by using a memory size select register (IMS).

2. The internal expansion RAM capacity can be changed by using an internal expansion RAM size select register (IXS).

Caution The μPD780058 subseries is under development.

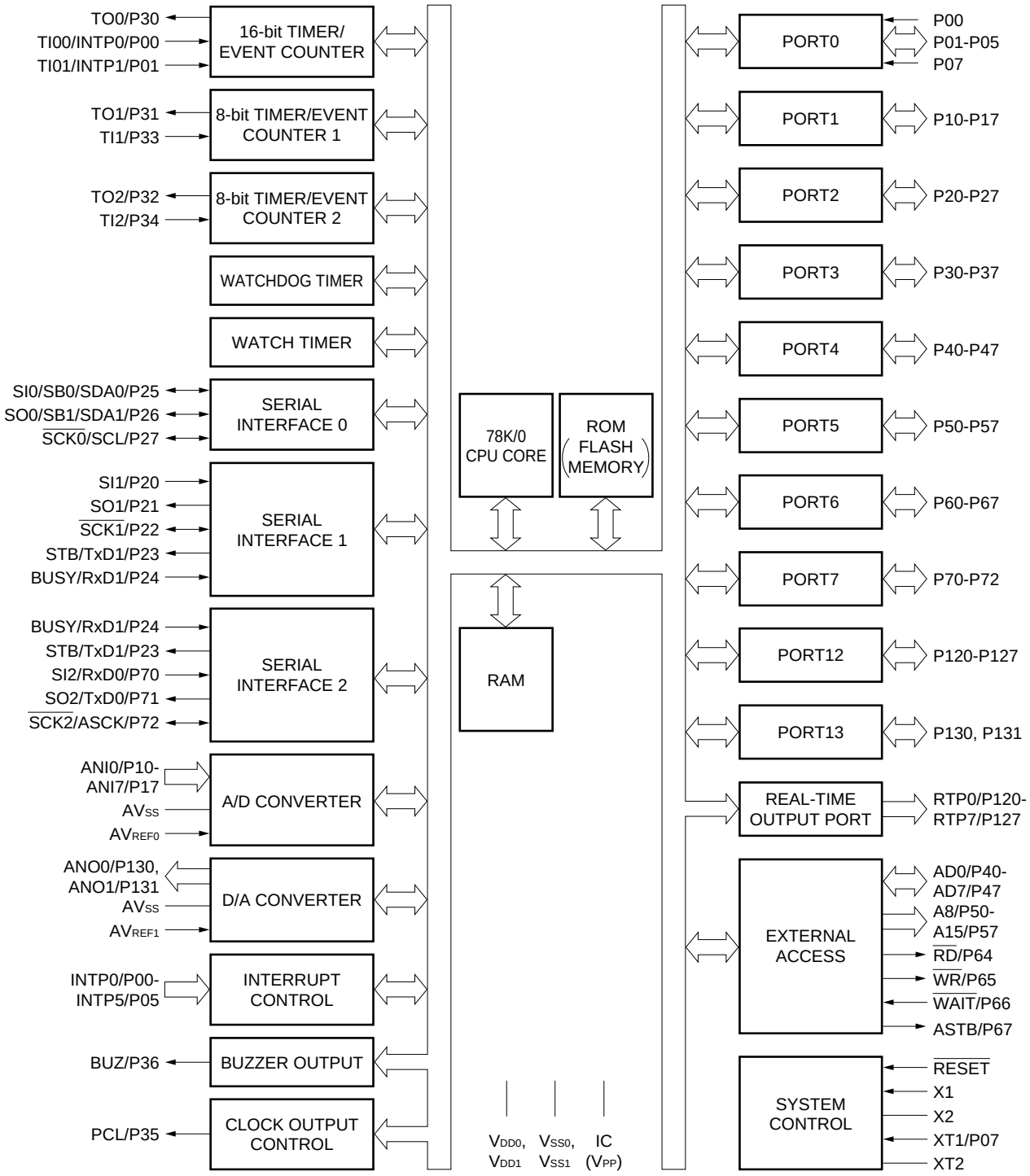
★ **Table 1-11. Functional Outline of μPD780058 Subseries (2/2)**

Item / Part Number		μPD780053	μPD780054	μPD780055	μPD780056	μPD780058	μPD78F0058
Buzzer output		1.2 kHz, 2.4 kHz, 4.9 kHz, 9.8 kHz (with main system clock of 5.0 MHz)					
Vectored interrupt source	Maskable	Internal: 13, external: 7					
	Non-maskable	Internal: 1					
	Software	1					
Test input		Internal: 1, external: 1					
Supply voltage		V_{DD} = 1.8 to 5.5 V					
Operating temperature		T_A = –40 to +85 °C					
Package		• 80-pin plastic QFP (14 × 14 mm, resin thickness 2.7 mm) • 80-pin plastic QFP (14 × 14 mm, resin thickness 1.4 mm)**Note** • 80-pin plastic TQFP (fine pitch) (12 × 12 mm)					

Note Under planning

Caution The μPD780058 subseries is under development.

★ **Figure 1-12. Block Diagram of μPD780058Y Subseries**



Remarks 1. The capacities of the internal ROM and RAM differ depending on the model.

2. ( ): μPD78F0058Y

★ **Table 1-12. Functional Outline of μPD780058Y Subseries (1/2)**

Item / Part Number		μPD780053Y	μPD780054Y	μPD780055Y	μPD780056Y	μPD780058Y	μPD78F0058Y
Internal memory	ROM	Mask ROM					Flash memory
		24K bytes	32K bytes	40K bytes	48K bytes	60K bytes	60K bytes[Note 1]
	High-speed RAM	1024 bytes					
	Buffer RAM	32 bytes					
	Expansion RAM	None				1024 bytes	1024 bytes[Note 2]
Memory space		64K bytes					
General-purpose register		8 bits × 8 × 4 banks					
Minimum instruction execution time	With main system clock	0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs/12.8 μs (at 5.0 MHz)					
	With subsystem clock	122 μs (at 32.768 kHz)					
Instruction set		• 16-bit operation • Multiplication/division (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, Boolean operation) • BCD adjustment, etc.					
I/O port		• Total : 68 • CMOS input : 2 • CMOS I/O : 62 • N-ch open-drain I/O : 4					
A/D converter		8-bit resolution × 8 channels					
D/A converter		8-bit resolution × 2 channels					
Serial interface		• 3-wire serial I/O/2-wire serial I/O/I^2C bus mode selectable : 1 channel • 3-wire serial I/O mode (with function to automatically transfer/receive up to 32 bytes) : 1 channel • 3-wire serial I/O/UART mode selectable (with time-division transfer function) : 1 channel					
Timer		• 16-bit timer/event counter : 1 channel • 8-bit timer/event counter : 2 channels • Watch timer : 1 channel • Watchdog timer : 1 channel					
Timer output		3 (14-bit PWM output: 1)					
Clock output		19.5 kHz, 39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5.0 MHz (with main system clock of 5.0 MHz), 32.768 kHz (with subsystem clock of 32.768 kHz)					

Notes 1. The capacities of the flash memory can be changed by using a memory size select register (IMS).

2. The internal expansion RAM capacity can be changed by using an internal expansion RAM size select register (IXS).

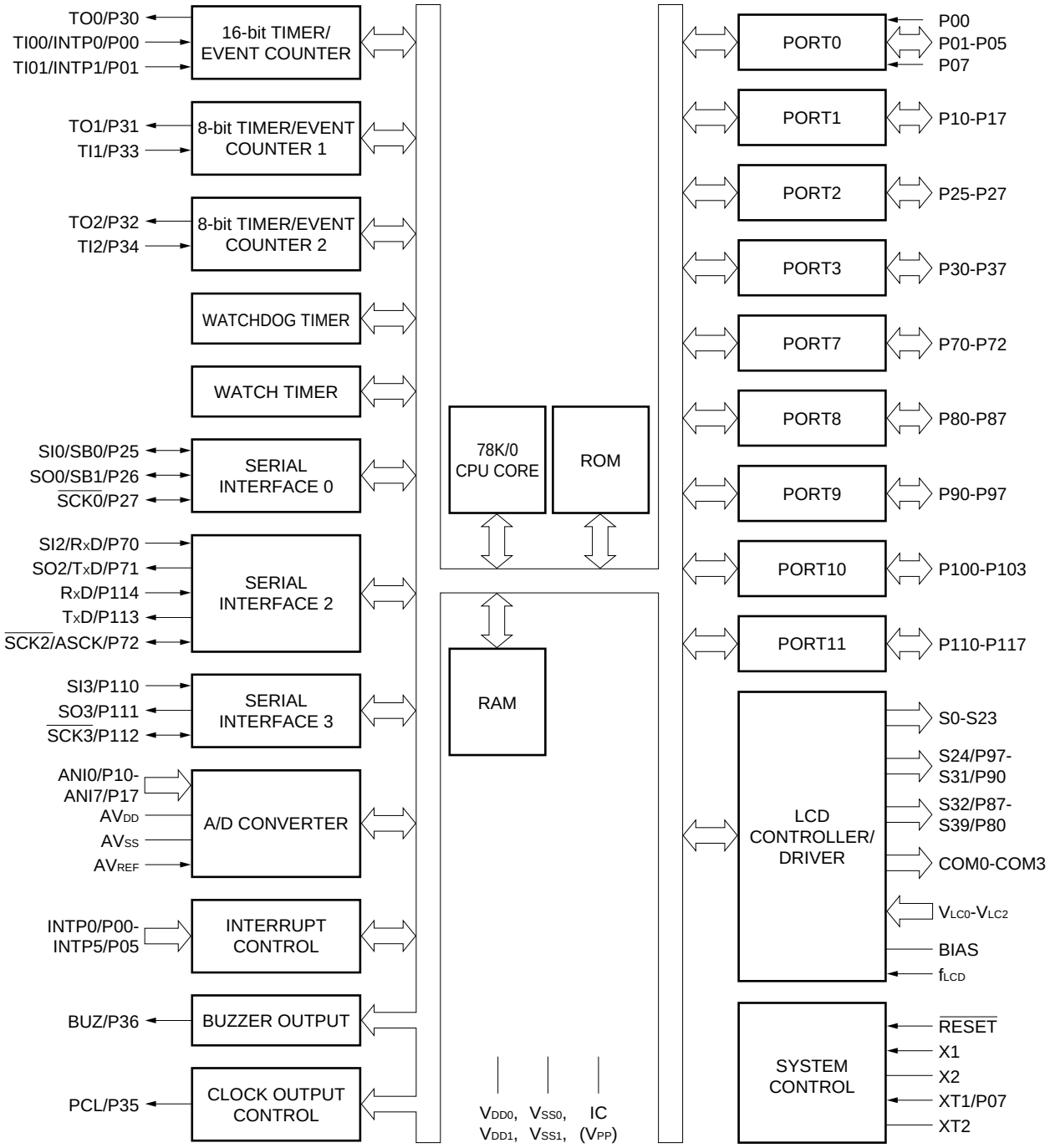
Caution The μPD780058Y subseries is under planning.

★ **Table 1-12. Functional Outline of μPD780058Y Subseries (2/2)**

Item / Part Number		μPD780053Y	μPD780054Y	μPD780055Y	μPD780056Y	μPD780058Y	μPD78F0058Y
Buzzer output		1.2 kHz, 2.4 kHz, 4.9 kHz, 9.8 kHz (with main system clock of 5.0 MHz)					
Vectored interrupt	Maskable	Internal: 13, external: 7					
	Non-maskable	Internal: 1					
	Software	1					
Test input		Internal: 1, external: 1					
Supply voltage		V_{DD} = 1.8 to 5.5 V					
Operating temperature		T_A = –40 to +85 °C					
Package		• 80-pin plastic QFP (14 × 14 mm, resin thickness 2.7 mm) • 80-pin plastic QFP (14 × 14 mm, resin thickness 1.4 mm) • 80-pin plastic TQFP (fine pitch) (12 × 12 mm)					

Caution The μPD780058Y subseries is under planning.

★ **Figure 1-13. Block Diagram of μPD780308 Subseries**



Remarks 1. The internal ROM capacity differs depending on the model.

2. ( ): μPD78P0308

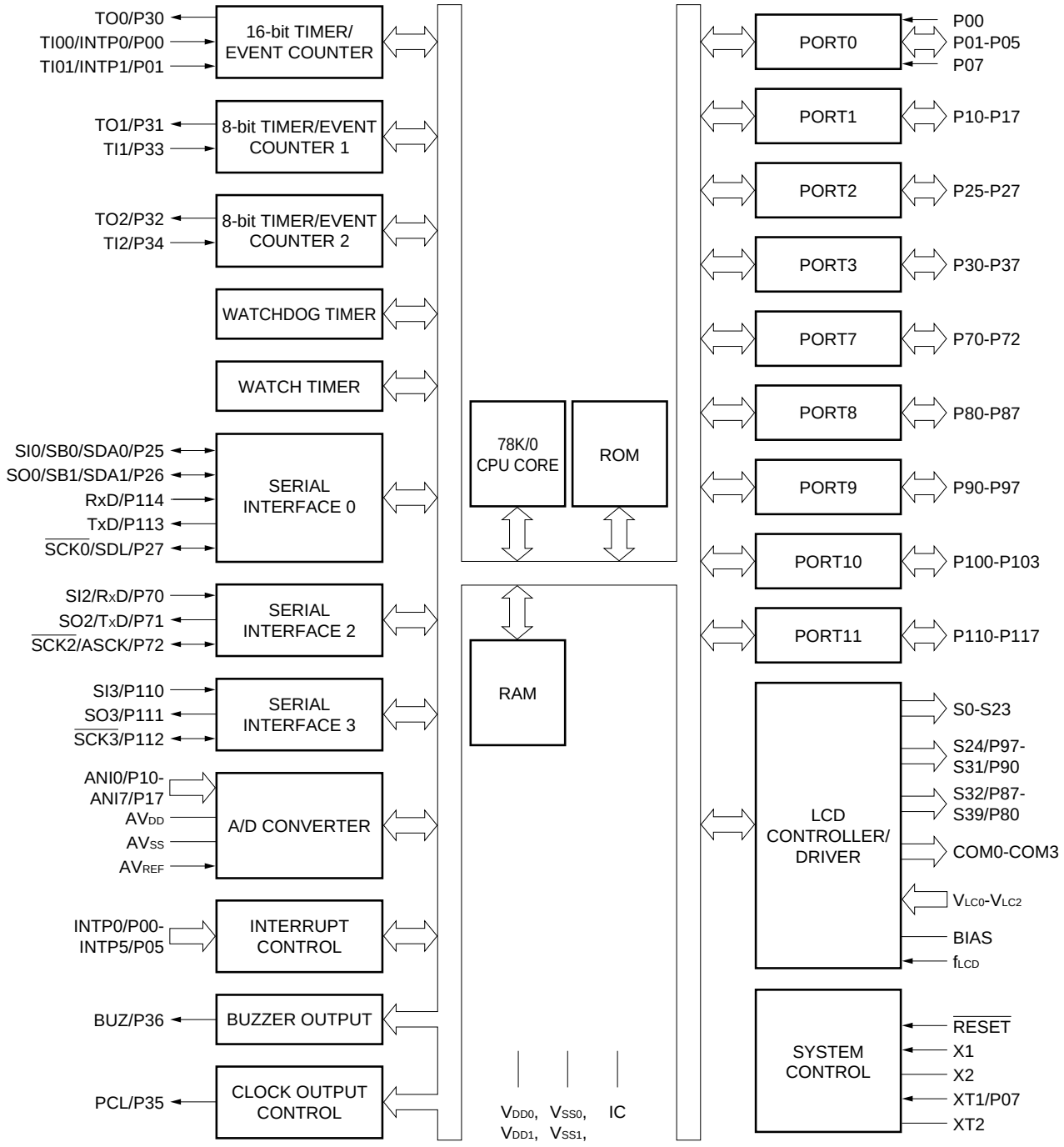
★ **Table 1-13. Functional Outline of μPD780308 Subseries**

Item / Part Number		μPD780306	μPD780308	μPD78P0308
Internal memory	ROM	Mask ROM		PROM
		48K bytes	60K bytes	60K bytes[Note]
	High-speed RAM	1024 bytes		
	Expansion RAM	1024 bytes		
	LCD display RAM	40 × 4 bits		
Memory space		64K bytes		
General-purpose register		8 bits × 8 × 4 banks		
Minimum instruction execution time	With main system clock	0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs/12.8 μs (at 5.0 MHz)		
	With subsystem clock	122 μs (at 32.768 kHz)		
Instruction set		• 16-bit operation • Multiplication/division (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, Boolean operation) • BCD adjustment, etc.		
I/O port (including pins multiplexed with segment signal output)		• Total : 57 • CMOS input : 2 • CMOS I/O : 55		
A/D converter		8-bit resolution × 8 channels		
LCD controller/driver		• Segment signal output : 40 max. • Common signal output : 4 max. • Bias : 1/2 or 1/3 bias selectable		
Serial interface		• 3-wire serial I/O/SBI/2-wire serial I/O mode selectable : 1 channel • 3-wire serial I/O/UART mode selectable : 1 channel • 3-wire serial I/O mode : 1 channel		
Timer		• 16-bit timer/event counter : 1 channel • 8-bit timer/event counter : 2 channels • Watch timer : 1 channel • Watchdog timer : 1 channel		
Timer output		3 (14-bit PWM output: 1)		
Clock output		19.5 kHz, 39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5.0 MHz (with main system clock of 5.0 MHz), 32.768 kHz (with subsystem clock of 32.768 kHz)		
Buzzer output		1.2 kHz, 2.4 kHz, 4.9 kHz, 9.8 kHz (with main system clock of 5.0 MHz)		
Vectored interrupt	Maskable	Internal: 13, external: 6		
	Non-maskable	Internal: 1		
	Software	1		
Test input		Internal: 1, external: 1		
Supply voltage		V_{DD} = 2.0 to 5.5 V		
Package		• 100-pin plastic QFP (fine pitch) (14 × 14 mm) • 100-pin plastic QFP (14 × 20 mm) • 100-pin ceramic WQFN (14 × 20 mm) (μPD78P0308 only)		

Note The capacity of the internal PROM can be changed by using a memory size select register (IMS).

Caution The μPD780308 subseries is under development.

★ **Figure 1-14. Block Diagram of μPD780308Y Subseries**



Remarks 1. The internal ROM capacity differs depending on the model.

2. ( ): μPD78P0308Y

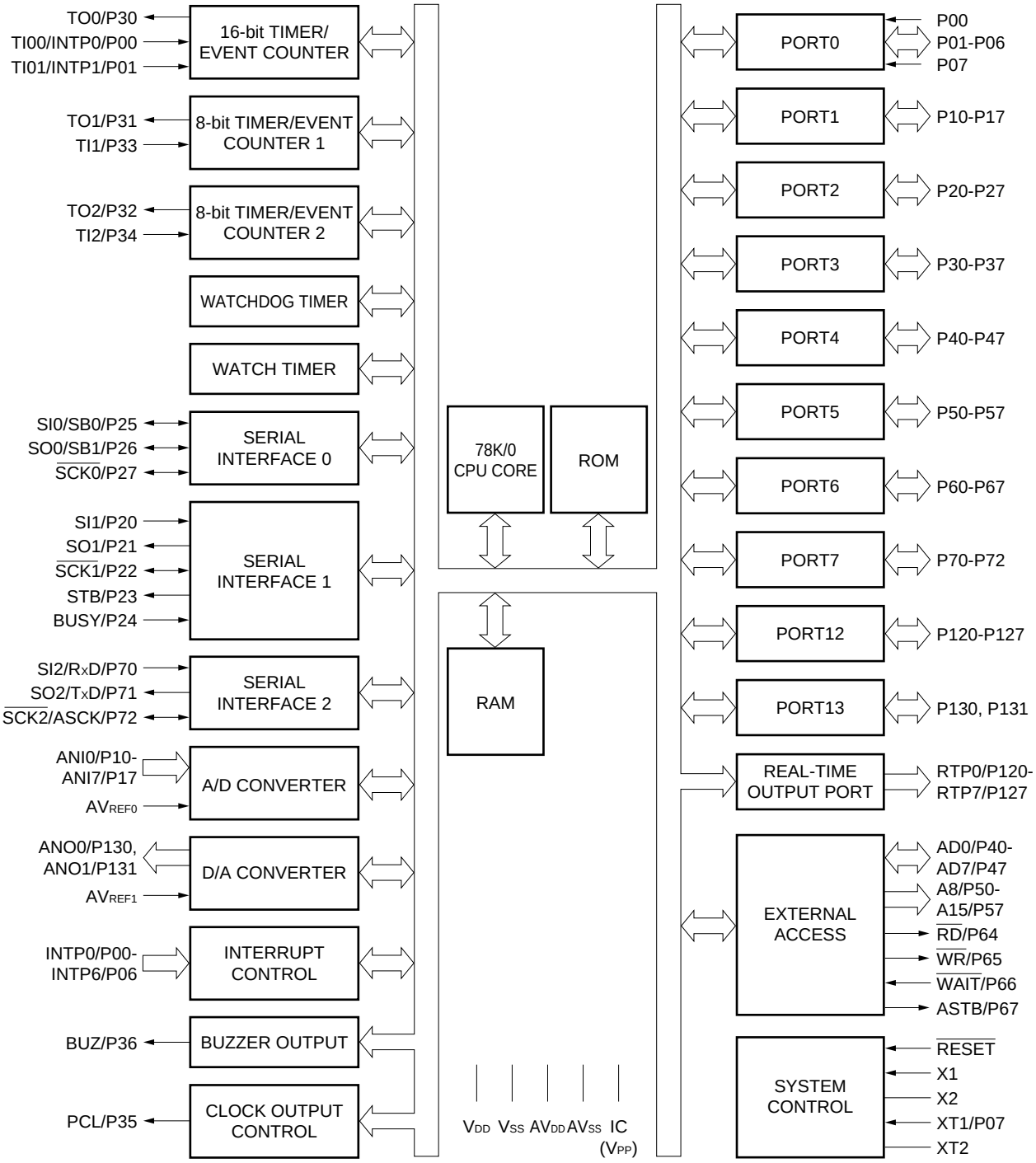
★ **Table 1-14. Functional Outline of μPD780308Y Subseries**

Item / Part Number		μPD780306Y	μPD780308Y	μPD78P0308Y
Internal memory	ROM	Mask ROM		PROM
		48K bytes	60K bytes	60K bytes[Note]
	High-speed RAM	1024 bytes		
	Expansion RAM	1024 bytes		
	LCD display RAM	40 × 4 bits		
Memory space		64K bytes		
General-purpose register		8 bits × 8 × 4 banks		
Minimum Instruction execution cycle	With main system clock	0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs/12.8 μs (at 5.0 MHz)		
	With subsystem clock	122 μs (at 32.768 kHz)		
Instruction set		• 16-bit operation • Multiplication/division (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, Boolean operation) • BCD adjustment, etc.		
I/O port (including pins multiplexed with segment signal output)		• Total : 57 • CMOS input : 2 • CMOS I/O : 55		
A/D converter		8-bit resolution × 8 channels		
LCD controller/driver		• Segment signal output : 40 max. • Common signal output : 4 max. • Bias : 1/2 or 1/3 bias selectable		
Serial interface		• 3-wire serial I/O/2-wire serial I/O/I^2C bus mode selectable : 1 channel • 3-wire serial I/O/UART mode selectable : 1 channel • 3-wire serial I/O mode : 1 channel		
Timer		• 16-bit timer/event counter : 1 channel • 8-bit timer/event counter : 2 channels • Watch timer : 1 channel • Watchdog timer : 1 channel		
Timer output		3 (14-bit PWM output: 1)		
Clock output		19.5 kHz, 39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5.0 MHz (with main system clock of 5.0 MHz), 32.768 kHz (with subsystem clock of 32.768 kHz)		
Buzzer output		1.2 kHz, 2.4 kHz, 4.9 kHz, 9.8 kHz (with main system clock of 5.0 MHz)		
Vectored interrupt source	Maskable	Internal: 13, external: 6		
	Non-maskable	Internal: 1		
	Software	1		
Test input		Internal: 1, external: 1		
Supply voltage		V_{DD} = 2.0 to 5.5 V		
Package		• 100-pin plastic QFP (14 × 20 mm) • 100-pin ceramic WQFN (14 × 20 mm) (μPD78P0308Y only)		

Note The capacity of the internal PROM can be changed by using a memory size select register (IMS).

Caution The μPD780308Y subseries is under development.

★ **Figure 1-15. Block Diagram of μPD78058F Subseries**



Remarks 1. The internal ROM and RAM capacities differ depending on the model.

2. ( ): μPD78P058F

★ **Table 1-15. Functional Outline of μPD78058F Subseries (1/2)**

Item / Part Number		μPD78056F	μPD78058F	μPD78P058F
Internal memory	ROM	Mask ROM		PROM
		48K bytes	60K bytes	60K bytes[Note 1]
	High-speed RAM	1024 bytes		
	Buffer RAM	32 bytes		
	Expansion RAM	None	1024 bytes	1024 bytes[Note 2]
Memory space		64K bytes		
General-purpose register		8 bits × 8 × 4 banks		
Minimum instruction execution time	With main system clock	0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs/12.8 μs (at 5.0 MHz)		
	With subsystem clock	122 μs (at 32.768 kHz)		
Instruction set		• 16-bit operation • Multiplication/division (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, Boolean operation) • BCD adjustment, etc.		
I/O port		• Total : 69 • CMOS input : 2 • CMOS I/O : 63 • N-ch open-drain I/O : 4		
A/D converter		8-bit resolution × 8 channels		
D/A converter		8-bit resolution × 2 channels		
Serial interface		• 3-wire serial I/O/SBI/2-wire serial I/O mode selectable : 1 channel • 3-wire serial I/O mode (with function to automatically transfer/receive up to 32 bytes) : 1 channel • 3-wire serial I/O/UART mode selectable : 1 channel		
Timer		• 16-bit timer/event counter : 1 channel • 8-bit timer/event counter : 2 channels • Watch timer : 1 channel • Watchdog timer : 1 channel		
Timer output		3 (14-bit PWM output: 1)		
Clock output		19.5 kHz, 39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5.0 MHz (with main system clock of 5.0 MHz), 32.768 kHz (with subsystem clock of 32.768 kHz)		

Notes 1. The capacity of the internal PROM can be changed by using a memory size select register (IMS).

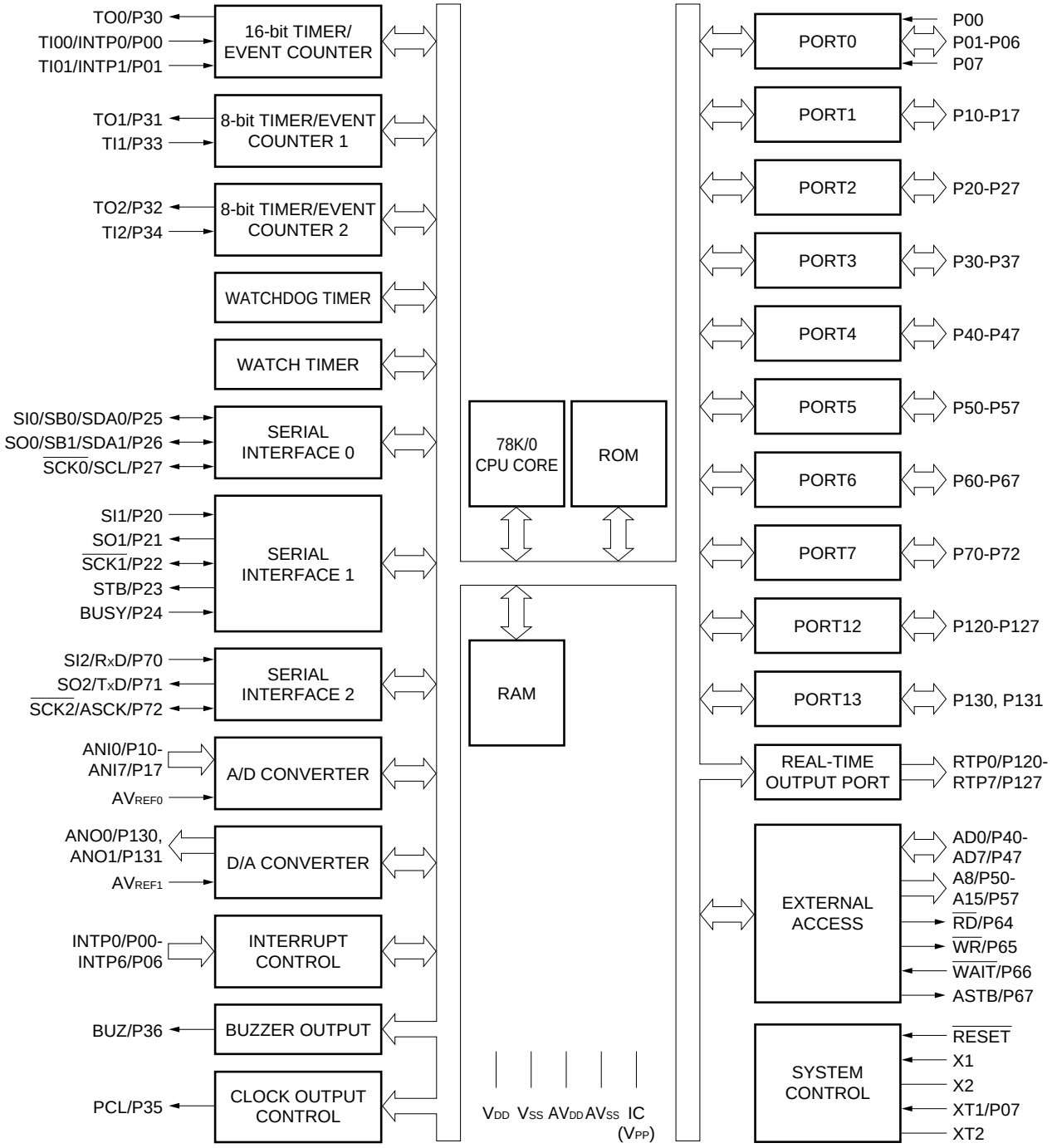
2. The internal expansion RAM capacity can be changed by using an internal expansion RAM size select register (IXS).

Table 1-15. Functional Outline of μPD78058F Subseries (2/2)

Item / Part Number		μPD78056F	μPD78058F	μPD78P058F
Buzzer output		1.2 kHz, 2.4 kHz, 4.9 kHz, 9.8 kHz (with main system clock of 5.0 MHz)		
Vectored interrupt source	Maskable	Internal: 13, external: 7		
	Non-maskable	Internal: 1		
	Software	Internal: 1		
Test input		Internal: 1, external: 1		
Supply voltage		V_{DD} = 2.7 to 6.0 V		
Package		• 80-pin plastic QFP (14 × 14 mm, resin thickness 2.7 mm) • 80-pin plastic QFP (14 × 14 mm, resin thickness 1.4 mm)**Note** • 80-pin plastic TQFP (fine pitch) (12 × 12 mm) (μPD78058F only)		

Note Under planning

★ **Figure 1-16. Block Diagram of μPD78058FY Subseries**



Remarks 1. The capacities of the internal ROM and RAM differ depending on the model.

2. ( ): μPD78P058FY

★ **Table 1-16. Functional Outline of μPD78058FY Subseries (1/2)**

Item / Part Number		μPD78056FY	μPD78058FY	μPD78P058FY
Internal memory	ROM	Mask ROM		PROM
		48K bytes	60K bytes	60K bytes[Note 1]
	High-speed RAM	1024 bytes		
	Buffer RAM	32 bytes		
	Expansion RAM	None	1024 bytes	1024 bytes[Note 2]
Memory space		64K bytes		
General-purpose register		8 bits × 8 × 4 banks		
Minimum instruction execution time	With main system clock	0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs/12.8 μs (at 5.0 MHz)		
	With subsystem clock	122 μs (at 32.768 kHz)		
Instruction set		• 16-bit operation • Multiplication/division (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, Boolean operation) • BCD adjustment, etc.		
I/O port		• Total : 69 • CMOS input : 2 • CMOS I/O : 63 • N-ch open-drain I/O : 4		
A/D converter		8-bit resolution × 8 channels		
D/A converter		8-bit resolution × 2 channels		
Serial interface		• 3-wire serial I/O/2-wire serial I/O/I^2C bus mode selectable : 1 channel • 3-wire serial I/O mode (with function to automatically transfer/receive up to 32 bytes) : 1 channel • 3-wire serial I/O/UART mode selectable : 1 channel		
Timer		• 16-bit timer/event counter : 1 channel • 8-bit timer/event counter : 2 channels • Watch timer : 1 channel • Watchdog timer : 1 channel		
Timer output		3 (14-bit PWM output: 1)		
Clock output		19.5 kHz, 39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5.0 MHz (with main system clock of 5.0 MHz), 32.768 kHz (with subsystem clock of 32.768 kHz)		

Notes 1. The capacity of the internal PROM can be changed by using a memory size select register (IMS).

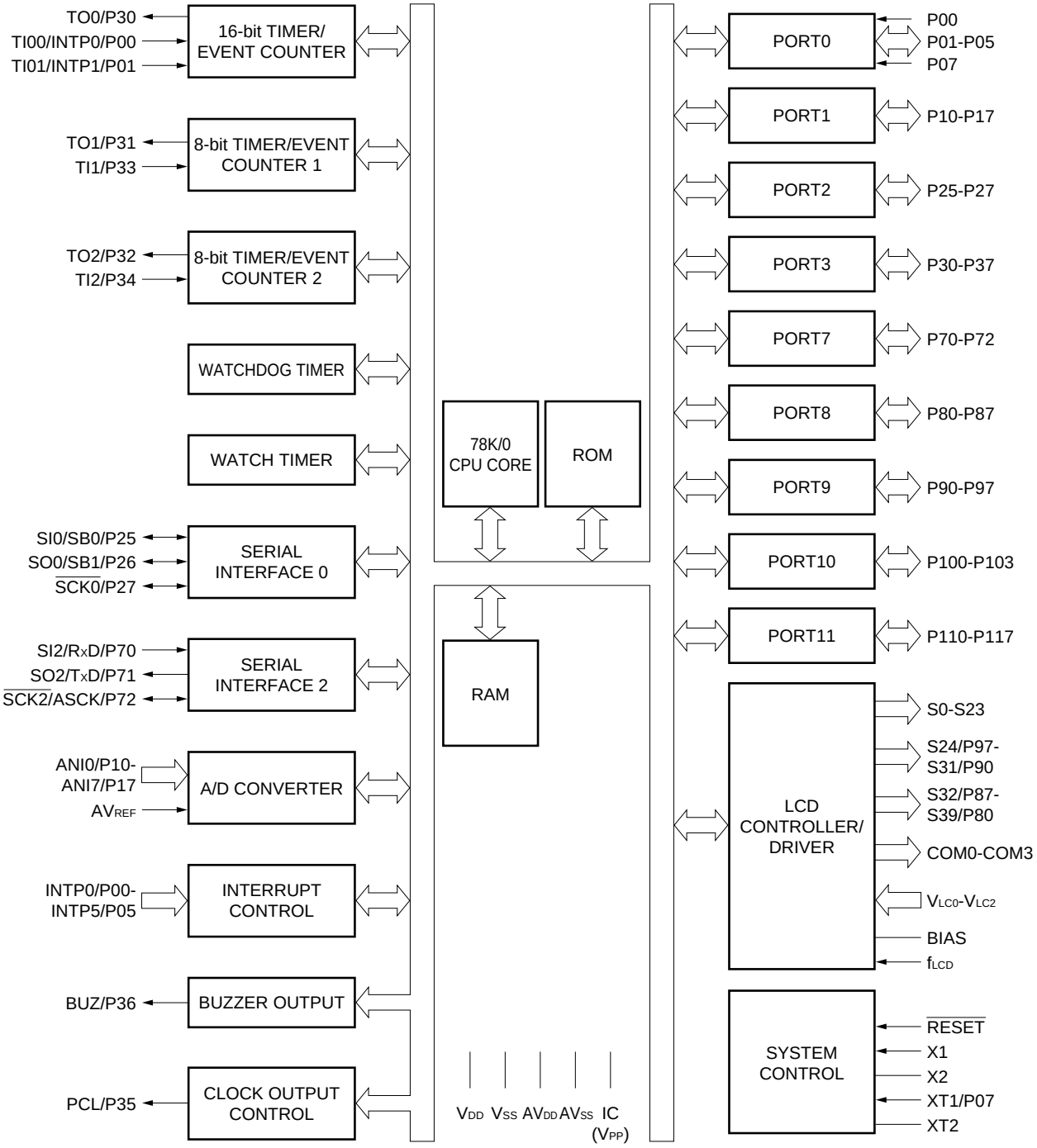
2. The internal expansion RAM capacity can be changed by using an internal expansion RAM size select register (IXS).

Table 1-16. Functional Outline of μPD78058FY Subseries (2/2)

Item / Part Number		μPD78056FY	μPD78058FY	μPD78P058FY
Buzzer output		1.2 kHz, 2.4 kHz, 4.9 kHz, 9.8 kHz (with main system clock of 5.0 MHz)		
Vectored interrupt source	Maskable	Internal: 13, external: 7		
	Non-maskable	Internal: 1		
	Software	1		
Test input		Internal: 1, external: 1		
Supply voltage		V_{DD} = 2.7 to 6.0 V		
Package		• 80-pin plastic QFP (14 × 14 mm, resin thickness 2.7 mm) • 80-pin plastic QFP (14 × 14 mm, resin thickness 1.4 mm)**Note** • 80-pin plastic TQFP (fine pitch)(12 × 12 mm) (μPD78058FY only)		

Note Under planning

★ **Figure 1-17. Block Diagram of μPD78064B Subseries**

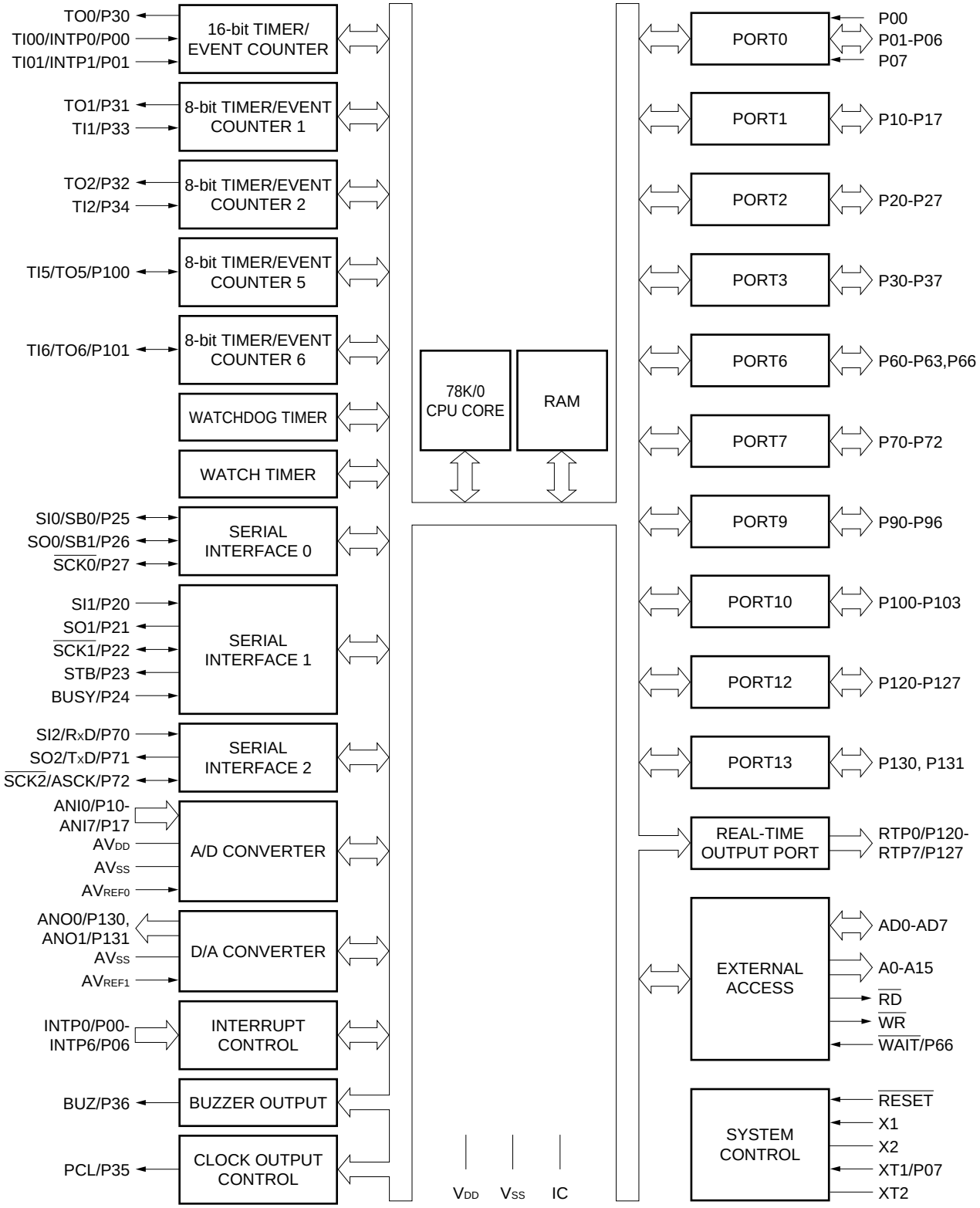


Remark ( ): μPD78P064B

★ **Table 1-17. Functional Outline of μPD78064B Subseries**

Item / Part Number		μPD78064B	μPD78P064B
Internal memory	ROM	Mask ROM	PROM
		32K bytes	
	High-speed RAM	1024 bytes	
	LCD display RAM	40 × 4 bits	
Memory space		64K bytes	
General-purpose register		8 bits × 8 × 4 banks	
Minimum instruction execution time	With main system clock	0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs/12.8 μs (at 5.0 MHz)	
	With subsystem clock	122 μs (at 32.768 kHz)	
Instruction set		• 16-bit operation • Multiplication/division (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, Boolean operation) • BCD adjustment, etc.	
I/O port (including pins multiplexed with segment signal output)		• Total : 57 • CMOS input : 2 • CMOS I/O : 55	
A/D converter		8-bit resolution × 8 channels	
LCD controller/driver		• Segment signal output : 40 max. • Common signal output : 4 max. • Bias : 1/2 or 1/3 bias selectable	
Serial interface		• 3-wire serial I/O/SBI/2-wire serial I/O mode selectable : 1 channel • 3-wire serial I/O/UART mode selectable : 1 channel	
Timer		• 16-bit timer/event counter : 1 channel • 8-bit timer/event counter : 2 channels • Watch timer : 1 channel • Watchdog timer : 1 channel	
Timer output		3 (14-bit PWM output: 1)	
Clock output		19.5 kHz, 39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5.0 MHz (with main system clock of 5.0 MHz), 32.768 kHz (with subsystem clock of 32.768 kHz)	
Buzzer output		1.2 kHz, 2.4 kHz, 4.9 kHz, 9.8 kHz (with main system clock of 5.0 MHz)	
Vectored interrupt source	Maskable	Internal: 12, external: 6	
	Non-maskable	Internal: 1	
	Software	1	
Test input		Internal: 1, external: 1	
Supply voltage		V_{DD} = 2.0 to 6.0 V	
Package		• 100-pin plastic QFP (fine pitch) (14 × 14 mm) • 100-pin plastic QFP (14 × 20 mm)	

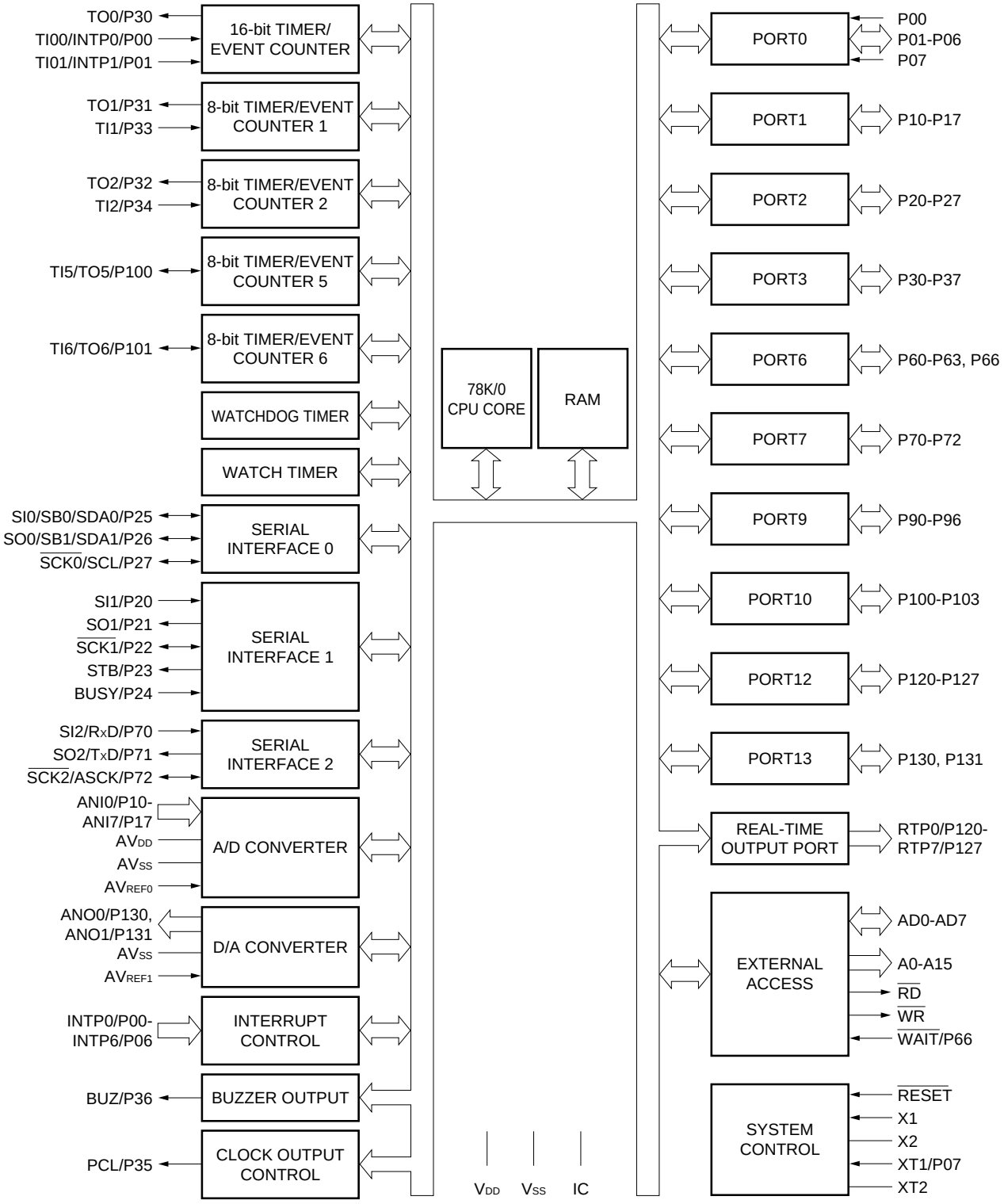
★ **Figure 1-18. Block Diagram of μPD78070A**



★ **Table 1-18. Functional Outline of μPD78070A**

Part Number		Functions
Internal memory	ROM	None
	High-speed RAM	1024 bytes
	Buffer RAM	32 bytes
Memory space		64K bytes
General-purpose register		8 bits × 8 × 4 banks
Minimum instruction execution time	With main system clock	0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs/12.8 μs (at 5.0 MHz)
	With subsystem clock	122 μs (at 32.768 kHz)
Instruction set		• 16-bit operation • Multiplication/division (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, Boolean operation) • BCD adjustment, etc.
I/O port		• Total : 61 • CMOS input : 2 • CMOS I/O : 51 • N-ch open-drain I/O : 8
A/D converter		8-bit resolution × 8 channels
D/A converter		8-bit resolution × 2 channels
Serial interface		• 3-wire serial I/O/SBI/2-wire serial I/O mode selectable : 1 channel • 3-wire serial I/O mode (with function to automatically transfer/receive up to 32 bytes) : 1 channel • 3-wire serail I/O/UART mode selectable : 1 channel
Timer		• 16-bit timer/event counter : 1 channel • 8-bit timer/event counter : 4 channels • Watch timer : 1 channel • Watchdog timer : 1 channel
Timer output		5 (14-bit PWM output: 1, 8-bit PWM output: 2)
Clock output		19.5 kHz, 39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5.0 MHz (with main system clock of 5.0 MHz), 32.768 kHz (with subsystem clock of 32.768 kHz)
Vectored interrupt source	Maskable	Internal: 15, external: 7
	Non-maskable	Internal: 1
	Software	1
Test input		Internal: 1
Supply voltage		V_{DD} = 2.7 to 5.5 V
Package		• 100-pin plastic QFP (fine pitch) (14 × 14 mm) • 100-pin plastic QFP (14 × 20 mm)

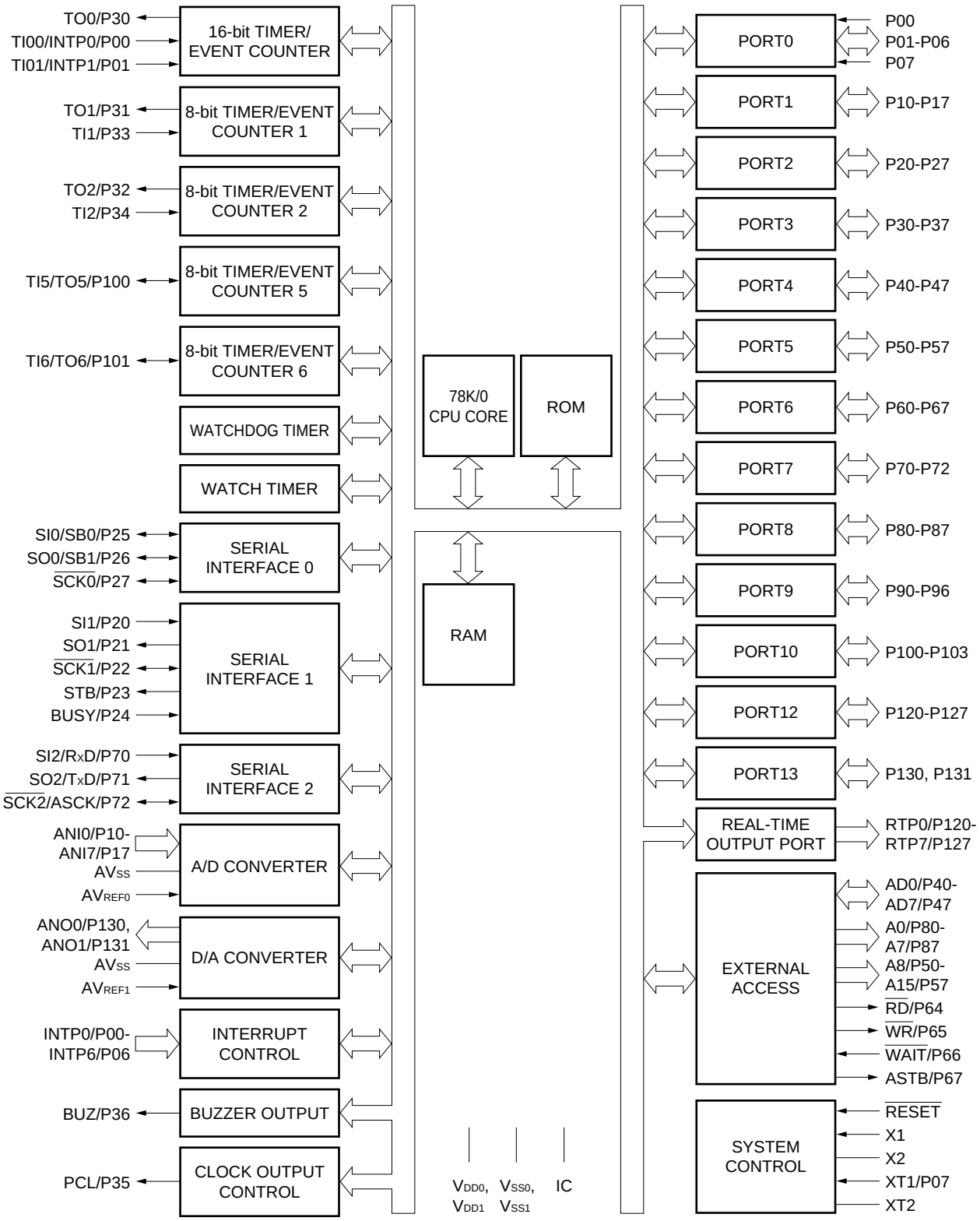
★ **Figure 1-19. Block Diagram of μPD78070AY**



★ **Table 1-19. Functional Outline of μPD78070AY**

Part Number		Functions
Internal memory	ROM	None
	High-speed RAM	1024 bytes
	Buffer RAM	32 bytes
Memory space		64K bytes
General-purpose register		8 bits × 8 × 4 banks
Minimum instruction execution time	With main system clock	0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs/12.8 μs (at 5.0 MHz)
	With subsystem clock	122 μs (at 32.768 kHz)
Instruction set		• 16-bit operation • Multiplication/division (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, Boolean operation) • BCD adjustment, etc.
I/O port		• Total : 61 • CMOS input : 2 • CMOS I/O : 51 • N-ch open-drain I/O : 8
A/D converter		8-bit resolution × 8 channels
D/A converter		8-bit resolution × 2 channels
Serial interface		• 3-wire serial I/O/2-wire serial I/O/I^2C bus mode selectable : 1 channel • 3-wire serial I/O mode (with function to automatically transfer/receive up to 32 bytes) : 1 channel • 3-wire serial I/O/UART mode selectable : 1 channel
Timer		• 16-bit timer/event counter : 1 channel • 8-bit timer/event counter : 4 channels • Watch timer : 1 channel • Watchdog timer : 1 channel
Timer output		5 (14-bit PWM output: 1, 8-bit PWM output: 2)
Clock output		19.5 kHz, 39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5.0 MHz (with main system clock of 5.0 MHz), 32.768 kHz (with subsystem clock of 32.768 kHz)
Buzzer output		1.2 kHz, 2.4 kHz, 4.9 kHz, 9.8 kHz (with main system clock of 5.0 MHz)
Vectored interrupt source	Maskable	Internal: 15, external: 7
	Non-maskable	Internal: 1
	Software	1
Test input		Internal: 1
Supply voltage		V_{DD} = 2.7 to 5.5 V
Package		• 100-pin plastic QFP (14 × 20 mm) • 100-pin plastic QFP (fine pitch) (14 × 14 mm)

★ **Figure 1-20. Block Diagram of μPD78075B Subseries**

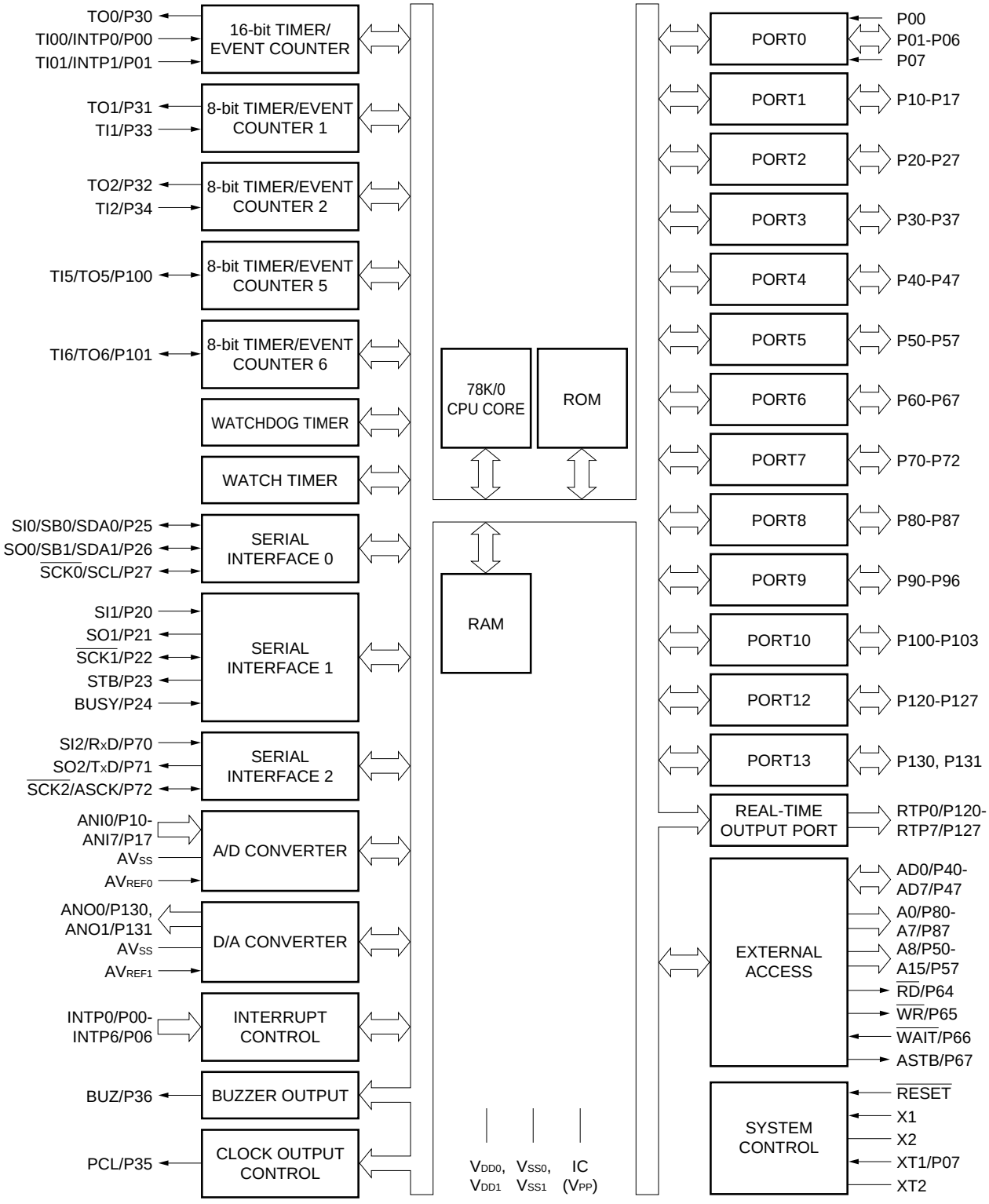


Remark The internal ROM capacity differs depending on the model.

★ **Table 1-20. Functional Outline of μPD78075B8 Subseries**

Item / Part Number		μPD78074B	μPD78075B
Internal memory	ROM	Mask ROM	
	High-speed RAM	32K bytes	40K bytes
	Buffer RAM	1024 bytes	
	Expansion RAM	32 bytes	
Memory space		64K bytes	
General-purpose register		8 bits × 8 × 4 banks	
Minimum instruction execution time	With main system clock	0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs/12.8 μs (at 5.0 MHz)	
	With subsystem clock	122 μs (at 32.768 kHz)	
Instruction set		• 16-bit operation • Multiplication/division (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, Boolean operation) • BCD adjustment, etc.	
I/O port		• Total : 88 • CMOS input : 2 • CMOS I/O : 78 • N-ch open-drain I/O : 8	
A/D converter		8-bit resolution × 8 channels	
D/A converter		8-bit resolution × 2 channels	
Serial interface		• 3-wire serial I/O/SBI/2-wire serial I/O mode selectable : 1 channel • 3-wire serial I/O mode (with function to automatically transfer/receive up to 32 bytes) : 1 channel • 3-wire serial I/O/UART mode selectable : 1 channel	
Timer		• 16-bit timer/event counter : 1 channel • 8-bit timer/event counter : 4 channels • Watch timer : 1 channel • Watchdog timer : 1 channel	
Timer output		5 (14-bit PWM output: 1, 8-bit PWM output: 2)	
Clock output		19.5 kHz, 39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5.0 MHz (with main system clock of 5.0 MHz), 32.768 kHz (with subsystem clock of 32.768 kHz)	
Vectored interrupt source	Maskable	Internal: 15, external: 7	
	Non-maskable	Internal: 1	
	Software	1	
Test input		Internal: 1, external: 1	
Supply voltage		V_{DD} = 1.8 to 5.5 V	
Package		• 100-pin plastic QFP (fine pitch) (14 × 14 mm, resin thickness 1.45 mm) • 100-pin plastic QFP (14 × 20 mm, resin thickness 2.7 mm)	

★ **Figure 1-21. Block Diagram of μPD78075BY Subseries**



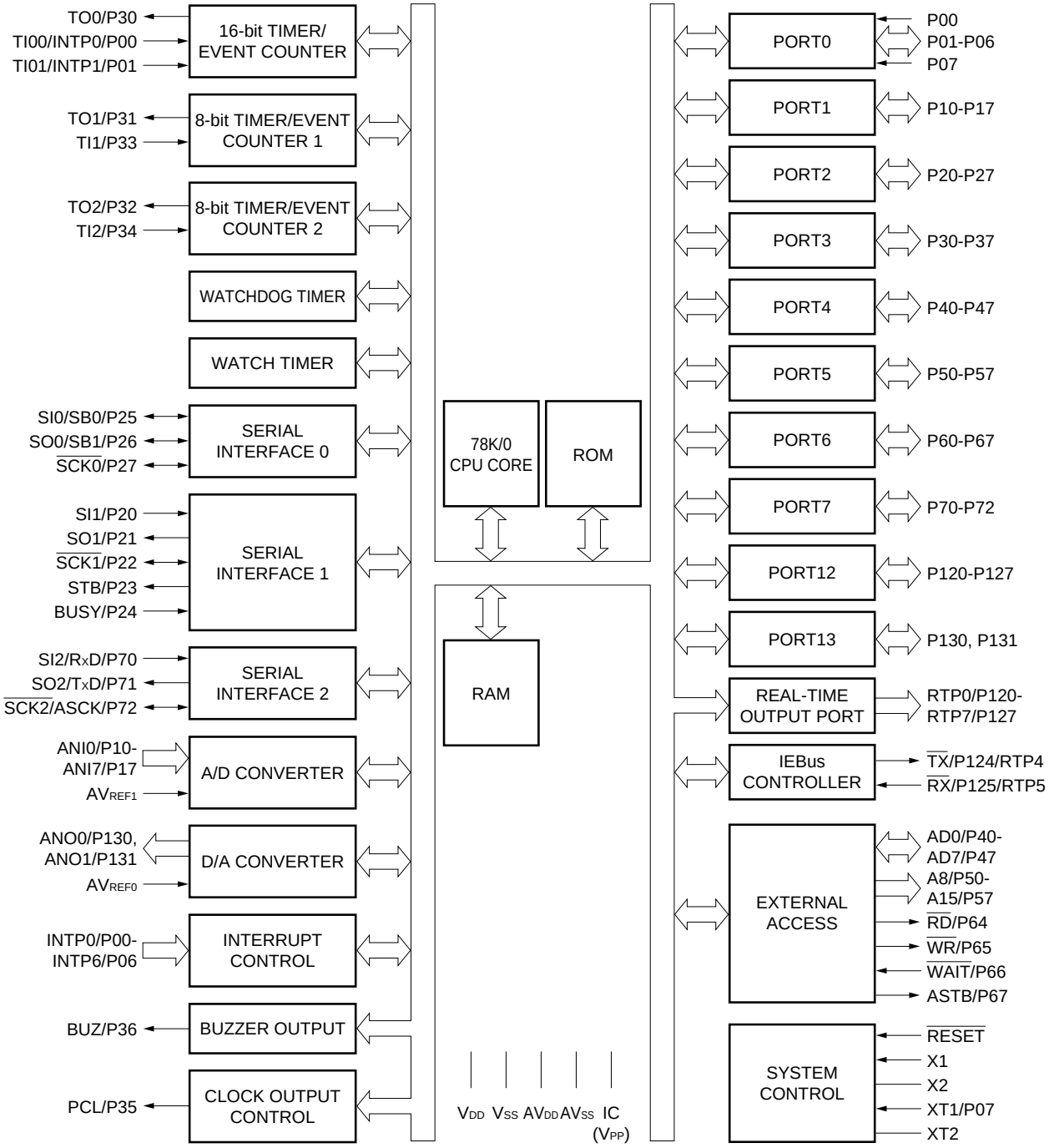
Remark The internal ROM capacity differs depending on the model.

★ **Table 1-21. Functional Outline of μPD78075BY Subseries**

Item / Part Number		μPD78074BY	μPD78075BY
Internal memory	ROM	Mask ROM	
		32K bytes	40K bytes
	High-speed RAM	1024 bytes	
	Buffer RAM	32 bytes	
Memory space		64K bytes	
General-purpose register		8 bits × 8 × 4 banks	
Minimum instruction execution time	With main system clock	0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs/12.8 μs (at 5.0 MHz)	
	With subsystem clock	122 μs (at 32.768 kHz)	
Instruction set		• 16-bit operation • Multiplication/division (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, Boolean operation) • BCD adjustment, etc.	
I/O port		• Total : 88 • CMOS input : 2 • CMOS I/O : 78 • N-ch open-drain I/O : 8	
A/D converter		8-bit resolution × 8 channels	
D/A converter		8-bit resolution × 2 channels	
Serial interface		• 3-wire serial I/O/2-wire serial I/O/I^2C bus mode selectable : 1 channel • 3-wire serial I/O mode (with function to automatically transfer/receive up to 32 bytes) : 1 channel • 3-wire serial I/O/UART mode selectable : 1 channel	
Timer		• 16-bit timer/event counter : 1 channel • 8-bit timer/event counter : 4 channels • Watch timer : 1 channel • Watchdog timer : 1 channel	
Timer output		5 (14-bit PWM output: 1, 8-bit PWM output: 2)	
Clock output		19.5 kHz, 39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5.0 MHz (with main system clock of 5.0 MHz), 32.768 kHz (with subsystem clock of 32.768 kHz)	
Buzzer output		1.2 kHz, 2.4 kHz, 4.9 kHz, 9.8 kHz (with main system clock of 5.0 MHz)	
Vectored interrupt source	Maskable	Internal: 15, external: 7	
	Non-maskable	Internal: 1	
	Software	1	
Test input		Internal: 1, external: 1	
Supply voltage		V_{DD} = 1.8 to 5.5 V	
Package		• 100-pin plastic QFP (14 × 20 mm, resin thickness 2.7 mm)	

Caution The μPD78075BY subseries is under development.

★ **Figure 1-22. Block Diagram of μPD78098B Subseries**



Remarks 1. The internal ROM and RAM capacities differ depending on the model.

2. ( ): μPD78P098B

★ **Table 1-22. Functional Outline of μPD78098B Subseries**

Item / Part Number		μPD78095B	μPD78096B	μPD78098B	μPD78P098B
Internal memory	ROM	Mask ROM			PROM
		40K bytes	48K bytes	60K bytes	60K bytes[Note 1]
	High-speed RAM	1024 bytes			
	Buffer RAM	32 bytes			
	Expansion RAM	None		2048 bytes	2048 bytes[Note 2]
Memory space		64K bytes			
General-purpose register		8 bits × 8 × 4 banks			
Minimum instruction execution time	With main system clock	0.5 μs/1.0 μs/2.0 μs/4.0 μs/8.0 μs/16.0 (at 6.0 MHz)			
	With subsystem clock	122 μs (at 32.768 kHz)			
Instruction set		• 16-bit operation • Multiplication/division (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, Boolean operation) • BCD adjustment, etc.			
I/O port		• Total : 69 • CMOS input : 2 • CMOS I/O : 63 • N-ch open-drain I/O: 4			
IEBus controller		Effective transfer rate: 3.9 kbps/17 kbps/26 kbps			
A/D converter		8-bit resolution × 8 channels			
D/A converter		8-bit resolution × 2 channels			
Serial interface		• 3-wire serial I/O/SBI/2-wire serial I/O mode selectable : 1 channel • 3-wire serial I/O mode (with function to automatically transfer/receive up to 32 bytes) : 1 channel • 3-wire serial I/O/UART mode selectable : 1 channel			
Timer		• 16-bit timer/event counter : 1 channel • 8-bit timer/event counter : 2 channels • Watch timer : 1 channel • Watchdog timer : 1 channel			
Timer output		3 (14-bit PWM output: 1)			
Clock output		15.6 kHz, 31.3 kHz, 62.5 kHz, 125 kHz, 250 kHz, 500 kHz, 1.0 MHz, 2.0 MHz, 4.0 MHz (with main system clock of 6.0 MHz), 32.768 kHz (with subsystem clock of 32.768 kHz)			
Buzzer output		977 Hz, 1.95 kHz, 3.9 kHz, 7.8 kHz (with main system clock of 6.0 MHz)			
Vectored interrupt source	Maskable	Internal: 14, external: 7			
	Non-maskable	Internal: 1			
	Software	1			
Test input		Internal: 1, external: 1			
Supply voltage		V_{DD} = 2.7 to 6.0 V			
Package		• 80-pin plastic QFP (14 × 14 mm) • 80-pin ceramic WQFN (14 × 14 mm) (μPD78P098B only)			

Notes 1. The internal PROM capacity can be changed by using a memory size select register (IMS).

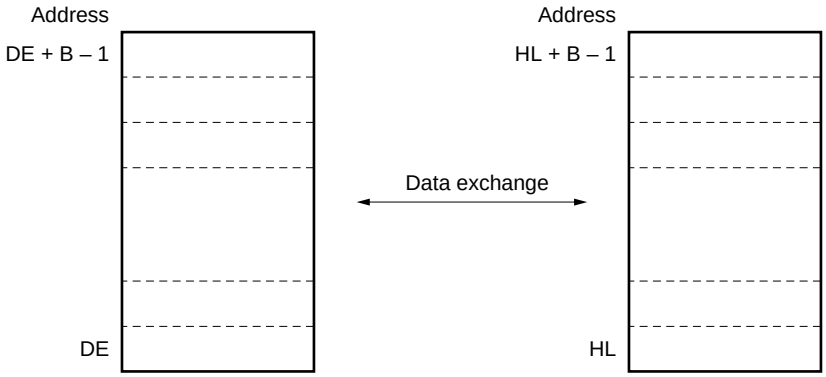
2. The internal expansion RAM can be changed by using an internal expansion RAM size select register (IXS).

Caution The μPD78098B subseries is under planning.

2.1 Data Transfer

Data is exchanged by using an address specified by the DE and HL registers as the first address. The number of bytes of the data to be exchanged is specified by the B register.

Figure 2-1. Data Exchange



(1) Registers used

A, B, DE, HL

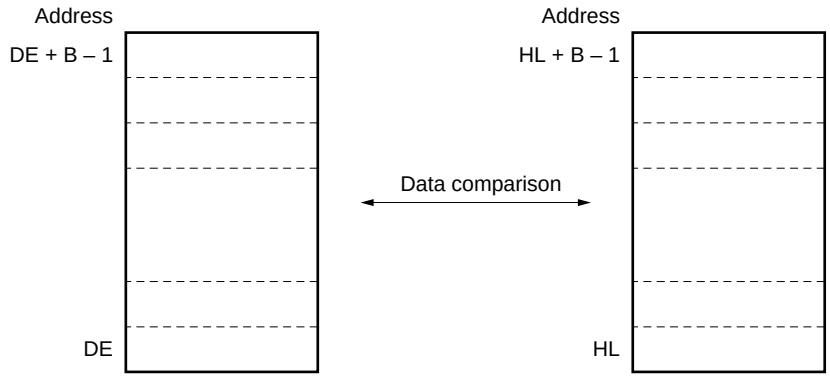
(2) Program list

```
EXCH:
        MOV     A,[DE]
        XCH     A,[HL]
        XCH     A,[DE]
        INCW    DE
        INCW    HL
        DBNZ    B,$EXCH
        RET
```

2.2 Data Comparison

Data is compared by using an address specified by the DE and HL registers as the first address. The number of bytes of the data to be compared is specified by the B register. If the result of comparison is equal, CY is cleared to 0; if not, CY is set to 1.

Figure 2-2. Data Comparison



(1) Registers used

A, B, DE, HL

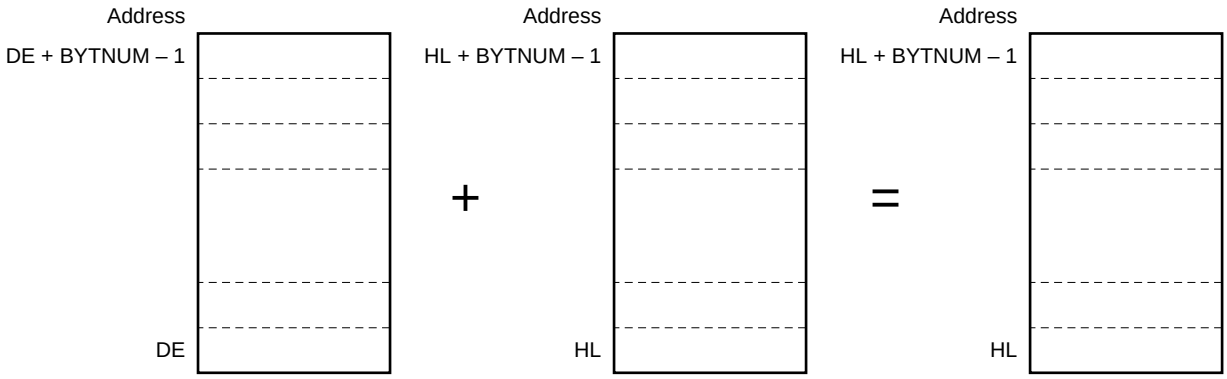
(2) Program list

```
COMP:
        MOV     A,[DE]
        CMP     A,[HL]
        BNZ     $ERROR
        INCW    DE
        INCW    HL
        DBNZ    B,$COMP
        CLR1    CY
        BR      RTN
ERROR:
        SET1    CY
RTN:
        RET
```

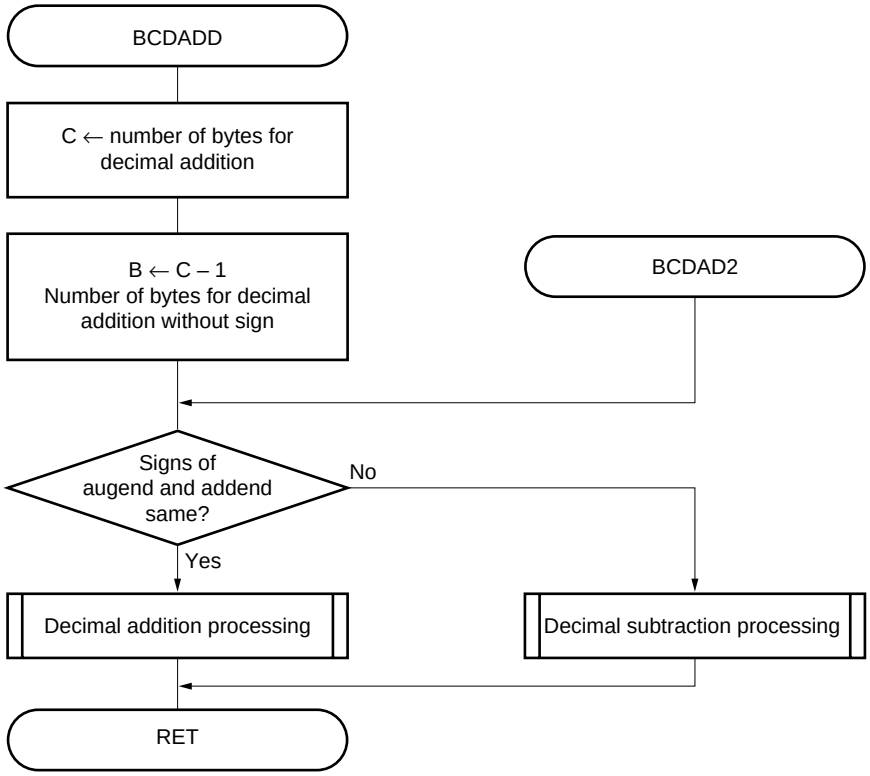
2.3 Decimal Addition

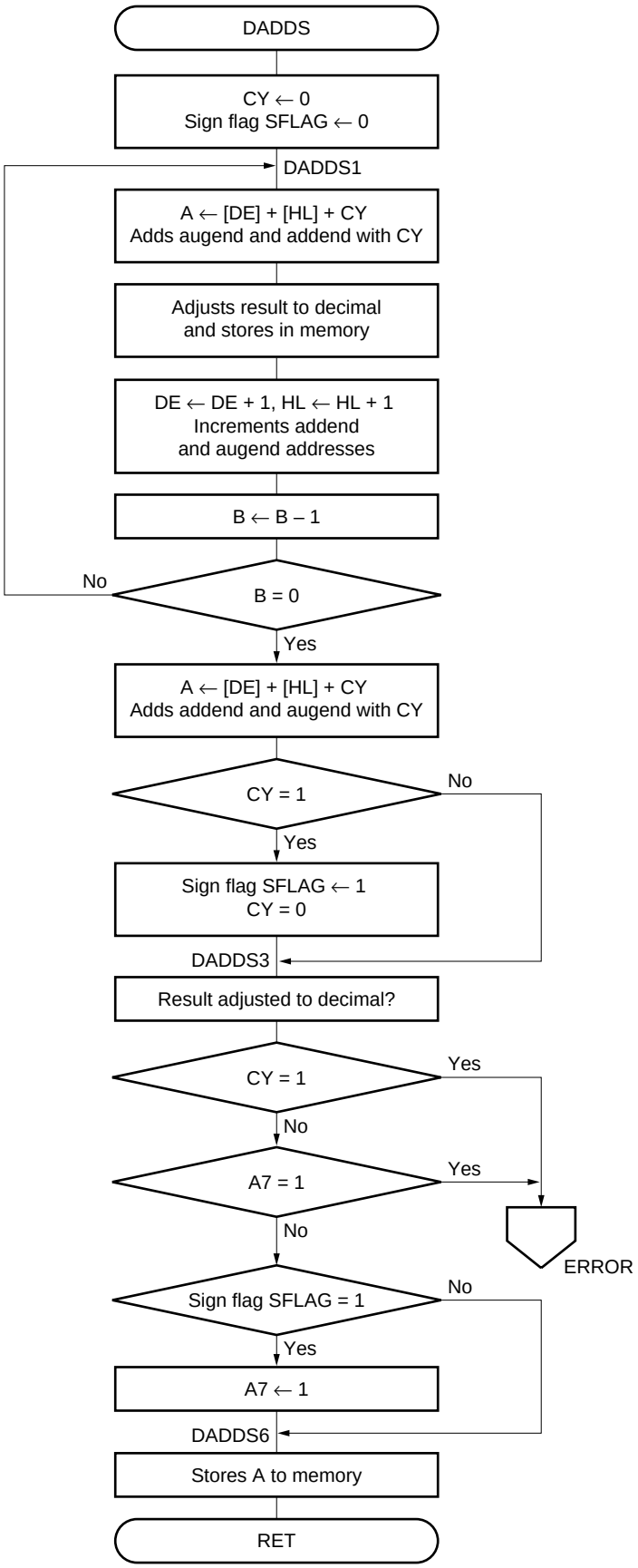
The lowest address for decimal addition is specified by the DE and HL registers, and the number of digits specified by BYTNUM is added. The result of the addition is stored to an area specified by the HL register. If an overflow or underflow occurs as a result of the addition, execution branches to error processing. Define the branch address as 'ERROR' in the main routine. Also declare it as PUBLIC.

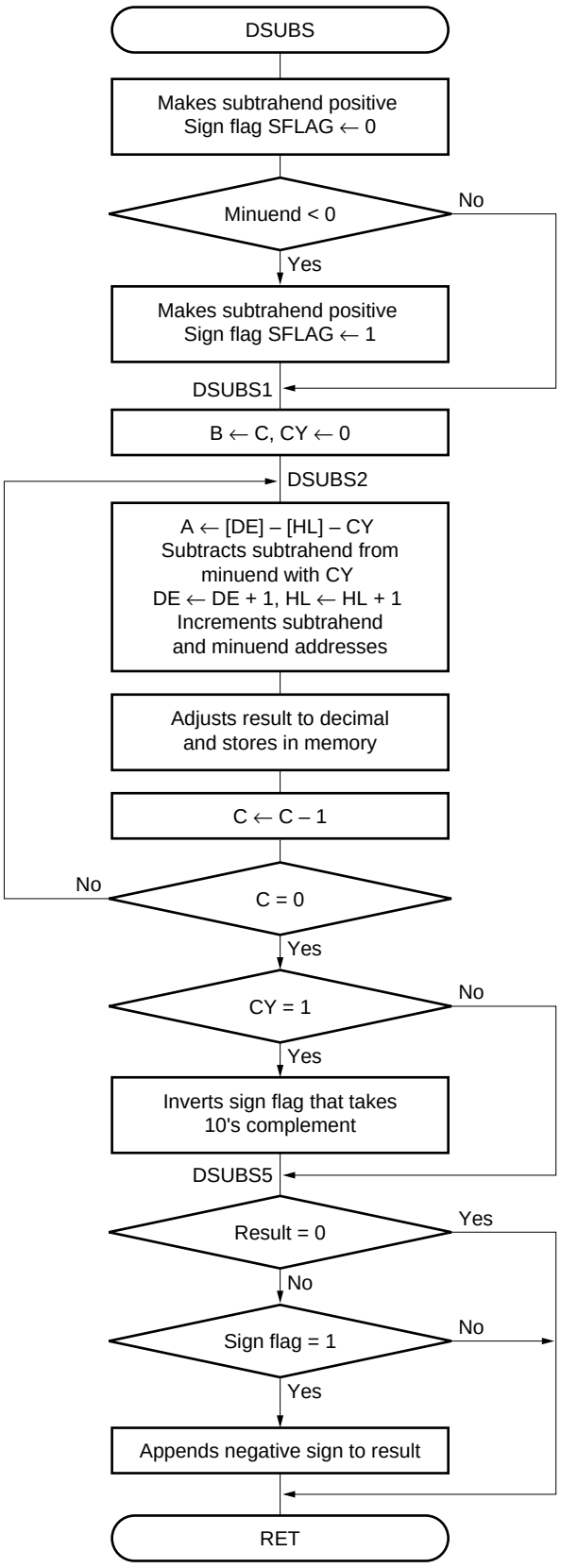
Figure 2-3. Decimal Addition



(1) Flowchart




DADDS
CY ← 0
Sign flag SFLAG ← 0
DADDS1
A ← [DE] + [HL] + CY
Adds augend and addend with CY
Adjusts result to decimal
and stores in memory
DE ← DE + 1, HL ← HL + 1
Increments addend
and augend addresses
B ← B – 1
No
B = 0
Yes
A ← [DE] + [HL] + CY
Adds addend and augend with CY
CY = 1
No
Yes
Sign flag SFLAG ← 1
CY = 0
DADDS3
Result adjusted to decimal?
CY = 1
Yes
No
A7 = 1
Yes
No
ERROR
Sign flag SFLAG = 1
No
Yes
A7 ← 1
DADDS6
Stores A to memory
RET


DSUBS
Makes subtrahend positive
Sign flag SFLAG ← 0
Minuend < 0
No
Yes
Makes subtrahend positive
Sign flag SFLAG ← 1
DSUBS1
B ← C, CY ← 0
DSUBS2
A ← [DE] – [HL] – CY
Subtracts subtrahend from
minuend with CY
DE ← DE + 1, HL ← HL + 1
Increments subtrahend
and minuend addresses
Adjusts result to decimal
and stores in memory
C ← C – 1
No
C = 0
Yes
CY = 1
No
Yes
Inverts sign flag that takes
10's complement
DSUBS5
Result = 0
Yes
No
Sign flag = 1
No
Yes
Appends negative sign to result
RET

(2) Registers used

AX, BC, DE, HL

(3) Program list

```
;*****************************************************************
;                                                                *
;       Input parameter                                          *
;              HL register: addend first address                 *
;              DE register: augend first address                 *
;       Output parameter                                         *
;              HL register: Operation result first address       *
;                                                                *
;*****************************************************************

        PUBLIC  BCDADD,BCDAD1,BCDAD2
        PUBLIC  DADDS
        PUBLIC  DSUBS
        EXTRN   ERROR                   ; Error processing branch address
        EXTBIT  SFLAG                   ; Sign flag
;
BYTNUM  EQU     4                       ; Sets number of digits for operation
;
        CSEG
BCDADD:
        MOV     C,#BYTNUM               ; Sets number of digits for operation to C register
BCDAD1:
        MOV     A,C
        MOV     B,A
        DEC     B
BCDAD2:
        MOV     A,[HL+BYTNUM-1]         ; Loads MSB (sign data) of augend
        XCHW    AX,DE
        XCHW    AX,HL
        XCHW    AX,DE
        XOR     A,[HL+BYTNUM-1]         ; Loads MSB (sign data) of augend
        XCHW    AX,HL
        XCHW    AX,DE
        XCHW    AX,HL

        BT      A.7,$BCDAD3             ; Signs coincide? ELSE subtraction processing
        CALL    !DADDS                  ; THEN addition processing
        RET
BCDAD3:
        CALL    !DSUBS
        RET
```

```
;=============================================================
;              ***** 10 Decimal addition *****
;=============================================================

DADDS:
        CLR1    CY
        CLR1    SFLAG
DADDS1:
        MOV     A,[DE]                  ; Starts addition from lowest digit
        ADDC    A,[HL]
        ADJBA
        MOV     [HL],A
        INCW    HL
        INCW    DE
        DBNZ    B,$DADDS1               ; End of addition (number of digits for operation – 1)

        MOV     A,[DE]
        ADDC    A,[HL]
DADDS2:
        BNC     $DADDS3                 ; Negative addition
        SET1    SFLAG                   ; THEN sets negative status
        CLR1    CY
DADDS3:
        ADJBA
        BNC     $DADDS4
        BR      ERROR
DADDS4:
        BF      A.7,$DADDS5
        BR      ERROR
DADDS5:
        BF      SFLAG,$DADDS6           ; Sets sign
        SET1    A.7
DADDS6:
        MOV     [HL],A
        RET
```

```
;=============================================================
;              ***** 10 Decimal subtraction *****
;=============================================================

DSUBS:
        PUSH    HL
        CLR1    SFLAG
        MOV     A,[HL+BYTNUM-1]         ; Sets subtrahend as positive value
        CLR1    A.7
        MOV     [HL+BYTNUM-1],A
        XCHW    AX,DE
        XCHW    AX,HL
        XCHW    AX,DE
        MOV     A,[HL+BYTNUM-1]
        BF      A.7,$DSUBS1             ; Minuend is negative
        CLR1    A.7                     ; THEN sets minuend as positive value
        MOV     [HL+BYTNUM-1],A
        SET1    SFLAG                   ; Sets sign as negative
DSUBS1:
        XCHW    AX,HL
        XCHW    AX,DE
        XCHW    AX,HL
        MOV     A,C
        MOV     B,A
        CLR1    CY
DSUBS2:
        MOV     A,[DE]
        SUBC    A,[HL]
        ADJBS
        MOV     [HL],A
        INCW    HL
        INCW    DE
        DBNZ    C,$DSUBS2               ; End of subtraction of number of digits for operation

        BNC     $DSUBS5                 ; THEN subtrahend > minuend
        POP     HL
        PUSH    HL
        MOV     A,B
        MOV     C,A
DSUBS3:
        MOV     A,#99H                  ; Complement operation of result of subtraction
        SUB     A,[HL]                  ;                          (result of subtraction – 99H)
        ADJBS
        MOV     [HL],A
        INCW    HL
        DBNZ    C,$DSUBS3


        POP     HL
        PUSH    HL
        SET1    CY

        MOV     A,B
        MOV     C,A
DSUBS4:
        MOV     A,#0                    ; Adds 1 to result of complement operation
        ADDC    A,[HL]
        ADJBA
```

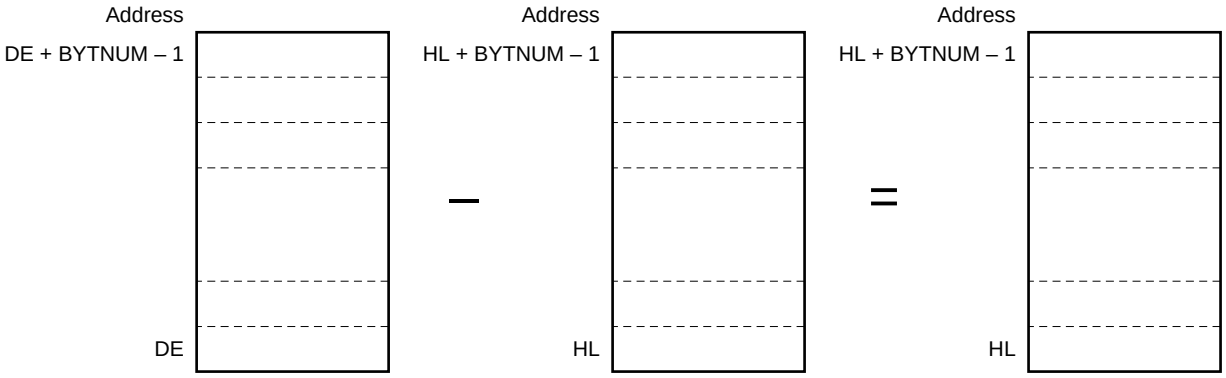
```
        MOV     [HL],A
        INCW    HL
        DBNZ    C,$DSUBS4
        MOV1    CY,SFLAG
        NOT1    CY
        MOV1    SFLAG,CY
;=====================================================
;       ***** 0 check of operation result *****
;=====================================================

DSUBS5:
        MOV     A,B
        MOV     C,A
        POP     HL
        PUSH    HL
        MOV     A,#0
DSUBS6:
        CMP     A,[HL]                  ; 0 check from lowest digit
        INCW    HL
        BNZ     $DSUBS7
        DBNZ    C,$DSUBS6               ; 0 check of all digits completed
        POP     HL                      ; THEN result of subtraction = 0
        RET
DSUBS7:
        BF      SFLAG,$DSUBS8           ; Result of subtraction is negative
        POP     HL                      ; THEN sets sign
        PUSH    HL
        MOV     A,[HL+BYTNUM-1]
        SET1    A.7
        MOV     [HL+BYTNUM-1],A
DSUBS8:
        POP     HL
        RET
```

2.4 Decimal Subtraction

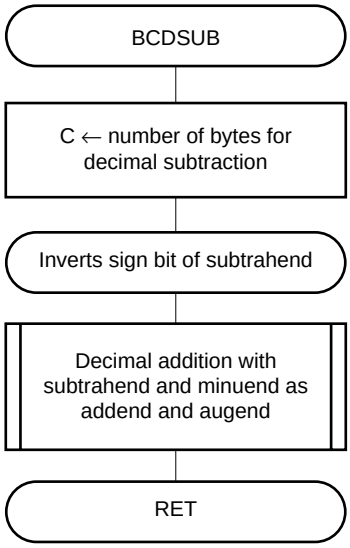
The lowest address for decimal subtraction is specified by the DE and HL registers, and the number of digits specified by BYTNUM is subtracted. The result of the subtraction is stored to an area specified by the HL register. If an overflow or underflow occurs as a result of the subtraction, execution branches to error processing. Define the branch address as 'ERROR' in the main routine. Also declare it as PUBLIC.

This program replaces minuend and subtrahend with augend and addend, and calls a program of decimal addition.

Figure 2-4. Decimal Subtraction



(1) Flowchart



(2) Registers used

AX, BC, DE, HL

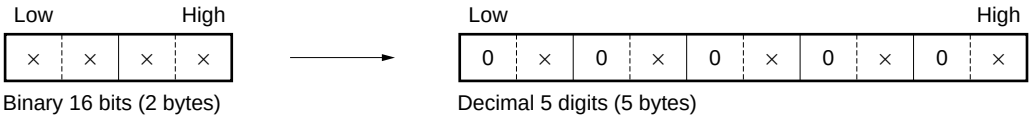
(3) Program list

```
;*****************************************************************
;                                                                *
;       Input parameter                                          *
;              HL register: subtrahend first address             *
;              DE register: minuend first address                *
;       Output parameter                                         *
;              HL register: Operation result first address       *
;                                                                *
;*****************************************************************

        PUBLIC  BYTNUM
        PUBLIC  BCDSUB
        EXTRN   BCDADD,BCDAD2
;
BYTNUM  EQU     4                       ; Sets number of digits for operation
;
        CSEG
BCDSUB:
        MOV     C,#BYTNUM               ; Sets number of digits for operation to C register
BCDSU1:
        MOV     A,C
        MOV     B,A
        DEC     B

        MOV     A,[HL+BYTNUM-1]         ; Sets MSB (sign data) of subtrahend for addition
        MOV1    CY,A.7                  ; Inverts sign data
        NOT1    CY
        MOV1    A.7,CY
        MOV     [HL+BYTNUM-1],A
        CALL    !BCDAD2                 ; Calls decimal addition processing
        RET
```

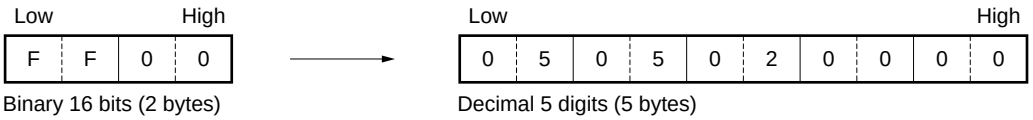
2.5 Binary-to-Decimal Conversion

Binary data of 16 bits in data memory is converted into 5-digit decimal data and stored in data memory. Binary data of 16 bits is divided by decimal 10 by the number of times equal to the number of digits (4 times), and conversion is carried out with the result of the operation and the value of the remainder at that time.

Figure 2-5. Binary-to-Decimal Conversion



Example To convert FFH into decimal number



(1) Registers used

AX, BC, HL

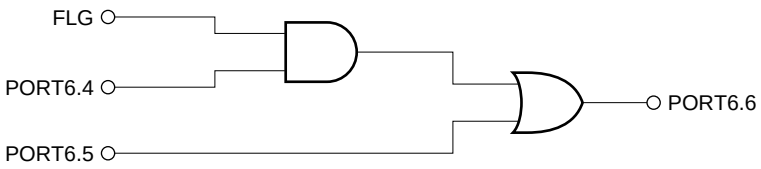
(2) Program list

```
        PUBLIC  B_DCONV
        DATDEC  EQU     10

        DSEG    SADDRP
REGA:   DS      2               ; Stores binary 16-bit data
REGB:   DS      5               ; Stores decimal 5-digit data

        COLNUM  EQU     4

B_DCONV:
        MOVW    AX,REGA
        MOV     B,#COLNUM
        MOVW    HL,#REGB
B_D1:
        MOV     C,#DATDEC
        DIVUW   C
        XCH     A,C
        MOV     [HL],A
        INCW    HL
        XCH     A,C
        DBNZ    B,$B_D1
        MOV     A,X
        MOV     [HL],A
        RET
```

2.6 Bit Manipulation Instruction

A 1 bit of a flag in the data memory is ANDed with the bit 4 of port 6, and the result is ANDed with the bit 5 of port 6 and is output to the bit 6 of port 6.

Figure 2-6. Bit Operation



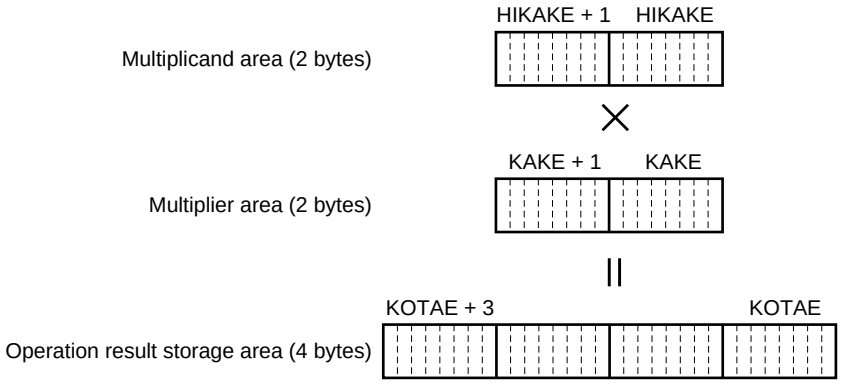
(1) Program list

```
        PUBLIC  BIT_OP,FLG

        BSEG
FLG     DBIT

BIT_OP:
        MOV1    CY,FLG
        AND1    CY,P6.4
        OR1     CY,P6.5
        MOV1    P6.6,CY
        RET
```

2.7 Binary Multiplication (16 bits × 16 bits)

Data in a multiplicand area (HIKAKE; 16 bits) and multiplier area (KAKE; 16 bits) are multiplied, and the result is stored in an operation result storage area (KOTAE).

Figure 2-7. Binary Multiplication



<Processing contents>

Multiplication is performed by adding the multiplicand by the number of bits of the multiplier that are “1”.

<Contents used>

Set the data in the multiplicand (HIKAKE) and multiplier (KAKE) areas, and call subroutine S_KAKERU.

```
        EXTRN   S_KAKERU
        EXTRN   HIKAKE,KAKE,KOTAE
MAIN:                           ; Multiplier
                .
                .
        HIKAKE=WORKA  (A)       ; Stores multiplicand data to multiplicand area
        HIKAKE+1=WORKA+1 (A)    ;
        KAKE=WORKB (A)          ; Stores multiplier data to multiplier area
        KAKE+1=WORKB+1 (A)      ;
        CALL    !S_KAKERU       ; Calls multiplication routine
        HL=#KOTAE               ; HL ← RAM address of operation result storage area
                .               ; Stores result by indirect address transfer
                .
                .
```

Caution Manipulate the data memory in 8-bit units.

(1) Input/output condition

- Input parameter

  HIKAKE : Store the multiplicand data in this area.

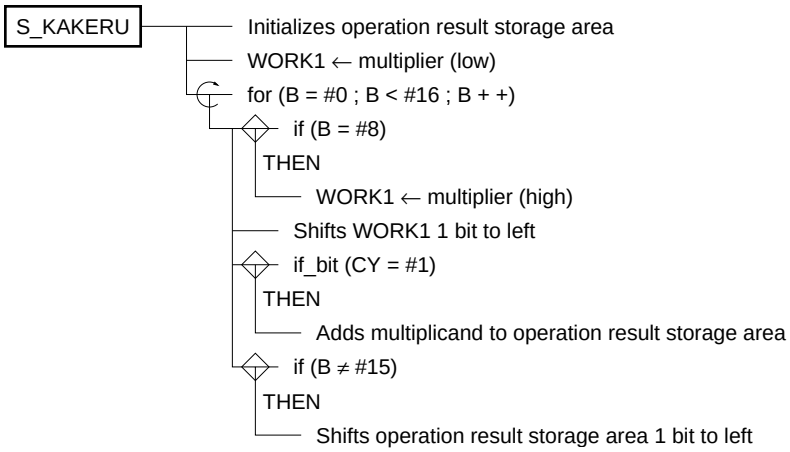
  KAKE : Store the multiplier data in this area.

- Output parameter

  KOTAE : Store the result of the operation in this area.

(2) SPD chart

[Multiplication subroutine]



(3) Registers used

A, B

(4) Program list

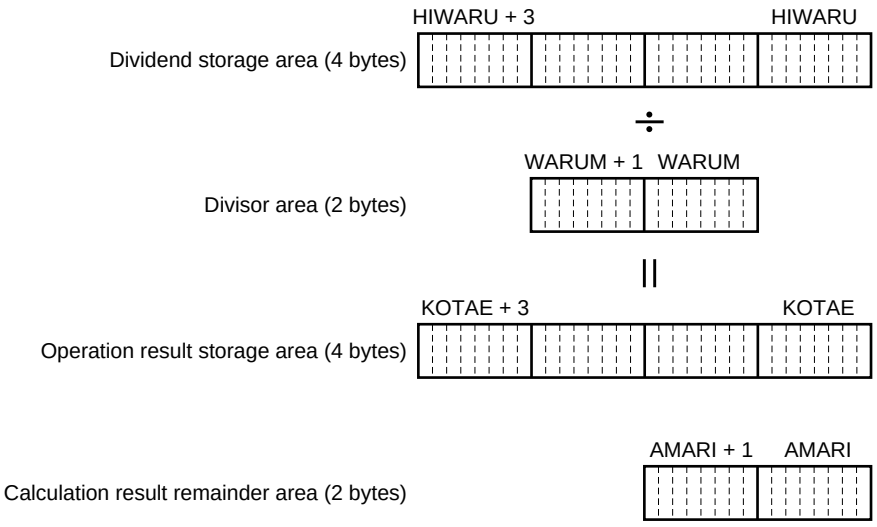
```
$PC(054)
;
PUBLIC HIKAKE,S_KAKERU,KAKE,KOTAE
;
;*********************************************
;            RAM definition
;*********************************************
           DSEG    SADDR
HIKAKE:    DS      2                          ; Multiplicand area
KAKE:      DS      2                          ; Multiplier area
WORK1:     DS      1                          ; Work area
KOTAE:     DS      4                          ; Operation result storage area
;
;*********************************************
;            Multiplication
;*********************************************
       CSEG                                   ;
S_KAKERU:                                     ;
       WORK1=KAKE+1 (A)                       ; Stores multiplier (low) in work area
       KOTAE=#0                               ; Initializes operation result storage area
       KOTAE+1=#0                             ;
       KOTAE+2=#0                             ;
       KOTAE+3=#0                             ;
       for(B=#0;B<#16;B++)(A)                 ; Stores higher multiplier in work area
           if(B == #8)(A)                     ; if low multiplication is completed
               WORK1=KAKE (A)                 ;
               endif                          ;
               A=WORK1                        ; Shifts multiplier 1 bit to left
               CLR1   CY                      ;
               ROLC   A,1                     ;
               WORK1=A                        ;
               if_bit(CY)                     ; Adds multiplicand to operation
                  KOTAE+=HIKAKE (A)           ;    result storage area if carry occurs
                  (KOTAE+1)+=HIKAKE+1,CY (A)  ;
                  (KOTAE+2)+=#0,CY (A)        ;
                  (KOTAE+3)+=#0,CY (A)        ;
               endif                          ;
               if(B != #15) (A)               ;
                   KOTAE+=KOTAE (A)           ; Shifts operation result storage area 1 bit to left
                   KOTAE+1+=KOTAE+1,CY (A)    ;
                   KOTAE+2+=KOTAE+2,CY (A)    ;
                   KOTAE+3+=KOTAE+3,CY (A)    ;
               endif                          ;
           next                               ;
           RET                                ;
           END
```

2.8 Binary Division (32 bits ÷ 16 bits)

Data in a dividend area (HIWARU; 32 bits) is divided by data in a divisor area (WARUM; 16 bits), and the result is stored in an operation result storage area (KOTAE). If a remainder is generated, it is stored in a calculation result reminder area (AMARI).

If division is executed with the divisor being 0, an error occurs.

Figure 2-8. Binary Division



<Processing contents>

The dividend is shifted to the left to the work area starting from the highest digit. If the contents of the work area are greater than the divisor, the divisor is subtracted from the work area, and the least significant bit of the dividend is set to 1. In this way, division is carried out by executing the program by the number of bits of the dividend.

If the divisor is 0, an error flag (F_ERR) is set.

<Usage>

Set data in the dividend area (HIWARU) and divisor area (WARUM), and call subroutine S_WARU.

```
          EXTRN   S_WARU
          EXTRN   HIWARU,WARUM,KOTAE
          EXBIT   F_ERR

  MAIN:
                  .                          ;
                  .                          ;
          HIWARU=WORKA  (A)                  ; Stores dividend data to dividend area
          HIWARU+1=WORKA+1 (A)               ;
          WARUM=WORKB (A)                    ; Stores divisor data to divisor area
          WARUM+1=WORKB+1 (A)                ;
          CALL    !S_WARU                    ; Calls division calculation routine
          HL=#KOTAE                          ; HL ← stores RAM address of operation result storage area
                  .                          ;
                  .                          ;
          if_bit(F_ERR)                      ;
            Calculation error processing     ;
          endif                              ;
                  .
                  .
                  .
```

Caution Manipulate the data memory in 8-bit units.

(1) Input/output conditions

- Input parameter

  HIWARU: Store the dividend data in this area.

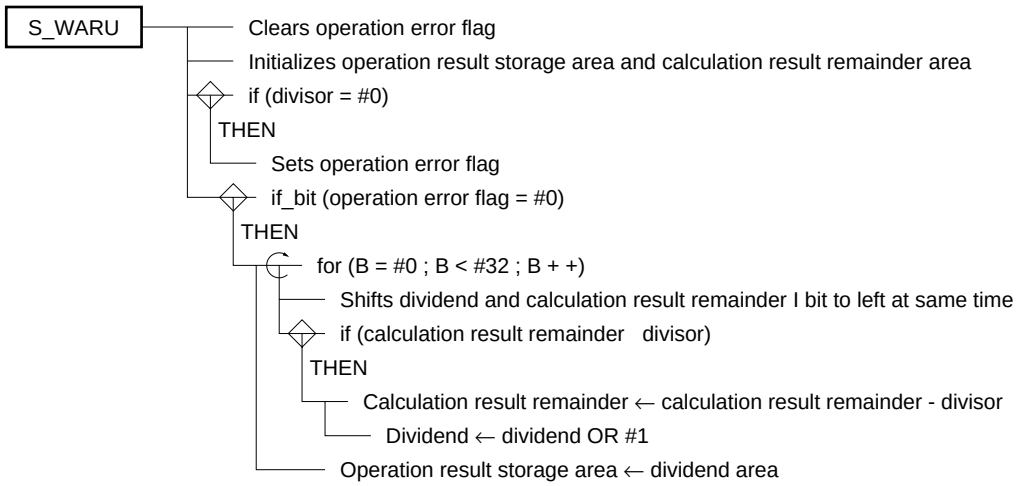
  WARUM : Store the divisor data in this area.

- Output parameter

  KOTAE : Store the result of the calculation in this area.

(2) SPD chart

[Division subroutine]



(3) Registers used

A, B

(4) Program list

```
$PC(054)
;
PUBLIC S_WARU,HIWARU,WARUM,F_ERR
EXTRN  KOTAE
;
;*********************************************
;            RAM definition
;*********************************************
          DSEG    SADDR
HIWARU:   DS      4                             ; Dividend area
WARUM:    DS      2                             ; Divisor area
AMARI:    DS      2                             ; Calculation result remainder storage area
          BSEG
F_ERR     DBIT                                  ; Operation error flag
;*********************************************
;                 Division
;*********************************************
       CSEG                                     ;
S_WARU:                                         ;
       CLR1       F_ERR                         ; Clears operation error flag
       AMARI=#0                                 ; Clears calculation result storage area to 0
       AMARI+1=#0                               ;
       KOTAE=#0                                 ; Clears operation result storage area to 0
       KOTAE+1=#0                               ;
       KOTAE+2=#0                               ;
       KOTAE+3=#0                               ;
       if(WARUM == #0)                          ; Divisor = 0?
           if(WARUM+1 == #0)                    ;
               SET1    F_ERR                    ; Sets operation error flag if divisor is 0
           endif                                ;
       endif                                    ;
       if_bit(!F_ERR)                           ; Operation error?
           for(B=#0;B < #32;B++) (A)            ; Starts 32-bit division
               HIWARU+=HIWARU (A)               ; Shifts dividend and remainder 1 bit to left
               HIWARU+1+=HIWARU+1,CY  (A)       ;
               HIWARU+2+=HIWARU+2,CY  (A)       ;
               HIWARU+3+=HIWARU+3,CY  (A)       ;
               AMARI+=AMARI,CY (A)              ;
               AMARI+1+=AMARI+1,CY (A)          ;
                                                ;
               if(AMARI+1 > WARUM+1) (A)        ; Remainder ≥ divisor?
                   AMARI-=WARUM (A)             ;    Remainder = remainder – divisor
                   AMARI+1-=WARUM+1,CY (A)      ;
                   HIWARU |= #1                 ; Stores 1 to first bit of dividend area
               elseif_bit(Z)                    ;
                   if(AMARI >= WARUM) (A)       ;
                       AMARI-=WARUM(A)          ;
                       AMARI+1-=WARUM+1,CY (A)  ;
                       HIWARU |= #1             ;
                   endif                        ;
               endif                            ;
           next                                 ;
           KOTAE=HIWARU (A)                     ; Stores operation result
           KOTAE+1=HIWARU+1 (A)                 ;
           KOTAE+2=HIWARU+2 (A)                 ;
           KOTAE+3=HIWARU+3 (A)                 ;
       endif                                    ;
       RET                                      ;
       END
```

CHAPTER 3 APPLICATION OF SYSTEM CLOCK SELECTION

The 78K/0 series allows you to select a CPU clock and controls the operation of the oscillator by rewriting the contents of the processor clock control register (PCC), oscillation mode select register (OSMS), and clock select registers 1 and 2 (IECL1 and IECL2).

When the CPU clock is changed, the time shown in Table 3-1 is required since the contents of the PCC have been rewritten until the CPU clock is actually changed. It is therefore not apparent for a while after the contents of the PCC have been rewritten, whether the processor operates on the new or old clock. To stop the main system clock or execute the STOP instruction, therefore, the wait time shown in Table 3-1 is necessary.

Caution IECL1 and IECL2 are provided to the μPD78098, 78098B subseries only.

Table 3-1. Maximum Time Required for Changing CPU Clock

Set Value before Change					Set Value after Change																							
MCS	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0
					0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	1	×	×	×
×	0	0	0	0					8 instructions				4 instructions				2 instructions				1 instruction				1 instruction			
×	0	0	0	1	16 instructions								4 instructions				2 instructions				1 instruction				1 instruction			
×	0	0	1	0	16 instructions				8 instructions								2 instructions				1 instruction				1 instruction			
×	0	0	1	1	16 instructions				8 instructions				4 instructions								1 instruction				1 instruction			
×	0	1	0	0	16 instructions				8 instructions				4 instructions				2 instructions								1 instruction			
1	1	×	×	×	$f_X/2f_{XT}$ instructions (77 instructions)				$f_X/4f_{XT}$ instructions (39 instructions)				$f_X/8f_{XT}$ instructions (20 instructions)				$f_X/16f_{XT}$ instructions (10 instructions)				$f_X/32f_{XT}$ instructions (5 instructions)							
0	1	×	×	×	$f_X/4_{XT}$ instructions (39 instructions)				$f_X/8f_{XT}$ instructions (20 instructions)				$f_X/16f_{XT}$ instructions (10 instructions)				$f_X/32f_{XT}$ instructions (5 instructions)				$f_X/64f_{XT}$ instructions (3 instructions)							

Caution Do not select dividing the CPU clock (PCC0-PCC2) and changing from the main system clock to subsystem clock (by setting CSS to 0 → 1) at the same time.
However, dividing the CPU clock (PCC0-PCC2) can be selected at the same time as changing from the subsystem clock to the main system clock.

Remarks 1. One instruction is the minimum instruction execution time of the CPU clock before change.
2. ( ): f_X = 5.0 MHz, f_{XT} = 32.768 kHz

Figure 3-1. Format of Processor Clock Control Register (μPD78054, 78054Y, 78064, 78064Y, 78078, 78078Y, 780058, 780058Y, 780308, 780308Y, 78058F, 78058FY, 78064B, 78075B, 78075BY subseries, μPD78070A, 78070AY)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
PCC	MCC	FRC	CLS	CSS	0	PCC2	PCC1	PCC0	FFFBH	04H	R/W[Note1]

R/W	CSS	PCC2	PCC1	PCC0	Selects CPU clock (f_{CPU})		
						MCS = 1	MCS = 0
	0	0	0	0	f_{XX}	f_X	$f_X/2$
		0	0	1	$f_{XX}/2$	$f_X/2$	$f_X/2^2$
		0	1	0	$f_{XX}/2^2$	$f_X/2^2$	$f_X/2^3$
		0	1	1	$f_{XX}/2^3$	$f_X/2^3$	$f_X/2^4$
		1	0	0	$f_{XX}/2^4$	$f_X/2^4$	$f_X/2^5$
	1	0	0	0	$f_{XT}/2$		
		0	0	1			
		0	1	0			
		0	1	1			
		1	0	0			
	Others				Setting prohibited		

R	CLS	Status of CPU clock
	0	Main system clock
	1	Subsystem clock

R/W	FRC	Selects feedback resistor of subsystem clock
	0	Uses internal feedback resistor
	1	Does not use internal feedback resistor

R/W	MCC	Controls oscillation of main system clock[Note 2]
	0	Enables oscillation
	1	Stops oscillation

Notes 1. Bit 5 is a read-only bit.

2. Use MCC to stop the oscillation of the main system clock when the CPU operates on the subsystem clock. Do not use the STOP instruction.

Caution Be sure to clear bit 3 to 0.

Remarks 1. f_{XX} : main system clock frequency (f_X or $f_X/2$)

2. f_X : main system clock oscillation frequency

3. f_{XT} : subsystem clock oscillation frequency

4. MCS: bit 0 of oscillation mode select register (OSMS)

Figure 3-2. Format of Processor Clock Control Register (μPD78083 subseries)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
PCC	0	0	0	0	0	PCC2	PCC1	PCC0	FFFBH	04H	R/W

PCC2	PCC1	PCC0	Selects CPU clock (f_{CPU})		
				MCS = 1	MCS = 0
0	0	0	f_{XX}	f_X	$f_X/2$
0	0	1	$f_{XX}/2$	$f_X/2$	$f_X/2^2$
0	1	0	$f_{XX}/2^2$	$f_X/2^2$	$f_X/2^3$
0	1	1	$f_{XX}/2^3$	$f_X/2^3$	$f_X/2^4$
1	0	0	$f_{XX}/2^4$	$f_X/2^4$	$f_X/2^5$
Others			Setting prohibited		

Caution Be sure to clear bits 3 through 7 to 0.

Remarks 1. f_{XX} : main system clock frequency (f_X or $f_X/2$)

2. f_X : main system clock oscillation frequency

3. MCS: bit 0 of oscillation mode select register (OSMS)

Figure 3-3. Format of Processor Clock Control Register (μPD78098, 78098B subseries)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
PCC	MCC	FRC	CLS	CSS	0	PCC2	PCC1	PCC0	FFFBH	04H	R/W[Note 1]

R/W	CSS	PCC2	PCC1	PCC0	Selects CPU clock (f_{CPU})
	0	0	0	0	f_{XX}
		0	0	1	$f_{XX}/2$
		0	1	0	$f_{XX}/2^2$
		0	1	1	$f_{XX}/2^3$
		1	0	0	$f_{XX}/2^4$
	1	0	0	0	$f_{XT}/2$
		0	0	1	
		0	1	0	
		0	1	1	
		1	0	0	
	Others				Setting prohibited

R	CLS	Status of CPU clock
	0	Main system clock
	1	Subsystem clock

R/W	FRC	Selects feedback resistor of subsystem clock
	0	Uses internal feedback resistor
	1	Does not use internal feedback resistor

R/W	MCC	Controls oscillation of main system clock[Note 2]
	0	Enables oscillation
	1	Stops oscillation

Notes 1. Bit 5 is a read-only bit.

2. Use MCC to stop the oscillation of the main system clock when the CPU operates on the subsystem clock. Do not use the STOP instruction.

Caution Be sure to clear bit 3 to 0.

Remarks 1. f_{XX}: main system clock frequency

2. f_{XT}: subsystem clock oscillation frequency

★ **Figure 3-4. Format of Processor Clock Control Register (μPD780018, 780018Y subseries)**

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
PCC	MCC	FRC	CLS	CSS	0	PCC2	PCC1	PCC0	FFFBH	04H	R/W[Note 1]

R/W	CSS	PCC2	PCC1	PCC0	Selects CPU clock (f_{CPU})	
	0	0	0	0	f_{XX}	f_X
		0	0	1	$f_{XX}/2$	$f_X/2$
		0	1	0	$f_{XX}/2^2$	$f_X/2^2$
		0	1	1	$f_{XX}/2^3$	$f_X/2^3$
		1	0	0	$f_{XX}/2^4$	$f_X/2^4$
	1	0	0	0	$f_{XT}/2$	
		0	0	1		
		0	1	0		
		0	1	1		
		1	0	0		
	Others				Setting prohibited	

R	CLS	Status of CPU clock
	0	Main system clock
	1	Subsystem clock

R/W	FRC	Selects feedback resistor of subsystem clock
	0	Uses internal feedback resistor
	1	Does not use internal feedback resistor

R/W	MCC	Controls oscillation of main system clock[Note 2]
	0	Enables oscillation
	1	Stops oscillation

Notes 1. Bit 5 is a read-only bit.

2. Use MCC to stop the oscillation of the main system clock when the CPU operates on the subsystem clock. Do not use the STOP instruction.

Caution Be sure to clear bit 3 to 0.

Remarks 1. f_{XX}: main system clock frequency (f_X)

2. f_X : main system clock oscillation frequency

3. f_{XT}: subsystem clock oscillation frequency

Figure 3-5. Format of Oscillation Mode Select Register (μPD78054, 78054Y, 78064, 78064Y, 78078, 78078Y, 78083, 780058, 780058Y, 780308, 780308Y, 78058F, 78058FY, 78064B, 78075B, 78075BY subseries, μPD78070A, 78070AY)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
OSMS	0	0	0	0	0	0	0	MCS	FFF2H	00H	W

MCS	Controls divider circuit of main system clock
0	Uses divider circuit
1	Does not use divider circuit

Cautions 1. When an instruction that writes a value to the OSMS is executed (including when the instruction is executed to write the same value), the main system clock cycle is extended up to 2/f_X only during the execution of the write instruction.
Consequently, a temporary error of the count clock cycle of the peripheral hardware units that operate on the main system clock, such as timers, occurs.
When the oscillation mode is changed, the clock supplied to the peripheral hardware, as well as the clock supplied to the CPU, is changed.
It is therefore recommended that you execute the instruction to write the OSMS only once after the reset signal has been deasserted, and before the peripheral hardware operates.
2. Set 1 to MCS after V_{DD} has risen to 2.7 V or more.

Figure 3-6. Format of Oscillation Mode Select Register (μPD78098, 78098B subseries)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
OSMS	0	0	0	0	0	0	0	MCS	FFF2H	00H	W

MCS	Controls divider circuit of main system clock (divider circuit 1)
0	Uses 1/2 divider circuit in divider circuit 1
1	Does not use 1/2 divider circuit in divider circuit 1

Caution When an instruction that writes a value to the OSMS is executed (including when the instruction is executed to write the same value), the main system clock cycle is extended up to 2/f_X only during the execution of the write instruction.
Consequently, `rary error of the count clock cycle of the peripheral hardware units that operate on the main system clock, such as timers, occurs.
When the oscillation mode is changed, the clock supplied to the peripheral hardware, as well as the clock supplied to the CPU, is changed.
It is therefore recommended that you execute the instruction to write the OSMS only once after the reset signal has been deasserted, and before the peripheral hardware operates.

★ **Figure 3-7. Format of Oscillation Mode Select Register (μPD780018, 780018Y subseries)**

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
OSMS	0	0	0	0	0	0	0	MCS	FFF2H	00H	W

MCS	Controls divider circuit of main system clock
0	Setting prohibited
1	Does not use divider circuit

Cautions 1. When an instruction that writes a value to the OSMS is executed (including when the instruction is executed to write the same value), the main system clock cycle is extended up to 2/fx only during the execution of the write instruction.
Consequently, a temporary error of the count clock cycle of the peripheral hardware units that operate on the main system clock, such as timers, occurs.
2. Setting MCS to 0 is prohibited. On $\overline{\text{RESET}}$ input, however, OSMS is reset to 00H. Therefore, be sure to set MCS to 1 at the start of a program or after clearing reset.

Figure 3-8. Format of Clock Select Register 1 (μPD78098, 78098B subseries)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
IECL1	0	0	0	0	0	0	IECL11	IECL10	F8E0H	00H	R/W

IECL10	Controls divider circuit of main system clock (divider circuit 1)
0	Uses 2/3 divider circuit in divider circuit 1
1	Does not use 2/3 divider circuit in divider circuit 1

IECL11	Controls divider circuit of main system clock (divider circuit 2)
0	Uses 1/2 divider circuit in divider circuit 2
1	Does not use1/2 divider circuit in divider circuit 2

Caution Be sure to clear bits 2 through 7 to 0.

Figure 3-9. Format of Clock Select Register 2 (μPD78098, 78098B subseries)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
IECL2	0	0	0	0	0	0	0	IECL20	F8E1H	00H	R/W

IECL20	Controls divider circuit of main system clock (divider circuit 1)
0	Uses 1/3 divider circuit in divider circuit 1
1	Does not use 1/3 divider circuit in divider circuit 1

Caution Be sure to clear bits 1 through 7 to 0.

The fastest instruction is executed in two CPU clocks. Therefore, the relation between the CPU clock (f_{CPU}) and minimum instruction execution time is as shown in Tables 3-2 and 3-3.

★ **Table 3-2. Relation between CPU Clock and Minimum Instruction Execution Time (other than μPD78098 and 78098B subseries)**

CPU Clock (f_{CPU})	Minimum Instruction Execution Time: $2/f_{CPU}$
f_X	0.4 μs
$f_X/2$	0.8 μs
$f_X/2^2$	1.6 μs
$f_X/2^3$	3.2 μs
$f_X/2^4$	6.4 μs
$f_X/2^5$ **Note 1**	12.8 μs
f_{XT} **Note 2**	122 μs

Notes 1. Except μPD780018 and 780018Y subseries

2. Except μPD78083 subseries

Remark f_X = 5.0 MHz, f_{XT} = 32.768 kHz

f_X : Main system clock oscillation frequency

f_{XT}: Subsystem clock oscillation frequency

✦ **Table 3-3. CPU Clock (f_{CPU}) List (μPD78098 and 78098B Subseries)**

CSS	PCC2	PCC1	PCC0	Selects CPU clock (f_{CPU})								
				MCS	0	0	0	0	1	1	1	1
				IECL20	0	0	1	1	0	0	1	1
				IECL10	0	1	0	1	0	1	0	1
0	0	0	0	f_{XX}	$f_X/2$ (0.67 μs)	$(2f_X/3)/2$ (1.00 μs)	$(f_X/3)/2$ (2.00 μs)	$(2f_X/9)/2$ (3.00 μs)	f_X (Setting prohibited)	$2f_X/3$ (0.50 μs)	$f_X/3$ (1.00 μs)	$2f_X/9$ (1.50 μs)
	0	0	1	$f_{XX}/2$	$f_X/2^2$ (1.33 μs)	$(2f_X/3)/2^2$ (2.00 μs)	$(f_X/3)/2^2$ (4.00 μs)	$(2f_X/9)/2^2$ (6.00 μs)	$f_X/2$ (0.67 μs)	$(2f_X/3)/2$ (1.00 μs)	$(f_X/3)/2$ (2.00 μs)	$(2f_X/9)/2$ (3.00 μs)
	0	1	0	$f_{XX}/2^2$	$f_X/2^3$ (2.67 μs)	$(2f_X/3)/2^3$ (4.00 μs)	$(f_X/3)/2^3$ (8.00 μs)	$(2f_X/9)/2^3$ (12.0 μs)	$f_X/2^2$ (1.33 μs)	$(2f_X/3)/2^2$ (2.00 μs)	$(f_X/3)/2^2$ (4.00 μs)	$(2f_X/9)/2^2$ (6.00 μs)
	0	1	1	$f_{XX}/2^3$	$f_X/2^4$ (5.33 μs)	$(2f_X/3)/2^4$ (8.00 μs)	$(f_X/3)/2^4$ (16.0 μs)	$(2f_X/9)/2^4$ (24.0 μs)	$f_X/2^3$ (2.67 μs)	$(2f_X/3)/2^3$ (4.00 μs)	$(f_X/3)/2^3$ (8.00 μs)	$(2f_X/9)/2^3$ (12.0 μs)
	1	0	0	$f_{XX}/2^4$	$f_X/2^5$ (10.7 μs)	$(2f_X/3)/2^5$ (16.0 μs)	$(f_X/3)/2^5$ (32.0 μs)	$(2f_X/9)/2^5$ (48.0 μs)	$f_X/2^4$ (5.33 μs)	$(2f_X/3)/2^4$ (8.00 μs)	$(f_X/3)/2^4$ (16.0 μs)	$(2f_X/9)/2^4$ (24.0 μs)
1	0	0	0	$f_{XT}/2$(122 μs)								
	0	0	1									
	0	1	0									
	0	1	1									
	1	0	0									
Others				Setting prohibited								

Remarks 1. f_X : Main system clock oscillation frequency

2. f_{XX} : Main system clock frequency

3. f_{XT} : Subsystem clock oscillation frequency

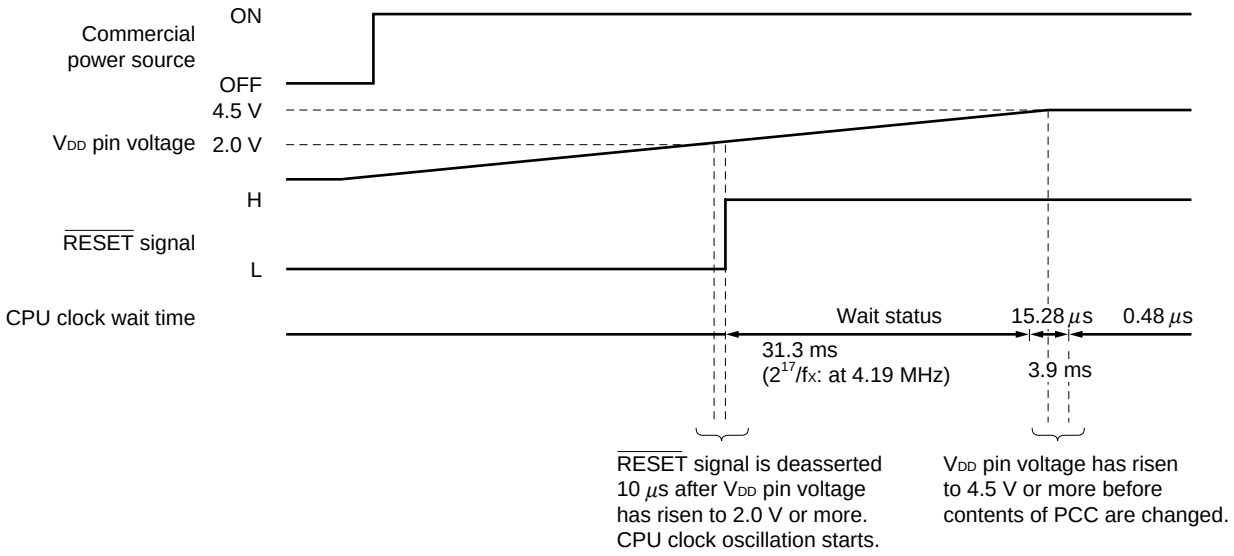
4. ( ) : Minimum instruction execution time with f_X = 6.0 MHz or f_{XT} = 32.768 kHz: $2/f_{CPU}$

3.1 Changing PCC Immediately after $\overline{\text{RESET}}$

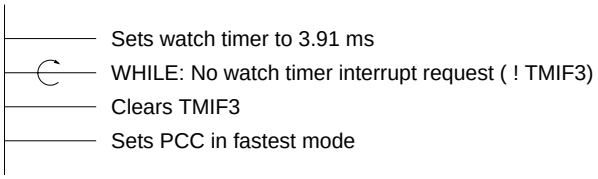
When the $\overline{\text{RESET}}$ signal is asserted, the slowest mode (processor clock control register: PCC = 04H, oscillation mode select register: OSMS = 00H) of the main system clock is selected for the CPU clock. To set the highest speed of the CPU clock, therefore, the contents of the PCC must be rewritten (PCC = 00H, OSMS = 01H). To use the fasted mode, however, the voltage on the V_{DD} pin has to have risen to a sufficient level and be stable.

In the following example, the CPU waits until the V_{DD} pin voltage has risen to the sufficient level by using the watch timer (the interval time is set to 3.91 ms). After that, the CPU operates on the fastest clock.

Figure 3-10. Example of Selecting CPU Clock after $\overline{\text{RESET}}$ (with μPD78054 subseries)



(1) SPD chart



(2) Program list

```
;************************************
;*       Sets wait time
;************************************
       TCL2=#00010000B
       TMC2=#00110110B          ; Sets watch timer to 3.91 ms
       while_bit(!TMIF3)        ; 3.91 ms?
       endw
       CLR1   WTIF
       OSMS=#00000001B          ; Does not use divider circuit
       PCC=#00000000B           ; Sets CPU clock in fastest mode
```

3.2 Selecting Power ON/OFF

The 78K/0 series can operate in an ultra low current consumption mode by using the processor clock control register (PCC) and selecting the subsystem clock. By providing a backup power supply such as a Ni-Cd battery or super capacitor to the system, therefore, the system can continue operating even if a power failure occurs.

In this example, a power failure is detected by using INTP1 (both the rising and falling edges are selected as the edge to be detected), and the contents of the PCC are changed depending on the port level at that time. Figure 3-11 shows a circuit example, and Figure 3-12 shows the system clock changing timing.

Figure 3-11. Example of System Clock Changing Circuit

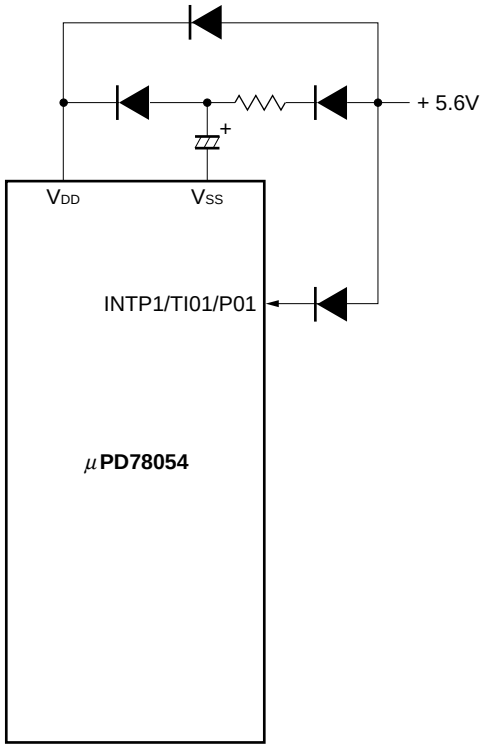
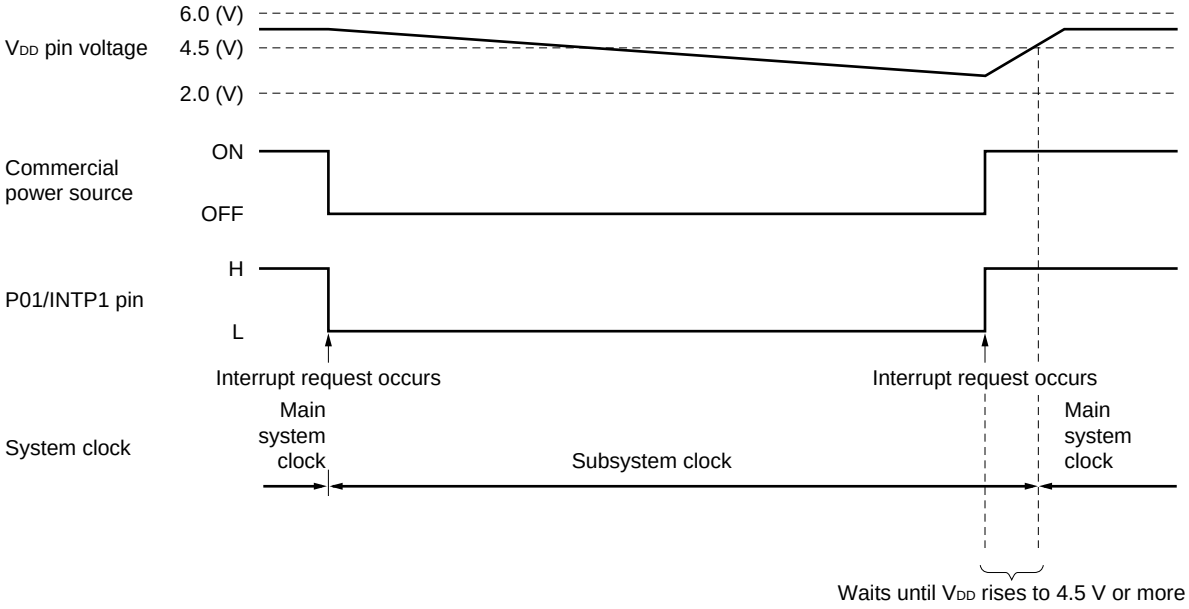
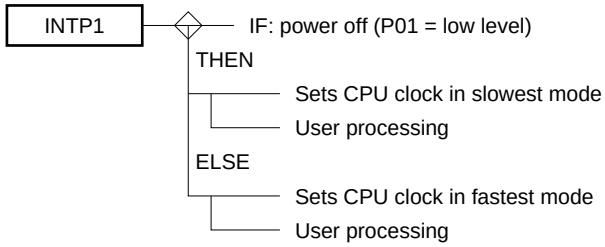


Figure 3-12. Example of Changing System Clock on Power Failure (μPD78054 subseries)



(1) SPD chart



(2) Program list

```
VEP0    CSEG    AT 08H
        DW      INTP1                   ; Sets vector address of INTP1

        MOV     INTM0,#00110000B        ; Both edge detection mode
        CLR1    PMK1
        EI

;*************************************
;*      Sets low-/high-speed mode
;*************************************
INTP1:
        if_bit(!P0.1)
;           Setting of internal hardware (low speed)
;           User processing

            PCC=#10010000B              ; Sets low-speed mode

        else
;           Sets internal hardware (high speed)
;           User processing

            PCC=#00000000B              ; Sets high-speed mode

    endif
    RETI
```

[MEMO]

CHAPTER 4 APPLICATIONS OF WATCHDOG TIMER

The watchdog timer of the 78K/0 series has two modes: watchdog timer mode in which a hang-up of the microcontroller is detected, and interval timer mode.

The watchdog timer is set by using timer clock select register 2 (TCL2) and watchdog timer mode register (WDTM).

Figure 4-1. Format of Timer Clock Select Register 2 (μPD78054 78054Y, 78064, 78064Y, 78078, 78078Y, 780058, 780058Y, 780308, 780308Y, 78058F, 78058FY, 78064B, 78075B, 78075BY subseries, μPD78070A, 78070AY)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
TCL2	TCL27	TCL26	TCL25	TCL24	0	TCL22	TCL21	TCL20	FF42H	00H	R/W

TCL22	TCL21	TCL20	Selects count clock of watchdog timer		
				MCS = 1	MCS = 0
0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)	$f_X/2^4$ (313 kHz)
0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)	$f_X/2^5$ (156 kHz)
0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)	$f_X/2^6$ (78.1 kHz)
0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)
1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.8 kHz)
1	1	0	$f_{XX}/2^9$	$f_X/2^9$ (9.8 kHz)	$f_X/2^{10}$ (4.9 kHz)
1	1	1	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.4 kHz)	$f_X/2^{12}$ (1.2 kHz)

TCL24	Selects count clock of watch timer		
		MCS = 1	MCS = 0
0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	f_{XT} (32.768 kHz)		

TCL27	TCL26	TCL25	Selects frequency of buzzer output		
				MCS = 1	MCS = 0
0	×	×	Disables buzzer output		
1	0	0	$f_{XX}/2^9$	$f_X/2^9$ (9.8 kHz)	$f_X/2^{10}$ (4.9 kHz)
1	0	1	$f_{XX}/2^{10}$	$f_X/2^{10}$ (4.9 kHz)	$f_X/2^{11}$ (2.4 kHz)
1	1	0	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.4 kHz)	$f_X/2^{12}$ (1.2 kHz)
1	1	1	Setting prohibited		

Caution To change the data of TCL2 except when writing the same data, once stop the timer operation.

Remarks 1. f_{XX} : main system clock frequency (f_X or $f_X/2$)
2. f_X : main system clock oscillation frequency
3. f_{XT} : subsystem clock oscillation frequency
4. × : don't care
5. MCS: bit 0 of oscillation mode select register (OSMS)
6. ( ) : f_X = 5.0 MHz or f_{XT} = 32.768 kHz

Figure 4-2. Format of Timer Clock Select Register 2 (μPD78083 subseries)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
TCL2	TCL27	TCL26	TCL25	0	0	TCL22	TCL21	TCL20	FF42H	00H	R/W

TCL22	TCL21	TCL20	Selects count clock of watchdog timer		
				MCS = 1	MCS = 0
0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)	$f_X/2^4$ (313 kHz)
0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)	$f_X/2^5$ (156 kHz)
0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)	$f_X/2^6$ (78.1 kHz)
0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)
1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.8 kHz)
1	1	0	$f_{XX}/2^9$	$f_X/2^9$ (9.8 kHz)	$f_X/2^{10}$ (4.9 kHz)
1	1	1	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.4 kHz)	$f_X/2^{12}$ (1.2 kHz)

TCL27	TCL26	TCL25	Selects frequency of buzzer output		
				MCS = 1	MCS = 0
0	×	×	Disables buzzer output		
1	0	0	$f_{XX}/2^9$	$f_X/2^9$ (9.8 kHz)	$f_X/2^{10}$ (4.9 kHz)
1	0	1	$f_{XX}/2^{10}$	$f_X/2^{10}$ (4.9 kHz)	$f_X/2^{11}$ (2.4 kHz)
1	1	0	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.4 kHz)	$f_X/2^{12}$ (1.2 kHz)
1	1	1	Setting prohibited		

Cautions 1. To change the data of TCL2 except when writing the same data, once stop the timer operation.

2. Be sure to clear bits 3 and 4 to 0.

Remarks 1. f_{XX} : main system clock frequency (f_X or $f_X/2$)

2. f_X : main system clock oscillation frequency

3. × : don't care

4. MCS: bit 0 of oscillation mode select register (OSMS)

5. ( ) : f_X = 5.0 MHz

Figure 4-3. Format of Timer Clock Select Register 2 (μPD78098, 78098B subseries)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
TCL2	TCL27	TCL26	TCL25	TCL24	0	TCL22	TCL21	TCL20	FF42H	00H	R/W

TCL22	TCL21	TCL20	Selects count clock of watchdog timer
0	0	0	$f_{XX}/2^3$ (500 kHz)
0	0	1	$f_{XX}/2^4$ (250 kHz)
0	1	0	$f_{XX}/2^5$ (125 kHz)
0	1	1	$f_{XX}/2^6$ (62.5 kHz)
1	0	0	$f_{XX}/2^7$ (31.3 kHz)
1	0	1	$f_{XX}/2^8$ (15.6 kHz)
1	1	0	$f_{XX}/2^9$ (7.8 kHz)
1	1	1	$f_{XX}/2^{11}$ (2.0 kHz)

TCL24	Selects count clock of watch timer
0	$f_{XX}/2^7$ (31.3 kHz)
1	f_{XT} (32.768 kHz)

TCL27	TCL26	TCL25	Selects frequency of buzzer output
0	×	×	Disables buzzer output
1	0	0	$f_{XX}/2^9$ (7.8 kHz)
1	0	1	$f_{XX}/2^{10}$ (3.9 kHz)
1	1	0	$f_{XX}/2^{11}$ (1.95 kHz)
1	1	1	Setting prohibited

Caution To change the data of TCL2 except when writing the same data, once stop the timer operation.

Remarks 1. f_{XX} : main system clock frequency

2. f_{XT} : subsystem clock oscillation frequency

3. × : don't care

4. ( ): f_{XX} = 4.0 MHz or f_{XT} = 32.768 kHz

★ **Figure 4-4. Format of Timer Clock Select Register 2 (μPD780018, 780018Y subseries)**

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
TCL2	TCL27	TCL26	TCL25	TCL24	0	TCL22	TCL21	TCL20	FF42H	00H	R/W

TCL22	TCL21	TCL20	Selects count clock of watchdog timer	
0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)
0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)
0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)
0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)
1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)
1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)
1	1	0	$f_{XX}/2^9$	$f_X/2^9$ (9.8 kHz)
1	1	1	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.4 kHz)

TCL24	Selects count clock of watch timer	
0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)
1	f_{XT} (32.768 kHz)	

TCL27	TCL26	TCL25	Selects frequency of buzzer output	
0	×	×	Disables buzzer output	
1	0	0	$f_{XX}/2^9$	$f_X/2^9$ (9.8 kHz)
1	0	1	$f_{XX}/2^{10}$	$f_X/2^{10}$ (4.9 kHz)
1	1	0	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.4 kHz)
1	1	1	Setting prohibited	

Caution To change the data of TCL2 except when writing the same data, once stop the timer operation.

Remarks 1. f_{XX} : main system clock frequency (f_X)

2. f_X : main system clock oscillation frequency

3. f_{XT} : subsystem clock oscillation frequency

4. × : don't care

5. ( ): f_X = 5.0 MHz or f_{XT} = 32.768 kHz

★ **Figure 4-5. Format of Watchdog Timer Mode Register**

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

WDTM4	WDTM3	Selects operation mode of watchdog time[Note 1]
0	×	Interval timer mode[Note 2] (maskable interrupt request occurs when overflow occurs)
1	0	Watchdog timer mode (non-maskable interrupt request occurs when overflow occurs)
1	1	Watchdog timer mode 2 (reset operation starts when overflow occurs)

RUN	Selects watchdog timer operation[Note 3]
0	Stops counting
1	Clears counter and starts counting

Notes 1. Once WDTM3 and WDTM4 have been set to 1, they cannot be cleared to 0 by software.

2. When RUN is set to 1, the WDTM starts interval timer operation.

3. Once RUN has been set to 1, it cannot be cleared to 0 by software. Therefore, when counting has been started, it cannot be stopped by any means other than the $\overline{\text{RESET}}$ signal.

Caution 1. When RUN is set to 1 and the watchdog timer is cleared, the actual overflow time is up to 0.5% shorter than the time set by the timer clock select register 2.

2. When using the watchdog timer modes 1 and 2, confirm that the interrupt request flag (TMIF4) is 0 and then set WDTM4 to 1. If WDTM4 is set to 1 while TMIF4 is 1, the non-maskable interrupt occurs regardless of the contents of WDTM3.

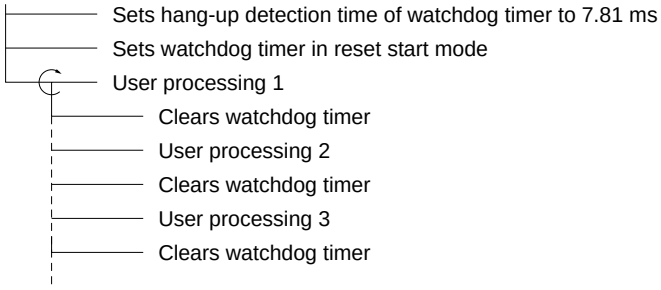
4.1 Setting Watchdog Timer Mode

Reset processing or non-maskable interrupt processing is performed after the watchdog timer has detected a hang-up. You can select which processing is to be performed by the watchdog timer mode register (WDTM). When the watchdog timer mode is used, the timer must be cleared at intervals shorter than the set hang-up detection time. If the timer is not cleared, an overflow occurs, and reset or interrupt processing is executed.

The hang-up detection time of the watchdog timer is set by the timer clock select register 2 (TCL2).

In the following example, the hang-up detection time is set to 7.81 ms and the reset processing is performed when an overflow occurs.

(1) SPD chart



(2) Program list

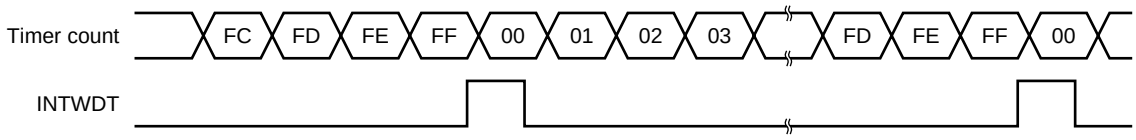
```
;************************************
;*   Sets watchdog timer
;************************************

        OSMS=#00000001B             ; Does not use divider circuit
        TCL2=#00000100B             ; Sets watchdog timer to 7.81 ms
        WDTM=#10011000B             ; Sets reset start mode
;           ¦
        User processing 1
            ¦
            ¦
        SET1    RUN                 ; Clears timer
;           ¦
            ¦
        User processing 2
            ¦
            ¦
        SET1    RUN                 ; Clears timer
;           ¦
            ¦
        User processing 3
            ¦
            ¦
        SET1    RUN                 ; Clears timer
            ¦
            ¦
```

4.2 Setting Interval Timer Mode

When the interval timer mode is used, the interval time is set by the timer clock select register 2 (TCL2) (interval time = 0.488 ms to 125 ms, at f_X = 4.19 MHz). In this mode, an interrupt request flag (TMIF4) is set when an overflow occurs in the timer.

In the following example, three types of times, 0.977 ms, 7.82 ms, and 125 ms, are set.

Figure 4-6. Count Timing of Watchdog Timer



(1) Program list

<1> To set 0.977 ms

```
TCL2=#00000001B      ; Sets 0.977 ms
WDTM=#10001000B      ; Selects interval timer mode
```

<2> To set 7.82 ms

```
TCL2=#00000100B      ; Sets 7.82 ms
WDTM=#10001000B      ; Selects interval timer mode
```

<3> To set 125 ms

```
TCL2=#00000111B      ; Sets 125 ms
WDTM=#10001000B      ; Selects interval timer mode
```

Remark The above interval time is the value when OSMS = 01H.

[MEMO]

CHAPTER 5 APPLICATIONS OF 16-BIT TIMER/EVENT COUNTER

The 16-bit timer/event counter of the 78K/0 series has the following six functions:

- Interval timer
- PWM output
- Pulse width measurement
- External event counter
- Square wave output
- One-shot pulse output

The 16-bit timer/event counter is set by the following registers:

- Timer clock select register 0 (TCL0)
- 16-bit timer mode control register (TMC0)
- Capture/compare control register 0 (CRC0)
- 16-bit timer output control register (TOC0)
- Port mode register 3 (PM3)
- External interrupt mode register (INTM0)
- Sampling clock select register (SCS)

Figure 5-1. Format of Timer Clock Select Register 0 (μPD78054, 78054Y, 78064, 78064Y, 78078, 78078Y, 780058, 780058Y, 780308, 780308Y, 78058F, 78058FY, 78064B, 78075B, 78075BY subseries, μPD78070A, 78070AY)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
TCL0	CLOE	TCL06	TCL05	TCL04	TCL03	TCL02	TCL01	TCL00	FF40H	00H	R/W

TCL03	TCL02	TCL01	TCL00	Selects clock of PCL output		
					MCS = 1	MCS = 0
0	0	0	0	f_{XT} (32.768 kHz)		
0	1	0	1	f_{XX}	f_X (5.0 MHz)	$f_X/2$ (2.5 MHz)
0	1	1	0	$f_{XX}/2$	$f_X/2$ (2.5 MHz)	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$ (1.25 MHz)	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
Others				Setting prohibited		

TCL06	TCL05	TCL04	Selects count clock of 16-bit timer register		
				MCS = 1	MCS = 0
0	0	0	TI00 (valid edge can be specified)		
0	0	1	$2f_{XX}$	Setting prohibited	f_X (5.0 MHz)
0	1	0	f_{XX}	f_X (5.0 MHz)	$f_X/2$ (2.5 MHz)
0	1	1	$f_{XX}/2$	$f_X/2$ (2.5 MHz)	$f_X/2^2$ (1.25 MHz)
1	0	0	$f_{XX}/2^2$	$f_X/2^2$ (1.25 MHz)	$f_X/2^3$ (625 kHz)
1	1	1	Watch timer output (INTTM3)		
Others			Setting prohibited		

CLOE	Controls PCL output
0	Disables output
1	Enables output

Cautions 1. The valid edge of the TI00/INTP0 pin is specified by the external interrupt mode register 0 (INTM0). The frequency of the sampling clock is selected by the sampling clock select register (SCS).

2. To enable PCL output, set TCL00 through TCL03, and then set CLOE to 1 by using a 1-bit memory manipulation instruction.

3. Read the count value from TM0, not from the capture/compare register 01(CR01), when TI00 is specified as the count clock of TM0.

4. Before writing new data to TCL0, stop the timer operation once.

Remarks 1. f_{XX} : main system clock frequency (f_X or $f_X/2$)

2. f_X : main system clock oscillation frequency

3. f_{XT} : subsystem clock oscillation frequency

4. TI00 : input pin of 16-bit timer/event counter

5. TM0 : 16-bit timer register

6. MCS: bit 0 of oscillation mode select register (OSMS)

7. ( ) : at f_X = 5.0 MHz or f_{XT} = 32.768 kHz

Figure 5-2. Format of Timer Clock Select Register 0 (μPD78098, 78098B subseries)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
TCL0	CLOE	TCL06	TCL05	TCL04	TCL03	TCL02	TCL01	TCL00	FF40H	00H	R/W

TCL03	TCL02	TCL01	TCL00	Selects clock of PCL output
0	0	0	0	f_{XT} (32.768 kHz)
0	1	0	1	f_{XX} (4.0 MHz)
0	1	1	0	$f_{XX}/2$ (2.0 MHz)
0	1	1	1	$f_{XX}/2^2$ (1.0 MHz)
1	0	0	0	$f_{XX}/2^3$ (500 kHz)
1	0	0	1	$f_{XX}/2^4$ (250 kHz)
1	0	1	0	$f_{XX}/2^5$ (125 kHz)
1	0	1	1	$f_{XX}/2^6$ (62.5 kHz)
1	1	0	0	$f_{XX}/2^7$ (31.3 kHz)
Others				Setting prohibited

TCL06	TCL05	TCL04	Selects count clock of 16-bit timer register
0	0	0	TI00 (valid edge can be specified)
0	0	1	$2f_{XX}$ **Note**
0	1	0	f_{XX} (4.0 MHz)
0	1	1	$f_{XX}/2$ (2.0 MHz)
1	0	0	$f_{XX}/2^2$ (1.0 MHz)
1	1	1	Watch timer output (INTTM3)
Others			Setting prohibited

CLOE	Controls PCL output
0	Disables output
1	Enables output

Note At f_{XX} > 2.5 MHz, setting prohibited.

Cautions 1. The valid edge of the TI00/INTP0 pin is specified by the external interrupt mode register 0 (INTM0). The frequency of the sampling clock is selected by the sampling clock select register (SCS).

2. To enable PCL output, set TCL00 through TCL03, and then set CLOE to 1 by using a 1-bit memory manipulation instruction.

3. Read the count value from TM0, not from the capture/compare register 01(CR01), when TI00 is specified as the count clock of TM0.

4. Before writing new data to TCL0, stop the timer operation once.

Remarks 1. f_{XX} : main system clock frequency

2. f_{XT} : subsystem clock oscillation frequency

3. TI00 : input pin of 16-bit timer/event counter

4. TM0 : 16-bit timer register

5. ( ) : at f_{XX} = 4.0 MHz or f_{XT} = 32.768 kHz

★ **Figure 5-3. Format of Timer Clock Select Register 0 (μPD780018, 780018Y subseries)**

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
TCL0	CLOE	TCL06	TCL05	TCL04	TCL03	TCL02	TCL01	TCL00	FF40H	00H	R/W

TCL03	TCL02	TCL01	TCL00	Selects clock of PCL output	
0	0	0	0	f_{XT} (32.768 kHz)	
0	1	0	1	f_{XX}	f_X (5.0 MHz)
0	1	1	0	$f_{XX}/2$	$f_X/2$ (2.5 MHz)
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$ (1.25 MHz)
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)
Others				Setting prohibited	

TCL06	TCL05	TCL04	Selects count clock of 16-bit timer register	
0	0	0	TI00 (valid edge can be specified)	
0	1	0	f_{XX}	f_X (5.0 MHz)
0	1	1	$f_{XX}/2$	$f_X/2$ (2.5 MHz)
1	0	0	$f_{XX}/2^2$	$f_X/2^2$ (1.25 MHz)
1	1	1	Watch timer output (INTTM3)	
Others			Setting prohibited	

CLOE	Controls PCL output
0	Disables output
1	Enables output

Cautions 1. The valid edge of the TI00/INTP0 pin is specified by the external interrupt mode register 0 (INTM0). The frequency of the sampling clock is selected by the sampling clock select register (SCS).

2. To enable PCL output, set TCL00 through TCL03, and then set CLOE to 1 by using a 1-bit memory manipulation instruction.

3. Read the count value from TM0, not from the capture/compare register 01(CR01), when TI00 is specified as the count clock of TM0.

4. Before writing new data to TCL0, stop the timer operation once.

Remarks 1. f_{XX} : main system clock frequency (f_X)

2. f_X : main system clock oscillation frequency

3. f_{XT} : subsystem clock oscillation frequency

4. TI00 : input pin of 16-bit timer/event counter

5. TM0 : 16-bit timer register

6. ( ) : at f_X = 5.0 MHz or f_{XT} = 32.768 kHz

Figure 5-4. Format of 16-Bit Timer Mode Control Register

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
TMC0	0	0	0	0	TMC03	TMC02	TMC01	OVF0	FF48H	00H	R/W

OVF0	Detects overflow of 16-bit timer register
0	Overflow does not occur
1	Overflow occurs

TMC03	TMC02	TMC01	Selects operation mode and clear mode	Selects output timing of TO0	Occurrence of interrupt request
0	0	0	Stops operation (clears TM0 to 0)	Not affected	Does not occur
0	0	1	PWM mode (free running)	PWM pulse output	Occurs if TM0 and CR00 coincide and if TM0 and CR01 coincide
0	1	0	Free running mode	Coincidence between TM0 and CR00 or between TM0 and CR01	
0	1	1		Coincidence between TM0 and CR00, or between TM0 and CR01, or valid edge of TI00	
1	0	0	Clears and starts at valid edge of TI00	Coincidence between TM0 and CR00 or between TM0 and CR01	
1	0	1		Coincidence between TM0 and CR00, or between TM0 and CR01, or valid edge of TI00	
1	1	0	Clears and start at coincidence between TM0 and CR00	Coincidence between TM0 and CR00 or between TM0 and CR01	
1	1	1		Coincidence between TM0 and CR00, or between TM0 and CR01, or valid edge of TI00	

Cautions 1. Before setting the clear mode or changing the output timing of TO0, stop the timer operation (by clearing TMC01 through TMC03 to 0, 0, 0).

2. The valid edge of the TI00/INTP0 pin is selected by the external interrupt mode register 0 (INTM0). The frequency of the sampling clock is selected by the sampling clock select register (SCS).

3. When using the PWM mode, set data to CR00 after setting the PWM mode.

4. When a mode in which the timer is cleared and started on coincidence between TM0 and CR00, the OVF0 flag is set to 1 when the set value of CR00 is FFFFH and the value of TM0 changes from FFFFH to 0000H.

5. The 16-bit timer register starts operating as soon as a value other than 0, 0, 0 (operation stop mode) is set to TMC01 through TMC03. To stop the operation, clear TMC01 through TMC03 to 0, 0, 0.

Remarks 1. TO0 : output pin of 16-bit timer/event counter
2. TI00 : input pin of 16-bit timer/event counter
3. TM0 : 16-bit timer register
4. CR00 : compare register 00
5. CR01 : compare register 01

Figure 5-5. Format of Capture/Compare Control Register

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
CRC0	0	0	0	0	0	CRC02	CRC01	CRC00	FF4CH	04H	R/W

CRC00	Selects operation mode of CR00
0	Operates as compare register
1	Operates as capture register

CRC01	Selects capture trigger of CR00
0	Captures at valid edge of TI01
1	Captures at valid edge of TI00

CRC02	Selects operation mode of CR01
0	Operates as compare register
1	Operates as capture register

Cautions 1. Be sure to stop the timer operation before setting CRC0.
2. When a mode in which the timer is cleared and started on coincidence between TM0 and CR00 is selected by the 16-bit timer mode control register, do not specify CR00 as the capture register.

Figure 5-6. Format of 16-Bit Timer Output Control Register

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
TOC0	0	OSPT	OSPE	TOC04	LVS0	LVR0	TOC01	TOE0	FF4EH	00H	R/W

TOE0	Controls output of 16-bit timer/event counter
0	Disables output (port mode)
1	Enables output

TOC01	PWM mode	Other than PWM mode
	Selects active level	Controls timer output F/F on coincidence between CR00 and TM0
0	Active high	Disables reverse operation
1	Active low	Enables reverse operation

LVS0	LVR0	Sets status of timer output F/F of 16-bit timer/ event counter
0	0	Not affected
0	1	Resets timer output F/F (to 0)
0	0	Sets timer output F/F (to 1)
0	1	Setting prohibited

TOC04	Controls timer output F/F on coincidence between CR01 and TM0
0	Disables reverse operation
0	Enables reverse operation

OSPE	Controls one-shot pulse output operation
0	Successive pulse output
1	One-shot pulse output

OSPT	Controls output trigger of one-shot pulse by software
0	No one-shot pulse trigger
1	One-shot pulse trigger

★ **Cautions 1. Be sure to stop the timer operation before setting TOC0 (except OSPT).**

2. LVS0 and LVR0 are always 0 when they are read immediately after data has been set.

3. OSPT is automatically cleared after data has been set. It is therefore always 0 when read.

Figure 5-7. Format of Port Mode Register 3

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W

PM3n	Selects input/output mode of P3n pin (n = 0-7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

Figure 5-8. Format of External Interrupt Mode Register 0

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
INTM0	ES31	ES30	ES21	ES20	ES11	ES10	0	0	FFECH	00H	R/W

ES11	ES10	Selects valid edge of INTP0
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES21	ES20	Selects valid edge of INTP1
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES31	ES30	Selects valid edge of INTP2
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

★ **Caution Before setting the valid edge of the INTP0/TI00/P00 pin, clear bits 1 through 3 (TMC01 through TMC03) of the 16-bit timer mode control register (TMC0) to 0, 0, 0, and stop the timer.**

Figure 5-9. Format of Sampling Clock Select Register (μPD78054, 78054Y, 78064, 78064Y, 78078, 78078Y, 780058, 780058Y, 780308, 780308Y, 78058F, 78058FY, 78064B, 78075B, 78075BY subseries, μPD78070A, 78070AY)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
SCS	0	0	0	0	0	0	SCS1	SCS0	FF47H	00H	R/W

SCS1	SCS0	Selects sampling clock of INTP0		
			MCS = 1	MCS = 0
0	0	$f_{XX}/2^N$		
0	1	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	0	$f_{XX}/2^5$	$f_X/2^5$ (156.3 kHz)	$f_X/2^6$ (78.1 kHz)
1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)

Caution $f_{XX}/2^N$ is the clock supplied to the CPU, and $f_{XX}/2^5$, $f_{XX}/2^6$, and $f_{XX}/2^7$ are the clocks supplied to the peripheral hardware. $f_{XX}/2^N$ is stopped in the HALT mode.

Remarks 1. N : Value (N = 0 to 4) set to the bits 0 through 2 (PCC0 through PCC2) of the processor clock control register (PCC)

2. f_{XX} : main system clock frequency (f_X or $f_X/2$)

3. f_X : main system clock oscillation frequency

4. MCS: bit 0 of oscillation mode select register (OSMS)

5. ( ) : at f_X = 5.0 MHz

Figure 5-10. Format of Sampling Clock Select Register (μPD78098, 78098B subseries)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
SCS	0	0	0	0	0	0	SCS1	SCS0	FF47H	00H	R/W

SCS1	SCS0	Selects sampling clock of INTP0
0	0	$f_{XX}/2^N$
0	1	$f_{XX}/2^7$ (31.3 kHz)
1	0	$f_{XX}/2^5$ (125.0 kHz)
1	1	$f_{XX}/2^6$ (62.5 kHz)

Caution $f_{XX}/2^N$ is the clock supplied to the CPU, and $f_{XX}/2^5$, $f_{XX}/2^6$, and $f_{XX}/2^7$ are the clocks supplied to the peripheral hardware. $f_{XX}/2^N$ is stopped in the HALT mode.

Remarks 1. N : Value (N = 0 to 4) set to the bits 0 through 2 (PCC0 through PCC2) of the processor clock control register (PCC)

2. f_{XX}: main system clock frequency

3. ( ) : at f_{XX} = 4.0 MHz

Figure 5-11. Format of Sampling Clock Select Register (μPD780018, 780018Y subseries)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
SCS	0	0	0	0	0	0	SCS1	SCS0	FF47H	00H	R/W

SCS1	SCS0	Selects sampling clock of INTP0	
0	0	$f_{XX}/2^N$	
0	1	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)
1	0	$f_{XX}/2^5$	$f_X/2^5$ (156.3 kHz)
1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)

Caution $f_{XX}/2^N$ is the clock supplied to the CPU, and $f_{XX}/2^5$, $f_{XX}/2^6$, and $f_{XX}/2^7$ are the clocks supplied to the peripheral hardware. $f_{XX}/2^N$ is stopped in the HALT mode.

Remarks 1. N : Value (N = 0 to 4) set to the bits 0 through 2 (PCC0 through PCC2) of the processor clock control register (PCC)

2. f_{XX}: main system clock frequency (f_X)

3. f_X : main system clock oscillation frequency

4. ( ) : at f_X = 5.0 MHz

5.1 Setting of Interval Timer

To set the 16-bit timer/event counter as an interval timer, first set the timer clock select register 0 (TCL0) and the 16-bit timer mode control register (TMC0). The clear mode of the 16-bit timer is set by TMC0 and the interval time is set by TCL0.

After that, set the value of the compare register (CR00) from the setup time and count clock. Determine the setup time by using the following expression:

Setup time = (Compare register value + 1) × Count clock cycle

This section shows two examples of setup times of the interval timer: 10 ms and 50 ms.

(a) Interval of 10 ms

<1> Setting of TMC0

Selects a mode in which the timer is cleared and started on coincidence between TM0 and CR00.

<2> Setting of TCL0

Select the f_{XX} mode in which an interval time of 10 ms or more can be set and the resolution is the highest (OSMS = 01H).

<3> Setting of CR00

$$10\ \text{ms} = (N + 1) \times \frac{1}{4.19\ \text{MHz}}$$

$$N = 10\ \text{ms} \times 4.19\ \text{MHz} - 1 \fallingdotseq 4.1899$$

(1) Program list

```
OSMS = #00000001B; Does not use divider circuit
CRC0 = #00000000B ; Selects CR00 as compare register
CR00 = #41899
TCL0 = #00100000B ; Selects count clock fXX
TMC0 = #00001100B ; Clears and starts 16-bit timer/event counter when TM0 and CR00 coincide
```

(b) Interval of 50 ms

<1> Setting of TMC0

Selects a mode in which the timer is cleared and started on coincidence between TM0 and CR00.

<2> Setting of TCL0

Select the $f_{XX}/2^2$ mode in which an interval time of 50 ms or more can be set and the resolution is the highest (OSMS = 01H).

<3> Setting of CR00

$$50 \text{ ms} = (N + 1) \times \frac{1}{4.19 \text{ MHz}/2^2}$$

$$N = 50 \text{ ms} \times 4.19 \text{ MHz}/2^2 - 1 \fallingdotseq .52374$$

(1) Program list

```
OSMS = #00000001B ; Does not use divider circuit
CRC0 = #00000000B ; Selects CR00 as compare register
CR00 = #52374
TCL0 = #01000000B ; Selects count clock fXX/2^2
TMC0 = #00001100B ; Clears and starts 16-bit timer/event counter when TM0 and CR00 coincide
```

5.2 PWM Output

When using the 16-bit timer/event counter in the PWM output mode, set the PWM mode by the 16-bit timer mode control register (TMC0) and enables the output of the 16-bit timer/event counter by the 16-bit timer output control register (TOC0).

The pulse width (active level) of PWM is determined by the value set to the capture/compare register 00 (CR00). Because the PWM of the 78K/0 series has a resolution of 14 bits, however, bits 2 through 15 of CR00 are valid (clear bits 0 and 1 of CR00 to '0, 0').

In the example below, the basic cycle of the PWM mode is set to 61.0 μs ($\frac{1}{f_{XX}} \times 2^8$) and the low level is selected as the active level. The high-order 4 bits of the pulse width are rewritten depending on the value of the parameter (00H to FFH). Therefore, in the following application example, PWM output can be performed in 16 steps (CR00 = 0FFCH to FFFCH).

(1) Description of package

<Public declaration symbol>

PWM : PWM output subroutine name
PWMOUT: input parameter of PWM active level

<Registers used>

AX

<RAM used>

Name	Usage	Attribute	Bytes
PWMOUT	Sets PWM active level	SADDR	1

<Nesting>

1 level 2 bytes

<Hardware units used>

- 16-bit timer/event counter
- P30/TO0

<Initial setting>

- OSMS = #00000001B ; Oscillation mode select register: does not use divider circuit
- Setting of 16-bit timer/event counter
  CRC0 = #00000000B ; Selects CR00 as compare register
  TMC0 = #00000010B ; PWM output mode
  TCL0 = #00100000B ; PWM basic cycle: 61.0 μs
  TOC0 = #00000011B ; Low-active output
- PM30 = 0 ; P30 output mode
- P30 = 0 ; P30 output latch

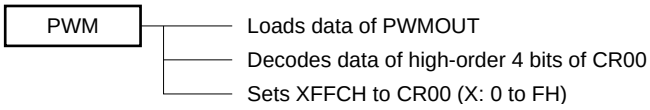
<Starting>

After setting data to PWMOUT in RAM, call subroutine PWM.

(2) Example of use

```
EXTRN     PWM, PWMOUT
  ⋮
OSMS = #00000001B     ; Does not use divider circuit
CRC0 = #00000000B     ; Selects CR00 as compare register
TOC0 = #00000011B     ; Sets low-active PWM output
TCL0 = #00100000B     ; Selects count clock fXX
TMC0 = #00000010B     ; Sets PWM mode
  ⋮
PWMOUT = A            ; Sets input parameter of active level
CALL      !PWM
```

(3) SPD chart



(4) Program list

```
        PUBLIC  PWM,PWMOUT
PWM_DAT DSEG    SADDR
PWMOUT: DS      1                       ; PWM output data area (0-15)
;*********************************
;*      PWM output (16 steps)
;*********************************
P0_SEG  CSEG
PWM:
        A=PWMOUT                        ; Loads high-order data of PWMOUT
        A<<=1
        A<<=1
        A<<=1
        A<<=1
        A|=#0FH                         ; Sets low-order 12 bits to 0FFCH
        X=#0FCH
        CR00=AX
        RET
```

5.3 Remote Controller Signal Reception

This section introduces two examples of programs that receives signals from a remote controller by using the 16-bit timer/event counter.

- The counter is cleared each time the valid edge of the remote controller signal has been detected, and measures a pulse width from the timer count value (capture register CR01) when the next valid edge has been detected.
- The timer operates in the free running mode to measure a pulse width from the difference of the counter between valid edges. PWM output is also performed at the same time.

The remote controller signal is received by a PIN receiver diode and is input to the P00/TI00/INTP0 pin via receive amplifier μPC1490. Figure 5-12 shows an example of a remote controller signal receiver circuit, and Figure 5-13 shows the format of the remote controller signal.

Figure 5-12. Example of Remote Controller Signal Receiver Circuit

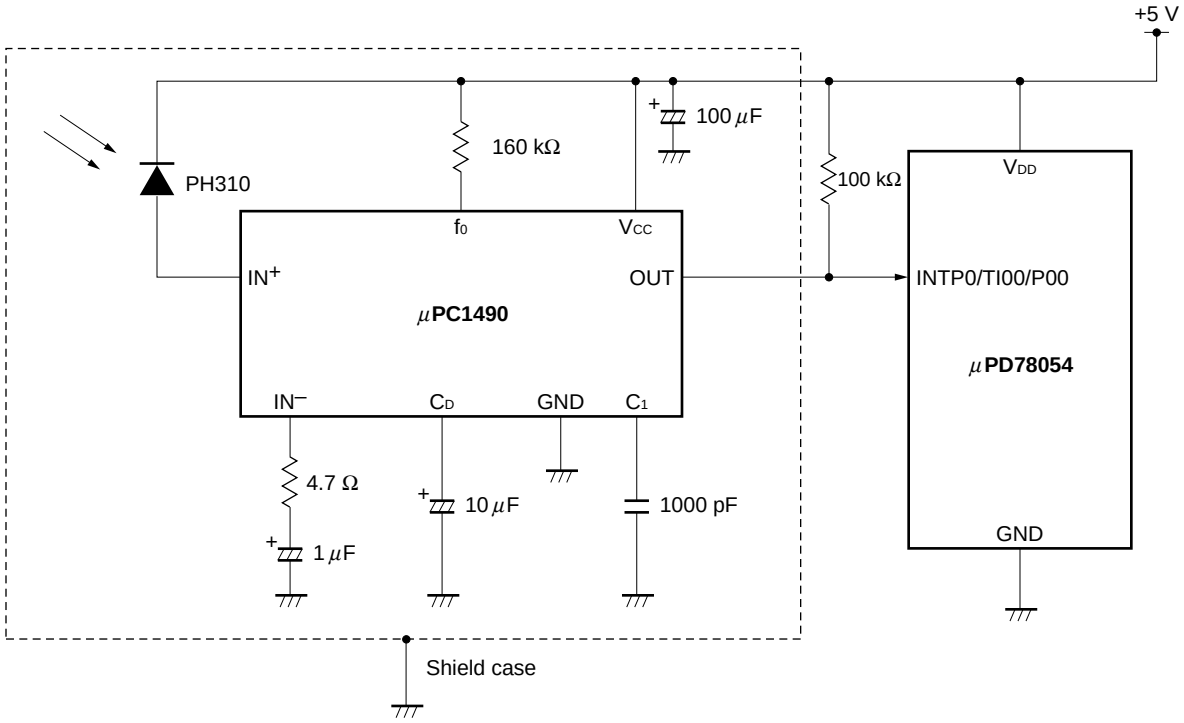
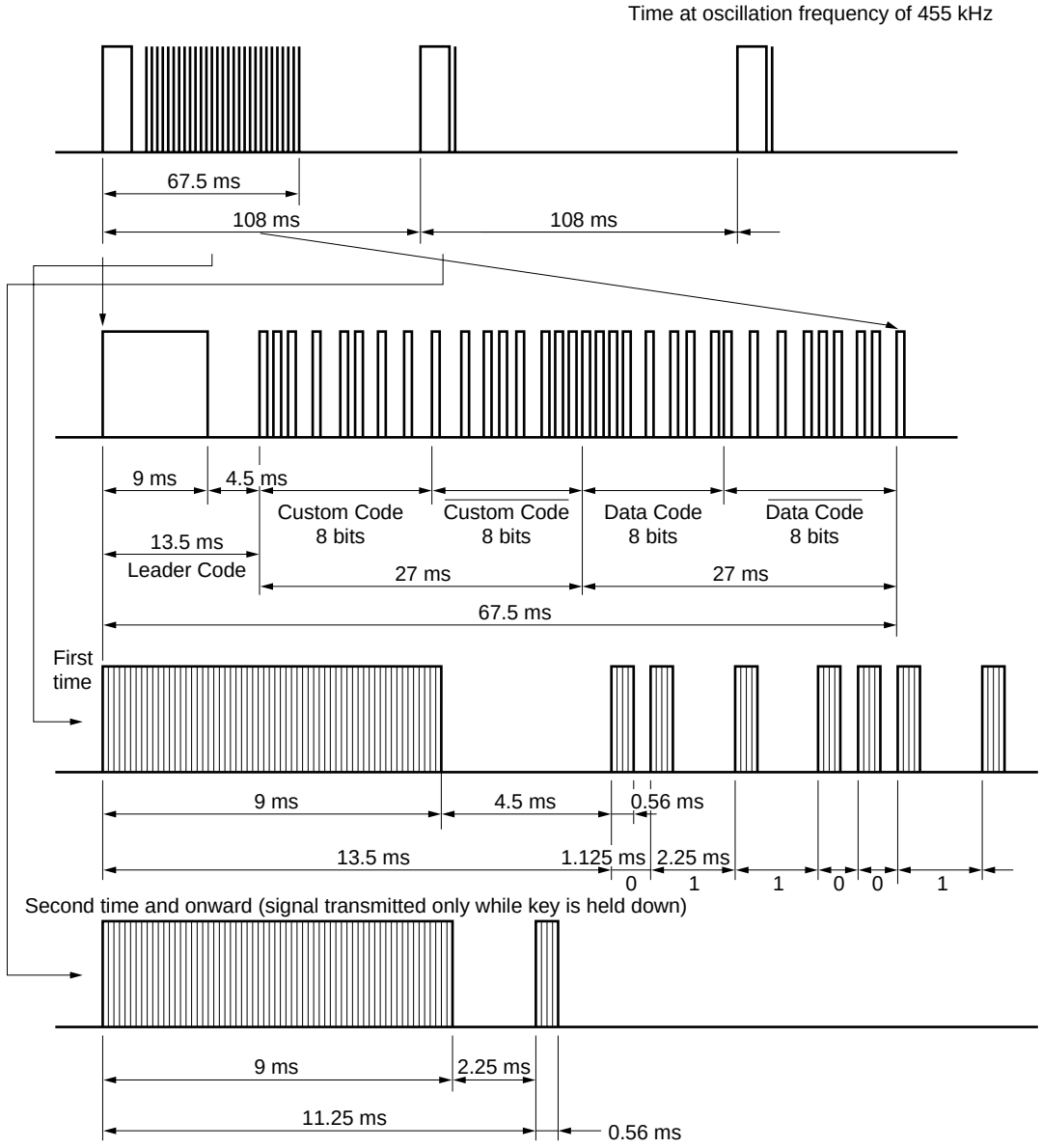
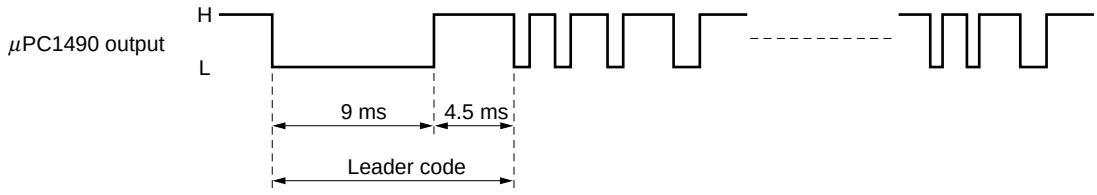


Figure 5-13. Remote Controller Signal Transmitter IC Output Signal



Because the receiver preamplifier μPC1490 used in the circuit example on the previous page is low-active, the level input to the μPD78054 subseries is the inverted data of the remote controller transmit data.

Figure 5-14. Output Signal of Receiver Preamplifier



5.3.1 Remote controller signal reception by counter clearing

Table 5-1 shows the valid pulse width for receiving a remote controller signal in the program example shown in this section, and <1> through <6> describes how to process each signal. The repeat signal of the remote controller signal is valid only within 250 ms after a valid signal has been input. If a signal input within 3 ms after the normal data has been loaded, the data is invalid.

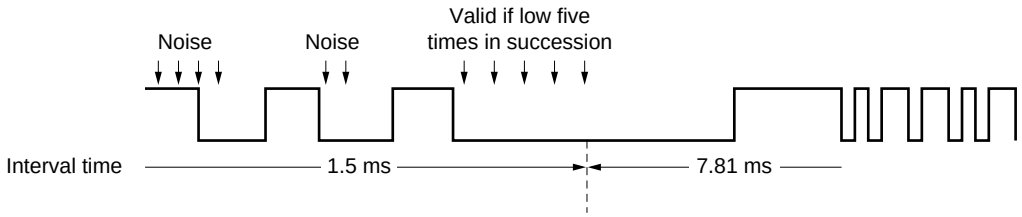
Table 5-1. Valid Time of Input Signal

Signal Name		Output Time	Valid Time
Leader code (low)		9 ms	6.8 ms-11.8 ms
Leader code (high)	Normal	4.5 ms	3 ms-5 ms
	Repeat	2.25 ms	1.8 ms-3 ms
Custom/data code	0	1.125 ms	0.5 ms-1.8 ms
	1	2.25 ms	1.8 ms-2.5 ms

<1> Leader code (low)

The interval time of the 16-bit timer/event counter is set to 1.5 ms, and the port level is sampled by means of interrupt processing. When five low levels have been detected in succession, these low levels are identified as a leader code, and the interval time is changed to 7.81 ms. After that, the pulse width of the low level of the leader code is measured by using rising-edge interrupt request INTP0.

Figure 5-15. Sampling of Remote Controller Signal



<2> Leader code (high)

The pulse width while the leader code is high is measured by using the falling-edge interrupt request INTP0 and the count value of the timer.

<3> Custom/data code

The pulse width of each 1 bit (1 cycle) is measured by using the falling-edge interrupt request INTP0. After the data of the 32nd bit has been loaded, the system tests if the inverted data and custom code coincide. It also checks that there is no data in the 33rd bit.

<4> Repeat code detection

When the high level of the leader code is less than 3 ms, the pulse width from output of the leader code to the rising edge of the INTP0 is measured.

<5> Valid period of repeat code

After the valid data has been input, sampling is performed by the interrupt processing (1.5 ms interval) of the 16-bit timer/event counter to measure the valid time of the repeat code of 250 ms.

<6> Time out during pulse width measurement

If the interrupt request of the 16-bit timer/event counter (7.81 ms) occurs during pulse width measurement, it is judged to be time out, and the data is invalid.

(1) Description of package

<Public declaration symbol>

RMDATA : Stores remote controller receive data
RPT : Repeat valid period identification flag
IPDTFG : Valid data identification flag
RMDTOK : Input signal validity identification flag
RMDTSET : Input signal identification flag

<Registers used>

Bank 0: AX, BC, HL

<RAM used>

Name	Usage	Attribute	Bytes
RPTCT	Repeat code valid time counter	SADDR	1
RMENDCT	No-input time counter after data input		
SELMOD	Mode selection		
LD_CT	Leader signal detection counter		
RMDATA	Valid data storage area		
WORKP	Input signal storage area	SADDRP	4

<Flags used>

Name	Usage
IPDTFG	Presence/absence of valid data
RMDTOK	Validity of input signal
RMDTSET	Presence/absence of input signal
RPT	Judgment whether repeat valid period elapsed

<Nesting>

5 levels 12 bytes

<Hardware used>

- 16-bit timer/event counter
- P00/TI00/INTP0

<Initial setting>

- OSMS = #00000001B ; Oscillation mode select register: does not use divider circuit
- Setting of 16-bit timer/event counter
  CRC0 = #00000100B ; Selects operation mode of CR00, CR01
  TMC0 = #00001100B ; Clears timer on coincidence between TM0 and CR00
  TCL0 = #00100000B ; Count clock f_{XX}
  CR00 = #6290 ; Compare register 00
- SCS = #00000011B ; INTP0 sampling clock $f_{XX}/2^6$
- PPR0 = 0 ; INTP0 high-priority interrupt
- TMMK0 = 0 ; Enables 16-bit timer/event counter interrupt
- Defines custom code to be CSTM and declares PUBLIC
- RAM clear

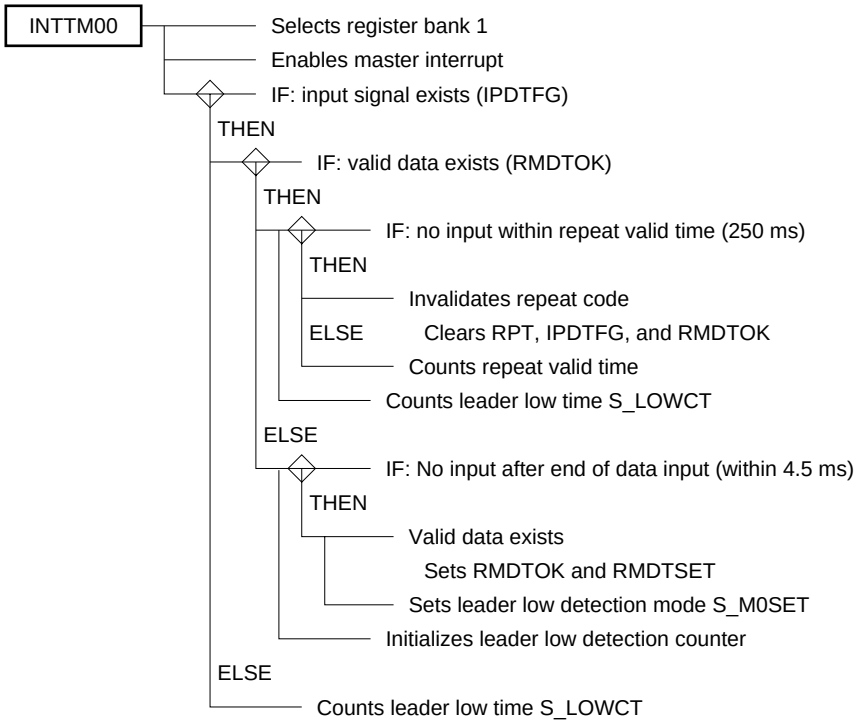
<Starting>

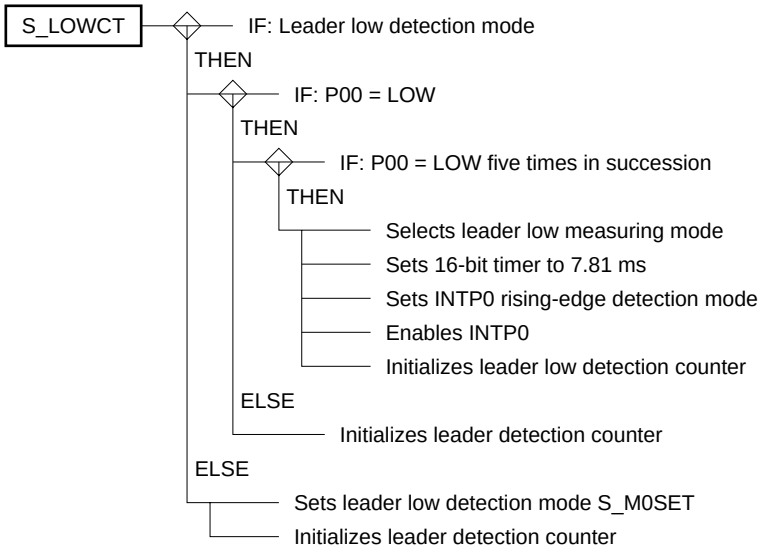
Started by INTP0 and INTTM00 interrupt requests

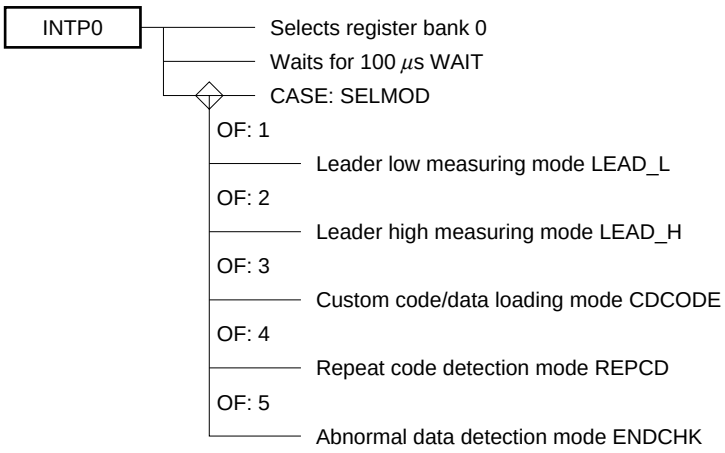
(2) Example of use

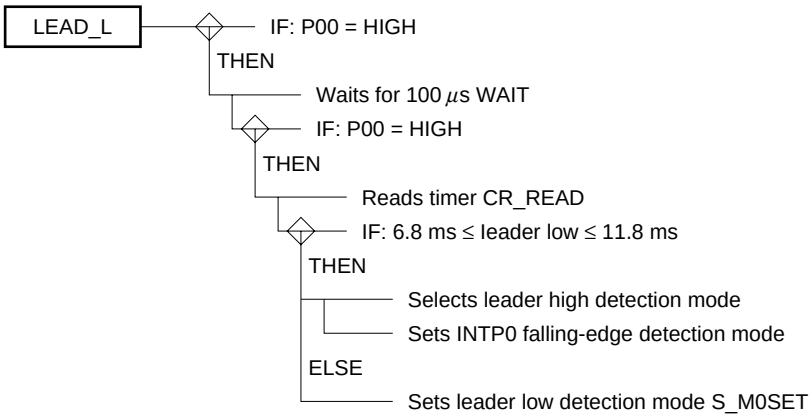
```
        PUBLIC  CSTM
        EXTRN   RMDATA,RPTCT
        EXTBIT  RPT,RMDTSET,IPDTFG

CSTM    EQU     9DH                     ; Remote controller custom code

        OSMS=#00000001B                 ; Does not use divider circuit
        CRC0=#00000100B                 ; Selects operation mode of CR00, CR01
        CR00=#6290
        TCL0=#00100000B                 ; Sets 1.5 ms
        TMC0=#00001100B
        SCS=#00000011B                  ; fxx/2^6 as INTP0 sampling clock

        CLR1    PPR0                    ; INTP0 with high priority
        CLR1    RPT                     ; Clears flag
        CLR1    IPDTFG
        CLR1    RMDTSET

        CLR1    TMMK0                   ; Enables timer interrupt
        EI

DT_TEST:
        if_bit(RMDTSET)
            CLR1    RMDTSET
            if_bit(RPT)
;
;               Repeat processing
;
            else
;
;               Processing when there is input
;
            endif
        else
            if_bit(!RPT)
;
;               Processing when there is no input
;
            endif
        endif
```

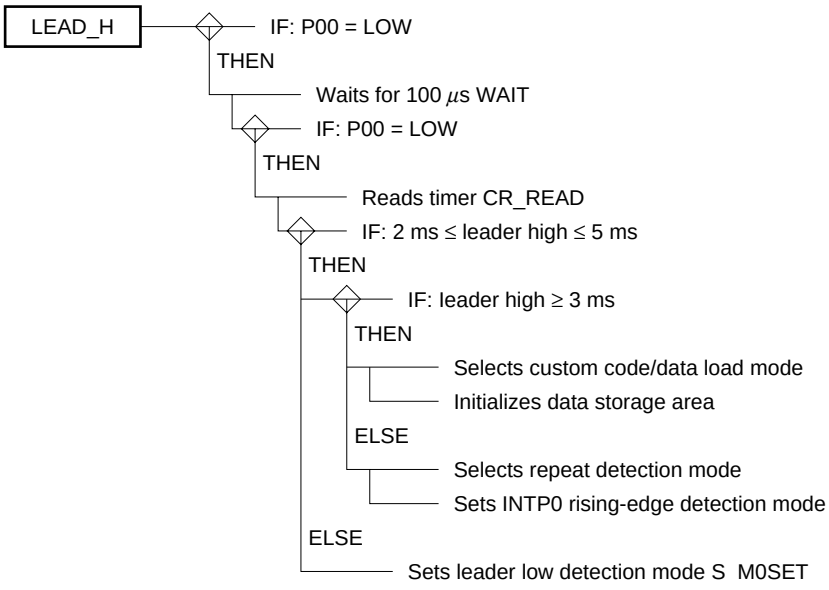
(3) SPD chart

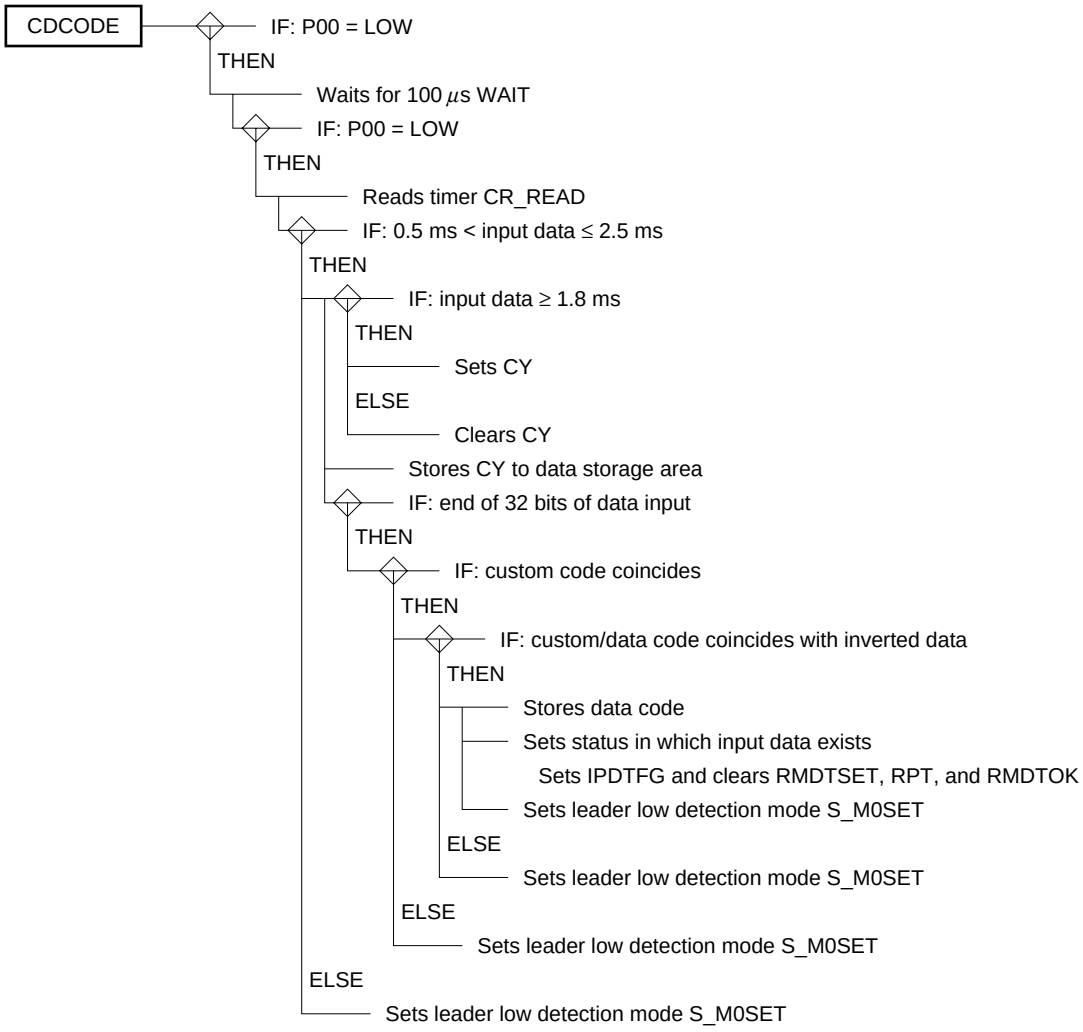

INTTM00
Selects register bank 1
Enables master interrupt
IF: input signal exists (IPDTFG)
THEN
IF: valid data exists (RMDTOK)
THEN
IF: no input within repeat valid time (250 ms)
THEN
Invalidates repeat code
ELSE
Clears RPT, IPDTFG, and RMDTOK
Counts repeat valid time
Counts leader low time S_LOWCT
ELSE
IF: No input after end of data input (within 4.5 ms)
THEN
Valid data exists
Sets RMDTOK and RMDTSET
Sets leader low detection mode S_M0SET
Initializes leader low detection counter
ELSE
Counts leader low time S_LOWCT

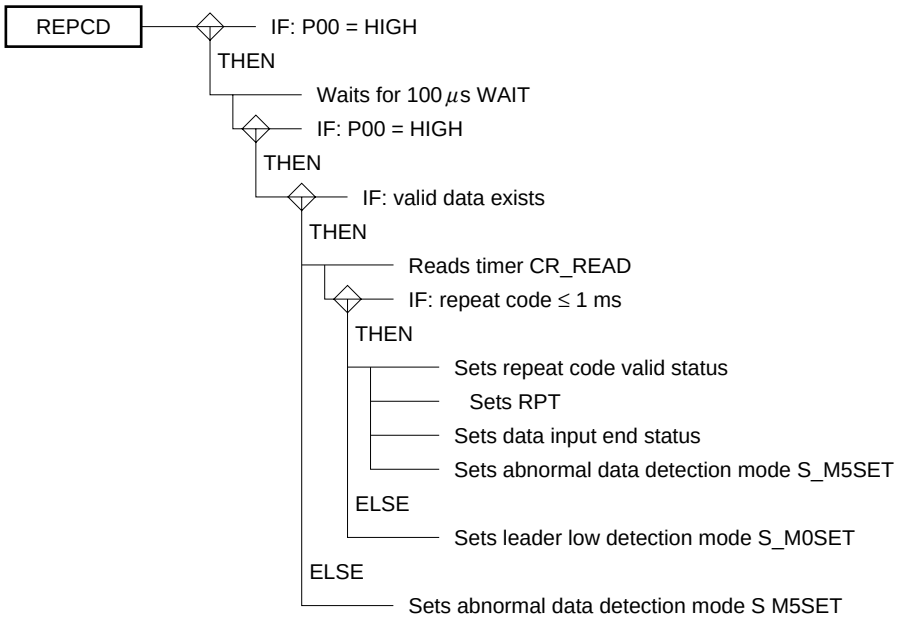

S_LOWCT
IF: Leader low detection mode
THEN
IF: P00 = LOW
THEN
IF: P00 = LOW five times in succession
THEN
Selects leader low measuring mode
Sets 16-bit timer to 7.81 ms
Sets INTP0 rising-edge detection mode
Enables INTP0
Initializes leader low detection counter
ELSE
Initializes leader detection counter
ELSE
Sets leader low detection mode S_M0SET
Initializes leader detection counter

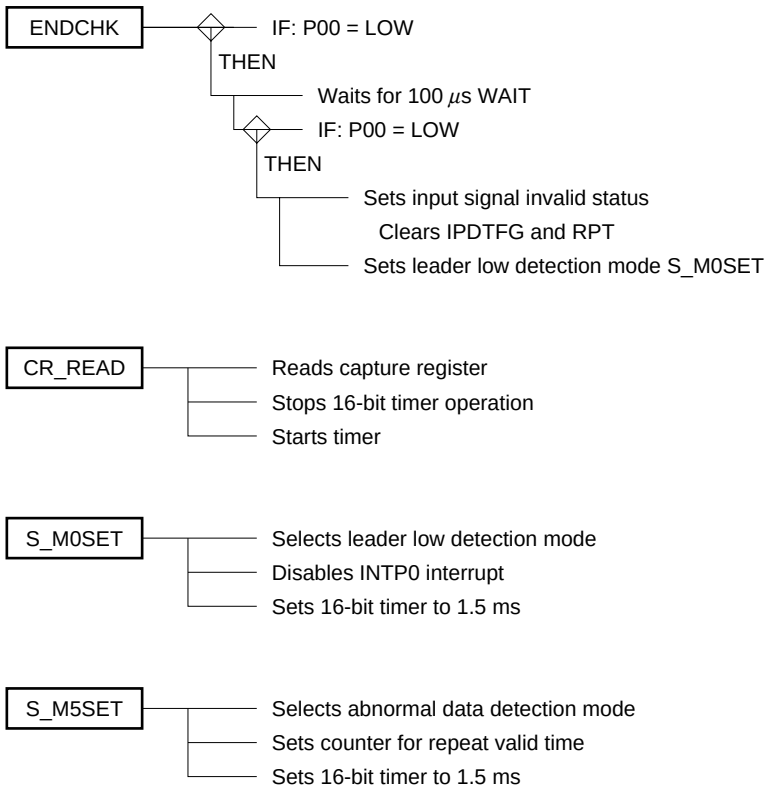

INTP0
Selects register bank 0
Waits for 100 μs WAIT
CASE: SELMOD
OF: 1
Leader low measuring mode LEAD_L
OF: 2
Leader high measuring mode LEAD_H
OF: 3
Custom code/data loading mode CDCODE
OF: 4
Repeat code detection mode REPCD
OF: 5
Abnormal data detection mode ENDCHK


LEAD_L
IF: P00 = HIGH
THEN
Waits for 100 μs WAIT
IF: P00 = HIGH
THEN
Reads timer CR_READ
IF: 6.8 ms ≤ leader low ≤ 11.8 ms
THEN
Selects leader high detection mode
Sets INTP0 falling-edge detection mode
ELSE
Sets leader low detection mode S_M0SET


LEAD_H
IF: P00 = LOW
THEN
Waits for 100 μs WAIT
IF: P00 = LOW
THEN
Reads timer CR_READ
IF: 2 ms ≤ leader high ≤ 5 ms
THEN
IF: leader high ≥ 3 ms
THEN
Selects custom code/data load mode
Initializes data storage area
ELSE
Selects repeat detection mode
Sets INTP0 rising-edge detection mode
ELSE
Sets leader low detection mode S M0SET


CDCODE
IF: P00 = LOW
THEN
Waits for 100 μs WAIT
IF: P00 = LOW
THEN
Reads timer CR_READ
IF: 0.5 ms < input data ≤ 2.5 ms
THEN
IF: input data ≥ 1.8 ms
THEN
Sets CY
ELSE
Clears CY
Stores CY to data storage area
IF: end of 32 bits of data input
THEN
IF: custom code coincides
THEN
IF: custom/data code coincides with inverted data
THEN
Stores data code
Sets status in which input data exists
Sets IPDTFG and clears RMDTSET, RPT, and RMDTOK
Sets leader low detection mode S_M0SET
ELSE
Sets leader low detection mode S_M0SET
ELSE
Sets leader low detection mode S_M0SET
ELSE
Sets leader low detection mode S_M0SET


REPCD
IF: P00 = HIGH
THEN
Waits for 100 μs WAIT
IF: P00 = HIGH
THEN
IF: valid data exists
THEN
Reads timer CR_READ
IF: repeat code ≤ 1 ms
THEN
Sets repeat code valid status
Sets RPT
Sets data input end status
Sets abnormal data detection mode S_M5SET
ELSE
Sets leader low detection mode S_M0SET
ELSE
Sets abnormal data detection mode S M5SET


ENDCHK
IF: P00 = LOW
THEN
Waits for 100 μs WAIT
IF: P00 = LOW
THEN
Sets input signal invalid status
Clears IPDTFG and RPT
Sets leader low detection mode S_M0SET
CR_READ
Reads capture register
Stops 16-bit timer operation
Starts timer
S_M0SET
Selects leader low detection mode
Disables INTP0 interrupt
Sets 16-bit timer to 1.5 ms
S_M5SET
Selects abnormal data detection mode
Sets counter for repeat valid time
Sets 16-bit timer to 1.5 ms

(4) Program list

```
        PUBLIC  RPT,IPDTFG,RMDTOK,RMDTSET
        PUBLIC  RMENDCT,RPTCT,SELMOD,LD_CT,RMDATA
        EXTRN   CSTM
RM_DAT  DSEG    SADDR
RPTCT:  DS      1                                       ; Repeat code valid time counter
RMENDCT:        DS                                      ; No-input time counter after data input
SELMOD: DS      1                                       ; Selects mode
LD_CT:  DS      1                                       ; Leader signal detection counter
RMDATA: DS      1                                       ; Valid data storage area

RM_DATP DSEG    SADDRP
WORKP:  DS      4                                       ; Input signal storage area

        BSEG
IPDTFG  DBIT                                            ; Valid data exists
RMDTOK  DBIT                                            ; Input signal is valid
RMDTSET DBIT                                            ; Input signal exists
RPT     DBIT                                            ; Repeat code valid period

VEP0    CSEG    AT 06H
        DW      INTP0                                   ; Sets vector address of INTP0

VETM0   CSEG    AT 20H
        DW      INTTM00                                 ; Sets vector address of 16-bit timer

;*****************************************************
;      Remote controller signal timer processing
;*****************************************************
TM0_SEG     CSEG
INTTM00:

        SEL RB1
        EI                                              ; Enables interrupt (INTP0)
        if_bit(IPDTFG)                                  ; Input signal exists?
            if_bit(RMDTOK)                              ; Valid data exists?
            RPTCT--
                if(RPTCT==#0)                           ; Repeat invalid time
                    CLR1    RPT                         ; Repeat code invalid status
                    CLR1    IPDTFG
                    CLR1    RMDTOK
                endif
                CALL        !S_LOWCT
            else
                RMENDCT--
                if(RMENDCT==#0)
                    SET1    RMDTOK                      ; Sets that valid data exists
                    SET1    RMDTSET
                    CALL    !S_M0SET                    ; Sets leader (low) detection mode
                endif
                LD_CT=#5
            endif
        else
            CALL            !s_LOWCT
        endif
        RETI
```

```
S_LOWCT:
        if(SELMOD==#0)                          ; Leader (low) detection mode?
            if_bit(!P0.0)
                LD_CT--
                if(LD_CT==#0)
                    SELMOD=#1                   ; Leader (low) measuring mode
                    TMC0=#00000000B
                    CR00=#32767                 ; Timer:  7.81 ms
                    TMC0=#00001100B
                    INTM0=#00000100B
                    CLR1    PIF0
                    CLR1    PMK0                ; Enables INTP0 interrupt
                    LD_CT=#5
                endif
            else
                LD_CT=#5
            endif
        else
            CALL            !S_MOSET            ; Sets leader (low) detection mode
            LD_CT=#5
        endif
        RET
$EJECT
;*********************************************************
;*    Remote controller signal edge detection processing
;*********************************************************
P0_SEG  CSEG
INTP0:

        SEL     RB0
        CALL    !WAIT                           ; Waits for 100 μs
        switch(SELMOD)
        case 1:
            CALL            !LEAD_L             ; Leader low detection processing
            break
        case 2:
            CALL            !LEAD_H             ; Leader high detection processing
            break
        case 3:
            CALL            !CDCODE             ; Custom/data code loading processing
            break
        case 4:
            CALL            !REPCD              ; Repeat code detection processing
            break
        case 5:
            CALL            !ENDCHK             ; Abnormal data detection processing
        ends
        RET1
```

```
;*************************************
;*        Leader low detection
;*************************************
LEAD_L:

        if_bit(P0.0)                            ; Level check P0.0 = 0:  noise
            CALL        !WAIT                   ; Waits for 100 μs
            if_bit(P0.0)
                CALL        !CR_READ            ; Reads timer value
                if(AX>=#3354)                   ; 6.8 ms – (1.5 ms * 4)
                    if(AX<#18035)               ; 11.8 ms – (1.5 ms * 5)
                        SELMOD=#2               ; Leader high detection mode
                        INTM0=#00000000B        ; INTP0 falling edge
                    else
                        CALL    !S_MOSET        ; Sets leader (low) detection mode
                    endif
                else
                    CALL    !S_MOSET            ; Sets leader (low) detection mode
                endif
            endif
        endif
        RET
$EJECT
;*************************************
;*      Leader high detection
;*************************************
LEAD_H:
        if_bit(!P0.0)                           ; Level check P0.0 = 1:  noise
            CALL        !WAIT                   ; Waits for 100 μs
            if_bit(!P0.0)
                CALL        !CR_READ            ; Reads timer value
                if(AX>=#6710-160/2)             ; 1.8 ms – 100 μs * 2 – 160 clocks (edge detection → timer starts)
                    if(AX<#20132-160/2)         ; 5 ms – 100 μs * 2 – 160 clocks (edge detection → timer starts)
                        if(AX>#11743-160/2)     ; Custom/data code (3 ms – 100 μs * 2)?
                            SELMOD=#3           ; Data loading mode
                            WORKP=#0000H        ; Initializes work area
                            (WORKP)+2=#8000H    ; Sets most significant bit to 1 (to check end of data)
                        else
                            SELMOD=#4           ; Repeat detection mode
                            INTM0=#00000100B    ; INTP0 rises
                        endif
                    else
                        CALL    !S_M0SET        ; Sets leader (low) detection mode
                    endif
                else
                    CALL    !S_M0SET            ; Sets leader (low) detection mode
                endif
            endif
        endif
        RET
$EJECT
```

```
;*****************************
;*  Custom/data code loading
;*****************************
CDCODE:
       if_bit(!P0.0)                          ; Level check P0.0 = 1:  noise
           CALL       !WAIT                  ; Waits for 100 μs
           if_bit(!P0.0)
               CALL       !CR_READ           ; Reads timer value
               if(AX>=#1257-190/2)            ; 0.5 ms – 100 μs * 2 – 190 clocks (edge detection → timer starts)
                   if(AX<#9646-190/2)         ; 2.5 ms – 100 μs * 2 –190 clocks (edge detection → timer starts)
                       if(AX>=#6710-190/2)    ; 1.8 ms – 100 μs * 2 – 190 clocks (edge detection → timer starts)
                           SET1    CY
                       else
                           CLR1    CY
                       endif
                       HL=#WORKP+3            ; Sets work area address
                       C=#4                   ; Sets number of digits of work area
                   WKSHFT:
                       A=[HL]                 ; Stores 1-bit data
                       RORC       A,1         ; Shifts 1 bit
                       [HL]=A
                       HL--
                       DBNZ       C,$WKSHFT   ; End of shifting all bits
                       if_bit(CY)             ; End of 32-bit input?
                           if(WORKP+0==#CSTM) (A)
                                              ; Custom code check
                               A^WORKP+1
                               if(A==#0FFH)   ; Custom code inverted data check
                                   A=WORKP+2
                                   A^=WORKP+3 ; Data code inverted data check
                                   if(A==#0FFH)
                                              ; Stores input data
                                       RMDATA=WORKP+2 (A)
                                              ; Sets status in which input data exists
                                       SET1    IPDTFG
                                       CLR1    RMDTSET
                                       CLR1    RPT
                                       CLR1    RMDTOK
                                       CALL    !S_M5SET
                                   else
                                              ; Sets leader (low) detection mode
                                       CALL    !S_M0SET
                                   endif
                               else
                                              ; Sets leader (low) detection mode
                                   CALL    !S_M0SET
                               endif
                           else
                               CALL       !S_M0SET
```

```
                        endif
                    endif
                else
                    CALL    !S_M0SET        ; Sets leader (low) detection mode
                endif
            else
                CALL    !S_M0SET            ; Sets leader (low) detection mode
            endif
        endif
    endif
    RET
$EJECT
;*********************************
;*      Repeat code detection
;*********************************
REPCD:
    if_bit(P0.0)                            ; Level check P0.0 = 0:  noise
        CALL       !WAIT                    ; Waits for 100 μs
        if_bit(P0.0)
            if_bit(RMDTOK)                  ; Valid data exists?
                CALL    !CR_READ            ; Reads timer value
                if(AX<=#3354-190/2)         ; 1 ms – 100 μs * 2 – 190 clocks (edge detection → timer starts)
                    SET1    RPT
                    CLR1    RMDTOK          ; Input signal check after end of data
                    CLR1    RMDTSET
                    CALL    !S_M5SET
                else
                    CALL    !S_M0SET        ; Sets leader (low) detection mode
                endif
            else
                CALL    !S_M0SET            ; Sets leader (low) detection mode
            endif
        endif
    endif
    RET
$EJECT
```

```
;*************************************
;*      Abnormal data detection
;*************************************
ENDCHK:
        if_bit(!P0.0)                       ; Level check P0.0 = 1:  noise
            CALL        !WAIT               ; Waits for 100 μs
            if_bit(!P0.0)
                CLR1        IPDTFG          ; Abnormal data input
                CLR1        RPT             ; Input signal invalid
                CALL        !S_M0SET        ; Sets leader (low) detection mode
            endif
        endif
        RET

;*************************************
;*          Waits for 100 μs
;*************************************
WAIT:
        B=#(838-14-12-8)/12                 ; CALL(14), RET(12), MOV(8)
WAITCT:                                     ; Sets 100 μs
        DBNZ    B,$WAITCT                   ; 1 instruction 12 clocks
        RET

;*************************************
;*   Sets leader (low) detection mode
;*************************************
S_M0SET:
        TMC0=#00000000B
        CR00=#6290
        TCL0=#00100000B                     ; Sets timer to 1.5 ms
        TMC0=#00001100B
        SELMOD=#0                           ; Leader (low) detection mode
        SET1    PMK0
        RET

;*************************************
;*   Sets abnormal data detection mode
;*************************************
S_M5SET:
        RPTCT=#173                          ; 250 ms measuring counter
        SELMOD=#5                           ; Data input end mode
        RMENDCT=#3                          ; No-input checking counter
        TMC0=#00000000B                     ; Stops operation
        CR00=#6290                          ; Sets 1.5 ms
        TMC0=#00001100B
        RET
;*************************************
;*      Reads timer count value
;*************************************
CR_READ:
        AX=CR01
        TMC0=#00000000B                     ; Stops operation
        TMC0=#00001100B                     ; Starts timer
        RET
```

5.3.2 Remote controller signal reception by PWM output and free running mode

Table 5-2 shows the valid pulse width when a remote controller signal is received by this program. <1> through <6> below describes how each signal is processed.

Table 5-2. Valid Time of Input Signal

Signal Name		Output Time	Valid Time
Leader code (low)		9 ms	3 ms-10 ms
Leader code (high)	Normal	4.5 ms	3 ms-5 ms
	Repeat	2.25 ms	1.8 ms-3 ms
Custom/data code	0	1.125 ms	0.5 ms-1.8 ms
	1	2.25 ms	1.8 ms-2.5 ms

<1> Leader code (low)

The value of the capture/compare register 01 (CR01) is stored to memory by an interrupt request that occurs when the falling edge of INTP0 is detected.

The pulse width is measured from the difference between the values of CR01 and the capture/compare register 00 (CR00) when the rising edge is generated.

<2> Leader code (high)

The pulse width between the high levels of the leader code is measured by the falling-edge interrupt request INTP0 and the count value of the timer.

<3> Custom/data code

The pulse width of each 1 bit (1 cycle) is measured by the falling-edge interrupt request INTP0. After the data of the 32nd bit has been loaded, the system tests if the inverted data and custom code coincide. It also checks that there is no data of the 33rd bit.

<4> Repeat code detection

When the high level of the leader code is less than 3 ms, the pulse width from output of the leader code to the rising edge of the INTP0 is measured.

<5> Valid period of repeat code

After the valid data has been input, the overflow flag (OVF0) of the 16-bit timer/event counter is tested by the main program, and the repeat code valid time of 250 ms is measured.

<6> Time out during pulse width measurement

The OVF0 of the 16-bit timer/event counter is tested during pulse width measurement. If it is detected two times, time out is assumed and the data is assumed to be invalid.

Because the 16-bit timer/event counter operates in the PWM mode in this example, the remote controller signal is received and, at the same time, PWM output can be performed by linking the program of **5.2 PWM Output.**

(1) Description of package

<Public declaration symbol>

TIM_PRO : name of subroutine processing timer overflow
RMDATA : stores remote controller receive data
RPT : repeat valid period identification flag
IPDTFG : valid data identification flag
RMDTOK : valid input signal identification flag
RMDTSET : input signal identification flag
OVSENS : INTP0 processing timer overflow detection flag

<Registers used>

Bank 0: AX, BC, HL

<RAM used>

Name	Usage	Attribute	Bytes
RPTCT	Repeat code invalid time counter	SADDR	1
RMENDCT	No-input time counter after data input		
SELMOD	Mode selection		
LD_CT	Leader signal detection counter		
RMDATA	Valid data storage area		
TO_CNT	Timer overflow detection counter		
CR01_NP	Newest timer count value storage area	SADDRP	2
CR01_OP	Previous timer count value storage area		
WORKP	Input signal storage area		4

<Flag used>

Name	Usage
IPDTFG	Presence/absence of valid data
RMDTOK	Presence/absence of valid input signal
RMDTSET	Presence/absence of input signal
RPT	Judgment whether repeat valid period elapsed
TO_FLG	Occurrence of timer overflow
OVSENS	Detection of timer overflow by INTP0 processing

<Nesting>

5 levels 11 bytes

<Hardware used>

- 16-bit timer/event counter
- P00/TI00/INTP0
- P30/TO0

<Initial setting>

- OSMS = #00000001B ; Oscillation mode select register: does not use divider circuit
- Setting of 16-bit timer/event counter
  CRC0 = #00000100B ; Selects operation mode of CR00, CR01
  TMC0 = #00000010B ; PWM output mode
  TCL0 = #00100000B ; PWM basic cycle: 61.0 μs
  TOC0 = #00000011B ; Low-active output
- PM30 = 0 ; P30 output mode
- SCS = #00000011B ; INTP0 sampling clock $f_{XX}/2^6$
- PPR0 = 0 ; INTP0 high-priority interrupt
- PMK0 = 0 ; Enables INTP0 interrupt
- Defines custom code to CSTM and declares PUBLIC
- RAM clear

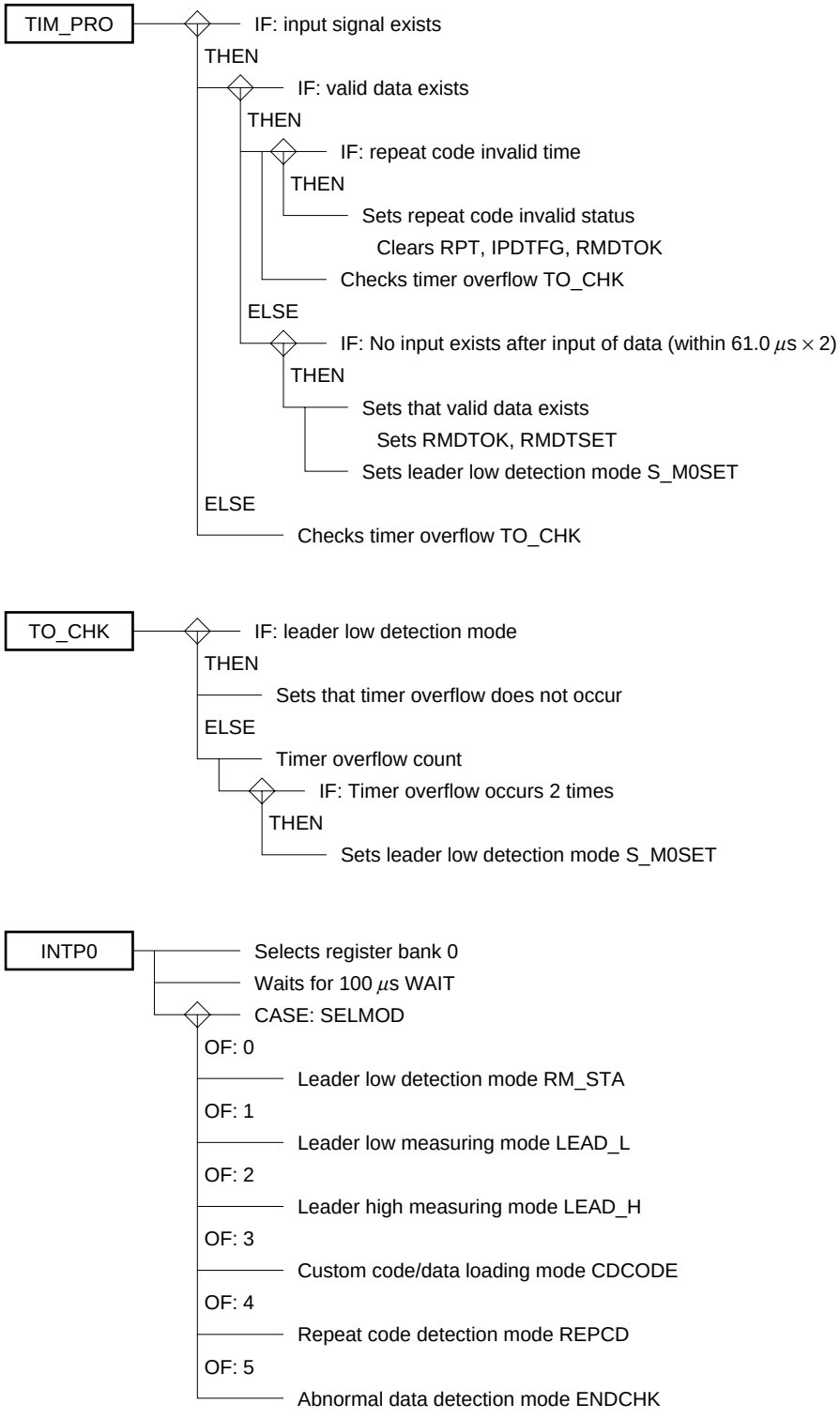
<Starting>

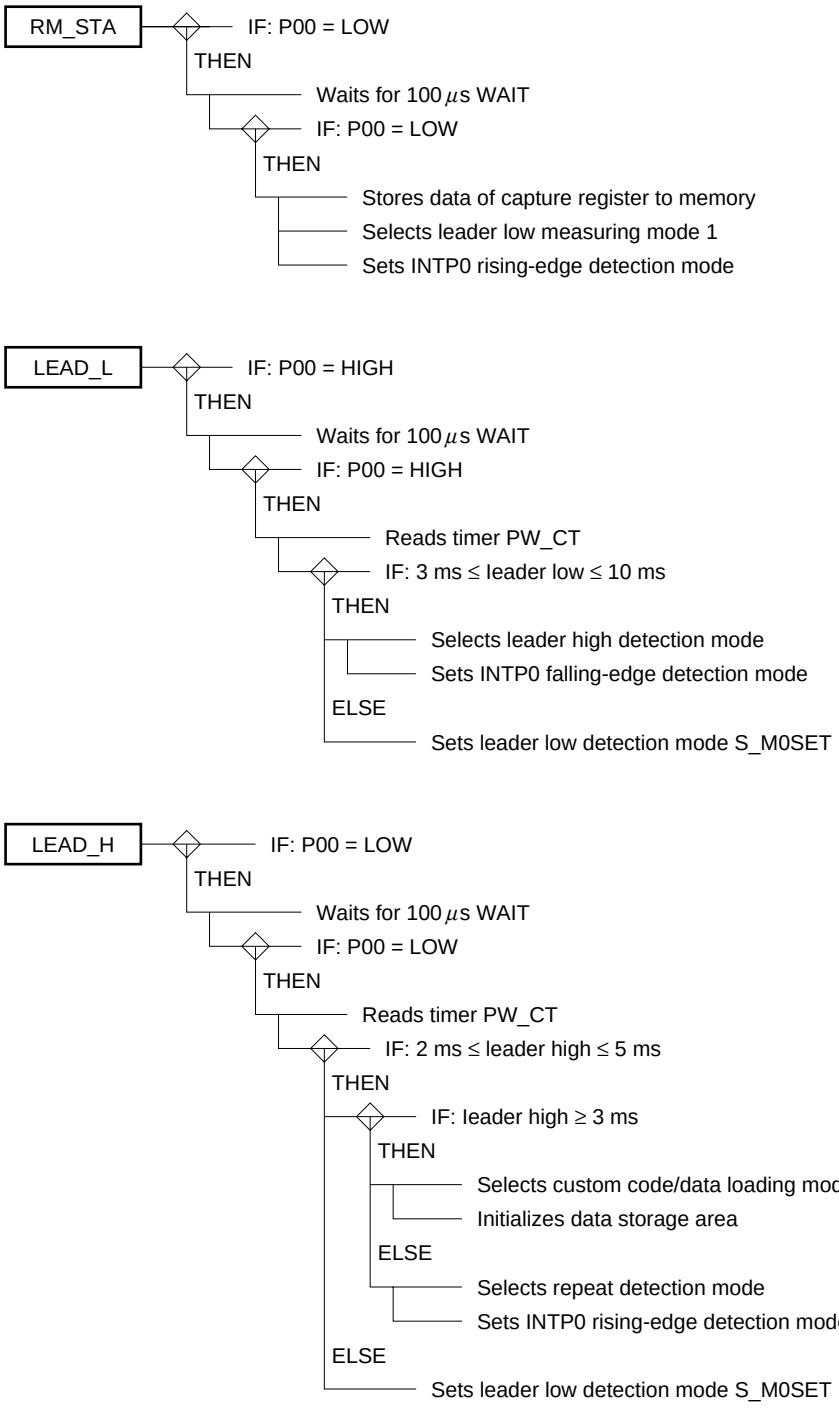
- Test the OVF0 of the 16-bit timer/event counter. When OVF0 is set, call subroutine TIM_PRO.
- Start by an interrupt request when the valid edge of the remote controller signal is detected.

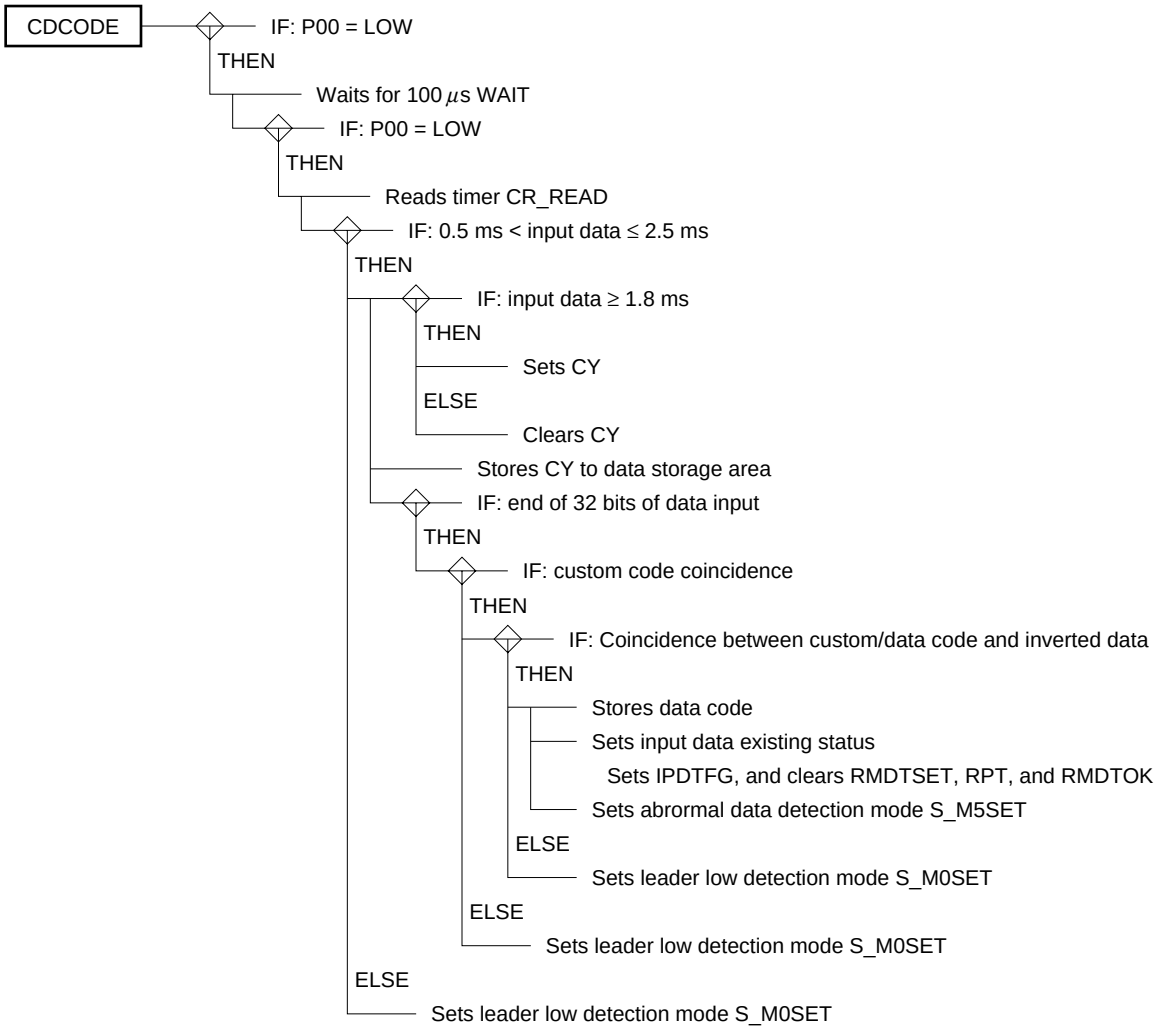
(2) Example of use

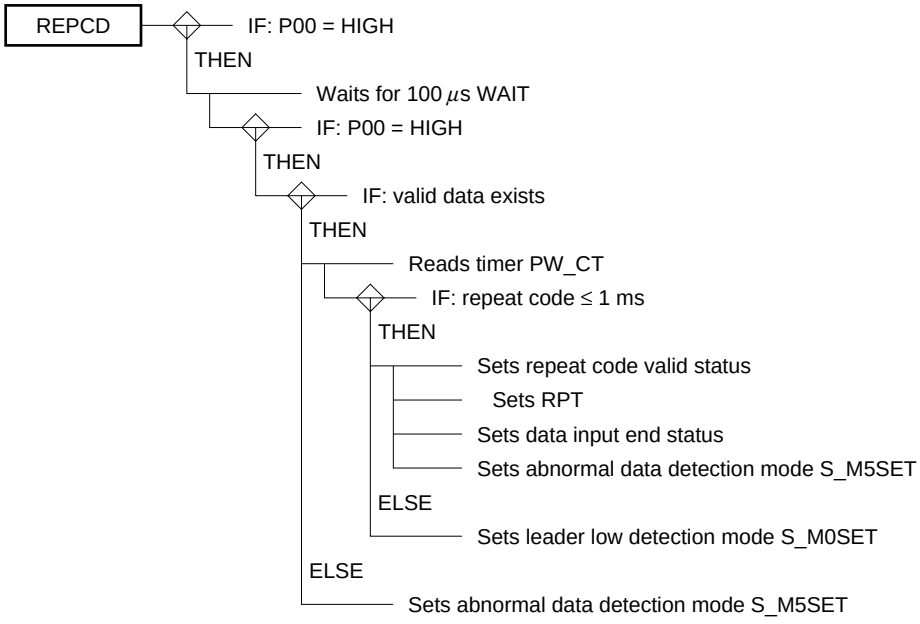
```
        PUBLIC  CSTM
        EXTRN   RMDATA,RPTCT,PWM,PWMOUT,TIM_PRO
        EXTBIT  RPT,RMDTSET,IPDTFG,TO_FLG,OVSENS
CSTM    EQU     9DH                         ; Custom code

        OSMS=#00000001B                     ; Does not use divider circuit
        CRC0=#00000100B                     ; Selects operation mode of CR00, CR01
        TOC0=#00000011B                     ; PWM output, low active setting
        TCL0=#00100000B                     ; Selects count clock fxx
        TMC0=#00000010B                     ; PWM mode, overflow occurs
        INTM0=#00000000B                    ; INTP0 falling edge
        SCS=#00000011B                      ; INTP0 sampling clock fxx/2^6

        CLR1    PPR0                        ; INTP0 with high priority
        CLR1    RPT                         ; Clears flag
        CLR1    IPDTFG
        CLR1    RMDTSET

        CLR1    PMK0                        ; Enables INTP0 interrupt
        EI
DT_TEST:
        if_bit(OVSENS)                      ; Detects timer overflow by INTP0 processing
            CLR1    OVSENS
            CALL    !TIM_PRO
        elseif_bit(OVF0)                    ; Timer overflow occurs
            CLR1    OVF0
            SET1    TO_FLG
            CALL    !TIM_PRO
        endif
        if_bit(RMDTSET)
            CLR1    RMDTSET
            if_bit(RPT)
;
;               Repeat processing
;
            else
;
;               Processing when input exists
;
            endif
        else
            if_bit(!RPT)
;
;               Processing when input does not exist
;
            endif
        endif
        MOV     PWMOUT,A
        CALL    !PWM
```

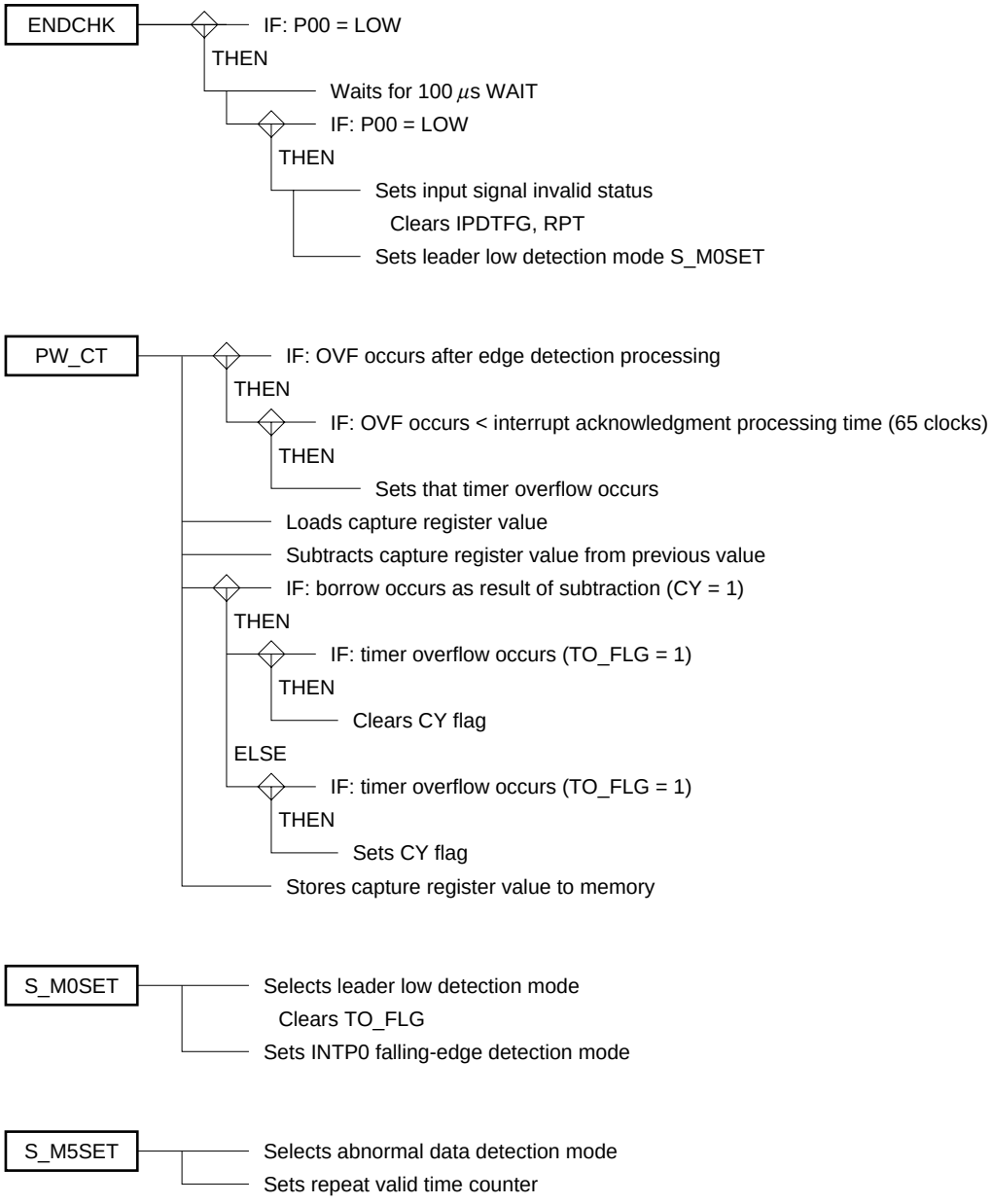
(3) SPD chart


TIM_PRO
IF: input signal exists
THEN
IF: valid data exists
THEN
IF: repeat code invalid time
THEN
Sets repeat code invalid status
Clears RPT, IPDTFG, RMDTOK
Checks timer overflow TO_CHK
ELSE
IF: No input exists after input of data (within 61.0 μs × 2)
THEN
Sets that valid data exists
Sets RMDTOK, RMDTSET
Sets leader low detection mode S_M0SET
ELSE
Checks timer overflow TO_CHK
TO_CHK
IF: leader low detection mode
THEN
Sets that timer overflow does not occur
ELSE
Timer overflow count
IF: Timer overflow occurs 2 times
THEN
Sets leader low detection mode S_M0SET
INTP0
Selects register bank 0
Waits for 100 μs WAIT
CASE: SELMOD
OF: 0
Leader low detection mode RM_STA
OF: 1
Leader low measuring mode LEAD_L
OF: 2
Leader high measuring mode LEAD_H
OF: 3
Custom code/data loading mode CDCODE
OF: 4
Repeat code detection mode REPCD
OF: 5
Abnormal data detection mode ENDCHK


RM_STA
IF: P00 = LOW
THEN
Waits for 100 μs WAIT
IF: P00 = LOW
THEN
Stores data of capture register to memory
Selects leader low measuring mode 1
Sets INTP0 rising-edge detection mode
LEAD_L
IF: P00 = HIGH
THEN
Waits for 100 μs WAIT
IF: P00 = HIGH
THEN
Reads timer PW_CT
IF: 3 ms ≤ leader low ≤ 10 ms
THEN
Selects leader high detection mode
Sets INTP0 falling-edge detection mode
ELSE
Sets leader low detection mode S_M0SET
LEAD_H
IF: P00 = LOW
THEN
Waits for 100 μs WAIT
IF: P00 = LOW
THEN
Reads timer PW_CT
IF: 2 ms ≤ leader high ≤ 5 ms
THEN
IF: leader high ≥ 3 ms
THEN
Selects custom code/data loading mode
Initializes data storage area
ELSE
Selects repeat detection mode
Sets INTP0 rising-edge detection mode
ELSE
Sets leader low detection mode S_M0SET


CDCODE
IF: P00 = LOW
THEN
Waits for 100 μs WAIT
IF: P00 = LOW
THEN
Reads timer CR_READ
IF: 0.5 ms < input data ≤ 2.5 ms
THEN
IF: input data ≥ 1.8 ms
THEN
Sets CY
ELSE
Clears CY
Stores CY to data storage area
IF: end of 32 bits of data input
THEN
IF: custom code coincidence
THEN
IF: Coincidence between custom/data code and inverted data
THEN
Stores data code
Sets input data existing status
Sets IPDTFG, and clears RMDTSET, RPT, and RMDTOK
Sets abrormal data detection mode S_M5SET
ELSE
Sets leader low detection mode S_M0SET
ELSE
Sets leader low detection mode S_M0SET
ELSE
Sets leader low detection mode S_M0SET


REPCD
IF: P00 = HIGH
THEN
Waits for 100 μs WAIT
IF: P00 = HIGH
THEN
IF: valid data exists
THEN
Reads timer PW_CT
IF: repeat code ≤ 1 ms
THEN
Sets repeat code valid status
Sets RPT
Sets data input end status
Sets abnormal data detection mode S_M5SET
ELSE
Sets leader low detection mode S_M0SET
ELSE
Sets abnormal data detection mode S_M5SET


ENDCHK
IF: P00 = LOW
THEN
Waits for 100 μs WAIT
IF: P00 = LOW
THEN
Sets input signal invalid status
Clears IPDTFG, RPT
Sets leader low detection mode S_M0SET
PW_CT
IF: OVF occurs after edge detection processing
THEN
IF: OVF occurs < interrupt acknowledgment processing time (65 clocks)
THEN
Sets that timer overflow occurs
Loads capture register value
Subtracts capture register value from previous value
IF: borrow occurs as result of subtraction (CY = 1)
THEN
IF: timer overflow occurs (TO_FLG = 1)
THEN
Clears CY flag
ELSE
IF: timer overflow occurs (TO_FLG = 1)
THEN
Sets CY flag
Stores capture register value to memory
S_M0SET
Selects leader low detection mode
Clears TO_FLG
Sets INTP0 falling-edge detection mode
S_M5SET
Selects abnormal data detection mode
Sets repeat valid time counter

(4) Program list

```
        PUBLIC  TIM_PRO,RPT,IPDTFG,RMDTOK,RMDTSET
        PUBLIC  RMENDCT,RPTCT,SELMOD,LD_CT,RMDATA
        PUBLIC  TO_FLG,OVSENS
        EXTRN   CSTM

RM_DAT  DSEG    SADDR
RPTCT:  DS      1                                   ; Repeat code valid time counter
RMENDCT:DS      1                                   ; No-input time counter after data input
SELMOD: DS      1                                   ; Mode selection
LD_CT:  DS      1                                   ; Leader signal detection counter
RMDATA: DS      1                                   ; Valid data storage area
TO_CNT: DS      1                                   ; Timer overflow counter

RM_DATP DSEG    SADDRP
CR01_NP:DS      2                                   ; Newest timer counter value storage area
CR01_OP:DS      2                                   ; Previous timer counter value storage area
WORKP:  DS      4                                   ; Input signal storage area

        BSEG
IPDTFG  DBIT                                        ; Valid data exists
RMDTOK  DBIT                                        ; Input signal valid
RMDTSET DBIT                                        ; Input signal exists
RPT     DBIT                                        ; Repeat code valid period
TO_FLG  DBIT                                        ; Timer overflow occurs
OVSENS  DBIT                                        ; Detects timer overflow by INTP0 processing

VEP0    CSEG    AT 06H
        DW      INTP0                               ; Sets vector address of INTP0

$EJECT
;*****************************************************
;      Remote controller signal timer processing
;*****************************************************
TM0_SEG CSEG
TIM_PRO:
        if_bit(IPDTFG)                              ; Input signal exists?
            if_bit(RMDTOK)                          ; Valid data exists?
            RPTCT--
                if(RPTCT==#0)                       ; Repeat invalid time
                    CLR1    RPT                     ; Repeat code valid status
                    CLR1    IPDTFG
                    CLR1    RMDTOK
                endif
            else
                RMENDCT--
                if(RMENDCT==#0)
                    SET1    RMDTOK                  ; Valid data exists
                    SET1    RMDTSET
                    CALL    !S_M0SET                ; Sets leader (low) detection mode
                endif
            endif
        else
            CALL    !TO_CHK                         ; Checks timer overflow
        endif
        RET
```

```
 TO_CHK:
        if(SELMOD==#0)
            CLR1        TO_FLG
        else
            TO_CNT++
            if(TO_CNT==#2)
                CALL        !S_M0SET                ; Sets start edge detection mode
            endif
        endif
        RET
    $EJECT
;*************************************************************
;*    Remote controller signal edge detection processing
;*************************************************************
P0_SEG  CSEG
INTP0:
        SEL RB0

        CALL    !WAIT                               ; Waits for 100 μs

        switch(SELMOD)

        case 0:
            CALL    !RM_STA                         ; Start edge detection processing
            break
        case 1:
            CALL    !LEAD_L                         ; Leader low detection processing
            break
        case 2:
            CALL    !LEAD_H                         ; Leader high detection processing
            break
        case 3:
            CALL    !CDCODE                         ; Custom/data code loading processing
            break
        case 4:
            CALL    !REPCD                          ; Repeat code detection processing
            break
        case 5:
            CALL    !ENDCHK                         ; Abnormal data detection processing
        ends
        RET1

;*************************************************************
;*                 Start edge detection
;*************************************************************
RM_STA:
        CLR1    TO_FLG                              ; Starts timer count
        if_bit(!P0.0)                               ; Level check P0.0 = 1:  noise
            CALL    !WAIT                           ; Waits for 100 μs
            if_bit(!P0.0)
                CR01_OP=CR01 (AX)                   ; Stores capture register
                SELMOD=#1                           ; Leader low detection mode
                INTM0=#00000100B                    ; INTP0 rising edge
                TO_CNT=#0
            endif
        endif
        RET
```

```
;*************************************
;*        Leader low detection
;*************************************
LEAD_L:
        if_bit(P0.0)                            ; Level check P0.0 = 1:  noise
            CALL        !WAIT                   ; Waits for 100 μs
            if_bit(P0.0)
                CALL        !PW_CT              ; Reads timer value
                if_bit(!CY)
                    TO_CNT=#0
                    if(AX>=#12582)              ; 3 ms
                        if(AX<#41942)           ; 10 ms
                            SELMOD=#2           ; Leader high detection mode
                            INTM0=#00000000B    ; INTP0 falling edge
                        else
                            CALL    !S_M0SET    ; Sets start edge detection mode
                        endif
                    else
                        CALL    !S_M0SET        ; Sets start edge detection mode
                    endif
                else
                    CALL    !S_M0SET            ; Sets start edge detection mode
                endif
            endif
        endif
        RET
    $EJECT
;*************************************
;*      Leader high detection
;*************************************
LEAD_H:
        if_bit(!P0.0)                           ; Level check P0.0 = 0:  noise
            CALL        !WAIT                   ; Waits for 100 μs
            if_bit(!P0.0)
                CALL        !PW_CT              ; Reads timer value
                if_bit(!CY)
                    TO_CNT=#0
                    if(AX>=#7549)               ; 1.8 ms
                        if(AX<#20971)           ; 5 ms
                            if(AX>#12582)       ; Custom/data code (3 ms)?
                                SELMOD=#3       ; Data loading mode
                                WORKP=#0000H    ; Initializes work area
                                (WORKP)+2=#8000H ; Sets most significant bit to 1 (to confirm end of data)
                            else
                                SELMOD=#4       ; Repeat detection mode
                                INTM0=#00000100B ; INTP0 rises
                            endif
                        else
                            CALL    !S_M0SET    ; Sets start edge detection mode
                        endif
                    else
                        CALL    !S_M0SET        ; Sets start edge detection mode
                    endif
                else
                    CALL    !S_M0SET            ; Sets start edge detection mode
                endif
            endif
        endif
        RET
    $EJECT
```

```
;*****************************
;*  Custom/data code loading
;*****************************
CDCODE:
       if_bit(!P0.0)                              ; Level check P0.0 = 1:  noise
           CALL    !WAIT                          ; Waits for 100 μs
           if_bit(!P0.0)
               CALL    !PW_CT                     ; Reads timer value
               if_bit(!CY)
                   TO_CNT=#0
                   if(AX>=#2096)                  ; 0.5 ms
                       if(AX<#10485)              ; 2.5 ms
                           if(AX>=#7549)          ; 1.8 ms
                               SET1    CY
                           else
                               CLR1    CY
                           endif

                           HL=#WORKP+3            ; Sets work area address
                           C=#4                   ; Sets number of work area digits
                       WKSHFT:
                           A=[HL]                 ; Stores 1-bit data
                           RORC    A,1            ; Shifts 1 bit
                           [HL]=A
                           HL--
                           DBNZ    C,$WKSHFT      ; End of shifting all digits

                           if_bit(CY)             ; End of input of 32 bits?
                                                  ; Checks custom code
                               if(WORKP+0==#CSTM) (A)
                                   A^=WORKP+1
                                   if(A==#0FFH)   ; Checks custom code inverted data
                                       A=WORKP+2
                                                  ; Checks data code inverted data
                                       A^=WORKP+3
                                       if(A==#0FFH)
                                                  ; Stores input data
                                           RMDATA=WORKP+2 (A)
                                                  ; Sets input data existing status
                                       SET1    IPDTFG
                                       CLR1    RMDTSET
                                       CLR1    RPT
                                       CLR1    RMDTOK
                                       CALL    !S_M5SET
                                   else
                                                  ; Sets start edge detection mode
                                       CALL    !S_M0SET
                                   endif
                               else
                                                  ; Sets start edge detection mode
                                   CALL    !S_M0SET
                               endif
                           else
                               CALL    !S_M0SET
                           endif
                       endif
                   else
                       CALL    !S_M0SET           ; Sets start edge detection mode
                   endif
               else
```

```
                    CALL        !S_M0SET        ; Sets start edge detection mode
                endif
            else
                CALL    !S_M0SET                ; Sets start edge detection mode
            endif
        endif
    endif
    RET
$EJECT

;*********************************
;*      Repeat code detection
;*********************************
REPCD:
    if_bit(P0.0)                                ; Level check P0.0 = 1:  noise
        CALL        !WAIT                       ; Waits for 100 μs
        if_bit(P0.0)
            if_bit(RMDTOK)                      ; Valid data?
                CALL    !PW_CT                  ; Reads timer value
                if_bit(!CY)
                    TO_CNT=#0
                    if(AX<=#4193)               ; 1 ms
                        SET1    RPT
                        CLR1    RMDTOK          ; Checks input signal after end of data
                        CLR1    RMDTSET
                        CALL    !S_M5SET
                    else
                        CALL    !S_M0SET        ; Sets start edge detection mode
                    endif
                else
                    CALL    !S_M0SET            ; Sets start edge detection mode
                endif
            else
                CALL    !S_M0SET                ; Sets start edge detection mode
            endif
        endif
    endif
    RET
$EJECT
```

```
;*****************************************
;*         Abnormal data detection
;*****************************************
ENDCHK:
        if_bit(!P0.0)                   ; Level check P0.0 = 1:  noise
            CALL    !WAIT               ; Waits for 100 μs
            if_bit(!P0.0)
                CLR1    IPDTFG          ; Abnormal data input
                CLR1    RPT             ; Input signal invalid
                CALL    !S_M0SET        ; Sets start edge detection mode
            endif
        endif
        RET

;*****************************************
;*   Calculation of capture register value
;*****************************************
PW_CT:
        if_bit(OVF0)                    ; OVF0 after edge detection?
            if(CR01<#10000-33) (AX)     ; Interrupt acknowledgment processing time = 65 clocks (MAX)
                CLR1    OVF0
                SET1    OVSENS
                SET1    TO_FLG
            endif
        endif

        CR01_NP=CR01 (AX)               ; Loads capture register value

        A=CR01_NP+0                     ; AX = CR01_NP – CR01_OP
        A-=CR01_OP
        X=A
        A=CR01_NP+1
        SUBC    A,CR01_OP+1

        BC=AX                           ; Saves operation result
        if_bit(CY)                      ; CR01_NP > CR01_OP
            if_bit(TO_FLG)              ; Timer overflow occurs (flag test)
                CLR1    CY              ; Normal data
            endif
        else
            if_bit(TO_FLG)              ; Timer overflow
                SET1    CY              ; Error occurs
            endif
        endif

        CR01_OP=CR01_NP (AX)
        AX=BC                           ; Restores operation result
        CLR1    TO_FLG
        RET
```

```
;*********************************************
;*             Waits for 100 μs
;*********************************************
WAIT:
        B=#(838-14-12-8)/12                     ; CALL (14), RET (12), MOV (8)
WAITCT:                                         ; Sets 100 μs
        DBNZ    B,$WAITCT                       ; 1 instruction 12 clocks
        RET

;*********************************************
;*      Sets start edge detection mode
;*********************************************
S_M0SET:
        TO_CNT=#0
        SELMOD=#0                               ; Start edge detection mode
        INTM0=#00000000B                        ; INTP0 falling edge
        RET

;*********************************************
;*   Setting of abnormal data detection mode
;*********************************************
S_M5SET:
        RPTCT=#16                               ; 250 ms measuring counter
        SELMOD=#5                               ; Data input end mode
        RMENDCT=#2                              ; No-input checking counter
        RET
```

5.4 One-Shot Pulse Output

The 16-bit timer/event counter has a function which outputs a one-shot pulse in synchronization with a software trigger and external trigger (INTP0/TI00/P00 pin input).

When using the one-shot pulse output function, the 16-bit timer mode control register (TMC0), capture/compare control register 0 (CRC0), and 16-bit timer output control register (TOC0) must be set.

In this section, an example for setting the one-shot pulse by using the software trigger is introduced.

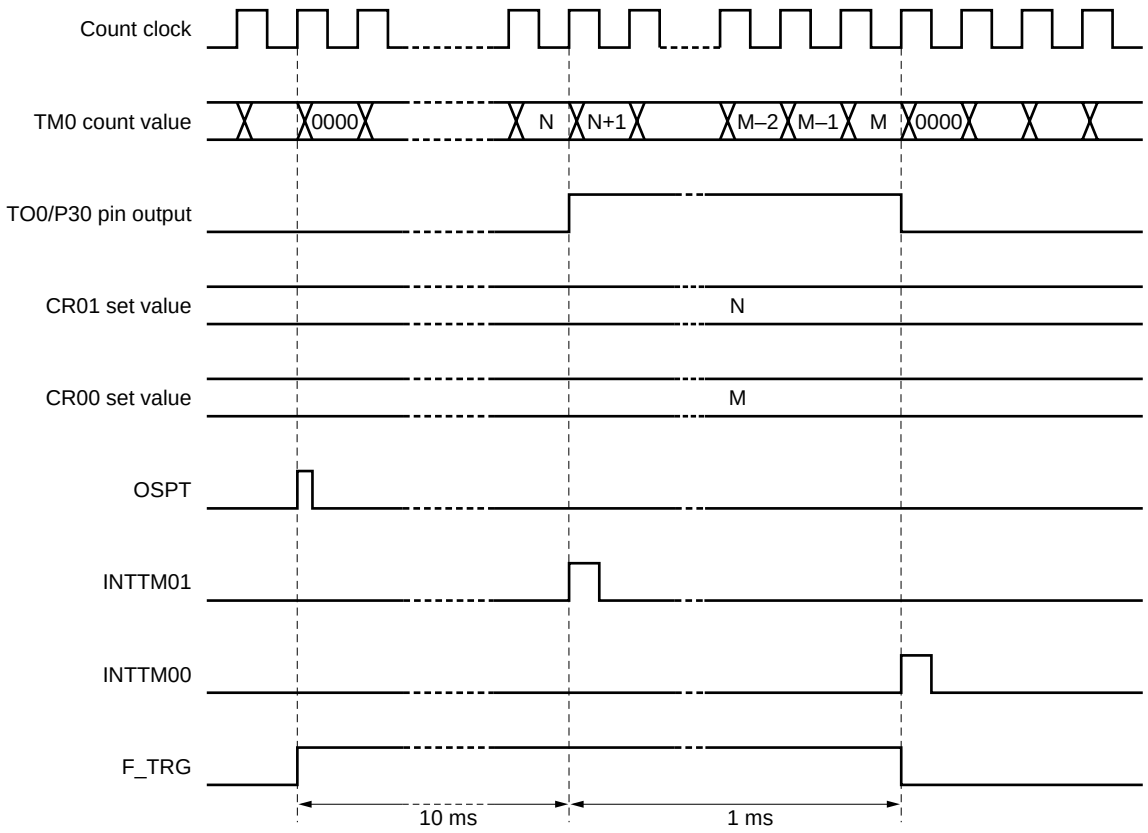
The OSPT flag (bit 6 of the TOC0 register) is set at arbitrary timing (such as key input).

After the software trigger has occurred, TM0 is cleared and started. When the value of TM0 coincides with the value set in advance to CR01, the TO0/P30 pin output is inverted (and becomes active). When the value of TM0 later coincides with the value set in advance to CR00, the TO0/P30 pin output is inverted again (and becomes inactive). The TM0 counter is cleared and counting up is started again after the value of TM0 has coincided with the value of CR00. The output of the TO0/P30 pin, however, is not inverted even if coincidence occurs next time. TM0 is cleared and started and the output of the TO0/P30 pin is inverted only when the software trigger is set. The active level of the TO0/P30 pin is determined by selecting the initial value of the TO0/P30 pin output of the TOC0 register.

Note that, when using the one-shot pulse output function with the software trigger, the OSPT flag must not be set to 1 while the one-shot pulse is output. To output the one-shot pulse again, do so after INTTM00, which is an interrupt request that occurs when TM0 coincides with CR00, has occurred.

In the example presented in this section, the software trigger is designed by using key input, and “H” active output is produced 10 ms after for 1 ms.

Figure 5-16. Timing of One-Shot Pulse Output Operation by Software Trigger



Remark F_TRG: flag indicating that output of the one-shot pulse is in progress. For details, refer to **(2) Example of use.**

(1) Description of package

<Public declaration symbol>

SOP_INIT: One-shot pulse output initial setting subroutine

<Register used>

None

<RAM used>

None

<Nesting level>

1 level 2 bytes

<Hardware used>

- 16-bit timer/event counter

<Initial setting>

```
• OSMS = #00000001B ; Oscillation mode select register: does not use divider circuit
• CLR1      P3.0      ; Clears output latch of bit 0 of port 3 to 0
• CLR1      PM3.0     ; Sets bit 0 of port mode register 3 in output mode
• CALL      !SOP_INIT ; Sets by subroutine SOP_INIT
```

<Starting>

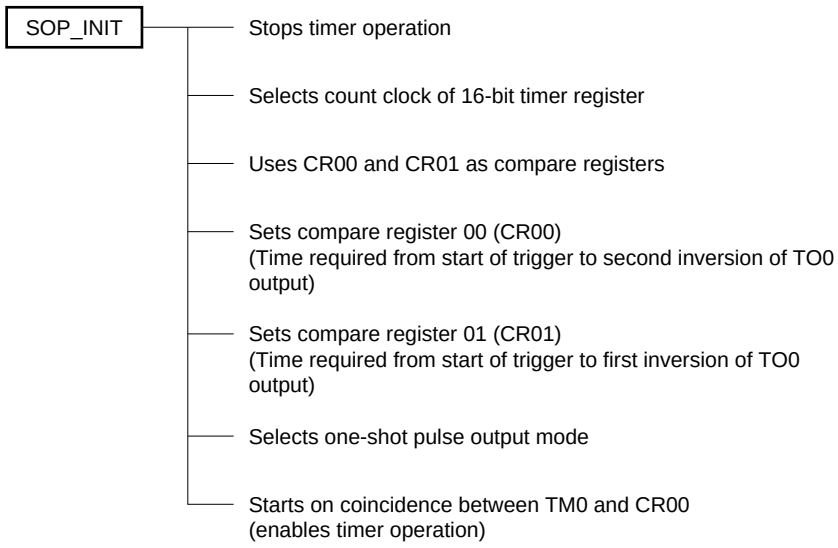
Set bit 6 (OSPT) of the 16-bit timer output control register (TOC0).

(2) Example of use

Because bit 6 (OSPT) of the 16-bit timer output control register (TOC0) is not set again while the pulse is output in the example of this package, the F_TRG flag is set as soon as the OSPT flag has been set as shown in Figure 5-16. Even if the next output request is issued while the F_TRG flag is set (i.e., while the pulse is output), the OSPT flag is not set. After the one-shot pulse has been output (INTTM00 occurs), clear the F_TRG flag.

```
EXTRN   SOP_INT

M1PR0       CSEG
RES_STA:
    OSMS=#00000001B             ; Does not use divider circuit
    CLR1    P3.0                ; Sets 0 to output latch if multiplexed pin is used
    CLR1    PM3.0               ; Sets output mode if multiplexed pin is used
    CALL    !SOP_INIT           ; One-shot pulse output initial setting routine
         .
         .
    if(key request issued)
        if_bit(!F_TRG)          ; Previous output ends?
            SET1    OSPT        ;     Clears and starts 16-bit counter
            SET1    F_TRG       ;     Sets one-shot pulse trigger flag
        endif                   ;
    endif                       ;
    if_bit(TMIF00)              ; End of one-shot pulse output?
        CLR1    F_TRG           ;     Clears one-shot trigger flag
        CLR1    TMIF00          ;     Clears TMIF00 request flag
    endif                       ;
         .
         .
```

(3) SPD chart



(4) Program list

```
PUBLIC  SOP_INIT

OPINIT     CSEG
SOP_INIT:
    TMC0=#00000000B          ; Stops timer operation
    TCL0=#01000000B          ; Count clock of 16-bit timer register:  1.05 MHz
    CRC0=#00000000B          ; Uses CR00 and CR01 as compare registers
    CR00=#11550–1            ; Sets compare register to 11 ms
    CR01=#10500–1            ; Sets compare register to 10 ms
    TOC0=#00110111B          ; Selects one-shot pulse mode
    TMC0=#00001100B          ; Starts on coincidence between TM0 and CR00 (enables timer operation)
    RET                      ;
    END
```

5.5 PPG Output

When using the 16-bit timer/event counter in the PPG (Programmable Pulse Generator) mode, the 16-bit timer mode control register (TMC0), capture/compare control register 0 (CRC0), and 16-bit timer output control register (TOC0) must be set.

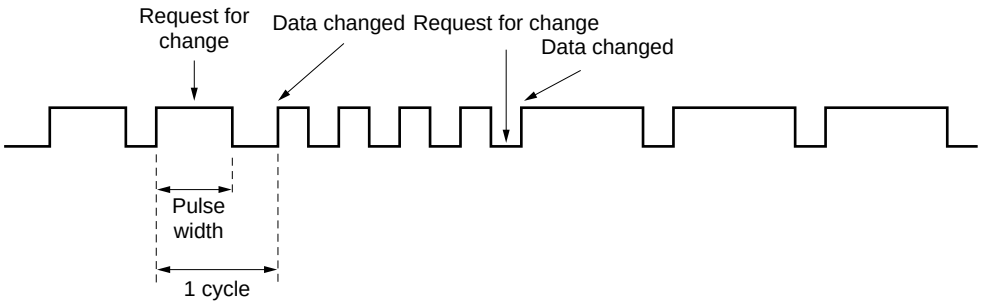
As the PPG output pulse, a square wave with a cycle specified by the count value set in advance to the 16-bit capture/compare register 00 (CR00) and a pulse width specified by the count value set in advance to the 16-bit capture/compare register 01 (CR01) is output from the TO0/P30 pin.

In the application example shown in this section, the output waveform is changed by using the PPG output. Data indicating the one cycle and pulse width of the output waveform is stored in ROM. This data is stored in the compare register.

The cycle and pulse width of the PPG output in this program can be changed in units of 1 ms to 10 ms. Therefore, the cycle can be set in a range of 2 to 10 ms, and the pulse width can be set in a range of 1 to 9 ms. If the cycle is equal to or less than the pulse width when the output waveform is changed, the data is not changed.

The output waveform is changed after the end of one output cycle. Figure 5-17 shows the PPG output waveform changing timing.

Figure 5-17. PPG Output Waveform Changing Timing



(1) Description of package

<Public declaration symbol>

- Subroutine name
  SPG_INIT : PPG output initial setting subroutine
- Data definition reference name of SPG_INIT routine
  PDAT : First address of data value for pulse width stored to compare register
  SDAT : First address of data value for cycle stored to compare register
- Input parameter of SPG_INIT routine
  PARUSU : Pulse width time storage area
  SAIKURU : 1 cycle time storage area
- Input/output parameters of SPG_INIT routine and INTTM00 interrupt
  PARUSUP : Pulse width time change data storage area
  SAIKURUP : 1-cycle time change data storage area

<Registers used>

SPG_INIT : Bank 0 AX, HL
INTTM00 : Bank 2 AX

<RAM used>

Name	Usage	Attribute	Bytes
PARUSU	Sets pulse width time	SADDR	1
SAIKURU	Sets 1-cycle time	SADDR	1
PARUSUP	Sets compare data value corresponding to pulse width time	SADDRP	2
SAIKURUP	Sets compare data value corresponding to 1 cycle time	SADDRP	2

<Flag used>

None

<Nesting level>

1 level 3 bytes

<Hardware used>

- 16-bit timer/event counter

<Initial setting>

- OSMS = #00000001B ; Oscillation mode select register: does not use divider circuit
- CLR1 P3.0 ; Clears output latch of bit 0 of port 3 to 0
- CLR1 PM3.0 ; Sets bit 0 of port mode register 3 in output mode
- CALL !PPG_INIT ; Sets by subroutine PPG_INIT

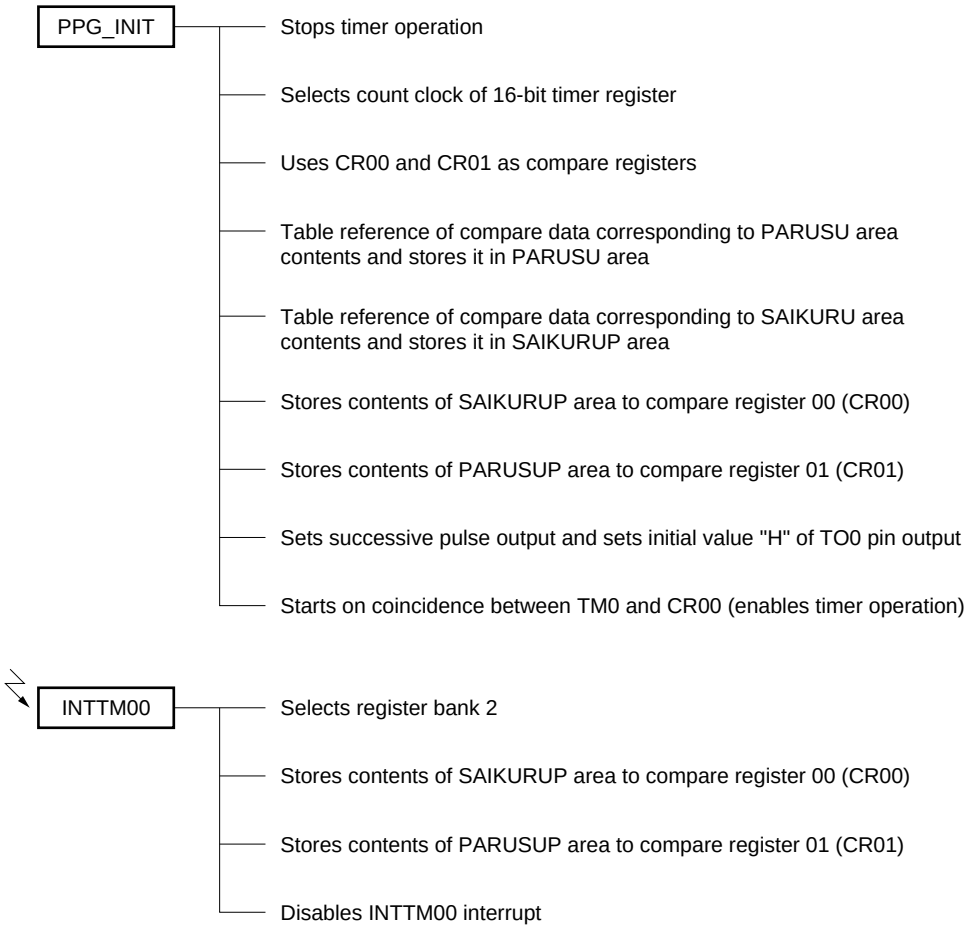
<Starting>

After the 16-bit timer/event counter has been reset and started, set pulse width time in the specified range to PARUSU in RAM and cycle time in the specified range to SAIKURU, and call subroutine PPG_INIT. When changing the PPG output waveform, clear the INTTM00 interrupt request flag to enable the interrupt after setting a compare data value corresponding to the pulse width in the specified range to PARUSUP, and a compare data value corresponding to the cycle time in the specified range to SAIKURUP.

(2) Example of use

```
EXTRN   SPG_INIT
EXTRN   SAIKURUP.PARUSUP
EXTRN   SAIKURU,PARUSU
EXTRN   PDAT,SDAT
;
SMIN        EQU     02H                 ; Minimum cycle time
PMIN        EQU     01H                 ; Minimum pulse width time
          .
          .
    OSMS=#00000001B                     ; Does not use divider circuit
    SAIKURU=#SMIN                       ; Sets initial cycle value
    PARUSU=#PMIN                        ; Sets initial pulse width value
    CLR1    P3.0                        ; Clears output latch to 0 if multiplexed pin is used
    CLR1    PM3.0                       ; Sets output mode if multiplexed pin is used
    CALL    !SPG_INIT                   ;
    EI                                  ;
          .
          .
    if(request for changing square wave)
        if(SAIKURU > PARUSU) (A)        ; If SAIKURUP > PARUSU
            A=PARUSU                    ; Data 1 → address XXX0
            A--                         ; Data 2 → address XXX2
            A <<= 1                     ; Data 3 → address XXX4
            X=A                         ;                .
            A=#0                        ; Table reference of low-order 8 bits of value stored to
            AX+=#PDAT                   ; compare register
            HL=AX                       ;     X register ← low-order 8 bits
            X=[HL] (A)                  ;
            HL++                        ; Table reference of high-order 8 bits of value stored to compare register
            A=[HL]                      ;     A register ← high-order 8 bits
            PARUSUP=AX                  ;
;                                       ;
            A=SAIKURU                   ; Cycle time storage processing
            A--                         ;
            A--                         ;
            A <<= 1                     ;
            X=A                         ;
            A=#0                        ;
            AX+=#SDAT                   ;
            HL=AX                       ;
            X=[HL] (A)                  ;
            HL++                        ;
            A=[HL]                      ;
            SAIKURUP=AX                 ;
            CLR1    TMIF00              ; Clears request flag
            CLR1    TMMK00              ; Enables compare register 00 interrupt
        endif                           ;     No data change
    endif                               ;
         .
         .
```

(3) SPD chart


PPG_INIT
Stops timer operation
Selects count clock of 16-bit timer register
Uses CR00 and CR01 as compare registers
Table reference of compare data corresponding to PARUSU area contents and stores it in PARUSU area
Table reference of compare data corresponding to SAIKURU area contents and stores it in SAIKURUP area
Stores contents of SAIKURUP area to compare register 00 (CR00)
Stores contents of PARUSUP area to compare register 01 (CR01)
Sets successive pulse output and sets initial value "H" of TO0 pin output
Starts on coincidence between TM0 and CR00 (enables timer operation)
INTTM00
Selects register bank 2
Stores contents of SAIKURUP area to compare register 00 (CR00)
Stores contents of PARUSUP area to compare register 01 (CR01)
Disables INTTM00 interrupt

(4) Program list

```
PUBLIC  SPG_INIT,PDAT,SDAT
PUBLIC  SAIKURU,PARUSU
EXTRN   SAIKURUP,PARUSUP
;
;*********************************************
;               RAM definition
;*********************************************
PPGRAM     DSEG    SADDR              ;
SAIKURU:   DS      1                  ; 1 cycle time storage area
PARUSU:    DS      1                  ; Pulse width storage area
;
;*********************************************
;          PPG output initial setting
;*********************************************
PPGINIT    CSEG
SPG_INIT:
    TMC0=#00000000B                   ; Stops timer operation
    TCL0=#00100000B                   ; Count clock of 16-bit timer register:  4.19 MHz
    CRC0=#00000000B                   ; Uses CR00 and CR01 as compare register
    A=PARUSU                          ;     Data 1 → address XXX0
    A--                               ;     Data 2 → address XXX2
    A <<= 1                           ;     Data 3 → address XXX4
    X=A                               ;                 .
    A=#0                              ; Table reference of low-order 8 bits of value stored to
    AX+=#PDAT                         ; compare register
    HL=AX                             ;     X register ← low-order 8 bits
    X=[HL] (A)                        ;
    HL++                              ; Table reference of high-order 8 bits of value stored to compare register
    A=[HL]                            ;     A register ← high-order 8 bits
    PARUSUP=AX                        ;
;                                     ;
    A=SAIKURU                         ; Cycle time storage processing
    A--                               ;
    A--                               ;
    A <<= 1                           ;
    X=A                               ;
    A=#0                              ;
    AX+=#SDAT                         ;
    HL=AX                             ;
    X=[HL] (A)                        ;
    HL++                              ;
    A=[HL]                            ;
    SAIKURUP=AX                       ;
    CR00=SAIKURUP (AX)                ; Sets compare register to 2 ms
    CR01=PARUSUP (AX)                 ; Sets compare register to 1 ms
    TOC0=#00011011B                   ; Sets successive pulse output and initial value 'H'
    TMC0=#00001100B                   ; Starts on coincidence between TM0 and CR00 (enables timer opera-
    RET                                 tion)
                                      ;
```

```
PDAT:                                   ;
    DW  4201                            ; Address XXX0
    DW  8403                            ; Address XXX2
    DW  12605                           ; Address XXX4
    DW  16807                           ;
    DW  21009                           ;
    DW  25211                           ;
    DW  29413                           ;
    DW  33615                           ;
    DW  37817                           ;
SDAT:                                   ;
    DW  8403                            ; Address XXX0
    DW  12605                           ; Address XXX2
    DW  16807                           ; Address XXX4
    DW  21009                           ;
    DW  25211                           ;
    DW  29413                           ;
    DW  33615                           ;
    DW  37817                           ;
    DW  42019                           ;
    END


PUBLIC      PARUSUP,SAIKURUP
;
VETM00      CSEG    AT 20H
            DW      INTTM00
;
P2RAM       DSEG    SADDRP
PARUSUP:    DS      2                   ; Pulse width time changing data storage area
SAIKURUP:   DS      2                   ; 1 cycle time changing data storage area
;*****************************************************************
;     PPG output (cycle pulse width time changing interrupt)
;*****************************************************************
TM00        CSEG                        ;
INTTM00:                                ;
    SEL     RB2                         ; Selects bank 2
    CR01=PARUSUP (AX)                   ; CR00, CR01 ← stores pulse width and cycle time changing data
    CR00=SAIKURUP (AX)                  ;
    SET1    TMMK00                      ; Disables compare register 00 interrupt
    RETI                                ;
    END
```

CHAPTER 6 APPLICATIONS OF 8-BIT TIMER/EVENT COUNTER

The 8-bit timer/event counter of the 78K/0 series has three functions: interval timer, external event counter, and square wave output. Two channels of 8-bit timers/event counters are provided and these timers/event counters can be used as a 16-bit timer/event counter when connected in cascade.

The 8-bit timers/event counters are set by the following registers:

- Timer clock select register 1 (TCL1)
- 8-bit timer mode control register (TMC1)
- 8-bit timer output control register (TOC1)
- Port mode register 3 (PM3)
- Port 3 (P3)

Figure 6-1. Format of Timer Clock Select Register 1 (μPD78054, 78054Y, 78064, 78064Y, 78078, 78078Y, 780058, 780058Y, 780308, 780308Y, 78058F, 78058FY, 78064B, 78075B, 78075BY subseries, μPD78070A, 78070AY)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
TCL1	TCL17	TCL16	TCL15	TCL14	TCL13	TCL12	TCL11	TCL10	FF41H	00H	R/W

TCL13	TCL12	TCL11	TCL10	Selects count clock of 8-bit timer register 1		
					MCS = 1	MCS = 0
0	0	0	0	Falling edge of TI1		
0	0	0	1	Rising edge of TI1		
0	1	1	0	$f_{XX}/2$	$f_X/2$ (2.5 MHz)	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$ (1.25 MHz)	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.8 kHz)
1	1	1	0	$f_{XX}/2^9$	$f_X/2^9$ (9.8 kHz)	$f_X/2^{10}$ (4.9 kHz)
1	1	1	1	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.4 kHz)	$f_X/2^{12}$ (1.2 kHz)
Others				Setting prohibited		

TCL17	TCL16	TCL15	TCL14	Selects count clock of 8-bit timer register 2		
					MCS = 1	MCS = 0
0	0	0	0	Falling edge of TI2		
0	0	0	1	Rising edge of TI2		
0	1	1	0	$f_{XX}/2$	$f_X/2$ (2.5 MHz)	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$ (1.25 MHz)	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.8 kHz)
1	1	1	0	$f_{XX}/2^9$	$f_X/2^9$ (9.8 kHz)	$f_X/2^{10}$ (4.9 kHz)
1	1	1	1	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.4 kHz)	$f_X/2^{12}$ (1.2 kHz)
Others				Setting prohibited		

Caution Before writing new data to TCL1, stop the timer operation once.

Remarks 1. f_{XX} : main system clock frequency (f_X or $f_X/2$)

2. f_X : main system clock oscillation frequency

3. TI1 : input pin of 8-bit timer register 1

4. TI2 : input pin of 8-bit timer register 2

5. MCS: bit 0 of oscillation mode select register (OSMS)

6. ( ) : at f_X = 5.0 MHz

Figure 6-2. Format of Timer Clock Select Register 1 (μPD78098, 78098B subseries)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
TCL1	TCL17	TCL16	TCL15	TCL14	TCL13	TCL12	TCL11	TCL10	FF41H	00H	R/W

TCL13	TCL12	TCL11	TCL10	Selects count clock of 8-bit timer register 1
0	0	0	0	Falling edge of TI1
0	0	0	1	Rising edge of TI1
0	1	1	0	$f_{XX}/2$ (2.0 MHz)
0	1	1	1	$f_{XX}/2^2$ (1.0 MHz)
1	0	0	0	$f_{XX}/2^3$ (500 kHz)
1	0	0	1	$f_{XX}/2^4$ (250 kHz)
1	0	1	0	$f_{XX}/2^5$ (125 kHz)
1	0	1	1	$f_{XX}/2^6$ (62.5 kHz)
1	1	0	0	$f_{XX}/2^7$ (31.3 kHz)
1	1	0	1	$f_{XX}/2^8$ (15.6 kHz)
1	1	1	0	$f_{XX}/2^9$ (7.8 kHz)
1	1	1	1	$f_{XX}/2^{11}$ (2.0 kHz)
Others				Setting prohibited

TCL17	TCL16	TCL15	TCL14	Selects count clock of 8-bit timer register 2
0	0	0	0	Falling edge of TI2
0	0	0	1	Rising edge of TI2
0	1	1	0	$f_{XX}/2$ (2.0 MHz)
0	1	1	1	$f_{XX}/2^2$ (1.0 MHz)
1	0	0	0	$f_{XX}/2^3$ (500 kHz)
1	0	0	1	$f_{XX}/2^4$ (250 kHz)
1	0	1	0	$f_{XX}/2^5$ (125 kHz)
1	0	1	1	$f_{XX}/2^6$ (62.5 kHz)
1	1	0	0	$f_{XX}/2^7$ (31.3 kHz)
1	1	0	1	$f_{XX}/2^8$ (15.6 kHz)
1	1	1	0	$f_{XX}/2^9$ (7.8 kHz)
1	1	1	1	$f_{XX}/2^{11}$ (2.0 kHz)
Others				Setting prohibited

Caution Before writing new data to TCL1, stop the timer operation once.

Remarks 1. f_{XX} : main system clock frequency

2. TI1: input pin of 8-bit timer register 1

3. TI2: input pin of 8-bit timer register 2

4. ( ): at f_{XX} = 4.0 MHz

★ **Figure 6-3. Format of Timer Clock Select Register 1 (μPD780018, 780018Y subseries)**

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
TCL1	TCL17	TCL16	TCL15	TCL14	TCL13	TCL12	TCL11	TCL10	FF41H	00H	R/W

TCL13	TCL12	TCL11	TCL10	Selects count clock of 8-bit timer register 1	
0	0	0	0	Falling edge of TI1	
0	0	0	1	Rising edge of TI1	
0	1	1	0	$f_{XX}/2$	$f_X/2$ (2.5 MHz)
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$ (1.25 MHz)
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)
1	1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)
1	1	1	0	$f_{XX}/2^9$	$f_X/2^9$ (9.8 kHz)
1	1	1	1	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.4 kHz)
Others				Setting prohibited	

TCL17	TCL16	TCL15	TCL14	Selects count clock of 8-bit timer register 2	
0	0	0	0	Falling edge of TI2	
0	0	0	1	Rising edge of TI2	
0	1	1	0	$f_{XX}/2$	$f_X/2$ (2.5 MHz)
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$ (1.25 MHz)
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)
1	1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)
1	1	1	0	$f_{XX}/2^9$	$f_X/2^9$ (9.8 kHz)
1	1	1	1	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.4 kHz)
Others				Setting prohibited	

Caution Before writing new data to TCL1, stop the timer operation once.

Remarks 1. f_{XX} : main system clock frequency (f_X)

2. f_X : main system clock oscillation frequency

3. TI1 : input pin of 8-bit timer register 1

4. TI2 : input pin of 8-bit timer register 2

5. ( ) : at f_X = 5.0 MHz

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

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
TMC1	0	0	0	0	0	TMC12	TCE2	TCE1	FF49H	00H	R/W

TCE1	Controls operation of 8-bit timer register 1
0	Stops operation (clears TM1 to 0)
1	Enables operation

TCE2	Controls operation of 8-bit timer register 2
0	Stops operation (clears TM2 to 0)
1	Enables operation

TMC12	Selects operation mode
0	8-bit timer register × 2 channel mode (TM1, TM2)
1	16-bit timer register × 1 channel mode (TMS)

Cautions 1. Before changing the operation mode, stop the timer operation.

2. When using the two 8-bit timer registers as a one 16-bit timer register, enable or stop the operation by using TCE1.

Figure 6-5. Format of 8-Bit Timer Output Control Register

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
TOC1	LVS2	LVR2	TOC15	TOE2	LVS1	LVR1	TOC11	TOE1	FF4FH	00H	R/W

TOE1	Controls output of 8-bit timer/event counter 1
0	Disables output (port mode)
1	Enables output

TOC11	Controls timer output F/F of 8-bit timer/event counter 1
0	Disables reverse operation
1	Enables reverse operation

LVS1	LVR1	Sets status of timer output F/F of 8-bit timer/ event counter 1
0	0	Not affected
0	1	Resets timer output F/F (to 0)
1	0	Sets timer output F/F (to 1)
1	1	Setting prohibited

TOE2	Controls output of 8-bit timer/event counter 2
0	Disables output (port mode)
1	Enables output

TOC15	Controls timer output F/F of 8-bit timer/event counter 2
0	Disables reverse operation
1	Enables reverse operation

LVS2	LVR2	Sets status of timer output F/F of 8-bit timer/ event counter 2
0	0	Not affected
0	1	Resets timer output F/F (to 0)
1	0	Sets timer output F/F (to 1)
1	1	Setting prohibited

Cautions 1. Before setting TOC1, be sure to stop the timer operation.

2. LVS1, LVS2, LVR1, and LVR2 are always 0 when they are read.

Figure 6-6. Format of Port Mode Register 3

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W

PM3n	Selects input/output mode of P3n pin (n = 0-7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

6.1 Setting of Interval Timer

When using an 8-bit timer/event counter as an interval timer, set an operation mode by the 8-bit timer mode control register (TMC1) and interval time by the timer clock select register 1 (TCL1).

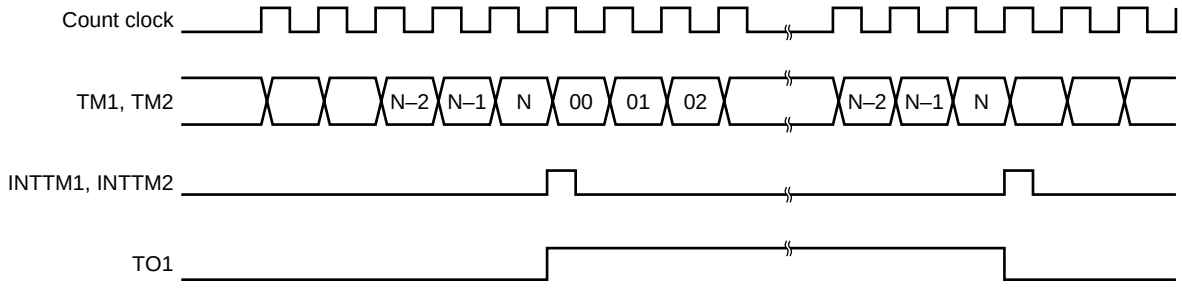
After that, set the values of the compare registers (CR10 and CR20) from the setup time and count clock. The setup time is determined by using the following expression:

Setup time = (Compare register value + 1) × Count clock cycle

The setup time can be calculated in the same manner regardless of whether each 8-bit timer/event counter is used or two 8-bit timers/event counters are used as a 16-bit timer/event counter. The count clock when two 8-bit timers/event counters are used as a 16-bit timer/event counter, however, is selected by the bits 0 through 3 (TCL10 through TCL13) of TCL1.

Examples of the modes of the 8-bit timers and 16-bit timer are described next.

Figure 6-7. Count timing of 8-Bit Timers



6.1.1 Setting of 8-bit timers

In this example, 8-bit timer 2 is used to set two types of interval times: 500 μs and 100 ms.

(a) To set interval of 500 μs

<1> Setting of TMC1
Select the 8-bit timer register × 2 channel mode and enables the operation of the 8-bit timer 2.
<2> Setting of TCL1
Select $f_{XX}/2^4$ that allows setting of 500 μs or more and has the highest resolution (OSMS = 01H).
<3> Setting of CR20

$$500\ \mu s = (N + 1) \times \frac{1}{4.19\ MHz/2^4}$$

$$N = 500\ \mu s \times 4.19\ MHz/2^4 - 1 \fallingdotseq 130$$

(1) Program list

```
OSMS = #00000001B   ;  Does not use divider circuit
TCL1 = #10011001B   ;  Selects fXX/2^4 as count clock
CR20  =  #130
TMC1  =  #00000010B
```

(b) To set interval of 100 ms

<1> Setting of TMC1
Select the 8-bit timer register × 2 channel mode and enables the operation of the 8-bit timer 2.
<2> Setting of TCL1
Select $f_{XX}/2^{11}$ that allows setting of 100 ms or more and has the highest resolution (OSMS = 01H).
<3> Setting of CR20

$$100\ ms = (N + 1) \times \frac{1}{4.19\ MHz/2^{11}}$$

$$N = 100\ ms \times 4.19\ MHz/2^{11} - 1 \fallingdotseq 204$$

(1) Program list

```
OSMS = #00000001B   ;  Does not use divider circuit
TCL1 = #11111111B   ;  Selects fXX/2^11 as count clock
CR20  =  #204
TMC1  =  #00000010B
```

6.1.2 Setting of 16-bit timer

In this example, 8-bit timers 1 and 2 are connected in cascade as a 16-bit timer to set two types of interval times: 500 ms and 10 s.

(a) To set interval of 500 ms

<1> Setting of TMC1
Select the 16-bit timer register × 1 channel mode and enables the operation of the 8-bit timers 1 and 2.

<2> Setting of TCL1
Select $f_{XX}/2^5$ that allows setting of 500 ms or more and has the highest resolution (OSMS = 01H).

<3> Setting of CR10 and CR20

$$500\ \text{ms} = \frac{N + 1}{4.19\ \text{MHz}/2^5}$$

$N = 500\ \text{ms} \times 4.19\ \text{MHz}/2^5 - 1 \fallingdotseq 65468 = \text{FF6CH}$
CR10 = 6CH, CR20 = FFH

(1) Program list

```
OSMS = #00000001B   ;  Does not use divider circuit
TCL1 = #00001010B
CR10 = #06CH        ;  Sets 65468 to CR10 and CR20
CR20 = #0FFH        ;  CR10 = 6CH, CR20 = FFH
TMC1 = #00000111B
```

(b) To set interval of 10 s

<1> Setting of TMC1
Select the 16-bit timer register × 1 channel mode and enable the operation of the 8-bit timers 1 and 2.

<2> Setting of TCL1
Select $f_{XX}/2^{11}$ that allows setting of 10 s or more and has the highest resolution (OSMS = 01H).

<3> Setting of CR10 and CR20

$$10\ \text{s} = \frac{N + 1}{4.19\ \text{MHz}/2^{11}}$$

$N = 10\ \text{s} \times 4.19\ \text{MHz}/2^{11} - 1 \fallingdotseq 20458 = \text{4FEAH}$
CR10 = EAH, CR20 = 4FH

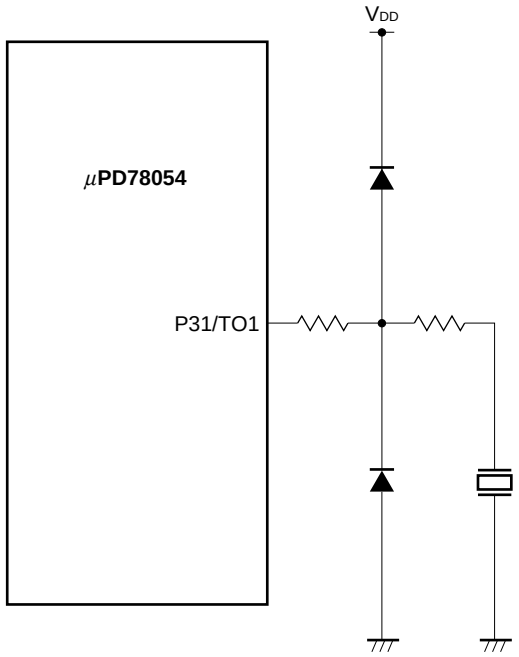
(1) Program list

```
OSMS = #00000001B   ;  Does not use divider circuit
TCL1 = #00001111B
CR10 = #0EAH        ;  Sets 20458 to CR10 and CR20
CR20 = #4FH         ;  CR10 = EAH, CR20 = 4FH
TMC1 = #00000111B
```

6.2 Musical Scale Generation

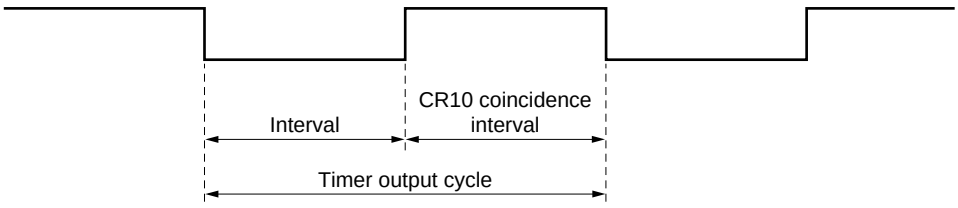
This section shows an example of a program that uses the square wave output (P31/TO1) of an 8-bit timer/event counter and generates a musical scale by supplying pulses to an external buzzer.

Figure 6-8. Musical Scale Generation Circuit



The output frequency of the P31/TO1 pin is set by the count clock and a compare register. In this example, the central frequency of the musical scale is set to a range of 523 to 1046 Hz. Therefore, $f_{XX}/2^5$ is selected as the count clock (oscillation mode select register: OSMS = 01H). Table 6-1 shows the musical scale, the set value of the compare register, and frequency of the output pulse. Because one cycle of the timer output is created when the value of the timer coincides with the value of the compare register two times, the interval time is set as half a cycle time.

Figure 6-9. Timer Output and Interval



As for the time length of a sound, the output time is determined by setting an interval time with 8-bit timer/event counter 2 and by counting the number of times the interrupt generated by the timer/event counter. In this example, 8-bit timer/event counter 2 is set to 20 ms.

Table 6-1. Musical Scale and Frequency

Musical Scale	Musical Scale Frequency Hz	Compare Register Value	Output Frequency Hz
Do	523.25	124	524.3
Re	587.33	111	585.1
Mi	659.25	98	662.0
Fa	698.46	93	697.2
So	783.98	83	780.2
La	880.00	73	885.6
Tee	987.77	65	993.0
Do	1046.5	62	1040

The format of the data table for this program is shown below.

```
TABLE:
        DB  musical scale data 1, sound length data 1
        DB  musical scale data 2, sound length data 2
                  ⋮                       ⋮
        DB  musical scale data n, sound length data n
        DB  0,                    0
```

The musical scale data is set to 0 for rest, and the sound length data is set to 0 for the end of data.

Example Number of counts of 8-bit timer/event counter to output sound for 1 second
Number of counts = 1 s/20 ms = 50 (50 is set as number of counts)

This program sequentially outputs do, re, mi, and so on, for 1 second each.

(1) Description of package

<Public declaration symbol>

MLDY: Subroutine name of musical scale generation program

<Registers used>

Bank 0: A, B, HL

<RAM used>

Name	Usage	Attribute	Bytes
POINT	Stores pointer value of table data	SADDR	1
LNG	Counts sound length data		

<Nesting>

1 level 3 bytes

<Hardware used>

- 8-bit timer/event counters 1 and 2
- P31/TO1

<Initial setting>

- Sets by subroutine MLDY
- Enables interrupt

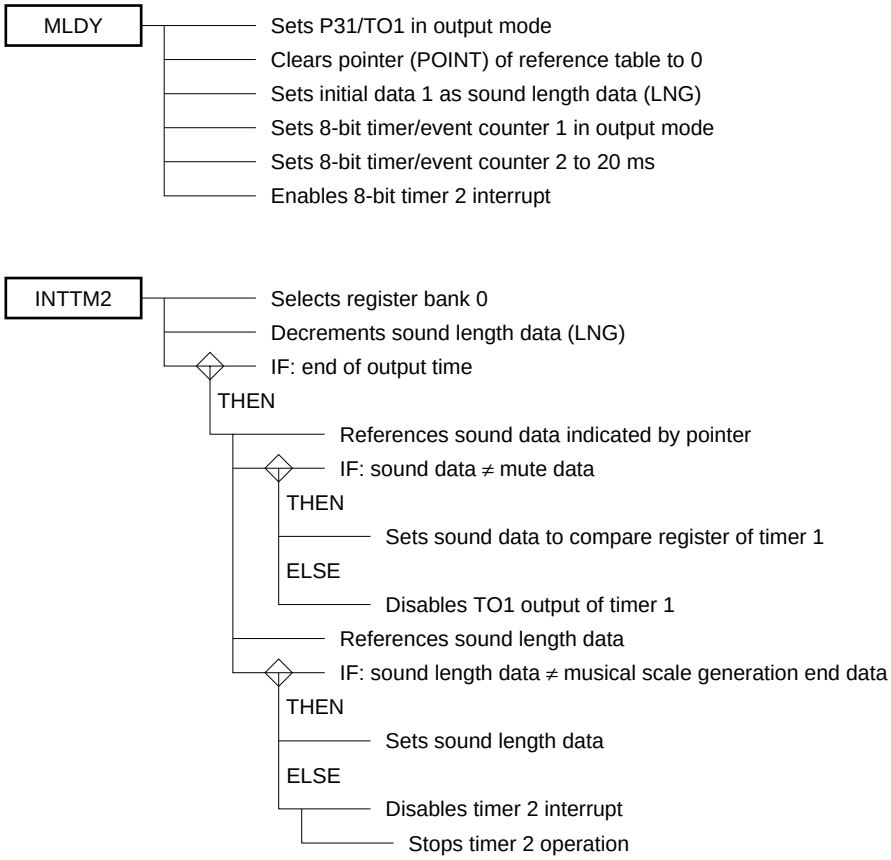
<Starting>

- Call subroutine MLDY

(2) Example of use

```
EXTRN  MLDY
  ⋮
CALL    !MLDY
⊟
```

(3) SPD chart


MLDY
Sets P31/TO1 in output mode
Clears pointer (POINT) of reference table to 0
Sets initial data 1 as sound length data (LNG)
Sets 8-bit timer/event counter 1 in output mode
Sets 8-bit timer/event counter 2 to 20 ms
Enables 8-bit timer 2 interrupt
INTTM2
Selects register bank 0
Decrements sound length data (LNG)
IF: end of output time
THEN
References sound data indicated by pointer
IF: sound data ≠ mute data
THEN
Sets sound data to compare register of timer 1
ELSE
Disables TO1 output of timer 1
References sound length data
IF: sound length data ≠ musical scale generation end data
THEN
Sets sound length data
ELSE
Disables timer 2 interrupt
Stops timer 2 operation

(4) Program list

```
        PUBLIC  MLDY

VETM2   CSEG    AT 26H
        DW      INTTM2                          ; Sets vector address of 8-bit timer/event counter

ML_DAT  DSEG    SADDR
POINT:  DS      1                               ; Pointer for table data
LNG:    DS      1                               ; Sound length data

;************************************************
;*      Musical scale generation initialize
;************************************************
ML_SEG  CSEG
MLDY:
        CLR     PM3.1                           ; Sets P3.1 in output mode
        POINT=#0                                ; Initial setting of pointer
        LGN=#1
        OSMS=#00000001B                         ; Does not use divider circuit
        TOC1=#00000011B                         ; Sets TO1 output mode
        TCL1=#11101010B
        CR20=#163                               ; Sets timer 2 to 20 ms
        TMC1=#00000010B                         ; Enables timer 2 operation
        CLR1    TMMK2                           ; Enables timer 2 interrupt
        RET
$EJECT
```

```
;*********************************************
;      Sets musical scale generation data
;*********************************************
TM2_SEG CSEG
INTTM2:
        SEL RB0
        LNG--
        if(LNG==#0)
            B=POINT (A)
            HL=#TABLE                          ; Sets table first address
            A=[HL+B]
            if(A!=#0)
                CLR1    TCE1                   ; Sets sound data
                CR10=A
                SET1    TOE1
                SET1    TCE1
            else
                CLR1    TOE1
            endif

            B++                                ; Increments pointer
            A=[HL+B]                           ; Loads sound length data
            if(A!=#0)                          ; Sound output in progress?
                LNG=A                          ; Sets sound length data
                B++
                POINT=B (A)
            else
                SET1    TMMK2                  ; Disables timer 2 interrupt
                CLR1    TCE2                   ; Stops timer 2 operation
            endif
        endif
        RETI
;*********************************************
;          Musical scale data table
;*********************************************
TABLE:
        DB  124,50                             ; Do
        DB  111,50                             ; Re
        DB  98,50                              ; Mi
        DB  93,50                              ; Fa
        DB  83,50                              ; So
        DB  73,50                              ; La
        DB  65,50                              ; Tee
        DB  62,50                              ; Do
        DB  00,00                              ; End
```

[MEMO]

CHAPTER 7 APPLICATIONS OF WATCH TIMER

The watch timer of the 78K/0 series has a watch timer function that causes the timer to overflow every 0.5 second by using the main system clock or subsystem clock as the clock source, and an interval timer function that allows you to set six types of reference times. These two functions can be simultaneously used.

The watch timer is set by using timer clock select register 2 (TCL2) and watch timer mode control register (TMC2).

Figure 7-1. Format of Timer Clock Select Register 2
(μPD78054, 78054Y, 78064, 78064Y, 78078, 78078Y, 780058, 780058Y, 780308, 780308Y, 78058F, 78058FY, 78064B, 78075B, 78075BY subseries, μPD78070A, 78070AY)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
TCL2	TCL27	TCL26	TCL25	TCL24	0	TCL22	TCL21	TCL20	FF42H	00H	R/W

TCL22	TCL21	TCL20	Selects count clock of watchdog timer		
				MCS = 1	MCS = 0
0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)	$f_X/2^4$ (313 kHz)
0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)	$f_X/2^5$ (156 kHz)
0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)	$f_X/2^6$ (78.1 kHz)
0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)
1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.8 kHz)
1	1	0	$f_{XX}/2^9$	$f_X/2^9$ (9.8 kHz)	$f_X/2^{10}$ (4.9 kHz)
1	1	1	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.4 kHz)	$f_X/2^{12}$ (1.2 kHz)

TCL24	Selects count clock of watch timer		
		MCS = 1	MCS = 0
0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	f_{XT} (32.768 kHz)		

TCL27	TCL26	TCL25	Selects frequency of buzzer output		
				MCS = 1	MCS = 0
0	×	×	Disables buzzer output		
1	0	0	$f_{XX}/2^9$	$f_X/2^9$ (9.8 kHz)	$f_X/2^{10}$ (4.9 kHz)
1	0	1	$f_{XX}/2^{10}$	$f_X/2^{10}$ (4.9 kHz)	$f_X/2^{11}$ (2.4 kHz)
1	1	0	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.4 kHz)	$f_X/2^{12}$ (1.2 kHz)
1	1	1	Setting prohibited		

Caution Before writing new data to TCL2, stop the timer operation once.

Remarks 1. f_{XX} : main system clock frequency (f_X or $f_X/2$)
2. f_X : main system clock oscillation frequency
3. f_{XT} : subsystem clock oscillation frequency
4. × : don't care
5. MCS: bit 0 of oscillation mode select register (OSMS)
6. ( ) : at f_X = 5.0 MHz or f_{XT} = 32.768 kHz

Figure 7-2. Format of Timer Clock Select Register 2 (μPD78098, 78098B subseries)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
TCL2	TCL27	TCL26	TCL25	TCL24	0	TCL22	TCL21	TCL20	FF42H	00H	R/W

TCL22	TCL21	TCL20	Selects count clock of watchdog timer
0	0	0	$f_{XX}/2^3$ (500 kHz)
0	0	1	$f_{XX}/2^4$ (250 kHz)
0	1	0	$f_{XX}/2^5$ (125 kHz)
0	1	1	$f_{XX}/2^6$ (62.5 kHz)
1	0	0	$f_{XX}/2^7$ (31.3 kHz)
1	0	1	$f_{XX}/2^8$ (15.6 kHz)
1	1	0	$f_{XX}/2^9$ (7.8 kHz)
1	1	1	$f_{XX}/2^{11}$ (2.0 kHz)

TCL24	Selects count clock of watch timer
0	$f_{XX}/2^7$ (31.3 kHz)
1	f_{XT} (32.768 kHz)

TCL27	TCL26	TCL25	Selects frequency of buzzer output
0	×	×	Disables buzzer output
1	0	0	$f_{XX}/2^9$ (7.8 kHz)
1	0	1	$f_{XX}/2^{10}$ (3.9 kHz)
1	1	0	$f_{XX}/2^{11}$ (1.95 kHz)
1	1	1	Setting prohibited

Caution Before writing new data to TCL2, stop the timer operation once.

Remarks 1. f_{XX}: main system clock frequency

2. f_{XT}: subsystem clock oscillation frequency

3. × : don't care

4. ( ): at f_{XX} = 4.0 MHz or f_{XT} = 32.768 kHz

★ **Figure 7-3. Format of Timer Clock Select Register 2 (μPD780018, 780018Y subseries)**

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
TCL2	TCL27	TCL26	TCL25	TCL24	0	TCL22	TCL21	TCL20	FF42H	00H	R/W

TCL22	TCL21	TCL20	Selects count clock of watchdog timer	
0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)
0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)
0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)
0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)
1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)
1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)
1	1	0	$f_{XX}/2^9$	$f_X/2^9$ (9.8 kHz)
1	1	1	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.4 kHz)

TCL24	Selects count clock of watch timer	
0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)
1	f_{XT}	

TCL27	TCL26	TCL25	Selects frequency of buzzer output	
0	×	×	Disables buzzer output	
1	0	0	$f_{XX}/2^9$	$f_X/2^9$ (9.8 kHz)
1	0	1	$f_{XX}/2^{10}$	$f_X/2^{10}$ (4.9 kHz)
1	1	0	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.4 kHz)
1	1	1	Setting prohibited	

Caution Before writing new data to TCL2, stop the timer operation once.

Remarks 1. f_{XX}: main system clock frequency (f_X)

2. f_X : main system clock oscillation frequency

3. f_{XT}: subsystem clock oscillation frequency

4. × : don't care

5. ( ): at f_X = 5.0 MHz or f_{XT} = 32.768 kHz

Figure 7-4. Format of Watch Timer Mode Control Register (μPD78054, 78054Y, 78064, 78064Y, 78078, 78078Y, 780018, 780018Y, 780058, 780058Y, 780308, 780308Y, 78058F, 78058FY, 78064B, 78075B, 78075BY subseries, μPD78070A, 78070AY)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
TMC2	0	TMC26	TMC25	TMC24	TMC23	TMC22	TMC21	TMC20	FF4AH	00H	R/W

TMC20	Selects watch operation mode
0	Normal operation mode (sets flag at $f_W/2^{14}$)
1	Fast-forward mode (sets flag at $f_W/2^5$)

TMC21	Controls operation of prescaler
0	Clears after operation stopped
1	Enables operation

TMC22	Controls operation of 5-bit counter
0	Clears after operation stopped
1	Enables operation

TMC23	Selects set time of watch timer		
	At f_{XX} = 5.0 MHz	At f_{XX} = 4.19 MHz	At f_{XT} = 32.768 kHz
0	$2^{14}/f_W$ (0.4 sec)	$2^{14}/f_W$ (0.5 sec)	$2^{14}/f_W$ (0.5 sec)
1	$2^{13}/f_W$ (0.2 sec)	$2^{13}/f_W$ (0.25 sec)	$2^{13}/f_W$ (0.25 sec)

TMC26	TMC25	TMC24	Selects interval time of prescaler		
			At f_{XX} = 5.0 MHz	At f_{XX} = 4.19 MHz	At f_{XT} = 32.768 kHz
0	0	0	$2^4/f_W$ (410 μs)	$2^4/f_W$ (488 μs)	$2^4/f_W$ (488 μs)
0	0	1	$2^5/f_W$ (819 μs)	$2^5/f_W$ (977 μs)	$2^5/f_W$ (977 μs)
0	1	0	$2^6/f_W$ (1.64 ms)	$2^6/f_W$ (1.95 ms)	$2^6/f_W$ (1.95 ms)
0	1	1	$2^7/f_W$ (3.28 ms)	$2^7/f_W$ (3.91 ms)	$2^7/f_W$ (3.91 ms)
1	0	0	$2^8/f_W$ (6.55 ms)	$2^8/f_W$ (7.81 ms)	$2^8/f_W$ (7.81 ms)
1	0	1	$2^9/f_W$ (13.1 ms)	$2^9/f_W$ (15.6 ms)	$2^9/f_W$ (15.6 ms)
Others			Setting prohibited		

Caution Do not often clear the prescaler when the watch timer is used.

Remarks 1. f_W : watch timer clock frequency ($f_{XX}/2^7$ or f_{XT})

2. f_{XX}: main system clock frequency (f_X or $f_X/2$)

3. f_X : main system clock oscillation frequency

4. f_{XT}: subsystem clock oscillation frequency

Figure 7-5. Format of Watch Timer Mode Control Register (μPD78098, 78098B subseries)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
TMC2	0	TMC26	TMC25	TMC24	TMC23	TMC22	TMC21	TMC20	FF4AH	00H	R/W

TMC23	TMC20	Selects set time of watch flag
0	0	2^{14}/fw (0.5s)
1		2^{13}/fw (0.25s)
0	1	2^{5}/fw (977 μs)
1		2^{4}/fw (488 μs)

TMC21	Controls operation of prescaler
0	Clears after operation stopped
1	Enables operation

TMC22	Controls operation of 5-bit counter
0	Clears after operation stopped
1	Enables operation

TMC26	TMC25	TMC24	Selects interval time of prescaler
0	0	0	2^{4}/fw (488 μs)
0	0	1	2^{5}/fw (977 μs)
0	1	0	2^{6}/fw (1.95 ms)
0	1	1	2^{7}/fw (3.91 ms)
1	0	0	2^{8}/fw (7.81 ms)
1	0	1	2^{9}/fw (15.6 ms)
Others			Setting prohibited

Caution Do not often clear the prescaler when the watch timer is used.

Remarks 1. fw : watch timer clock frequency ($f_X/2^8$ or f_{XT})

2. ( ): at fw = 32.768 kHz

7.1 Watch and LED Display Program

As an example of using the watch timer, this section introduces a program that counts time by using an 0.5 second overflow and dynamically displays LED at intervals of 1.95 ms.

To count time, an overflow flag is tested each time a subroutine is called. When the flag is set, time is counted up in seconds. Because an overflow occurs every 0.5 second, it takes 1 minute to count 120 times. The overflow flag is tested at intervals of 1.95 ms so that the flag is tested without fail. The watch of this program is 24-hour watch. The high-order and low-order digits of minute and hour data are stored in separate areas of memory.

Figure 7-6. Concept of Watch Data

Second data	Minute data		Hour data	
0-120	Low-order digit 0-9	High-order digit 0-5	Low-order digit 0-9	High-order digit 0-2

As LED dynamic display, four digits are displayed with the display digit changed at intervals of 1.95 ms. In this example, the high-order 4 bits of P3 are used as a digit signal, and P5 that can directly drive an LED is selected as a segment signal.

The digit of an LED specified by a display digit area (DIGCT) in an LED display area is displayed. To change the digit signal, the segment signal is turned off so that the adjacent digits are not displayed.

Figure 7-7. LED Display Timing

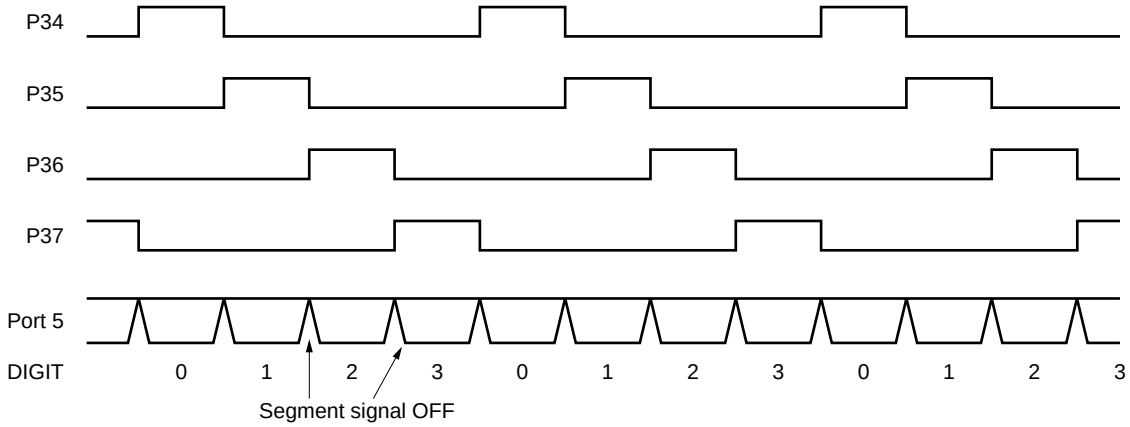
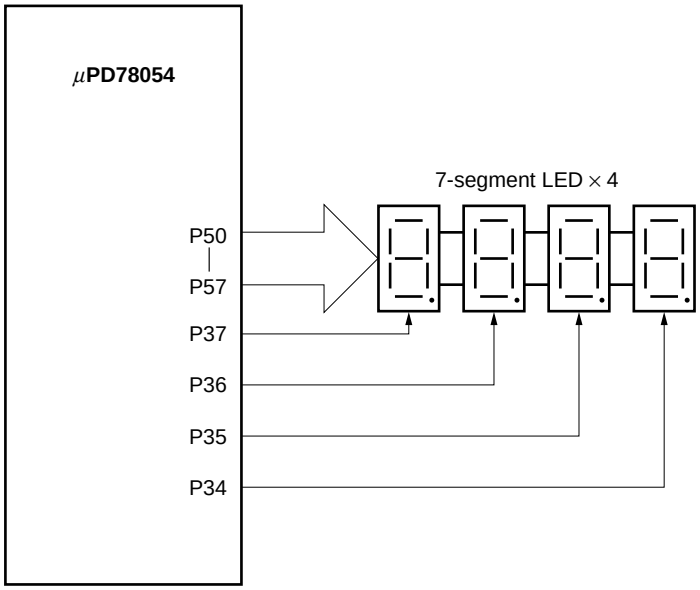


Figure 7-8. Circuit Example of Watch Timer



(1) Description of package

<Public declaration symbol>

SECD : second data storage area
MINDP : minute data storage area
HOURDP : hour data storage area
LEDDP : LED display area

<Register used>

Bank 0: AX, B, HL

<RAM used>

Name	Usage	Attribute	Bytes
MINDP	Stores minute data	SADDRP	2
HOURDP	Stores hour data		
SECD	Stores second data		1
DIGCT	Stores LED display digit data		
LEDDP	LED display data		4

<Hardware used>

- Watch timer
- P34-37
- P5

<Initial setting>

- TMC2 = #00100110B ; 0.5-second watch operation at 1.95 ms interval
- TMMK3 = 0 ; enables watch timer interrupt

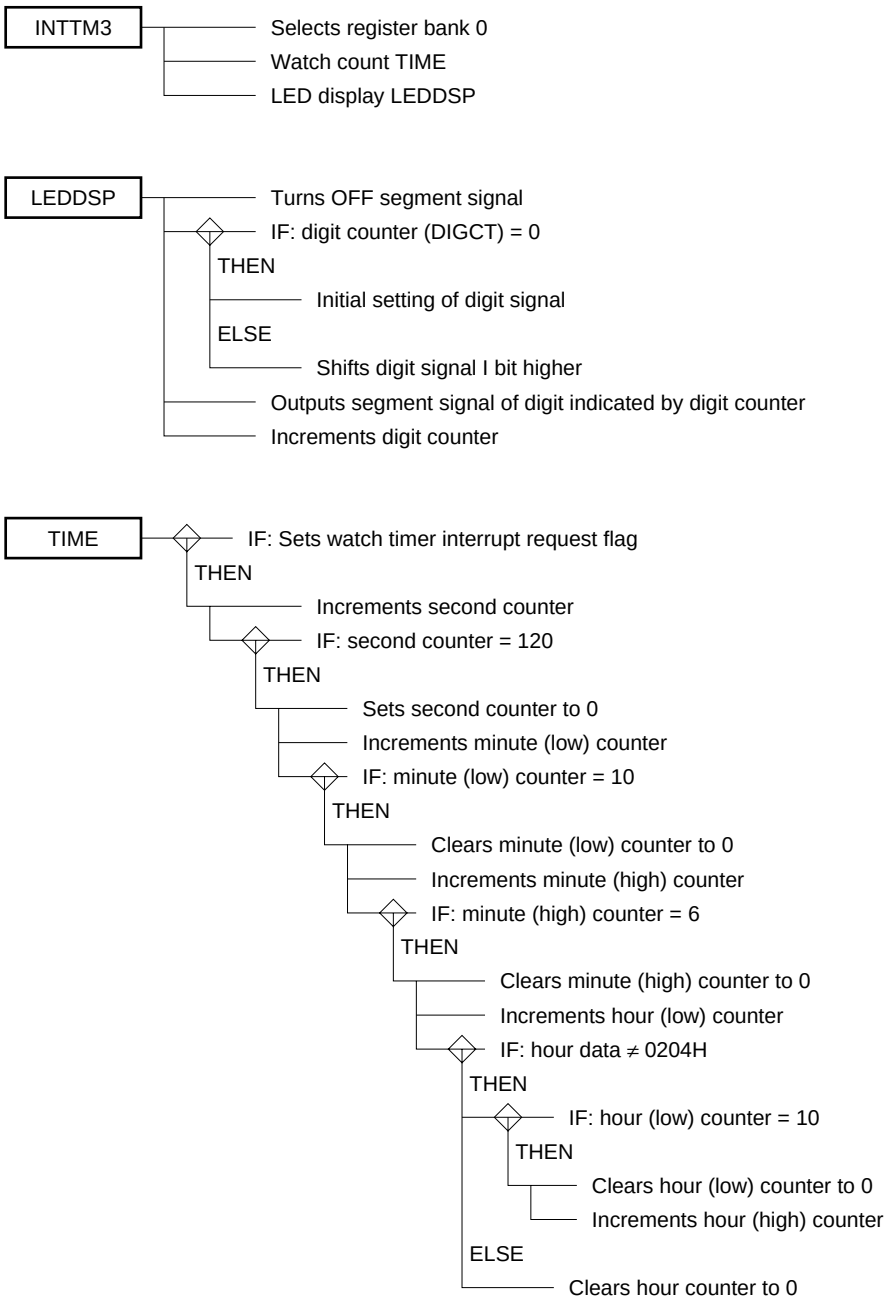
<Starting>

Started by the interval timer interrupt request of the watch timer.

(2) Example of use

```
EXTRN  MINDP, HOURDP, SECD, LEDDP

TMC2 = #00100110B   ;  0.5-second watch operation at 1.95 ms interval
CLR1    TMMK3       ;  Enables watch timer interrupt
EI
```

(3) SPD chart


INTTM3
Selects register bank 0
Watch count TIME
LED display LEDDSP
LEDDSP
Turns OFF segment signal
IF: digit counter (DIGCT) = 0
THEN
Initial setting of digit signal
ELSE
Shifts digit signal I bit higher
Outputs segment signal of digit indicated by digit counter
Increments digit counter
TIME
IF: Sets watch timer interrupt request flag
THEN
Increments second counter
IF: second counter = 120
THEN
Sets second counter to 0
Increments minute (low) counter
IF: minute (low) counter = 10
THEN
Clears minute (low) counter to 0
Increments minute (high) counter
IF: minute (high) counter = 6
THEN
Clears minute (high) counter to 0
Increments hour (low) counter
IF: hour data ≠ 0204H
THEN
IF: hour (low) counter = 10
THEN
Clears hour (low) counter to 0
Increments hour (high) counter
ELSE
Clears hour counter to 0

(4) Program list

```
        PUBLIC  HOURDP,MINDP,SECD,LEDDP

WT_DATP DSEG    SADDRP
MINDP:  DS      2                       ; Minute data storage area
HOURDP: DS      2                       ; Hour data storage area
SECD:   DS      1                       ; Second data storage area
DIGCT:  DS      1                       ; LED display digit area
LEDDP:  DS      4                       ; LED display area

VETM3   CSEG    AT 1EH
        DW      INTTM3                  ; Sets vector address of watch timer

;************************************
;*    Interval interrupt processing
;************************************
TM3_SEG CSEG
INTTM3:
        SEL RB0
        CALL    !TIME
        CALL    !LEDDPSP
        RETI
```

```
;**********************************
;           LED display
;**********************************
LEDDPSP:
       P5=#0FFH                     ; Turns OFF segment output
       DIGCT&=#00000011B            ; Adjusts digit counter (0-3)
       if(DIGCT==#0)
           A=P3
           A&=#00001111B            ; Initial setting of digit signal (high-order 4 bits)
           A|=#00010000B
           P3=A
       else
           A=P3
           A&=#11110000B            ; Shifts high-order 4 bits
           X=A
           A=P3
           A+=X
           P3=A
       endif

       B=DIGCT (A)                  ; Sets address of display data
       HL=#LEDDP                    ; Display area first address
       B=[HL+B] (A)                 ; Sets display data
       HL=#SEGDT                    ; Conversion to segment data
       P5=[HL+B] (A)                ; Outputs segment signal

       DIGCT++
       RET
SEGDT:
       DB  11000000B                ; 0
       DB  11111001B                ; 1
       DB  10100100B                ; 2
       DB  10110000B                ; 3
       DB  10011001B                ; 4
       DB  10010010B                ; 5
       DB  10000010B                ; 6
       DB  11111000B                ; 7
       DB  10000000B                ; 8
       DB  10010000B                ; 9
       DB  10001000B                ; A
       DB  10000011B                ; B
       DB  11000110B                ; C
       DB  10100001B                ; D
       DB  10000110B                ; E
       DB  10001110B                ; F
$EJECT
```

```
;*******************************
;*       Watch count up
;*******************************
TIME:                                           ; 0.5 second test
       if_bit(WTIF)
           CLR1       WTIF                      ; 120 = 60 seconds/0.5
           SECD++
           if(SECD==#120)
               SECD=#0                          ; Increments minute (low)
               (MINDP+0)++                      ; Carry occurs
               if((MINDP+0)==#10)
                   (MINDP+0)=#0                 ; Increments minute (high)
                   (MINDP+1)++                  ; Carry occurs
                   if(MINDP+1==#6)
                       (MINDP+1)=#0
                       (HOURDP+0)++             ; Hour data 24?
                       if(HOURDP!=#0204H) (AX)  ; Carry occurs
                           if((HOURDP+0)==#10)
                               (HOURDP+0)=#0
                               (HOURDP+1)++
                           endif
                       else
                           HOURDP=#0000H
                       endif
                   endif
               endif
           endif
       endif
       RET
```

[MEMO]

CHAPTER 8 APPLICATIONS OF SERIAL INTERFACE

The 78K/0 series is provided with the serial interface shown in Table 8-1.

★ **Table 8-1. Serial Interface Channel of Each Subseries**

Configuration of Serial Interface / Subseries	Channel 0				Channel 1		Channel 2		Channel 3	Channel 4	Channel 5
	3-wire	2-wire	SBI	I^2C bus	3-wire	3-wire with automatic transmission/ reception function	3-wire	UART	3-wire	3-wire with time-division function	I^2C bus (multi-master supporting)
μPD78054	○	○	○	×	○	○	○	○	×	×	×
μPD78054Y	○	○	×	○	○	○	○	○	×	×	×
μPD78064	○	○	○	×	×	×	○	○	×	×	×
μPD78064Y	○	○	×	○	×	×	○	○	×	×	×
μPD78078	○	○	○	×	○	○	○	○	×	×	×
μPD78078Y	○	○	×	○	○	○	○	○	×	×	×
μPD78083	×	×	×	×	×	×	○	○	×	×	×
μPD78098	○	○	○	×	○	○	○	○	×	×	×
μPD780018	×	×	×	×	○	○	×	×	×	○	×
μPD780018Y	×	×	×	×	○	○	×	×	×	○	○
μPD780058	○	○	○	×	○	○	○	○ Note	×	×	×
μPD780058Y	○	○	×	○	○	○	○	○ Note	×	×	×
μPD780308	○	○	○	×	×	×	○	○ Note	○	×	×
μPD780308Y	○	○	×	○	×	×	○	○ Note	○	×	×
μPD78058F	○	○	○	×	○	○	○	○	×	×	×
μPD78058FY	○	○	×	○	○	○	○	○	×	×	×
μPD78064B	○	○	○	×	×	×	○	○	×	×	×
μPD78070A	○	○	○	×	○	○	○	○	×	×	×
μPD78070AY	○	○	×	○	○	○	○	○	×	×	×
μPD78075B	○	○	○	×	○	○	○	○	×	×	×
μPD78075BY	○	○	×	○	○	○	○	○	×	×	×
μPD78098B	○	○	○	×	○	○	○	○	×	×	×

Note With time-division transfer function

Remark ○: Function provided, ×: Function not provided

The serial interface of the 78K/0 series has a different function depending on the subseries, as shown in Table 8-1. This chapter explains each function and application example of the serial interface. The function supported by each subseries are listed in Table 8-2. For details of application examples of using the serial interface function of a specific subseries, refer to the section or paragraph marked ○ in this table.

★ **Table 8-2. Items Supported by Each Subseries**

Item / Subseries	8.1.1 Communication in 2-wire serial I/O mode	8.1.2 Communication in I^2C bus mode	8.2 Interface with OSD LSI (μPD6451A)	8.3 Interface in SBI Mode	8.4 Interface in 3-Wire Serial I/O Mode	8.5 Interface in Asynchronous Serial Interface (UART) Mode
μPD78054	○	–	○	○	○	○
μPD78054Y	○	○	○	–	○	○
μPD78064	○	–	–	○	○	○
μPD78064Y	○	○	–	–	○	○
μPD78078	○	–	○	○	○	○
μPD78078Y	○	○	○	–	○	○
μPD78083	–	–	–	–	–	○
μPD78098	○	–	○	○	○	○
μPD780018	–	–	○	–	–	–
μPD780018Y	–	–	○	–	–	–
μPD780058	○	–	○	○	○	○
μPD780058Y	○	○	○	–	○	○
μPD780308	○	–	–	○	○	○
μPD780308Y	○	○	–	–	○	○
μPD78058F	○	–	○	○	○	○
μPD78058FY	○	○	○	–	○	○
μPD78064B	○	–	–	○	○	○
μPD78070A	○	–	○	○	○	○
μPD78070AY	○	○	○	–	○	○
μPD78075B	○	–	○	○	○	○
μPD78075BY	○	○	○	–	○	○
μPD78098B	○	–	○	○	○	○

The functions and operations of the serial interface are specified by using the following registers:

★ **Table 8-3. Registers of Serial Interface**

Serial Interface	Register Used
Channel 0	• Timer clock select register (TCL3) • Serial operation mode register 0 (CSIM0) • Serial bus interface control register (SBIC) • Interrupt timing specification register (SINT)
Channel 1	• Timer clock select register (TCL3) • Serial operation mode register 1 (CSIM1) • Automatic data transmission/reception control register (ADTC) • Automatic data transmission/reception interval specification register (ADTI)
Channel 2	• Serial operation mode register 2 (CSIM2) • Asynchronous serial interface mode register (ASIM) • Asynchronous serial interface status register (ASIS) • Baud rate generator control register (BRGC) • Serial interface pin select register (SIPS)**Note**

Note This register is provided only on the μPD780058, 780058Y, 780308, and 780308Y subseries.

Remark This chapter describes the register formats and application examples of serial interface channels 0, 1, and 2. For details of the register formats of channels 3, 4, and 5, refer to the User's Manual of each subseries.

Figure 8-1. Format of Timer Clock Select Register 3
(μPD78054, 78078, 780058, 78058F, 78075B subseries, μPD78070A)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
TCL3	TCL37	TCL36	TCL35	TCL34	TCL33	TCL32	TCL31	TCL30	FF43H	88H	R/W

TCL33	TCL32	TCL31	TCL30	Selects serial clock of serial interface channel 0		
					MCS = 1	MCS = 0
0	1	1	0	$f_{XX}/2$	Setting prohibited	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$ (1.25 MHz)	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.8 kHz)
Others				Setting prohibited		

TCL37	TCL36	TCL35	TCL34	Selects serial clock of serial interface channel 1		
					MCS = 1	MCS = 0
0	1	1	0	$f_{XX}/2$	Setting prohibited	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$ (1.25 MHz)	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.8 kHz)
Others				Setting prohibited		

Caution Before writing new data to TCL3, stop serial transfer once.

Remarks 1. f_{XX} : main system clock frequency (f_X or $f_X/2$)
2. f_X : main system clock oscillation frequency
3. MCS : bit 0 of oscillation mode select register (OSMS)
4. ( ) : at f_X = 5.0 MHz

Figure 8-2. Format of Timer Clock Select Register 3
(μPD78054Y, 78078Y, 780058Y, 78058FY, 78075BY subseries, μPD78070AY)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
TCL3	TCL37	TCL36	TCL35	TCL34	TCL33	TCL32	TCL31	TCL30	FF43H	88H	R/W

TCL33	TCL32	TCL31	TCL30	Selects serial clock of serial interface channel 0					
				Serial clock in I^2C bus mode			Serial clock in 3-wire serial I/O or 2-wire serial I/O mode		
					MCS = 1	MCS = 0		MCS = 1	MCS = 0
0	1	1	0	$f_{XX}/2^5$	Setting prohibited	$f_X/2^6$ (78.1 kHz)	$f_{XX}/2$	Setting prohibited	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)	$f_{XX}/2^2$	$f_X/2^2$ (1.25 MHz)	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.77 kHz)	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_{XX}/2^9$	$f_X/2^9$ (9.77 kHz)	$f_X/2^{10}$ (4.88 kHz)	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_{XX}/2^{10}$	$f_X/2^{10}$ (4.88 kHz)	$f_X/2^{11}$ (2.44 kHz)	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.44 kHz)	$f_X/2^{12}$ (1.22 kHz)	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	1	0	1	$f_{XX}/2^{12}$	$f_X/2^{12}$ (1.22 kHz)	$f_X/2^{13}$ (0.61 kHz)	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.8 kHz)
Others				Setting prohibited					

TCL37	TCL36	TCL35	TCL34	Selects serial clock of serial interface channel 1		
					MCS = 1	MCS = 0
0	1	1	0	$f_{XX}/2$	Setting prohibited	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$ (1.25 MHz)	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.8 kHz)
Others				Setting prohibited		

Caution Before writing new data to TCL3, stop serial transfer once.

Remarks 1. f_{XX} : main system clock frequency (f_X or $f_X/2$)
2. f_X : main system clock oscillation frequency
3. MCS : bit 0 of oscillation mode select register (OSMS)
4. ( ) : at f_X = 5.0 MHz

Figure 8-3. Format of Timer Clock Select Register 3 (μPD78064, 780308, 78064B subseries)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
TCL3	1	0	0	0	TCL33	TCL32	TCL31	TCL30	FF43H	88H	R/W

TCL33	TCL32	TCL31	TCL30	Selects serial clock of serial interface channel 0		
					MCS = 1	MCS = 0
0	1	1	0	$f_{XX}/2$	Setting prohibited	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$ (1.25 MHz)	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.8 kHz)
Others				Setting prohibited		

Cautions 1. Clear bits 4 through 6 to 0 and set bit 7 to 1.

2. Before writing new data to TCL3, stop serial transfer once.

Remarks 1. f_{XX} : main system clock frequency (f_X or $f_X/2$)

2. f_X : main system clock oscillation frequency

3. MCS : bit 0 of oscillation mode select register (OSMS)

4. ( ) : at f_X = 5.0 MHz

Figure 8-4. Format of Timer Clock Select Register 3 (μPD78064Y, 780308Y subseries)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
TCL3	1	0	0	0	TCL33	TCL32	TCL31	TCL30	FF43H	88H	R/W

TCL33	TCL32	TCL31	TCL30	Selects serial clock of serial interface channel 0					
				Serial clock in I^2C bus mode			Serial clock in 3-wire serial I/O or 2-wire serial I/O mode		
					MCS = 1	MCS = 0		MCS = 1	MCS = 0
0	1	1	0	$f_{XX}/2^5$	Setting prohibited	$f_X/2^6$ (78.1 kHz)	$f_{XX}/2$	Setting prohibited	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)	$f_{XX}/2^2$	$f_X/2^2$ (1.25 MHz)	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.77 kHz)	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_{XX}/2^9$	$f_X/2^9$ (9.77 kHz)	$f_X/2^{10}$ (4.88 kHz)	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_{XX}/2^{10}$	$f_X/2^{10}$ (4.88 kHz)	$f_X/2^{11}$ (2.44 kHz)	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.44 kHz)	$f_X/2^{12}$ (1.22 kHz)	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	1	0	1	$f_{XX}/2^{12}$	$f_X/2^{12}$ (1.22 kHz)	$f_X/2^{13}$ (0.61 kHz)	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.8 kHz)
Others				Setting prohibited					

Cautions 1. Clear bits 4 through 6 to 0 and set bit 7 to 1.

2. Before writing new data to TCL3, stop serial transfer once.

Remarks 1. f_{XX} : main system clock frequency (f_X or $f_X/2$)

2. f_X : main system clock oscillation frequency

3. MCS : bit 0 of oscillation mode select register (OSMS)

4. ( ) : at f_X = 5.0 MHz

Figure 8-5. Format of Timer Clock Select Register 3 (μPD78098, 78098B subseries)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
TCL3	TCL37	TCL36	TCL35	TCL34	TCL33	TCL32	TCL31	TCL30	FF43H	88H	R/W

TCL33	TCL32	TCL31	TCL30	Selects serial clock of serial interface channel 0
0	1	1	0	$f_{XX}/2$**Note**
0	1	1	1	$f_{XX}/2^2$ (1.0 MHz)
1	0	0	0	$f_{XX}/2^3$ (500 kHz)
1	0	0	1	$f_{XX}/2^4$ (250 kHz)
1	0	1	0	$f_{XX}/2^5$ (125 kHz)
1	0	1	1	$f_{XX}/2^6$ (62.5 kHz)
1	1	0	0	$f_{XX}/2^7$ (31.3 kHz)
1	1	0	1	$f_{XX}/2^8$ (15.6 kHz)
Others				Setting prohibited

TCL37	TCL36	TCL35	TCL34	Selects serial clock of serial interface channel 1
0	1	1	0	$f_{XX}/2$**Note**
0	1	1	1	$f_{XX}/2^2$ (1.0 MHz)
1	0	0	0	$f_{XX}/2^3$ (500 kHz)
1	0	0	1	$f_{XX}/2^4$ (250 kHz)
1	0	1	0	$f_{XX}/2^5$ (125 kHz)
1	0	1	1	$f_{XX}/2^6$ (62.5 kHz)
1	1	0	0	$f_{XX}/2^7$ (31.3 kHz)
1	1	0	1	$f_{XX}/2^8$ (15.6 kHz)
Others				Setting prohibited

Note Can be set only when the main system clock frequency is 5.0 MHz or less.

Caution Before writing new data to TCL3, stop serial transfer once.

Remarks 1. f_{XX} : main system clock frequency

2. ( ) : at f_{XX} = 4.0 MHz

★ **Figure 8-6. Format of Timer Clock Select Register 3 (μPD780018, 780018Y subseries)**

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
TCL3	TCL37	TCL36	TCL35	TCL34	1	0	0	0	FF43H	88H	R/W

TCL37	TCL36	TCL35	TCL34	Selects serial clock of serial interface channel 1	
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$ (1.25 MHz)
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)
1	1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)
Others				Setting prohibited	

Caution Before writing new data to TCL3, stop serial transfer once.

Remarks 1. f_{XX} : main system clock frequency (f_X)

2. f_X : main system clock oscillation frequency

3. ( ) : at f_X = 5.0 MHz

Figure 8-7. Format of Serial Operating Mode Register 0 (μPD78054, 78064, 78078, 78098, 780058, 780308, 78058F, 78064B, 78075B, 78098B subseries, μPD78070A)(1/2)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W Note 1

R/W	CSIM01	CSIM00	Selects clock of serial interface channel 0
	0	×	Clock externally input to $\overline{SCK0}$ pin
	1	0	Output of 8-bit timer register 2 (TM2)
	1	1	Clock specified by bits 0 through 3 of timer clock select register 3 (TCL3)

R/W	CSIM04	CSIM03	CSIM02	PM25	P25	PM26	P26	PM27	P27	Operation mode	First bit	Function of SI0/SB0/P25 pin	Function of SO0/SB1/P26 pin	Function of $\overline{SCK0}$/P27 pin
	0	×	0	Note 2	Note 2	0	0	0	1	3-wire serial	MSB	SI0 Note 2	SO0	$\overline{SCK0}$
			1	1	×					I/O mode	LSB	(input)	(CMOS output)	(CMOS I/O)
	1	0	0	Note 3	Note 3	0	0	0	1	SBI mode	MSB	P25	SB1	$\overline{SCK0}$
				×	×							(CMOS I/O)	(N-ch open drain I/O)	(CMOS I/O)
			1	0	0	Note 3	Note 3	0	1			SB0	P26	
						×	×					(N-ch open drain I/O)	(CMOS I/O)	
	1	1	0	Note 3	Note 3	0	0	0	1	2-wire serial	MSB	P25	SB1	$\overline{SCK0}$
				×	×					I/O mode		(CMOS I/O)	(N-ch open drain I/O)	(N-ch open drain I/O)
			1	0	0	Note 3	Note 3	0	1			SB0	P26	
						×	×					(N-ch open drain I/O)	(CMOS I/O)	

R/W	WUP	Controls wake-up function Note 4
	0	Generates interrupt request signal in all modes each time serial transfer is executed
	1	Generates interrupt request signal when address received after bus has been released (when CMDD = RELD = 1) coincides with data of slave address register in SBI mode

Notes 1. Bit 6 (COI) is a read-only bit.

2. When only the transmission function is used, this pin can be used as P25 (CMOS I/O).

3. These pins can be used as port pins.

★ **4.** When using the wake-up function (WUP = 1), clear bit 5 (SIC) of the interrupt timing specification register (SINT) to 0.

★ **Caution Do not change the operation mode (3-wire serial I/O/2-wire serial I/O/SBI) while the operation of the serial interface channel 0 is enabled. To change the operation mode, stop the serial operation.**

Remark × : don't care
PM××: Port mode register
P×× : Output latch of port

Figure 8-7. Format of Serial Operating Mode Register 0 (μPD78054, 78064, 78078, 78098, 780058, 780308, 78058F, 78064B, 78075B, 78098B subseries, μPD78070A)(2/2)

R

COI	Slave address comparison result flag[Note]
0	Data of slave address register does not coincide with data of serial I/O shift register
1	Data of slave address register coincides with data of serial I/O shift register

R/W

CSIE0	Controls operation of serial interface channel 0
0	Stops operation
1	Enables operation

Note COI is 0 when CSIE0 = 0.

★ **Caution Do not change the operation mode (3-wire serial I/O/2-wire serial I/O/SBI) while the operation of the serial interface channel 0 is enabled. To change the operation mode, stop the serial operation.**

Figure 8-8. Format of Serial Operating Mode Register 0 (μPD78054Y, 78064Y, 78078Y, 780058Y, 780308Y, 78058FY, 78075BY subseries, μPD78070AY) (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W Note 1

R/W	CSIM01	CSIM00	Selects clock of serial interface channel 0
	0	×	Clock externally input to $\overline{\text{SCK0}}$/SCL pin
	1	0	Output of 8-bit timer register 2 (TM2) Note 2
	1	1	Clock specified by bits 0 through 3 of timer clock select register 3 (TCL3)

R/W	CSIM04	CSIM03	CSIM02	PM25	P25	PM26	P26	PM27	P27	Operation mode	First bit	Function of SI0/SB0/SDA0/P25 pin	Function of SO0/SB1/SDA1/P26 pin	Function of $\overline{\text{SCK0}}$/SCL/P27 pin
	0	×	0	Note 3	Note 3	0	0	0	1	3-wire serial	MSB	SI0 Note 3	SO0	$\overline{\text{SCK0}}$
			1	1	×					I/O mode	LSB	(input)	(CMOS output)	(CMOS I/O)
	1	1	0	Note 4	Note 4	0	0	0	1	2-wire serial I/O mode or I^2C bus mode	MSB	P25 (CMOS I/O)	SB1 (N-ch open drain I/O)	$\overline{\text{SCK0}}$/SCL (N-ch open drain I/O)
			1	0	0	Note 4	Note 4	0	1			SB0/SDA0 (N-ch open drain I/O)	P26 (CMOS I/O)	

R/W	WUP	Controls wake-up function Note 5
	0	Generates interrupt request signal in all modes each time serial transfer is executed
	1	Generates interrupt request signal when address received after start condition has been detected (when CMDD = 1) coincides with data of slave address register in I^2C mode

Notes 1. Bit 6 (COI) is a read-only bit.

2. In the I^2C bus mode, the clock frequency is 1/16 of the clock frequency output by TO2

3. When only the transmission function is used, this pin can be used as P25 (CMOS I/O).

4. These pins can be used as port pins.

★ **5.** When using the wake-up function (WUP = 1), clear bit 5 (SIC) of the interrupt timing specification register (SINT) to 0. While WUP = 1, do not execute an instruction that writes data to the I/O shift register 0 (SIO0).

★ **Caution Do not change the operation mode (3-wire serial I/O/2-wire serial I/O/I^2C bus) while the operation of the serial interface channel 0 is enabled. To change the operation mode, stop the serial operation.**

Remark × : don't care
PM××: Port mode register
P×× : Output latch of port

Figure 8-8. Format of Serial Operating Mode Register 0 (μPD78054Y, 78064Y, 78078Y, 780058Y, 780308Y, 78058FY, 78075BY subseries, μPD78070AY) (2/2)

R

COI	Slave address comparison result flag[Note]
0	Data of slave address register does not coincide with data of serial I/O shift register
1	Data of slave address register coincides with data of serial I/O shift register

R/W

CSIE0	Controls operation of serial interface channel 0
0	Stops operation
1	Enables operation

Note COI is 0 when CSIE0 = 0.

★ **Caution Do not change the operation mode (3-wire serial I/O/2-wire serial I/O/I^2C bus) while the operation of the serial interface channel 0 is enabled. To change the operation mode, stop the serial operation.**

Figure 8-9. Format of Serial Bus Interface Control Register (μPD78054, 78064, 78078, 78098, 780058, 780308, 78058F, 78064B, 78075B, 78098B subseries, μPD78070A) (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
SBIC	BSYE	ACKD	ACKE	ACKT	CMDD	RELD	CMDT	RELT	FF61H	00H	R/W[Note]

R/W		
R/W	RELT	Used to output bus release signal. When RELT = 1, SO latch is set to 1. After SO latch has been set, this bit is automatically cleared to 0. It is also cleared to 0 when CSIE = 0.

R/W		
R/W	CMDT	Used to output command signal. When CMDT = 1, SO latch is cleared to 0. After SO latch has been cleared, this bit is automatically cleared to 0. It is also cleared to 0 when CSIE0 = 0.

R	RELD	Bus release detection
	Clear condition (RELD = 0)	Set condition (RELD = 1)
	• On execution of transfer start instruction • If values of SIO0 and SVA do not coincide when address is received • When CSIE0 = 0 • At $\overline{\text{RESET}}$ input	• When bus release signal (REL) is detected

R	CMDD	Command detection
	Clear condition (CMDD = 0)	Set condition (CMDD = 1)
	• On execution of transfer start instruction • When bus release signal (REL) is detected • When CSIE0 = 0 • At $\overline{\text{RESET}}$ input	• When command signal (CMD) is detected

R/W		
R/W	ACKT	Outputs acknowledge signal in synchronization with falling edge of $\overline{\text{SCK0}}$ clock immediately after instruction that sets this bit to 1 has been executed. After acknowledge signal has been output, this bit is automatically cleared to 0. ACKE is cleared to 0. This bit is also cleared to 0 when transfer of serial interface is started and when CSIE0 = 0.

Note Bits 2, 3, and 6 (RELD, CMDD, and ACKD) are read-only bits.

Remarks 1. Bits 0, 1, and 4 (RELD, CMDT, and ACKT) are cleared to 0 when they are read after data has been set.

2. CSIE0: Bit 7 of the serial operating mode register 0 (CSIM0)

Figure 8-9. Format of Serial Bus Interface Control Register (μPD78054, 78064, 78078, 78098, 780058, 780308, 78058F, 78064B, 78075B, 78098B subseries, μPD78070A) (2/2)

R/W

ACKE	Controls acknowledge signal output	
0	Disables automatic output of acknowledge signal (output by ACKT is enabled)	
1	Before completion of transfer	Acknowledge signal is output in synchronization with falling edge of 9th clock of $\overline{SCK0}$ (automatically output when ACKE = 1)
	After completion of transfer	Acknowledge signal is output in synchronization with falling edge of $\overline{SCK0}$ clock immediately after instruction that sets this bit to 1 has been executed (automatically output when ACKE = 1). However, this bit is not automatically cleared to 0 after acknowledge signal has been output.

R

ACKD	Acknowledge detection
Clear condition (ACKD = 0)	Set condition (ACKD = 1)
• Falling edge of $\overline{SCK0}$ clock immediately after busy mode has been released after execution of transfer start instruction • When CSIE0 = 0 • At $\overline{RESET}$ input	• When acknowledge signal ($\overline{ACK}$) is detected at rising edge of $\overline{SCK0}$ clock after completion of transfer

R/W

BSYE[Note]	Controls output of synchronization busy signal
0	Disables output of busy signal in synchronization with falling edge of $\overline{SCK0}$ clock immediately after instruction that clears this bit to 0 has been executed
1	Outputs busy signal at falling edge of $\overline{SCK0}$ clock following acknowledge signal

Note The busy mode can be released by starting serial interface transfer and receiving of an address signal. However, the BSYE flag is not cleared to 0.

Remark CSIE0: Bit 7 of the serial operating mode register 0 (CSIM0)

Figure 8-10. Format of Serial Bus Interface Control Register (μPD78054Y, 78064Y, 78078Y, 780058Y, 780308Y, 78058FY, 78075BY subseries, μPD78070AY) (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
SBIC	BSYE	ACKD	ACKE	ACKT	CMDD	RELD	CMDT	RELT	FF61H	00H	R/W[Note]

R/W		
R/W	RELT	Used to output stop condition. When RELT = 1, SO latch is set to 1. After SO latch has been set, this bit is automatically cleared to 0. It is also cleared to 0 when CSIE0 = 0.

R/W		
R/W	CMDT	Used to output start condition. When CMDT = 1, SO latch is cleared to 0. After SO latch has been cleared, this bit is automatically cleared to 0. It is also cleared to 0 when CSIE0 = 0.

R	RELD	Stop condition detection
	Clear condition (RELD = 0)	Set condition (RELD = 1)
	• On execution of transfer start instruction • If values of SIO0 and SVA do not coincide when address is received • When CSIE0 = 0 • At $\overline{\text{RESET}}$ input	• Stop condition is detected

R	CMDD	Start condition detection
	Clear condition (CMDD = 0)	Set condition (CMDD = 1)
	• On execution of transfer start instruction • When stop condition is detected • When CSIE0 = 0 • At $\overline{\text{RESET}}$ input	• When start condition is detected

R/W		
R/W	ACKT	Makes SDA0 (SDA1) low immediately after instruction that sets this bit to 1 (ACKT = 1) until next SCL falls. Used to generate $\overline{\text{ACK}}$ signal by software when 8-clock wait is selected. Cleared to 0 when transfer by serial interface is started and CSIE0 = 0

Note Bits 2, 3, and 6 (RELD, CMDD, and ACKD) are read-only bits.

Remark CSIE0: Bit 7 of the serial operating mode register 0 (CSIM0)

Figure 8-10. Format of Serial Bus Interface Control Register (μPD78054Y, 78064Y, 78078Y, 780058Y, 780308Y, 78058FY, 78075BY subseries, μPD78070AY) (2/2)

R/W

ACKE	Controls automatic output of acknowledge signal[Note 1]
0	Disables automatic output of acknowledge signal (output by ACKT is enabled). Used for transmission or reception with 8-clock wait selected[Note 2].
1	Enables automatic output of acknowledge signal. Acknowledge signal is output in synchronization with falling edge of 9th clock of $\overline{SCL}$ (automatically output when ACKE = 1). After output, this bit is not automatically cleared to 0. Used for reception when 9-clock wait is selected.

R

ACKD	Acknowledge detection	
Clear condition (ACKD = 0)		Set condition (ACKD = 1)
• On execution of transfer start instruction • When CSIE0 = 0 • At $\overline{RESET}$ input		• When acknowledge signal is detected at rising edge of SCL clock after completion of transfer

R/W

BSYE[Note 3]	Controls transmission N-ch open drain output in I^2C bus mode[Note 4]
0	Enables output (transmission)
1	Disables output (reception)

Notes 1. Set this bit before starting transfer.

2. Output the acknowledge signal on reception by using ACKT when 8-clock wait is selected.

3. The wait status can be released by starting transfer of serial interface or receiving an address signal. However, BSYE is not cleared to 0.

4. Be sure to set BSYE to 1 when using the wake-up function.

Remark CSIE0: Bit 7 of the serial operating mode register 0 (CSIM0)

Figure 8-11. Format of Interrupt Timing Specification Register (μPD78054, 78064, 78078, 78098, 780058, 780308, 78058F, 78064B, 78075B, 78098B subseries, μPD78070A)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
SINT	0	CLD	SIC	SVAM	0	0	0	0	FF63H	00H	R/W[Note 1]

R/W

SVAM	Bits of SVA used as slave address
0	Bits 0 through 7
1	Bits 1 through 7

R/W

SIC	Selects INTCSI0 interrupt source[Note 2]
0	Sets CSIIF0 at end of transfer of serial interface channel 0
1	Sets CSIIF0 at end of transfer of serial interface channel 0 or on detection of bus release

R

CLD	Level of $\overline{SCK0}$ pin[Note 3]
0	Low level
1	High level

Notes 1. Bit 6 (CLD) is a read-only bit.

2. Clear SIC to 0 when using the wake-up function in the SBI mode.

3. CLD is 0 when CSIE0 = 0.

Caution Be sure to clear bits 0 through 3 to 0.

Remark SVA : slave address register
CSIIF0 : interrupt request flag corresponding to INTCSI0
CSIE0 : bit 7 of the serial operating mode register 0 (CSIM0)

Figure 8-12. Format of Interrupt Timing Specification Register (μPD78054Y, 78064Y, 78078Y, 780058Y, 780308Y, 78058FY, 78075BY subseries, μPD78070AY) (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
SINT	0	CLD	SIC	SVAM	CLC	WREL	WAT1	WAT0	FF63H	00H	R/W[Note 1]

R/W	WAT1	WAT0	Controls wait and interrupt processing request
	0	0	Generates interrupt request at rising edge of 8th clock of $\overline{SCK0}$ (clock output goes into high-impedance state)
	0	1	Setting prohibited
	1	0	Used in I^2C bus mode (8-clock wait). Generates interrupt processing request at rising edge of 8th clock of SCL (master makes SCL output low and waits after outputting 8 clocks. Slave makes SCL pin low and requests for wait after inputting 8 clocks).
	1	1	Used in I^2C bus mode (9-clock wait). Generates interrupt processing request at rising edge of 9th clock of SCL (master makes SCL output low and waits after outputting 9 clocks. Slave makes SCL pin low and requests for wait after inputting 9 clocks).

R/W	WREL	Controls wait release
	0	Wait release status
	1	Releases wait status. After wait status has been released, this bit is automatically cleared to 0 (used to release wait status set by WAT1 and WAT0)

R/W	CLC	Controls clock level[Note 2]
	0	Used in I^2C bus mode. Makes output level of SCL pin low when serial transfer is not executed
	1	Used in I^2C bus mode. Makes output level of SCL pin high impedance when serial transfer is not executed (clock line goes high). Used by master to generate start/stop condition.

Notes 1. Bit 6 (CLD) is a read-only bit.

2. Clear CLC to 0 when the I^2C bus mode is not used.

Figure 8-12. Format of Interrupt Timing Specification Register (μPD78054Y, 78064Y, 78078Y, 780058Y, 780308Y, 78058FY, 78075BY subseries, μPD78070AY) (2/2)

R/W	SVAM	Bits of SVA used as slave address
	0	Bits 0 through 7
	1	Bits 1 through 7

R/W	SIC	Selects INTCSI0 interrupt source[Note 1]
	0	Sets CSIIF0 to 1 at end of transfer of serial interface channel 0
	1	Sets CSIIF0 to 1 at end of transfer of serial interface channel 0 or on detection of stop condition

R/W	CLD	Level of $\overline{SCK0}$/SCL/P27 pin[Note 2]
	0	Low level
	1	High level

Notes 1. Sets SIC to 1 when using the wake-up function in the I^2C mode.

2. CLD is 0 when CSIE0 = 0.

Remark SVA : slave address register
CSIIF0 : interrupt request flag corresponding to INTCSI0
CSIE0 : bit 7 of the serial operating mode register 0 (CSIM0)

Figure 8-13. Format of Serial Operating Mode Register 1
(μPD78054, 78054Y, 78078, 78078Y, 78098, 780018, 780018Y, 780058, 780058Y, 78058F, 78058FY, 78075B, 78075BY, 78098B subseries, μPD78070A, 78070AY)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
CSIM1	CSIE 1	DIR	ATE	0	0	0	CSIM 11	CSIM 10	FF68H	00H	R/W

CSIM 11	CSIM 10	Selects clock of serial interface channel 1
0	×	Clock externally input to $\overline{\text{SCK1}}$ pin**Note 1**
1	0	Output of 8-bit timer register 2 (TM2)
1	1	Clock specified by bits 4 through 7 of timer clock select register 3 (TCL3)

ATE	Selects operation mode of serial interface channel 1
0	3-wire serial I/O mode
1	3-wire serial I/O mode with automatic transfer/reception function

DIR	First bit	Function of SI1 pin	Function of SO1 pin
0	MSB	SI1/P20	SO1
1	LSB	(input)	(CMOS output)

CSIE 1	CSIM 11	PM20	P20	PM21	P21	PM22	P22	Operation of shift register 1	Controls operation of counter of serial clock	Function of SI1/P20 pin	Function of SO1/P21 pin	Function of $\overline{\text{SCK1}}$/P22
0	×	**Note 2** ×	**Note 2** ×	**Note 2** ×	**Note 2** ×	**Note 2** ×	**Note 2** ×	Stops operation	Clear	P20 (CMOS I/O)	P21 (CMOS I/O)	P22 (CMOS I/O)
1	0	**Note 3** 1	**Note 3** ×	0	0	1	×	Enables operation	Count operation	SI1**Note 3** (input)	SO1 (CMOS output)	$\overline{\text{SCK1}}$ (input)
	1					0	1					$\overline{\text{SCK1}}$ (CMOS output)

Notes 1. Clear bit 2 (STRB) and bit 1 (BUSY1) of the automatic data transfer/reception control register (ADTC) to 0, 0 when the external clock input is selected by clearing CSIM11 to 0.

2. These pins can be used as port pins.

3. When only transmit is executed, this pin can be used as P20 (CMOS I/O). (Set bit 7 (RE) of the automatic data transfer/reception control register (ADTC) to 0.)

Remark × : don't care
PM××: Port mode register
P×× : Output latch of port

Figure 8-14. Format of Automatic Data Transfer/Reception Control Register (μPD78054, 78054Y, 78078, 78078Y, 78098, 780018, 780018Y, 78058F, 78058FY, 78075B, 78075BY, 78098B subseries, μPD78070A, 78070AY)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
ADTC	RE	ARLD	ERCE	ERR	TRF	STRB	BUSY1	BUSY0	FF69H	00H	R/W Note 1

R/W

BUSY1	BUSY0	Controls busy input
0	×	Does not use busy input
1	0	Enables busy input (active high)
1	1	Enables busy input (low active)

R/W

STRB	Controls strobe output
0	Disables strobe output
1	Enables strobe output

R

TRF	Status of automatic transfer/reception function Note 2
0	Detects end of automatic transfer/reception (0 when automatic transfer/reception is stopped or when ARLD = 0)
1	Automatic transfer/reception in progress (1 when SIO1 is written)

R

ERR	Detects error of automatic transfer/reception function
0	No error on automatic reception (0 when 1 is written to SIO1)
1	Error on automatic transfer/reception

R/W

ERCE	Controls error check of automatic transfer/reception function
0	Disables error check on automatic transfer/reception
1	Enables error check on automatic transfer/reception (only when BUSY1 = 1)

R/W

ARLD	Selects operation mode of automatic transfer/ reception function
0	Single mode
1	Repetitive mode

R/W

RE	Controls reception of automatic transfer/reception function
0	Disables reception
1	Enables reception

Notes 1. Bits 3 and 4 (TRF and ERR) are read-only bits.

2. Identify the end of automatic transfer/reception by using TRF instead of CSIIF1. (interrupt request flag)

Caution When external clock input is selected by clearing bit 1 (CSIM11) of the serial operating mode register 1 (CSIM1) to 0, clear STRB and BUSY1 of ADTC to 0, 0.

Remark ×: don't care

★ **Figure 8-15. Format of Automatic Data Transfer/Reception Control Register (μPD780058, 780058Y subseries)**

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
ADTC	RE	ARLD	ERCE	ERR	TRF	STRB	BUSY1	BUSY0	FF69H	00H	R/W Note 1

R/W

BUSY1	BUSY0	Controls busy input
0	×	Does not use busy input
1	0	Enables busy input (active high)
1	1	Enables busy input (low active)

R/W

STRB	Controls strobe output
0	Disables strobe output
1	Enables strobe output

R

TRF	Status of automatic transfer/reception function Note 2
0	Detects end of automatic transfer/reception (0 when automatic transfer/reception is stopped or when ARLD = 0)
1	Automatic transfer/reception in progress (1 when SIO1 is written)

R

ERR	Detects error of automatic transfer/reception function
0	No error on automatic reception (0 when 1 is written to SIO1)
1	Error on automatic transfer/reception

R/W

ERCE	Controls error check of automatic transfer/reception function
0	Disables error check on automatic transfer/reception
1	Enables error check on automatic transfer/reception (only when BUSY1 = 1)

R/W

ARLD	Selects operation mode of automatic transfer/ reception function
0	Single mode
1	Repetitive mode

R/W

RE	Controls reception of automatic transfer/reception function
0	Disables reception
1	Enables reception

Notes 1. Bits 3 and 4 (TRF and ERR) are read-only bits.

2. Identify the end of automatic transfer/reception by using TRF instead of CSIIF1. (interrupt request flag)

Cautions 1. When external clock input is selected by clearing bit 1 (CSIM11) of the serial operating mode register 1 (CSIM1) to 0, clear STRB and BUSY1 of ADTC to 0, 0.

2. When using the P23/STB/TxD1 and P24/BUSY/RxD1 pins in the asynchronous serial interface (UART) mode of serial interface channel 2, the busy control option and busy & strobe control option are invalid.

Remark ×: don't care

Figure 8-16. Format of Automatic Data Transfer/Reception Interval Specification Register (μPD78054, 78054Y, 78078, 78078Y, 780018, 780018Y, 780058, 780058Y, 78058F, 78058FY, 78075B, 78075BY subseries, μPD78070A, 78070AY) (1/4)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI7	Controls interval time of data transfer
0	Does not control interval time by ADTI[Note 1]
1	Controls interval time by ADTI (ADTI0 through ADTI4)

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Specifies interval time of data transfer (f_{XX} = 5.0 MHz)	
					Minimum value[Note 2]	Maximum value[Note 2]
0	0	0	0	0	18.4 μs + 0.5/f_{SCK}	20.0 μs + 1.5/f_{SCK}
0	0	0	0	1	31.2 μs + 0.5/f_{SCK}	32.8 μs + 1.5/f_{SCK}
0	0	0	1	0	44.0 μs + 0.5/f_{SCK}	45.6 μs + 1.5/f_{SCK}
0	0	0	1	1	56.8 μs + 0.5/f_{SCK}	58.4 μs + 1.5/f_{SCK}
0	0	1	0	0	69.6 μs + 0.5/f_{SCK}	71.2 μs + 1.5/f_{SCK}
0	0	1	0	1	82.4 μs + 0.5/f_{SCK}	84.0 μs + 1.5/f_{SCK}
0	0	1	1	0	95.2 μs + 0.5/f_{SCK}	96.8 μs + 1.5/f_{SCK}
0	0	1	1	1	108.0 μs + 0.5/f_{SCK}	109.6 μs + 1.5/f_{SCK}
0	1	0	0	0	120.8 μs + 0.5/f_{SCK}	122.4 μs + 1.5/f_{SCK}
0	1	0	0	1	133.6 μs + 0.5/f_{SCK}	135.2 μs + 1.5/f_{SCK}
0	1	0	1	0	146.4 μs + 0.5/f_{SCK}	148.0 μs + 1.5/f_{SCK}
0	1	0	1	1	159.2 μs + 0.5/f_{SCK}	160.8 μs + 1.5/f_{SCK}
0	1	1	0	0	172.0 μs + 0.5/f_{SCK}	173.6 μs + 1.5/f_{SCK}
0	1	1	0	1	184.8 μs + 0.5/f_{SCK}	186.4 μs + 1.5/f_{SCK}
0	1	1	1	0	197.6 μs + 0.5/f_{SCK}	199.2 μs + 1.5/f_{SCK}
0	1	1	1	1	210.4 μs + 0.5/f_{SCK}	212.0 μs + 1.5/f_{SCK}

Notes 1. The interval time is dependent on only the CPU processing.

2. The interval time of data transfer includes an error. The minimum and maximum values of the interval time for data transfer can be calculated by the following expressions (where n is the value set to ADTI0 through ADTI4). However, if the minimum value calculated by the expression below is less than 2/f_{SCK}, the minimum interval time is 2/f_{SCK}.

$$\text{Minimum value} = (n+1) \times \frac{2^6}{f_{XX}} + \frac{28}{f_{XX}} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum value} = (n+1) \times \frac{2^6}{f_{XX}} + \frac{36}{f_{XX}} + \frac{1.5}{f_{SCK}}$$

Cautions 1. Do not write ADTI during automatic transmission/reception operation.

2. Be sure to clear bits 5 and 6 to 0.

★ **3. When controlling interval time of data transfer by automatic transfer/reception using ADTI, the busy control option is invalid.**

Remarks 1. f_{XX} : main system clock frequency (f_X or $f_X/2$)

2. f_X : main system clock oscillation frequency

3. f_{SCK}: serial clock frequency

Figure 8-16. Format of Automatic Data Transfer/Reception Interval Specification Register (μPD78054, 78054Y, 78078, 78078Y, 780018, 780018Y, 780058, 780058Y, 78058F, 78058FY, 78075B, 78075BY subseries, μPD78070A, 78070AY) (2/4)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Specifies interval time of data transfer (f_{XX} = 5.0 MHz)	
					Minimum value[Note]	Maximum value[Note]
1	0	0	0	0	223.2 μs + 0.5/f_{SCK}	224.8 μs + 1.5/f_{SCK}
1	0	0	0	1	236.0 μs + 0.5/f_{SCK}	237.6 μs + 1.5/f_{SCK}
1	0	0	1	0	248.8 μs + 0.5/f_{SCK}	250.4 μs + 1.5/f_{SCK}
1	0	0	1	1	261.6 μs + 0.5/f_{SCK}	263.2 μs + 1.5/f_{SCK}
1	0	1	0	0	274.4 μs + 0.5/f_{SCK}	276.0 μs + 1.5/f_{SCK}
1	0	1	0	1	287.2 μs + 0.5/f_{SCK}	288.8 μs + 1.5/f_{SCK}
1	0	1	1	0	300.0 μs + 0.5/f_{SCK}	301.6 μs + 1.5/f_{SCK}
1	0	1	1	1	312.8 μs + 0.5/f_{SCK}	314.4 μs + 1.5/f_{SCK}
1	1	0	0	0	325.6 μs + 0.5/f_{SCK}	327.2 μs + 1.5/f_{SCK}
1	1	0	0	1	338.4 μs + 0.5/f_{SCK}	340.0 μs + 1.5/f_{SCK}
1	1	0	1	0	351.2 μs + 0.5/f_{SCK}	352.8 μs + 1.5/f_{SCK}
1	1	0	1	1	364.0 μs + 0.5/f_{SCK}	365.6 μs + 1.5/f_{SCK}
1	1	1	0	0	376.8 μs + 0.5/f_{SCK}	378.4 μs + 1.5/f_{SCK}
1	1	1	0	1	389.6 μs + 0.5/f_{SCK}	391.2 μs + 1.5/f_{SCK}
1	1	1	1	0	402.4 μs + 0.5/f_{SCK}	404.0 μs + 1.5/f_{SCK}
1	1	1	1	1	415.2 μs + 0.5/f_{SCK}	416.8 μs + 1.5/f_{SCK}

Note The interval time of data transfer includes an error margin. The minimum and maximum values of the interval time for data transfer can be calculated by the following expressions (where n is the value set to ADTI0 through ADTI4). However, if the minimum value calculated by the expression below is less than 2/f_{SCK}, the minimum interval time is 2/f_{SCK}.

$$\text{Minimum value} = (n+1) \times \frac{2^6}{f_{XX}} + \frac{28}{f_{XX}} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum value} = (n+1) \times \frac{2^6}{f_{XX}} + \frac{36}{f_{XX}} + \frac{1.5}{f_{SCK}}$$

Cautions 1. Do not write ADTI during automatic transfer/reception operation.

2. Be sure to clear bits 5 and 6 to 0.

★ **3. When controlling interval time of data transfer by automatic transfer/reception using ADTI, the busy control option is invalid.**

Remarks 1. f_{XX} : main system clock frequency (f_X or f_X/2)

2. f_X : main system clock oscillation frequency

3. f_{SCK}: serial clock frequency

Figure 8-16. Format of Automatic Data Transfer/Reception Interval Specification Register (μPD78054, 78054Y, 78078, 78078Y, 780018, 780018Y, 780058, 780058Y, 78058F, 78058FY, 78075B, 78075BY subseries, μPD78070A, 78070AY) (3/4)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI7	Controls interval time of data transfer
0	Does not control interval time by ADTI[Note 1]
1	Controls interval time by ADTI (ADTI0 through ADTI4)

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Specifies interval time of data transfer (f_{XX} = 2.5 MHz)	
					Minimum value[Note 2]	Maximum value[Note 2]
0	0	0	0	0	36.8 μs + 0.5/f_{SCK}	40.0 μs + 1.5/f_{SCK}
0	0	0	0	1	62.4 μs + 0.5/f_{SCK}	65.6 μs + 1.5/f_{SCK}
0	0	0	1	0	88.0 μs + 0.5/f_{SCK}	91.2 μs + 1.5/f_{SCK}
0	0	0	1	1	113.6 μs + 0.5/f_{SCK}	116.8 μs + 1.5/f_{SCK}
0	0	1	0	0	139.2 μs + 0.5/f_{SCK}	142.4 μs + 1.5/f_{SCK}
0	0	1	0	1	164.8 μs + 0.5/f_{SCK}	168.0 μs + 1.5/f_{SCK}
0	0	1	1	0	190.4 μs + 0.5/f_{SCK}	193.6 μs + 1.5/f_{SCK}
0	0	1	1	1	216.0 μs + 0.5/f_{SCK}	219.2 μs + 1.5/f_{SCK}
0	1	0	0	0	241.6 μs + 0.5/f_{SCK}	244.8 μs + 1.5/f_{SCK}
0	1	0	0	1	267.2 μs + 0.5/f_{SCK}	270.4 μs + 1.5/f_{SCK}
0	1	0	1	0	292.8 μs + 0.5/f_{SCK}	296.0 μs + 1.5/f_{SCK}
0	1	0	1	1	318.4 μs + 0.5/f_{SCK}	321.6 μs + 1.5/f_{SCK}
0	1	1	0	0	344.0 μs + 0.5/f_{SCK}	347.2 μs + 1.5/f_{SCK}
0	1	1	0	1	369.6 μs + 0.5/f_{SCK}	372.8 μs + 1.5/f_{SCK}
0	1	1	1	0	395.2 μs + 0.5/f_{SCK}	398.4 μs + 1.5/f_{SCK}
0	1	1	1	1	420.8 μs + 0.5/f_{SCK}	424.0 μs + 1.5/f_{SCK}

Notes 1. The interval time is dependent on only the CPU processing.

2. The interval time of data transfer includes an error margin. The minimum and maximum values of the interval time for data transfer can be calculated by the following expressions (where n is the value set to ADTI0 through ADTI4). However, if the minimum value calculated by the expression below is less than 2/f_{SCK}, the minimum interval time is 2/f_{SCK}.

$$\text{Minimum value} = (n+1) \times \frac{2^6}{f_{XX}} + \frac{28}{f_{XX}} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum value} = (n+1) \times \frac{2^6}{f_{XX}} + \frac{36}{f_{XX}} + \frac{1.5}{f_{SCK}}$$

Cautions 1. Do not write ADTI during automatic transfer/reception operation.

2. Be sure to clear bits 5 and 6 to 0.

★ **3. When controlling interval time of data transfer by automatic transfer/reception using ADTI, the busy control option is invalid.**

Remarks 1. f_{XX} : main system clock frequency (f_X or $f_X/2$)

2. f_X : main system clock oscillation frequency

3. f_{SCK}: serial clock frequency

Figure 8-16. Format of Automatic Data Transfer/Reception Interval Specification Register (μPD78054, 78054Y, 78078, 78078Y, 780018, 780018Y, 780058, 780058Y, 78058F, 78058FY, 78075B, 78075BY subseries, μPD78070A, 78070AY) (4/4)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Specifies interval time of data transfer (f_{XX} = 2.5 MHz)	
					Minimum value[Note]	Maximum value[Note]
1	0	0	0	0	446.4 μs + 0.5/f_{SCK}	449.6 μs + 1.5/f_{SCK}
1	0	0	0	1	472.0 μs + 0.5/f_{SCK}	475.2 μs + 1.5/f_{SCK}
1	0	0	1	0	497.6 μs + 0.5/f_{SCK}	500.8 μs + 1.5/f_{SCK}
1	0	0	1	1	523.2 μs + 0.5/f_{SCK}	526.4 μs + 1.5/f_{SCK}
1	0	1	0	0	548.8 μs + 0.5/f_{SCK}	552.0 μs + 1.5/f_{SCK}
1	0	1	0	1	574.4 μs + 0.5/f_{SCK}	577.6 μs + 1.5/f_{SCK}
1	0	1	1	0	600.0 μs + 0.5/f_{SCK}	603.2 μs + 1.5/f_{SCK}
1	0	1	1	1	625.6 μs + 0.5/f_{SCK}	628.8 μs + 1.5/f_{SCK}
1	1	0	0	0	651.2 μs + 0.5/f_{SCK}	654.4 μs + 1.5/f_{SCK}
1	1	0	0	1	676.8 μs + 0.5/f_{SCK}	680.0 μs + 1.5/f_{SCK}
1	1	0	1	0	702.4 μs + 0.5/f_{SCK}	705.6 μs + 1.5/f_{SCK}
1	1	0	1	1	728.0 μs + 0.5/f_{SCK}	731.2 μs + 1.5/f_{SCK}
1	1	1	0	0	753.6 μs + 0.5/f_{SCK}	756.8 μs + 1.5/f_{SCK}
1	1	1	0	1	779.2 μs + 0.5/f_{SCK}	782.4 μs + 1.5/f_{SCK}
1	1	1	1	0	804.8 μs + 0.5/f_{SCK}	808.0 μs + 1.5/f_{SCK}
1	1	1	1	1	830.4 μs + 0.5/f_{SCK}	833.6 μs + 1.5/f_{SCK}

Note The interval time of data transfer includes an error margin. The minimum and maximum values of the interval time for data transfer can be calculated by the following expressions (where n is the value set to ADTI0 through ADTI4). However, if the minimum value calculated by the expression below is less than 2/f_{SCK}, the minimum interval time is 2/f_{SCK}.

$$\text{Minimum value} = (n+1) \times \frac{2^6}{f_{XX}} + \frac{28}{f_{XX}} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum value} = (n+1) \times \frac{2^6}{f_{XX}} + \frac{36}{f_{XX}} + \frac{1.5}{f_{SCK}}$$

Cautions 1. Do not write ADTI during automatic transfer/reception operation.

2. Be sure to clear bits 5 and 6 to 0.

★ **3. When controlling interval time of data transfer by automatic transfer/reception using ADTI, the busy control option is invalid.**

Remarks 1. f_{XX} : main system clock frequency (f_X or f_X/2)

2. f_X : main system clock oscillation frequency

3. f_{SCK}: serial clock frequency

Figure 8-17. Format of Automatic Data Transfer/Reception Interval Specification Register (μPD78098, 78098B subseries) (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI7	Controls interval time of data transfer
0	Does not control interval time by ADTI[Note 1]
1	Controls interval time by ADTI (ADTI0 through ADTI4)

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Specifies interval time of data transfer (f_{XX} = 4.0 MHz)	
					Minimum value[Note 2]	Maximum value[Note 2]
0	0	0	0	0	23.0 μs + 0.5/f_{SCK}	25.0 μs + 1.5/f_{SCK}
0	0	0	0	1	39.0 μs + 0.5/f_{SCK}	41.0 μs + 1.5/f_{SCK}
0	0	0	1	0	55.0 μs + 0.5/f_{SCK}	57.0 μs + 1.5/f_{SCK}
0	0	0	1	1	71.0 μs + 0.5/f_{SCK}	73.0 μs + 1.5/f_{SCK}
0	0	1	0	0	87.0 μs + 0.5/f_{SCK}	89.0 μs + 1.5/f_{SCK}
0	0	1	0	1	103.0 μs + 0.5/f_{SCK}	105.0 μs + 1.5/f_{SCK}
0	0	1	1	0	119.0 μs + 0.5/f_{SCK}	121.6 μs + 1.5/f_{SCK}
0	0	1	1	1	135.0 μs + 0.5/f_{SCK}	137.0 μs + 1.5/f_{SCK}
0	1	0	0	0	151.0 μs + 0.5/f_{SCK}	153.0 μs + 1.5/f_{SCK}
0	1	0	0	1	167.0 μs + 0.5/f_{SCK}	169.0 μs + 1.5/f_{SCK}
0	1	0	1	0	183.0 μs + 0.5/f_{SCK}	185.0 μs + 1.5/f_{SCK}
0	1	0	1	1	199.0 μs + 0.5/f_{SCK}	201.0 μs + 1.5/f_{SCK}
0	1	1	0	0	215.0 μs + 0.5/f_{SCK}	217.0 μs + 1.5/f_{SCK}
0	1	1	0	1	231.0 μs + 0.5/f_{SCK}	233.0 μs + 1.5/f_{SCK}
0	1	1	1	0	247.0 μs + 0.5/f_{SCK}	249.0 μs + 1.5/f_{SCK}
0	1	1	1	1	263.0 μs + 0.5/f_{SCK}	265.0 μs + 1.5/f_{SCK}

Notes 1. The interval time is dependent on only the CPU processing.

2. The interval time of data transfer includes an error margin. The minimum and maximum values of the interval time for data transfer can be calculated by the following expressions (where n is the value set to ADTI0 through ADTI4). However, if the minimum value calculated by the expression below is less than $2/f_{SCK}$, the minimum interval time is $2/f_{SCK}$.

$$\text{Minimum value} = (n+1) \times \frac{2^6}{f_{XX}} + \frac{28}{f_{XX}} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum value} = (n+1) \times \frac{2^6}{f_{XX}} + \frac{36}{f_{XX}} + \frac{1.5}{f_{SCK}}$$

Cautions 1. Do not write ADTI during automatic transfer/reception operation.

2. Be sure to clear bits 5 and 6 to 0.

★ **3. When controlling interval time of data transfer by automatic transfer/reception using ADTI, the busy control option is invalid.**

Remarks 1. f_{XX} : main system clock frequency

2. f_{SCK}: serial clock frequency

Figure 8-17. Format of Automatic Data Transfer/Reception Interval Specification Register (μPD78098, 78098B subseries) (2/2)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Specifies interval time of data transfer (f_{XX} = 4.0 MHz)	
					Minimum value[Note]	Maximum value[Note]
1	0	0	0	0	279.0 μs + 0.5/f_{SCK}	281.0 μs + 1.5/f_{SCK}
1	0	0	0	1	295.0 μs + 0.5/f_{SCK}	297.0 μs + 1.5/f_{SCK}
1	0	0	1	0	311.0 μs + 0.5/f_{SCK}	313.0 μs + 1.5/f_{SCK}
1	0	0	1	1	327.0 μs + 0.5/f_{SCK}	329.0 μs + 1.5/f_{SCK}
1	0	1	0	0	343.0 μs + 0.5/f_{SCK}	345.0 μs + 1.5/f_{SCK}
1	0	1	0	1	359.0 μs + 0.5/f_{SCK}	361.0 μs + 1.5/f_{SCK}
1	0	1	1	0	375.0 μs + 0.5/f_{SCK}	377.0 μs + 1.5/f_{SCK}
1	0	1	1	1	391.0 μs + 0.5/f_{SCK}	393.0 μs + 1.5/f_{SCK}
1	1	0	0	0	407.0 μs + 0.5/f_{SCK}	409.0 μs + 1.5/f_{SCK}
1	1	0	0	1	423.0 μs + 0.5/f_{SCK}	425.0 μs + 1.5/f_{SCK}
1	1	0	1	0	439.0 μs + 0.5/f_{SCK}	441.0 μs + 1.5/f_{SCK}
1	1	0	1	1	455.0 μs + 0.5/f_{SCK}	457.0 μs + 1.5/f_{SCK}
1	1	1	0	0	471.0 μs + 0.5/f_{SCK}	473.0 μs + 1.5/f_{SCK}
1	1	1	0	1	487.0 μs + 0.5/f_{SCK}	489.0 μs + 1.5/f_{SCK}
1	1	1	1	0	503.0 μs + 0.5/f_{SCK}	505.0 μs + 1.5/f_{SCK}
1	1	1	1	1	519.0 μs + 0.5/f_{SCK}	521.0 μs + 1.5/f_{SCK}

Note The interval time of data transfer includes an error margin. The minimum and maximum values of the interval time for data transfer can be calculated by the following expressions (where n is the value set to ADTI0 through ADTI4). However, if the minimum value calculated by the expression below is less than 2/f_{SCK}, the minimum interval time is 2/f_{SCK}.

$$\text{Minimum value} = (n+1) \times \frac{2^6}{f_{XX}} + \frac{28}{f_{XX}} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum value} = (n+1) \times \frac{2^6}{f_{XX}} + \frac{36}{f_{XX}} + \frac{1.5}{f_{SCK}}$$

Cautions 1. Do not write ADTI during automatic transfer/reception operation.

2. Be sure to clear bits 5 and 6 to 0.

★ **3. When controlling interval time of data transfer by automatic transfer/reception using ADTI, the busy control option is invalid.**

Remarks 1. f_{XX} : main system clock frequency

2. f_{SCK}: serial clock frequency

Figure 8-18. Format of Serial Operating Mode Register 2 (μPD78054, 78054Y, 78064, 78064Y, 78078, 78078Y, 78083, 78098, 780058, 780058Y, 780308, 780308Y, 78058F, 78058FY, 78064B, 78075B, 78075BY, 78098B subseries, μPD78070A, 78070AY)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
CSIM2	CSIE2	0	0	0	0	CSIM22	CSCK	0	FF72H	00H	R/W

CSCK	Selects clock in 3-wire serial I/O mode
0	Clock externally input to $\overline{\text{SCK2}}$ pin
1	Output of dedicated baud rate generator

CSIM22	Specifies first bit
0	MSB
1	LSB

CSIE2	Controls operation in 3-wire serial I/O mode
0	Stops operation
1	Enables operation

Cautions 1. Be sure to clear bits 0 and 3 through 6 to 0.

2. Set CSIM2 to 00H in the UART mode.

Figure 8-19. Format of Asynchronous Serial Interface Mode Register (μPD78054, 78054Y, 78064, 78064Y, 78078, 78078Y, 78083, 78098, 780058, 780058Y, 780308, 780308Y, 78058F, 78058FY, 78064B, 78075B, 78075BY, 78098B subseries, μPD78070A, 78070AY)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
ASIM	TXE	RXE	PS1	PS0	CL	SL	ISRM	SCK	FF70H	00H	R/W

SCK	Selects clock of asynchronous serial interface mode
0	Clock externally input to ASCK pin
1	Output of dedicated baud rate generator[Note]

ISRM	Controls receive end interrupt on occurrence of error
0	Generates receive end interrupt request when error occurs
1	Does not generate receive end interrupt request when error occurs

SL	Specifies stop bit length of transmit data
0	1 bit
1	2 bits

CL	Specifies character length
0	7 bits
1	8 bits

PS1	PS0	Specifies parity bit
0	0	No parity
0	1	Always append 0 parity during transmission. Does not check parity during reception (does not generate parity error)
1	0	Odd parity
1	1	Even parity

RXE	Controls reception operation
0	Stops reception operation
1	Enables reception operation

TXE	Controls transmission operation
0	Stops transmission operation
1	Enables transmission operation

Note When the baud rate generator output is selected by setting SCK to 1, the ASCK pin can be used as an I/O port pin.

Cautions 1. Set ASIM to 00H when the 3-wire serial I/O mode is selected.

2. Before changing the operation mode, stop the serial transfer/reception operation.

Table 8-4. Setting of Operation Modes of Serial Interface Channel 2 (μPD78054, 78054Y, 78064, 78064Y, 78078, 78078Y, 78083, 78098, 78058F, 78058FY, 78064B, 78075B, 78075BY, 78098B subseries, μPD78070A, 78070AY)

(1) Operation stop mode

ASIM			CSIM2			PM70	P70	PM71	P71	PM72	P72	First bit	Shift clock	Function of P70/SI2/RxD pin	Function of P71/SO2/TxD pin	Function of P72/$\overline{SCK2}$/ASCK pin
TXE	RXE	SCK	CSIE2	CSIM22	CSCK											
0	0	×	0	×	×	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	–	–	P70	P71	P72
Others												Setting prohibited				

(2) 3-wire serial I/O mode

ASIM			CSIM2			PM70	P70	PM71	P71	PM72	P72	First bit	Shift clock	Function of P70/SI2/RxD pin	Function of P71/SO2/TxD pin	Function of P72/$\overline{SCK2}$/ASCK pin
TXE	RXE	SCK	CSIE2	CSIM22	CSCK											
0	0	0	1	0	0	1Note 2	×Note 2	0	1	1	×	MSB	External clock	SI2Note 2	SO2 (CMOS output)	$\overline{SCK2}$ input
					1					0	1		Internal clock			$\overline{SCK2}$ output
			1	1	0					1	×	LSB	External clock	SI2Note 2	SO2 (CMOS output)	$\overline{SCK2}$ input
					1					0	1		Internal clock			$\overline{SCK2}$ output
Others												Setting prohibited				

(3) Asynchronous serial interface mode

ASIM			CSIM2			PM70	P70	PM71	P71	PM72	P72	First bit	Shift clock	Function of P70/SI2/RxD pin	Function of P71/SO2/TxD pin	Function of P72/$\overline{SCK2}$/ASCK pin
TXE	RXE	SCK	CSIE2	CSIM22	CSCK											
1	0	0	0	0	0	×Note 1	×Note 1	0	1	1	×	LSB	External clock	P70	TxD (CMOS output)	ASCK input
		1								×Note 1	×Note 1		Internal clock			P72
0	1	0	0	0	0	1	×	×Note 1	×Note 1	1	×		External clock	RxD	P71	ASCK input
		1								×Note 1	×Note 1		Internal clock			P72
1	1	0	0	0	0	1	×	0	1	1	×		External clock		TxD (CMOS output)	ASCK input
		1								×Note 1	×Note 1		Internal clock			P72
Others												Setting prohibited				

Notes 1. These pins can be used as port pins.

2. This pin can be used as P70 (CMOS I/O) when only transmission is executed.

Remark × : don't care

PM××: port mode register

P×× : output latch of port

✦ **Table 8-5. Setting of Operation Modes of Serial Interface Channel 2 (μPD780058 and 780058Y Subseries) (1/2)**

(1) Operation stop mode

ASIM			CSIM2			SIPS		PM70	P70	PM71	P71	PM23	P23	PM24	P24	PM72	P72	First bit	Shift clock	Function of P70/SI2/RxD0 pin	Function of P71/SO2/TxD0 pin	Function of P23/STB/TxD1 pin	Function of P24/BUSY/RxD1 pin	Function of P72/$\overline{\text{SCK2}}$/ASCK pin
TXE	RXE	SCK	CSIE2	CSIM22	CSCK	SIPS21	SIPS20																	
0	0	×	0	×	×	×	×	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	–	–	P70	P71	P23/STB	P24/BUSY	P72
Others																		Setting prohibited						

(2) 3-wire serial I/O mode

ASIM			CSIM2			SIPS		PM70	P70	PM71	P71	PM23	P23	PM24	P24	PM72	P72	First bit	Shift clock	Function of P70/SI2/RxD0 pin	Function of P71/SO2/TxD0 pin	Function of P23/STB/TxD1 pin	Function of P24/BUSY/RxD1 pin	Function of P72/$\overline{\text{SCK2}}$/ASCK pin
TXE	RXE	SCK	CSIE2	CSIM22	CSCK	SIPS21	SIPS20																	
0	0	0	1	0	0	×	×	1Note 2	×Note 2	0	1	×Note 1	×Note 1	×Note 1	×Note 1	1	×	MSB	External clock	SI2Note 2	SO2 (CMOS output)	P23/STB	P24/BUSY	$\overline{\text{SCK2}}$ input
					1											0	1		Internal clock					$\overline{\text{SCK2}}$ output
			1	1	0											1	×	LSB	External clock	SI2Note 2	SO2 (CMOS output)			$\overline{\text{SCK2}}$ input
					1											0	1		Internal clock					$\overline{\text{SCK2}}$ output
Others																		Setting prohibited						

Notes 1. These pins can be used as port pins.

2. This pin can be used as P70 (CMOS I/O) when only transmission is executed.

Remark × : don't care

PM××: port mode register

P×× : output latch of port

Table 8-5. Setting of Operation Modes of Serial Interface Channel 2 (μPD780058 and 780058Y Subseries) (2/2)

(3) Asynchronous serial interface mode

ASIM			CSIM2			SIPS		PM70	P70	PM71	P71	PM23	P23	PM24	P24	PM72	P72	First bit	Shift clock	Function of P70/SI2/RxD0 pin	Function of P71/SO2/TxD0 pin	Function of P23/STB/TxD1 pin	Function of P24/BUSY/RxD1 pin	Function of P72/$\overline{SCK2}$/ASCK pin
TXE	RXE	SCK	CSIE2	CSIM22	CSCK	SIPS21	SIPS20																	
1	0	0	0	0	0	0	0	×Note	×Note	0	1	×Note	×Note	×Note	×Note	1	×	LSB	External clock	P70	TxD0 (CMOS output)	P23/STB	P24/BUSY	ASCK input
		1														×Note	×Note		Internal clock					P72
0	1	0	0	0	0	0	0	1	×	×Note	×Note	×Note	×Note	×Note	×Note	1	×		External clock	RxD0	P71			ASCK input
		1														×Note	×Note		Internal clock					P72
1	1	0	0	0	0	0	0	1	×	0	1	×Note	×Note	×Note	×Note	1	×		External clock		TxD0 (CMOS output)			ASCK input
		1														×Note	×Note		Internal clock					P72
1	0	0	0	0	0	1	0	×Note	×Note	0	1	0	1	×Note	×Note	1	×		External clock	P70	Output high	TxD1	P24/BUSY	ASCK input
		1														×Note	×Note		Internal clock					P72
0	1	0	0	0	0	0	1	1	×	×Note	×Note	×Note	×Note	1	×	1	×		External clock	P70 (input)	P71	P23/STB	RxD1	ASCK input
		1														×Note	×Note		Internal clock					P72
1	1	0	0	0	0	1	1	1	×	0	1	0	1	1	×	1	×		External clock	P70 (input)	Output high	TxD1	RxD1	ASCK input
		1														×Note	×Note		Internal clock					P72
Others																		Setting prohibited						

Note These pins can be used as port pins.

Remark × : don't care
PM××: port mode register
P×× : output latch of port

✦ **Table 8-6. Setting of Operation Modes of Serial Interface Channel 2 (μPD780308 and 780308Y Subseries) (1/2)**

(1) Operation stop mode

ASIM			CSIM2			SIPS		PM70	P70	PM71	P71	PM113	P113	PM114	P114	PM72	P72	First bit	Shift clock	Function of P70/SI2/RxD0 pin	Function of P71/SO2/TxD0 pin	Function of P113/TxD pin	Function of P114/RxD pin	Function of P72/$\overline{SCK2}$/ASCK pin
TXE	RXE	SCK	CSIE2	CSIM22	CSCK	SIPS21	SIPS20																	
0	0	×	0	×	×	×	×	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	–	–	P70	P71	P113	P114	P72
Others																		Setting prohibited						

(2) 3-wire serial I/O mode

ASIM			CSIM2			SIPS		PM70	P70	PM71	P71	PM113	P113	PM114	P114	PM72	P72	First bit	Shift clock	Function of P70/SI2/RxD0 pin	Function of P71/SO2/TxD0 pin	Function of P113/TxD pin	Function of P114/RxD pin	Function of P72/$\overline{SCK2}$/ASCK pin
TXE	RXE	SCK	CSIE2	CSIM22	CSCK	SIPS21	SIPS20																	
0	0	0	1	0	0	×	×	1Note 2	×Note 2	0	1	×Note 1	×Note 1	×Note 1	×Note 1	1	×	MSB	External clock	SI2Note 2	SO2 (CMOS output)	P113	P114	$\overline{SCK2}$ input
					1											0	1		Internal clock					$\overline{SCK2}$ output
			1	1	0											1	×	LSB	External clock	SI2Note 2	SO2 (CMOS output)			$\overline{SCK2}$ input
					1											0	1		Internal clock					$\overline{SCK2}$ output
Others																		Setting prohibited						

Notes 1. These pins can be used as port pins.

2. This pin can be used as P70 (CMOS I/O) when only transmission is executed.

Remark × : don't care

PM××: port mode register

P×× : output latch of port

Table 8-6. Setting of Operation Modes of Serial Interface Channel 2 (μPD780308 and 780308Y Subseries) (2/2)

(3) Asynchronous serial interface mode

ASIM			CSIM2			SIPS		PM70	P70	PM71	P71	PM113	P113	PM114	P114	PM72	P72	First bit	Shift clock	Function of P70/SI2/RxD0 pin	Function of P71/SO2/TxD0 pin	Function of P113/TxD pin	Function of P114/RxD pin	Function of P72/$\overline{SCK2}$/ASCK pin
TXE	RXE	SCK	CSIE2	CSIM22	CSCK	SIPS21	SIPS20																	
1	0	0	0	0	0	0	0	×Note	×Note	0	1	×Note	×Note	×Note	×Note	1	×	LSB	External clock	P70	TxD (CMOS output)	P113	P114	ASCK input
		1														×Note	×Note		Internal clock					P72
0	1	0	0	0	0	0	0	1	×	×Note	×Note	×Note	×Note	×Note	×Note	1	×		External clock	RxD	P71			ASCK input
		1														×Note	×Note		Internal clock					P72
1	1	0	0	0	0	0	0	1	×	0	1	×Note	×Note	×Note	×Note	1	×		External clock		TxD (CMOS output)			ASCK input
		1														×Note	×Note		Internal clock					P72
1	0	0	0	0	0	1	0	×Note	×Note	0	1	0	1	×Note	×Note	1	×		External clock	P70	Output high	TxD	P114	ASCK input
		1														×Note	×Note		Internal clock					P72
0	1	0	0	0	0	0	1	1	×	×Note	×Note	×Note	×Note	1	×	1	×		External clock	P70 (input)	P71	P113	RxD	ASCK input
		1														×Note	×Note		Internal clock					P72
1	1	0	0	0	0	1	1	1	×	0	1	0	1	1	×	1	×		External clock	P70 (input)	Output high	TxD	RxD	ASCK input
		1														×Note	×Note		Internal clock					P72
Others																		Setting prohibited						

Note These pins can be used as port pins.

Remark × : don't care
PM××: port mode register
P×× : output latch of port

Figure 8-20. Format of Asynchronous Serial Interface Status Register (μPD78054, 78054Y, 78064, 78064Y, 78078, 78078Y, 78083, 78098, 780058, 780058Y, 780308, 780308Y, 78058F, 78058FY, 78064B, 78075B, 78075BY, 78098B subseries, μPD78070A, 78070AY)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
ASIS	0	0	0	0	0	PE	FE	OVE	FF71H	00H	R

OVE	Overrun error flag
0	Overrun error does not occur
1	Overrun error occurs[Note 1] (if next reception operation is completed before data is read from receive buffer register)

FE	Framing error flag
0	Framing error does not occur
1	Framing error occurs[Note 2] (if stop bit is not detected)

PE	Parity error flag
0	Parity error does not occur
1	Parity error occurs (if parity of transmit data does not coincide)

Notes 1. If an overrun error occurs, be sure to read the receive buffer register (RXB). The overrun error persists each time data is received until RXB is read.

2. Even if the stop bit length is set to 2 bits by the bit 2 (SL) of the asynchronous serial interface mode register (ASIM), only 1 stop bit is detected during reception.

Figure 8-21. Format of Baud Rate Generator Control Register (μPD78054, 78054Y, 78064, 78064Y, 78078, 78078Y, 78083, 780058, 780058Y, 780308, 780308Y, 78058F, 78058FY, 78064B, 78075B, 78075BY subseries, μPD78070A, 78070AY) (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
BRGC	TPS3	TPS2	TPS1	TPS0	MDL3	MDL2	MDL1	MDL0	FF73H	00H	R/W

MDL3	MDL2	MDL1	MDL0	Selects input clock of baud rate generator	k
0	0	0	0	$f_{SCK}/16$	0
0	0	0	1	$f_{SCK}/17$	1
0	0	1	0	$f_{SCK}/18$	2
0	0	1	1	$f_{SCK}/19$	3
0	1	0	0	$f_{SCK}/20$	4
0	1	0	1	$f_{SCK}/21$	5
0	1	1	0	$f_{SCK}/22$	6
0	1	1	1	$f_{SCK}/23$	7
1	0	0	0	$f_{SCK}/24$	8
1	0	0	1	$f_{SCK}/25$	9
1	0	1	0	$f_{SCK}/26$	10
1	0	1	1	$f_{SCK}/27$	11
1	1	0	0	$f_{SCK}/28$	12
1	1	0	1	$f_{SCK}/29$	13
1	1	1	0	$f_{SCK}/30$	14
1	1	1	1	f_{SCK} **Note**	–

Note Can be used only in the 3-wire serial I/O mode.

Remarks 1. f_{SCK}: source clock of 5-bit counter

2. k : value set by MDL0 through MDL3 ($0 \leq k \leq 14$)

Figure 8-21. Format of Baud Rate Generator Control Register (μPD78054, 78054Y, 78064, 78064Y, 78078, 78078Y, 78083, 780058, 780058Y, 780308, 780308Y, 78058F, 78058FY, 78064B, 78075B, 78075BY subseries, μPD78070A, 78070AY) (2/2)

TPS3	TPS2	TPS1	TPS0	Selects source clock of 5-bit counter			n
					MCS = 1	MCS = 0	
0	0	0	0	$f_{XX}/2^{10}$	$f_X/2^{10}$ (4.9 kHz)	$f_X/2^{11}$ (2.4 kHz)	11
0	1	0	1	f_{XX}	f_X (5.0 MHz)	$f_X/2$ (2.5 MHz)	1
0	1	1	0	$f_{XX}/2$	$f_X/2$ (2.5 MHz)	$f_X/2^2$ (1.25 MHz)	2
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$ (1.25 MHz)	$f_X/2^3$ (625 kHz)	3
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (625 kHz)	$f_X/2^4$ (313 kHz)	4
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (313 kHz)	$f_X/2^5$ (156 kHz)	5
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (156 kHz)	$f_X/2^6$ (78.1 kHz)	6
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)	7
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)	8
1	1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.8 kHz)	9
1	1	1	0	$f_{XX}/2^9$	$f_X/2^9$ (9.8 kHz)	$f_X/2^{10}$ (4.9 kHz)	10
Others				Setting prohibited			

Caution If data is written to BRGC during communication, the output of the baud rate generator is disturbed and communication cannot be executed normally.
Therefore, do not write data to BRGC during communication.

Remarks 1. f_{XX} : main system clock frequency (f_X or $f_X/2$)
2. f_X : main system clock oscillation frequency
3. MCS : bit 0 of oscillation mode select register (OSMS)
4. n : value set by TPS0 through TPS3 ($1 \leq n \leq 11$)
5. ( ) : at f_X = 5.0 MHz

Figure 8-22. Format of Baud Rate Generator Control Register (μPD78098, 78098B subseries) (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
BRGC	TPS3	TPS2	TPS1	TPS0	MDL3	MDL2	MDL1	MDL0	FF73H	00H	R/W

MDL3	MDL2	MDL1	MDL0	Selects input clock of baud rate generator	k
0	0	0	0	$f_{SCK}/16$	0
0	0	0	1	$f_{SCK}/17$	1
0	0	1	0	$f_{SCK}/18$	2
0	0	1	1	$f_{SCK}/19$	3
0	1	0	0	$f_{SCK}/20$	4
0	1	0	1	$f_{SCK}/21$	5
0	1	1	0	$f_{SCK}/22$	6
0	1	1	1	$f_{SCK}/23$	7
1	0	0	0	$f_{SCK}/24$	8
1	0	0	1	$f_{SCK}/25$	9
1	0	1	0	$f_{SCK}/26$	10
1	0	1	1	$f_{SCK}/27$	11
1	1	0	0	$f_{SCK}/28$	12
1	1	0	1	$f_{SCK}/29$	13
1	1	1	0	$f_{SCK}/30$	14
1	1	1	1	f_{SCK} **Note**	–

Note Can be used only in the 3-wire serial I/O mode.

Remarks 1. f_{SCK} : source clock of 5-bit counter

2. k : value set by MDL0 through MDL3 ($0 \leq k \leq 14$)

Figure 8-22. Format of Baud Rate Generator Control Register (μPD78098, 78098B subseries) (2/2)

TPS3	TPS2	TPS1	TPS0	Selects source clock of 5-bit counter	n
0	0	0	0	$f_{XX}/2^{10}$ (3.91 kHz)	11
0	1	0	1	f_{XX} (4.0 MHz)	1
0	1	1	0	$f_{XX}/2$ (2.0 MHz)	2
0	1	1	1	$f_{XX}/2^2$ (1.0 MHz)	3
1	0	0	0	$f_{XX}/2^3$ (500 kHz)	4
1	0	0	1	$f_{XX}/2^4$ (250 kHz)	5
1	0	1	0	$f_{XX}/2^5$ (125 kHz)	6
1	0	1	1	$f_{XX}/2^6$ (62.5 kHz)	7
1	1	0	0	$f_{XX}/2^7$ (31.3 kHz)	8
1	1	0	1	$f_{XX}/2^8$ (15.6 kHz)	9
1	1	1	0	$f_{XX}/2^9$ (7.81 kHz)	10
Others				Setting prohibited	

Caution If data is written to BRGC during communication, the output of the baud rate generator is disturbed and communication cannot be executed normally.
Therefore, do not write data to BRGC during communication.

Remarks 1. f_{XX} : main system clock frequency
2. n : value set by TPS0 through TPS3 ($1 \leq n \leq 11$)
3. ( ) : at f_{XX} = 4.0 MHz

★ **Figure 8-23. Format of Serial Interface Pin Select Register (μPD780058 and 780058Y Subseries)**

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
SIPS	0	0	SIPS21	SIPS20	0	0	0	0	FF75H	00H	R/W

SIPS21	SIPS20	Selects I/O pin of asynchronous serial interface
0	0	Input pin : RxD0/SI2/P70 Output pin : TxD0/SO2/P71
0	1	Input pin : RxD1/BUSY/P24 Output pin : TxD0/SO2/P71
1	0	Input pin : RxD0/SI2/P70 Output pin : TxD1/STB/P23
1	1	Input pin : RxD1/BUSY/P24 Output pin : TxD1/STB/P23

Cautions 1. Change the mode of an I/O Pin after stopping the serial transfer/reception operation.

2. When using the busy control option or busy & strobe control option in the 3-wire serial I/O mode with automatic transfer/reception function of the serial interface channel 1, the RxD1/BUSY/P24 and TxD1/STB/P23 pins cannot be used as data I/O pins.

★ **Figure 8-24. Format of Serial Interface Pin Select Register (μPD780308 and 780308Y Subseries)**

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
SIPS	0	0	SIPS21	SIPS20	0	0	0	0	FF75H	00H	R/W

SIPS21	SIPS20	Selects I/O pin of asynchronous serial interface
0	0	Input pin : RxD/SI2/P70 Output pin : TxD/SO2/P71
0	1	Input pin : RxD/P114 Output pin : TxD/SO2/P71
1	0	Input pin : RxD0/SI2/P70 Output pin : TxD/P113
1	1	Input pin : RxD/P114 Output pin : TxD/P113

Cautions 1. Change the mode of an I/O Pin after stopping the serial transfer/reception operation.

2. Port 11 has a falling edge detection function. Do not input a falling edge to the pin used as a multiplexed pin of this port.

8.1 Interface with EEPROM™ (μPD6252)

The μPD6252[Note] is a 2048-bit EEPROM which can be electrically written or erased. To write or read data to or from the μPD6252, the 3-wire serial interface is used.

★ **Note** μPD6252 is for maintenance use.

Figure 8-25. Pin Configuration of μPD6252

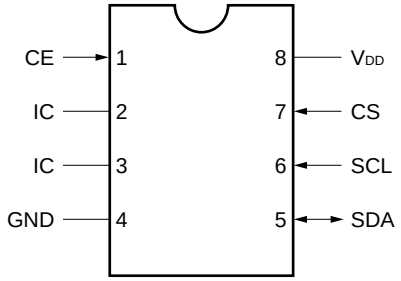
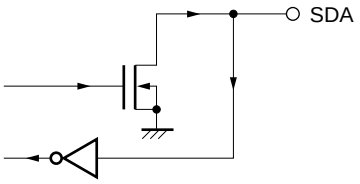


Table 8-7. Pin Function of μPD6252

Pin Number	Pin Name	I/O	Function
1	CE	CMOS input	Keep this pin high during data transfer. **Caution Do not change the level of this pin from high to low during data transfer.** To change the level of this pin from high to low, make sure that the CS pin (pin 7) is low. By making both the CE and CS pins low, you can set the standby status in which the power consumption is reduced.
2 3	IC	–	Fix the IC pins to the high or low level via resistor.
4	GND	–	Ground
5	SDA	CMOS input/ N-ch open-drain output	Data input/output pin. Because this pin is an N-ch open-drain I/O pin, externally pull it up with a resistor. 
6	SCL	CMOS input	Inputs a clock for data transfer.
7	CS	CMOS input	Chip select pin. When this pin is high, the μPD6252 is enabled to operate. When it is low, memory cells cannot be read or written. When the level of this pin is changed from high to low with the SCL pin high, the operation of the serial bus interface is started. To end the operation of the serial bus interface, change the level of this pin from high to low.
8	V_{DD}	–	Positive power: +5 V ±10%

8.1.1 Communication in 2-wire serial I/O mode

The 3-wire mode of the μPD6252[Note] is implemented by serial clock (SCL), data (SDA), and chip select (CS) lines. Excluding the handshaking line, therefore, only two lines, clock and data lines, are necessary for interfacing. To interface the μPD6252 with a 78K/0 series microcontroller, the 2-wire serial I/O mode is used. In the example shown in this section, the μPD78054 subseries is used.

★ **Note** μPD6252 is for maintenance use.

Figure 8-26. Example of Connection of μPD6252

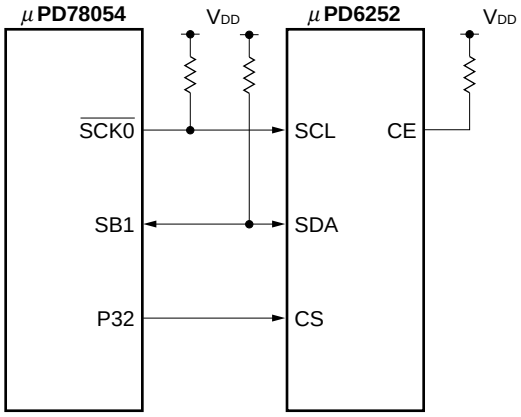


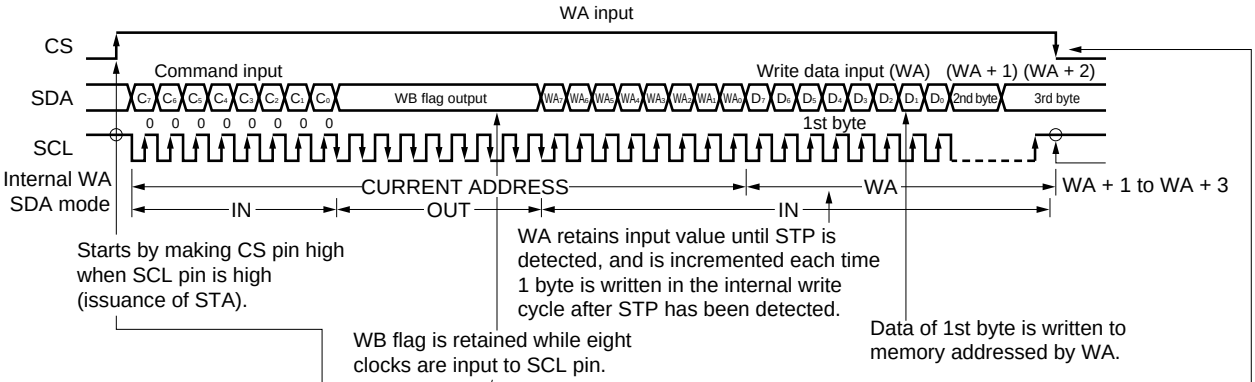
Table 8-8 and Figure 8-27 shows the commands to write and read data to/from the μPD6252 and communication format.

Table 8-8. μPD6252 Commands

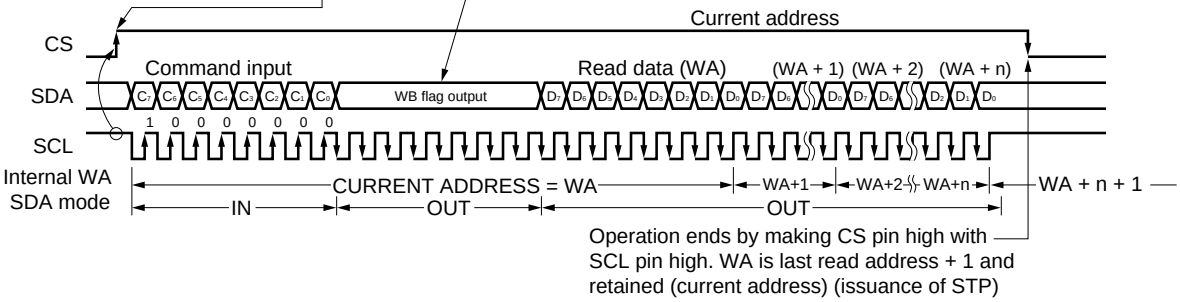
Command Name	Command	Operation
RANDOM WRITE	00000000B [00H] MSB C_7-C_0	Transfers write data after setting an 8-bit word address (WA). Up to 3 bytes of write data can be set successively. Correspondence between word address and data WA Data of first byte WA+1 Data of second byte WA+2 Data of third byte The write operation is executed in the internal write cycle after the CS pin has gone low.
CURRENT READ	10000000B [80H] MSB C_7-C_0	Transfers the contents of memory specified by the word address (WA) (current address) specified when the command is set, to the read data buffer. Each time 8 bits of data have been read from the SDA pin, the word address (WA) is incremented, and the corresponding memory contents are transferred to the data buffer.
RANDOM READ	11000000B [C0H] MSB C_7-C_0	Executes data read starting from a set word address (WA) after the word address has been set. The difference from CURRENT READ is that this command sets a word address (WA) after it has been executed. After the word address has been set, this command performs the same operation as CURRENT READ.

Figure 8-27. Communication Format of μPD6252

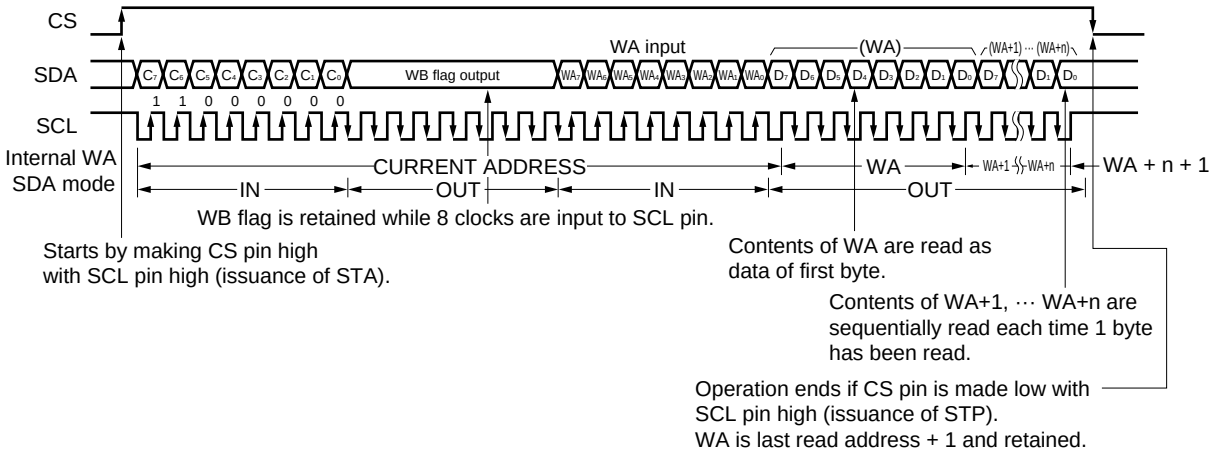
(1) RANDOM WRITE



(2) CURRENT READ



(3) RANDOM READ



Steps <1> through <5> below are the operating procedure of the μPD6252. In this example, the number of data to be written or read per interface operation is fixed to 1 byte. If the μPD6252 is in the write busy (WB) status when interfaced, the busy flag is set.

<1> Make the CS pin (P32) high to start interfacing.
<2> Transmit the write or read command.
<3> Receive the data of WRITE BUSY. If interfacing the μPD6252 is enabled, 00H is received. If a code other than 00H is received, it is judged that the μPD6252 is in the WRITE BUSY status. In this case, communication is stopped.
<4> Transfer the data corresponding to the command.
<5> Make the CS pin (P32) low to end the communication.

(1) Description of package

<Public declaration symbol>

T3_6252 : μPD6252 transfer subroutine name
RWRITE : RANDOM WRITE command value
RREAD : RANDOM READ command value
CREAD : CURRENT READ command value
WADAT : Word address storage area
TRNDAT : Transmit data storage area
RCVDAT : Receive data storage area
CMDDAT : Command data storage area
BUSYFG : Busy status test flag
CS6252 : CS pin (P32) of μPD6252

<Register used>

A

<RAM used>

Name	Usage	Attribute	Bytes
WAADR	Stores word address (before start of transfer)	SADDR	1
TRNDAT	Stores transmit data (before start of transfer)		
RCVDAT	Stores receive data (after end of transfer)		
CMDDAT	Stores command data (before start of transfer)		

<Flag used>

Name	Usage
BUSYFG	Sets WRITE BUSY status

<Nesting>

1 level 3 bytes

<Hardware used>

- Serial interface channel 0
- P32

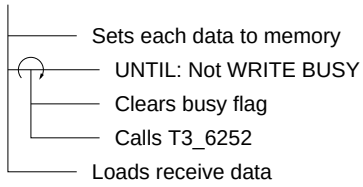
<Initial setting>

- OSMS = #00000001B ; Oscillation mode select register: does not use divider circuit
- Setting of serial interface channel 0
  CSIM0 = #10011011B ; Selects 2-wire serial I/O mode and SB1 pin
- TCL3 = #××××1001B ; Serial clock $f_{XX}/2^4$
- RELT = 1 ; Makes SB1 latch high

<Starting>

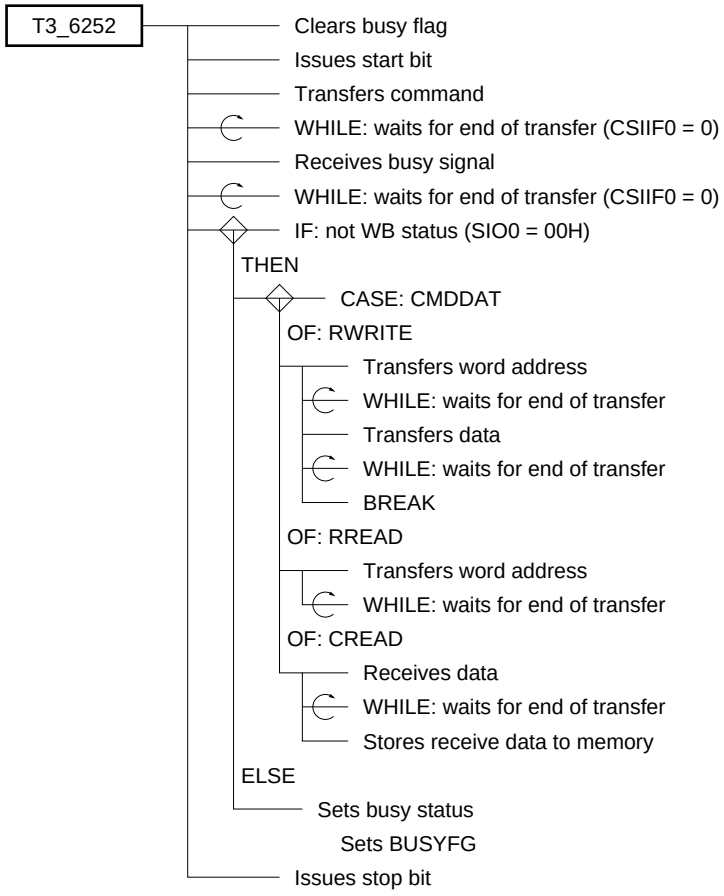
Set the necessary data corresponding to the commands and call T3_6252. After execution returns from the subroutine, the busy flag (BUSYFG) is tested. If the busy flag is set, transfer is not executed. It is therefore necessary to execute transfer again. In the receive mode, the receive data is stored RCVDAT after execution has returned from the subroutine.

(2) Example of use




```
EXTRN   RWRITE,RREAD,CREAD
EXTRN   WADAT,TRNDAT,RCVDAT,CMDDAT,T3_6252
EXTBIT  BUSYFG,CS6252

    OSMS=#00000001B                     ; Does not use divider circuit
    CSIM0=#10011011B                    ; Sets 2-wire serial I/O mode and SB1 pin
    TCL3=#10011001B                     ; Sets SCK0 = 262 kHz
    CLR1    SB0
    CLR1    CS6252                      ; Makes CS of μPD6252 low
    CLR1    PM3.2

    CMDDAT=A
      .
      .
      .
    WADAT=A
      .
      .
      .
    TRNDAT=A
      .
      .
      .
    repeat
            CLR1    BUSYFG
            CALL    !T3_6252
    until_bit(!BUSYFG)
      .
      .
      .
    A=RCVDAT
```

(3) SPD chart


T3_6252
Clears busy flag
Issues start bit
Transfers command
WHILE: waits for end of transfer (CSIIF0 = 0)
Receives busy signal
WHILE: waits for end of transfer (CSIIF0 = 0)
IF: not WB status (SIO0 = 00H)
THEN
CASE: CMDDAT
OF: RWRITE
Transfers word address
WHILE: waits for end of transfer
Transfers data
WHILE: waits for end of transfer
BREAK
OF: RREAD
Transfers word address
WHILE: waits for end of transfer
OF: CREAD
Receives data
WHILE: waits for end of transfer
Stores receive data to memory
ELSE
Sets busy status
Sets BUSYFG
Issues stop bit

(4) Program list

```
        PUBLIC  RWRITE,RREAD,CREAD
        PUBLIC  WADAT,TRNDAT,RCVDAT,CMDDAT,T3_6252
        PUBLIC  BUSYFG,CS6252
CSI_DAT DSEG    SADDR
WADAT:  DS      1                       ; Word address storage area
TRNDAT: DS      1                       ; Transmit data storage area
RCVDAT: DS      1                       ; Receive data storage area
CMDDAT: DS      1                       ; Command data storage area

CSI_FLG BSEG
BUSYFG  DBIT                            ; Sets busy status

RWRITE  EQU     00H                     ; RANDOM WRITE mode
RREAD   EQU     0C0H                    ; RANDOM READ mode
CREAD   EQU     080H                    ; CURRENT READ mode
CS6252  EQU     0FF03H.2                ; 0FF03H=PORT3

CSI_SEG CSEG
;************************************
;*   μPD6252 (3-wire) communication
;************************************
T3_6252:
        CLR1    BUSYFG
        SET1    CS6252                  ; Issues start bit
        SI00=CMDDAT (A)                 ; Transfers command
        while_bit(!CSIIF0)              ; Waits for end of transfer
        endw
        CLR1    CSIIF0
        SIO0=#0FFH                      ; Starts reception of busy signal
        while_bit(!CSIIF0)              ; Waits for end of transfer
        endw
        CLR1    CSIIF0
        if(SIO0==#00H)                  ; Busy check
            switch (CMDDAT)
                case RWRITE:
                    SIO0=WADAT (A)      ; Transfers word address
                    while_bit(!CSIIF0)  ; Waits for end of transfer
                    endw
                    CLR1    CSIIF0
                    SIO0=TRNDAT (A)     ; Starts data transfer
                    while_bit(!CSIIF0)  ; Waits for end of transfer
                    endw
                    CLR1    CSIIF0
                break

                case RREAD:
                    SIO0=WADAT (A)      ; Transfers word address
                    while_bit(!CSIIF0)  ; Waits for end of transfer
                    endw
                    CLR1    CSIIF0
                case CREAD:
                    SIO0=#0FFH          ; Starts data reception
                    while_bit(!CSIIF0)  ; Waits for end of transfer
                    endw
                    CLR1    CSIIF0
                    RCVDAT=SIO0 (A)     ; Stores receive data
            ends
```

```
else
    SET1    BUSYFG          ; Sets busy status
endif
CLR1    CS6252
RET
```

8.1.2 Communication in I^2C bus mode

In the 2-wire mode of the μPD6252[Note], two lines, serial clock (SCL) and data (SDA) lines are used for communication. This mode conforms to the communication format of I^2C. Therefore, the I^2C mode is selected when communicating with the μPD6252 by using the μPD78054Y, 78064Y, 78078Y, 780058Y, 780308Y, 78058FY, 78075BY subseries, or μPD78070AY.

In the example shown in this section, the μPD78054Y subseries is used.

★ **Note** μPD6252 is for maintenance use.

Figure 8-28. Example of Connection between μPD6252 and I^2C Bus Mode

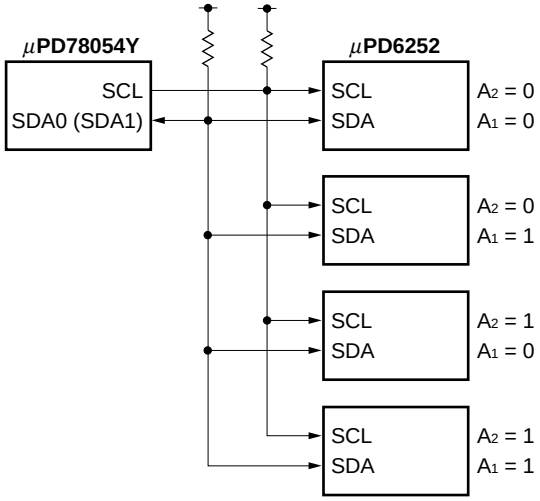
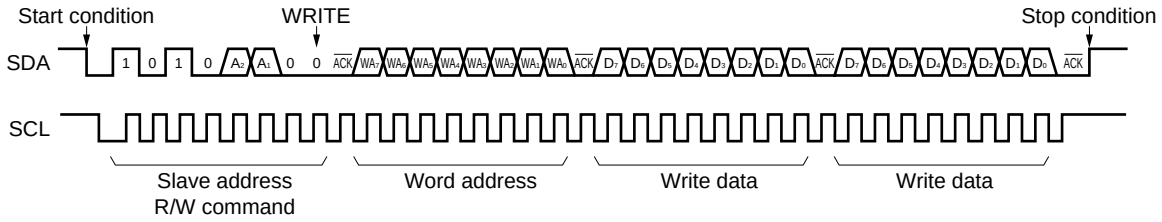


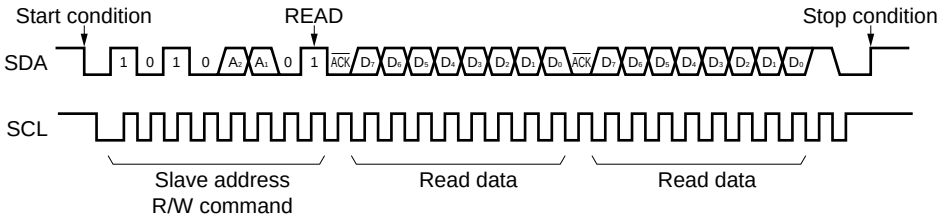
Figure 8-29 shows the communication format in which data is written to or read from the μPD6252.

Figure 8-29. μPD6252 Operation Timing

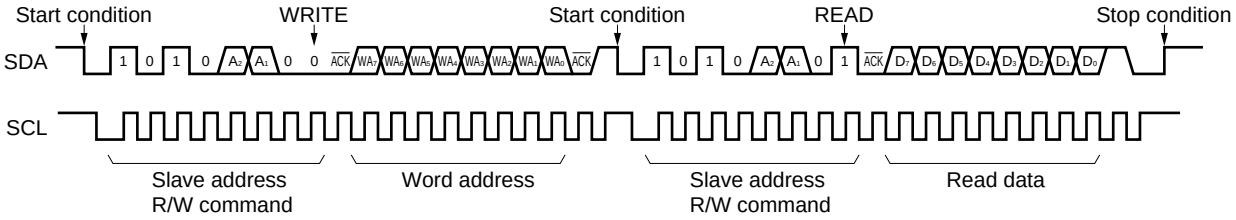
(a) Transmission to μPD6252



(b) Reception from μPD6252 (without word address specification)



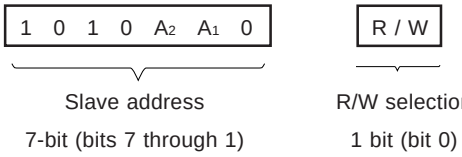
(c) Reception from μPD6252 (with word address specification)



Steps <1> through <5> below are the communication procedure of the μPD6252. In this example, the number of data to be written or read is fixed to 1 byte. If the master receives data in the I^2C bus format, and if it has received the last data, the $\overline{ACK}$ signal is not output. Because the master does not output the $\overline{ACK}$ signal in this example, ACKE is always 0.

<1> Set a start condition to start communication.
Fall the data with the serial clock high.

<2> Transmit the slave address value (bits 1 through 7) of the μPD6252 and write (bit 0 = 0)/read (bit 0 = 1) select bit.



Remark A_2 and A_1 are set by external pins.

<3> Transfer the data.

- In transmission mode
  (i) Transmit the word address of the μPD6252.
  (ii) Transmit the write data.

- In reception mode
  Receive the read data.

<4> Set an end condition to end the communication.
Rise the data with the serial clock high.

<5> Because a word address is specified only in the write mode, to read data by specifying an address, the address must be specified by once setting the write mode.

If the μPD6252 does not return the $\overline{ACK}$ signal during data transfer, communication is stopped.

The start and end conditions are set by CLC when the serial clock is manipulated, and by RELT and CMDT when data is manipulated.

(1) Description of package

<Public declaration symbol>

T2_6252 : μPD6252 transfer subroutine name
WAADR : Word address storage area
TRNDAT : Transmit data storage area
RCVDAT : Receive data storage area
SLVADR : Slave address storage area
BUSYFG : Busy status test flag
WRCHG : Write → read mode change flag
ERRFG : Error status test flag

<Register used>

A

<RAM used>

Name	Usage	Attribute	Bytes
WAADR	Stores word address (before start of transfer)	SADDR	1
TRNDAT	Stores transmit data (before start of transfer)		
RCVDAT	Stores receive data (after end of transfer)		
SLVADR	Stores slave address		

<Flag used>

Name	Usage
BUSYFG	Sets WRITE BUSY status
WRCHG	Changes write mode to read mode
ERRFG	Sets error status

<Nesting>

1 level 2 bytes

<Hardware used>

Serial interface 0

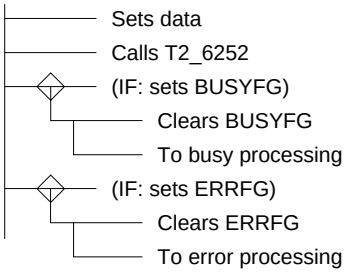
<Initial setting>

- OSMS = #00000001B ; Oscillation mode select register: does not use divider circuit
- Setting of serial interface channel 0
  CSIM0 = #10011011B ; Selects 2-wire serial I/O mode and SB0 pin
- TCL3 = #××××1000B ; Selects serial clock $f_{XX}/2^3$ and 16
- SINT = #00001011B ; Generates interrupts at rising edge of 9th serial clock and sets clock line to high level

<Starting>

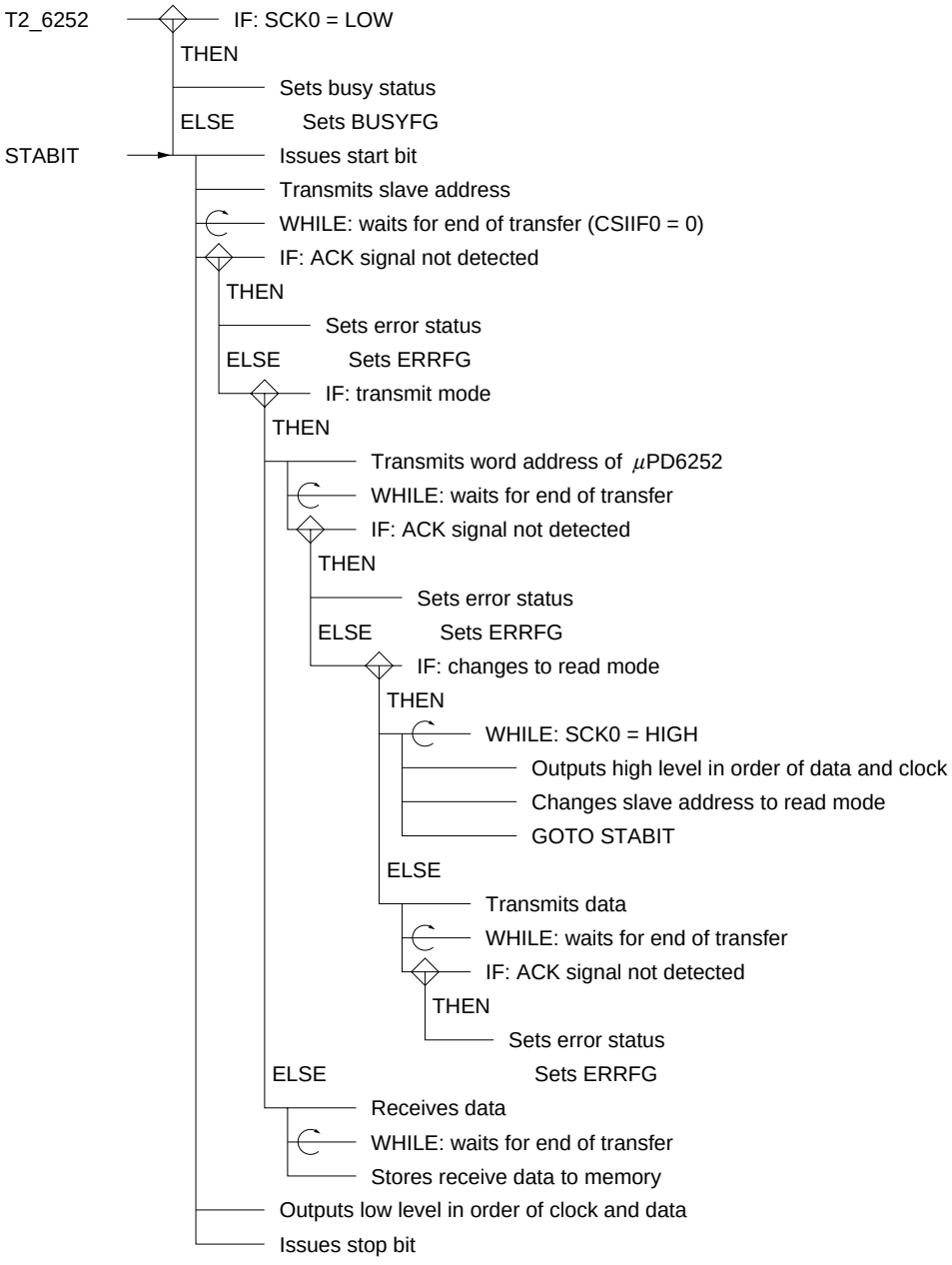
- Set the necessary data corresponding to the commands and call T2_6252. In the reception mode, the receive data is stored to RCVDAT after execution has returned from the subroutine.
- If the serial clock is low (busy status) when communication is started or if ACK cannot be received during data transfer, the BUSYFG and ERRFG are set. Test and clear these flags with the main processing.

(2) Example of use




```
EXTRN   WAADR,TRNDAT,RCVDAT,SLVADR,T2_6252
EXTBIT  BUSYFG,WRCHG,ERRFG

SET1    SB0
OSMS=#00000001B          ; Does not use divider circuit
CSIM0=#10011011B         ; Serial interface 2-wire, SB0
SINT=#00001011B          ; Sets I2C mode
TCL3=#10001000B          ; SCK = 32.7 kHz
SET1    RELT
SET1    SCK0
CLR1    SB0
  ⋮
WAADR=A
  ⋮
TRNDAT=A

SLVADR=A
CALL    !T2_6252
if_bit(BUSYFG)
        CLR1    BUSYFG
    ⋮
endif
    ⋮
if|bit(ERRFG)
        CLR1    ERRFG
    ⋮
ENDIF
```

(3) SPD chart


T2_6252
IF: SCK0 = LOW
THEN
Sets busy status
ELSE Sets BUSYFG
STABIT
Issues start bit
Transmits slave address
WHILE: waits for end of transfer (CSIIF0 = 0)
IF: ACK signal not detected
THEN
Sets error status
ELSE Sets ERRFG
IF: transmit mode
THEN
Transmits word address of μPD6252
WHILE: waits for end of transfer
IF: ACK signal not detected
THEN
Sets error status
ELSE Sets ERRFG
IF: changes to read mode
THEN
WHILE: SCK0 = HIGH
Outputs high level in order of data and clock
Changes slave address to read mode
GOTO STABIT
ELSE
Transmits data
WHILE: waits for end of transfer
IF: ACK signal not detected
THEN
Sets error status
ELSE Sets ERRFG
Receives data
WHILE: waits for end of transfer
Stores receive data to memory
Outputs low level in order of clock and data
Issues stop bit

(4) Program list

```
        PUBLIC  WAADR,TRNDAT,RCVDAT,SLVADR,T2_6252
        PUBLIC  BUSYFG,WRCHG,ERRFG

CSI_DAT DSEG    SADDR
WAADR:  DS      1                       ; Word address storage area
TRNDAT: DS      1                       ; Transmit data storage area
RCVDAT: DS      1                       ; Receive data storage area
SLVADR: DS      1                       ; Salve address storage area

CSI_FLG BSEG
BUSYFG  DBIT                            ; Sets busy status
WRCHG   DBIT                            ; Changes mode
ERRFG   DBIT                            ; Sets error status

SCK0    EQU     P2.7

CSI_SEG CSEG
;************************************
;*   μPD6252 (2-wire) communication
;************************************
T2_6252:
        if_bit(!CLD)
            SET1    BUSYFG              ; Busy status
        else
STABIT:
            SET1    CMDT                ; Issues start bit
            NOP                         ; Waits for start bit valid width
            NOP
            NOP
            NOP
            NOP
            CLR1    CLC                 ; Changes clock to low level
            SIO0=SLVADR (A)             ; Starts transmitting slave address
            while_bit(!CSIIF0)          ; Waits for end of transfer
            endw
            CLR1    CSIIF0
            if_bit(!ACKD)               ; ACK signal not detected
                SET1    ERRFG
            elseif_bit(!SLVADR.0)       ; Transmission mode
                SI00=WAADR (A)          ; Starts transmitting word address
                while_bit(!CSIIF0)      ; Waits for end of transfer
                endw
                CLR1    CSIIF0
                if_bit(!ACKD)           ; ACK signal not detected
                    SET1    ERRFG
                elseif_bit(WRCHG)
                    while_bit(CLD)
                    endw
                    SET1    RELT
                    SET1    CLC
                    while_bit(!CLD)     ; Checks high level of clock
                    endw
                    NOP                 ; Waits for high level valid width of clock
                    NOP
                    NOP
                    NOP
                    NOP
                    NOP
                    NOP
                    NOP
                    SET1    SLVADR.0    ; Changes to read mode address
                    goto    STABIT

                else
```

```
            SIO0=TRNDAT (A)         ; Starts transmitting data
            while_bit(!CSIIF0)      ; Waits for end of transfer
            endw
            CLR1    CSIIF0
            if_bit(!ACKD)           ; ACK signal detected
                SET1    ERRFG
            endif
        endif
    else
        SIO0=#0FFH                  ; Starts data reception
        while_bit(!CSIIF0)          ; Waits for end of transfer
        endw
        CLR1    CSIIF0
        RCVDAT=SIO0 (A)             ; Stores receive data
    endif

    while_bit(CLD)
    endw
    SET1    CMDT
    SET1    CLC
    while_bit(!CLD)                 ; Checks high level of clock
    endw
    NOP                             ; Waits for high level valid width of clock
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    SET1    RELT                    ; Issues stop bit
endif
RET
```

★ **(5) Limitation when using I^2C bus mode**

The following limitation applies when the μPD78054Y, 78064Y, 78078Y, 780058Y, 780308Y, 78058FY, 78075BY subseries, and μPD78070AY are used. This section explains an example using the μPD78054Y.

- **Limitation when the device is used as a slave device in the I^2C bus mode**

Description: If the wake-up function is executed (by setting the WUP flag (bit 5 of serial operation mode register 0 (CSIM0) to 1) in the serial transfer status[Note], the data between other slave device and the master devices is checked as an address. If that data coincides with the slave address of the μPD78054Y, therefore, the μPD78054Y takes part in communication, destroying the communication data.

Note The serial transfer status is the status from when the serial I/O shift register 0 (SIO0) has been written until the interrupt request flag (CSIIF0) is set to 1 by completion of serial transfer.

Preventive measures: The above problem can be prevented by modifying the program.
Before executing the wake-up function, execute the following program that clears the serial transfer status. When executing the wake-up function, do not execute an instruction that writes data to SIO0. Even if such an instruction is executed, data can be received when the wake-up function is executed.
This program is to clear the serial transfer status. To clear the serial transfer status, serial interface channel 0 must be stopped (by clearing the CSIE0 flag (bit 7 of the serial operation mode register (CSIM0) to 0). If the serial interface channel 0 is stopped in the I^2C bus mode, however, the SCL pin outputs a high level and the SDA0 (SDA1) pin outputs a low level, affecting communication on the I^2C bus. Therefore, this program allows the SCL and SDA0 (SDA1) pin to go into a high-impedance state to prevent the I^2C bus from being affected.
Note that, in this example, the serial data input/output pin is SDA0 (/P25). If SDA1 (/P26) is used as the serial data input/output pin, take P2.5 and PM2.5 in the program as P2.6 and PM2.6.

- **Example of program that clears serial transfer status**

```
SET1    P2.5     ; <1>
SET1    PM2.5    ; <2>
SET1    PM2.7    ; <3>
CLR1    CSIE0    ; <4>
SET1    CSIE0    ; <5>
SET1    RELT     ; <6>
CLR1    PM2.7    ; <7>
CLR1    P2.5     ; <8>
CLR1    PM2.5    ; <9>
```

<1> When the I^2C bus mode is restored by instruction <5>, the SDA0 pin does not output a low level. The output of the SDA0 pin goes into a high-impedance state.

<2> The P25(/SDA0) pin is set in the input mode to prevent the SDA0 line from being affected when the port mode is set by instruction <4>. The P25 pin is set in the input mode when instruction <2> is executed.

<3> The P27 (/SCL) pin is set in the input mode to prevent the SCL line from being affected when the port mode is set by instruction <4>. The P27 pin is set in the input mode when instruction <3> is executed.

<4> The I^2C bus mode is changed to the port mode.

<5> The port mode is changed to the I^2C bus mode.

<6> Instruction <8> prevents the SDA0 pin from outputting a low level.

<7> The P27 pin is set in the output mode because it must be in the output mode in the I^2C bus mode.

<8> The output latch of the P25 pin is cleared to 0 because it must be cleared to 0 in the I^2C bus mode.

<9> The P25 pin is set in the output mode because it must be in the output mode in the I^2C bus mode.

Remark RELT: Bit 0 of serial bus interface control register (SBIC)

8.2 Interface with OSD LSI (μPD6451A)

The OSD (On Screen Display) LSI μPD6451A displays the program information of a VCR and TV channels on a display when used in combination with a microcontroller. The μPD6451A is interface with four lines: DATA, CLK, STB, and BUSY. In the example shown in this section, the μPD78054 subseries is used to interface the μPD6451A.

Figure 8-30. Example of Connecting μPD6451A

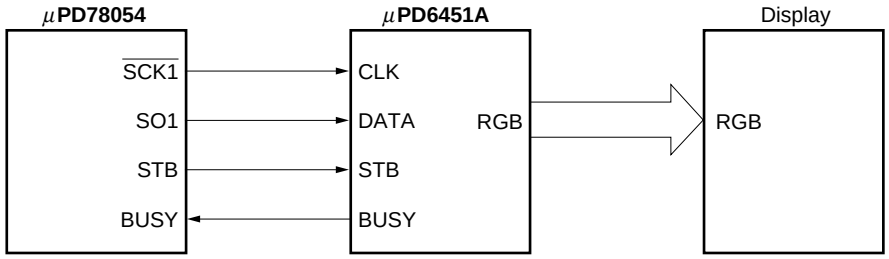
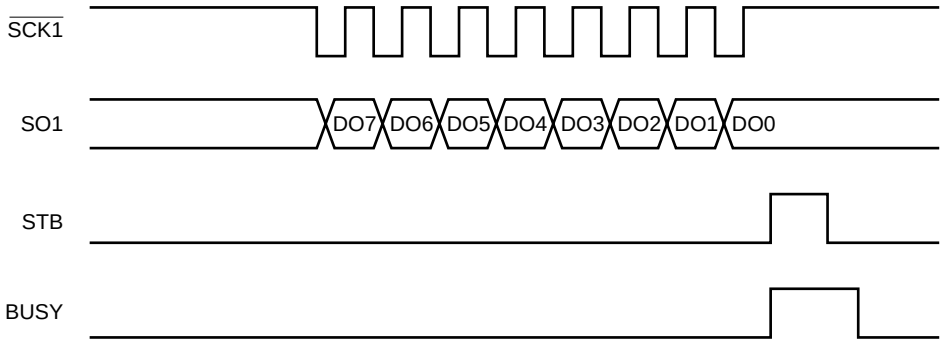


Figure 8-31. Communication Format of μPD6451A



The strobe signal (STB) is output and busy signal (BUSY) is tested automatically by the serial interface channel 1 of the 78K/0 series to establish handshaking with and to interface the μPD6451A. To match the communication format of the μPD6451A, the μPD78054 subseries is set in a mode in which output of the strobe signal and input of the busy signal (high active) are enabled. By setting the transmit data (32 bytes MAX) in a buffer area (FAC0H through FADFH) and the number of transmit data to the automatic data transmit/receive address pointer (ADTP), you can automatically transmit plural data successively.

(1) Description of package

<Public declaration symbol>

TR6451 : μPD6451A transfer subroutine name

DTVAL : Number of transmit data setting area

<Register used>

A

<RAM used>

Name	Usage	Attribute	Bytes
DTVAL	Stores number of transmit data	SADDR	1

<Nesting>

1 level 2 bytes

<Hardware used>

- Serial interface channel 1

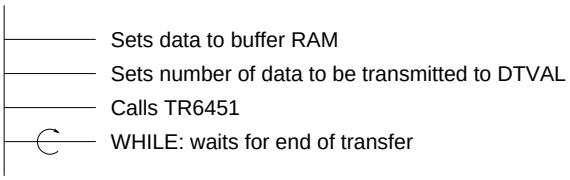
<Initial setting>

- Setting of serial interface channel 1
  CSIM1 = #10100011B ; Enables automatic transmission/reception with MSB first
  ADTC = #00000110B ; Enables busy input (high active) and strobe output in single mode
- ADTI = #00000000B ; Interval time of data transfer
- OSMS = #00000001B ; Oscillation mode select register; does not use divider circuit
- TCL3 = #1001××××B ; Serial clock $f_{XX}/2^4$
- Makes P22 output latch high
- PM2 = #×××1000×B ; Sets P21, P22, and P23 in output mode and P24 in input mode

<Starting>

Set the data to be transmitted to the buffer RAM (starting from the highest address), and the number of data to be transmitted to DTVAL, and call TR6451. You can check the end of data transfer by testing the bit 3 (TRF) of the automatic data transfer/reception control register (ADTC).

(2) Example of use

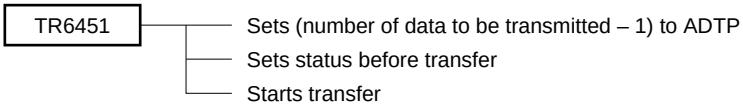



```
        EXTRN   TR6451,DTVAL

SCK1    EQU     P2.2
             :
             :
             :
        OSMS=#00000001B          ; Does not use divider circuit
        P2=#00000100B
        PM2=#11110001B
        CSIM1=#10100011B         ; Sets automatic transfer/reception function
        TCL3=#10011001B          ; SCK1 = 262 kHz
        ADTC=#00000110B          ; Enable strobe and busy signals
        ADTI=#00000000B          ; Interval time of data transfer
             :
             :
             :
        DE=#TABLE1               ; Sets table reference address of transmit data
        HL=#0FAC0H               ; Sets first address of buffer RAM
        B=32                     ; Sets number of data to be transmitted

        while(B>#0)              ; Transfers transmit data to buffer RAM
                B--
                [HL+B]=[DE] (A)
                DE++
        endw

        DATVAL=#32               ; Sets number of data to be transmitted
        CALL    !TR6451
        while_bit(TRF)           ; Waits for end of transfer
        endw
```

```
TABLE1:
        DB      11111111B               ; Power-ON reset, command 1
        DB      01000000B               ; Vertical address 0
        DB      11000000B               ; Horizontal address 0
        DB      10000000B               ; Character size
        DB      11111100B               ; Command 0
        DB      11101001B               ; Turns LC transmission ON, blinking OFF, display ON

        DB      10001100B               ; Turns blinking ON. Character: red

        DB      11011011B               ; Color specification, background filled in cyan
        DB      10010101B               ; Number of display lines: 5
        DB      10100000B               ; Number of display digits: 0


        DB      07H                     ; 7
        DB      08H                     ; 8
        DB      1BH                     ; K
        DB      6DH                     ; /
        DB      00H                     ; 0
        DB      10H
        DB      11H                     ; A
        DB      20H                     ; P
        DB      20H                     ; P
        DB      1CH                     ; L
        DB      19H                     ; I
        DB      13H                     ; C
        DB      11H                     ; A
        DB      24H                     ; T
        DB      19H                     ; I
        DB      00H                     ; O
        DB      1EH                     ; N
        DB      10H
        DB      1EH                     ; N
        DB      00H                     ; O
        DB      24H                     ; T
        DB      15H                     ; E
```

Remark For the command and data of the output table data, refer to **μPD6451A Data Sheet (Document No. IC-2337)**.

(3) SPD chart



(4) Program list

```
        PUBLIC  TR6451,DTVAL

CSI_DAT DSEG    SADDR
DTVAL:  DS      1                       ; Number of data setting area

CSI_SEG CSEG
;**********************************
;*       μPD6451A communication
;**********************************
TR6451:
        A=DTVAL                         ; Sets number of data
        A--
        ADTP=A
        SIO1=#0FFH                      ; Starts transfer
        RET
```

8.3 Interface in SBI Mode

The 78K/0 series has an SBI mode conforming to NEC serial bus format. In this mode, one master CPU can communicate with two or more slave CPUs by using two lines: clock and data. In the example shown in this section, the μPD78054 subseries is used.

Figure 8-32 shows an example of connection to use the SBI mode, and Figure 8-33 shows the communication format.

Figure 8-32. Example of Connection in SBI Mode

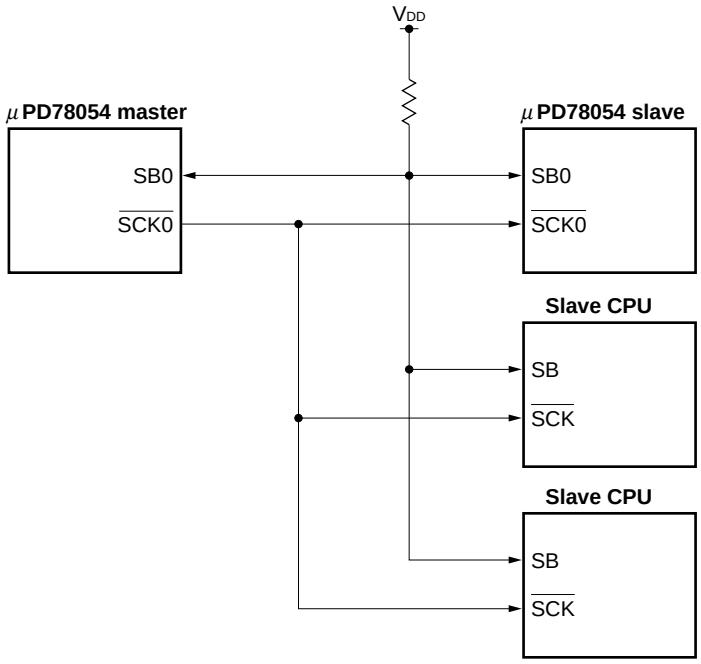
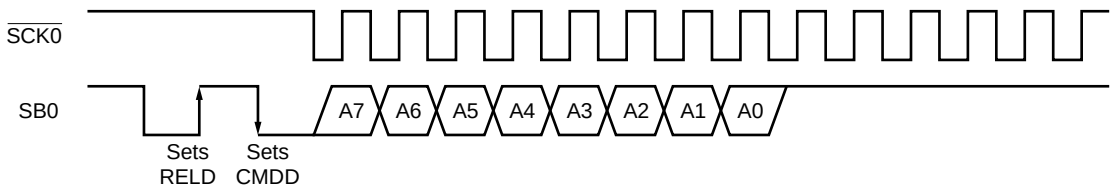
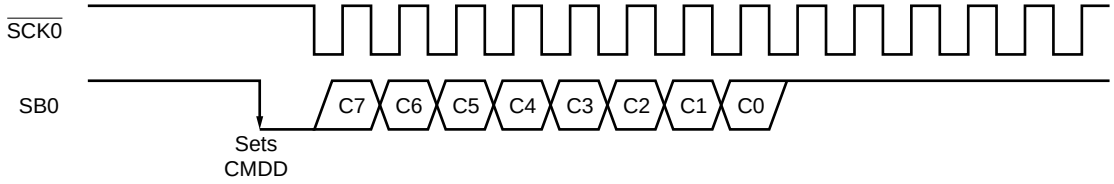


Figure 8-33. Communication Format in SBI Mode

(a) Address transmission



(b) Command transmission



(c) Data transmission/reception

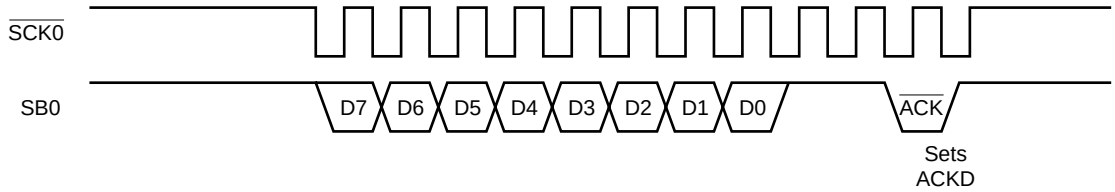


Table 8-9. Signals in SBI Mode

Signal Name	Output by:	Meaning
Address	Master	Selects slave device
Command	Master	Command to slave device
Data	Master/slave	Data to be processed by slave or master
Clock	Master	Serial data transmission/reception synchronization signal
$\overline{ACK}$	Receiver side[Note]	Reception acknowledge signal
$\overline{BUSY}$	Slave	Busy status

Note This signal is output by the receiver side during normal operation. However, it is output by the master CPU in case of an error such as time out.

8.3.1 Application as master CPU

When the μPD78054 subseries is used as a master CPU, it performs processing (a) through (d) below with respect to slave CPUs.

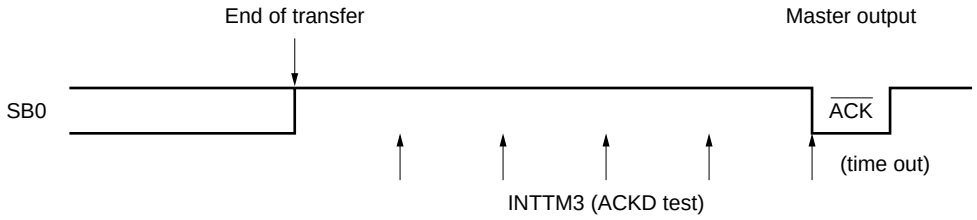
(a) Address transmission
(b) Command transmission
(c) Data transmission
(d) Data reception

While the above processing is performed, errors <1> and <2> below are checked.

<1> Time out processing

If the master CPU transmits data and a slave does not return the $\overline{ACK}$ signal within a specific time (in this example, before the watch interrupt request occurs five times), the master judges that an error has occurred. The master CPU then outputs an $\overline{ACK}$ signal and terminates the processing.

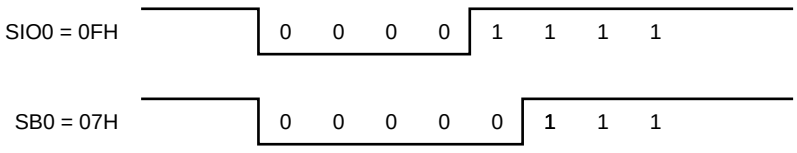
Figure 8-34. $\overline{ACK}$ Signal in Case of Time out



<2> Testing bus line

The master CPU tests whether data has been correctly output to the bus line by setting the transmit data to the serial I/O shift register 0 (SIO0) and the slave address register (SVA). Because the data on the bus line is received by SIO0, it confirms that the data has been output normally by testing bit 6 (COI) of the serial operating mode register 0 (CSIM0) (that is set when SIO0 coincides with SVA) at the end of transfer.

Figure 8-35. Testing Bus Line



In Figure 8-35, the values of SIO0 and SVA do not coincide (SIO0 = 07H and SVA = 0FH). Consequently, COI = 0, and an error has occurred on the bus line.

(1) Description of package

<Public declaration symbol>

M_TRANS : Master SBI transfer subroutine name
TR_MODE : Storage area of transfer mode select value
TRNDAT : Transmit data storage area
RCVDAT : Receive data storage area
TRADR : Address transmit mode select value
TRCMD : Command transmit mode select value
TRDAT : Data transmit mode select value
RCDAT : Data reception mode select value
ERRORF : Error status test flag

<Register used>

Subroutine A

<RAM used>

Name	Usage	Attribute	Bytes
TR_MODE	Stores transfer mode select value	SADDR	1
ACKCT	Time out counter		
TRNDAT	Stores transmit data		
RCVDAT	Stores receive data		

<Flag used>

Name	Usage
RCVFLG	Sets reception mode
BUSYFG	Sets busy status
ERRORF	Sets error status
ACKWFG	Sets $\overline{\text{ACK}}$ signal wait status

<Nesting>

2 levels 5 bytes

<Hardware used>

- Serial interface channel 0
- Watch timer

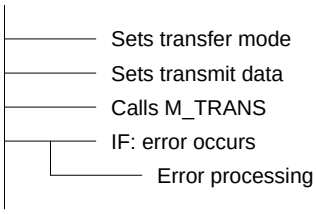
<Initial setting>

- OSMS=#00000001B ; Oscillation mode select register: does not use divider circuit
- Sets serial interface channel 0
  CSIM0=#10010011B ; Selects SBI mode and SB1 pin
- TCL3=#××××1001B ; Serial clock: $f_{XX}/2^4$
- RELT=1 ; Makes SO0 latch high
- P27=1 ; Makes P27 output latch high
- TMC2=#00100110B ; Watch timer interval: 1.95 ms
- Enables watch timer interrupt

<Starting>

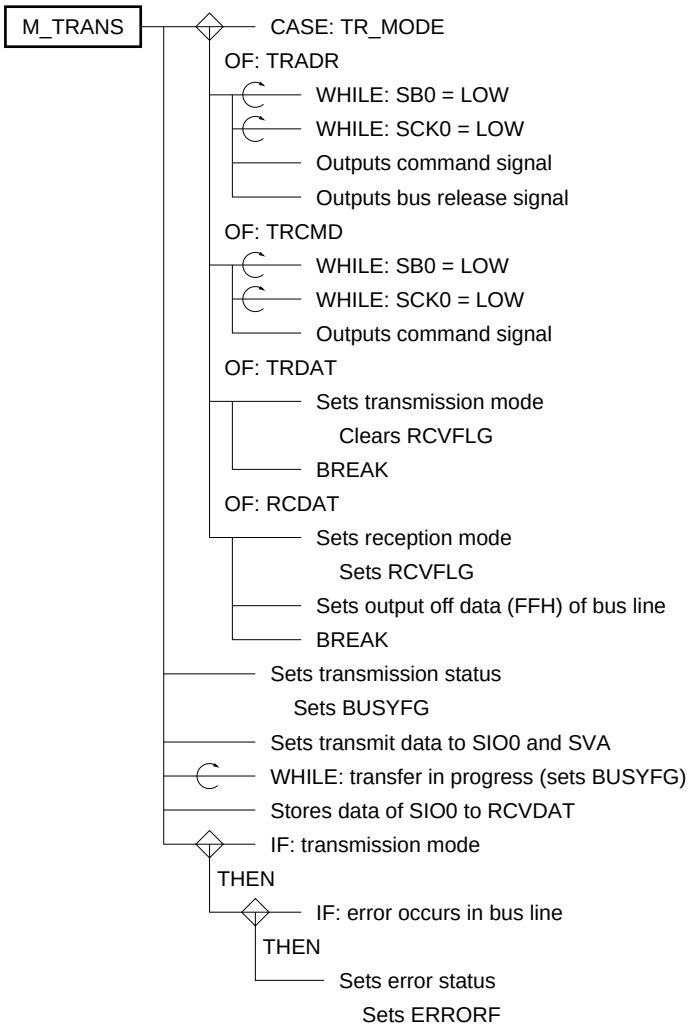
Set the transfer mode and necessary data, and call M_TRANS. When execution has returned from the subroutine, occurrence of a transfer error can be checked by testing the error flag (ERRORF). In the reception mode, the receive data is stored to RCVDAT after execution has returned from the subroutine.

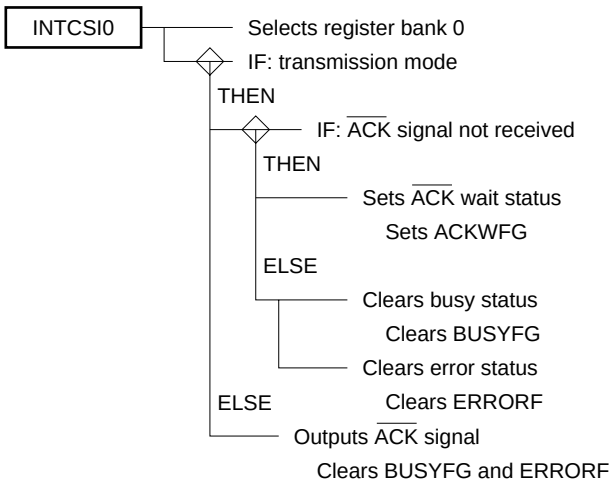
(2) Example of use

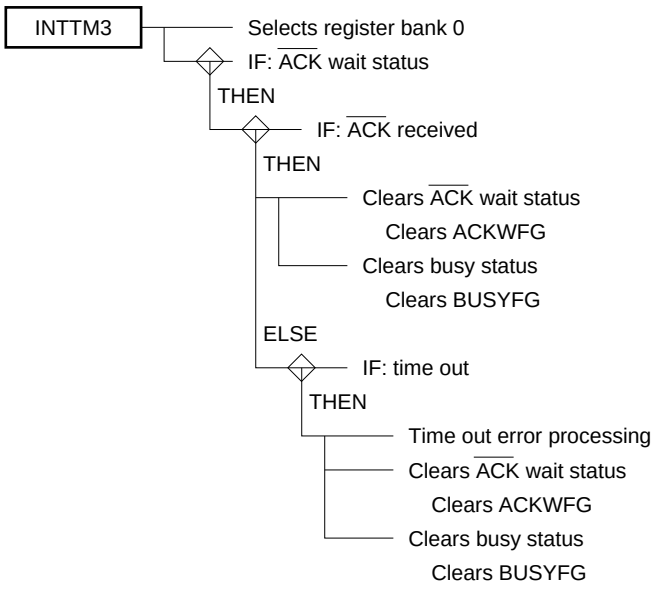



```
EXTRN   M_TRANS,TR_MODE,TRADR,TRCMD,TRDAT,RCDAT
EXTRN   TRNDAT,RCVDAT
EXTBIT  ERRORF

SCK0    EQU     P2.7
SB1     EQU     P2.5
      :
      :
OSMS=#00000001B                 ; Does not use divider circuit
SET1    SB1
CSIM0=#10010111B                ; Operates in SBI mode
TCL3=#10011001B                 ; SCK0 = 262 kHz
TMC2=#00100110B                 ; Sets interval of watch timer to 1.95 ms
CLR1    BSYE                    ; Disables output of busy signal
SET1    RELT                    ; Sets output latch
SET1    SCK0
CLR1    SB1
CLR1    CSIMK0                  ; Enables serial interface channel 0 interrupt
CLR1    TMMK3                   ; Enables watch timer interrupt
EI                              ; Enables master interrupt
      :
      :
TR_MODE=#TRADR
TRNDAT=#5AH
CALL    !M_TRANS
if_bit(ERRORF)
    Error processing
endif
```

(3) SPD chart


M_TRANS
CASE: TR_MODE
OF: TRADR
WHILE: SB0 = LOW
WHILE: SCK0 = LOW
Outputs command signal
Outputs bus release signal
OF: TRCMD
WHILE: SB0 = LOW
WHILE: SCK0 = LOW
Outputs command signal
OF: TRDAT
Sets transmission mode
Clears RCVFLG
BREAK
OF: RCDAT
Sets reception mode
Sets RCVFLG
Sets output off data (FFH) of bus line
BREAK
Sets transmission status
Sets BUSYFG
Sets transmit data to SIO0 and SVA
WHILE: transfer in progress (sets BUSYFG)
Stores data of SIO0 to RCVDAT
IF: transmission mode
THEN
IF: error occurs in bus line
THEN
Sets error status
Sets ERRORF


INTCSI0
Selects register bank 0
IF: transmission mode
THEN
IF: $\overline{ACK}$ signal not received
THEN
Sets $\overline{ACK}$ wait status
Sets ACKWFG
ELSE
Clears busy status
Clears BUSYFG
Clears error status
Clears ERRORF
ELSE
Outputs $\overline{ACK}$ signal
Clears BUSYFG and ERRORF


INTTM3
Selects register bank 0
IF: $\overline{ACK}$ wait status
THEN
IF: $\overline{ACK}$ received
THEN
Clears $\overline{ACK}$ wait status
Clears ACKWFG
Clears busy status
Clears BUSYFG
ELSE
IF: time out
THEN
Time out error processing
Clears $\overline{ACK}$ wait status
Clears ACKWFG
Clears busy status
Clears BUSYFG

(4) Program list

```
        PUBLIC  M_TRANS,TR_MODE,TRADR,TRCMD,TRDAT,RCDAT
        PUBLIC  TRNDAT,RCVDAT,ERRORF

VECSI0  CSEG    AT 14H
        DW      INTCSI0         ; Sets vector address of serial interface channel 0
VETM3   CSEG    AT 1EH
        DW      INTTM3          ; Sets vector address of watch timer

SBI_DAT DSEG    SADDR
TRNDAT: DS      1               ; Transmit data
RCVDAT: DS      1               ; Receive data
TR_MODE:DS      1               ; Sets transfer mode
ACKCT:  DS      1               ; ACK time out count

SBI_FLG BSEG
RCVFLG  DBIT                    ; Sets reception mode
BUSYFG  DBIT                    ; Transfer status
ERRORF  DBIT                    ; Error display
ACKWFG  DBIT                    ; ACK wait status

SB0     EQU     P2.5
SCK0    EQU     P2.7

TRADR   EQU     1               ; Selects address transmission mode
TRCMD   EQU     2               ; Selects command transmission mode
TRDAT   EQU     3               ; Selects data transmission mode
RCDAT   EQU     4               ; Selects data reception mode
```

```
;****************************************
;*       SBI data transfer processing
;****************************************
SBI_SEG CSEG
M_TRANS:
    switch(TR_MODE)
    case TRADR:
        SET1    PM2.5
        while_bit(!SB0)                     ; SB0 = high?
        CLR1    PM2.5
        endw
        while_bit(!SCK0)                    ; SCK = high?
        endw
        SET1    CMDT                        ; Outputs command signal
        NOP                                 ; Wait
        SET1    RELT                        ; Outputs bus release signal
        A=#TRCMD
    case TRCMD:
        SET1    PM2.5
        while_bit(!SB0)                     ; SB0 = high?
        CLR1    PM2.5
        endw
        while_bit(!SCK0)                    ; SCK = high?
        endw
        SET1    CMDT                        ; Outputs command signal
        A=#TRDAT
    case TRDAT:
        CLR1    RCVFLG                      ; Sets transmission mode
        A=TRNDAT                            ; Sets transmit data
        break
    case RCDAT:
        SET1    RCVFLG                      ; Sets reception mode
        MOV     A,#0FFH                     ; Turns off receive buffer
        break
    ends

        SET1    BUSYFG                      ; Sets transfer status
        SVA=A                               ; Tests bus line
        SIO0=A                              ; Starts transfer

        while_bit(BUSYFG)                   ; Transfer in progress
        endw
        RCVDAT=SIO0 (A)                     ; Stores receive data
        if_bit(!RCVFLG)                     ; Transmission mode
            if_bit(!COI)                    ; Bus line output abnormal
                SET1    ERRORF              ; Sets error status
            endif
        endif
        RET
```

```
;************************************
;*     INTCSI0 interrupt processing
;************************************
CSI_SEG CSEG
INTCSI0:
        SEL RB0
        if_bit(!RCVFLG)                     ; Transmission mode
            if_bit(!ACKD)                   ; Acknowledge signal not received
                ACKCT=#5                    ; Sets acknowledge signal wait status
                SET1    ACKWFG
            else
                CLR1    BUSYFG              ; Clears busy status
                CLR1    ERRORF              ; Clears error status
            endif
        else
            SET1    ACKT                    ; Outputs acknowledge signal
            CLR1    BUSYFG                  ; Clears busy status
            CLR1    ERRORF                  ; Clears error status
        endif
        RET

;************************************
;*         Time out processing
;************************************
TM3_SEG CSEG
INTTM3:
        SEL RB0
        if_bit(ACKWFG)                      ; Acknowledge signal wait status?
            if_bit(ACKD)                    ; Acknowledge signal received?
                CLR1     ACKWFG             ; Clears acknowledge signal wait status
                CLR1     BUSYFG             ; Clears busy status
            else
                ACKCT--
                if(ACKCT==#0)               ; Time out?
                    SET1    ACKT            ; Time out error processing
                    SET1    ERRORF
                    CLR1    ACKWFG          ; Clears acknowledge signal wait status
                    CLR1    BUSYFG          ; Clears busy status
                endif
            endif
        endif
```

8.3.2 Application as slave CPU

A slave CPU receives addresses, commands, and data from the master CPU and transmits data to the master CPU.

In the example shown in this section, addresses are received by using the wake-up function. This function is to generate an interrupt only when the address value transmitted by the master to the slave coincides with the value set to the slave address register (SVA) of the slave in the SBI mode. Therefore, only the slave CPU selected by the master CPU generates INTCSI0, and the slave CPUs not selected operates without generating an inadvertent interrupt request.

The slave CPU clears the wake-up function when it has been selected by the master (the interrupt request signal is generated at the end of transmission), and interfaces with the master CPU. Addresses, commands, and data being transmitted are identified by using bits 2 and 3 (RELD and CMDD) of the serial bus interface control register (SB IC).

Because the slave CPU is not automatically placed in the unselect status, a program that returns the slave CPU to the unselect status must be prepared by processing commands between the master and CPU.

(1) Description of package

<Public declaration symbol>

RCVDAT: Receive data storage area

<Register used>

Bank 0: A

<RAM used>

Name	Usage	Attribute	Bytes
RCVDAT	Stores receive data	SADDR	1

<Flag used>

Name	Usage
RCVFLG	Sets reception mode

<Nesting>

1 level 3 bytes

<Hardware used>

- Serial interface channel 0

<Initial setting>

- Setting of serial interface channel 0
  CSIM0=#10010011B; Sets SBI mode, SBI pin, and wake-up mode, and inputs serial clock from external source
- BYSE=1 Outputs synchronous busy signal
- RELT=1 Makes SO0 latch high
- SVA=#SLVADR; Slave address
- Enables serial interface channel 0 interrupt

<Starting>

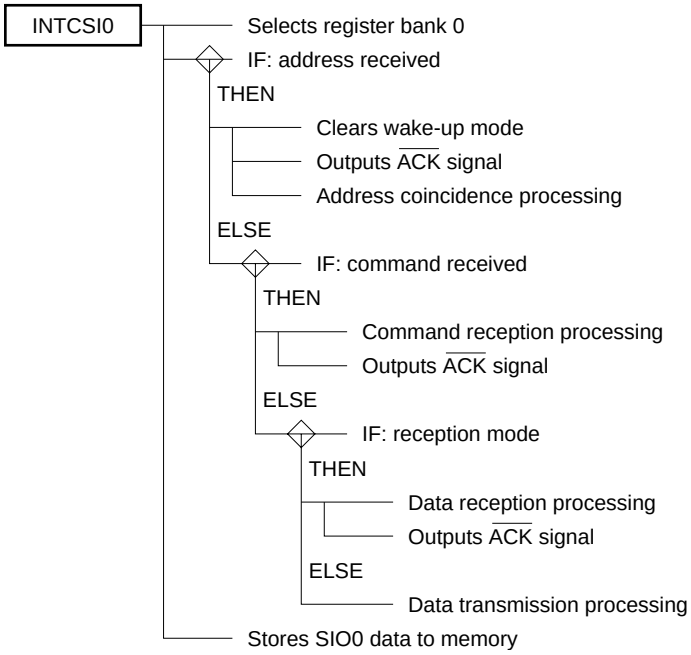
The interrupt processing is started by generation of INTCSI0. The interrupt processing performs the following processing:

- Identifies address/command/data
- Outputs $\overline{ACK}$ signal
- Stores receive data to RCVDAT

(2) Example of use

```
EXTRN   RCVDAT
EXTBIT  RCVFLG

SLVADR  EQU     5AH
SB1     EQU     P2.5
     :
     :
SET1    SB1
CSIM0=#10110100B        ; Inputs external clock, sets SB1 pin, and selects wake-up mode
SET1    RELT            ; Sets output latch to high level
SET1    BSYE            ; Sets busy automatic output
SVA=#SLVADR             ; Sets slave address
SIO0=#0FFH              ; Serial transfer start command
CLR1    SB1
CLR1    CSIMK0          ; Enables serial interface channel 0 interrupt
EI                      ; Enables master interrupt
```

(3) SPD chart



(4) Program list

```
VECSI0  CSEG   AT 14H
        DW     INTCSI0                                  ; Sets vector address of serial
                                                          interface channel 0
SCI_DAT DSEG   SADDR
RCVDAT: DS     1                                        ; Receive data storage area

CSI_FLG BSEG
RCVFLG  DBIT                                            ; Sets reception mode

CSI_SEG CSEG
;*************************************
;*     INTCSI0 interrupt processing
;*************************************
INTCSI0:
        SEL RB0
        if_bit(RELD)                                    ; To address reception
            CLR1       WUP                              ; Clears wake-up mode
            SET1       ACKT                             ; Outputs acknowledge signal
;    User processing (address reception)

;*************************************

        elseif_bit(CMDD)                                ; To command reception
;           User processing (command reception)

            SET1       ACKT                             ; Outputs acknowledge signal
        else

            if_bit(RCVFLG)
;               User processing (data reception processing)
                SET1    ACKT                            ; Outputs acknowledge signal
            else
;               User processing (data transmission processing)
            endif
;************************************
        endif
        RCVDAT=SIO0 (A)

        RETI
```

8.4 Interface in 3-Wire Serial I/O Mode

In this section, examples of communication between the master and a slave by using the 3-wire serial I/O mode (serial clock, data input, data output) of the serial channel 0 of the 78K/0 series are shown. In these examples, one extra busy signal is used as a handshake signal for simultaneous transmission/reception between the master and slave. This busy signal is active-low and is output by the slave. The data is 8 bits long and transmitted with the MSB first. In the examples in this section, the μPD78054 subseries is used.

Figure 8-36. Example of Connection in 3-Wire Serial I/O Mode

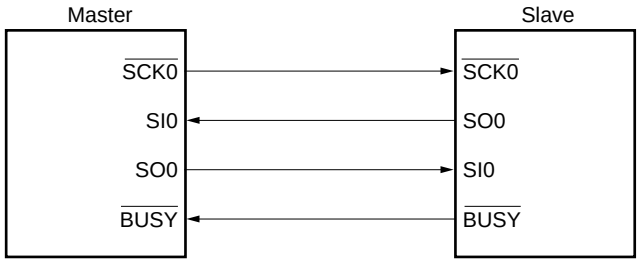
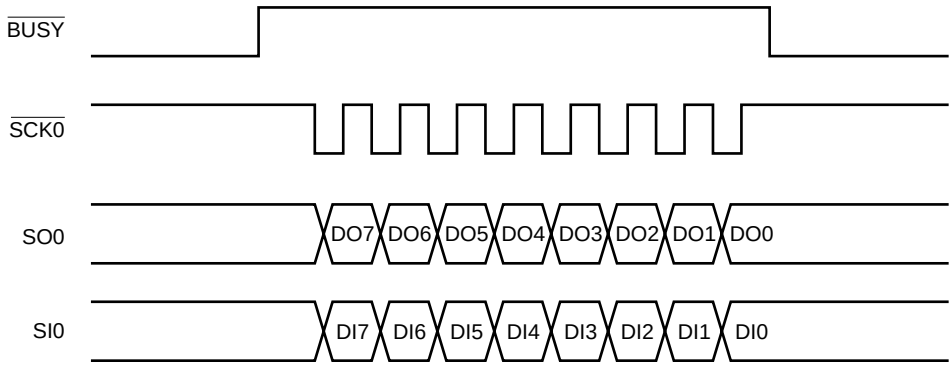


Figure 8-37. Communication Format in 3-Wire Serial I/O Mode



8.4.1 Application as master CPU

The serial clock is set to $f_{XX}/2^4$, and communication is executed in synchronization with this serial clock between the master and slave CPUs.

The master CPU starts transmission after it has set the transmit data. If the slave CPU is busy (when the busy signal is low), however, the master does not transmit data and sets the busy flag (BUSYFG).

(1) Description of package

<Public declaration symbol>

TRANS : Name of 3-wire transfer subroutine of master
TDATA : Transmit data storage area
RDATA : Receive data storage area
BUSY : Busy signal input port
TREND : Transfer end test flag
BUSYFG : Busy status test flag

<Register used>

Interrupt : Bank 0, A
Subroutine : A

<RAM used>

Name	Usage	Attribute	Bytes
TDATA	Stores transmit data	SADDR	1
RDATA	Stores receive data		

<Flag used>

Name	Usage
TREND	Sets transfer end status
BUSYFG	Sets busy status

<Nesting>

2 levels 5 bytes

<Hardware used>

- Serial interface channel 0
- P33

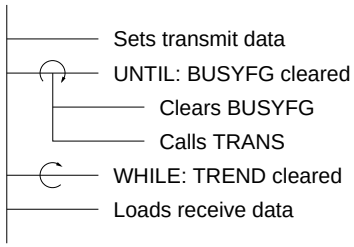
<Initial setting>

- OSMS=#00000001B ; Oscillation mode select register: does not used divider circuit
- Setting of serial interface channel 0
  CSIM0=#10000011B ; 3-wire serial I/O mode, MSB first
- TCL3=#××××1001B ; Serial clock $f_{XX}/2^4$
- P27=1 ; Makes P27 output latch high
- P33 input mode
- Enables serial interface channel 0 interrupt

<Starting>

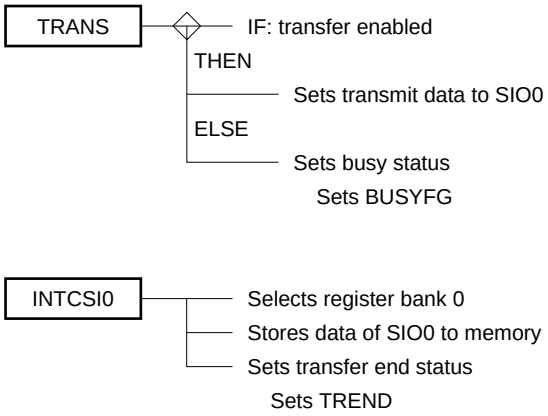
Set the transmit data to TDATA and call TRANS. After execution has returned from the subroutine, test the busy flag (BUSYFG). If the busy flag is set, transfer has not been executed and therefore, you must execute it again. If the busy flag is cleared, transfer has ended and the receive data has been stored to RDATA.

(2) Example of use




```
EXTRN   TDATA,RDATA,TRANS
EXTBIT  TREND,BUSYFG,BUSY

SCK0    EQU     P2.7
          .
          .
          .
    OSMS=#00000001B             ; Does not use divider circuit
    CSIM0=#10000011B            ; Sets 3-wire serial I/O mode with MSB first
    TCL3=#10011001B             ; Sets SCK0 = 262 kHz
    SET1    SCK0
    SET1    PM3.3               ; Sets P3.3 input mode
    CLR1    CSIMK0              ; Enables serial interface channel 0
    EI
          .
          .
          .
    TDATA=A                     ; Sets transmit data
    repeat
            CLR1    BUSYFG      ; Busy test
            CALL    !TRANS
    until_bit(!BUSYFG)
    while_bit(!TREND)           ; Ends transfer
    endw
    A=RDATA                     ; Loads receive data
```

(3) SPD chart



(4) Program list

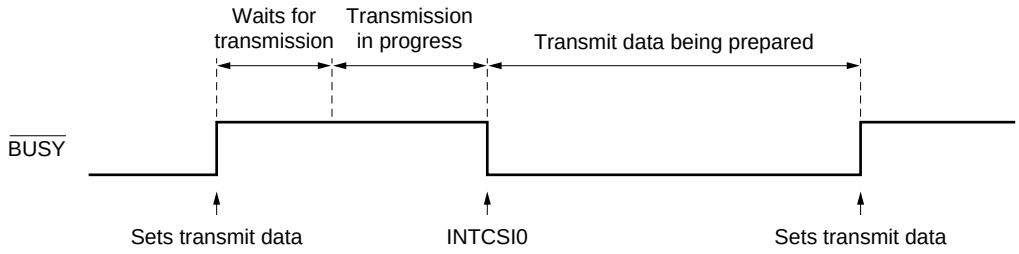
```
        PUBLIC  TRANS,RDATA,TDATA,BUSY,TREND,BUSYFG
VECSI0  CSEG    AT 14H
        DW      INTCSI0                         ; Sets vector address of serial interface channel 0

BUSY    EQU     0FF03H.3                        ; 0FF03H = PORT3

CSI_DAT DSEG    SADDR
RDATA:  DS      1                               ; Receive data storage area
TDATA:  DS      1                               ; Transmit data storage area

CSI_FLG BSEG
TREND   DBIT                                    ; Sets transfer end status
BUSYFG  DBIT                                    ; Sets busy status

CSI_SEG CSEG

;************************************
;:*    INTCSI0 interrupt processing
;************************************
INTCSI0:
        SEL RB0
        RDATA=SIO0 (A)                          ; Stores receive data
        SET1    TREND                           ; Sets transfer end status
        RETI

;************************************
;*         3-wire (master)
;************************************
TRANS:
        if_bit(BUSY)
            SIO0=TDATA (A)                      ; Enables transfer
        else                                    ; Sets transmit data
            SET1    BUSYFG
        endif                                   ; Sets busy status
        RET
```

8.4.2 Application as slave CPU

In this example, a slave CPU simultaneously transmits and receives 8-bit data in synchronization with the serial clock from the master CPU. The busy signal output by the slave CPU is low (busy status) while the transmit data is prepared. This busy signal is cleared (high level) when the transmit data is set (CALL !TRANS), and is output (low level) when interrupt INTCSI0 occurs at the end of transfer.

Therefore, the busy status remains after the end of transfer until the data is set.

Figure 8-38. Output of Busy Signal



(1) Description of package

<Public declaration symbol>

TRANS : Name of 3-wire transfer subroutine of slave
TDATA : Transmit data storage area
RDATA : Receive data storage area
BUSY : Busy signal output port
TREND : Transfer end test flag

<Register used>

Interrupt : Bank 0, A
Subroutine : A

<RAM used>

Name	Usage	Attribute	Bytes
TDATA	Stores transmit data	SADDR	1
RDATA	Stores receive data		

<Flag used>

Name	Usage
TREND	Sets transfer end status

<Nesting>

2 level 5 bytes

<Hardware used>

- Serial interface channel 0
- P33

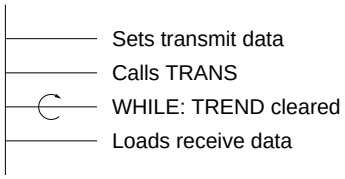
<Initial setting>

- Setting of serial interface channel 0
  CSIM0=#10000000B ; Sets 3-wire serial I/O mode with MSB first, and inputs external clock
- P33=0 ; P33 output mode
- Setting of busy status
- Enables serial interface channel 0

<Starting>

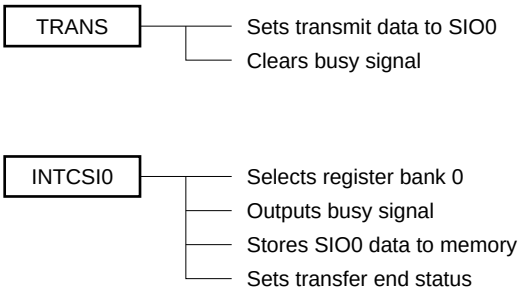
Set the transmit data to TDATA and call TRANS. Because the busy signal is cleared by the processing of TRANS, the slave waits for communication with the master. After the communication has ended, INTCSI0 occurs and interrupt processing is started. You can check the end of transfer by testing TREND. After TREND has been set, the receive data has been stored to RDTA.

(2) Example of use




```
EXTRN   TDATA,RDATA,TRANS
EXTBIT  TREND,BUSY
    .
    .
    .
CSIM0=#10000000B            ; Sets 3-wire I/O mode with MSB first
CLR1    BUSY                ; Busy status
CLR1    PM3.3               ; P3.3 output mode
CLR1    CSIMK0              ; Enables serial interface channel 0
EI
    .
    .
    .
TDATA=A                     ; Sets transmit data
CALL    !TRANS
while_bit(!TREND)           ; Ends transfer
endw
A=RDATA                     ; Loads receive data
```

(3) SPD chart



(4) Program list

```
        PUBLIC  RDATA,TDATA,BUSY,TREND,BUSYFG
        PUBLIC  TRANS
VECSI0  CSEG    AT 14H
        DW      INTCSI0                         ; Sets vector address of serial interface channel 0

CSI_DAT DSEG    SADDR
RDATA:  DS      1                               ; Stores receive data
TRADA:  DS      1                               ; Stores transmit data

CSI_FLG BSEG
TREND   DBIT                                    ; Sets transfer end status
BUSYFG  DBIT                                    ; Sets busy status

BUSY    EQU     0FF03H.3                        ; 0FF03H = PORT3

CSI_SEG CSE
;************************************
;*    INTCSI0 interrupt processing
;************************************
INTCSI0:
        SEL RB0
        CLR1    BUSY                            ; Sets busy status
        RDATA=SI00 (A)                          ; Stores receive data
        SET1    TREND                           ; Sets transfer end status
        RETI

;************************************
;*        3-wire (slave)
;************************************
TRANS:
        SIO0=TDATA (A)                          ; Sets transmit data
        SET1    BUSY                            ; Clears busy status
        RET
```

8.5 Interface in Asynchronous Serial Interface (UART) Mode

Serial interface channel 2 has two modes: asynchronous serial interface (hereafter referred to as “UART”) and 3-wire serial I/O modes.

Serial interface channel 2 is set by the following registgers:

- Serial operating mode register 2 (CSIM2)
- Asynchronous serial interface mode register (ASIM)
- Asynchronous serial interface status register (ASIS)
- Baud rate generator control register (BRGC)
- Oscillation mode select register (OSMS)

UART using serial interface channel 2 is briefly described below.

The UART mode of serial interface channel 2 is to transmit or receive 1-byte data following a start bit and can perform full-duplex operation.

The operations of UART communication are described below.

(a) Communication format

One data frame of transmit/receive data consists of a start bit, character bits, parity bit, and stop bit. The character bit length, parity, and stop bit length in one data frame are specified by using the asynchronous serial interface mode register (ASIM).

(b) Setting of baud rate

A UART dedicated baud rate generator is provided that can set a wide range of baud rates. A baud rate can also be defined by dividing the clock input to the ASCK pin.

The transmit/receive clock for the band rate is generated by dividing the main system clock. The baud rate generated from the main system clock can be calculated by the following expression. Table 8-10 shows the relations between the main system clock and baud rate (at fx = 4.19 MHz).

$$[\text{Baud rate}] = \frac{f_{XX}}{2^n \times (k + 16)} \text{ [Hz]}$$

Remarks 1. f_{XX}: main system clock frequency (f_X or $f_X/2$)

2. f_X : main system clock oscillation frequency

3. n : value set by TPS0-TPS3
(bits 4-7 of the baud rate generator control register (BRGC) ($1 \leq n \leq 11$)

4. k : value set by MDL0-MDL3 (bits 0-3 of BRGC) ($0 \leq k \leq 14$)

Table 8-10. Relations between Main System Clock and Baud Rate (at f_X = 4.19 MHz)

Baud rate (bps)	MCS = 1		MCS = 0	
	Set value of BRGC	Error (%)	Set value of BRGC	Error (%)
75	0BH	1.14	EBH	1.14
110	03H	–2.01	E3H	–2.01
150	EBH	1.14	DBH	1.1.4
300	DBH	1.14	CBH	1.14
600	CBH	1.14	BBH	1.14
1200	BBH	1.14	ABH	1.14
2400	ABH	1.14	9BH	1.14
4800	9BH	1.14	8BH	1.14
9600	8BH	1.14	7BH	1.14
19200	7BH	1.14	6BH	1.14
31250	71H	–1.31	61H	–1.31
38400	6BH	1.14	5BH	1.14
★ 76800	5BH	1.14	–	–

Remark MCS: bit 0 of the oscillation mode select register (OSMS)

(c) Transmission

Transmission is started when transmit data has been written to the transmit shift register (TXS). The start bit and parity bit are automatically appended.

(d) Reception

Reception is enabled when bit 6 (RXE) of the asynchronous serial interface mode register (ASIM) is set to 1, and the data input to the RxD pin is sampled. When reception of one frame of data has been completed, the receive data in the shift register is transferred to the receive buffer register (RXB) and a receive end interrupt request (INTSR) occurs.

(e) Receive error

During reception, three types of errors may occur: parity error, framing error, and overrun error. If the error flag of the asynchronous serial interface status register (ASIS) is set as a result of data reception, a receive error interrupt (INTSER) occurs. By reading the contents of ASIS in the receive error interrupt processing (INTSER), which error has occurred can be identified. The contents of ASIS are reset (0) by either reading the receive buffer register (RXB) or receiving the next data. (if the next data includes an error, that error flag is set).

Cautions 1. The contents of the asynchronous serial interface status register (ASIS) are reset to 0 when the receive buffer register (RXB) is read or the next data is received. To determine the nature of the error, be sure to read ASIS before reading RXB.

2. Be sure to read the receive buffer register (RXB) when a reception error has occurred. Otherwise, an overrun error will occur when the next data is received, and the reception error status will persist.

During communication, transmission and reception with a terminal is performed and $\overline{\text{RTS}}$ and $\overline{\text{CTS}}$ are controlled for handshaking. The communication protocol is shown below.

- Baud rate: 9600 bps
- No parity bit
- Stop bit: 2 bits
- LSB first
- $\overline{\text{CTS}}$ input pin: P31
- $\overline{\text{RTS}}$ output pin: P32

When transmission is started, the end of the previous transmission (in which case the transmission end interrupt request flag (STIF) is set to 1) is checked, and transmission is executed if the $\overline{\text{CTS}}$ input status is ready (“L”).
During reception, the busy signal (“H”) is output to the $\overline{\text{RTS}}$ output pin when a reception end interrupt request (INTSR) occurs. “L” is output to the $\overline{\text{RTS}}$ output pin when reception is enabled.
A receive error interrupt request (INTSER) occurs if a receive error (parity error, framing error, or overrun error) occurs, and the error flag is set. Figure 8-39 shows a communication block diagram, and Figures 8-40 and 8-41 show the transmission/reception format.

Figure 8-39. Communication Block Diagram

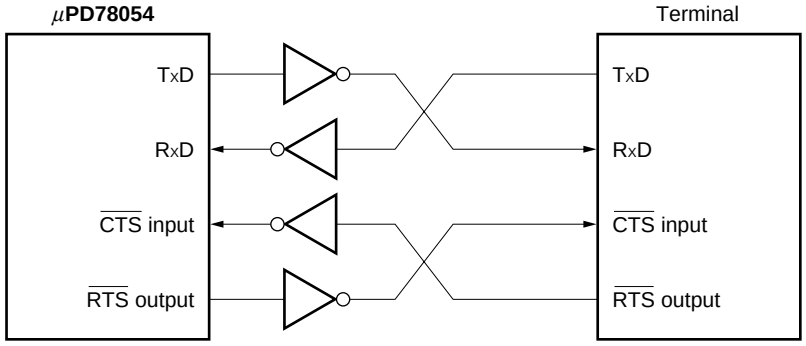


Figure 8-40. Communication Format

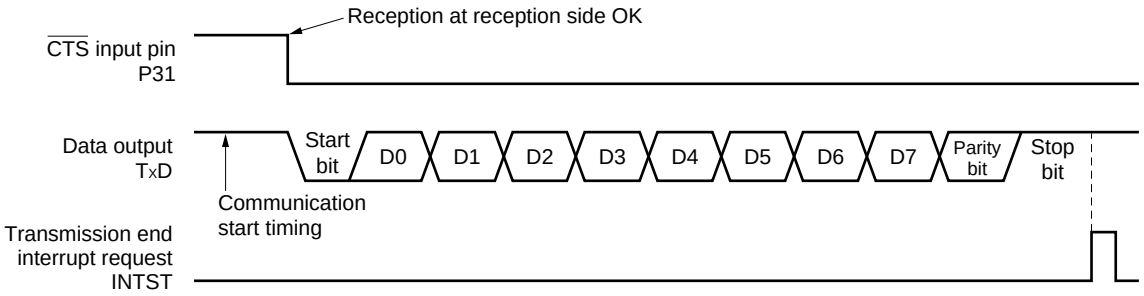
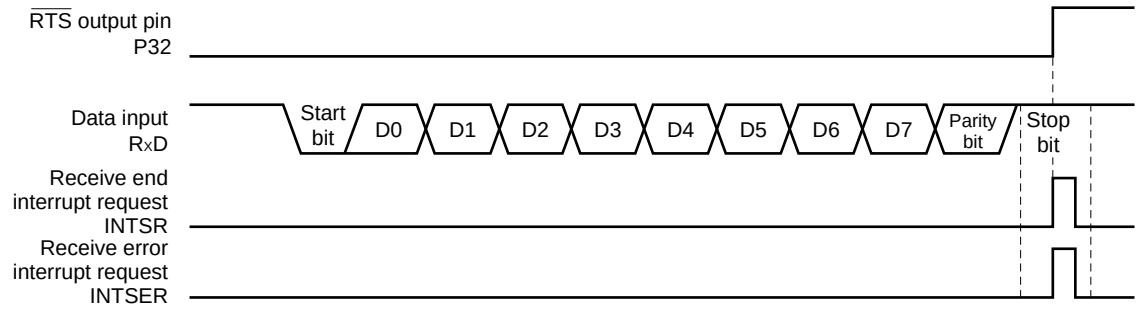


Figure 8-41. Reception Format



(1) Description of package

<Public declaration symbol>

- Subroutine name
  S_SOSHIN : Transmit routine
- Input parameter of S_SOSHIN routine
  SOSHIN : Transmit data storage area
- Output parameter of S_SOSHIN routine
  F_BUSY : Transmit busy flag
- Output parameter of INTSR interrupt
  JUSHIN : Receive data storage area
  F_TUSHIN : Reception end flag
- Output parameter of INTSER interrupt
  F_ERR : Receive error flag

<Register used>

S_SOSHIN : Bank 0, A
INTSR : Bank 3, A
INTSER : Bank 3, A

<RAM used>

Name	Usage	Attribute	Bytes
SOSHIN	Transmit data storage area	SADDR	1
JUSHIN	Receive data storage area	SADDR	1

<Flag used>

Name	Usage
F_TUSHIN	Set at end of reception
F_BUSY	Set if transmission cannot be started by $\overline{\text{CTS}}$ input pin; cleared if transmission can be started
F_ERR	Set if receive error occurs

<Nesting level>

2 levels 5 bytes

<Hardware used>

- Serial interface channel 2 (UART mode)

<Initial setting>

- OSMS=#00000001B ; Oscillation mode select register: does not use divider circuit
- CLR1 P3.2 ; P31 = $\overline{\text{CTS}}$ input, P32 = $\overline{\text{RTS}}$ output
  PM3=#××××10×B
- BRGC=#10001011B ; Sets baud rate to 9600 bps (error: 1.14%)
- CSIM2=#00000000B ; Sets 0 to serial operation mode register 2 when UART is used
- ASIM=#11001101B ; Sets asynchronous serial interface mode register
- CLR1 SRIF ; Clears reception end receive error interrupt request flags
  CLR1 SERIF
- SET1 STIF ; Sets transmission end interrupt request flag (to end transmission)
- CLR1 SRMK ; Enables reception end and receive error interrupts
  CLR1 SERMK

Caution Before starting transmission, check the transmission end interrupt request flag (STIF) so that transmission is not executed during transmission. Therefore, set the transmission end interrupt request flag (STIF) after reset and start.

Remark To use the transmission end interrupt (to generate the interrupt request), use an additional flag. Set the additional flag as the initial setting. Clear the flag at the start of transmission, and set it in the interrupt processing.

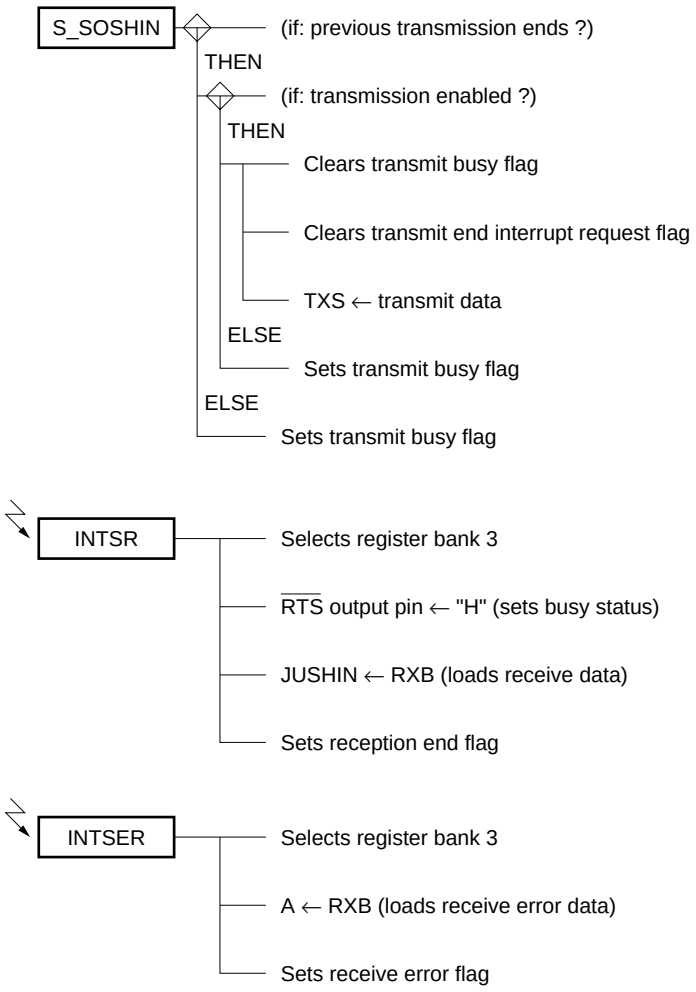
<Starting>

Store the transmit data to the SOSHIN area at the start of transmission and call the S_SOSHIN routine.

(2) Example of use

```
EXTRN   S_SOSHIN
EXTRN   SOSHIN,JUSHIN
EXTBIT  F_TUSHIN,F_ERR,F_BUSY
;
RTS_0   EQU     P3.2                ; RTS output port
;
    OSMS=#00000001B                 ; Does not use divider circuit
    CLR1    RTS_0                   ;
    PM3=#11111011B                  ; P31 = CTS input, P32 = RTS output
    BRGC=#10001011B                 ; 9600 bps (error: 1.41%)
    CSIM2=#00000000B                ; Initial setting when UART is used
    ASIM=#11001101B                 ; Enables receive error interrupt. Stop bit: 2 bits
                                    ; Transmit data: 8 bits. No parity. Enables reception and transmission.
    CLR1    SERIF                   ; Clears receive error interrupt request flag
    CLR1    SRIF                    ; Clears reception end interrupt request flag
    SET1    STIF                    ; Sets transmit end interrupt request flag
;                                   ;   →Ends transmission
    CLR1    SERMK                   ; Enables receive error interrupt
    CLR1    SRMK                    ; Enables reception end interrupt
    EI                              ;

          :
          :
          :
    if_bit(transmission request)    ; Sets transmission request flag?
       SOSHIN=A                     ;   Stores transmit data
       CALL    !S_SOSHIN            ;   Calls transmit routine
    endif
    if_bit(F_BUSY)                  ;   End of transmission?
          .
       Communication busy processing;
          .
    endif                           ;
          :
    if_bit(F_TUSHIN)                ; Sets reception end flag?
       CLR1    F_TUSHIN             ;   Clears reception end flag
       A=JUSHIN                     ;   Reads receive data
          .
       Reception processing         ;
          .
       CLR1    RTS_0                ; RTS output pin ← “L” (ready status)
    endif                           ;
          :
    if_bit(F_ERR)                   ; Receive error occurs?
       CLR1    F_ERR                ;
          .
       Receive error processing     ;
          .
    endif                           ;
          :
```

(3) SPD chart


S_SOSHIN
(if: previous transmission ends ?)
THEN
(if: transmission enabled ?)
THEN
Clears transmit busy flag
Clears transmit end interrupt request flag
TXS ← transmit data
ELSE
Sets transmit busy flag
ELSE
Sets transmit busy flag
INTSR
Selects register bank 3
RTS output pin ← "H" (sets busy status)
JUSHIN ← RXB (loads receive data)
Sets reception end flag
INTSER
Selects register bank 3
A ← RXB (loads receive error data)
Sets receive error flag

(4) Program list

```
PUBLIC  S_SOSHIN
PUBLIC  SOSHIN,JUSHIN
PUBLIC  F_TUSHIN,F_ERR,F_BUSY
;
VESR    CSEG    AT 1AH
        DW      INTSR
VESER   CSEG    AT 18H
        DW      INTSER
;
RTS_0       EQU     P3.2            ; RTS output port
CTS_I       EQU     P3.1            ; CTS input port
;
UARTRAM     DSEG    SADDR
SOSHIN:     DS      1               ; Transmit data storage area
JUSHIN:     DS      1               ; Receive data storage area
;
UARTFLG     BSEG
F_TUSHIN    DBIT                    ; Reception end flag
F_BUSY      DBIT                    ; Communication busy flag
F_ERR       DBIT                    ; Reception error flag
;
;*********************************
;        Transmission routine
;*********************************
UARTPR0     CSEG
S_SOSHIN:
    if_bit(STIF)                    ; Previous transmission end?
        if_bit(!CTS_I)              ;   Enables transmission?
            CLR1    STIF            ;     Clears transmit end interrupt request flag
            TXS=SOSHIN (A)          ;     Stores transmit data
            CLR1    F_BUSY          ;     Clears transmit busy flag
        else                        ;
            SET1    F_BUSY          ;   Disables transmission → sets transmission busy flag
        endif                       ;
    else                            ;
        SET1    F_BUSY              ;
    endif                           ;
    RET
;*********************************
;        Reception end routine
;*********************************
INTSR:                              ;
    SEL     RB3                     ; RTS ← H
    SET1    RTS_0                   ; Loads receive data
    JUSHIN=RXB (A)                  ; Sets reception end flag
    SET1    F_TUSHIN                ;
    RETI
;*********************************
;       Reception error routine
;*********************************
INTSER:                             ; Selects bank 3
    SEL     RB3                     ; Reads error data
    A=RXB                           ; Sets receive error flag
    SET1    F_ERR                   ;
    RETI
    END
```

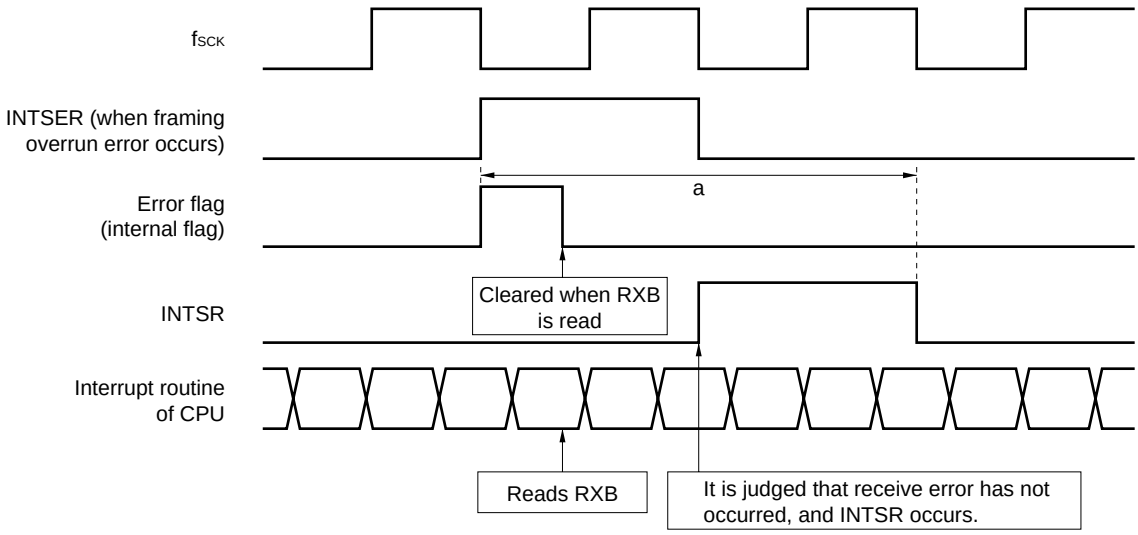
(f) Limitation when using UART mode

In the UART mode, the reception completion interrupt (INTSR) occurs a certain time after the reception error interrupt (INTSER) has occurred and cleared. As a result, the following phenomenon may take place.

- **Description**

  If bit 1 (ISRM) of the asynchronous serial interface mode register (ASIM) is set to 1, the reception completion interrupt (INTSR) does not occur when a reception error occurs. If the receive buffer register (RXB) is read at certain timing (a in Figure 8-42) during reception error interrupt (INTSER) processing, the internal error flag is cleared to 0. Therefore, it is judged that a reception error has not occurred, and INTSR, which should not occur, occurs. Figure 8-42 illustrates this operation.

Figure 8-42. Timing of Reception Completion Interrupt (when ISRM = 1)



Remark ISRM : Bit 1 of asynchronous serial interface mode register (ASIM)
f_{SCK} : Source clock of 5-bit counter of baud rate generator
RXB : Receive buffer register

To prevent this phenomenon, take the following measures:

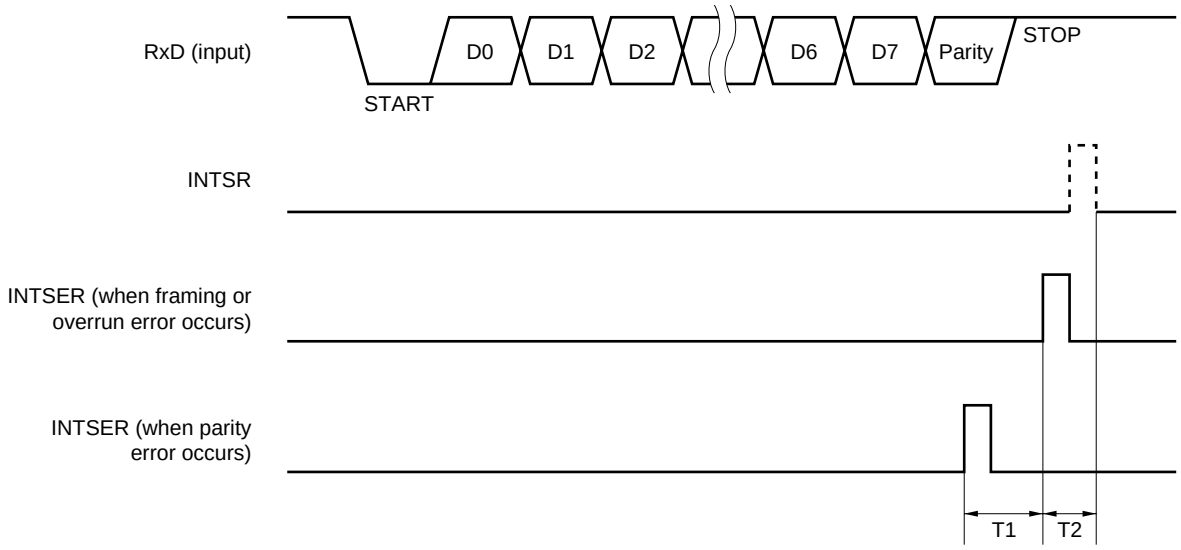
- **Preventive measures**
  - In case of framing error or overrun error

    Disable the receive buffer register (RXB) from being read for a certain period (T2 in Figure 8-43) after the receive error interrupt (INTSER) has occurred.

  - In case of parity error

    Disable the receive buffer register (RXB) from being read for a certain period (T1 + T2 in Figure 8-43) after the reception error interrupt (INTSER) has occurred.

Figure 8-43. Receive Buffer Register Reading Disabled Period



T1: Time of one data of baud rate selected by baud rate generator control register (BRGC) (1/baud rate)

T2: Time of two source clocks (f_{SCK}) of 5-bit counter selected by BRGC

- **Example of preventive measures**

  An example of preventive measures is shown below.

  [Condition]

  f_X = 5.0 MHz

  Processor clock control register (PCC) = 00H

  Oscillation mode select register (OSMS) = 01H

  Baud rate generator control register (BRGC) = 80H (2400 bps is selected as baud rate)

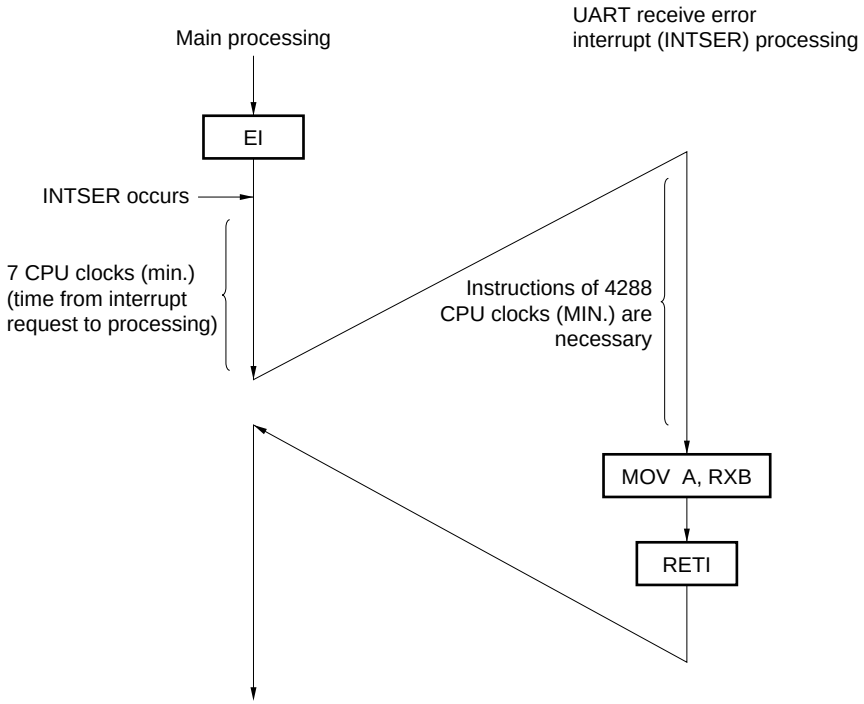
  t_{CY} = 0.4 μs (t_{CY} = 0.2 μs)

  $$T1 = \frac{1}{2400} = 833.4\ \mu s$$

  $$T2 = 12.8 \times 2 = 25.6\ \mu s$$

  $$\frac{T1 + T2}{t_{CY}} = 4295 \text{ (clocks)}$$

[Example]


Main processing
UART receive error
interrupt (INTSER) processing
EI
INTSER occurs
7 CPU clocks (min.)
(time from interrupt
request to processing)
Instructions of 4288
CPU clocks (MIN.) are
necessary
MOV A, RXB
RETI

[MEMO]

CHAPTER 9 APPLICATIONS OF A/D CONVERTER

The A/D converter of the 78K/0 series is a successive approximation type with an 8-bit resolution and eight channels. Although only a select mode is supported as the operation mode, conversion can be started by an external trigger. If the external trigger is not used, the analog data of a selected channel is repeatedly converted into a digital signal.

The A/D converter is set by the A/D converter mode register (ADM), A/D converter input select register (ADIS), external interrupt mode register 1 (INTM1), and A/D current cut select register (IEAD).

Caution IEAD is provided only to the μPD78098 and 78098B subseries.

Figure 9-1. Format of A/D Converter Mode Register
(μPD78054, 78054Y, 78064, 78064Y, 78078, 78078Y, 78083, 780058, 780058Y, 780308, 780308Y, 78058F, 78058FY, 78064B, 78075B, 78075BY subseries, μPD78070A, 78070AY)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
ADM	CS	TRG	FR1	FR0	ADM3	ADM2	ADM1	HSC	FF80H	01H	R/W

ADM3	ADM2	ADM1	Selects analog input channel
0	0	0	ANI0
0	0	1	ANI1
0	1	0	ANI2
0	1	1	ANI3
1	0	0	ANI4
1	0	1	ANI5
1	1	0	ANI6
1	1	1	ANI7

FR1	FR0	HSC	Selects A/D conversion time[Note 1]			
			At f_X = 5.0 MHz		At f_X = 4.19 MHz	
			MCS = 1	MCS = 0	MCS = 1	MCS = 0
0	0	1	80/f_X (setting prohibited)[Note 2]	160/f_X (32.0 μs)	80/f_X (19.1 μs)	160/f_X (38.1 μs)
0	1	1	40/f_X (setting prohibited)[Note 2]	80/f_X (setting prohibited)[Note 2]	40/f_X (setting prohibited)[Note 2]	80/f_X (19.1 μs)
1	0	0	50/f_X (setting prohibited)[Note 2]	100/f_X (20.0 μs)	50/f_X (setting prohibited)[Note 2]	100/f_X (23.8 μs)
1	0	1	100/f_X (20.0 μs)	200/fx (40.0 μs)	100/f_X (23.8 μs)	200/f_X (47.7 μs)
Others			Setting prohibited			

TRG	Selects external trigger
0	No external trigger (software start)
1	Conversion started by external trigger (hardware start)

CS	Controls A/D conversion operation
0	Stops operation
1	Starts operation

Notes 1. Set the A/D conversion time to 19.1 μs or longer.

2. These settings are prohibited because the A/D conversion time is less than 19.1 μs.

Cautions 1. To reduce the power consumption of the A/D converter when the standby function is used, stop the A/D conversion operation by clearing bit 7 (CS) to 0, and then execute the HALT or STOP instruction.

2. To resume the A/D conversion operation which has been once stopped, clear the interrupt request flag (ADIF) to 0 and then start the A/D conversion operation.

Remarks 1. f_X : main system clock oscillation frequency

2. MCS : bit 0 of the oscillation mode select register (OSMS)

Figure 9-2. Format of A/D Converter Mode Register (μPD78098, 78098B subseries)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
ADM	CS	TRG	FR1	FR0	ADM3	ADM2	ADM1	HSC	FF80H	01H	R/W

ADM3	ADM2	ADM1	Selects analog input channel
0	0	0	ANI0
0	0	1	ANI1
0	1	0	ANI2
0	1	1	ANI3
1	0	0	ANI4
1	0	1	ANI5
1	1	0	ANI6
1	1	1	ANI7

FR1	FR0	HSC	Selects A/D conversion time[Note 1]
0	0	1	80/f_{XX} (20.0 μs)
0	1	1	40/f_{XX} (setting prohibited)[Note 2]
1	0	0	50/f_{XX} (setting prohibited)[Note 2]
1	0	1	100/f_{XX} (25.0 μs)
Others			Setting prohibited

TRG	Selects external trigger
0	No external trigger (software start)
1	Conversion started by external trigger (hardware start)

CS	Controls A/D conversion operation
0	Stops operation
1	Starts operation

Notes 1. Set the A/D conversion time to 19.1 μs or longer.

2. These settings are prohibited because the A/D conversion time is less than 19.1 μs.

Cautions 1. To reduce the power consumption of the A/D converter when the standby function is used, stop the A/D conversion operation by clearing bit 7 (CS) to 0, and then execute the HALT or STOP instruction.

2. To resume the A/D conversion operation which has been once stopped, clear the interrupt request flag (ADIF) to 0 and then start the A/D conversion operation.

Remarks 1. f_{XX} : main system clock frequency

2. ( ) : f_{XX} = 4.0 MHz

★ **Figure 9-3. Format of A/D Converter Mode Register (μPD780018, 780018Y subseries)**

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
ADM	CS	TRG	FR1	FR0	ADM3	ADM2	ADM1	HSC	FF80H	01H	R/W

ADM3	ADM2	ADM1	Selects analog input channel
0	0	0	ANI0
0	0	1	ANI1
0	1	0	ANI2
0	1	1	ANI3
1	0	0	ANI4
1	0	1	ANI5
1	1	0	ANI6
1	1	1	ANI7

FR1	FR0	HSC	Selects A/D conversion time[Note 1]	
			At f_X = 5.0 MHz	At f_X = 4.19 MHz
0	0	1	80/f_X (setting prohibited)[Note 2]	80/f_X (19.1 μs)
0	1	1	40/f_X (setting prohibited)[Note 2]	40/f_X (setting prohibited)[Note 2]
1	0	0	50/f_X (setting prohibited)[Note 2]	50/f_X (setting prohibited)[Note 2]
1	0	1	100/f_X (20.0 μs)	
Others			Setting prohibited	

TRG	Selects external trigger
0	No external trigger (software start)
1	Conversion started by external trigger (hardware start)

CS	Controls A/D conversion operation
0	Stops operation
1	Starts operation

Notes 1. Set the A/D conversion time to 19.1 μs or longer.

2. These settings are prohibited because the A/D conversion time is less than 19.1 μs.

Cautions 1. To reduce the power consumption of the A/D converter when the standby function is used, stop the A/D conversion operation by clearing bit 7 (CS) to 0, and then execute the HALT or STOP instruction.

2. To resume the A/D conversion operation which has been once stopped, clear the interrupt request flag (ADIF) to 0 and then start the A/D conversion operation.

Remark f_X : main system clock oscillation frequency

Figure 9-4. Format of A/D Converter Input Select Register

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
ADIS	0	0	0	0	ADIS3	ADIS2	ADIS1	ADIS0	FF84H	00H	R/W

ADIS3	ADIS2	ADIS1	ADIS0	Selects number of analog input channels
0	0	0	0	No analog input channel (P10-P17)
0	0	0	1	1 channel (ANl0, P11-P17)
0	0	1	0	2 channels (ANl0, ANl1, P12-P17)
0	0	1	1	3 channels (ANl0-ANl2, P13-P17)
0	1	0	0	4 channels (ANl0-ANl3, P14-P17)
0	1	0	1	5 channels (ANl0-ANI4, P15-P17)
0	1	1	0	6 channels (ANl0-ANI5, P16-P17)
0	1	1	1	7 channels (ANI0-ANl6, P17)
1	0	0	0	8 channels (ANI0-ANI7)
Others				Setting prohibited

Cautions 1. Set analog input channels in the following steps:

<1> Set the number of analog input channels by using ADIS.

<2> Select one channel whose data is to be converted, from the channels selected by ADIS, by using the A/D converter mode register (ADM).

2. The internal pull-up resistor is not used to the channel selected by ADIS as an analog input channel, regardless of the value of the bit 1 (PUO1) of the pull-up resistor option register L (PUOL).

Figure 9-5. Format of External Interrupt Mode Register 1 (μPD78054, 78054Y, 78078, 78078Y, 78098, 780018, 780018Y, 78058F, 78058FY, 78075B, 78075BY, 78098B subseries, μPD78070A, 78070AY)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
INTM1	ES71	ES70	ES61	ES60	ES51	ES50	ES41	ES40	FFEDH	00H	R/W

ES41	ES40	Selects valid edge of INTP3
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES51	ES50	Selects valid edge of INTP4
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES61	ES60	Selects valid edge of INTP5
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES71	ES70	Selects valid edge of INTP6
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

Figure 9-6. Format of External Interrupt Mode Register 1
(μPD78064, 78064Y, 780058, 780058Y, 780308, 780308Y, 78064B subseries)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
INTM1	0	0	ES61	ES60	ES51	ES50	ES41	ES40	FFEDH	00H	R/W

ES41	ES40	Selects valid edge of INTP3
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES51	ES50	Selects valid edge of INTP4
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES61	ES60	Selects valid edge of INTP5
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

Figure 9-7. Format of External Interrupt Mode Register 1 (μPD78083 subseries)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
INTM1	0	0	0	0	0	0	ES41	ES40	FFEDH	00H	R/W

ES41	ES40	Selects valid edge of INTP3
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

Caution Be sure to clear bits 2 through 7 to 0.

Figure 9-8. Format of A/D Current Cut Select Register (μPD78098, 78098B subseries)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
IEAD	0	0	0	0	0	0	0	IEAD0	F8E2H	00H	R/W

IEAD0	Controls connection between AV_{DD} and AV_{REF0}
0	Disconnects AV_{DD} from AV_{REF0}
1	Connects AV_{DD} and AV_{REF0}

9.1 Level Meter

In this application example, the analog voltage input to the A/D converter is displayed on an LED matrix consisting of 4 × 4, i.e., 16 LEDs.

Because a level meter has been included in this example, the LED display is given in decibel units. Figure 9-9 shows the circuit of the level meter, and Figure 9-10 shows the relations between the result of the A/D conversion and the number of display digits.

Figure 9-9. Example of Level Meter Circuit

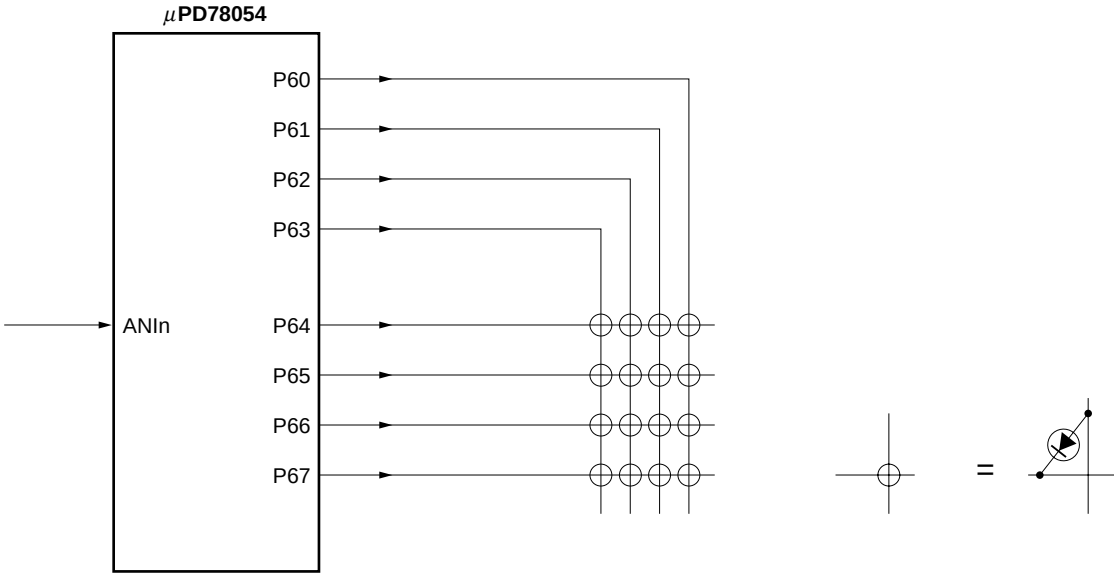
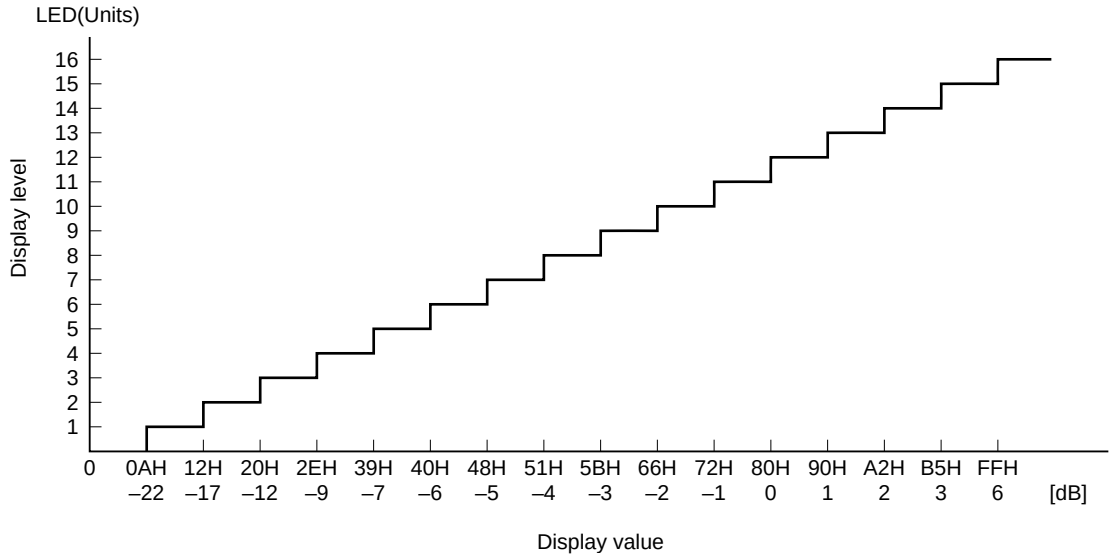


Figure 9-10. A/D Conversion Result and Display



The level meter in this example operates with specifications <1> through <3> below.

<1> Measurement method

A/D conversion is performed every 20 ms, and the average value of four previous data is calculated and displayed on the LEDs.

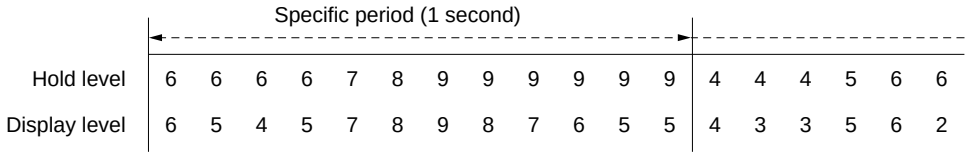
<2> Display method

The LED display is updated every 20 ms. The LED matrix consists of $4 \times 4 = 16$ LEDs and performs dynamic display. For the dynamic display, 8-bit timer/event counter 1 (interval time: 2 ms) is used.

<3> Peak hold

Holding the maximum display level for a specific period (1 second) is called peak hold. Even if the display level drops during a specific period, only the LED at the maximum display level is held. Therefore, the hold period of the hold level is 20 ms to 1 s.

Figure 9-11. Concept of Peak Hold



(1) Description of package

<Public declaration symbol>

LEVEL : Name of LED display subroutine
DSPLEV : Display level storage area
HLDLEV : Hold level storage area
CT20MS : Counter measuring 20 ms
CT1S : Counter measuring 1 s

<Register used>

AX, HL, BC (subroutine processing)
Bank 0: A, HL, B (interrupt processing)

<RAM used>

Name	Usage	Attribute	Bytes
ADDAT	Stores A/D conversion value	SADDR	4
DSPLEV	Stores display level		1
HLDLEV	Stores hold level		
CT20MS	Counter measuring 20 ms		
CT1S	Counter measuring 1 s		
DIGCNT	Display digit counter		
DSPDAT	Stores display data		4
WORKCT	Work counter for loop processing		1

<Flag used>

Name	Usage
T20MSF	Set every 20 ms
T1SF	Set every 1 s

<Nesting>

2 levels 5 bytes

<Hardware used>

- A/D converter
- 8-bit timer/event counter 1
- P6

<Initial setting>

- OSMS = #00000001B ; Oscillation mode select register: does not use divider circuit
- ADM = #1000×××1B ; Selects channel of A/D converter and starts operation
- TCL1 = #10111011B ; Interval time of 2 ms of 8-bit timer/event counter 1
  TMC1 = #00000001B
  CR10 = 130
- P6 output mode
- Makes P6 output latch low
- Enables INTTM1 interrupt

<Starting>

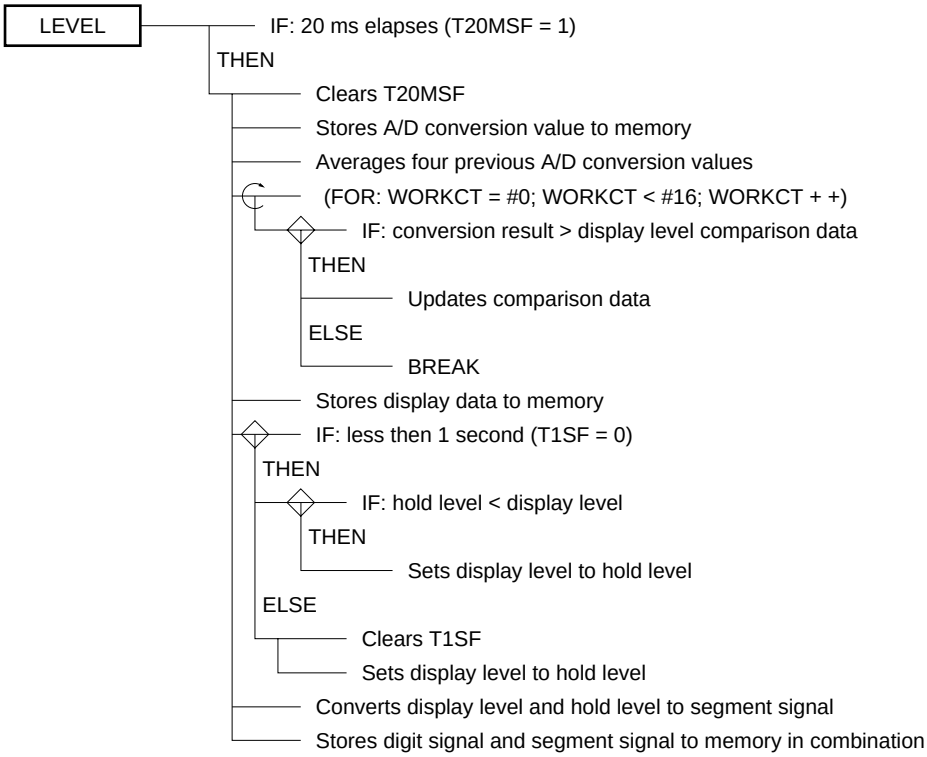
This program performs two types of processing: A/D conversion (subroutine) and LED display (interrupt).

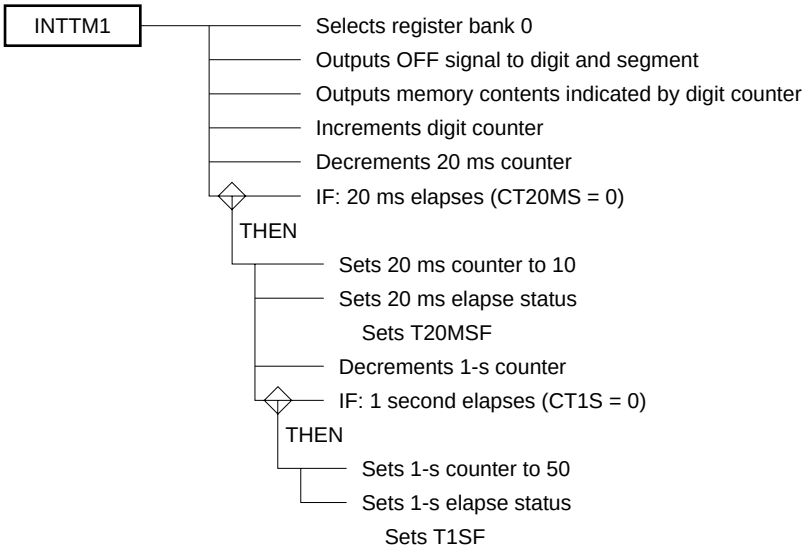
- A/D conversion processing
  Call LEVEL at least once every 20 ms from the main processing. The LEVEL processing performs A/D conversion processing only when 20 has elapsed.
- LED display
  The 4 × 4 LED matrix performs dynamic display by using the interrupt processing of 8-bit timer/event counter 1 (interval: 2 ms). The interrupt processing of 8-bit timer/event counter 1 sets the T20MSF (loading of A/D conversion value) and T1SF (end of hold period) used for the A/D conversion processing at an interval of 2 ms.

(2) Example of use

```
EXTRN    LEVEL,CT20MS,CT1S

MOV      CT20MS,#10
MOV      CT1S,#50
MOV      TMC2,#00100110B
CLR1     TMMK3
                    ; Turns OFF LED display
P6=#00H
PM6=#00000000B      ; Does not use divider circuit
OSMS=#00000001B     ; ANI0 pin starts operation
ADM=#10000001B      ; Sets 8-bit timer/event counter 1 to 2 ms
TCL1=#10111011B

CR10=#130
TMC1=#00000001B     ; Enables 8-bit timer/event counter 1 interrupt
CLR1     TMMK1
EI
```

(3) SPD chart


LEVEL
IF: 20 ms elapses (T20MSF = 1)
THEN
Clears T20MSF
Stores A/D conversion value to memory
Averages four previous A/D conversion values
(FOR: WORKCT = #0; WORKCT < #16; WORKCT + +)
IF: conversion result > display level comparison data
THEN
Updates comparison data
ELSE
BREAK
Stores display data to memory
IF: less then 1 second (T1SF = 0)
THEN
IF: hold level < display level
THEN
Sets display level to hold level
ELSE
Clears T1SF
Sets display level to hold level
Converts display level and hold level to segment signal
Stores digit signal and segment signal to memory in combination


INTTM1
Selects register bank 0
Outputs OFF signal to digit and segment
Outputs memory contents indicated by digit counter
Increments digit counter
Decrements 20 ms counter
IF: 20 ms elapses (CT20MS = 0)
THEN
Sets 20 ms counter to 10
Sets 20 ms elapse status
Sets T20MSF
Decrements 1-s counter
IF: 1 second elapses (CT1S = 0)
THEN
Sets 1-s counter to 50
Sets 1-s elapse status
Sets T1SF

(4) Program list

```
        PUBLIC  LEVEL,HLDLEV,DSPLEV,CT20MS,CT1S

AD_DAT  DSEG    SADDR
ADDAT:  DS      4                               ; A/D conversion result storage area
DSPLEV: DS      1                               ; Display level value
HLDLEV: DS      1                               ; Hold level value
CT20MS: DS      1                               ; 20 ms counter
CT1S:   DS      1                               ; 1 s counter
DIGCNT: DS      1                               ; Display digit counter
DSPDAT: DS      4                               ; Display data
WORKCT: DS      1

AD_FLG  BSEG
T20MSF  DBIT                                    ; Measures 20 ms
T1SF    DBIT                                    ; Measures 1 s

VETM1   CSEG    AT 24H
        DW      INTTM1                          ; Sets vector address of 8-bit timer/event counter 1

AD_SEG  CSEG
;*******************************
*      Sets level meter data
;*******************************
LEVEL:
        IF_BIT(T20MSF)                          ; Checks 20 ms
            CLR1    T20MSF
            A=ADCR                              ; Inputs A/D conversion value
            A<->ADDAT                           ; Stores A/D conversion value
            A<->ADDAT+1
            A<->ADDAT+2
            A<->ADDAT+3
                                                ; Averages four A/D conversion values
            AX=#0H
            HL=#ADDAT                           ; Data storage address
            for(WORKCT=#0;WORKCT<#4;WORKCT++)
                A+=[HL]
                HL++
                if_bit(CY)                      ; Carry
                    X++                         ; Higher digit
                endif
            next

            A<->X
            C=#4                                ; Averages four values
            AX/=C                               ; AX/C = AX (quotient) ... C (remainder)
            if(C>=#2) (A)                       ; Remainder processing (2 or higher is carried)
                X++                             ; Carry processing
            endif

            HL=#LEVTBL
            B=#0                                ; Conversion result storage register
            for(WORKCT=#0;WORKCT<#16;WORKCT++)
                if(X>=[HL+B]) (A)               ; Compares data
                    B++
                else
                    break
                endif
            next
```

```
    DSPLEV=B (A)                    ; Determines display data

    if_bit(!T1SF)                   ; 1 s (hold level updated)
        X=HLDLEV (A)                ; Compares hold and display levels
        if(X<DSPLEV) (A)
            HLDLEV=DSPLEV (A)
        endif
    else
        CLR1    T1SF
        HLDLEV=DSPLEV (A)
    endif

    HL=#DSPTBL
    A=DSPLEV                        ; Creates display level
    A+=A
    B=A
    A=HLDLEV
    A+=A
    C=A
    X=[HL+B] (A)
    B++
    A=[HL+B]
    HL=#HLDTBL                      ; Creates hold level
    A<->X
    A|=[HL+C]
    A<->X
    C++
    A|=[HL+C]
    BC=AX

    HL=#DSPDAT                      ; Sets segment signal of first digit
    A=C
    A&=#0FH
    A|=#00010000B                   ; Sets digit signal
    [HL]=A
    HL++
    A=C                             ; Sets segment signal of second digit
    A>>=1
    A>>=1
    A>>=1
    A>>=1
    A&=#0FH
    A|=#00100000B                   ; Sets digit signal
    [HL]=A
    HL++
    A=B                             ; Sets segment signal of third digit
    A&=#0FH
    A|=#01000000B                   ; Sets digit signal
    [HL]=A
    HL++
    A=B                             ; Sets segment signal of fourth digit
    A>>=1
    A>>=1
    A>>=1
    A>>=1
    A&=#0FH                         ; Sets digit signal
    A|=#10000000B
    [HL]=A
endif
```

```
        RET
LEVTBL:
        DB      0AH
        DB      12H
        DB      20H
        DB      2EH
        DB      39H
        DB      40H
        DB      48H
        DB      51H
        DB      5BH
        DB      66H
        DB      72H
        DB      80H
        DB      90H
        DB      0A2H
        DB      0B5H
        DB      0FFH

DSPTBL:
        DW      0000000000000000B
        DW      0000000000000001B
        DW      0000000000000011B
        DW      0000000000000111B
        DW      0000000000001111B
        DW      0000000000011111B
        DW      0000000000111111B
        DW      0000000001111111B
        DW      0000000011111111B
        DW      0000000111111111B
        DW      0000001111111111B
        DW      0000011111111111B
        DW      0000111111111111B
        DW      0001111111111111B
        DW      0011111111111111B
        DW      0111111111111111B
        DW      1111111111111111B

HLDTBL:
        DW      0000000000000000B
        DW      0000000000000001B
        DW      0000000000000010B
        DW      0000000000000100B
        DW      0000000000001000B
        DW      0000000000010000B
        DW      0000000000100000B
        DW      0000000001000000B
        DW      0000000010000000B
        DW      0000000100000000B
        DW      0000001000000000B
        DW      0000010000000000B
        DW      0000100000000000B
        DW      0001000000000000B
        DW      0010000000000000B
        DW      0100000000000000B
        DW      1000000000000000B
$EJECT
```

```
;*******************************
*        Level meter data
;*******************************
TM1_SEG CSEG
INTTM1:
        SEL RB0                     ; Turns OFF digit and segment signals
        P6=#00000000B
        HL=#DSPDAT
        B=DIGCNT (A)
        P6=[HL+B] (A)
        DIGCNT++
        DIGCNT&=#00000011B          ; 20 ms?
        CT20MS--
        if(CT20MS==#0)              ; Sets initial counter value
            CT20MS=#10
            SET1    T20MSF          ; 1s?
            CT1S--
            if(CT1S==#0)            ; Sets initial counter value
                CT1S=#50
                SET1    T1SF
            endif
        endif
        RETI
```

9.2 Thermometer

In this application example, a temperature in a range of –20°C to +50°C is measured by using a thermistor (6 kΩ/0°C) as a temperature sensor. Changes in the resistance of the thermistor with respect to temperature are given by the following expression:

$$R = R_0 \exp \{ B (1/T - 1/T_0) \}$$

where,

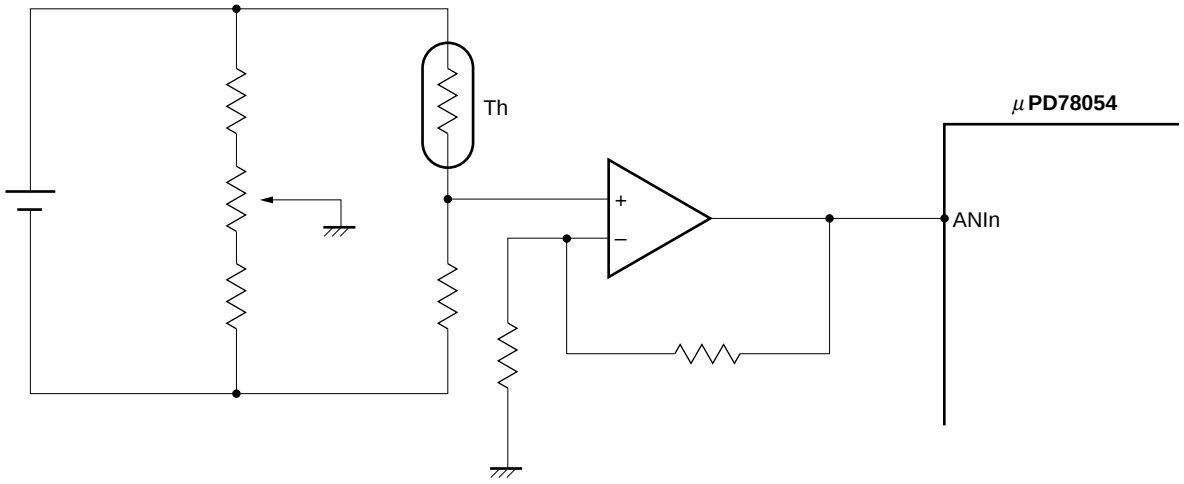
- R : resistance at given temperature T [°K]
- T : given temperature [°K]
- R_0 : resistance at reference temperature T_0 [°K]
- T_0 : reference temperature [°K]
- B : constant obtained by reference temperature T_0 [°K] and T_0 [°K]

Constant B changes with the temperature. This constant can be calculated by changing the above expression as follows:

$$B = \frac{1}{(1/T - 1/T_0)} \ln \frac{R}{R_0}$$

Figure 9-12 shows a circuit example. This circuit is designed to input 0 V at –20°C, and 5 V at + 50°C.

Figure 9-12. Circuit Example of Thermometer



Because the characteristic of the thermistor is non linear in this example, the input analog voltage is not converted to a temperature in a range of –20 °C to +50 °C through calculation but by comparison with table data. This conversion result is stored to RAM (DSPDAT) as 2-digit BCD. Figure 9-13 shows the characteristics of the thermistor, and Table 9-1 shows the relations between temperature and A/D conversion value.

To measure the temperature, four conversion values are averaged and converted to a temperature. The result of the conversion is stored in a display area. Therefore, the data is updated once every four times. For example, if measurement processing is executed every 250 ms, the display updating cycle is 1 second.

Figure 9-13. Temperature vs. Output Characteristic

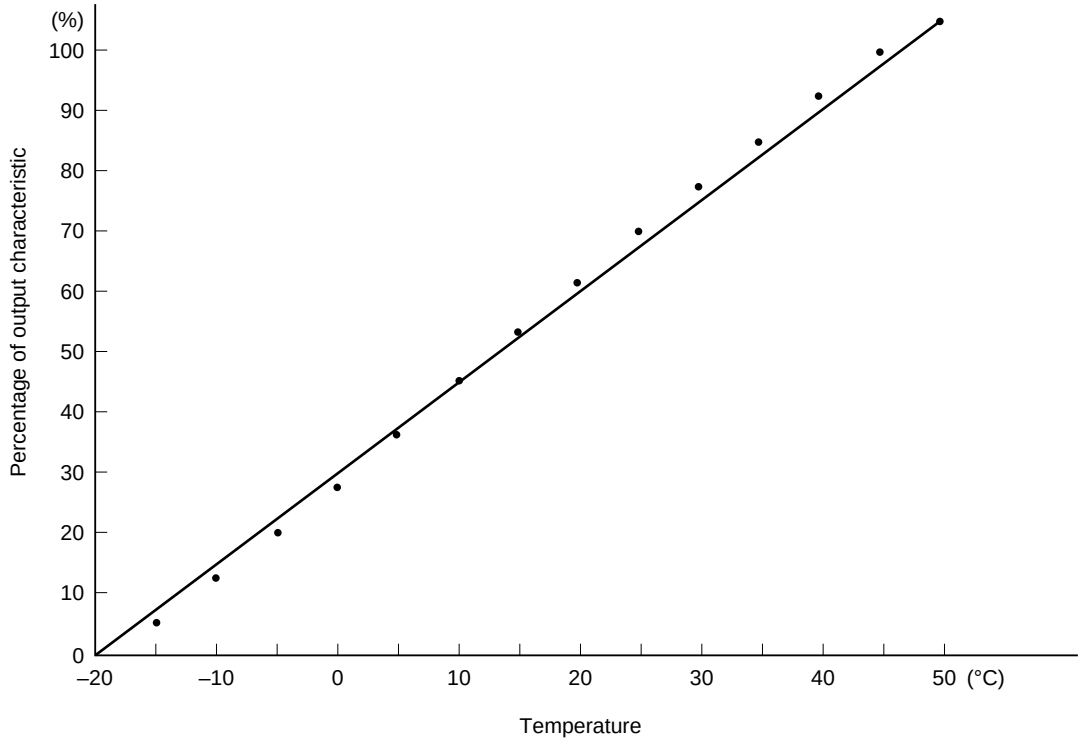


Table 9-1. A/D Conversion Value and Temperature

Conversion Value	Temperature [°C]	Conversion Value	Temperature [°C]	Conversion Value	Temperature [°C]	Conversion Value	Temperature [°C]
00	–20.0	38	–2.5	82	15.5	CB	33.5
01	–19.5	3C	–1.5	86	16.5	CE	34.5
04	–18.5	40	–0.5	8B	17.5	D2	35.5
07	–17.5	44	0.5	8F	18.5	D6	36.5
0A	–16.5	48	1.5	93	19.5	D9	37.5
0C	–15.5	4C	2.5	97	20.5	DC	38.5
0F	–14.5	50	3.5	9B	21.5	E0	39.5
12	–13.5	54	4.5	9F	22.5	E3	40.5
16	–12.5	58	5.5	A3	23.5	E7	41.5
19	–11.5	5C	6.5	A8	24.5	EA	42.5
1C	–10.5	60	7.5	AC	25.5	ED	43.5
1F	–9.5	64	8.5	B0	26.5	F0	44.5
23	–8.5	69	9.5	B4	27.5	F3	45.5
26	–7.5	6D	10.5	B7	28.5	F6	46.5
2A	–6.5	71	11.5	BB	29.5	F9	47.5
2D	–5.5	75	12.5	BF	30.5	FC	48.5
31	–4.5	7A	13.5	C3	31.5	FE	49.5
35	–3.5	7E	14.5	C7	32.5	FF	50.0

(1) Description of package

<Public declaration symbol>

THMETER : Thermometer subroutine call name
DSPDAT : Display data storage area
CNTPRO : Test counter counting number of inputs
MINUSF : Minus temperature display flag
T250MSF : 250-ms setting flag

<Register used>

AX, BC, HL

<RAM used>

Name	Usage	Attribute	Bytes
ADDAT	Stores A/D conversion value	SADDR	4
DSPDAT	Stores display data		2
CNTPRO	Test counter for number of inputs		1
WORKCT	Work counter for loop processing		

<Flag used>

Name	Usage
T250MSF	Executes measurement processing when set
MINUSF	Set when temperature is below zero

<Nesting>

1 level 2 bytes

<Hardware used>

A/D converter

<Initial setting>

ADM = #1000×××1B; Selects A/D converter channel and starts operation

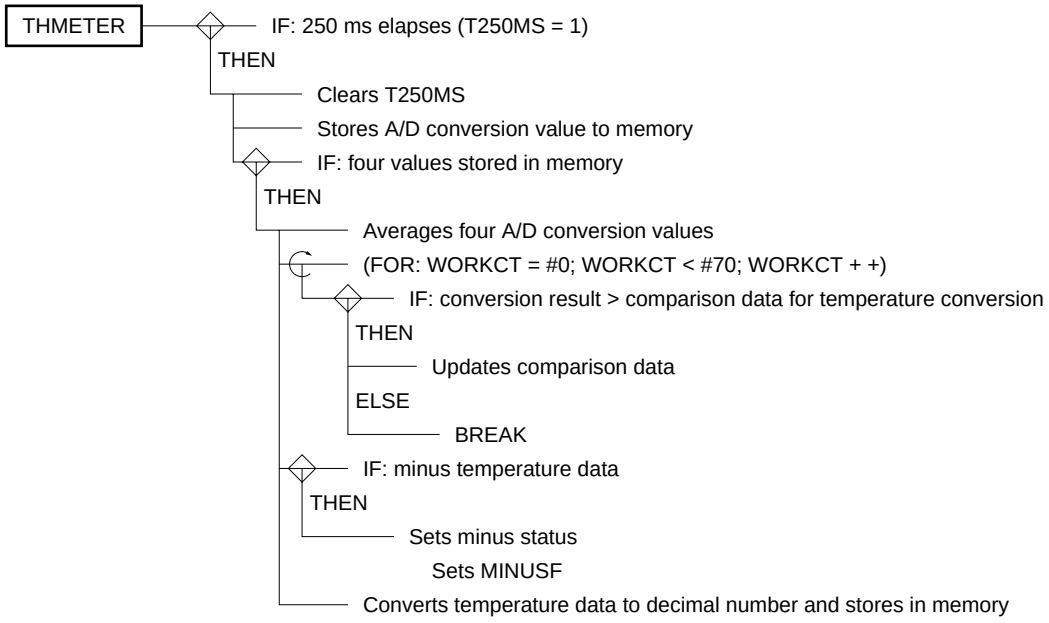
<Starting>

Set the T250MSF flag in each measurement cycle by using timer processing. After that, call THMETER at least once in measurement cycle.

(2) Example of use

```
          EXTRN   THMETER,DSPDAT,CNTPRO
          EXTBIT  MINUSF,T250MSF

AD_DAT    DSEG    SADDR
CT250MS:  DS      1                               ; 250 ms counter
LEDD:     DS      4                               ; LED display area
DIGCT:    DS      1                               ; LED display digit counter

VETM3     CSEG    AT 1EH
          DW      INTTM3                          ; Sets vector address of watch timer

          MOV     TMC2,#00100110B                 ; Sets watch timer to 1.95 ms
          CLR1    TMMK3
            .
            .
            .
          CT250MS=#128
          CNTPR0=#4
          ADM=#10000011B                          ; Selects ANI1 pin and starts operation
            .
            .
            .

;*****************************************
;       Watch timer interrupt processing
;       Interval time:  1.95 ms
;*****************************************
INTTM3:                                           ; 1.95 ms interrupt processing
            .
            .
            .
          DBNZ    CT250MS,$RTNTM3
          MOV     CT250MS,#128                    ; 250 ms elapses
          SET1    T250MSF
RTNTM3:
            .
            .
            .
          RETI
```

(3) SPD chart


THMETER
IF: 250 ms elapses (T250MS = 1)
THEN
Clears T250MS
Stores A/D conversion value to memory
IF: four values stored in memory
THEN
Averages four A/D conversion values
(FOR: WORKCT = #0; WORKCT < #70; WORKCT + +)
IF: conversion result > comparison data for temperature conversion
THEN
Updates comparison data
ELSE
BREAK
IF: minus temperature data
THEN
Sets minus status
Sets MINUSF
Converts temperature data to decimal number and stores in memory

(4) Program list

```
        PUBLIC  THMETER,DSPDAT,CNTPR0,T250MSF,MINUSF

AD_DAT  DSEG    SADDR
ADDAT:  DS      4                               ; A/D conversion result storage area
DSPDAT: DS      2                               ; Display data
CNTPR0: DS      1                               ; Tests number of inputs
WORKCT: DS      1

AD_FLG  BSEG
T250MSF DBIT                                    ; Sets 250 ms
MINUSF  DBIT                                    ; Sets minus data

TH_SEG  CSEG
;*******************************
*      Sets temperature data
;*******************************
THMETER:
        if_bit(T250MSF)                         ; 250 ms
            CLR1    T250MSF
            A=ADCR
            A<->ADDAT
            A<->ADDAT+1
            A<->ADDAT+2
            A<->ADDAT+3

            CNTPR0--
            if(CNTPR0==#0)
                CNTPR0=#4
                AX=#0H
                HL=#ADDAT                       ; Data storage address
                for(WORKCT=#0;WORKCT<#4;WORKCT++)
                    A+=[HL]
                    HL++
                    if_bit(CY)                  ; Carry occurs
                        X++                     ; Carry
                    endif
                next

                A<->X
                C=#4
                AX/=C                           ;  AX/C = AX (quotient) ... C (remainder)
                if(C>=#2) (A)                   ; Remainder processing (2 digits or more carried)
                    X++                         ; Carry processing
                endif

                A=X                             ; Converts to temperature data
                B=#0
                HL=#THRTBL
                if(A==#0FFH)
                    B=#70
                else
                    for(WORKCT=#0;WORKCT<#70;WORKCT++)
                        if(X>=[HL+B]) (A)
                            B++
                        else
                            break
                        endif
                    next
```

```
        endif

        CLR1    MINUSF
        A=#20                   ; Temperature data 20
        B-=A
        if_bit(CY)              ; To decimal conversion
            SET1    MINUSF
            A=#0
            A-=B                ; Absolute value of data
            A<->B
        endif
        X=#0                    ; Decimal conversion
        A=B
        A<->X
        C=#10
        AX/=C                   ; Temperature data/10
        DSPDAT=C (A)            ; Updates display data
        (DSPDAT+1)=X (A)
    endif
endif
RET
```

```
THRTBL;
;
        DB      1                       ; -19.5
        DB      4                       ; -18.5
        DB      7                       ; -17.5
        DB      0AH                     ; -16.5
        DB      0CH                     ; -15.5
        DB      0FH                     ; -14.5
        DB      12H                     ; -13.5
        DB      16H                     ; -12.5
        DB      19H                     ; -11.5
        DB      1CH                     ; -10.5
        DB      1FH                     ; -9.5
        DB      23H                     ; -8.5
        DB      26H                     ; -7.5
        DB      2AH                     ; -6.5
        DB      2DH                     ; -5.5
        DB      31H                     ; -4.5
        DB      35H                     ; -3.5
        DB      38H                     ; -2.5
        DB      3CH                     ; -1.5
        DB      40H                     ; -0.5
        DB      44H                     ; +0.5
        DB      48H                     ; 1.5
        DB      4CH                     ; 2.5
        DB      50H                     ; 3.5
        DB      54H                     ; 4.5
        DB      58H                     ; 5.5
        DB      5CH                     ; 6.5
        DB      60H                     ; 7.5
        DB      64H                     ; 8.5
        DB      69H                     ; 9.5
        DB      6DH                     ; 10.5
        DB      71H                     ; 11.5
        DB      75H                     ; 12.5
        DB      7AH                     ; 13.5
        DB      7EH                     ; 14.5
        DB      82H                     ; 15.5
        DB      86H                     ; 16.5
        DB      8BH                     ; 17.5
        DB      8FH                     ; 18.5
        DB      93H                     ; 19.5
        DB      97H                     ; 20.5
        DB      9BH                     ; 21.5
        DB      9FH                     ; 22.5
        DB      0A3H                    ; 23.5
        DB      0A8H                    ; 24.5
        DB      0ACH                    ; 25.5
        DB      0B0H                    ; 26.5
        DB      0B4H                    ; 27.5
        DB      0B7H                    ; 28.5
        DB      0BBH                    ; 29.5
        DB      0BFH                    ; 30.5
        DB      0C3H                    ; 31.5
        DB      0C7H                    ; 32.5
        DB      0CBH                    ; 33.5
        DB      0CEH                    ; 34.5
        DB      0D2H                    ; 35.5
        DB      0D6H                    ; 36.5
```

```
DB      0D9H                ; 37.5
DB      0DCH                ; 38.5
DB      0E0H                ; 39.5
DB      0E3H                ; 40.5
DB      0E7H                ; 41.5
DB      0EAH                ; 42.5
DB      0EDH                ; 43.5
DB      0F0H                ; 44.5
DB      0F3H                ; 45.5
DB      0F6H                ; 46.5
DB      0F9H                ; 47.5
DB      0FCH                ; 48.5
DB      0FFH                ; 49.5
```

9.3 Analog Key Input

In this example, sixteen keys are input by using the A/D converter. To input keys, a circuit must be designed so that a voltage peculiar to a key is input to the A/D converter when the key is pressed.

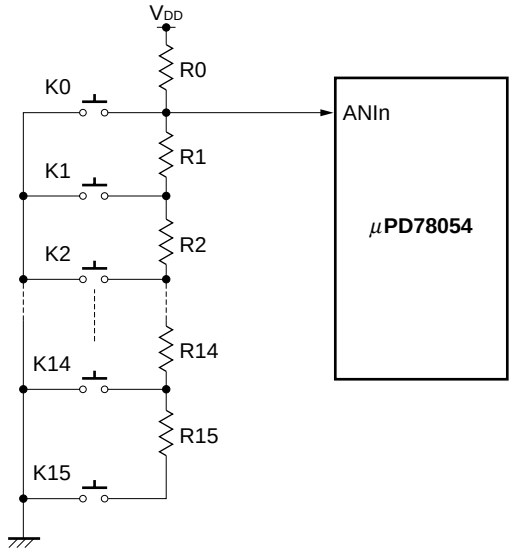
Because sixteen keys are input in this example, V_{DD} is divided by 16 and the voltage of each key is converted into a key code. Table 9-2 shows the relations between the input voltages and key codes (00H through 0FH). When no key input is made, the key code is 10H.

Table 9-2. Input Voltage and Key Code

Input Voltage V	A/D Conversion Value	Key Code
GND	00-07H	00H
$1/16V_{DD}$	08-17H	01H
$2/16V_{DD}$	18-27H	02H
$3/16V_{DD}$	28-37H	03H
$4/16V_{DD}$	38-47H	04H
$5/16V_{DD}$	48-57H	05H
$6/16V_{DD}$	58-67H	06H
$7/16V_{DD}$	68-77H	07H
$8/16V_{DD}$	78-87H	08H
$9/16V_{DD}$	88-97H	09H
$10/16V_{DD}$	98-A7H	0AH
$11/16V_{DD}$	A8-B7H	0BH
$12/16V_{DD}$	B8-C7H	0CH
$13/16V_{DD}$	C8-D7H	0DH
$14/16V_{DD}$	D8-E7H	0EH
$15/16V_{DD}$	E8-F7H	0FH
V_{DD}	F8-FFH	10H

Figure 9-14 shows an example of the circuit that satisfies the above relations between the input voltages and key codes. Note, however, that this circuit gives a priority to the key with the lower number if two or more keys are pressed at the same time.

Figure 9-14. Example of Analog Key Input Circuit



Resistances R0 through R15 used in the circuit in Figure 9-14 can be calculated by the following expression:

$$\sum_{K=1}^{n} R_K = \frac{n \times R0}{16-n}$$

Table 9-3 shows the resistances of R1 through R15 where R0 is 1 kΩ in the above expression (the calculation result of a resistance may slightly different from the resistance of commercial resistors indicated by a color code).

Table 9-3. Resistances of R1 through R5

Resistor No.	Resistance Value Ω	Resistor No.	Resistance Value Ω	Resistor No.	Resistance Value Ω
R1	68	R6	150	R11	560
R2	75	R7	180	R12	750
R3	82	R8	220	R13	1.3 k
R4	100	R9	270	R14	2.7 k
R5	120	R10	390	R15	8.2 k

This program converts an input analog voltage into the corresponding key code shown in Table 9-2, absorbs chattering, and then stores the input voltage to RAM. To absorb chattering, a key code is assumed to be valid when it coincides with a given value five times in succession. For example, if an analog voltage is sampled every 5 ms, chattering of 20 to 25 ms is absorbed. If a key input is changed, a key change flag (KEYCHG) is set.

(1) Description of package

<Public declaration symbol>

AKEYIN : Analog key input subroutine name
KEYDAT : Key code storage area
PASTDT : Key code storage area for chattering absorption
CHATCT : Chattering absorption counter
KEYCHG : Key change test flag
CHTENDF : Flag to test end of chattering absorption
KEYOFF : Key code when there is no key input

<Register used>

A

<RAM used>

Name	Usage	Attribute	Bytes
PASTDAT	Stores key code for chattering absorption	SADDR	1
KEYDAT	Stores key code		
CHATCNT	Chattering counter		

<Flag used>

Name	Usage
KEYCHG	Set when key is changed
CHTENDF	Sets when chattering absorption ends

<Nesting>

1 level 2 bytes

<Hardware used>

A/D converter

<Initial setting>

ADM = #1000×××1B; Selects A/D converter channel and starts operation

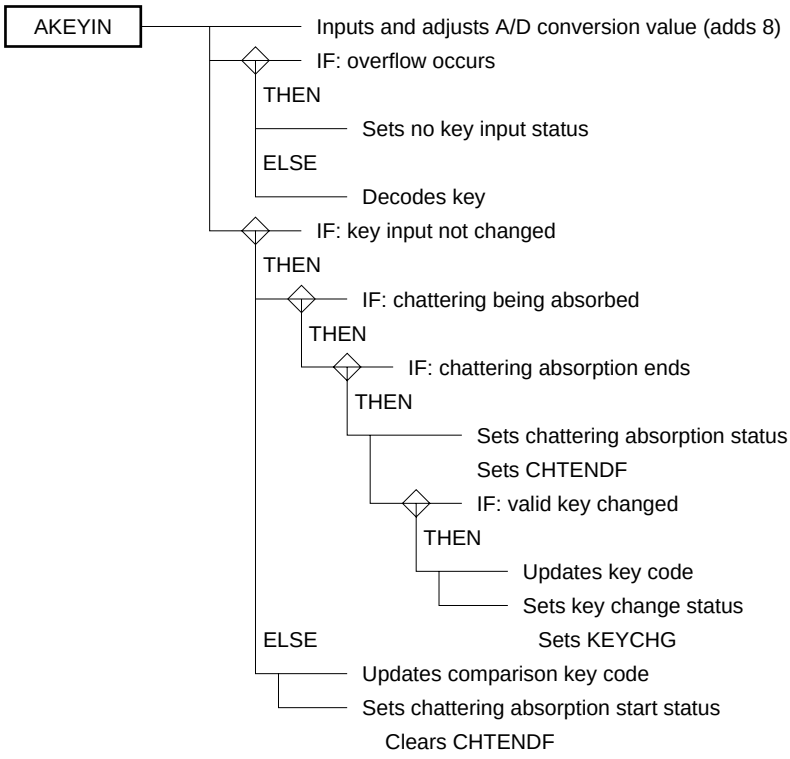
<Starting>

- Call AKEYIN at fixed interval.
- Input a key code after testing the key change flag. Note that this flag is not cleared by the subroutine and must be cleared after the flag has been tested.

(2) Example of use

```
          EXTRN   AKEYIN,KEYDAT,PASTDT,CHATCT
          EXTRN   KEYOFF

          EXTBIT  KEYCHG,CHTENDF

VETM3     CSEG    AT 1EH
          DW      INTTM3                    ; Sets vector address of watch timer

MAINDAT   DSEG    SADDR
CT5MS:    DS      1

          TMC2=#00100110B
          CLR1    TMMK3
          CT5MS=#3

          KEYDAT=#KEYOFF                    ; Sets OFF data as key data
          PASTDT=#KEYOFF
          CHATCT=#CHAVAL                    ; Sets number of times of chattering to five
          CLR1    CHTENDF
          CLR1    KEYCHG
          ADM=#10000101B                    ; Selects ANI2 pin and starts operation
          EI
            .
            .
            .
          if_bit(KEYCHG)                    ; Key changed?
                  CLR1   KEYCHG
                  ; Key input processing
          endif
            .
            .
            .
;*******************************************
;       Watch timer interrupt processing
;              Interval:  1.95 ms
;*******************************************
INTTM3:                                     ; 1.95 ms interrupt processing
            .
            .
            .
          DBNZ    CT5MS,$RTNTM3
          MOV     CT5MS,#3                  ; 1.95 ms × 3 elapses
          CALL    !AKEYIN
RTNTM3:
            .
            .
            .
          RETI
```

(3) SPD chart


AKEYIN
Inputs and adjusts A/D conversion value (adds 8)
IF: overflow occurs
THEN
Sets no key input status
ELSE
Decodes key
IF: key input not changed
THEN
IF: chattering being absorbed
THEN
IF: chattering absorption ends
THEN
Sets chattering absorption status
Sets CHTENDF
IF: valid key changed
THEN
Updates key code
Sets key change status
Sets KEYCHG
ELSE
Updates comparison key code
Sets chattering absorption start status
Clears CHTENDF

(4) Program list

```
        PUBLIC  AKEYIN,KEYDAT,PASTDT
        PUBLIC  CHATCT,KEYOFF
        PUBLIC  KEYCHG,CHTENDF
AK_DAT  DSEG    SADDR                   ; Key data storage area
KEYDAT: DS      1                       ; Chattering key data
PASTDT: DS      1                       ; Chattering counter
CHATCT: DS      1

AK_FLG  BSEG                            ; Key changed
KEYCHG  DBIT                            ; Chattering absorption end status
CHTENDF DBIT

                                        ; OFF key data
KEYOFF  EQU     10H                     ; Number of times of chattering absorption
CHAVAL  EQU     5

AK_SEG  CSEG
;*****************************
*       Analog key input
;*****************************
AKEYIN:                                 ; Inputs A/D conversion value
        A=ADCR                          ; Corrects data
        A+=#8
        if_bit(CY)                      ; Sets no key input status
            A=#KEYOFF
        else                            ; Decodes key
            A>>=1
            A>>=1
            A>>=1
            A>>=1
            A&=0FH
        endif                           ; No key change
        if(A==PASTDT)                   ; Chattering being absorbed
            if_bit(!CHTENDF)            ; End of chattering absorption
                CHATCT--
                if(CHATCT==#0)          ; Sets chattering absorption status
                    SET1    CHTENDF
                    A=PASTDT            ; Valid key changed
                    if(A!=KEYDAT)       ; Updates key data
                        KEYDAT=A        ; Sets key change status
                        SET1    KEYCHG
                    endif
                endif
            endif
        endif                           ; Updates previous key data
            PASTDT=A                    ; Starts chattering absorption
            CHATCT=#CHAVAL-1
            CLR1    CHTENDF
        endif
        RET
```

9.4 4-Channel Input A/D Conversion

This section describes the method to scan four channels for A/D conversion. The A/D conversion operation is started by the software.

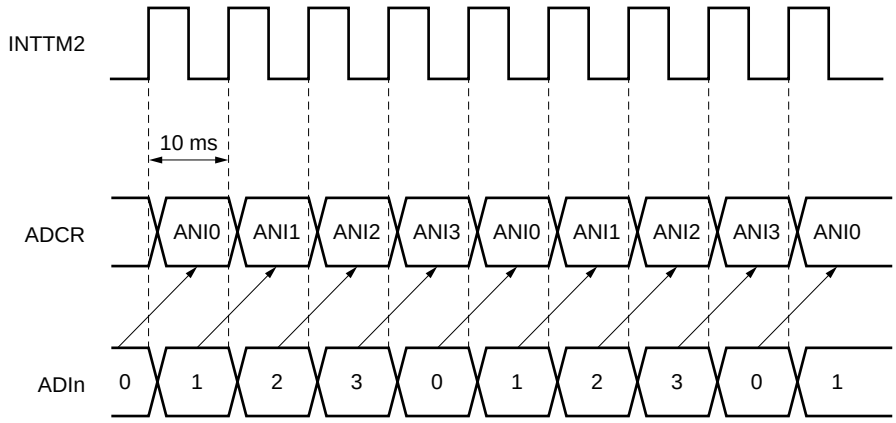
The analog voltages input to the selected four channels are converted into digital signals. The result of the A/D conversion of each channel is stored in RAM.

An interrupt request is generated by using 8-bit timer/event counter 1. The result of the conversion is loaded and channel is converted in the processing of this interrupt request. Because 8-bit timer/event counter 1 is set to 10 ms, it is not necessary to measure the wait time of the A/D conversion.

Caution To change the interrupt time, make the following setting:

- **Set timer longer than { A/D conversion end time + Interrupt entry return time + Interrupt processing time. }**
- **Test flags that indicate the end of the conversion.**

Figure 9-15. Timing Chart in 4-Channel Scan Mode



(1) Description of package

<Public declaration symbol>

- Output parameter

  M_CH0 : Stores conversion result of channel 0

  M_CH1 : Stores conversion result of channel 1

  M_CH2 : Stores conversion result of channel 2

  M_CH3 : Stores conversion result of channel 3

<Register used>

A

<RAM used>

Name	Usage	Attribute	Bytes
M_CH0	Channel 0 conversion result storage area	SADDR	1
M_CH1	Channel 1 conversion result storage area	SADDR	1
M_CH2	Channel 2 conversion result storage area	SADDR	1
M_CH3	Channel 3 conversion result storage area	SADDR	1
M_MODE	Mode storage area	SADDR	1

<Nesting>

1 level 3 bytes

<Hardware used>

- A/D converter
- 8-bit timer/event counter 1
- Port 1 (P10-P13)

<Initial setting>

- OSMS = #00000001B ; Oscillation mode select register: does not use divider circuit
- ADM = #1000××××B ; Selects A/D converter channel and starts operation
- ADIS = #00000100B ; Selects number of A/D converter channels
- TCL1 = #00001110B ; Interval time of 8-bit timer/event counter 1: 10 ms

  TMC1 = #00000001B

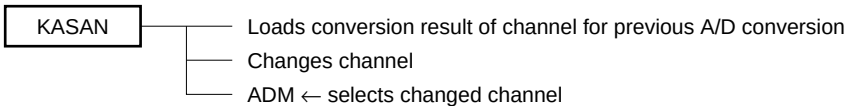
  CR10 = #81
- Enables TMMK1 interrupt

(2) Example of use

```
EXTRN   M_CH0,M_CH1,M_CH2,M_CH3,M_MODE
;**************************************
;           Initialize
;**************************************
M4          CSEG                        ;
RES_STA:
    SEL RB0                             ;
    DI                                  ;
          :
          :
          :
    OSMS=#00000001B                     ; Does not use divider circuit
    ADM=#10000001B                      ; Starts A/D operation and selects external trigger channel 0
    ADIS=#00000100B                     ; Selects analog input channel 4
    CR10=#81                            ; Sets modulo register 81
    TCL1=#00001110B                     ; Count clock: 8.2 kHz
    TMC1=#00000001B                     ; Enables 8-bit timer/register 1 operation
    CLR1    TMIF1                       ; Clears timer 1 interrupt request flag
    CLR1    TMMK1                       ; Enables timer 1 interrupt
    EI                                  ;
    M_MODE=#0                           ; Sets initial value (0 channel) to mode area
          :
    while(forever)                      ;
          :
        A=M_CH0                         ; A ← data of channel 0
          :
        A=M_CH1                         ; A ← data of channel 1
          :
        A=M_CH2                         ; A ← data of channel 2
          :
        A=M_CH3                         ; A ← data of channel 3
          :
```

(3) SPD chart

[A/D conversion processing]



(4) Program list

```
;*****************************************
;              A/D conversion
;*****************************************
;
$PC(054)                                    ;
;
PUBLIC  M_CH0,M_CH1,M_CH2,M_CH3,M_MODE      ;
;
VEINTM1 CSEG    AT 24H
        DW  KASAN
;*****************************************
;              RAM definition
;*****************************************
            DSEG    SADDR
M_CH0:      DS      1                       ; Area for channel 0 addition
M_CH1:      DS      1                       ; Area for channel 1 addition
M_CH2:      DS      1                       ; Area for channel 2 addition
M_CH3:      DS      1                       ; Area for channel 3 addition
M_MODE:     DS      1                       ; Mode storage area
;
        CSEG                                ;
KASAN:
        SEL RB2                             ; Selects bank 2
        switch(M_MODE)                      ; Channel currently selected?
        case 0:                             ;   Channel 0:
            M_CH0=ADCR (A)                  ;           Transfers conversion result to RAM
            M_MODE++                        ;
            ADM=#10000011B                  ;           Select channel 1:
            break                           ;
        case 1:                             ;   Channel 1:
            M_CH1=ADCR (A)                  ;           Transfers conversion result to RAM
            M_MODE++                        ;
            ADM=#10000101B                  ;           Selects channel 2
            break                           ;
        case 2:                             ;   Channel 2:
            M_CH2=ADCR (A)                  ;           Transfers conversion result to RAM
            M_MODE++                        ;
            ADM=#10000111B                  ;           Selects channel 3
            break                           ;
        case 3:                             ;   Channel 3:
            M_CH3=ADCR (A)                  ;           Transfers conversion result to RAM
            M_MODE=#0                       ;
            ADM=#10000001B                  ;           Selects channel 0
            break                           ;
        ends                                ;
        RETI                                ;
        END
```

[MEMO]

CHAPTER 10 APPLICATIONS OF D/A CONVERTER

The D/A converter of the 78K/0 series consists of two voltage output type D/A converter channels with an 8-bit resolution. This D/A can be used in two modes: normal mode and real-time output mode. In the normal mode, the output trigger is writing data to the D/A conversion value setting registers 0 and 1 (DACS0 and 1). In the real-time output mode, the output is triggered by the interrupt requests (INTTM1 and 2) of 8-bit timer/event counters 1 and 2. In this mode, set data to DACS0 and DACS1 after an output trigger has been generated until the next output trigger is generated.

The D/A converter is set by the D/A converter mode register.

Figure 10-1. Format of D/A Converter Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
DAM	0	0	DAM5	DAM4	0	0	DACE1	DACE0	FF98H	00H	R/W

DACE0	Controls D/A converter channel 0
0	Stops D/A conversion operation
1	Enables D/A conversion operation

DACE1	Controls D/A converter channel 1
0	Stops D/A conversion operation
1	Enables D/A conversion operation

DAM4	Operation mode of D/A converter channel 0
0	Normal mode
1	Real-time output mode

DAM5	Operation mode of D/A converter channel 1
0	Normal mode
1	Real-time output mode

Cautions 1. To use the D/A converter, set the multiplexed port pins in the input mode and disconnect the pull-up resistor.

2. Be sure to clear bits 2, 3, 6, and 7 to 0.

3. The output goes into a high-impedance state when D/A conversion operation is stopped.

4. The output trigger in the real-time output mode is INTTM1 for channel 0 and INTTM2 for channel 1.

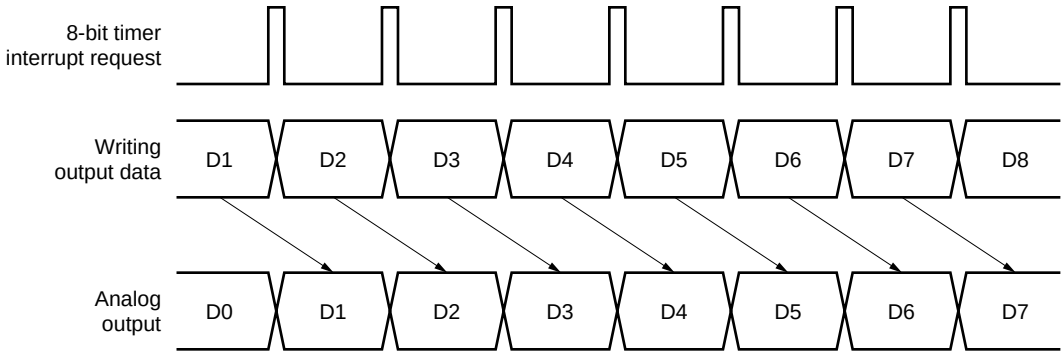
10.1 SIN Wave Output

This section introduces an example that outputs a SIN wave with a frequency of 50 Hz by using the real-time output mode of D/A converter channel 0.

After the output operation has been started, an analog value resulting from the D/A conversion specified by the D/A conversion value setting register 0 (DACS0) is output, and the next output data is set to DACS0 by interrupt processing. The value set by the interrupt processing is output at the next timing of 8-bit timer/event counter 1.

Figure 10-2 shows the output data writing timing and analog output timing.

Figure 10-2. Analog Output and Output Data Storage Timing



The interval time of 8-bit timer/event counter 1 is set to about 668 μs and a 50-Hz D/A output wave is generated as shown in Figure 10-3.

The SIN wave output data is stored in ROM. Data are sequentially referenced by the interrupt processing of 8-bit timer/event counter 1 and written to DACS0.

Table 10-1 shows the voltages for SIN wave output and set values.

Figure 10-3. D/A Output Waveform

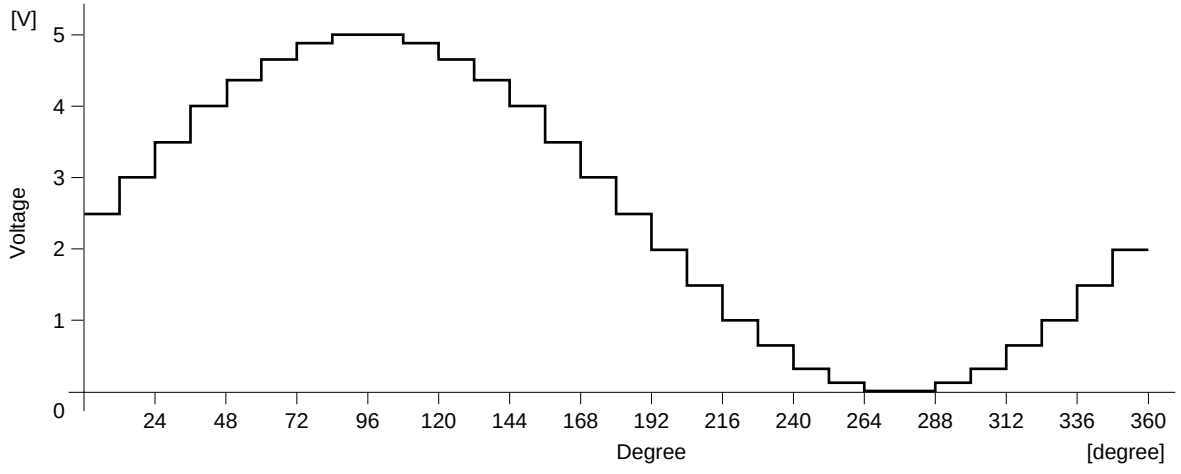


Table 10-1. Voltage of SIN Wave Output and Preset Value

Degree	Voltage (V)	Set Value	Degree	Voltage (V)	Set Value
0	2.5000	80H	180	2.5000	80H
12	3.0200	9BH	192	1.9802	65H
24	3.5168	B4H	204	1.4832	4CH
36	3.9695	CBH	216	1.0305	35H
48	4.3579	DFH	228	0.6421	21H
60	4.6651	EFH	240	0.3349	11H
72	4.8776	FAH	252	0.1224	06H
84	4.9863	FFH	264	0.0137	01H
96	4.9863	FFH	276	0.0137	01H
108	4.8776	FAH	288	0.1224	06H
120	4.6651	EFH	300	0.3349	11H
132	4.3579	DFH	312	0.6421	21H
144	3.9695	CBH	324	1.0305	35H
156	3.5168	B4H	336	1.4832	4CH
168	3.0200	9BH	348	1.9802	65H

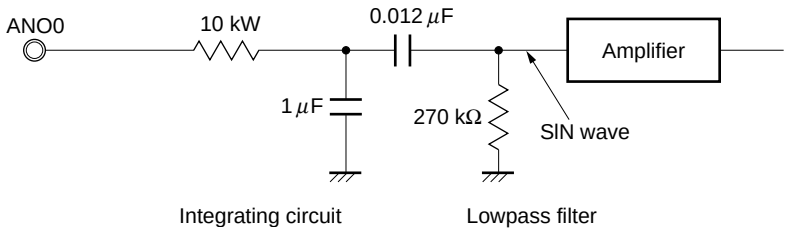
Remark The analog voltage output to the ANO0 pin is determined by the following expression:

$$\text{ANO0 pin output voltage} = \frac{AV_{REF1} \times DACS0}{256}$$

Caution The voltage values shown in Table 10-1 is rounded off at the fifth position after the decimal point. However, the preset value is calculated with the data before rounding off. The resultant data is rounded off at the first position after the decimal point.

The output analog value is processed by the SIN wave conversion circuit shown in Figure 10-4 to create a SIN wave without step.

Figure 10-4. SIN Wave Conversion Circuit



(1) Description of package

<Public declaration symbol>

- Data definition reference name
  SDATA : First address of SIN wave output data to be stored to DACS0 register
  ENDDAT : Last pointer of SIN wave data
- Input/output parameter
  C_DATA : ROM data counter

<Register used>

Bank 3; AX, HL, B

<RAM used>

Name	Usage	Attribute	Bytes
C_DATA	Counter indicating pointer that extracts SIN wave output data	SADDR	1

<Flag used>

None

<Nesting level>

1 level 3 bytes

<Hardware used>

- D/A converter
- 8-bit timer/event counter 1

<Initial setting>

- OSMS = #00000001B ; Oscillation mode select register: does not use divider circuit
- PM13 = #×××××××1B ; Sets port 13 in input mode
- TCL1 = #××××1001B ; Interval of 8-bit timer/event counter: 668 μs
  TMC1 = #000000×0B
  CR10 = #174
- DACS0 = #80H ; Sets D/A converter
  DAM = #00000001B
- SET1 DAM.4 ; Sets D/A converter in real-time output mode
- SET1 TCE1 ; Enables operation of 8-bit timer/event counter 1 and enables interrupt
  CLR1 TMIF1
  CLR1 TMMK1

Caution To prevent output of a value on resetting and starting, once set the normal mode and write the initial value to the D/A conversion value setting register 0 (DACS0), and then output the initial value. After that, set the real-time output mode, and enable the operation of 8-bit timer/event counter 1 and interrupt.
If D/A conversion is started in the real-time output mode after reset and start with the initial value set to the DACS0 register, 0 V (data D0 in Figure 10-2. Analog Output and Output Data Storage Timing) is output.

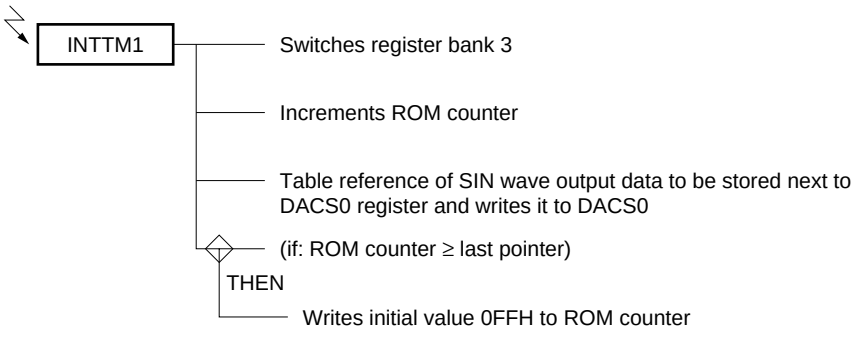
<Starting>

When starting output, enable the operation of the D/A converter (by setting bit 4 (DAM4) of the D/A converter mode register(DAM)), the operation of 8-bit timer/event counter 1 (by setting bit 0 (TCE) of the 8-bit timer mode control register (TMC1)), and interrupts (by clearing TMIF1 and TMMK1).

(2) Example of use

```
EXTRN   C_DATA,SDATA,ENDDAT
;
F_RIARU     EQU         DAM.4          ; Real-time output port setting flag
;
          :
    OSMS=#00000001B                    ; Does not use divider circuit
    TCL1=#00001001B                    ; SIN_DAT; 8-bit timer 1. Count clock: 262 kHz
    CR10=#175-1                        ; 8-bit timer 1. Interval: 668 μs
    TMC1=#00000000B                    ; Disables 8-bit timer 1 operation
;                                      ;
    HL=#SDATA                          ;
    B=C_DATA (A)                       ;
    DACS0=[HL+B] (A)                   ;
    DAM=#00000001B                     ; Enables D/A conversion operation of channel 0 in normal mode
    PM13=#11111111B                    ; Sets P130 in input port mode
    EI
          :
          :
          :
    if_bit(SIN wave output data start);
       SET1    F_RIARU                 ; Sets channel 0 in real-time output mode
       C_DATA=#0                       ; Sets initial value to conversion value setting register
       HL=#SDATA                       ;
       B=C_DATA (A)                    ;
       DACS0=[HL+B] (A)                ;
       SET1    TCE1                    ; Enables 8-bit timer 1 operation
       CLR1    TMIF1                   ; Clears 8-bit timer 1 request flag
       CLR1    TMMK1                   ; Enables 8-bit timer 1 interrupt
       SET1    DACE0                   ; Enables D/A operation
    endif                              ;
          :
          :
          :
```

(3) SPD chart



(4) Program list

```
PUBLIC  C_DATA,SDATA,ENDDAT
;
VETIM1  CSEG    AT 24H
        DW      INTTM1

ENDDAT      EQU         1DH                         ; SIN wave data 1 cycle end value
;
SINRAM      DSEG    SADDR
C_DATA:     DS      1                               ; ROM data counter
;*******************************************************
;      SIN wave data change interrupt processing
;*******************************************************
SINDAT  CSEG
INTTM1:
    SEL     RB3                                     ; Sets bank 3
    C_DATA++                                        ; Increments ROM data counter
    B=C_DATA (A)                                    ; Refers to SIN wave output data
    HL=#SDATA                                       ;
    DACS0=[HL+B] (A)                                ; Stores data
    if(C_DATA >= #ENDDAT)                           ; End of 1 cycle of SIN wave?
        C_DATA=#0FFH                                ;    Initializes ROM data counter
    endif                                           ;
    RETI                                            ;
SDATA:                                              ;
    DB      09BH                                    ; SIN wave data
    DB      0B4H                                    ;
    DB      0CBH                                    ;
    DB      0DFH                                    ;
    DB      0EFH                                    ;
    DB      0FAH                                    ;
    DB      0FFH                                    ;
    DB      0FFH                                    ;
    DB      0FAH                                    ;
    DB      0EFH                                    ;
    DB      0DFH                                    ;
    DB      0CBH                                    ;
    DB      0B4H                                    ;
    DB      09BH                                    ;
    DB      080H                                    ;
    DB      065H                                    ;
    DB      04CH                                    ;
    DB      035H                                    ;
    DB      021H                                    ;
    DB      011H                                    ;
    DB      006H                                    ;
    DB      001H                                    ;
    DB      001H                                    ;
    DB      006H                                    ;
    DB      011H                                    ;
    DB      021H                                    ;
    DB      035H                                    ;
    DB      04CH                                    ;
    DB      065H                                    ;
    DB      080H                                    ;
    END
```

[MEMO]

CHAPTER 11 APPLICATION OF REAL-TIME OUTPUT PORT

This chapter describes the real-time output function of the 78K/0 series.

The real-time output function is used to output data set in advance in the real-time output buffer registers (RTBL and RTBH) to an external device by transferring the data to an output latch by hardware as soon as a timer interrupt request or external interrupt request occurs.

By using the real-time output port function, a jitter free signal can be output. Therefore, this function is ideal for controlling a stepping motor. The real-time output port can be set in the port mode or real-time output mode in 1-bit units.

The real-time output data is written to the real-time output buffer registers (RTBL and RTBH). RTBL and RTBH are mapped to independent addresses in the SFR area.

When an operation mode of 4 bits × 2 channels is selected, RTBL and RTBH can independently set data.

When an operation mode of 8 bits × 1 channel is specified, data can be set to RTBL or RTBH by writing 8-bit data to either of RTBL or RTBH.

The real-time output port is set by using the real-time output port mode register (RTPM), real-time output control register (RTPC), and port mode register 12 (PM12).

Figure 11-1. Format of Real-Time Output Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
RTPM	RTPM7	RTPM6	RTPM5	RTPM4	RTPM3	RTPM2	RTPM1	RTPM0	FF34H	00H	R/W

RTPMn	Selects real-time output port (n = 0-7)
0	Port mode
1	Real-time output port mode

Cautions 1. When the real-time output port mode is used, the port that performs real-time output must be set in the output mode (by clearing the corresponding bits of the port mode register 12 (PM12) to 0).

2. Data cannot be set to the output latch of the port set in the real-time output port mode. To set an initial value, therefore, set data to the output latch before setting the real-time output port mode.

Figure 11-2. Format of Real-Time Output Port Control Register

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
RTPC	0	0	0	0	0	0	BYTE	EXTR	FF36H	00H	R/W

EXTR	Controls real-time output by INTP2
0	Does not use INTP2 as real-time output trigger
1	Uses INTP2 as real-time output trigger

BYTE	Operation mode of real-time output port
0	4 bits × 2 channels
1	8 bits × 1 channel

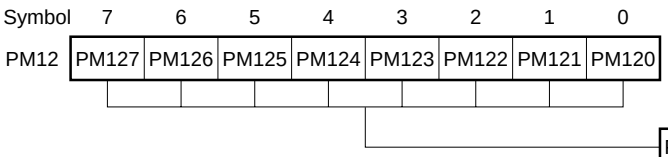
The relationship between operation mode and output trigger of the real-time output port is shown in Table 11-1.

Table 11-1. Operation Mode and Output Trigger of Real-Time Output Port

BYTE	EXTR	Output Mode	RTBH → Port Output	RTBL → Port Output
0	0	4 bits × 2 channels	INTTM2	INTTM1
	1		INTTM1	INTP2
1	0	8 bits × 1 channel	INTTM1	
	1		INTP2	

★

Figure 11-3. Format of Port Mode Register 12



PM12n	Selects input/output mode of P12n pin (n = 0-7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

11.1 Stepping Motor

A 4-phase stepping motor is connected to the real-time output port (P120 through P123) and is controlled with 1-phase excitation pattern. A motor that rotates 1.8 degree per step is used for 1-phase excitation and is driven 200 revolutions per minute.

The time required for 1 step is calculated by the following expression:

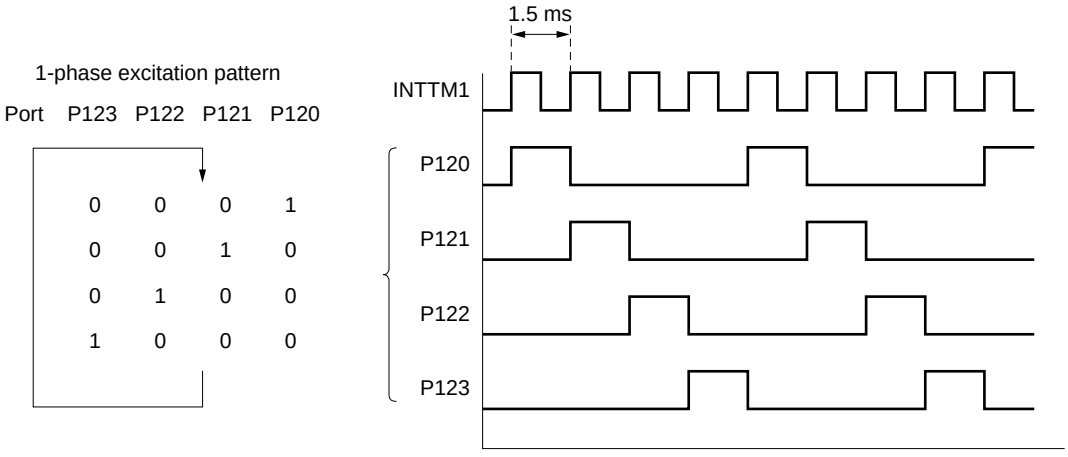
$$1 \text{ step} = \frac{60 \text{ seconds}}{200 \text{ revolutions} \times \dfrac{360 \text{ degrees}}{1.8 \text{ degree}} \text{ Step}} = 1.5 \text{ ms}$$

The compare register (CR01) of 8-bit timer/event counter 1 is set to 1.5 ms and the real-time output buffer register (RTBL) is set.

By using the real-time output port control register (RTPC), set the 4 bit × 2 channel real-time output mode, and the coincidence interrupt (INTTM1) of the 8-bit timer/event counter 1 as the output trigger (refer to **Table 11-1**).

Figure 11-4 shows the phase excitation output pattern and output timing.

Figure 11-4. Phase Excitation Output Pattern and Output Timing



(1) Description of package

<Public declaration symbol>

None

<Register used>

Bank 3, A

<RAM used>

None

<Flag used>

None

<Nesting level>

1 level 3 bytes

<Hardware used>

- Real-time output port
- 8-bit timer/event counter 1

<Initial setting>

- OSMS = #00000001B ; Oscillation mode select register: does not use divider circuit
- P12 = #××××0000B ; Sets P120-P123 in output port mode
  PM12 = #××××0000B
- TCL1 = #××××1010B ; Timer clock select register 1 (count clock: 131 kHz)
- CR10 = #195 ; Compare register (set to 1.5 ms)
- TMC1 = #000000×1B ; 8-bit timer mode control register 1 (enables operation of 8-bit timer/event counter 1)
- RTPM = ××××1111B ; Real-time output port mode register (lower 4 bits are used as real-time output port)
- RTPC = #00000000B ; Real-time output port control register (selects 4 bit × 2 channel mode and INTTM1 as output trigger)
- RTBL = #00000001B ; Initial setting of real-time output buffer register
- CLR1 TMIF1 ; Clears 8-bit timer/event counter 1 interrupt request flag
- CLR1 TMMK1 ; Enables 8-bit timer/event counter 1

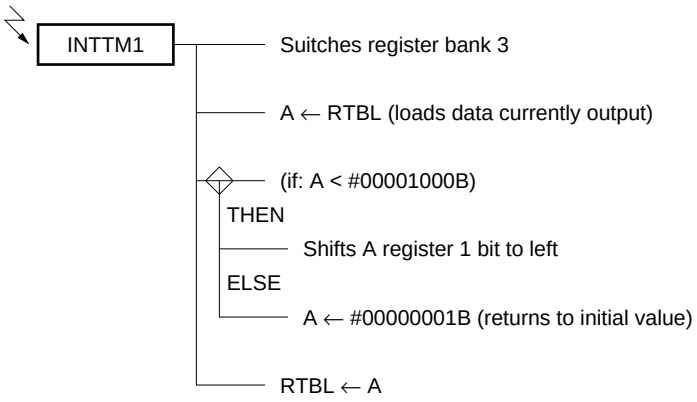
<Starting>

Clear the interrupt request flag of 8-bit timer/event counter 1 and enable the interrupt when the operation is started.

(2) Example of use

```
          :
OSMS=#00000001B
TCL1=#00001010B             ; Does not use divider circuit
CR10=#196-1                 ; MORTER_DAT: 8-bit timer 1. Count clock: 131 kHz
P12=#00000000B              ; Sets compare register to 1.5 ms
PM12=#11110000B             ;
RTPM=#00001111B             ; Sets P120-P123 in output port mode
RTPC=#00000000B             ; Sets low-order 4 bits in output port mode
RTBL=#00000001B             ; Uses INTTM1 as output trigger
TMC1=#00000001B             ;
CLR1    TMIF1               ; Enables 8-bit timer 1 operation
CLR1    TMMK1               ; Clears 8-bit timer 1 interrupt request flag
EI                          ; Enables 8-bit timer 1 interrupt
                            ;
            :
```

(3) SPD chart



(4) Program list

```
VETIM1      CSEG    AT 24H
            DW      INTTM1
;*********************************************
;       Stepping motor data output processing
;*********************************************
MOTER       CSEG
INTTM1:
    SEL     RB3                 ; Bank 3
    A=RTBL                      ;
    if(A < #8)                  ; Prepares next output data
        A <<= 1                 ;
    else                        ;
        A=#01H                  ;
    endif                       ;
    RTBL=A                      ; RTPL ← output data
    RETI                        ;
    END
```

CHAPTER 12 APPLICATIONS OF LCD CONTROLLER/DRIVER

The LCD controller/driver of the μPD78064, 78064Y, 780308, 780308Y, and 78064B subseries is set by using the LCD display mode register (LCDM) and LCD display control register (LCDC).

Figure 12-1. Format of LCD Display Mode Register (μPD78064, 78064Y, 78064B subseries)

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
LCDM	LCDON	LCDM6	LCDM5	LCDM4	0	LCDM2	LCDM1	LCDM0	FFB0H	00H	R/W

LCDM2	LCDM1	LCDM0	Time division	Bias
0	0	0	4	1/3
0	0	1	3	1/3
0	1	0	2	1/2
0	1	1	3	1/2
1	0	0	Static	
Others			Setting prohibited	

LCDM6	LCDM5	LCDM4	Selects LCD clock[Note]		
			At f_{XX} = 5.0 MHz	At f_{XX} = 4.19 MHz	At f_{XT} = 32.768 kHz
0	0	0	$f_W/2^9$ (76 Hz)	$f_W/2^9$ (64 Hz)	$f_W/2^9$ (64 Hz)
0	0	1	$f_W/2^8$ (153 Hz)	$f_W/2^8$ (128 Hz)	$f_W/2^8$ (128 Hz)
0	1	0	$f_W/2^7$ (305 Hz)	$f_W/2^7$ (256 Hz)	$f_W/2^7$ (256 Hz)
0	1	1	$f_W/2^6$ (610 Hz)	$f_W/2^6$ (512 Hz)	$f_W/2^6$ (512 Hz)
Others			Setting prohibited		

LCDON	Enables/disables LCD display
0	Display off (all segment outputs are unselect signal outputs)
1	Display on

Note The LCD clock is supplied by the watch timer. To perform LCD display, set the bit 1 (TMC21) of watch timer mode control register (TMC2) to 1.
If TMC21 is reset to 0 during LCD display, supply of the LCD clock is stopped and the display is disturbed.

Remarks 1. f_W : watch timer clock frequency ($f_{XX}/2^7$ or f_{XT})
2. f_{XX} : main system clock frequency (f_X or $f_X/2$)
3. f_X : main system clock oscillation frequency
4. f_{XT} : subsystem clock oscillation frequency

★ **Figure 12-2. Format of LCD Display Mode Register (μPD780308, 780308Y subseries)**

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
LCDM	LCDON	LCDM6	LCDM5	LCDM4	LCDM3	LCDM2	LCDM1	LCDM0	FFB0H	00H	R/W

LCDM2	LCDM1	LCDM0	Time division	Bias
0	0	0	4	1/3
0	0	1	3	1/3
0	1	0	2	1/2
0	1	1	3	1/2
1	0	0	Static	
Others			Setting prohibited	

Note 1 LCDM3	Operation mode of LCD controller/driver	Supply voltage of LCD controller/driver		
		Static display mode	1/3 bias mode	1/2 bias mode
0	Normal operation	2.0 to 5.5 V	2.5 to 5.5 V	2.7 to 5.5 V
1	Low-voltage operation	2.0 to 3.4 V		

LCDM6	LCDM5	LCDM4	Selects LCD clock**Note**		
			At f_{XX} = 5.0 MHz	At f_{XX} = 4.19 MHz	At f_{XT} = 32.768 kHz
0	0	0	$f_W/2^9$ (76 Hz)	$f_W/2^9$ (64 Hz)	$f_W/2^9$ (64 Hz)
0	0	1	$f_W/2^8$ (153 Hz)	$f_W/2^8$ (128 Hz)	$f_W/2^8$ (128 Hz)
0	1	0	$f_W/2^7$ (305 Hz)	$f_W/2^7$ (256 Hz)	$f_W/2^7$ (256 Hz)
0	1	1	$f_W/2^6$ (610 Hz)	$f_W/2^6$ (512 Hz)	$f_W/2^6$ (512 Hz)
Others			Setting prohibited		

LDON	Enables/disables LCD display
0	Display off (all segment outputs are unselect signal outputs)
1	Display on

Notes 1. To lower the power consumption, clear LCDM3 to 0 when LCD display is not used. To manipulate LCDM3, be sure to turn off the LCD display.
If TMC21 is cleared to 0 during LCD display, the supply of the LCD clock is stopped and the display is disturbed.

2. The LCD clock is supplied by the watch timer. To perform LCD display, set the bit 1 (TMC21) of watch timer mode control register (TMC2) to 1.
If TMC21 is reset to 0 during LCD display, supply of the LCD clock is stopped and the display is disturbed.

Remarks 1. f_W : watch timer clock frequency ($f_{XX}/2^7$ or f_{XT})

2. f_{XX} : main system clock frequency (f_X or $f_X/2$)

3. f_X : main system clock oscillation frequency

4. f_{XT} : subsystem clock oscillation frequency

Figure 12-3. Format of LCD Display Control Register

Symbol	7	6	5	4	3	2	1	0	Address	At reset	R/W
LCDC	LCDC7	LCDC6	LCDC5	LCDC4	0	0	LEPS	LIPS	FFB2H	00H	R/W

LEPS	LIPS	Selects LCD drive power supply
0	0	Does not supply power for LCD driving
0	1	Supplies LCD driving power from V_{DD}
1	0	Supplies driving power from BIAS pin (BIAS and V_{LC0} pins are internally short-circuited)
1	1	Setting prohibited

LCDC7	LCDC6	LCDC5	LCDC4	Function of P80/S39-P97/ S24 pins	
				Port pin	Segment pin
0	0	0	0	P80-P97	None
0	0	0	1	P80-P95	S24, S25
0	0	1	0	P80-P93	S24-S27
0	0	1	1	P80-P91	S24-S29
0	1	0	0	P80-P87	S24-S31
0	1	0	1	P80-P85	S24-S33
0	1	1	0	P80-P83	S24-S35
0	1	1	1	P80, P81	S24-S37
1	0	0	0	None	S24-S39
Others				Setting prohibited	

Cautions 1. Pins that output segments cannot be used as output port pins even if 0 is set to the corresponding port mode register.

2. When pins that output segments are read as port pins, 0 is returned.

3. Pins set by LCDC to output segments are not used with the internal pull-up resistor, regardless of the values of the bits 0 and 1 (PUO8 and PUO9) of the pull-up resistor option register H (PUOH).

The LCD controller/driver of the μPD78064, 78064Y, 780308, and 780308Y subseries is described next.

(a) Function of LCD controller/driver

The LCD controller/driver has the following functions:

<1> Automatically outputs segment and common signals by automatically reading the display data memory.

<2> Five types of display modes are available:

- Static mode
- 1/2 duty mode (1/2 bias)
- 1/3 duty mode (1/2 bias)
- 1/3 duty mode (1/3 bias)
- 1/4 duty mode (1/3 bias)

<3> Four types of frame frequencies can be selected in each display mode.

<4> Up to 40 segment signal outputs (S0 through S39) and four common signal outputs (COM0 through COM3) are available. Sixteen segment outputs can be set in the input/output port mode in 2-bit units (P80/S39 through P87/S32, and P90/S31 through P97/S24).

<5> Divider resistors for generating the LCD drive voltage can be provided to the mask ROM model by mask option.

<6> Can operate on the subsystem clock.

Table 12-1 shows the maximum number of pixels that can be displayed in each display mode.

Table 12-1. Maximum Number of Pixels for Display

Bias	Time Division	Common Signal Used	Maximum Number of Pixels
–	Static	COM0 (COM0-COM3)	40 (40 segments × 1 common)Note 1
1/2	2	COM0, COM1	80 (40 segments × 2 common)Note 2
	3	COM0-COM2	120 (40 segments × 3 common)Note 3
1/3	3	COM0-COM2	
	4	COM0-COM3	160 (40 segments × 4 common)Note 4

Notes 1. Can display 5 digits with eight segments for each digit on an 8-segment LCD panel.

2. Can display 10 digits with four segments for each digit on an 8-segment LCD panel.

3. Can display 13 digits with three segments for each digit on an 8-segment LCD panel.

4. Can display 20 digits with two segments for each digit on an 8-segment LCD panel.

(b) Setting of LCD controller/driver

Set the LCD controller/driver as follows. When using the LCD controller/driver, set the watch timer in the operating status in advance.

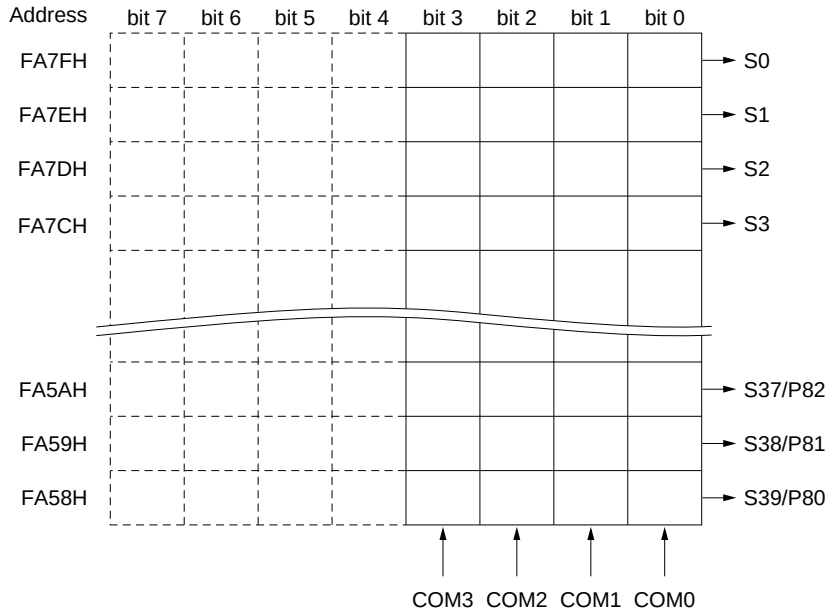
<1> Enables the watch operation by using the timer clock select register 2 (TCL2) and watch timer mode control register (TMC2).
<2> Set the initial value to the display data memory (FA58H through FA7FH).
<3> Specify the pins used for segment output by using the LCD display control register (LCDC).
<4> Set the display mode and LCD clock by using the LCD display mode register.

After that, set data to the display data memory according to the contents to be displayed.

(c) LCD display data memory

The LCD display data memory is mapped to addresses FA58H through FA7FH. The data stored to the LCD display data memory can be displayed on the LCD panel by using the LCD controller/driver.
Figure 12-4 shows the relations between the contents of LCD display data memory and the segment/common outputs.
The area not used for display can be used as an ordinary RAM area.

Figure 12-4. Relations between Contents of LCD Display Data Memory and Segment/Common Output



Caution The high-order 4 bits of the LCD display data memory are not used as memory bits. Be sure to clear these bits to 0.

(d) Common and segment signals

Each pixel on an LCD panel lights when the potential difference between the corresponding common and segment signals reaches to a specific level (LCD drive voltage V_{LCD}).

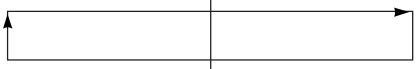
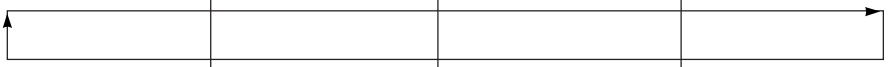
Because an LCD panel degrades if DC voltages are applied as common and segment signals, it is driven by AC voltages.

<1> Common signal

The common signal is selected as shown in Table 12-2 according to the set number of time divisions and repeatedly operates in the cycles shown in the table. In the static mode, the same signal is output to COM0 through COM3.

In the 2-time division mode, open the COM2 and COM3 pins. Open the COM3 pin in the 3-time division mode.

Table 12-2. COM Signal

COM Signal / Number of Time Divisions	COM0	COM1	COM2	COM3
Static				
2 time divisions			Open	Open
3 time divisions				Open
4 time divisions				

<2> Segment signal

Segment signals correspond to a 40-byte LCD display data memory (FA58H through FA7FH). Bits 0, 1, 2, and 3 of the display data memory are read in synchronization of COM0, COM1, COM2, and COM3, respectively. If the content of each bit is 1, the corresponding segment signal is converted to a select voltage and is output to the segment pin (S0 to S39). If the content of a bit is 0, the segment signal is converted to an unselect voltage. (Note that S24 through S39 are multiplexed with input/output port pins.) Therefore, confirm what combination of the front panel electrode (corresponding to a segment signal) and rear panel electrode (corresponding to a common signal) of the LCD panel generates a display pattern, and write the bit data corresponding to the pattern to be displayed on a one-to-one basis to the LCD display memory.

In the static mode, bits 1, 2, and 3 of the LCD display data memory are not used for LCD display. In the 2- and 3-time division modes, bits 2 and 3, and bit 3 are not used for LCD display, respectively. These bits therefore can be used for any other purposes.

Bits 4 through 7 are fixed to 0.

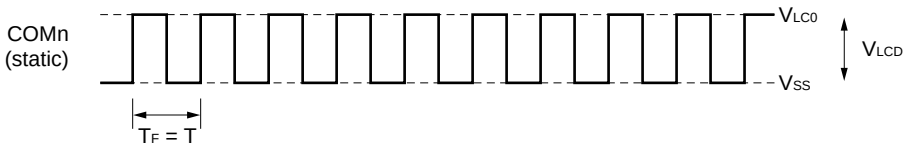
<3> Output waveforms of common and segment signals

Only when the voltage levels of specific common and segment signals reach the select levels, ±V_{LCD} (LCD drive voltage) level is reached and the corresponding pixel on the LCD panel lights. The pixel remains dark with any other combination of the common and segment signals.

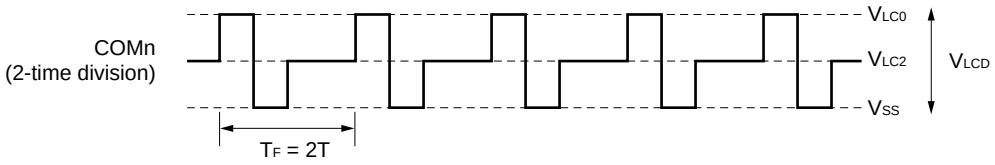
Figure 12-5 shows the waveform of the common signal, and Figure 12-6 shows the phase difference in voltage between the common and segment signals.

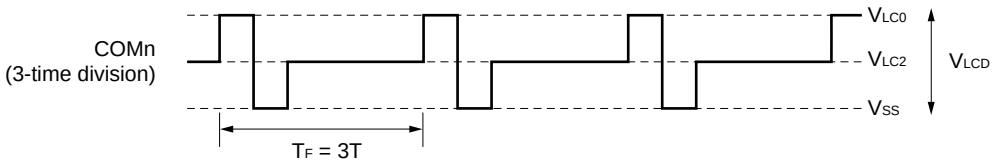
Figure 12-5. Common Signal Waveform

(a) Static display mode

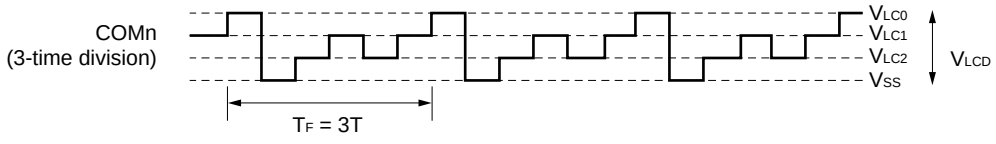


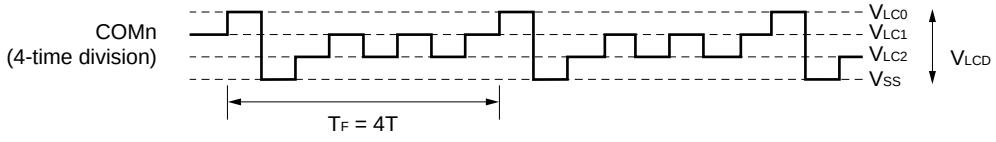
(b) 1/2 bias





(c) 1/3 bias





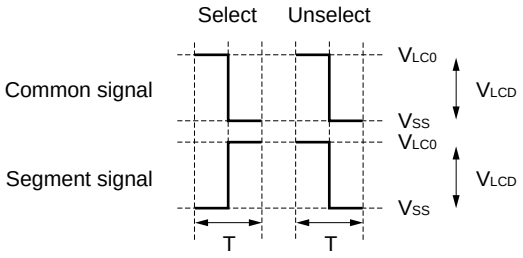
Remarks 1. T : one cycle of LCDCL

2. T_F : frame frequency

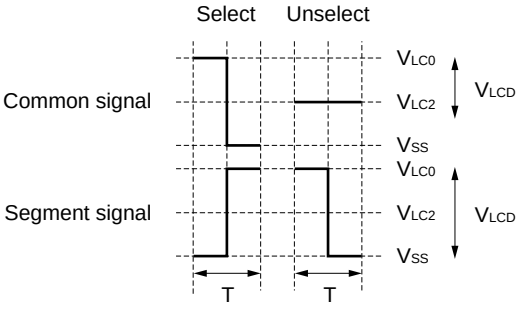
3. V_{LCD} : LCD drive voltage

Figure 12-6. Phase Difference in Voltage between Command Signal and Segment Signal

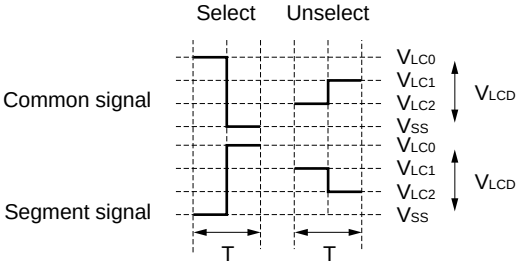
(a) Static display mode



(b) 1/2 bias



(c) 1/3 bias



Remarks 1. T : one cycle of LCDCL

2. V_{LCD} : LCD drive voltage

(e) Supplying LCD drive voltage

The mask ROM model can be provided by mask option with a divider resistor that is used to create the LCD drive voltage (the PROM model is not provided with a divider resistor). By providing the divider resistor, an LCD drive voltage corresponding to each bias can be created without an external divider resistor.

In addition, an LCD drive voltage can be supplied to the BIAS pin to support various LCD drive voltages.

12.1 Static Display

This section explains an example using the μPD78064 subseries. A 4-digit static LCD is driven by using the 32 segment signals (S0 through S31) and a common signal (COM0). Figure 12-7 shows the display pattern and electrode wiring of the static LCD. Figure 12-8 shows the connections among the segment signals and common signal. Figure 12-9 shows an example of connecting an LCD driving power supply in the static display mode (with an external divider resistor, V_{DD} = 5 V, and V_{LCD} = 5 V). The display example in Figure 12-8 is "1234", and the contents of the display data memory (addresses FA60H through FA7FH) correspond to this.

In this section, how to display the second digit, "3", is described. According to the display pattern in Figure 12-8, the select and unselect voltages must be output to the S8 through S15 pins in the timing of the common signal COM0, as shown in Table 12-3.

Table 12-3. Select and Unselect Voltages (COM0)

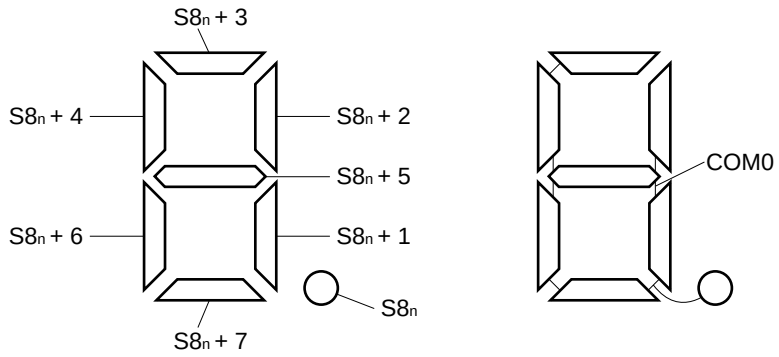
Segment / Common	S8	S9	S10	S11	S12	S13	S14	S15
COM0	Unselect	Select	Select	Select	Unselect	Select	Unselect	Select

From Table 12-3, it is clear that 10101110 must be set to bit 0 of the display data memory (addresses FA70H through FA77H) corresponding to S8 through S15.

Figure 12-10 shows the LCD driving waveforms of S11, S12, and COM0.

Because the same waveform as COM0 is output to COM1, 2, and 3, the driving capability can be increased by connecting COM0, 1, 2, and 3.

Figure 12-7. Display Pattern and Electrode Wiring of Static LCD



Remark n = 0-3

Figure 12-8. Connection of Static LCD

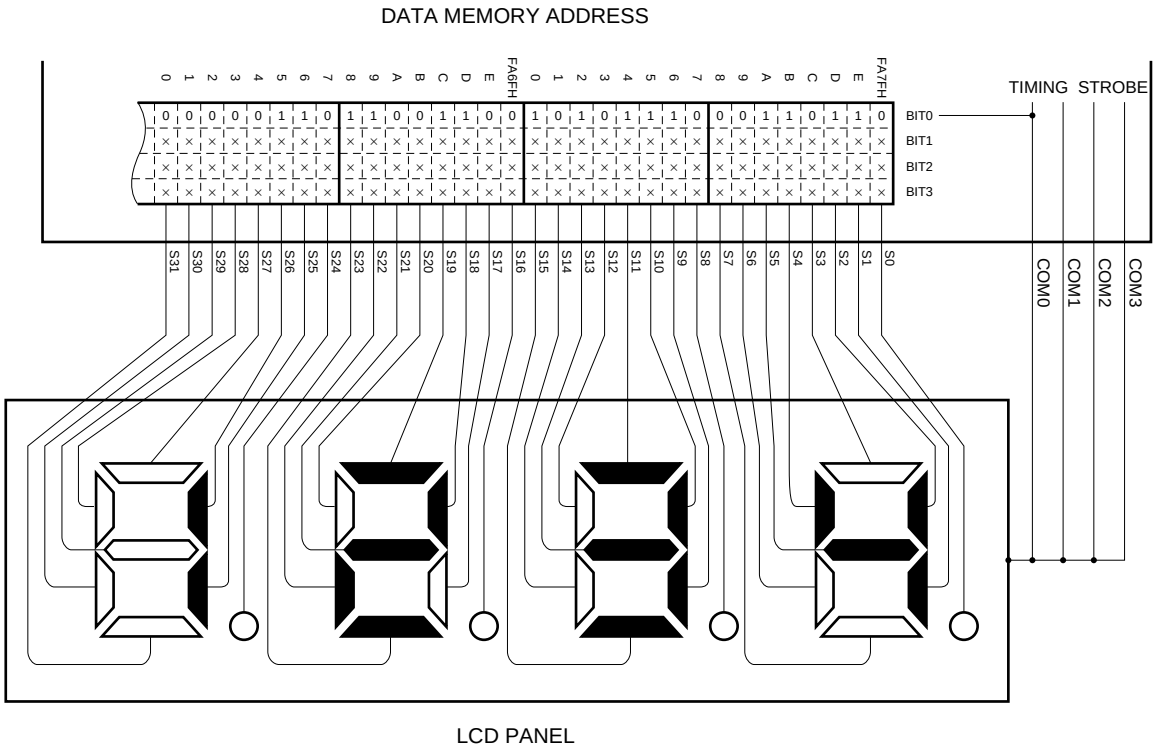


Figure 12-9. Example of Connecting LCD Driving Power in Static Display Mode (with external divider resistor, V_{DD} = 5 V, and V_{LCD} = 5 V)

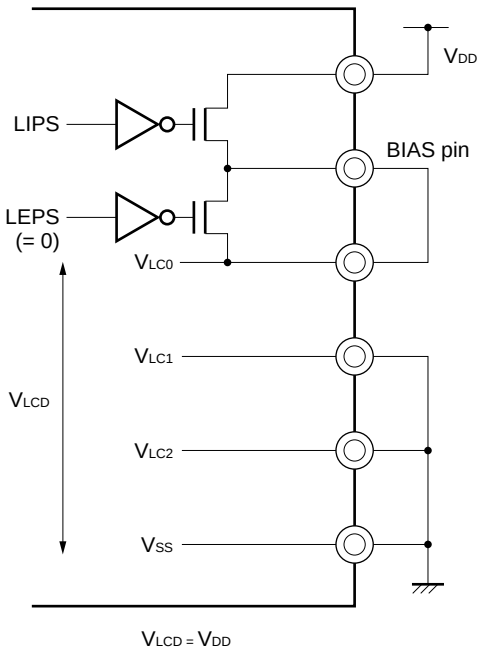
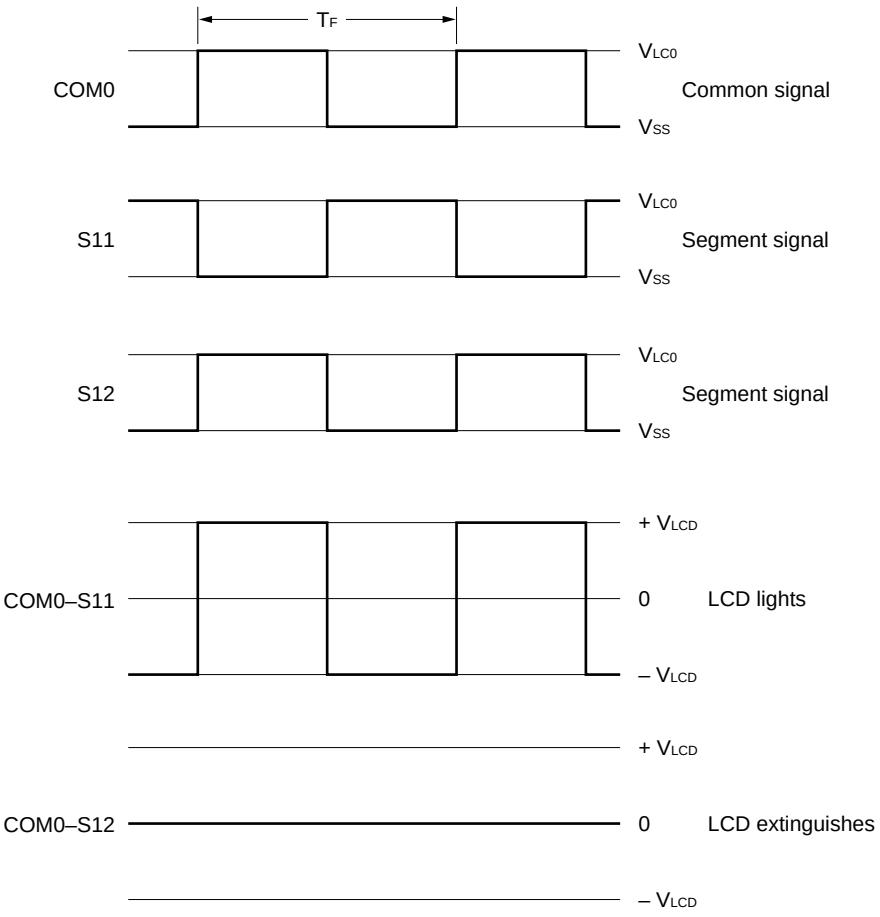


Figure 12-10. Example of Static LCD Driving Waveform



Remark T_F: frame frequency

To display the LCD, segment signals are output based on the waveform of the common signal.

The static LCD is lit by a segment signal (S11) output at a frame frequency half a cycle shifted from that of the common signal (COM0) as shown in Figure 12-10. This means that a potential difference is generated between the common signal and segment signal, and this potential difference is responsible for lighting the LCD. As can be seen from COM0 and S11 in Figure 12-10, a potential difference $\pm V_{LCD}$ (LCD drive voltage) is generated between these signals.

To extinguish the LCD, the segment signal (S12) is output in a waveform synchronous to that of the common signal (COM0). In this way, the potential difference between COM0 and S12 is eliminated and the LCD remains dark.

(1) Description of package

<Public declaration symbol>

- Subroutine name
  S_LCD : Static display data storage routine
- Input parameter
  B_LCD : LCD display content storage buffer area
- Data definition reference name
  S0 : LCD display data memory reference address (FA7FH)

<Register used>

Bank 0 : AX, DE, HL

<RAM used>

Name	Usage	Attribute	Bytes
B_LCD	LCD display data storage buffer area	SADDR	1
i	Display digit loop counter	SADDR	1
j	Segment setting loop counter	SADDR	1
WORKP	Display data storage area address saving area	SADDRP	2

<Flag used>

None

<Nesting level>

1 level 2 bytes

<Hardware used>

- LCD controller/driver

<Initial setting>

- OSMS = #00000001B ; Oscillation mode select register
- TCL2 = #×××00×××B ; Count clock of watch timer = selects system clock
- TMC2 = #0××××1×B ; Supplies LCD clock (enables prescaler operation)
- LCDC = #01000010B ; LCD display control register (supplies LCD drive power from BIAS pin with segment pins S24 through S31 used)
- LCDM = #10100100B ; LCD display mode register (sets static display, selects LCD clock, and turns on display)

Caution Set the initial value to the LCD display data memory (FA58H through FA7FH) before turning on the LCD display.

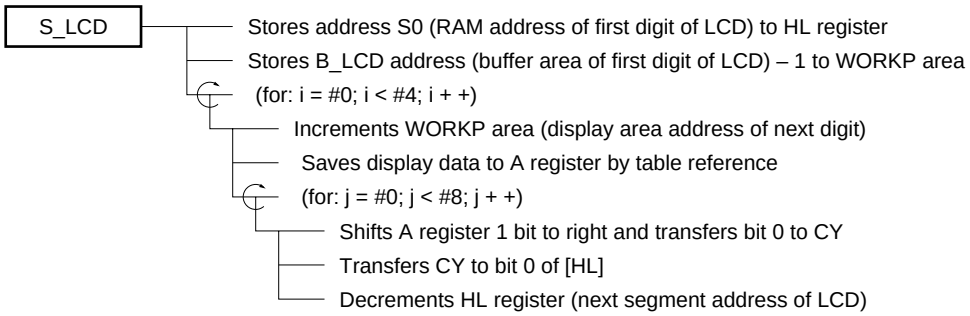
<Starting>

Set the display contents in the B_LCD area and call the S_LCD routine.

(2) Example of use

```
EXTRN   SLCD,B_LCD,S0
        :
    HL=#S0                  ; Clears LCDRAM
    BC=#0FA80H-0FA58H       ;   (from FA58H to FA7FH)
    while(BC!=#0) (AX)      ;
        A=#0                ;
        [HL]=A              ;
        HL--                ;
        BC--                ;
    endw                    ;
;
    B_LCD=#0                ;
    B_LCD+1=#0              ;
    B_LCD+2=#0              ;
    B_LCD+3=#0              ;
;
    TCL2=#00000000B         ; Count clock of watch timer = selects main system clock
    TMC2=#00000010B         ; Enables operation of prescaler
    LCDC=#01000010B         ; Supplies LCD drive power from BIAS pin with segments S24 through 31 used
    LCDM=#10100100B         ; Turns ON static display with 256-Hz clock selected
;
        :
      B_LCD+3=A             ;
        :
      B_LCD+2=A             ;
        :
      B_LCD+1=A             ;
        :
      B_LCD=A               ;
      CALL    !S_LCD        ;
        :
```

(3) SPD chart



(4) Program list

```
PUBLIC  S_LCD,B_LCD,S0
;
S0          EQU         0FA7FH                  ; 1st digit of LCD
;
LCDRAM1     DSEG    SADDR
B_LCD:      DS      4                           ; Display BUF area
i:          DS      1                           ; Work counter
j:          DS      1                           ; Work counter
LCDRAM2     DSEG    SADDRP
WORKP:      DS      2                           ; Work area
;*************************************************
;      LCD display (static display) processing
;*************************************************
LSDS        CSEG
S_LCD:
    HL=#S0                                      ; HL ← address S0
    WORKP=#B_LCD-1                              ; Work area ← address of B_LCD - 1
    for(i=#0;i<#4;i++)                          ;
        DE=WORKP (AX)                           ;
        DE++                                    ; References display data of contents of next digit
        WORKP=DE (AX)                           ;
        X=[DE] (A)                              ;
        A=#0                                    ;
        AX+=#LCDDAT                             ;
        DE=AX                                   ;
        A=[DE]                                  ;
        for(j=#0;j<#8;j++)                      ;
            RORC    A,1                         ; Stores display data to bit 0 from address S0
            [HL].0=CY                           ;
            HL--                                ;
        next                                    ;
    next                                        ;
    RET                                         ;
LCDDAT:
    DB  11011110B                               ; 0
    DB  00000110B                               ; 1
    DB  11101100B                               ; 2
    DB  10101110B                               ; 3
    DB  00110110B                               ; 4
    DB  10111010B                               ; 5
    DB  11111010B                               ; 6
    DB  00011110B                               ; 7
    DB  11111110B                               ; 8
    DB  10111110B                               ; 9
    DB  01111110B                               ; A
    DB  11110010B                               ; B
    DB  11011000B                               ; C
    DB  11100110B                               ; D
    DB  11111000B                               ; E
    DB  01111000B                               ; F
    DB  00000000B                               ; Extinguishes
    END
```

12.2 4-Time Division Display

This section explains an example using the μPD78064 subseries. Four LCD digits are driven by means of 1/3 bias and 4-time division by using the 16 segment signals (S0 through S15) and four common signals (COM0 through COM3). Figure 12-12 shows the connection of a 4-time division 4-digit LCD panel with 10 display patterns shown in Figure 12-11 and the segment (S0 through S15) and common (COM0 through COM4) signals of the μPD78064 subseries. Figure 12-13 shows an example of connecting an LCD drive power supply in the 4-time division display mode (with external divider resistor, V_{DD} = 5 V, and V_{LCD} = 5 V). The display example in Figure 12-12 is "12345678", and the contents of the display data memory (addresses FA70H through FA7FH) correspond to this.

In this case, "6" at the third digit has been taken as an example. According to the display pattern in Figure 12-12, the select and unselect voltages shown in Table 12-4 must be output to the S4 and S5 pins in the timing of the common signals COM0 through COM3.

Table 12-4. Select and Unselect Voltages (COM0, 1, 2, 3)

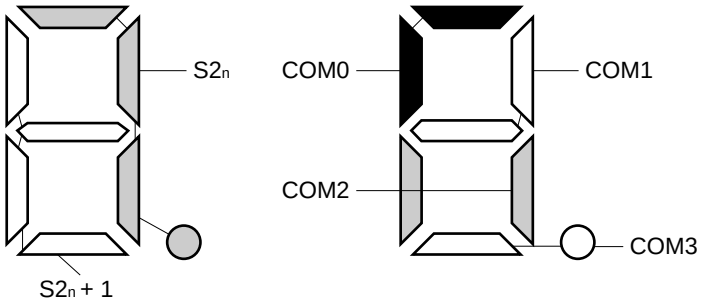
Segment / Common	S4	S5
COM0	Select (a)	Select (e)
COM1	Unselect (b)	Select (f)
COM2	Select (c)	Select (g)
COM3	Unselect (d)	Select (h)

Remark (a) through (h) in the table corresponds to the segments a through h in Figure 12-12.

Table 12-4 indicates that 0101 should be stored to the display data memory address (FA7BH) corresponding to S4.

Figure 12-14 shows the LCD drive waveforms between S4 and COM0 and COM1 signals.

Figure 12-11. Display Pattern of 4-Time Division LCD and Electrode Wiring



Remark n = 0-7

Figure 12-12. Connections of 4-Time Division LCD Panel

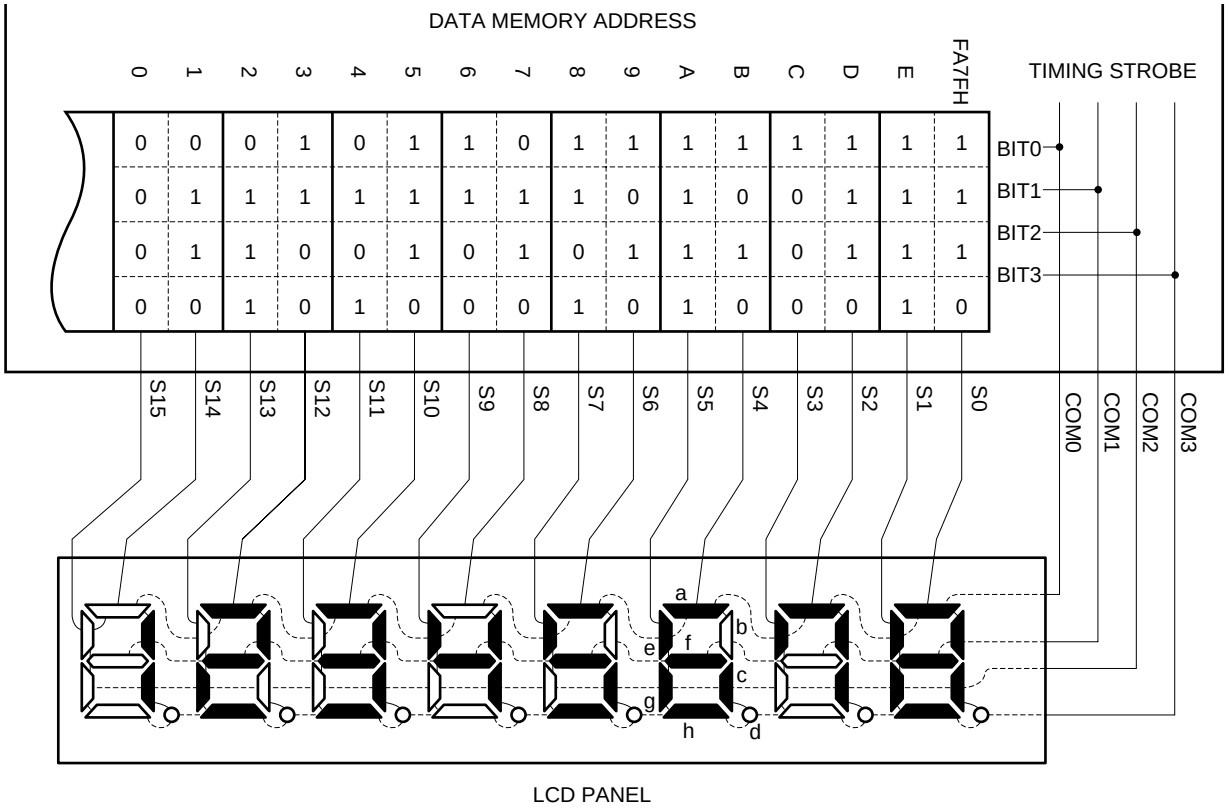


Figure 12-13. Example of Connecting LCD Drive Power in 4-Time Division Mode (with external divider resistor, V_{DD} = 5 V, V_{LCD} = 5 V)

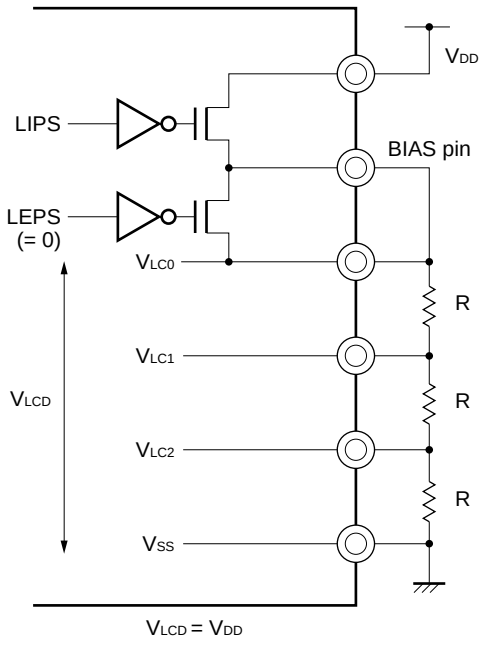
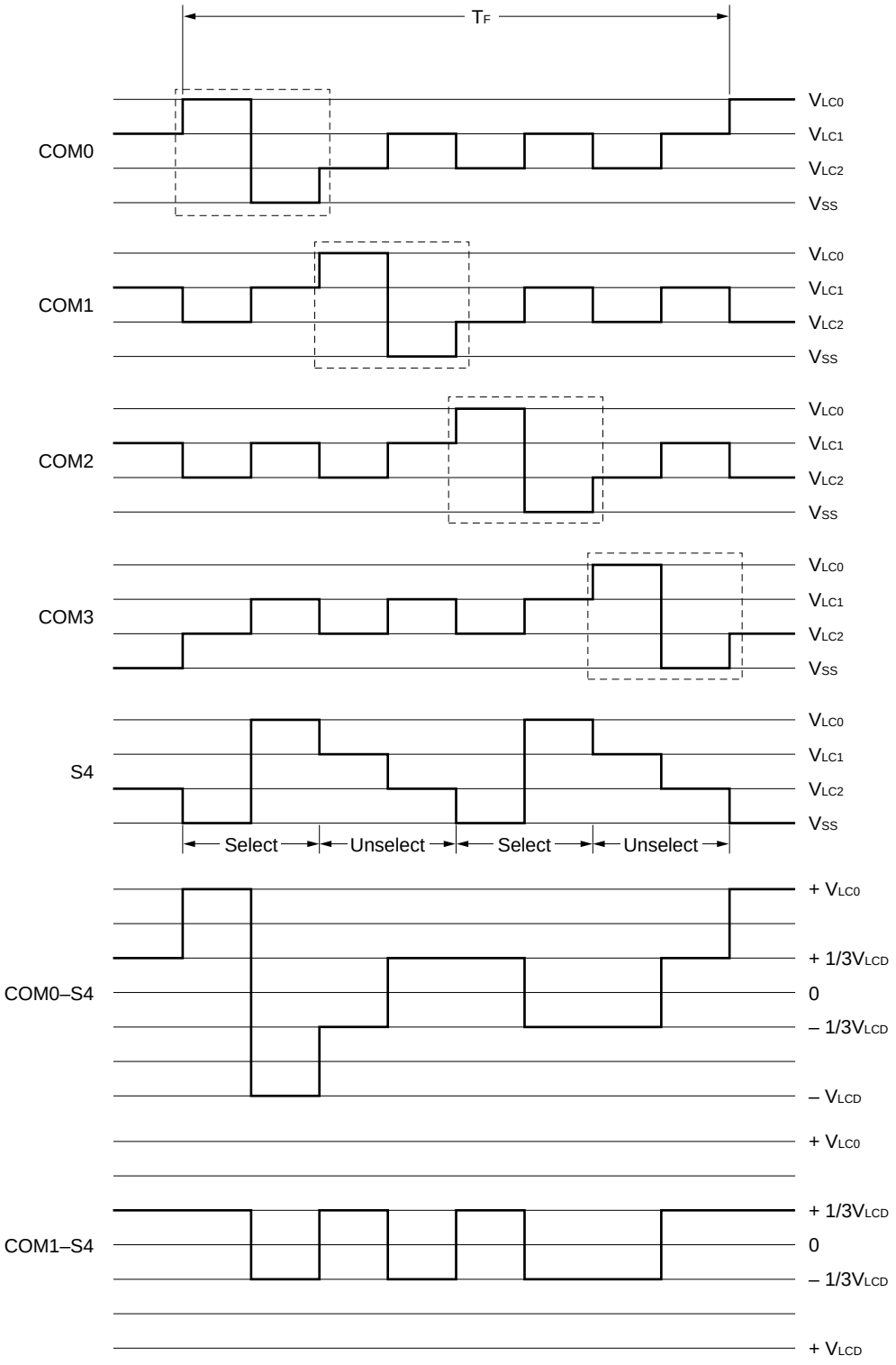


Figure 12-14. Example of 4-Time Division LCD Driving Waveform



Remarks 1. T_F: frame frequency

2. The valid waveform of each common signal is enclosed in dotted line.

For 4-time division LCD display, the valid timing (enclosed in dotted line in Figure 12-14) of each common signal is output in a cycle 1/4 of the frame frequency (T_F) as shown in Figure 12-14. In this timing,each segment signal is output to light or extinguish the LCD.

For example, segment signal S4 outputs a waveform that lights the LCD in the timing of COM0 and COM2, in respect to each common signal (COM0 through COM3) in Figure 12-14.

When the relations between each common signal and S4 is examined, it can be seen that a potential difference of $\pm V_{LCD}$ (LCD drive voltage) is generated at the COM0 select timing between COM0 and S4, as can be seen from the waveform of COM0-S4. In the case of COM2 and S4, a voltage difference of $\pm V_{LCD}$ (LCD drive voltage) is also generated between COM2 and S4 at the COM2 select timing. Therefore, the segment indicated by COM0, COM2 and S4 lights.

Because a voltage difference between COM1 and S4 is always $\pm 1/3\ V_{LCD}$ (COM1-S4 in Figure 12-14) at the select timing of COM1 (COM1 in Figure 12-14), the LCD remains dark.

(1) Description of package

<Public declaration symbol>

- Subroutine name
  S_4LCD: 4-time division display data storage routine
- Input parameter
  B_LCD : LCD display content storage buffer area
- Data definition reference name
  S0 : LCD display data memory reference address (FA7FH)

<Register used>

Bank 0; AX, DE, HL

<RAM used>

Name	Usage	Attribute	Bytes
B_LCD	LCD display data storage buffer area	SADDR	1
i	Display digit loop counter	SADDR	1
WORKP	Display data storage area address saving area	SADDRP	2

<Flag used>

None

<Nesting level>

1 level 2 bytes

<Hardware used>

- LCD controller/driver

<Initial setting>

- OSMS = #00000001B ; Oscillation mode select register
- TCL2 = #×××00×××B ; Count clock of watch timer = selects system clock
- TMC2 = #0××××1×B ; Supplies LCD clock (enables operation of prescaler)
- LCDC = #00000001B ; LCD display control register (LCD driving power is supplied from V_{DD} with segment signal pins S24 through S31 not used)
- LCDM = #10100000B ; LCD display mode register (sets 4-time division display, selects LCD clock, turns ON display)

Caution Set the initial value to the LCD display data memory (FA58H through FA7FH) before turning ON the LCD.

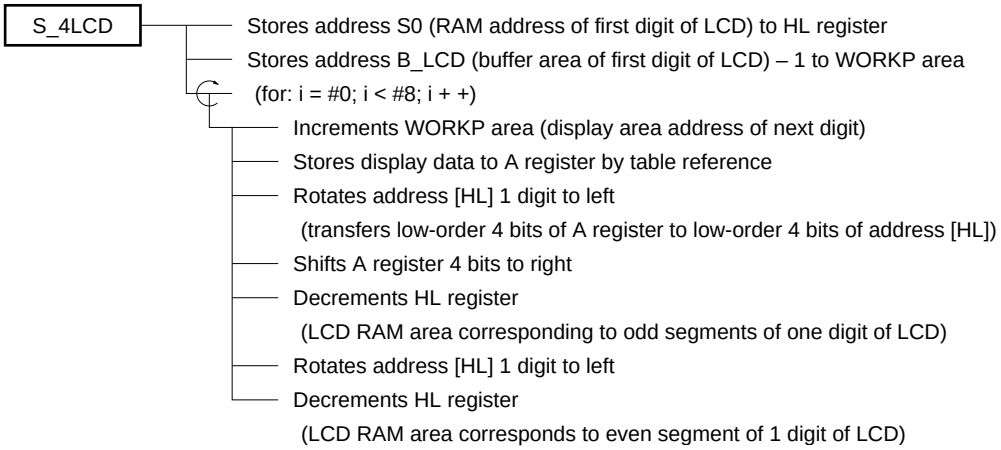
<Starting>

Set the display contents to the B_LCD area and call the S_4LCD routine.

(2) Example of use

```
EXTRN   S_4LCD,B_LCD,S0
        :
    HL=#S0                  ; Clears LCDRAM
    BC=#0FA80H-0FA58H       ;   (from FA58H to FA7FH)
    while(BC!=#0) (AX)      ;
        A=#0                ;
        [HL]=A              ;
        HL--                ;
        BC--                ;
    endw                    ;
;
    B_LCD=#0                ;
    B_LCD+1=#0              ;
    B_LCD+2=#0              ;
    B_LCD+3=#0              ;
;
    TCL2=#00000000B         ; Count clock of watch timer = selects main system clock
    TMC2=#00000010B         ; Enables prescaler operation
    LCDC=#00000001B         ; Supplies driving power from VDD with segments S24 through S31 not used
    LCDM=#10100000B         ; Turns on 4-time division display with 256-Hz clock selected, turns ON display
;
        :
      B_LCD+3=A             ;
        :
      B_LCD+2=A             ;
        :
      B_LCD+1=A             ;
        :
      B_LCD=A               ;
      CALL    !S_4LCD       ;
        :
```

(3) SPD chart



(4) Program list

```
PUBLIC  S_4LCD,B_LCD,S0

S0          EQU         0FA7FH                      ; 1st digit of LCD

LCD4RAM1    DSEG    SADDR
B_LCD:      DS      8                               ; Display BUF area
i:          DS      1                               ; Work counter
LCD4RAM2    DSEG    SADDRP
WO4RKP:     DS      2                               ; Work area
;*************************************************************
;      LCD display (4-time division display) processing
;*************************************************************
LSD4        CSEG
S_4LCD:
    HL=#S0                                          ; HL ← address S0
    WORKP=#B_LCD-1                                  ;
    for(i=#0;i<#8;i++)                              ;
        DE=WORKP (AX)                               ;
        DE++                                        ; References display data of contents of next digit
        WORKP=DE (AX)                               ;
        X=[DE] (A)                                  ;
        A=#0                                        ;
        AX+=#LCDDAT                                 ;
        DE=AX                                       ;
        A=[DE]                                      ;
        ROL4    [HL]                                ;  [HL] ← low-order 4 bits of A register
        A >>= 1                                     ; Shifts high-order 4 bits of A register to lower 4 bits
        A >>= 1                                     ;
        A >>= 1                                     ;
        A >>= 1                                     ;
        HL--                                        ; HL−−
        ROL4    [HL]                                ; [HL] ← low-order 4 bits of A register
        HL--                                        ; HL−−
    next                                            ;
    RET                                             ;
LCDDAT:
    DB  11010111B                                   ; 0
    DB  00000110B                                   ; 1
    DB  11100011B                                   ; 2
    DB  10100111B                                   ; 3
    DB  00110110B                                   ; 4
    DB  10110101B                                   ; 5
    DB  11110101B                                   ; 6
    DB  00010111B                                   ; 7
    DB  11110111B                                   ; 8
    DB  10110111B                                   ; 9
    DB  01110111B                                   ; A
    DB  11110100B                                   ; B
    DB  11010001B                                   ; C
    DB  11100110B                                   ; D
    DB  11110001B                                   ; E
    DB  01110001B                                   ; F
    DB  00000000B                                   ; Extinguishes
    END
```

CHAPTER 13 APPLICATIONS OF KEY INPUT

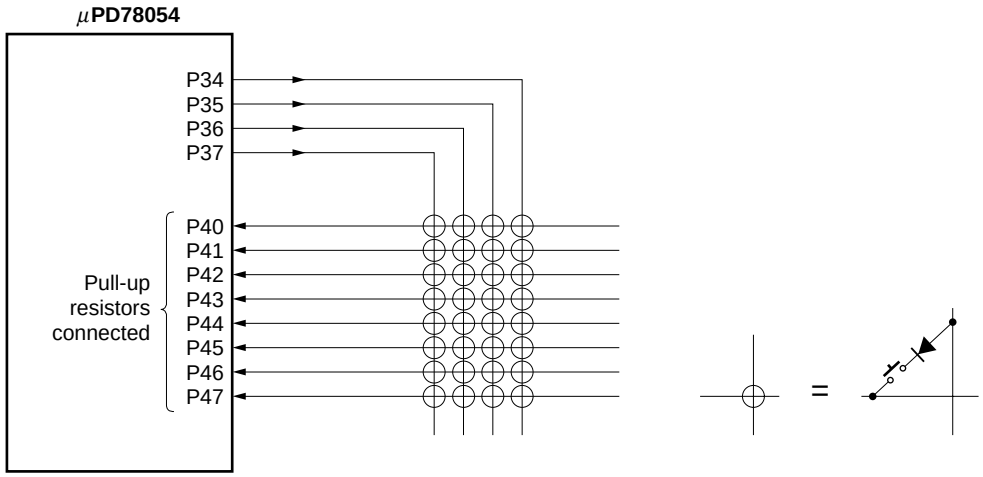
This chapter introduces an example of a program that inputs signals from a key matrix of 4 × 8 keys. The key scan be pressed successively, and two or more keys can be pressed simultaneously. In the circuit shown in this section, the high-order 4 bits of port 3 (P34 through P37) are used as key scan signals, and port 4[Note] is used as key return signals. As the pull-up resistor of port 4 for key return, the internal pull-up resistor set by software is used (refer to **Figure 13-1**).

Port 4 of the 78K/0 series has a function to detect the falling edges of the eight port pins in parallel. If port 4 is used for key return signals, therefore, the standby mode can be released through detection of a falling edge, i.e., by key input.

In this example, the μPD78054 subseries is used.

Note With the μPD78064, 78064Y, 780308, 780308Y, and 78064B subseries, port 11 is used instead of port 4.

Figure 13-1. Key Matrix Circuit



The input keys are stored to RAM on a one key-to-1 bit basis. The RAM bit corresponding to a pressed key is set and the bit corresponding to a released key is cleared. By testing the RAM data on a 1-bit-by-1-bit basis starting from the first bit, the key status can be checked. To absorb chattering, the key is assumed to be valid when four successive key codes coincide with a given code. For example, if a key code is sampled every 5 ms, chattering of 15 ms to 20 ms can be absorbed. If the key input is changed, a key change flag (KEYCHG) is set.

(1) Description of package

<Public declaration symbol>

KEYIN : Key input subroutine name
KEYDATA : Key data storage area
CHATCT : Chattering counter
KEYCHG : Key change test flag

<Register used>

AX, DE, HL

<RAM used>

Name	Usage	Attribute	Bytes
KEYDATA	Stores valid key data	SADDR	4
WORK	Stores key data during chattering		
CHATCT	Chattering counter		1
WORKCT	Loop processing work counter		

<Flag used>

Name	Usage
CHGFG	Set if key input changes
KEYCHG	Set if valid key changes
CHTEND	Confirms end of chattering

<Nesting>

1 level 2 bytes

<Hardware used>

- P4
- P3 (P34-P37)

<Initial setting>

- PUO4 = 1 ; Connects pull-up resistor to P4
- PM3 = #0000××××B ; Sets high-order 4 bits of P3 in output mode

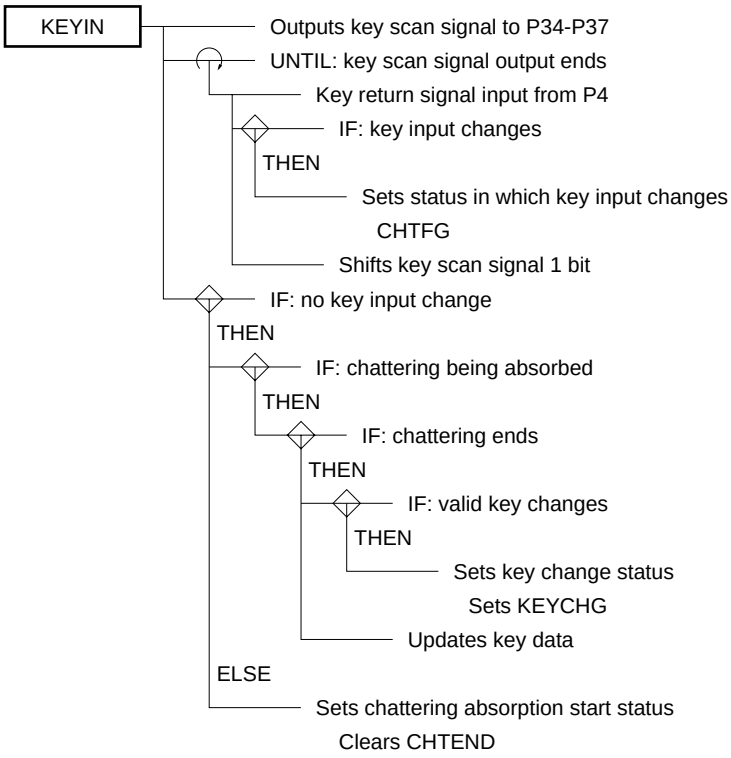
<Starting>

- Call KEYIN at specific intervals.
- Before inputting the key data, test the key change flag. The key change flag is not cleared by the subroutine. Clear the flag after it has been tested.

(2) Example of use

```
          EXTRN   KEYDATA,CHATCT,KEYIN
          EXTBIT  KEYCHG

VETM3     CSEG    AT 1EH
          DW      INTTM3                          ; Sets vector address of watch timer

MAINDAT   DSEG    SADDR
CT5MS:    DS      1
            .
            .
            .
          TMC2=#00100110B
          CLR1    TMMK3
          CT5MS=#3

          PM3=#00000000B                          ; Sets P3 in output mode
          SET1    PUO.4                           ; Pulls P4 up
          CHATCT=#3                               ; Initial setting of chattering counter
            .
            .
            .
          if_bit(KEYCHG)                          ; Key changed?
                  CLR1    KEYCHG
                  ; Key input processing
          endif
            .
            .
            .
;*****************************************
;       Watch timer interrupt processing
;       Interval time:  1.95 ms
;*****************************************
INTTM3:                                           ; 1.95 ms interrupt
            .
            .
            .
          DBNZ    CT5MS,$RTNTM3
          MOV     CT5MS,#3                        ; 1.95 ms × 3 elapses
          CALL    !ANKEYIN
RTNTM3:
            .
            .
            .
          RETI
```

(3) SPD chart


KEYIN
Outputs key scan signal to P34-P37
UNTIL: key scan signal output ends
Key return signal input from P4
IF: key input changes
THEN
Sets status in which key input changes
CHTFG
Shifts key scan signal 1 bit
IF: no key input change
THEN
IF: chattering being absorbed
THEN
IF: chattering ends
THEN
IF: valid key changes
THEN
Sets key change status
Sets KEYCHG
Updates key data
ELSE
Sets chattering absorption start status
Clears CHTEND

(4) Program list

```
        PUBLIC  KEYDATA,KEYCHG,KEYIN,CHATCT

KEY_DAT DSEG    SADDR
KEYDATA:DS      4                                   ; Key data storage area
WORK:   DS      4                                   ; Chattering key data
CHATCT: DS      1                                   ; Chattering counter
WORKCT: DS      1

KEY_FLG BSEG
CHGFG   DBIT                                        ; Key change status
KEYCHG  DBIT                                        ; Key changed
CHTEND  DBIT                                        ; Chattering absorption end status

KEY_SEG CSEG
;*****************************
*       Matrix key input
;*****************************
KEYIN:
        CLR1    CHGFG
        P3&=#00001111B
        P3|=#00010000B
        HL=#WORK                                    ; Sets address of key work area
        repeat
            A=P4
            A^=#11111111B                           ; Data inverted
            if(A!=[HL])                             ; Key changed?
                SET1    CHGFG
                [HL]=A
            endif
            HL++
            A=P3                                    ; Shifts key scan 1 bit
            A&=#11110000B
            X=A
            A=P3
            A+=X
            P3=A
        until_bit(CY)

        if_bit(!CHGFG)                              ; Key changed
            if_bit(!CHTEND)                         ; Chattering absorbed
                CHATCT--                            ; Chattering ends
                if(CHATCT==#0)
                    SET1    CHTEND
                    DE=#WORK
                    HL=#KEYDATA
                    for(WORKCT=#0;WORKCT<#4;WORKCT++)
                        if([DE]!=[HL]) (A)          ; Key changed
                            SET1    KEYCHG
                        endif
                        A<->[HL]                    ; Transfers WORK to KEYDATA
                        HL++
                        DE++
                    next
                endif
            endif
        else
            CHATCT=#3
            CLR1    CHTEND
        endif
        RET
```

[MEMO]

APPENDIX A DESCRIPTION OF SPD CHART

SPD stands for Structured Programming Diagrams.

Structuring means structuring the logical processing of a program, and designing and formulating the logic by using the basic structure of the logic elements.

All programs can be created by only combining the basic structure of logic elements, (sequentially, selectively, or repeatedly). (This is called a structured theorem). Through structuring, the flow of a program is clarified, and the reliability is improved. Although various methods are available for expressing the structuring of a program, NEC employs a diagram technique called SPD.

The following table describes the SPD symbols used for the SPD technique and compares them with flowchart symbols.

Table A-1. Comparison between SPD Symbols and Flowchart Symbol (1/2)

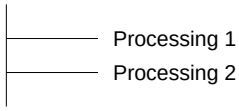
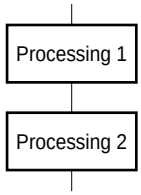
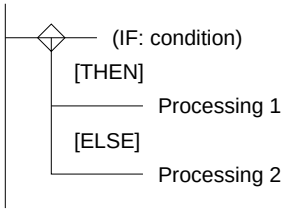
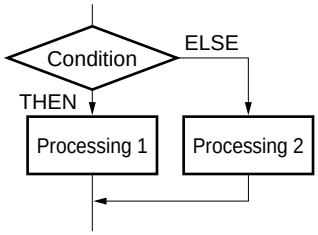
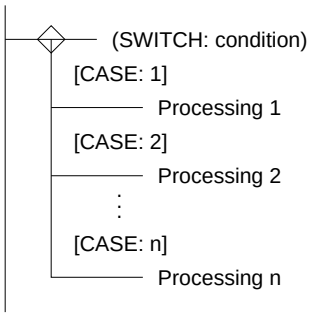
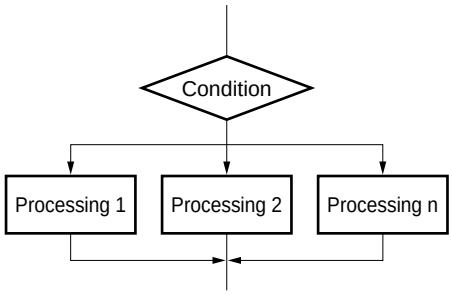
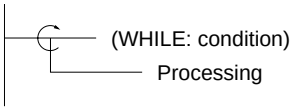
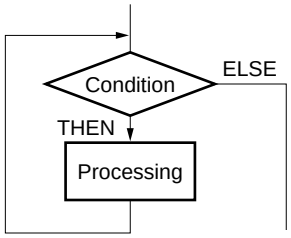
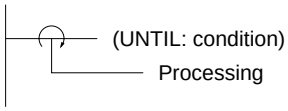
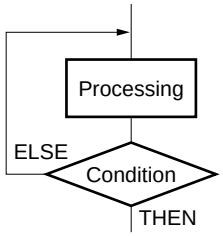
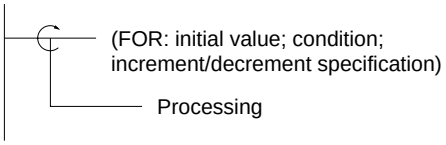
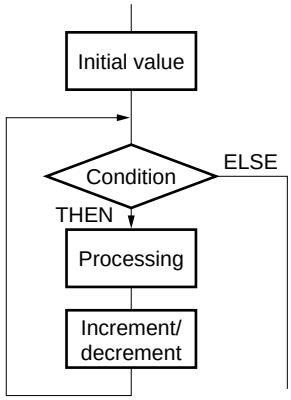
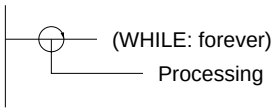
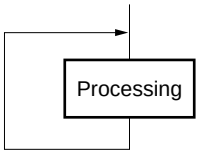
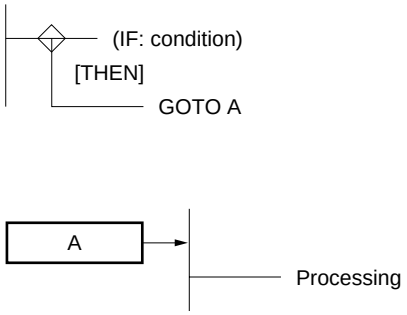
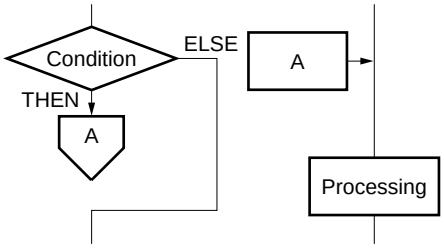
Processing Name	SPD Symbol	Flowchart Symbol
Sequential processing		
Conditional branch (IF)		
Conditional branch (SWITCH)		

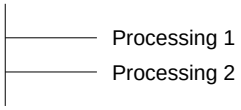
Table A-1. Comparison between SPD Symbols and Flowchart Symbol (2/2)

Processing Name	SPD Symbol	Flowchart Symbol
Conditional loop (WHILE)		
Conditional loop (UNTIL)		
Conditional loop (FOR)		
Infinite loop		
Connector		

1. Sequential processing

Sequential processing executes processing from top to bottom in the sequence in which processing appears.

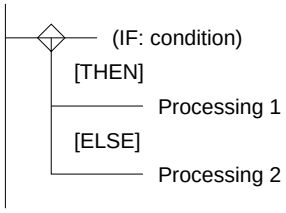
- **SPD chart**



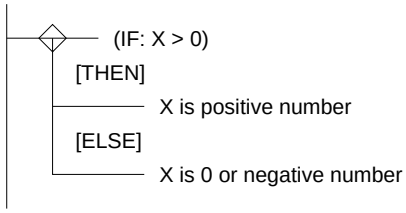
2. Conditional branch: 2 branch (IF)

Processing contents are selected according to the condition specified by IF is true or false (THEN/ELSE).

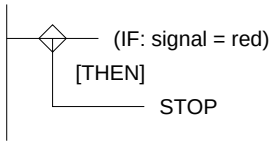
- **SPD chart**



Example 1. Identification of positive or negative of X



2. STOP if signal is red

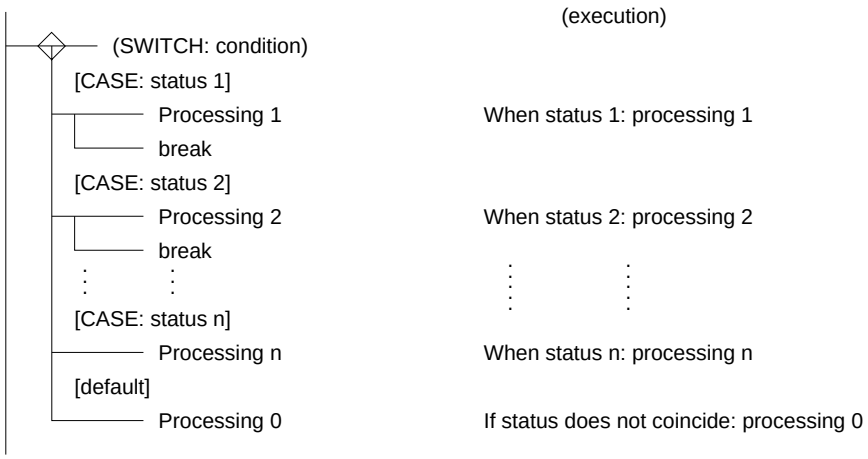


3. Conditional branch: multiple branch (SWITCH)

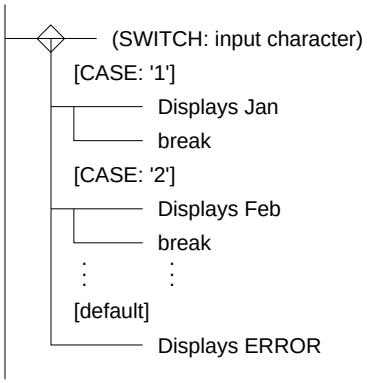
The condition specified by SWITCH is compared with the status indicated by CASE to select the processing. The processing of the SWITCH statement may be executed only when the given values coincide, or continued downward starting from when the given values coincide (if the processing is not continued downward, 'break' is described). If there is no coincide status, 'default' processing is executed (description of 'default' is arbitrary).

(1) Execution only on coincidence

- **SPD chart**

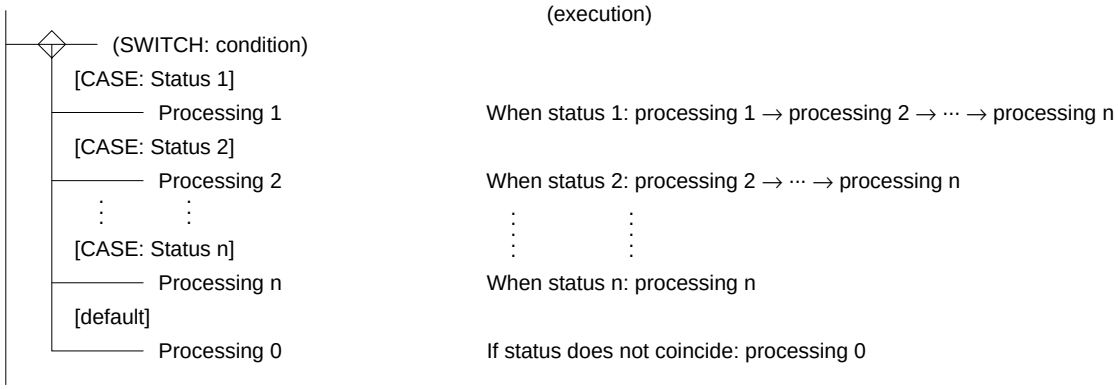


Example Displays name of month by input characters

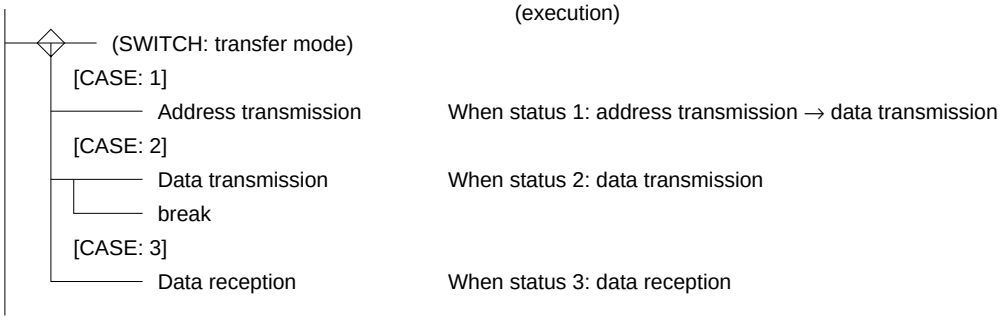


(2) If processing continues from coincidence status

- **SPD chart**



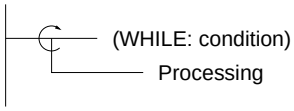
Example Transmission/reception of serial interface



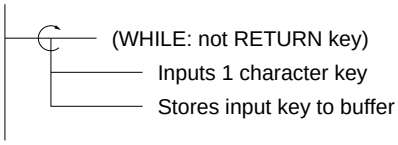
4. Conditional Loop (WHILE)

The condition indicated by WHILE is judged. If the condition is satisfied, processing is repeatedly executed (if the condition is not satisfied from the start, the processing is not executed).

- **SPD chart**



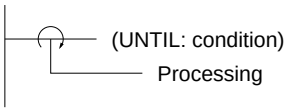
Example Buffers key until RETURN key is input



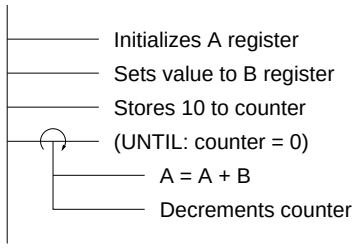
5. Conditional Loop (UNTIL)

The condition indicated by UNTIL is judged after processing has been executed, and the processing is repeatedly executed until a given condition is satisfied (even if the condition is not satisfied from the start, the processing is executed once).

- **SPD chart**



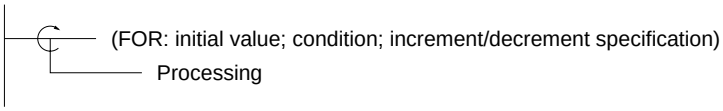
Example Multiplies value of B register by 10 and stores result to A register



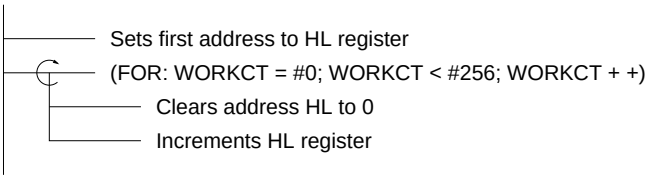
6. Conditional Loop (FOR)

While the condition of the parameter indicated by FOR is satisfied, processing is repeatedly executed.

- **SPD chart**



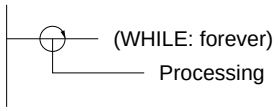
Example Clears 256 bytes to 0 starting from address HL



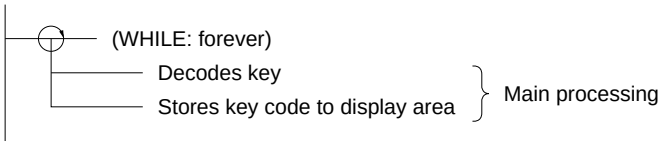
7. Infinite Loop

If 'forever' is set as the condition of WHILE, processing is infinitely executed.

- **SPD chart**



Example To execute main processing repeatedly

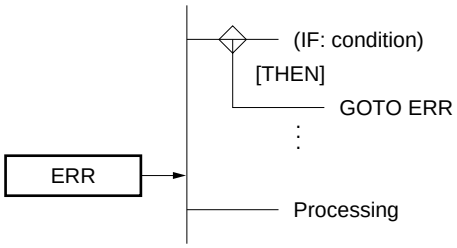


8. Connector (GOTO)

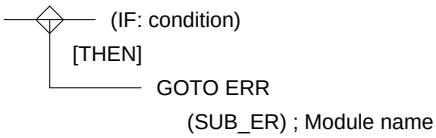
Unconditionally branches to a specified address.

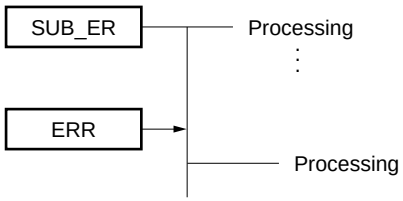
- **SPD chart**

(1) To branch to same module

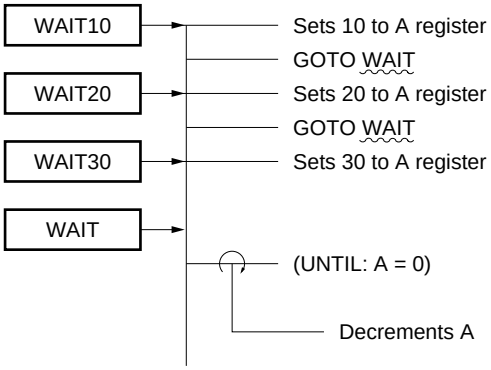


(2) To branch to different module





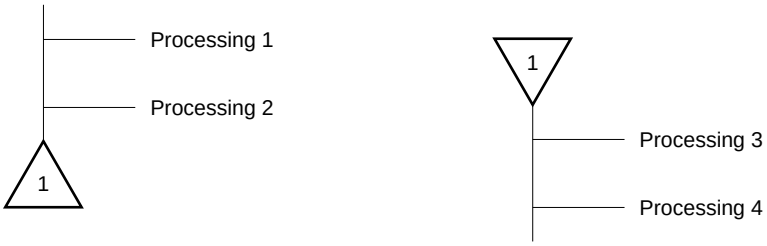
Example To select a parameter at the start address of a subroutine and set wait state



9. Connector (continuation)

Used when the SPD of one module requires two or more pages to indicate the flow of processing.

- **SPD chart**



★

APPENDIX B REVISION HISTORY

The revision history of this document is as follows. “Chapter” indicates the chapter number in the preceding edition.

(1/2)

Edition	Major Revision from Preceding Edition	Chapter
2nd edition	Addition of following products as target products: μPD780018, 780018Y, 780058, 780058Y, 780308, 780308Y, 78058F, 78058FY, 78064B, 78075B, 78075BY, 78098B subseries, μPD78070A, 78070AY μPD78052(A), 78053(A), 78054(A) μPD78062(A), 78063(A), 78064(A) μPD78081(A), 78082(A), 78P083(A), 78081(A2) μPD78058F(A), 78058FY(A) μPD78064B(A)	Throughout
	Deletion of following products as target products: μPD78P054Y, 78P064Y, 78074, 78075, 78074Y, 78075Y	
	Addition of Note 2 and Caution 2 to Figure 4-5 Format of Watchdog Timer Mode Register	CHAPTER 4 APPLICATION OF WATCHDOG TIMER
	Addition of Caution to Figure 5-8 Format of External Interrupt Mode Register 0	CHAPTER 5 APPLICATION OF 16-BIT TIMER/EVENT COUNTER
	Addition of Table 8-2 Items Supported by Each Subseries	CHAPTER 8 APPLICATION OF SERIAL INTERFACE
	Addition of Table 8-3 Registers of Serial Interface	
	Addition of note on using wake-up function and note on changing operation mode to Figures 8-7 and 8-8 Format of Serial Operating Mode Register 0	
	Addition of Caution to Figures 8-16 and 8-17 Format of Automatic Data Transmission/Reception Interval Specification Register	
	Addition of Figures 8-23 and 8-24 Format of Serial Interface Pin Select Register	
	μPD6252 as maintenance product in 8.1 Interface with EEPROM™ (μPD6252)	
	Addition of (5) Limitations when using I^2C bus mode to 8.1.2 Communication in I^2C bus mode	
	Addition of (f) Limitations when using UART mode to 8.5 Interface in Asynchronous Serial Interface (UART) Mode	

(2/2)

Edition	Major Revision from Preceding Edition	Chapter
2nd edition	Description of following register formats and tables for each subseries: Figures 8-14 and 8-15 Format of Automatic Data Transfer/ Reception Control Register Tables 8-4, 8-5, and 8-6 Setting of Operating Modes of Serial Interface Channel 2 Figures 12-1 and 12-2 Format of LCD Display Mode Register	CHAPTER 8 APPLICATION OF SERIAL INTERFACE
	Addition of Figure 11-3 Format of Port Mode Register 12	CHAPTER 11 APPLICATION OF REAL-TIME OUTPUT PORT
	Description of following register formats for each subseries: Figures 12-1 and 12-2 Format of LCD Display Mode Register	CHAPTER 12 APPLICATION OF LCD CONTROLLER/DRIVER

NEC

Facsimile Message

From:

Name

Company

Tel. FAX

Address

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

Thank you for your kind support.

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

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

South America
NEC do Brasil S.A.
Fax: +55-11-889-1689

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

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

Taiwan
NEC Electronics Taiwan Ltd.
Fax: 02-719-5951

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

Japan
NEC Corporation
Semiconductor Solution Engineering Division
Technical Information Support Dept.
Fax: 044-548-7900

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

Document title: _____

Document number: _____ Page number: _____

If possible, please fax the referenced page or drawing.

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