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

MOS INTEGRATED CIRCUIT

# μPD17147GS

## SMALL GENERAL-PURPOSE 4 BIT SINGLE-CHIP MICROCONTROLLER

The μPD17147GS is a 4-bit single-chip microcontroller containing an 8-bit A/D converter (four channels), three timers, and a serial interface.

For the CPU, the μPD17147GS uses general registers. It can perform operations directly on data memory and transfer data directly between memories, allowing programming to be done efficiently. Each instruction is 16 bits (one word) long.

Since the μPD17147GS has an on-chip A/D converter and serial interface, it is suitable for electronic control in various applications such as electric home appliances. The μPD17P149, a one-time PROM product, is available for evaluation of the μPD17147GS and for small-scale production of general electronic equipment.

Note that the number of pins and the package of the μPD17P149 are different from those of the μPD17147GS.

**The following user's manual completely describes the functions of the μPD17147GS. Be sure to read it before designing an application system.**

**μPD17145 Sub-Series User's Manual: IEU-1383**

### FEATURES

- 17K architecture : General registers, 16-bit instructions
- Program memory (ROM) : 4K bytes (2048 × 16 bits)
- Data memory (RAM) : 110 × 4 bits
- External interrupt : 1 line (INT pin, with sensor input)
- Instruction execution time : 2 μs (at fx = 8 MHz, ceramic oscillation)
- 8-bit A/D converter : 4 channels
  Absolute accuracy: ±1.5 LSB or lower ($V_{DD}$ = 4.0 to 5.5 V)
- Timer function : 3 channels
- Serial interface : 1 channel (three-wire synchronous mode)
- POC circuit (mask option)
- Supply voltage : $V_{DD}$ = 2.7 to 5.5 V (at fx = 400 kHz to 2 MHz)
  $V_{DD}$ = 4.5 to 5.5 V (at fx = 400 kHz to 8 MHz)

### APPLICATIONS

Electric appliances, battery chargers, cameras, electronic measuring instruments



Document No. U10961EJ1V0DS00 (1st edition)
Date Published March 1996 P
Printed in Japan

## ORDERING INFORMATION

| Part number | Package |
|---|---|
| μPD17147GS-xxx-GJG | 30-pin plastic shrink SOP (300 mil) |

**Remark** xxx is a ROM code number.

## FUNCTIONS

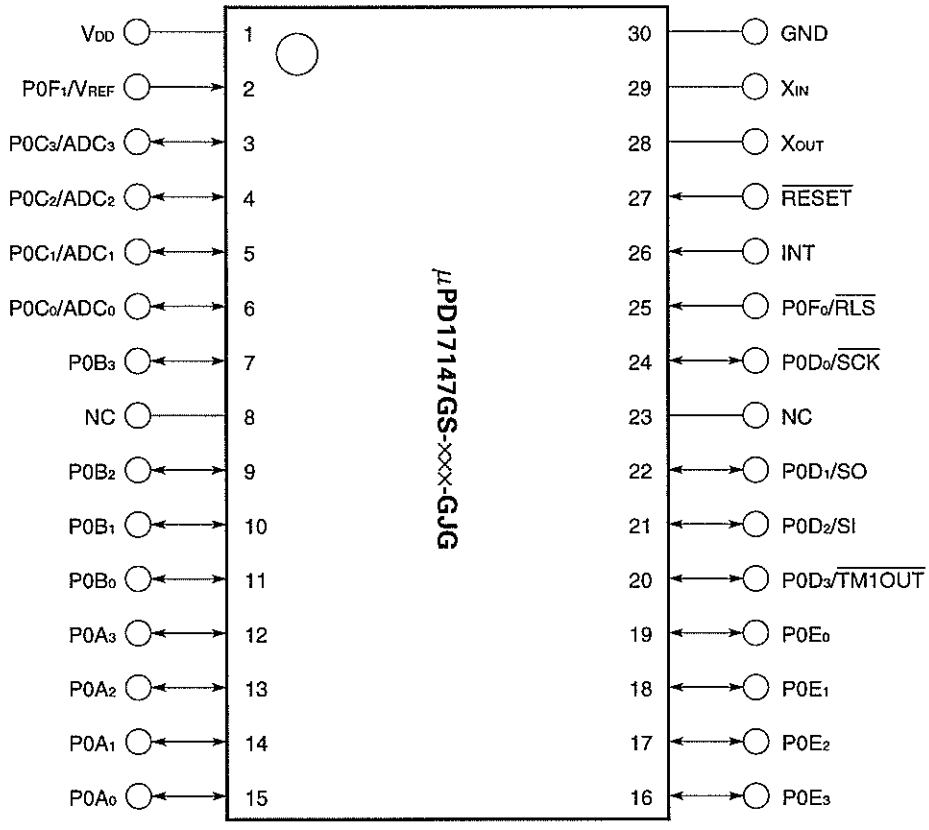
| Item \ Product | μPD17147GS |
|---|---|
| ROM | 4K bytes (2048 × 16 bits) |
| RAM | 110 × 4 bits |
| Stack | 5 address stacks, 3 interrupt stacks |
| Number of I/O ports | 23 { • 20 I/O ports<br>• 2 general input ports<br>• 1 sensor input port (INT pin[Note]) |
| A/D converter input | 4 channels (shared with ports) with an absolute accuracy of ±1.5 LSB or less |
| Timer | 3 channels { • 2 channels for 8-bit timer counter (They can be used together as one 16-bit timer.)<br>• 1 channel for 7-bit basic interval timer (can be used as a watchdog timer) |
| Serial interface | 1 channel (3-wire type) |
| Interrupt | • Up to 3 levels of multiple hardware interrupt<br>• 1 external interrupt (INT) { Detection of the rising edge / Detection of the falling edge / Detection of both edges } Selectable<br>• 4 internal interrupts { • Timer 0 (TM0)<br>• Timer 1 (TM1)<br>• Basic interval timer (BTM)<br>• Serial interface (SIO) |
| Execution time of an instruction | 2 μs (when operating at fx = 8 MHz with ceramic oscillation) |
| Standby function | HALT/STOP |
| POC circuit | Mask option<br>(Can be used in an application circuit where $V_{DD}$ is 5 V ±10 % and fx = 400 kHz to 4 MHz) |
| Supply voltage | $V_{DD}$ = 2.7 to 5.5 V (at fx = 400 kHz to 2 MHz)<br>$V_{DD}$ = 4.5 to 5.5 V (at fx = 400 kHz to 8 MHz) |
| Package | 30-pin plastic shrink SOP (300 mil) |
| One-time PROM | μPD17P149<br>• The package of the μPD17P149 is different from that of the μPD17147GS.<br>28-pin plastic shrink DIP (400 mil)<br>28-pin plastic SOP (375 mil) |

**Note** The INT pin can be used as an input pin (sensor input) when the external interrupt function is not used. The status of the pin is read with the INT flag of the control register, not with the port register.

**Caution Although a PROM product is highly compatible with a masked ROM product in respect of functions, they differ in internal ROM circuits and part of electrical characteristics. Before changing the PROM product to the masked ROM product in an application system, evaluate the system carefully using the masked ROM product.**

## PIN CONFIGURATION (TOP VIEW)

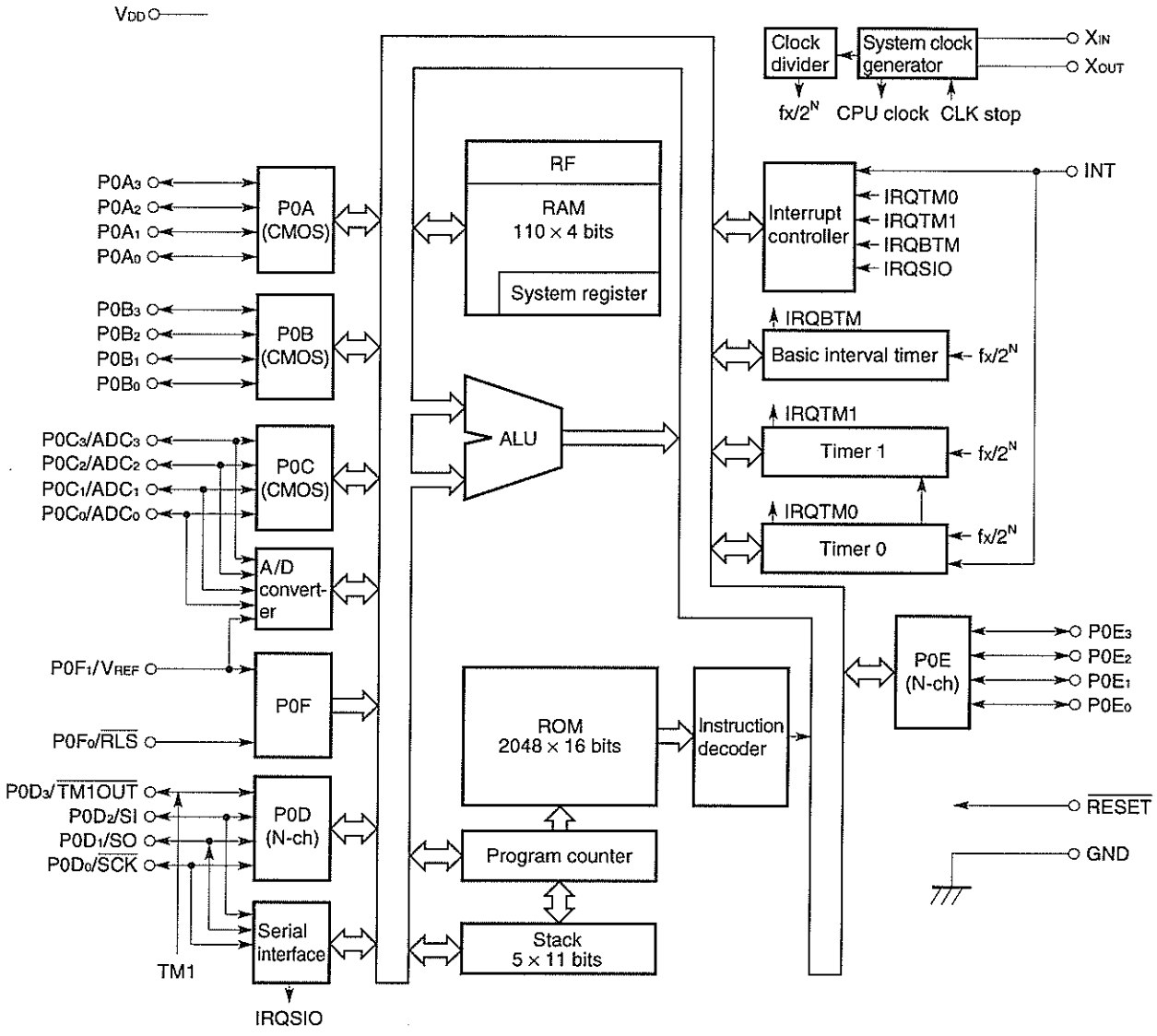
### 30-pin plastic shrink SOP (300 mil)




ADC0 - ADC3 : Analog input
GND : Ground
INT : External interrupt input
NC : No connection
P0A0 - P0A3 : Port 0A
P0B0 - P0B3 : Port 0B
P0C0 - P0C3 : Port 0C
P0D0 - P0D3 : Port 0D
P0E0 - P0E3 : Port 0E
P0F0 and P0F1 : Port 0F

$\overline{\text{RESET}}$ : Reset input
$\overline{\text{RLS}}$ : Standby release signal input
$\overline{\text{SCK}}$ : Serial clock input/output
SI : Serial data input
SO : Serial data output
$\overline{\text{TM1OUT}}$ : Timer 1 output
VDD : Power supply
VREF : Reference voltage for the A/D converter
XIN, XOUT : System clock oscillation

## BLOCK DIAGRAM




**Remark** The terms CMOS and N-ch in parentheses indicate the output form of the port.

CMOS : CMOS push-pull output

N-ch : N-channel open-drain output

CONTENTS

# 1. PINS

## 1.1 PIN FUNCTIONS

(1/2)

| Pin No. | Symbol | Function | Output | Upon reset |
|---|---|---|---|---|
| 1 | $V_{DD}$ | Power supply | – | – |
| 2 | $P0F_1/V_{REF}$ | Port 0F. The reference voltage is supplied to the A/D converter through this pin.<br>• Pull-up resistor incorporation specifiable by mask option<br>• $P0F_1$<br>• Bit 1 of 2-bit input port P0F<br>• $V_{REF}$<br>• Reference voltage input for the A/D converter | Input | Input ($P0F_1$) |
| 3 - 6 | $P0C_3/ADC_3$ - $P0C_0/ADC_0$ | Port 0C. Analog voltage is supplied to the A/D converter through these pins.<br>• $P0C_3$ - $P0C_0$<br>• 4-bit input/output port<br>• Input/output setting allowed in units of 1 bit<br>• $ADC_3$ - $ADC_0$<br>• Analog input for the A/D converter | CMOS push-pull | Input (P0C) |
| 7 | $P0B_3$ | Port 0B<br>• 4-bit input/output port<br>• Input/output setting allowed in units of 4 bits<br>• Pull-up resistor incorporation specifiable by program in units of 4 bits<br>• Pin 8 is left open. | CMOS push-pull | Input |
| 8 | NC | | – | – |
| 9<br>10<br>11 | $P0B_2$<br>$P0B_1$<br>$P0B_0$ | | CMOS push-pull | Input |
| 12<br>13<br>14<br>15 | $P0A_3$<br>$P0A_2$<br>$P0A_1$<br>$P0A_0$ | Port 0A<br>• 4-bit input/output port<br>• Input/output setting allowed in units of 4 bits<br>• Pull-up resistor incorporation specifiable by program in units of 4 bits | CMOS push-pull | Input |
| 16<br>17<br>18<br>19 | $P0E_3$<br>$P0E_2$<br>$P0E_1$<br>$P0E_0$ | Port 0E<br>• Withstand voltage is $V_{DD}$ (Max.).<br>• 4-bit input/output port<br>• Input/output setting allowed in units of 4 bits<br>• Pull-up resistor incorporation specifiable by program in units of 4 bits | N-ch open drain | Input |

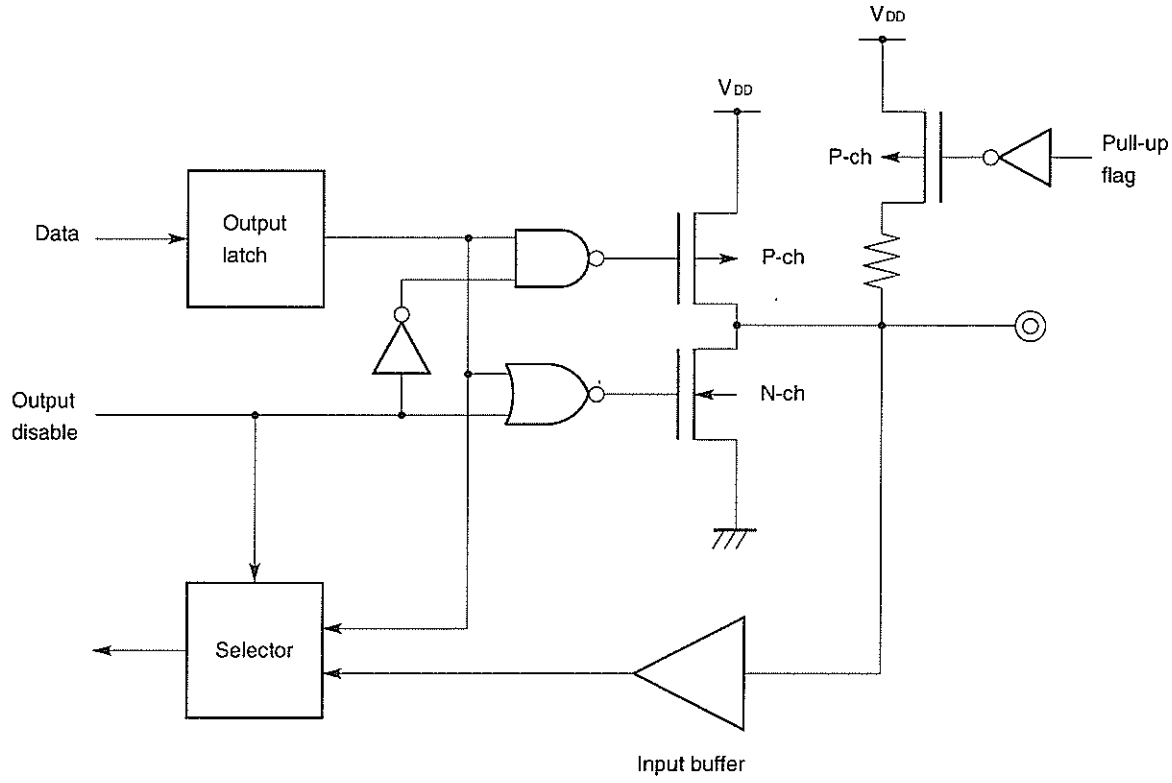
(1/2)

| Pin No. | Symbol | Function | Output | Upon reset |
|---|---|---|---|---|
| 20 | P0D3/$\overline{TM1OUT}$ | Pin for port 0D, timer 1 output, serial data input, serial data output, and serial clock input/output.<br>• Pull-up resistor incorporation specified by program bit by bit<br>• Withstand voltage is $V_{DD}$ (Max.).<br>• P0D3 - P0D0<br>• 4-bit input/output port<br>• Input/output setting allowed bit by bit<br>• $\overline{TM1OUT}$<br>• Timer 1 output | N-ch open drain | Input (P0D) |
| 21 | P0D2/SI | • SI<br>• Serial data input | | |
| 22 | P0D1/SO | • SO<br>• Serial data output | | |
| 23 | NC | • $\overline{SCK}$<br>• Serial clock input/output<br>• Pin 23 is left open. | – | – |
| 24 | P0D0/$\overline{SCK}$ | | N-ch open drain | Input (P0D) |
| 25 | P0F0/$\overline{RLS}$ | Pin for port 0F and input for standby mode release signal<br>• Pull-up resistor incorporation specifiable by mask option<br>• P0F0<br>• Bit 0 of 2-bit input port P0F<br>• $\overline{RLS}$<br>• Input for standby mode release signal | Input | Input (P0F0) |
| 26 | INT | Input for an external interrupt request signal and standby mode release signal.<br>• Pull-up resistor incorporation specifiable by mask option | Input | Input |
| 27 | $\overline{RESET}$ | System reset input pin<br>• Pull-up resistor incorporation specifiable by mask option | Input | Input |
| 28<br>29 | $X_{OUT}$<br>$X_{IN}$ | For system clock oscillation<br>The ceramic resonator is connected between $X_{IN}$ and $X_{OUT}$. | – | – |
| 30 | GND | Ground | – | – |

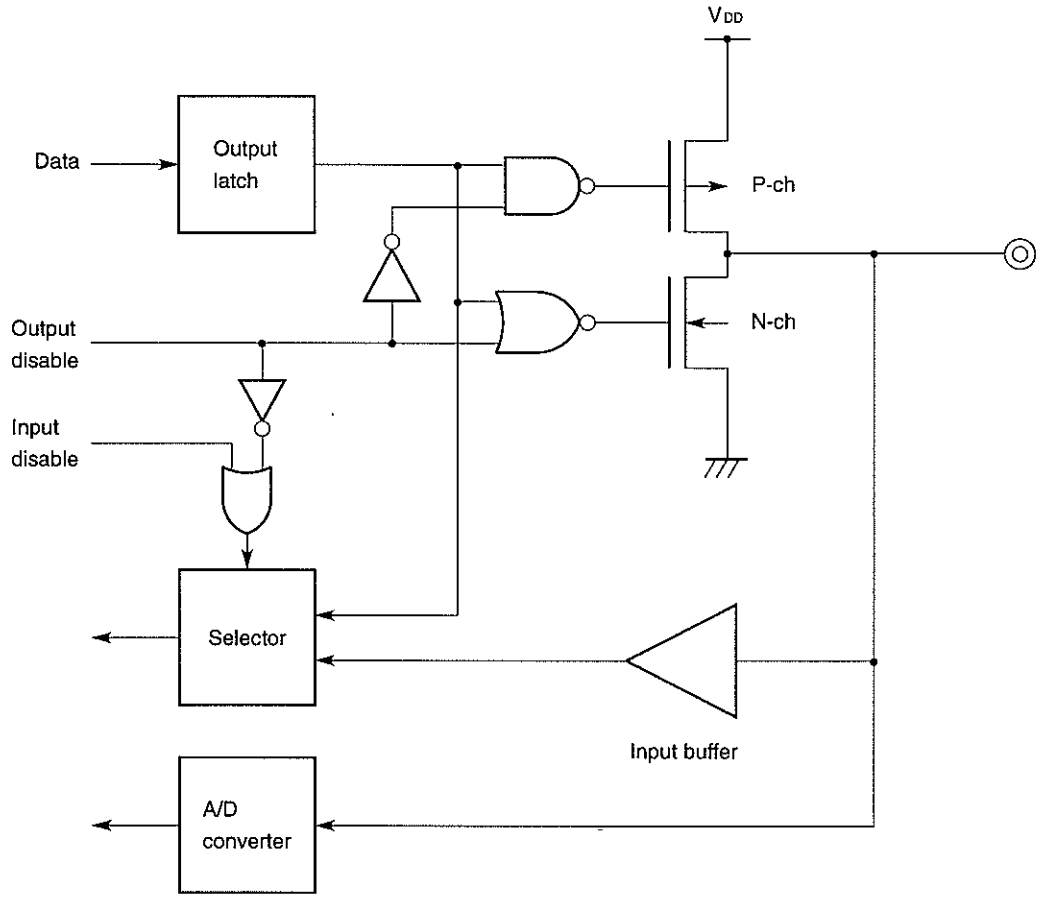
### 1.2 EQUIVALENT INPUT/OUTPUT CIRCUITS

Below are simplified diagrams of the input/output circuits for each pin.

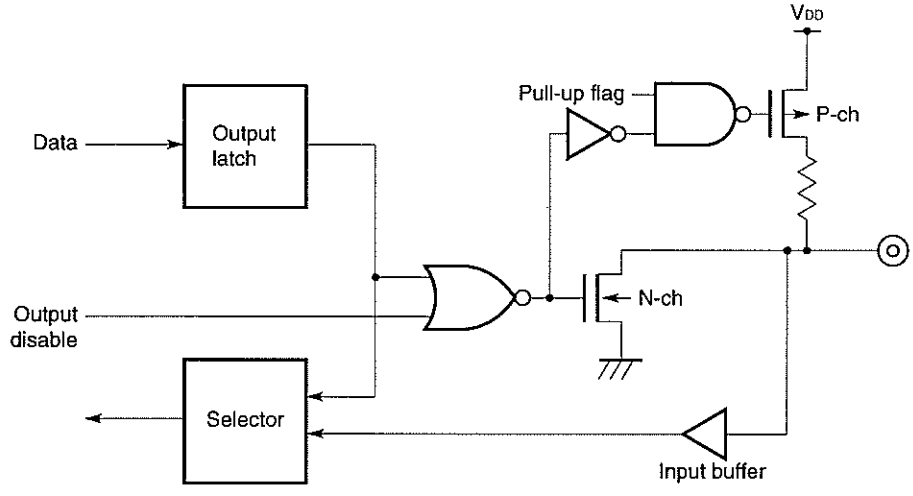
**(1) $P0A_0$ - $P0A_3$, $P0B_0$ - $P0B_3$**



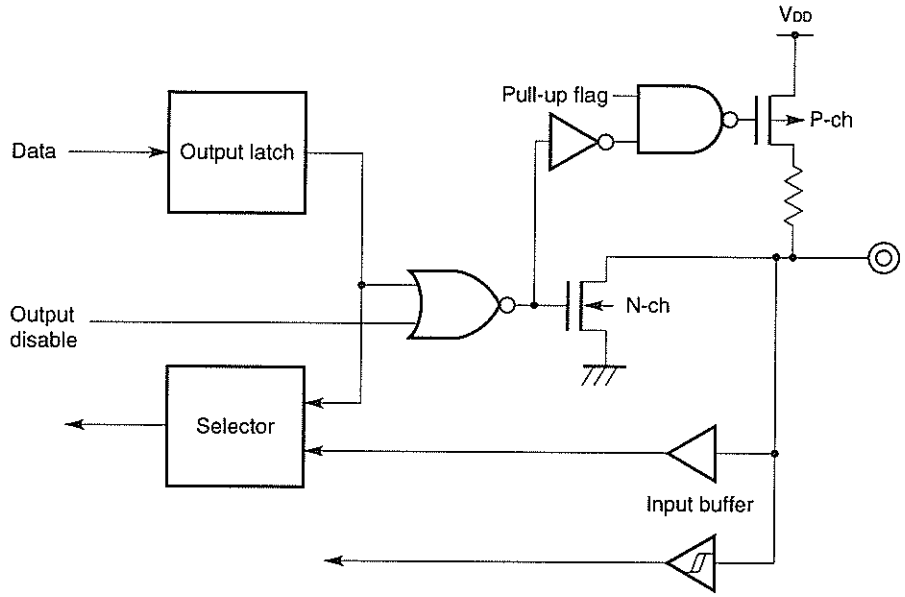
(2) P0C0/ADC0 - P0C3/ADC3



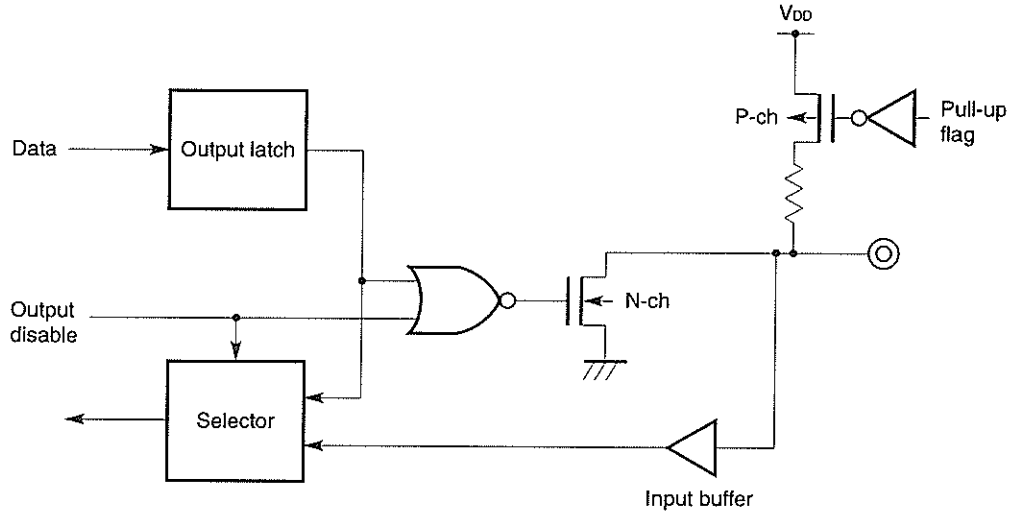

(3) P0D3/$\overline{\text{TM1OUT}}$, P0D1/SO



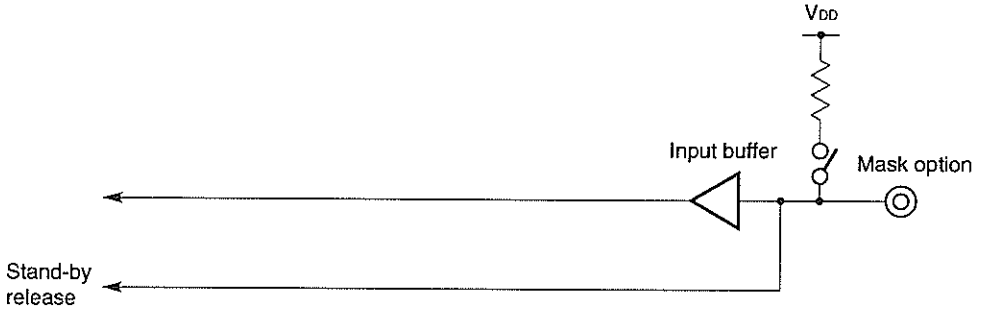
(4) P0D2/SI, P0D0/$\overline{SCK}$



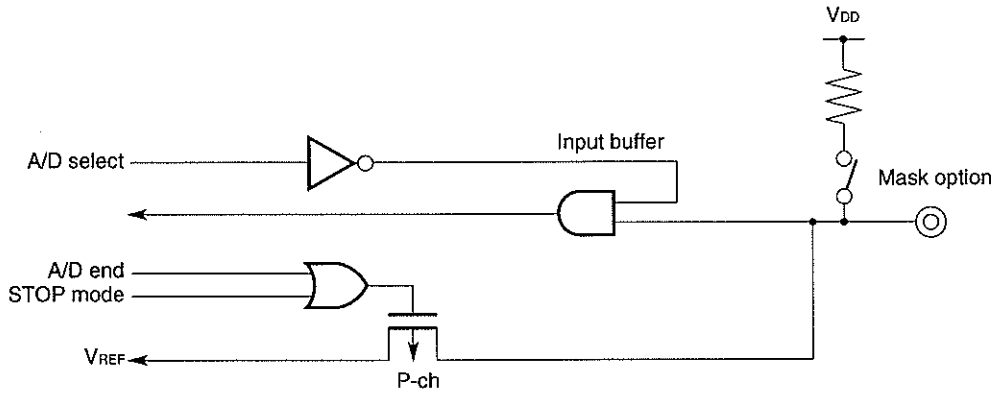

(5) P0E0 - P0E3



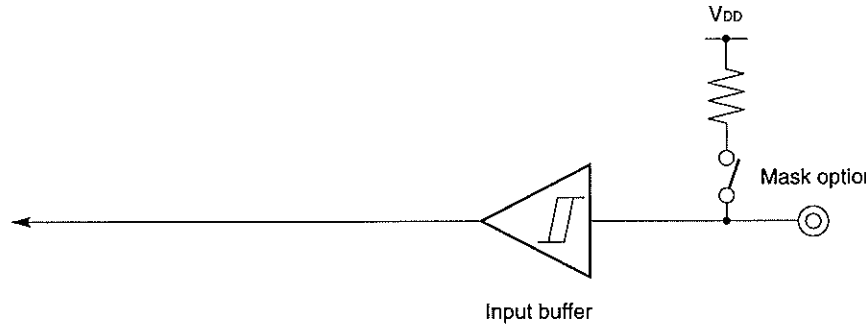

(6) P0F0/$\overline{RLS}$



(7) P0F1/VREF




(8) $\overline{\text{RESET}}$, INT



### 1.3 HANDLING UNUSED PINS

Connect unused pins as follows:

**Table 1-1 Handling Unused Pins**

| Pin | | | Conditions and handling | |
|---|---|---|---|---|
| | | | Internal | External |
| Port | Input mode | P0A, P0B, P0D, P0E | Pull-up resistors that can be specified with the software are incorporated. | Leave open. |
| | | P0C | — | Connect to $V_{DD}$ through pull-up resistors. Or, connect to ground through pull-down resistors.Note 1 |
| | | $P0F_1$ | Pull-up resistors that can be specified with the mask option are not incorporated. | Connect directly to $V_{DD}$ or ground. |
| | | | Pull-up resistors that can be specified with the mask option are incorporated. | Leave open. |
| | | $P0F_0$Note 2 | Pull-up resistors that can be specified with the mask option are not incorporated. | Connect directly to ground. |
| | Output mode | P0A, P0B, P0C (CMOS ports) | — | Leave open. |
| | | P0D (N-ch open-drain port) | Outputs low level. | |
| | | P0E (N-ch open-drain port) | Outputs low level without pull-up resistors that can be specified with the software. | |
| | | | Outputs low level with pull-up resistors that can be specified with the software. | |
| External interrupt (INT) | | | Pull-up resistors that can be specified with the mask option are not incorporated. | Connect directly to $V_{DD}$ or ground. |
| | | | Pull-up resistors that can be specified with the mask option are incorporated. | Leave open. |
| $\overline{RESET}$Note 3 (When use only internal POC circuit) | | | Pull-up resistors that can be specified with the mask option are not incorporated. | Connect directly to $V_{DD}$. |
| | | | Pull-up resistors that can be specified with the mask option are incorporated. | |

**Notes 1.** When a pin is pulled up to $V_{DD}$ or pulled down to ground outside the chip, take the driving capacity and maximum current consumption of a port into consideration. When using high-resistance pull-up or pull-down resistors, apply appropriate countermeasures to ensure that noise is not attracted by the resistors. Although the optimum pull-up or pull-down resistor varies with the application circuit, in general, a resistor of 10 to 100 kilohms is suitable.

**Notes 2.** Since the P0F0/$\overline{RLS}$ pin is also used for setting the test mode, connect it directly to ground without incorporating a pull-up resistor that can be specified with the mask option, when the pin is not used.

3. When designing an application circuit which requires high reliability, be sure to design a circuit to which an external $\overline{RESET}$ signal can be input. Since the $\overline{RESET}$ pin is also used for setting the test mode, connect it to $V_{DD}$ directly when not used.

**Caution To fix the I/O mode, pull-up resistors that can be specified with the software, and output level of a pin, it is recommended that they should be specified repeatedly within a loop in a program.**

### 1.4 NOTES ON USE OF THE $\overline{RESET}$ AND P0F0/$\overline{RLS}$ PINS

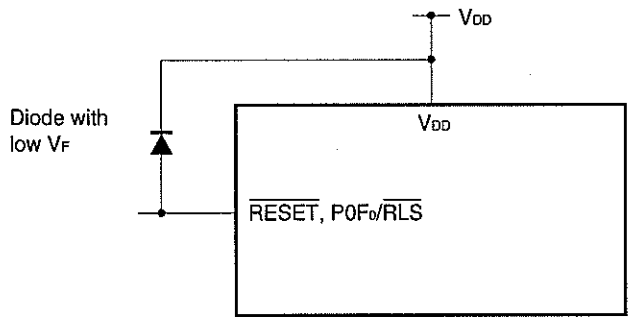
The $\overline{RESET}$ and P0F0/$\overline{RLS}$ pins have the test mode selecting function for testing the internal operation of the μPD17147GS (IC test), besides the functions shown in **Section 1.1**.

Applying a voltage exceeding $V_{DD}$ to the $\overline{RESET}$ and/or P0F0/$\overline{RLS}$ pin causes the μPD17147GS to enter the test mode. When noise exceeding $V_{DD}$ comes in during normal operation, the device is switched to the test mode.

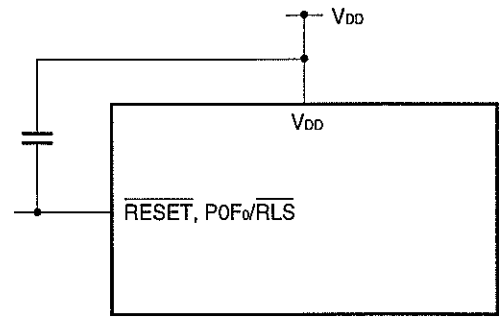
For example, if the wiring from the $\overline{RESET}$ or P0F0/$\overline{RLS}$ pin is too long, noise may be induced on the wiring, causing this mode switching.

When installing the wiring, lay the wiring in such a way that noise is suppressed as much as possible. If noise yet arises, use an external part to suppress it as shown below.

- **Connect a diode with low $V_F$ between the pin and $V_{DD}$.**




- **Connect a capacitor between the pin and $V_{DD}$.**



## 2. PROGRAM MEMORY (ROM)

Table 2-1 lists the program memory configuration.

**Table 2-1 Program Memory Configuration**

| Product | Program memory capacity | Address range |
|---|---|---|
| μPD17147GS | 4K bytes (2048 × 16 bits) | 0000H - 07FFH |

Program memory stores the program and the constant data table.

The program memory address is specified by the program counter.

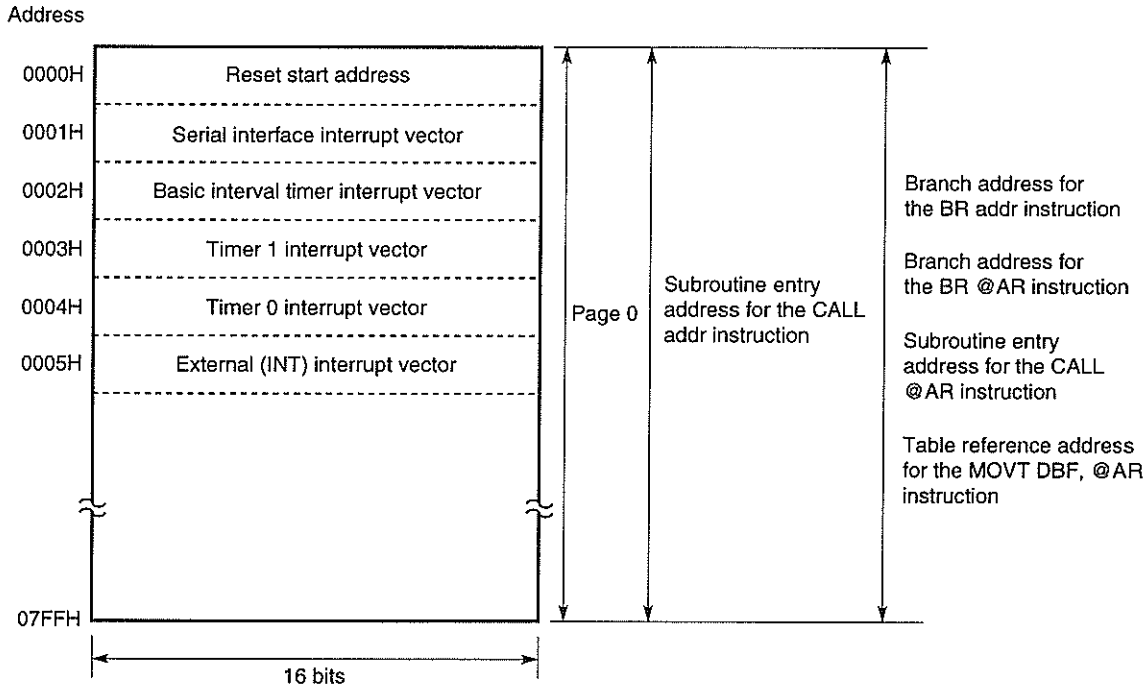
The reset start address and interrupt vector addresses are assigned to program memory 0000H to 0005H.

### 2.1 PROGRAM MEMORY CONFIGURATION

Fig. 2-1 shows the program memory map. A step consists of 16 bits of program memory. A 2K-step area is called a page.

Direct subroutine calls can specify address 0000H to 07FFH (page 0) in program memory. Branch instructions, indirect subroutine calls, and table references can specify any address in program memory.

**Fig. 2-1 Program Memory Map**



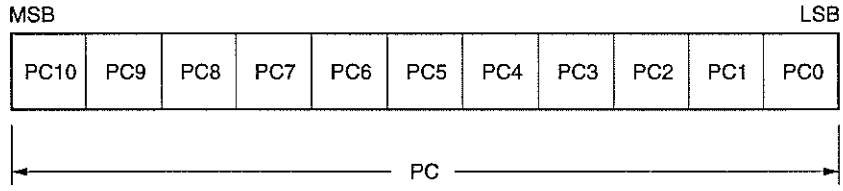
## 3. PROGRAM COUNTER (PC)

The program counter is used to specify an address in program memory.

### 3.1 PROGRAM COUNTER CONFIGURATION

As shown in Fig. 3-1, the program counter is a 11-bit binary counter.

**Fig. 3-1 Program Counter**




### 3.2 PROGRAM COUNTER OPERATION

Normally, the program counter is automatically incremented each time a command is executed. The memory address at which the next instruction to be executed is stored is assigned to the program counter under the following conditions: At reset; when a branch, subroutine call, return, or table referencing instruction is executed; or when an interrupt is received.

**Fig. 3-2 Value of the Program Counter After an Instruction Is Executed**

| Instruction \ Program counter bit | Program counter value | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | PC10 | PC9 | PC8 | PC7 | PC6 | PC5 | PC4 | PC3 | PC2 | PC1 | PC0 |
| During reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| BR addr | Value set by addr | | | | | | | | | | |
| CALL addr | | | | | | | | | | | |
| BR @AR<br>CALL @AR<br>(MOVT DBF, @AR) | Value in the address register (AR) | | | | | | | | | | |
| RET<br>RETSK<br>RETI | Value in the address stack location pointed to by the stack pointer (return address) | | | | | | | | | | |
| During interrupt | Vector address for the interrupt | | | | | | | | | | |

## 4. STACK

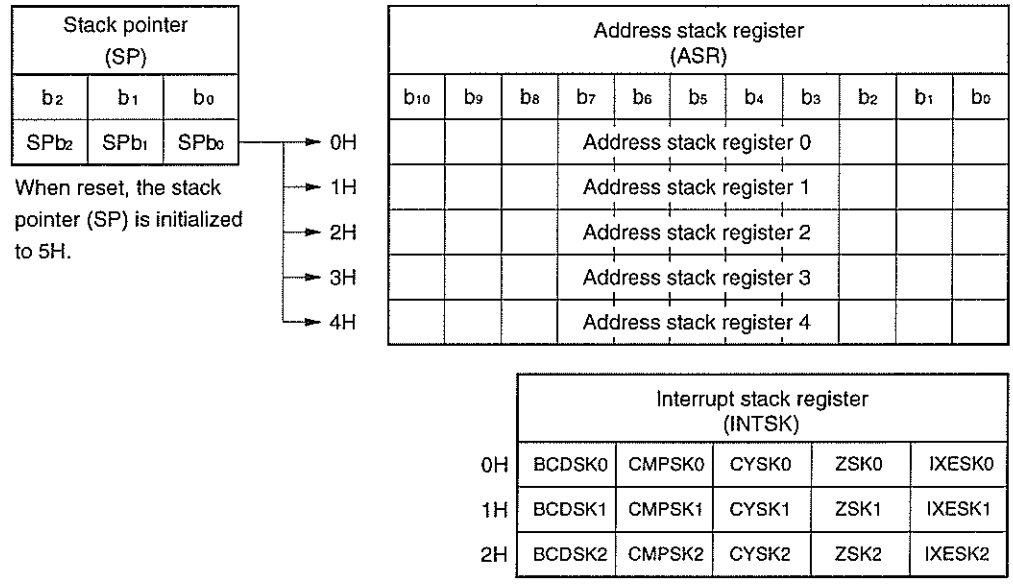
The stack is a register used to save information such as the program return address and the contents of the system register during execution of subroutine calls, interrupts and similar operations.

### 4.1 STACK CONFIGURATION

Fig. 4-1 shows the configuration of the stack.

The stack consists of the following parts: one 3-bit binary counter stack pointer (SP), five 11-bit address stack registers (ASR), and three 5-bit interrupt stack registers (INTSK).

**Fig. 4-1 Stack Configuration**




### 4.2 FUNCTIONS OF THE STACK

The stack is used to save the return address during execution of subroutine calls and table reference instructions. When an interrupt occurs, the program return address and the program status word (PSWORD) are automatically saved in the stack. Then, all bits of the PSWORD are cleared to 0.

## 5. DATA MEMORY (RAM)

Data memory (RAM) stores data such as operation and control data. Data can be read from or written to data memory with an instruction during normal operation.

### 5.1 DATA MEMORY CONFIGURATION

Data memory locations have 7-bit addresses. The three high-order bits of each address are called the row address, and the four low-order bits are called the column address.

For example, the row address of address 1AH is 1H. The column address is 0AH.

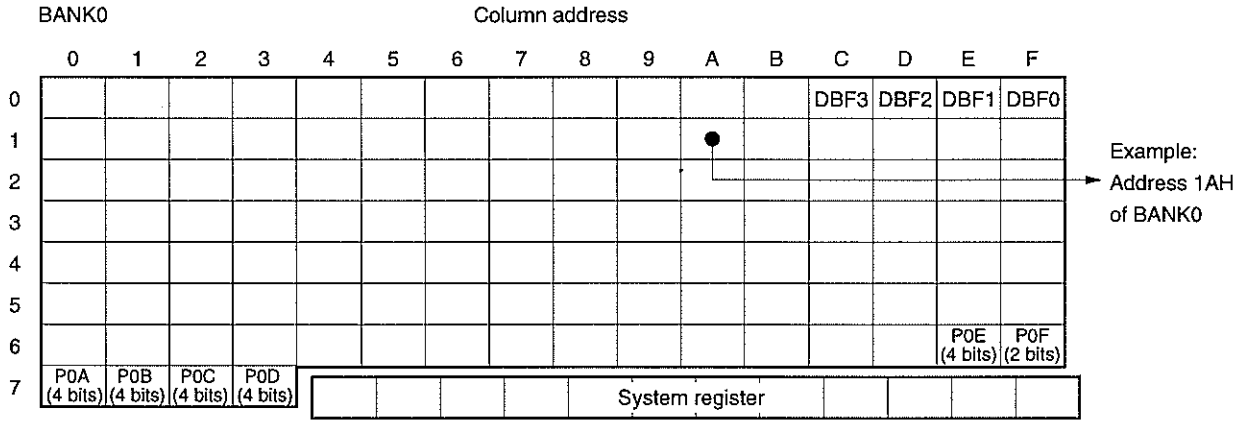
Each addressed memory location is 4-bits (one nibble) long.

Data memory contains an area to which the user is allowed to store data freely, as well as areas which are reserved for the use of specific functions.

The areas reserved for specific functions are as follows:

- System register (SYSREG) (See **Chapter 7.**)
- Data buffer (DBF) (See **Chapter 9.**)
- Port registers (See **Chapter 11.**)

**Fig. 5-1 Organization of Data Memory**



## 6. GENERAL REGISTER (GR)

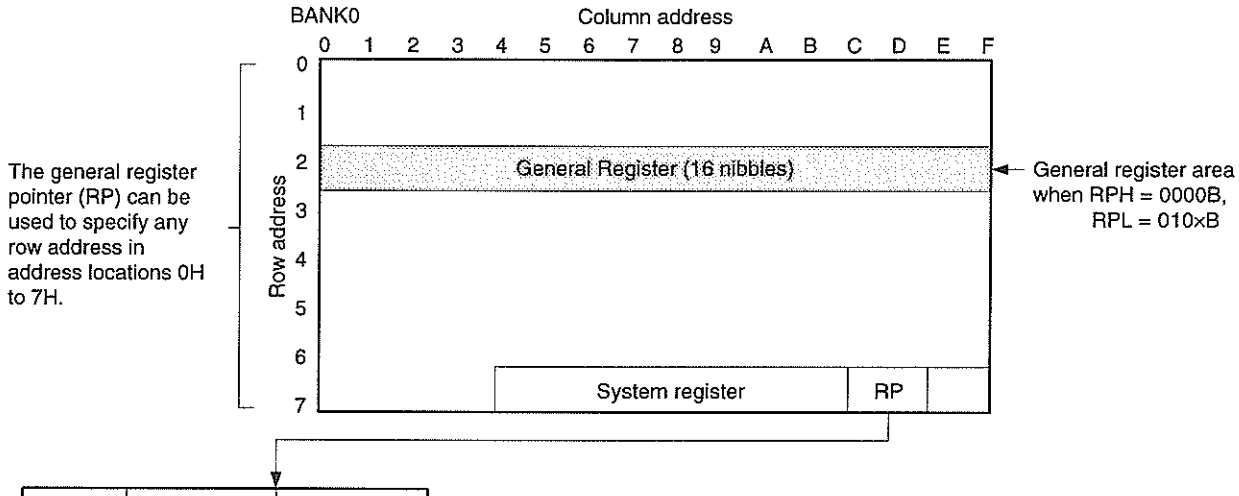
The general register, as the name implies, is a general register used for data transfer and manipulation. In the 17K series, the location of the general register is not fixed. The area used for the general register is in data memory, as specified by the general register pointer (RP). Thus, part of the data memory area can be specified as the general register as required, allowing data transfer in data memory and data memory manipulation to be performed with a single instruction.

### 6.1 GENERAL REGISTER POINTER (RP)

RP is a pointer used to specify part of data memory as the general register. In RP, specify a desired data memory bank and row address for the general register. RP consists of seven bits: 7DH (RPH), and the three high-order bits of 7EH (RPL) in the system register (see **Chapter 7**).

Set a bank in RPH, and a data memory row address in RPL.

**Fig. 6-1 General Register Pointer Configuration**




| Address | 7DH | | | | 7EH | | | |
|---|---|---|---|---|---|---|---|---|
| Name | General register pointer (RP) | | | | | | | |
| Symbol | RPH | | | | RPL | | | |
| Bits | $b_3$ | $b_2$ | $b_1$ | $b_0$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ |
| Data | 0 | 0 | 0 | 0 | | | | BCD |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

## 7. SYSTEM REGISTER (SYSREG)

The system register (SYSREG), located in data memory, is used for direct control of the CPU.

### 7.1 SYSTEM REGISTER CONFIGURATION

Fig. 7-1 shows the allocation address of the system register in data memory. As shown in Fig. 7-1, the system register is allocated in addresses 74H to 7FH of data memory.

Since the system register is allocated in data memory, it can be manipulated using any of the instructions available for manipulating data memory. Therefore, it is also possible to put the system register in the general register.

**Fig. 7-1 Allocation of System Register in Data Memory**

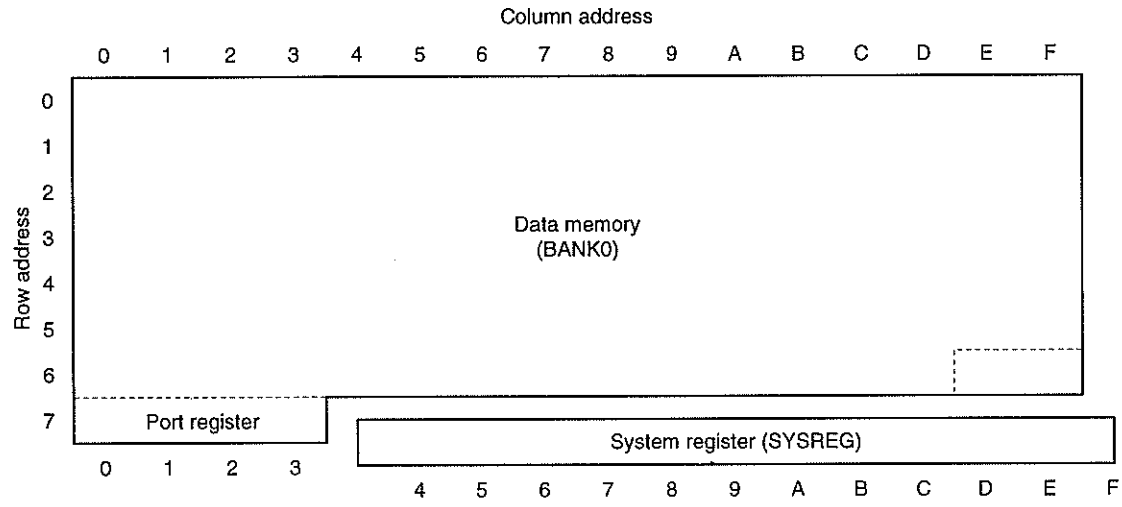



Fig. 7-2 shows the configuration of the system register. As shown in Fig. 7-2, the system register consists of the following seven registers.

- Address register (AR)
- Window register (WR)
- Bank register (BANK)
- Index register (IX)
- Data memory row address pointer (MP)
- General register pointer (RP)
- Program status word (PSWORD)

**Fig. 7-2 System Register Configuration**

| Address | 74H | 75H | 76H | 77H | 78H | 79H | 7AH | 7BH | 7CH | 7DH | 7EH | 7FH |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Address register (AR) | | | | Window register (WR) | Bank register (BANK) | Index register (IX) Data memory row address pointer (MP) | | | General register pointer (RP) | | Program status word (PSWORD) |
| Symbol | AR3 | AR2 | AR1 | AR0 | WR | BANK | IXH / MPH | IXM / MPL | IXL | RPH | RPL | PSW |
| Bit | $b_3$ $b_2$ $b_1$ $b_0$ | $b_3$ $b_2$ $b_1$ $b_0$ | $b_3$ $b_2$ $b_1$ $b_0$ | $b_3$ $b_2$ $b_1$ $b_0$ | $b_3$ $b_2$ $b_1$ $b_0$ | $b_3$ $b_2$ $b_1$ $b_0$ | $b_3$ $b_2$ $b_1$ $b_0$ | $b_3$ $b_2$ $b_1$ $b_0$ | $b_3$ $b_2$ $b_1$ $b_0$ | $b_3$ $b_2$ $b_1$ $b_0$ | $b_3$ $b_2$ $b_1$ $b_0$ | $b_3$ $b_2$ $b_1$ $b_0$ |
| Data[Note] | 0 0 0 0 (AR) | 0 (AR) | (AR) | (AR) | (WR) | 0 0 0 0 (BANK) | MPE 0 0 0 (MP) | (IX) (MP) | (IX) | 0 0 0 0 (RP) | (RP) | BCD CMP CY Z IXE |
| Initial value when reset | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | Not defined | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 |

**Note** A bit indicating zero is fixed to zero.

## 8. REGISTER FILE (RF)

The register file is a register used mainly for specifying conditions for peripheral hardware.

### 8.1 REGISTER FILE CONFIGURATION

#### 8.1.1 Configuration of the Register File

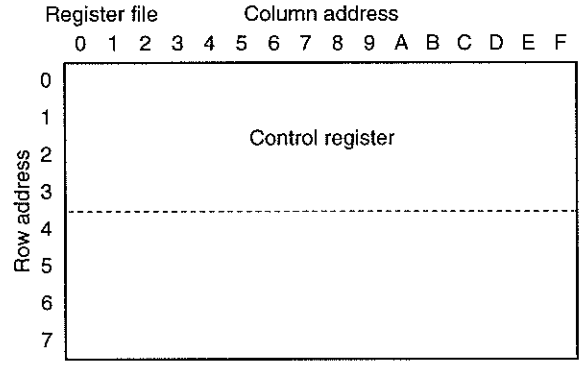
Fig. 8-1 shows the configuration of the register file.

As shown in Fig. 8-1, the register file is a register consisting of 128 nibbles (128 × 4 bits).

In the same way as with data memory, the register file is divided into addresses in units of four bits. It has a total of 128 nibbles specified in row addresses from 0H to 7H and column addresses from 0H to 0FH.

Address locations 00H to 3FH define an area called the control register.

**Fig. 8-1 Register File Configuration**



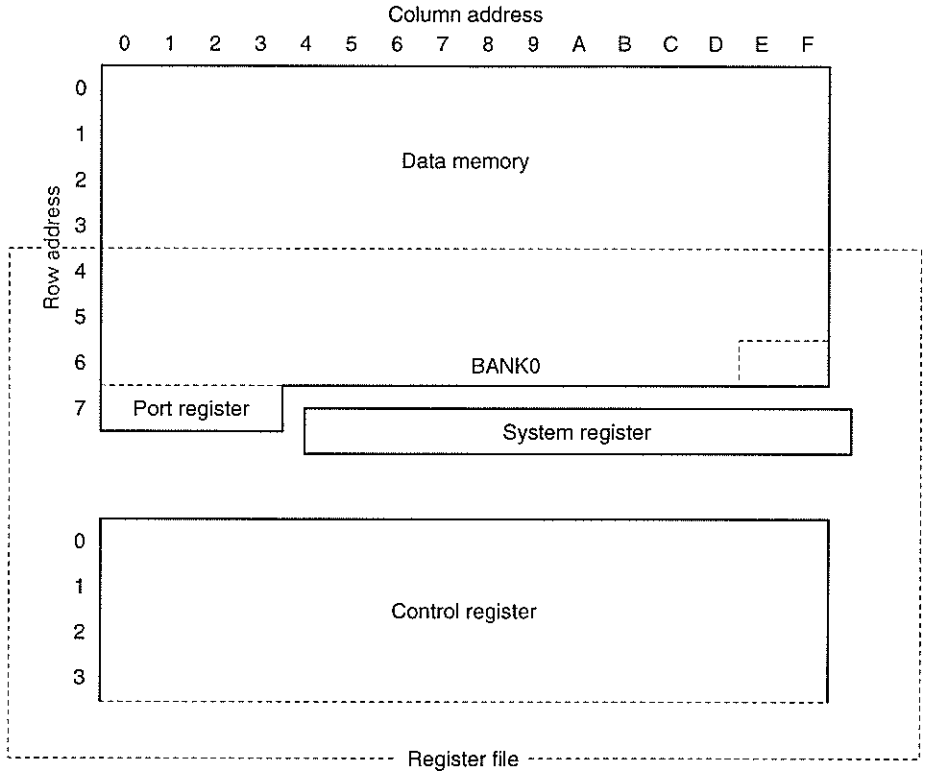

#### 8.1.2 Relationship between the Register File and Data Memory

Fig. 8-2 shows the relationship between the register file and data memory.

As shown in Fig. 8-2, the register file overlaps with data memory at addresses 40H to 7FH.

This means that, on a program, it seems that the same memory exists in the register file at addresses 40H to 7FH and in the data memory at addresses 40H to 7FH.

Fig. 8-2 Relationship between the Register File and Data Memory




## 8.2 FUNCTIONS OF THE REGISTER FILE

### 8.2.1 Functions of the Register File

The register file is a collection of registers in which peripheral hardware conditions are set with the PEEK instruction or POKE instruction.

The register used to control the peripheral hardware is located at addresses 00H to 3FH. This area is called the control register.

Addresses 40H to 7FH of the register file constitute normal data memory. Thus, not only the MOV instruction, but also the PEEK and POKE instructions, can be used to enable this part to perform read and write operations.

### 8.2.2 Control Register Functions

The peripheral hardware whose conditions can be controlled by control registers is listed below.

For details concerning peripheral hardware and the control register, see the section for the peripheral hardware concerned.

- Ports
- 8-bit timer counter (TM0, TM1)
- Basic interval timer (BTM)
- A/D converter
- Serial interface (SIO)
- Interrupt function
- Stack pointer (SP)

## 9. DATA BUFFER (DBF)

The data buffer consists of four nibbles allocated in addresses 0CH to 0FH in BANK0.

The data buffer acts as a data storage area for the CPU peripheral hardware (address register, serial interface, timer 0, timer 1, and A/D converter) through use of the GET and PUT instructions. It also acts as data storage used for receiving and transferring data. By using the MOVT DBF, @AR instruction, fixed data in program memory can be read into the data buffer.

### 9.1 DATA BUFFER CONFIGURATION

Fig. 9-1 shows the allocation of the data buffer in data memory.

As shown in Fig. 9-1, the data buffer is allocated in address locations 0CH to 0FH in data memory and consists of four nibbles (4 × 4 bits).

**Fig. 9-1 Allocation of the Data Buffer**

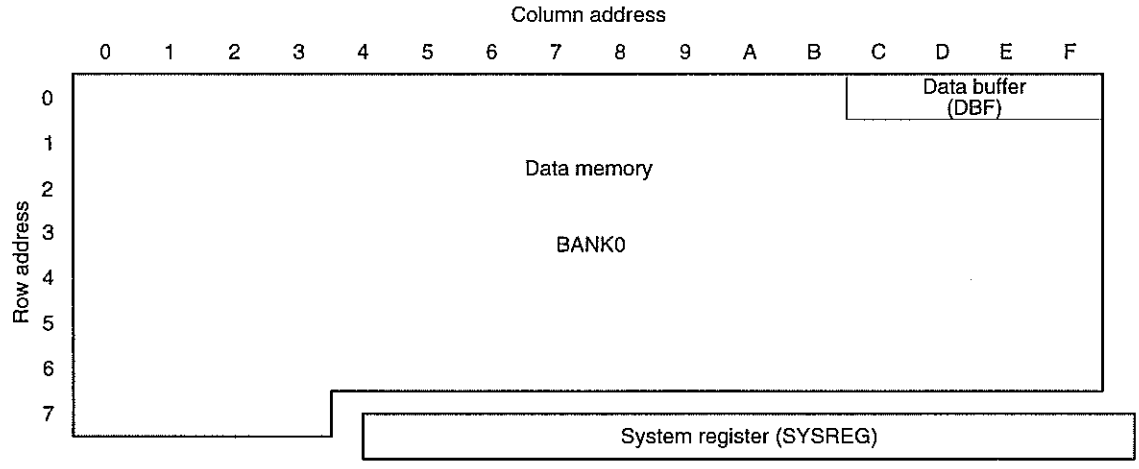



Fig. 9-2 shows the configuration of the data buffer. As shown in Fig. 9-2, the data buffer is made up of sixteen bits with its least significant bit in bit 0 of address 0FH and its most significant bit in bit 3 of address 0CH.

**Fig. 9-2 Data Buffer Configuration**

| Data memory BANK0 | Address | 0CH | | | | 0DH | | | | 0EH | | | | 0FH | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Bit | $b_3$ | $b_2$ | $b_1$ | $b_0$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ |
| Data buffer | Bit | $b_{15}$ | $b_{14}$ | $b_{13}$ | $b_{12}$ | $b_{11}$ | $b_{10}$ | $b_9$ | $b_8$ | $b_7$ | $b_6$ | $b_5$ | $b_4$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ |
| | Symbol | DBF3 | | | | DBF2 | | | | DBF1 | | | | DBF0 | | | |
| | Data | ^ MSB v | | | | Data | | | | | | | | | | | ^ LSB v |

Because the data buffer is allocated in data memory, it can be used in any of the data memory manipulation instructions.

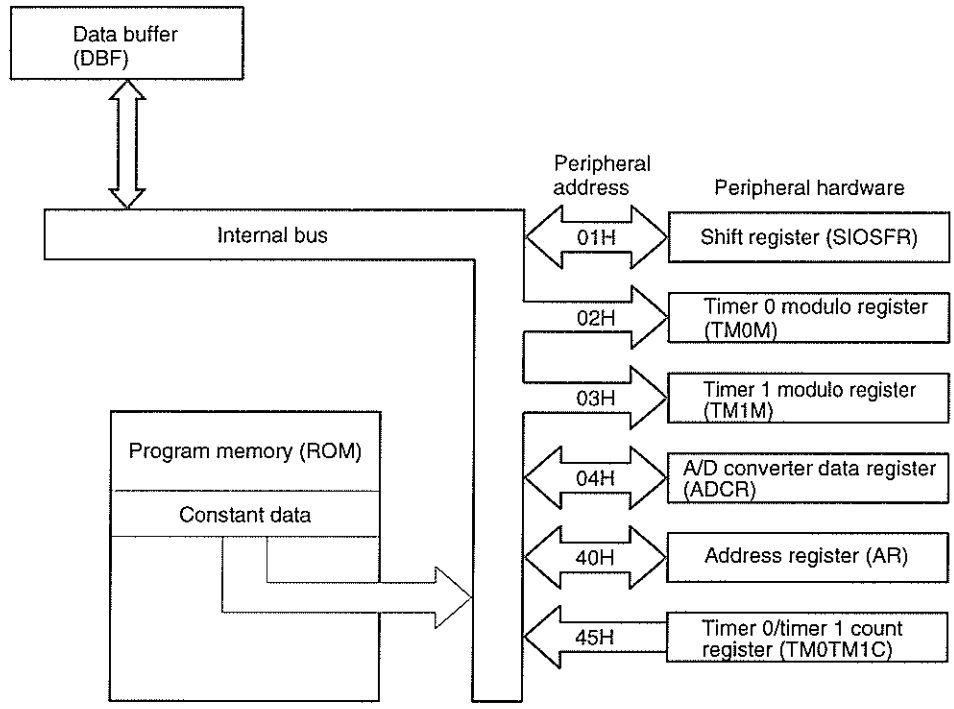
When reset, all 16 bits are undefined.

### 9.2 FUNCTIONS OF THE DATA BUFFER

The data buffer has two separate functions.

The data buffer is used for data transfer with peripheral hardware. The data buffer is also used for reading constant data in program memory. Fig. 9-3 shows the relationship between the data buffer and peripheral hardware.

**Fig. 9-3 Relationship between the Data Buffer and Peripheral Hardware**



## 10. ALU BLOCK

The ALU is used for performing arithmetic operations, logical operations, bit evaluations, comparison evaluations, and rotations on 4-bit data.

### 10.1 ALU BLOCK CONFIGURATION

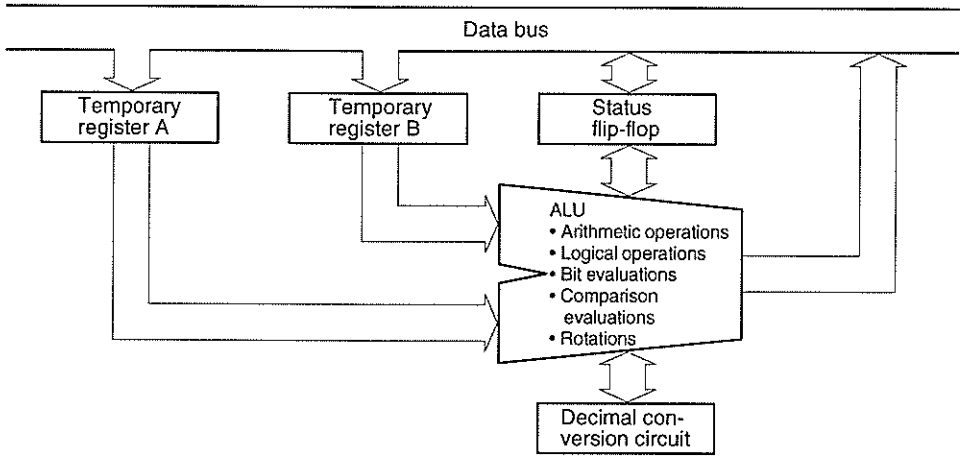
Fig. 10-1 shows the configuration of the ALU block.

As shown in Fig. 10-1, the ALU block consists of the main 4-bit data processor, temporary registers A and B, the status flip-flop for controlling the status of the ALU, and the decimal conversion circuit for use during arithmetic operations in BCD.

As shown in Fig. 10-1, the status flip-flop consists of the following flags: Zero flag flip-flop, carry flag flip-flop, compare flag flip-flop, and the BCD flag flip-flop.

Each flag in the status flip-flop corresponds directly to a flag in the program status word (PSWORD: addresses 7EH, 7FH) located in the system register. The flags in the program status word are the following: Zero flag (Z), carry flag (CY), compare flag (CMP), and the BCD flag (BCD).

Fig. 10-1 Configuration of the ALU




| Address | 7EH | 7FH | | | |
|---|---|---|---|---|---|
| Name | Program status word (PSWORD) | | | | |
| Bit | $b_0$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ |
| Flag | BCD | CMP | CY | Z | IXE |

| Status flip-flop | | | |
|---|---|---|---|
| BCD flag flip-flop | CMP flag flip-flop | CY flag flip-flop | Z flag flip-flop |

| Function outline |
|---|
| Indicates when the result of an arithmetic operation is 0. |
| Stores the borrow or carry from an arithmetic operation. |
| Used to indicate whether to store the result of an arithmetic operation. |
| Used to indicate whether to perform decimal correction for arithmetic operations. |

## 11. PORTS

### 11.1 PORT 0A ($P0A_0$, $P0A_1$, $P0A_2$, $P0A_3$)

Port 0A is a 4-bit input/output port with an output latch. It is mapped into address 70H of BANK0 in data memory. The output format is CMOS push-pull output.

Input or output can be specified in 4-bit units. Input/output is specified by P0AGIO (bit 0 at address 2CH) in the register file.

When P0AGIO is 0, each pin of port 0A is used as input port. If a read instruction is executed for the port register, pin statuses are read.

When P0AGIO is 1, each pin of port 0A is used as output port and the contents written in the output latch are output to pins. If a read instruction is executed when pins are output ports, the contents of the output latch, rather than pin statuses, are fetched.

Port 0A contains a software controlled pull-up resistor. P0AGPU (bit 0 at address 0CH) of the register file is used to determine whether port 0A contains the pull-up resistor. When P0AGPU is 1, all 4-bit pins are pulled up. If P0AGPU is 0, the pull-up resistor is not contained.

At reset, P0AGIO and P0AGPU are set to 0 and all P0A pins become input ports without a pull-up resistor. The contents of the port output latch are 0.

**Table 11-1 Writing into and Reading from the Port Register (0.70H)**

| P0AGIO RF: 2CH, bit 0 | Pin input/output | BANK0 70H | |
|---|---|---|---|
| | | Write | Read |
| 0 | Input | Possible Write to the P0A latch | P0A pin status |
| 1 | Output | | Data in P0A latch |

### 11.2 PORT 0B ($P0B_0$, $P0B_1$, $P0B_2$, $P0B_3$)

Port 0B is a 4-bit input/output port with an output latch. It is mapped into address 71H of BANK0 in data memory. The output format is CMOS push-pull output.

Input or output can be specified in 4-bit units. Input/ output is specified by P0BGIO (bit 1 at address 2CH) in the register file.

When P0BGIO is 0, all pins of port 0B are used as input ports. If a read instruction is executed for the port register, pin statuses are read.

When P0BGIO is 1, all pins of port 0B are used as output ports. The contents written in the output latch are output to pins. If a read instruction is executed when pins are used as output ports, the contents of the output latch, rather than pin statuses, are fetched.

Port 0B contains a software controlled pull-up resistor. P0BGPU (bit 1 at address 0CH) is used to determine whether or not port 0B contains a pull-up resistor. When P0BGPU is 1, all 4-bit pins are pulled up. When P0BGPU is 0, a pull-up resistor is not contained.

At reset, P0BGIO and P0BGPU are 0 and all P0B pins are input ports without a pull-up resistor. The value of the port 0B output latch is 0.

**Table 11-2 Writing into and Reading from the Port Register (0.71H)**

| P0BGIO RF: 2CH, bit 1 | Pin input/output | BANK0 71H | |
|---|---|---|---|
| | | Write | Read |
| 0 | Input | Possible Write to the P0B latch | P0B pin status |
| 1 | Output | | Data in P0B latch |

### 11.3 PORT 0C (P0C0/ADC0, P0C1/ADC1, P0C2/ADC2, P0C3/ADC3)

Port 0C is a 4-bit input/output port with an output latch. It is mapped into address 72H of BANK0 in data memory. The output format is CMOS push-pull output.

Input or output can be specified bit-by-bit. Input/output can be specified by P0CBIO0 to P0CBIO3 (address 1CH) in the register file.

If P0CBIOn is 0 (n = 0 to 3), the P0Cn pins are used as input port. If a data read instruction is executed for the port register, the pin statuses are read. If P0CBIOn is 1 (n = 0 to 3), the P0Cn pins are used as output port and the contents written in the output latch are output to pins. If a read instruction is executed when pins are used as output ports, the contents of the latch, rather than pin statuses, are fetched.

At reset, P0CBIO0 to P0CBIO3 are 0 and all P0C pins are input ports. The contents of the port output latch are 0.

Port 0C can also be used as an analog input to the A/D converter. P0C0IDI to P0C3IDI (1BH address) in the register file are used to switch the port and analog input pin.

If P0CnIDI is 0 (n = 0 to 3), the P0Cn/ADCn pin functions as a port. If P0CnIDI is 1 (n = 0 to 3), the P0Cn/ADCn pin functions as the analog input pin for the A/D converter. If any bit of P0CnIDI (n = 0 to 3) is 1, the P0F1/$V_{REF}$ pin is used as the $V_{REF}$ pin.

When using these pins for the analog input for the A/D converter, set P0CnIDI to 1 for the pins to which analog voltage is input, immediately after reset. This setting disables the port function for the pins. Then clear P0CBIOn (n = 0 to 3) to 0 to use the pins for input. Select which pins are used for analog input, using ADCCH0 and ADCCH1 (bits 0 and 1 at address 22H) in the register file.

At reset, P0CBIO0 to P0CBIO3, P0C0IDI to P0C3IDI, ADCCH0, and ADCCH1 are set to 0 and the P0C pins are used as input ports.

**Table 11-3 Switching the Port and A/D Converter**

(n = 0 to 3)

| P0CnIDI RF: 1BH | P0CBIOn RF: 1CH | Function | BANK0 72H | |
|---|---|---|---|---|
| | | | Write | Read |
| 0 | 0 | Input port | Possible P0C latch | Pin status |
| | 1 | Output port | Possible P0C latch | Data in P0C latch |
| 1 | 0 | A/D converter analog input[Note 1] | Possible P0C latch | Data in P0C latch |
| | 1 | Output port and A/D converter analog input[Note 2] | Possible P0C latch | Data in P0C latch |

**Notes 1.** Normal setting when the pins are used as A/D converter analog input pins.

**2.** Functions as an output port. The analog input voltage is changed by the output from the port. To use the pins for analog input, be sure to set P0CBIOn to 0.

### 11.4 PORT 0D (P0D0/$\overline{SCK}$, P0D1/SO, P0D2/SI, P0D3/$\overline{TM1OUT}$)

Port 0D is a 4-bit input/output port with an output latch. It is mapped into address 73H of BANK0 in data memory. The output format is N-ch open-drain output.

Input or output can be specified bit-by-bit. Input/output is specified with P0DBIO0 to P0DBIO3 (address 2BH) in the register file.

If P0DBIOn is 0 (n = 0 to 3), the P0Dn pins are used as input port. Pin statuses are read if a data read instruction is executed for the port register. If P0DBIOn is 1, the P0Dn pins are used as output port and the value written in the output latch are output to pins. If a data read instruction is executed when pins are used as output ports, the output latch value, rather than pin statuses, is fetched.

Port 0D contains a software controlled pull-up resistor. P0DBPU0 to P0DBPU3 (address 0DH) of the register file are used to determine whether each bit of port 0D contains the pull-up resistor. When P0DBPUn is 1, the P0Dn pin is pulled up. If P0DBPUn is 0, the pull-up resistor is not contained.

At reset, P0DBIOn is set to 0 and all P0D pins become input ports. The contents of the port output latch become 0. The output latch contents remain unchanged even if P0DBIOn changes from 1 to 0.

Port 0D can also be used for serial interface input/output or timer 1 output. SIOEN (0BH bit 0) in the register file is used to switch ports (P0D0 to P0D2) to serial interface input/output ($\overline{SCK}$, SO, SI) and vice versa. TM1OSEL (bit 3 at address 0BH) in the register file is used to switch a port (P0D3) to timer 1 output ($\overline{TM1OUT}$) and vice versa. If TM1OSEL = 1 is selected, 1 is output at timer 1 reset. This output is inverted every time a timer 1 count value matches the modulo register contents.

**Table 11-4 Register File Contents and Pin Functions**

(n = 0 to 3)

| Register file value | | | Pin function | | | |
|---|---|---|---|---|---|---|
| TM1OSEL RF: 0BH Bit 3 | SIOEN RF: 0BH Bit 0 | P0DBIOn RF: 2BH Bit n | P0D0/$\overline{SCK}$ | P0D1/SO | P0D2/SI | P0D3/$\overline{TM1OUT}$ |
| 0 | 0 | 0 | Input port | | | |
| | | 1 | Output port | | | |
| | 1 | 0 | $\overline{SCK}$ | SO | SI | Input port |
| | | 1 | | | | Output port |
| 1 | 0 | 0 | Input port | | | $\overline{TM1OUT}$ |
| | | 1 | Output port | | | |
| | 1 | 0 | $\overline{SCK}$ | SO | SI | |
| | | 1 | | | | |

**Table 11-5 Data Read from the Port Register (0.73H)**

| Port mode | | Data read from the port register (0.73H) |
|---|---|---|
| Input port | | Pin status |
| Output port | | Data in output latch |
| $\overline{SCK}$ | An internal clock is selected as a serial clock. | Data in output latch |
| | An external clock is selected as a serial clock. | Pin status |
| SI | | Pin status |
| SO | | Data in output latch |
| $\overline{TM1OUT}$ | | Data in output latch |

## 11.5 PORT 0E ($P0E_0$, $P0E_1$, $P0E_2$, $P0E_3$)

Port 0E is a 4-bit input/output port with an output latch. It is mapped into address 6EH of BANK0 in data memory. The output format is N-ch open-drain output.

Input or output can be specified in units of four bits. Input/output is specified by P0EGIO (bit 2 at address 2CH) in the register file.

When P0EGIO is 0, each pin of port 0E is used as input port. If a read instruction is executed for the port register, pin statuses are read. When P0EGIO is 1, each pin of port 0E is used as output port and the contents written in the output latch are output to pins. If a read instruction is executed when pins are output ports, the contents of the output latch, rather than pin statuses, are fetched.

Port 0E contains a software controlled pull-up resistor. P0EGPU (bit 2 at address 0CH) of the register file in used to determine whether port 0E contains the pull-up resistor. When P0EGPU is 1, all 4-bit pins are pulled up. If P0EGPU is 0, the pull-up resistor is not contained.

At reset, P0EGIO is set to 0 and all P0E pins become input ports. The contents of the port output latch are 0.

**Table 11-6 Writing into and Reading from the Port Register (0.6EH)**

(n = 0 to 3)

| P0EGIO RF: 2CH, bit 2 | Pin input/output | BANK0 6EH | |
|---|---|---|---|
| | | Write | Read |
| 0 | Input | Possible | P0E pin status |
| 1 | Output | Write to the P0E latch | Data in P0E latch |

## 11.6 PORT 0F ($P0F_0/\overline{RLS}$, $P0F_1/V_{REF}$)

Port 0F is a 2-bit input-dedicated port. It is mapped into address 6FH of BANK0 in data memory. Mask option can be used to specify whether each pin uses a built-in pull-up resistor.

If a pin of port 0F is used as an input port, a pin status is read in the two low-order bits of the port register when a data read instruction is executed for the port register (the two high-order bits are always 0). A data write instruction does not affect the port register.

The $P0F_0/\overline{RLS}$ pin can also be used for the input pin for the signal for releasing the standby mode.

The $P0F_1/V_{REF}$ pin is used as the $V_{REF}$ pin (reference voltage input pin for the A/D converter) when any bit of P0CnIDI (address 1BH in the register file, n = 0 to 3) is 1. If the $P0F_1/V_{REF}$ pin functions as the $V_{REF}$ pin, bit 1 at address 6FH is always 0 when a data read instruction is executed for the port register.

## 12. 8-BIT TIMER COUNTER (TM0, TM1)

Timer 0 (TM0) and timer 1 (TM1) are available as 8 bit-timer counters.

By using the timer 0 counting signal as the timer 1 count clock, these two 8-bit counters can be used as a 16-bit timer.

The timers are controlled by hardware operation with the PUT/GET instruction or by register operation in the register file with the PEEK/POKE instruction.

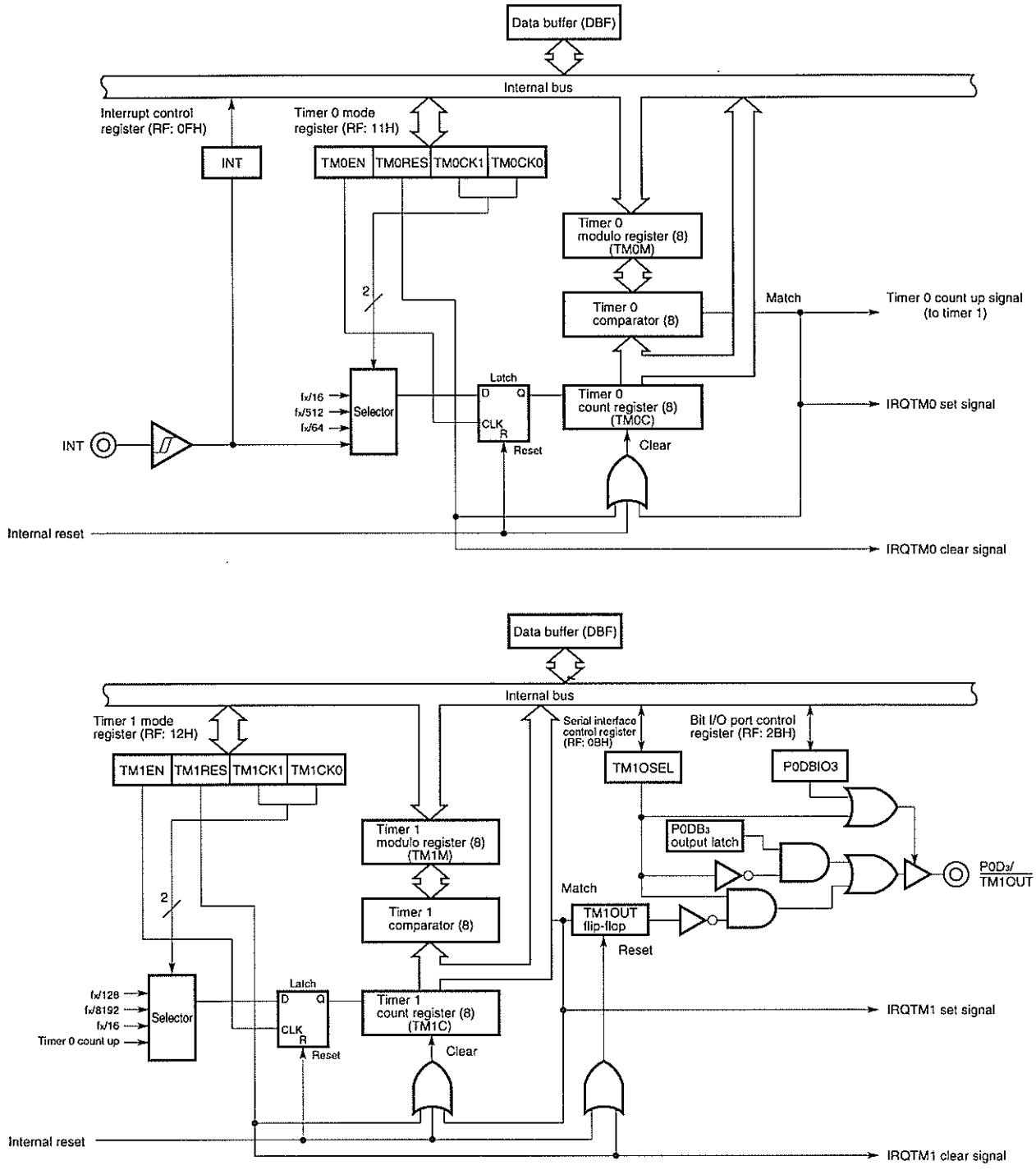
### 12.1 CONFIGURATION OF 8-BIT TIMER COUNTERS

Fig. 12-1 shows the configuration of the 8-bit timer counters. An 8-bit timer consists of an 8-bit counter register, 8-bit modulo register, comparator (compares counter register values and modulo register values), and selector (for count pulse selection).

**Cautions 1. The modulo register is a write-only register.**

**2. The count register is a read-only register.**

Fig. 12-1 Configuration of the 8-Bit Timer Counters




**Remark** fx: System clock oscillation frequency

**Fig. 12-2 Timer 0 Mode Register**

RF: 11H

| | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|
| | TM0EN | TM0RES | TM0CK1 | TM0CK0 |
| Read/write | R/W | | | |
| Initial value when reset | 0 | 0 | 0 | 0 |

Read = R, write = W

| TM0CK1 | TM0CK0 | Select a count pulse of timer 0 |
|---|---|---|
| 0 | 0 | fx/16 |
| 0 | 1 | fx/512 |
| 1 | 0 | fx/64 |
| 1 | 1 | INT pin |

| TM0RES | Resets timer 0 |
|---|---|
| 0 | No effect on timer 0 |
| 1 | The timer 0 count register and IRQTM0 are reset. |

**Remark** TM0RES is automatically cleared to 0 after it is set to 1. 0 is always read.

| TM0EN | Instruction to start timer 0 |
|---|---|
| 0 | Timer 0 counting stops. |
| 1 | Timer 0 counting starts. |

**Remark** TM0EN can be used as a status flag for detecting the counting status of timer 0.
(1: Counting, 0: Not counting)

Fig. 12-3 Timer 1 Mode Register

RF: 12H

| | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|
| | TM1EN | TM1RES | TM1CK1 | TM1CK0 |
| Read/write | R/W | | | |
| Initial value when reset | 1 | 0 | 0 | 0 |

| TM1CK1 | TM1CK0 | Select a count pulse of timer 0 |
|---|---|---|
| 0 | 0 | fx/128 |
| 0 | 1 | fx/8192 |
| 1 | 0 | fx/16 |
| 1 | 1 | Count up signal from timer 0 |

| TM1RES | Resets timer 1 |
|---|---|
| 0 | No effect on timer 1 |
| 1 | The timer 1 count register and IRQTM1 are reset. |

**Remark** TM1RES is automatically cleared to 0 after it is set to 1. 0 is always read.

| TM1EN | ← Instruction to start timer 1 |
|---|---|
| 0 | Timer 1 counting stops. |
| 1 | Timer 1 counting starts. |

**Remark** TM1EN can be used as a status flag for detecting the counting status of timer 1. (1: Counting, 0: Not counting)

## 13. BASIC INTERVAL TIMER (BTM)

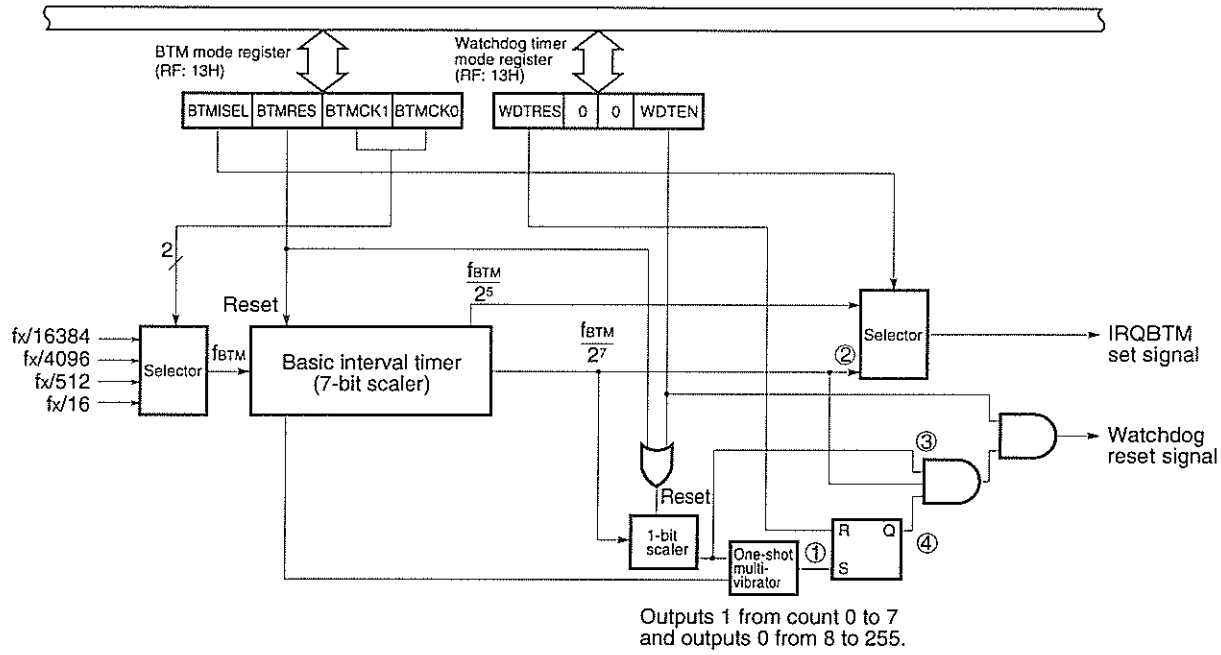
The μPD17147GS provides a 7-bit basic interval timer. This timer has the following functions:

(1) Reference time generation
(2) Selection and counting of a wait time when standby mode is released
(3) Watchdog timer operation for detecting software errors (infinite loops, etc.)

### 13.1 CONFIGURATION OF THE BASIC INTERVAL TIMER

Fig. 13-1 shows the configuration of the basic interval timer.

**Fig. 13-1 Configuration of the Basic Interval Timer**




**Remark** ① to ④ in the figure indicate the signals in the timing chart in Fig. 13-4.
fx: System clock oscillation frequency

## 13.2 REGISTERS FOR CONTROLLING THE BASIC INTERVAL TIMER

The basic interval timer is controlled by the BTM mode register and watchdog timer mode register.

Fig. 13-2 and 13-3 show the configurations of the registers.

**Fig. 13-2 BTM Mode Register**

RF: 13H

| | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|
| | BTMISEL | BTMRES | BTMCK1 | BTMCK0 |
| Read/write | R/W | | | |
| Initial value when reset | 0 | 0 | 0 | 0 |

Read = R, write = W

| BTMCK1 | BTMCK0 | Select a BTM count pulse |
|---|---|---|
| 0 | 0 | fx/16 (1-instruction execution time) |
| 0 | 1 | fx/16384 (1024-instruction execution time) |
| 1 | 0 | fx/4096 (256-instruction execution time) |
| 1 | 1 | fx/512 (32-instruction execution time) |

| BTMRES | Specifies whether to reset the basic interval timer (BTM) |
|---|---|
| 0 | Does not affect the basic interval timer (BTM). |
| 1 | Resets the binary count of the BTM. |

**Remark** BTMRES is automatically cleared to 0 after it is set to 1. 0 is always read.

| BTMISEL | Selects an interval |
|---|---|
| 0 | Sets the count pulse divided by 128. |
| 1 | Sets the count pulse divided by 32. |

Fig. 13-3 Watchdog Timer Mode Register

RF: 03H

| | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|
| | WDTRES | 0 | 0 | WDTEN |
| Read/write | R/W | | | |
| Initial value when reset | 0 | 0 | 0 | 0 |

Read = R, write = W

| WDTEN | Enables or disables the watchdog timer function |
|---|---|
| 0 | Disables the watchdog timer. |
| 1 | Enables the watchdog timer. |

**Remarks 1.** WDTEN cannot be cleared to 0 by the program.

**2.** WDTEN is automatically cleared to 0 after it is set to 1. 0 is always read.

| WDTRES | Specifies whether to reset the watchdog timer |
|---|---|
| 0 | Does not affect the watchdog timer. |
| 1 | Resets the flip-flop used to retain a BTM overflow carry used for the watchdog timer. |

**Remark** WDTRES is automatically cleared to 0 after it is set to 1. 0 is always read.

### 13.3 WATCHDOG TIMER FUNCTION

The basic interval timer can also be used as a watchdog timer which detects a system hang.

#### 13.3.1 Overview of the Watchdog Timer

The watchdog timer is a counter that generates a reset signal at constant intervals. When the generation of a reset signal is being disabled every time by the program, this function enables the system to be reset (starting from address 0000H) when the system hangs up (the watchdog timer is not reset within the expected time) for some reason, such as due to external noise.

Even if a program branches to an unexpected routine due to external noise and enters an infinite loop, the system can be recovered within a certain time by the reset signal that is generated by the watchdog timer.

#### 13.3.2 Operation of the Watchdog Timer

If WDTEN is set to 1, the 1-bit scaler starts operating, causing the basic interval timer to operate as an 8-bit watchdog timer.

Once the watchdog timer runs, the watchdog timer function can be stopped only when the device is reset and WDTEN is cleared to 0.

A reset by the watchdog timer can be disabled in the following two ways:

(1) Setting WDTRES to 1 repeatedly in the program
(2) Setting BTMRES to 1 repeatedly in the program

For (1), it is necessary to set WDTRES to 1 while the watchdog timer count is between 8 and 191 (immediately before 192), as shown in Fig. 13-4. Therefore, the program must be written so that SET1 WDTRES is executed at least once in a shorter period than that required for the watchdog timer count to reach 184.

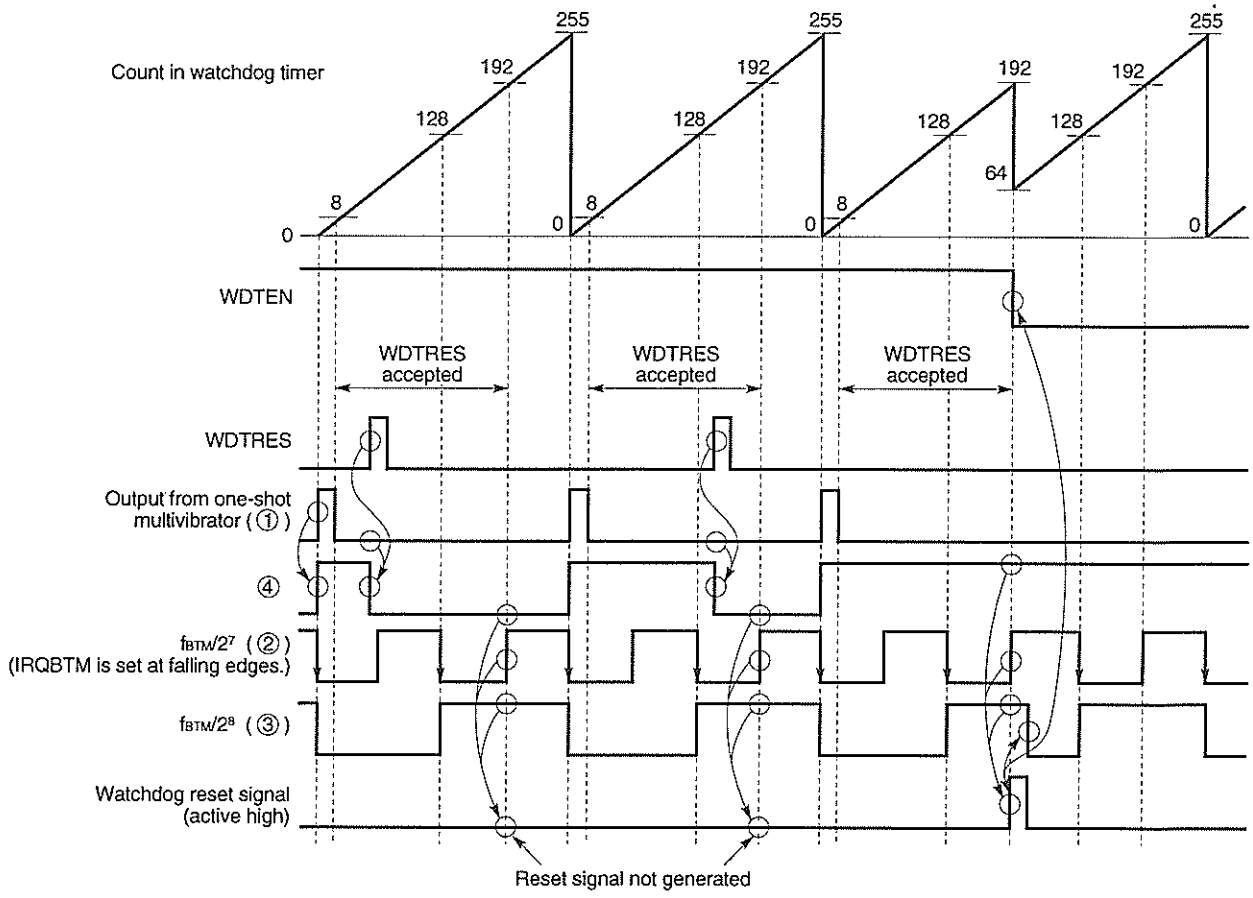
For (2), it is necessary to set BTMRES to 1 before the basic interval timer (BTM) counts to reach 128. Therefore, the program must be written so that SET1 BTMRES is executed at least once in a shorter period than that required for the basic interval timer count to reach 128. However, using this method, interrupt handling by the basic interval timer is disabled.

**Caution Setting WDTEN to 1 does not reset the basic interval timer. So, set BTMRES to 1 before setting WDTEN to 1, to reset the basic interval timer.**

**Example**

```
      ⋮
SET1  BTMRES
SET2  WDTEN, WDTRES
      ⋮
```

Fig. 13-4 Timing Chart for the Watchdog Timer (When the WDTRES Flag Is Used)



## 14. A/D CONVERTER

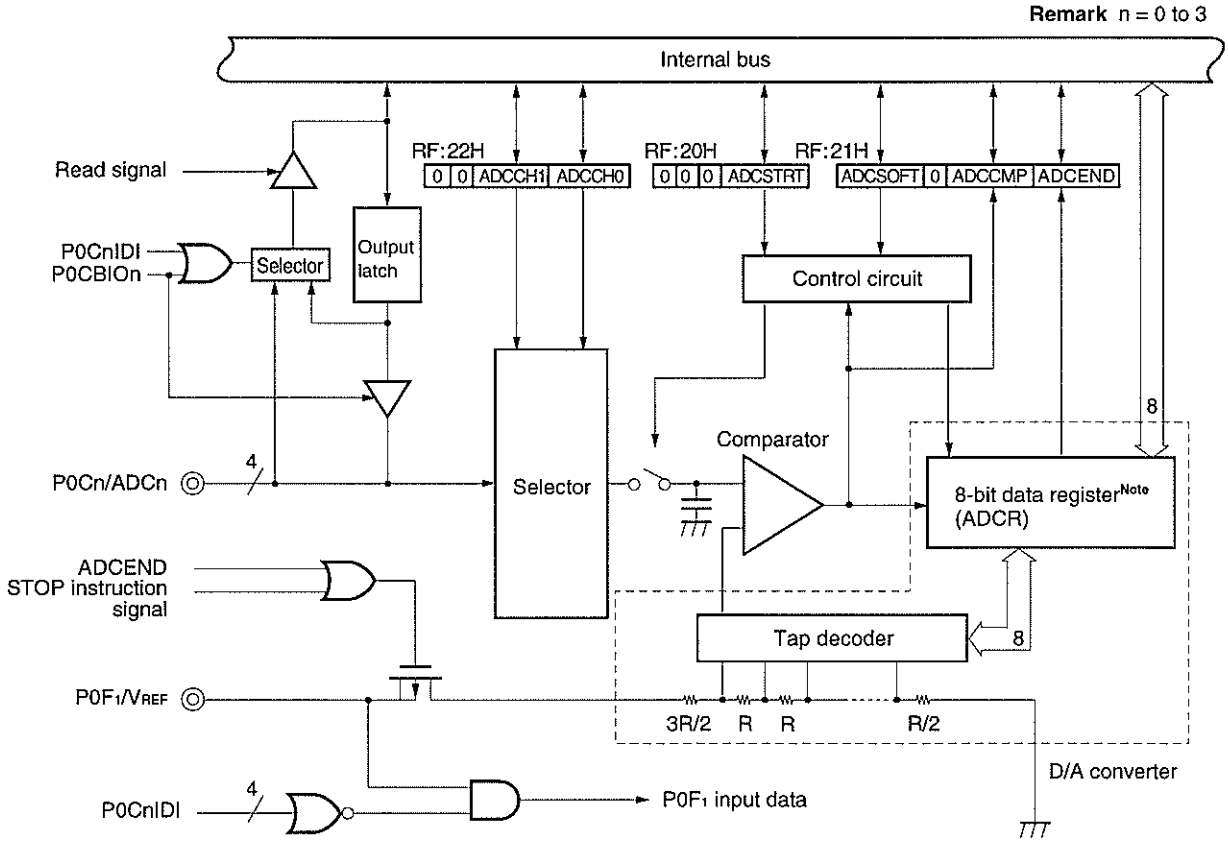
μPD17147GS contains an 8-bit resolution A/D converter with 4-channel analog input (P0C0/ADC0 - P0C3/ADC3). The A/D converter uses the successive approximation method. The following two operation modes are available:

① Continuous mode: 8-bit A/D conversion occurs starting at high-order bits.
② Single mode: Comparison occurs with an arbitrary voltage value set in the 8-bit data register.

### 14.1 A/D CONVERTER CONFIGURATION

Fig. 14-1 shows the A/D converter configuration.

**Fig. 14-1 Block Diagram for the A/D Converter**




**Note** The 8-bit data register (ADCR) is cleared to 00H at the execution of STOP instruction.

## 14.2 A/D CONVERTER FUNCTIONS

### (1) $ADC_0$ - $ADC_3$ pins

These pins are used to input 4-channel analog voltage to the A/D converter. The A/D converter contains a sample hold circuit. Analog input voltage is internally retained during A/D conversion.

### (2) $V_{REF}$ pin

This pin is used to input the reference voltage for the A/D converter.

A signal input to $ADC_0$ to $ADC_3$ is converted to a digital signal based on voltage applied across $V_{REF}$ and GND.

To reduce the current consumption of the microcontroller, the A/D converter has a function for automatically stopping the current which flows into the $V_{REF}$ pin when the converter is not operating. Current flows into the $V_{REF}$ pin in the following cases.

① **Continuous mode (ADCSOFT=0)**
From when the ADCSTRT flag is set (1) until the ADCEND flag is set (1).

② **Single mode (ADCSOFT=1)**
From when the ADCSTRT flag is set (1) or from when a value of the 8-bit data register is written until the result of comparison by the comparator is written in the ADCCMP flag.

**Remarks 1.** Even if the HALT instruction is executed during the A/D conversion, the A/D converter operates until the ADCEND flag is set in continuous mode or until the result of comparison is saved in the ADCCMP flag in single mode. A current flows into the $V_{REF}$ pin while the A/D converter is operating.

**2.** The A/D conversion stops when the STOP instruction is executed. The A/D converter is initialized and a current does not flow into the $V_{REF}$ pin. (The A/D converter remains stopped after the STOP mode is released.)

### (3) 8-bit data register (ADCR)

In the continuous mode, this 8-bit data register stores A/D conversion results for successive approximation. It is read by the GET instruction. In the single mode, the data in this register is converted to analog voltage by the internal D/A converter and the comparator compares this voltage with an analog signal input from the ADCn pin. A value can be written in this register by using the PUT instruction.

### (4) Comparator

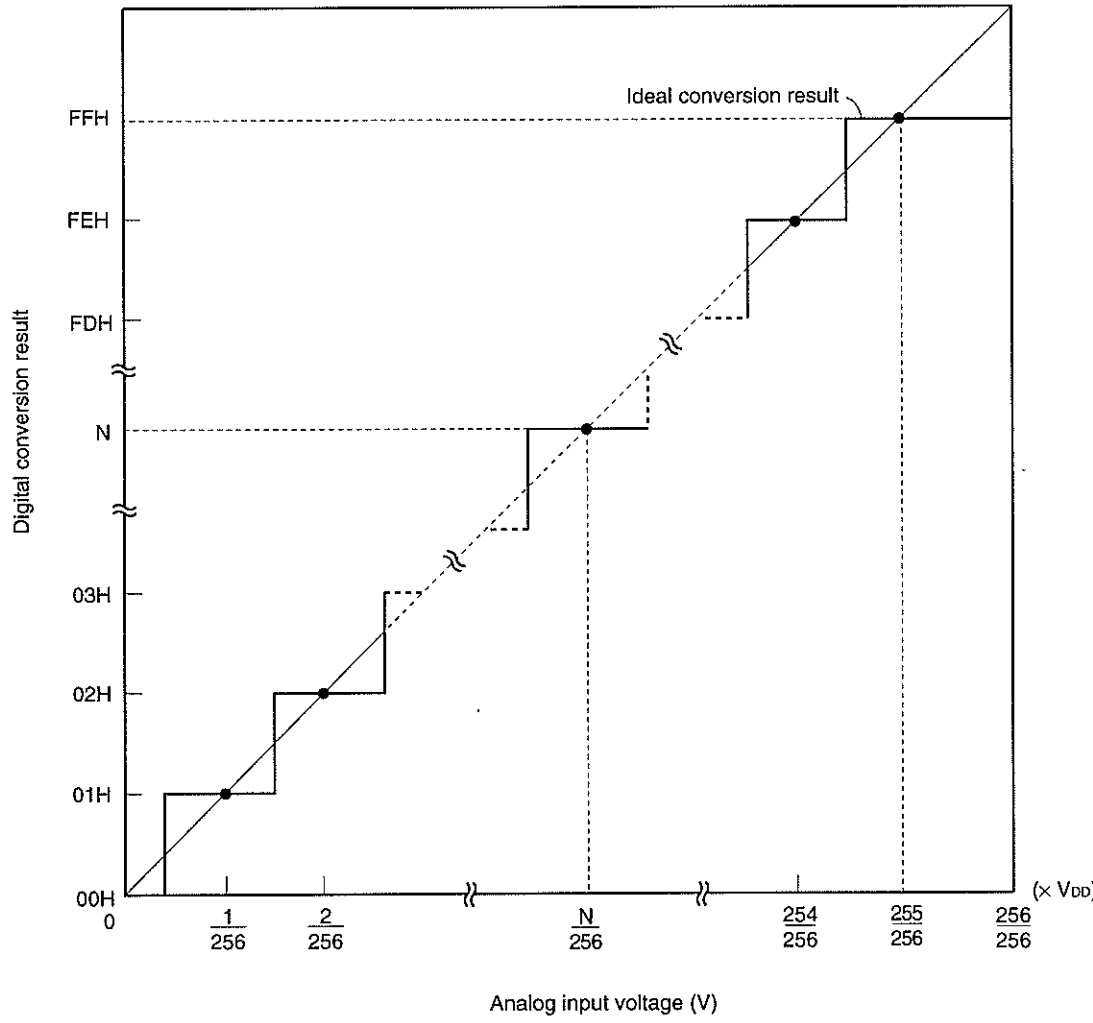
The comparator compares an analog input voltage with voltage output from the D/A converter. Value 1 is output if analog input voltage is higher. Value 0 is output if this voltage is lower. The comparison result is stored in the 8-bit data register (ADCR) in the continuous mode. It is stored in the ADCCMP flag in the single mode.

### 14.3 A/D CONVERTER OPERATION

The A/D converter operates in two modes: continuous mode and single mode. The mode can be switched by setting the ADCSOFT flag.

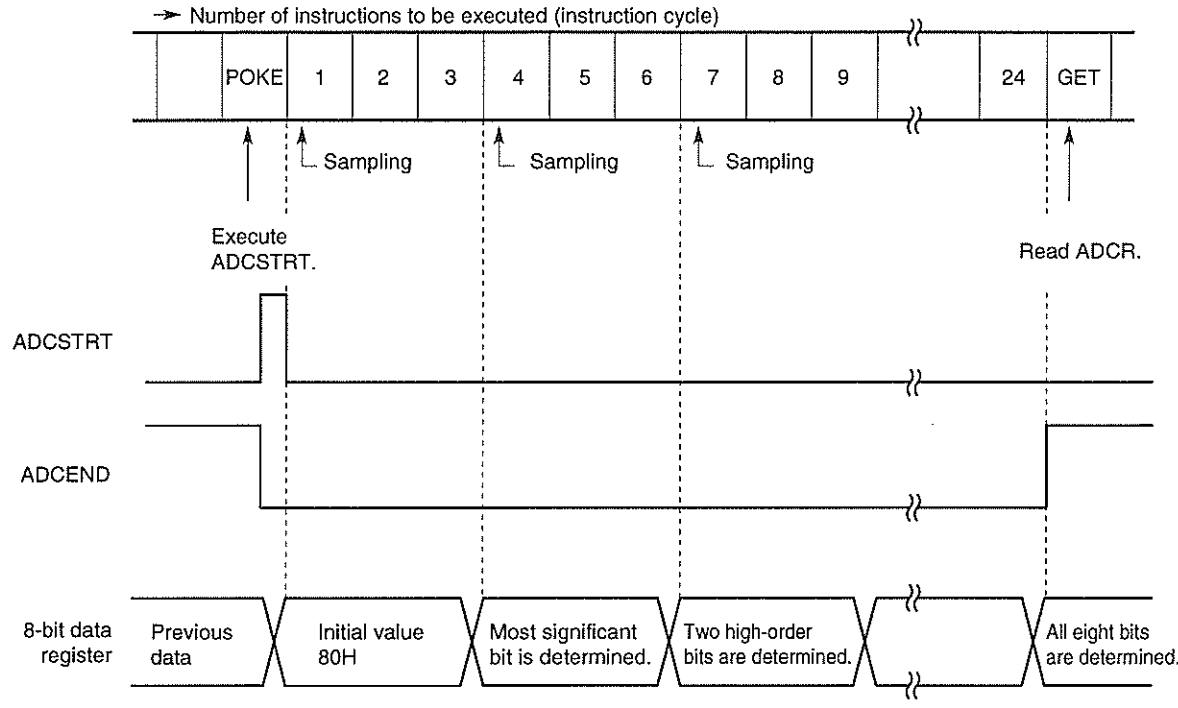
| ADCSOFT | A/D converter operation mode |
|---|---|
| 0 | Continuous mode (A/D conversion) |
| 1 | Single mode (comparison) |

**Fig. 14-2 Relation between the Analog Input Voltage and Digital Conversion Result**



(1) Timing in the continuous mode operation (A/D conversion)

Fig. 14-3 Timing in the Continuous Mode Operation (A/D Conversion)

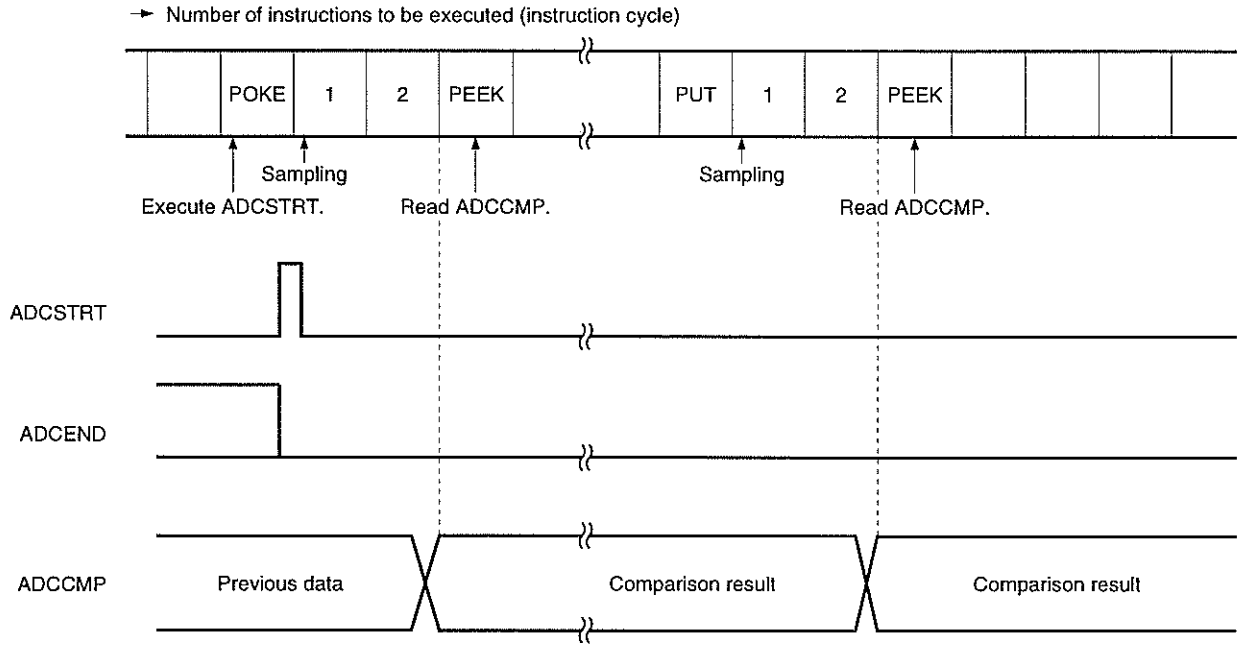



**Caution** **Sampling is performed eight times for each A/D conversion. If the analog input voltage changes considerably during A/D conversion, accurate A/D conversion cannot be performed. To obtain accurate conversion, minimize any change in the analog input voltage during A/D conversion.**

**Time required for one sampling operation = 14/fx (1.75 μs at fx = 8 MHz)**
**Sampling cycle period = 48/fx (6 μs at fx = 8 MHz)**

(2) Timing in the single mode operation (comparison)

Fig. 14-4 Timing in the Single Mode Operation (Comparison)




In the single mode, after 1 is written to ADCSTRT (by executing the POKE instruction), a value is stored in ADCCMP and the comparison result is read by the PEEK instruction at the execution of the third instruction. Setting a value in ADCR (by executing the PUT instruction) also starts the comparison and the result is read at the third instruction after the setting.

ADCCMP is cleared to 0 by reset or by the execution of a write instruction to ADCR.

**Caution Before setting a value in ADCR, always set ADCSOFT to 1. If ADCSOFT = 0, no value can be set in ADCR. (The PUT ADCR,DBF instruction is ineffective.)**

**Remark** Sampling time = 14/fx (1.75 μs at fx = 8 MHz)

## 15. SERIAL INTERFACE (SIO)

The serial interface consists of an 8-bit shift register (SIOSFR), serial mode register, and serial clock counter. It is used for serial data input/output.

### 15.1 FUNCTIONS OF THE SERIAL INTERFACE

This serial interface provides three signal lines: serial clock input pin ($\overline{SCK}$), serial data output pin (SO), and serial data input pin (SI). It allows 8 bits to be sent or received in synchronization with clocks. It can be connected to peripheral input/output devices using any method with a mode compatible to that used by the μPD7500 or 75X series.

#### (1) Serial clock

Three types of internal clocks and one type of external clock are able to be selected. If an internal clock is selected as a serial clock, it is automatically output to the $P0D_0/\overline{SCK}$ pin.

**Table 15-1 Serial Clocks**

| SIOCK1 | SIOCK0 | Serial clock to be selected |
|---|---|---|
| 0 | 0 | External clock input to the $\overline{SCK}$ pin |
| 0 | 1 | fx/16 |
| 1 | 0 | fx/128 |
| 1 | 1 | fx/1024 |

fx: System clock oscillation frequency

#### (2) Transmission operation

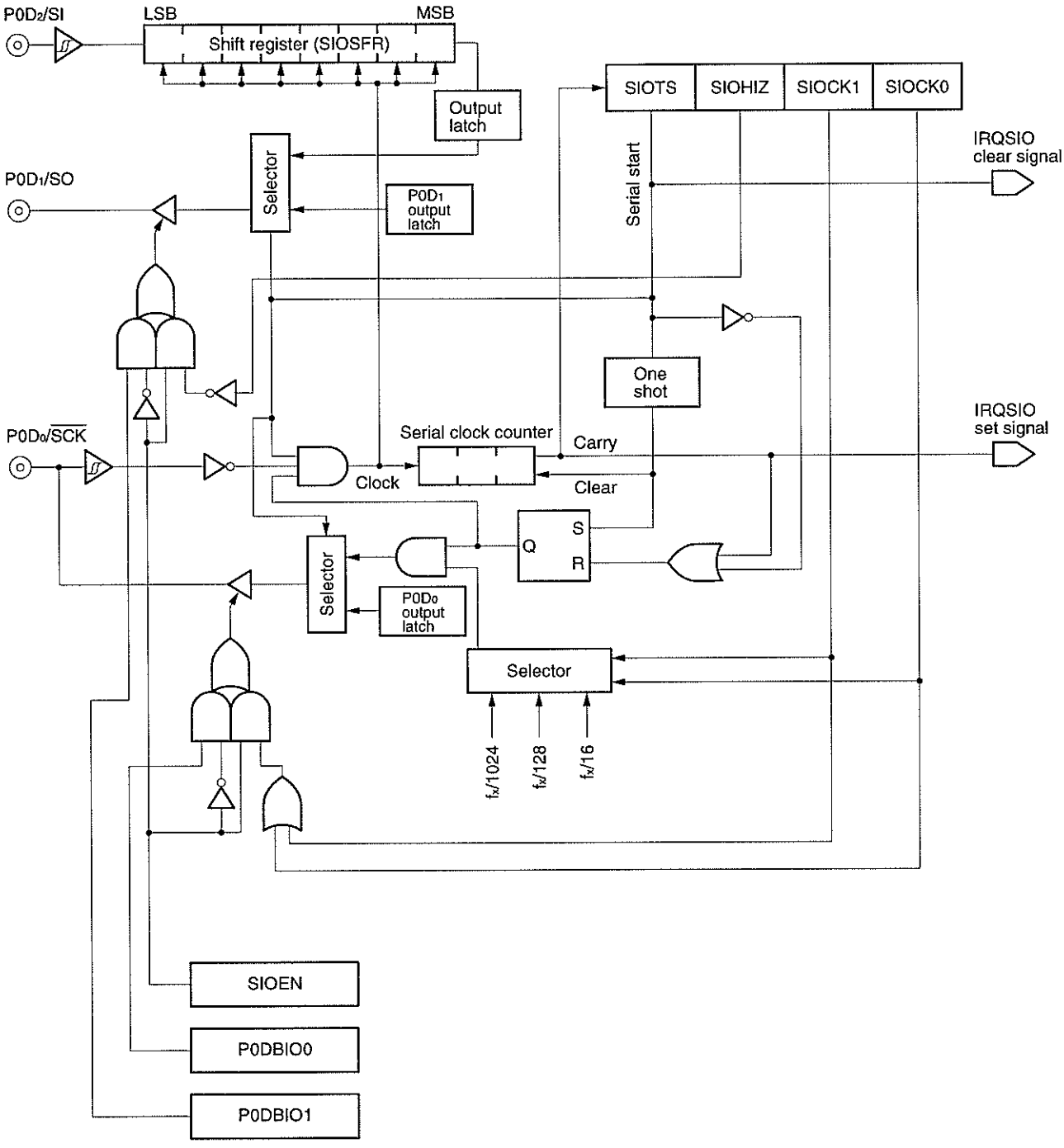
When SIOEN is set to 1, the pins of port 0D ($P0D_0/\overline{SCK}$, $P0D_1$/SO, $P0D_2$/SI) function as the pins of the serial interface. The serial interface operates in synchronization with the falling edge of the external or internal clock by setting SIOTS to 1. When SIOTS is set, IRQSIO is automatically cleared.

Transmission starts from the most significant bit of the shift register in synchronization with the falling edge of the serial clock. SI pin information is stored in the shift register starting at the least significant bit in synchronization with the rising edge of the serial clock.

When the 8-bits data transmission is terminated, SIOTS is automatically cleared and IRQSIO is set.

**Remark** Serial transmission starts only from the most significant bit of the shift register contents. It is not possible to transmit from the least significant bit. SI pin status is always stored in the shift register in synchronization with the rising edge of the serial clock.

Fig. 15-1 Block Diagram of the Serial Interface




**Caution** The output latch of the shift register is independent of that of the P0D1 pin. Therefore, even if an output instruction is executed for the P0D1 pin, the output latch status of the shift register does not change. Inputting a reset signal clears the output latch of the shift register to 0. After that, the latch retains the LSB of the data transmitted previously.

### 15.2 3-WIRE SERIAL INTERFACE OPERATION MODES

Two modes can be used for the serial interface. If the serial interface function is selected, the P0D2/SI pin always takes in data in synchronization with the serial clock.

- 8-bit transmission and reception mode (simultaneous transmission and reception)
- 8-bit reception mode (with the SO pin set to the high impedance status)

**Table 15-2 Serial Interface Operation Mode**

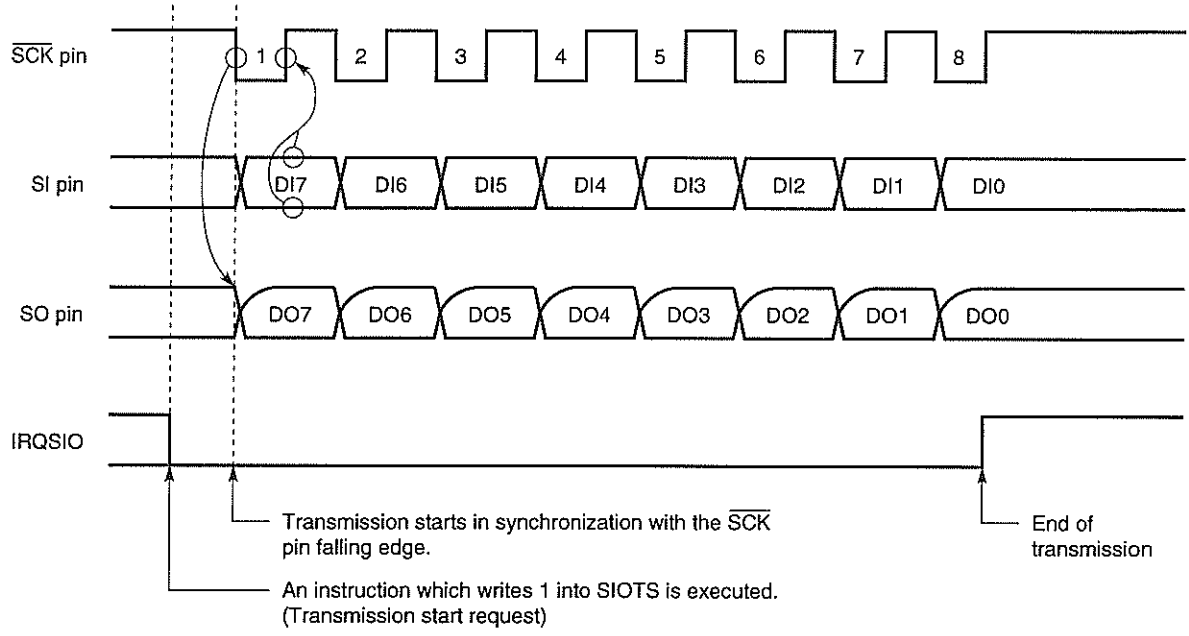
| SIOEN | SIOHIZ | P0D0/SI pin | P0D1/SO pin | Serial interface operation mode |
|---|---|---|---|---|
| 1 | 0 | SI | SO | 8-bit transmission and reception mode |
| 1 | 1 | SI | P0D1 (input) | 8-bit reception mode |
| 0 | × | P0D0 (I/O) | P0D1 (I/O) | General port mode |

×: Don't care

**(1) 8-bit transmission and reception mode (simultaneous transmission and reception)**

Serial data input/output is controlled by a serial clock. The most significant bit of the shift register is output from the SO line with a falling edge of the serial clock ($\overline{SCK}$ pin signal). The contents of the shift register is shifted one bit and at the same time, data on the SI line is loaded into the least significant bit of the shift register. The serial clock counter (3-bit counter) counts serial clock pulses. Every time it counts eight clocks , the internal interrupt request flag (IRQSIO) is set to 1.

**Fig. 15-2 Timing of 8-Bit Transmission and Reception Mode (Simultaneous Transmission and Reception)**



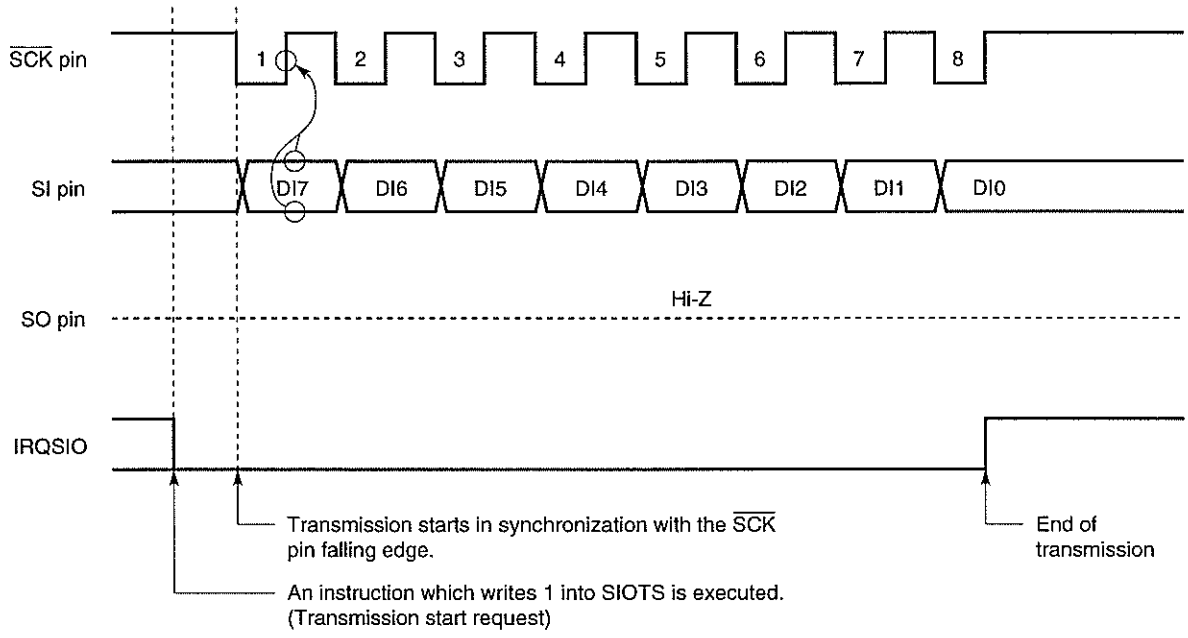

**Remark** DI : Input serial data
DO: Output serial data

**(2) 8-bit transmission and reception mode (SO pin in the high impedance status)**

When SIOHIZ is 1, the P0D1/SO pin is in the high impedance status. If serial clock supply starts by writing 1 in SIOTS, only the reception function of the serial interface operates.

The P0D1/SO pin is in the high impedance status and can be used for input port (P0D1).

**Fig. 15-3 Timing Chart in 8-Bit Reception Mode**




**Remark** DI: Input serial data

**(3) Operation stop mode**

If the value in SIOTS (RF: address 02H, bit 3) is 0, the serial interface enters operation stop mode. In this mode, no serial transfer occurs.

In this mode, the shift register does not perform shifting and can be used as an ordinary 8-bit register.

## 16. INTERRUPT FUNCTIONS

The μPD17147GS has four internal interrupt functions and one external interrupt function. It can be used in various applications.

The interrupt control circuit of the μPD17147GS has the features listed below. This circuit enables very high-speed interrupt handling.

(a) Used to determine whether an interrupt can be accepted with the interrupt mask enable flag (INTE) and interrupt enable flag (IPxxx).
(b) The interrupt request flag (IRQxxx) can be tested or cleared. (Interrupt generation can be checked by software.)
(c) Multiple interrupts are possible (up to three levels).
(d) Standby mode (STOP, HALT) can be released by an interrupt request. (Release conditions can be selected by the interrupt enable flag.)

**Caution In interrupt handling, only the BCD, CMP, CY, Z, and IXE flags are saved in the stack automatically by the hardware for up to three levels of multiple interrupts. The DBF and WR are not saved by the hardware when peripheral hardware such as the timers or A/D converter is accessed in interrupt handling. It is recommended that the DBF and WR be saved in RAM by the software at the beginning of interrupt handling. Saved data can be loaded back into the DBF and WR immediately before the end of interrupt handling.**

### 16.1 INTERRUPT SOURCES AND VECTOR ADDRESSES

For every interrupt in the μPD17147GS, when the interrupt is accepted, a branch occurs to the vector address associated with the interrupt source. This method is called the vectored interrupt method. Table 16-1 lists the interrupt source types and vector addresses.

If two or more interrupt requests occur or multiple suspended interrupt requests are enabled at the same time, they are handled according to priorities shown in Table 16-1.

**Table 16-1 Interrupt Source Types**

| Interrupt source | Priority | Vector address | IRQ flag | IP flag | IEG flag | Internal/ external | Remarks |
|---|---|---|---|---|---|---|---|
| INT pin (RF: 0FH, bit 0) | 1 | 0005H | IRQ RF: 3FH, bit 0 | IP RF: 2FH, bit 0 | IEGMD0,1 RF: 1FH | External | Rising edge or falling edge can be selected. |
| Timer 0 | 2 | 0004H | IRQTM0 RF: 3EH, bit 0 | IPTM0 RF: 2FH, bit 1 | – | Internal | – |
| Timer 1 | 3 | 0003H | IRQTM1 RF: 3DH, bit 0 | IPTM1 RF: 2FH, bit 2 | – | Internal | – |
| Basic interval timer | 4 | 0002H | IRQBTM RF: 3CH, bit 0 | IPBTM RF: 2FH, bit 3 | – | Internal | – |
| Serial interface | 5 | 0001H | IRQSIO RF: 3BH, bit 0 | IPSIO RF: 2EH, bit 0 | – | Internal | – |

## 16.2 HARDWARE COMPONENTS OF THE INTERRUPT CONTROL CIRCUIT

The flags of the interrupt control circuit are explained below.

### (1) Interrupt request flag and the interrupt enable flag

The interrupt request flag (IRQxxx) is set to 1 when an interrupt request occurs. When interrupt handling is executed, the flag is automatically cleared to 0.

An interrupt enable flag (IPxxx) is provided for each interrupt request flag. If the flag is 1, an interrupt is enabled. If it is 0, the interrupt is disabled.

### (2) EI/DI instruction

The EI/DI instruction is used to determine whether an accepted interrupt is to be executed.

If the EI instruction is executed, the interrupt enable flag (INTE) for enabling interrupt reception is set to 1 (when the interrupt is received, INTE is cleared to 0). Since the INTE flag is not registered in the register file, flag status cannot be checked by instructions.

The DI instruction clears the INTE flag to 0 and disables all interrupts.

At reset the INTE flag is cleared to 0 and all interrupts are disabled.

**Table 16-2 Interrupt Request Flag and Interrupt Enable Flag**

| Interrupt request flag | Signal for setting the interrupt request flag | Interrupt enable flag |
|---|---|---|
| IRQ | Set by edge detection of an INT pin input signal. A detection edge is selected by IEGMD0 or IEGMD1. | IP |
| IRQTM0 | Set by a match signal from timer 0. | IPTM0 |
| IRQTM1 | Set by a match signal from timer 1. | IPTM1 |
| IRQBTM | Set by an overflow (reference time interval signal) from the basic interval timer. | IPBTM |
| IRQSIO | Set by a serial data transmission end signal from the serial interface. | IPSIO |

# 17. STANDBY FUNCTION

## 17.1 OVERVIEW OF THE STANDBY FUNCTION

The μPD17147GS can reduce its current by using the standby function. The standby function supports STOP and HALT modes.

In the STOP mode, the system clock is stopped and the CPU current is reduced to almost only a leak current. This mode is useful in retaining data memory contents without operating the CPU.

In the HALT mode, the oscillation of the system clock continues. However, the system clock is not supplied to the CPU, stopping CPU operation. In this mode, current reduction is less than that in the STOP mode. However, since the system clock is oscillating, operation can be started immediately after the HALT mode is released. In both STOP and HALT modes, the statuses of the data memory, registers, and output latches of the output port used immediately before the standby mode is set are maintained (except STOP 0000B). Therefore, in order to lower consumption current for the entire system, input/output port statuses should be set beforehand.

Table 17-1 Standby Mode Status

| | | STOP mode | HALT mode |
|---|---|---|---|
| Programmed instruction | | STOP instruction | HALT instruction |
| System clock oscillator | | Oscillation stopped | Oscillation continued |
| Operation status | CPU | • Operation stopped | |
| | RAM | • The contents held immediately before setting standby mode are retained. | |
| | Port | • The status existing immediately before setting standby mode is retained.[Note] | |
| | TM0 | • Operable only when the INT input is selected as the count pulse.<br>• Stopped when the system clock is selected. (The count is retained.) | • Operable |
| | TM1 | • Operation stopped. (The count is reset to 0.) (Count-up is also inhibited.) | • Operable |
| | BTM | • Operation stopped. (The count is retained.) | • Operable |
| | SIO | • Operable only when the external clock is selected as the serial clock.[Note] | • Operable |
| | A/D | • Operation stopped[Note] (ADCR ← 00H) | • Operable |
| | INT | • Operable | • Operable |

**Note** When STOP 0000B is executed including the case when port pins are being used as dual-function pins other than for the port, this port changes to an input port immediately.

**Cautions 1. Always specify a NOP instruction immediately before STOP and HALT instructions.**

**2. When an interrupt request flag and the corresponding interrupt enable flag are both set, and the associated interrupt is specified as the standby mode release condition, the system does not enter the standby mode even if a STOP or HALT instruction is executed.**

## 17.2 HALT MODE

### 17.2.1 Setting HALT Mode

Executing a HALT instruction sets HALT mode.

Operand $b_3b_2b_1b_0$ of the HALT instruction indicates the HALT mode release conditions.

**Table 17-2 HALT Mode Release Conditions**

Format: HALT $b_3b_2b_1b_0$B

| Bit | HALT mode release conditions[Note 1] |
|---|---|
| $b_3$ | When this bit is 1, release by IRQxxx is permitted.[Note 2] |
| $b_2$ | Always 0 |
| $b_1$ | When this bit is 1, forced release by IRQTM1 is permitted.[Notes 2, 3] |
| $b_0$ | When this bit is 1, release by $\overline{RLS}$ input is permitted.[Note 2] |

**Notes 1.** When HALT 0000B is specified, HALT mode can be released only by reset ($\overline{RESET}$ input or POC).

**2.** IPxxx must be 1. If a HALT instruction is executed when IRQxxx = 1, the HALT instruction is ignored (treated as a NOP instruction), and HALT mode is not set.

**3.** HALT mode is released regardless of the IPTM1 status.

### 17.2.2 Starting Address After HALT Mode Is Released

The starting address depends on the release conditions and interrupt enable conditions.

**Table 17-3 Starting Address After HALT Mode Is Released**

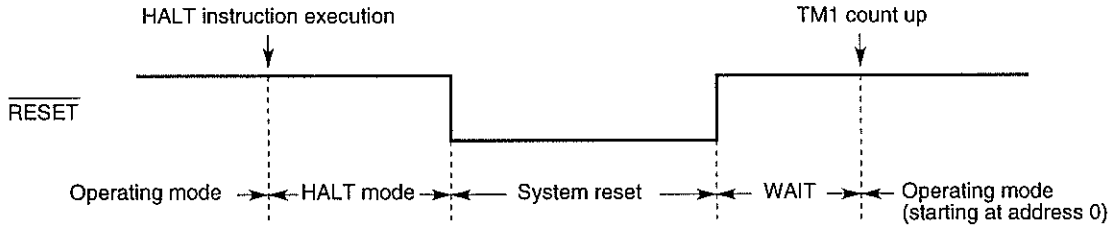
| Release condition | Starting address after release |
|---|---|
| Reset[Note 1] | Address 0 |
| $\overline{RLS}$ | Address subsequent to the HALT instruction |
| IRQxxx[Note 2] | For DI, address subsequent to the HALT instruction |
| | For EI, interrupt vector<br>(When more than one IRQxxx is set, the interrupt vector having the highest priority) |

**Notes 1.** $\overline{RESET}$ input and POC are valid.

**2.** Except when forced release is made with IRQTM1, IPxxx must be 1.

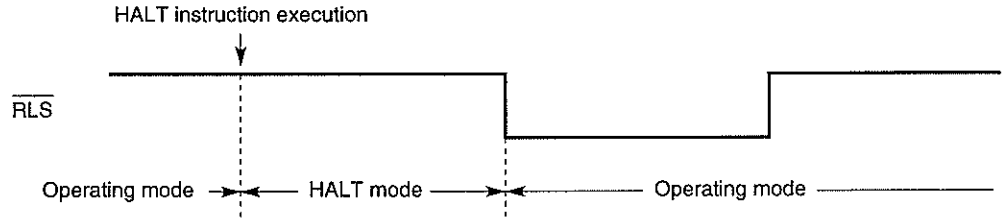
Fig. 17-1 Releasing HALT Mode

(a) Releasing HALT mode by $\overline{\text{RESET}}$ input

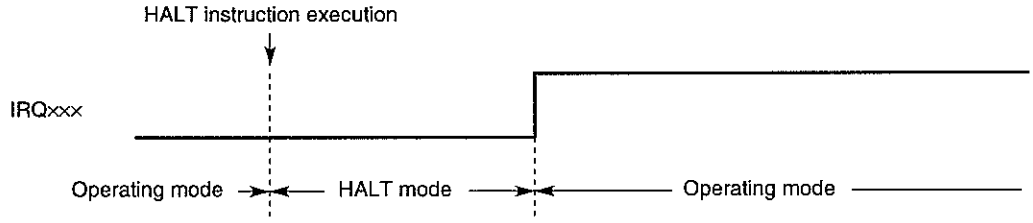



WAIT : Wait time until TM1 counts 256 source clock pulses (system clock/128)
$256 \times 128/fx + \alpha$ (approx. 4 ms + α, at fx = 8 MHz)
α : Oscillation development time (which depends on the resonator)

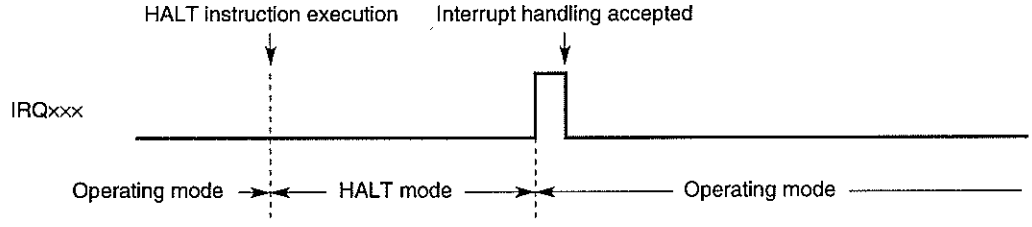
(b) Releasing HALT mode by $\overline{\text{RLS}}$ input




(c) Releasing HALT mode by IRQxxx (for DI status)




(d) Releasing HALT mode by IRQxxx (for EI status)



## 17.3 STOP MODE

### 17.3.1 Setting STOP Mode

Executing a STOP instruction results in STOP mode being set.

Operand $b_3b_2b_1b_0$ of the STOP instruction indicates the STOP mode release conditions.

**Table 17-4 STOP Mode Release Conditions**

Format: STOP $b_3b_2b_1b_0$B

| Bit | STOP mode release condition[Note 1] |
|---|---|
| $b_3$ | When this bit is 1, release by IRQxxx is permitted.[Notes 2, 4] |
| $b_2$ | Always 0 |
| $b_1$ | Always 0 |
| $b_0$ | When this bit is 1, release by $\overline{RLS}$ input is permitted.[Notes 3, 4] |

**Notes 1.** When STOP 0000B is specified, STOP mode can be released only with reset ($\overline{RESET}$ input or POC). When STOP 0000B is executed, the microcomputer is initialized to the state existing immediately after the reset.

**2.** IPxxx must be 1. STOP mode cannot be released with IRQTM1.

**3.** Setting only $b_0$ to 1 is not allowed. (STOP 0001B is inhibited.) To set $b_0$ to 1, $b_3$ must also be set to 1.

**4.** If the STOP instruction is executed when IRQxxx = 1 or when the $\overline{RLS}$ input is low, the STOP instruction is ignored (treated as a NOP instruction), and STOP mode is not set.

### 17.3.2 Starting Address After STOP Mode Is Released

The starting address depends on the release conditions and interrupt enable conditions.

**Table 17-5 Starting Address After STOP Mode Is Released**

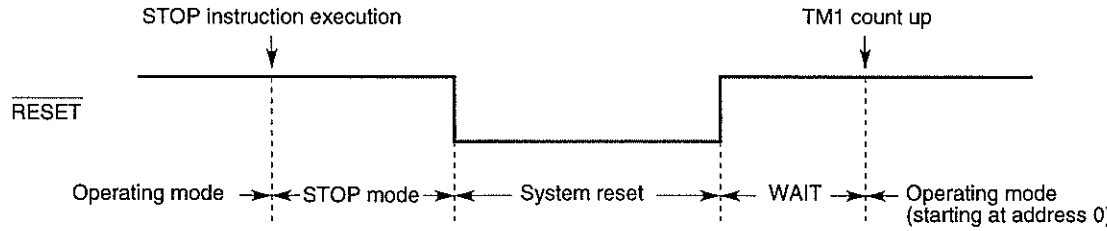
| Release condition | Starting address after release |
|---|---|
| Reset[Note 1] | Address 0 |
| $\overline{RLS}$ | Address subsequent to the STOP instruction |
| IRQxxx[Note 2] | For DI, address subsequent to the STOP instruction |
| | For EI, interrupt vector<br>(When more than one IRQxxx is set, the interrupt vector having the highest priority) |

**Notes 1.** $\overline{RESET}$ input and POC are valid.

**2.** IPxxx must be 1. STOP mode cannot be released with IRQTM1.

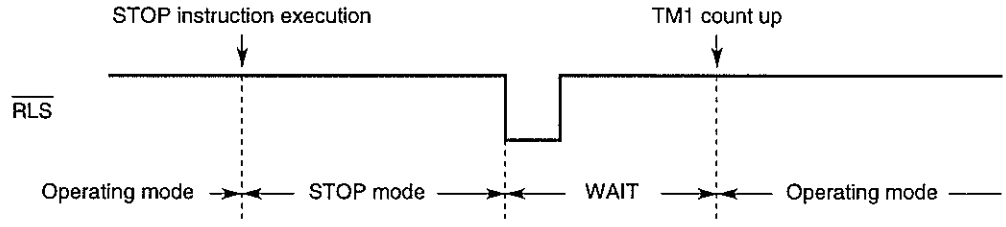
Fig. 17-2 Releasing STOP Mode (1/2)

(a) Releasing STOP mode by $\overline{\text{RESET}}$ input



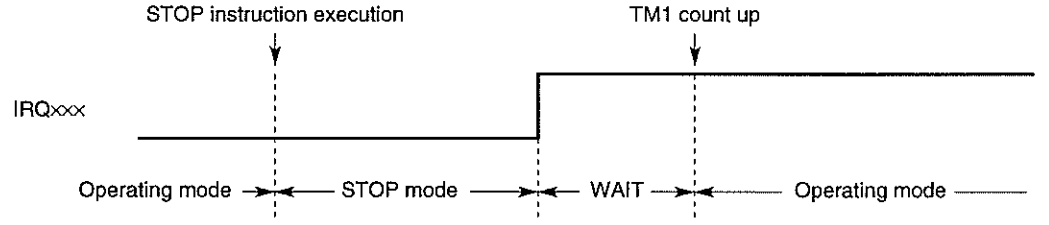

WAIT : Wait time until TM1 counts 256 source clock pulses (system clock/128)
256 × 128/fx + α (approx. 4 ms + α, at fx = 8 MHz)
α : Oscillation development time (which depends on the resonator)

(b) Releasing STOP mode by $\overline{\text{RLS}}$ input




WAIT : Wait time until TM1 counts n+1 source clock pulses (system clock/m)
(n+1) × m/fx + α (n and m are the values used immediately before STOP mode is set)
α : Oscillation development time (which depends on the resonator)

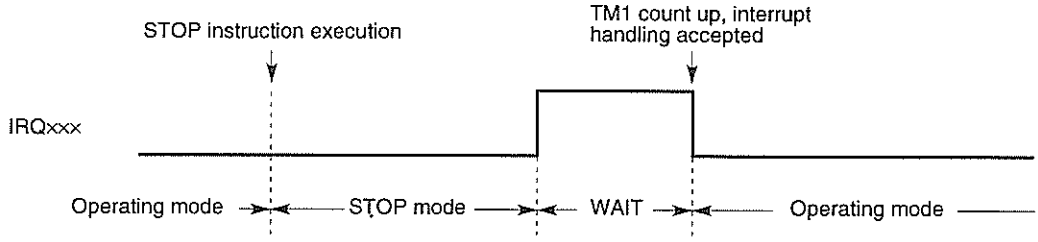
(c) Releasing STOP mode by IRQ×××  (for DI status)




WAIT : Wait time until TM1 counts n+1 source clock pulses (system clock/m)
(n+1) × m/fx + α (n and m are the values used immediately before STOP mode is set)
α : Oscillation development time (which depends on the resonator)

Fig. 17-2 Releasing STOP Mode (2/2)

(d) Releasing STOP mode by IRQxxx (for EI status)




WAIT : Wait time until TM1 counts n+1 source clock pulses (system clock/m)
(n+1) × m/fx + α (n and m are the values used immediately before STOP mode is set)

α : Oscillation development time (which depends on the resonator)

## 18. RESET

The μPD17147GS is reset when a reset signal is applied to the $\overline{\text{RESET}}$ pin, when the incorporated POC circuit detects a supply voltage drop, when the watchdog timer function detects a program crush, or when the address stack overflows or underflows. The incorporated POC circuit is specified with the mask option.

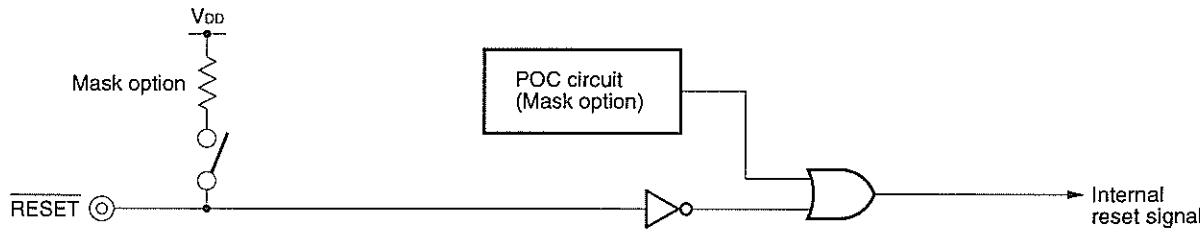
### 18.1 RESET FUNCTIONS

The reset functions are used to initialize device operations. The operations initialized depend on the reset type.

**Table 18-1 Hardware Statuses after Reset**

| Hardware \ Reset type | | • $\overline{\text{RESET}}$ input during operation<br>• Reset by the incorporated POC circuit | • $\overline{\text{RESET}}$ input in the standby mode<br>• Reset by the incorporated POC circuit in the standby mode | • Watchdog timer overflow<br>• Stack overflow or underflow |
|---|---|---|---|---|
| Program counter | | 0000H | 0000H | 0000H |
| Port | Input/output | Input | Input | Input |
| | Output latch content | 0 | 0 | Not defined |
| General-purpose data memory | General- purpose data memory (excluding DBF) | Not defined | Statuses before reset are retained. | Not defined |
| | DBF | Not defined | Not defined | Not defined |
| | System register (excluding WR) | 0 | 0 | 0 |
| | WR | Not defined | Statuses before reset are retained. | Not defined |
| Control register | | SP = 5H, IRQTM1 = 1, TM1EN = 1, IRQBTM = 1, and INT indicate the current status of the INT pin. The others are 0. See **Chapter 8.** | | SP = 5H and INT indicate the current status of the INT pin. The others retain statuses before reset. |
| Timer 0 and timer 1 | Count register | 00H | 00H | Timer 0: 00H<br>Timer 1: Not defined |
| | Modulo register | FFH | FFH | FFH |
| Basic interval timer binary counter | | Not defined | Not defined | Not defined. 40H for watchdog timer overflow |
| Serial interface | Shift register (SIOSFR) | Not defined | Statuses before reset are retained. | Not defined |
| | Output latch | 0 | 0 | Not defined |
| A/D converter data register (ADCR) | | 00H | 00H | 00H |

Fig. 18-1 Reset Block Configuration




## 18.2 RESETTING

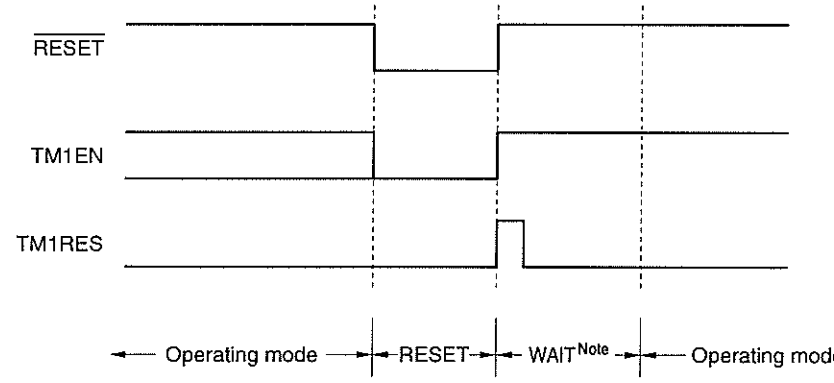
Operation when system reset is caused by $\overline{\text{RESET}}$ input is shown in Fig. 18-2.

If the $\overline{\text{RESET}}$ pin is set from low to high, system clock generation starts and the timer 1 generates an oscillation settling time. Program execution starts from address 0000H.

The controller also operates in the same way when the POC circuit causes a reset.

At watchdog timer overflow reset or address stack register overflow and underflow reset, an oscillation settling time (WAIT) is not generated. Operation starts from address 0000H after initial statuses are internally set.

Fig. 18-2 Resetting




**Note** This is an oscillation settling time. Operating mode is set when timer 1 counts system clocks (fx)128 × 256 times (approx. 4 ms at fx = 8 MHz).

## 19. POC CIRCUIT (MASK OPTION)

The POC circuit monitors the power supply voltage. It resets the microcomputer when the power is turned on/off. It can be used in application circuits using clock frequencies (fx) of between 400 kHz and 4 MHz.

The POC circuit can be included in the μPD17147GS by specifying a mask option.

**Caution The PROM product (μPD17P149) cannot include the POC circuit.**

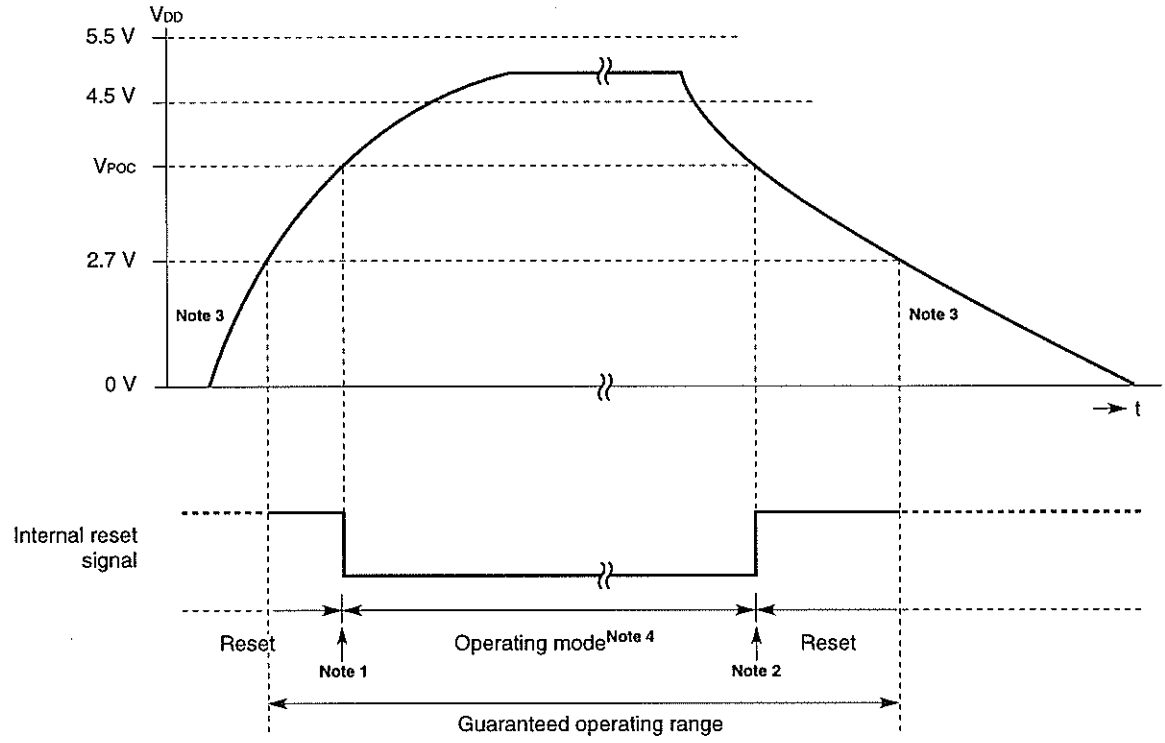
### 19.1 FUNCTIONS OF THE POC CIRCUIT

The POC circuit operates as follows:

- When $V_{DD} \leq V_{POC}$, an internal reset signal is generated.
- When $V_{DD} > V_{POC}$, the internal reset signal is released.

($V_{DD}$: power voltage, $V_{POC}$: POC-detected voltage)

**Fig. 19-1 Operation of the POC Circuit**




**Notes 1.** In actual operation, an oscillation settling time, controlled by timer 1, is inserted before the operating mode is set. The oscillation settling time is equal to the time needed to execute approximately 2048 instructions (approx. 8 ms at fx = 4 MHz).

**2.** A power supply voltage fall can cause a reset only when $V_{POC}$ or a lower voltage level is maintained for a duration equal to at least the reset detection pulse width $t_{SAMP}$. Thus, there is a delay of up to $t_{SAMP}$ before reset.

**3.** When the supply voltage ($V_{DD}$) drops below 2.7 V, operations of all functions of the μPD17147GS are unpredictable. The POC circuit, however, is designed to generate an internal reset signal whenever possible, regardless of whether an oscillation is received. So, an internal reset should occur when the voltage reaches a level at which the internal circuit can operate.

**4.** When the supply voltage rises abruptly (3 V/ms or more), even during operation mode, the POC circuit may generate an internal reset signal to prevent the program running away out of control.

**Remark** For the values of $V_{POC}$ and $t_{SAMP}$, see **Chapter 22.**

### 19.2 CONDITIONS UNDER WHICH THE POC CIRCUIT MAY BE USED

The POC circuit can be used when the application circuit satisfies the following conditions:

- The application circuit is not required to provide high reliability.
- The supply voltage ($V_{DD}$) of the application circuit is between 4.5 V and 5.5 V.
- The system clock frequency ($f_X$) of the application circuit is between 400 kHz and 4 MHz.
- The supply voltage ($V_{DD}$) characteristics satisfy the POC circuit specifications.

**Cautions 1. When the POC circuit is used with an application circuit requiring high reliabilty, be sure to design the POC circuit so that the $\overline{\text{RESET}}$ signal is input from the outside.**

**2. When the POC circuit is used, the current drawn in standby mode will be slightly higher than when the circuit is not used.**

**Remark** POC circuit operation is guranteed at 2.7 to 5.5 V.

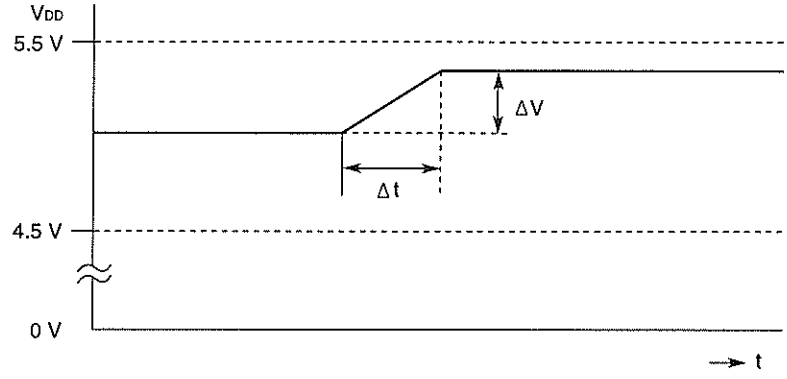
### 19.3 CAUTIONS FOR USING THE POC CIRCUIT

The POC circuit is designed in a fail-safe configuration. It has an auxiliary function[Note] by which if the supply voltage changes abruptly even within the rated range ($V_{DD}$ = 4.5 to 5.5 V), a reset signal is issued as much as possible to avoid program crashes.

Note that if the supply voltage does not meet the conditions listed below, the POC circuit may cause a reset.

- The variation ($\Delta V$) of the supply voltage must be within 100 mV.
- If the variation of the supply voltage is not within 100 mV, the variation slant ($\Delta V/\Delta t$) must be within 3 mV/μs.

**Fig. 19-2 Supply Voltage Variation**




**Note** The reset function based on supply voltage variation is of auxiliary nature. It does not assure that a reset occurs without failure. There is no condition to assure that a reset occurs based on this auxiliary function. Do not count on this function when designing.

## 20. INSTRUCTION SET

### 20.1 OVERVIEW OF THE INSTRUCTION SET

| $b_{14}$-$b_{11}$ \ $b_{15}$ | | 0 | | | 1 | |
|---|---|---|---|---|---|---|
| BIN | HEX | | | | | |
| 0 0 0 0 | 0 | ADD | r, m | | ADD | m, #n4 |
| 0 0 0 1 | 1 | SUB | r, m | | SUB | m, #n4 |
| 0 0 1 0 | 2 | ADDC | r, m | | ADDC | m, #n4 |
| 0 0 1 1 | 3 | SUBC | r, m | | SUBC | m, #n4 |
| 0 1 0 0 | 4 | AND | r, m | | AND | m, #n4 |
| 0 1 0 1 | 5 | XOR | r, m | | XOR | m, #n4 |
| 0 1 1 0 | 6 | OR | r, m | | OR | m, #n4 |
| 0 1 1 1 | 7 | INC | AR | | | |
| | | INC | IX | | | |
| | | MOVT | DBF, @AR | | | |
| | | BR | @AR | | | |
| | | CALL | @AR | | | |
| | | RET | | | | |
| | | RETSK | | | | |
| | | EI | | | | |
| | | DI | | | | |
| | | RETI | | | | |
| | | PUSH | AR | | | |
| | | POP | AR | | | |
| | | GET | DBF, p | | | |
| | | PUT | p, DBF | | | |
| | | PEEK | WR, rf | | | |
| | | POKE | rf, WR | | | |
| | | RORC | r | | | |
| | | STOP | s | | | |
| | | HALT | h | | | |
| | | NOP | | | | |
| 1 0 0 0 | 8 | LD | r, m | | ST | m, r |
| 1 0 0 1 | 9 | SKE | m, #n4 | | SKGE | m, #n4 |
| 1 0 1 0 | A | MOV | @r, m | | MOV | m, @r |
| 1 0 1 1 | B | SKNE | m, #n4 | | SKLT | m, #n4 |
| 1 1 0 0 | C | BR | addr | (page 0) | CALL | addr |
| 1 1 0 1 | D | BR | addr | (page 1) | MOV | m, #n4 |
| 1 1 1 0 | E | | | | SKT | m, #n |
| 1 1 1 1 | F | | | | SKF | m, #n |

## 20.2 LEGEND

| | | |
|---|---|---|
| AR | : | Address register |
| ASR | : | Address stack register pointed to by the stack pointer |
| addr | : | Program memory address (11 low-order bits) |
| BANK | : | Bank register |
| CMP | : | Compare flag |
| CY | : | Carry flag |
| DBF | : | Data buffer |
| h | : | Halt release condition |
| INTEF | : | Interrupt enable flag |
| INTR | : | Register automatically saved in the stack when an interrupt occurs |
| INTSK | : | Interrupt stack register |
| IX | : | Index register |
| MP | : | Data memory row address pointer |
| MPE | : | Memory pointer enable flag |
| m | : | Data memory address specified by $m_R$ and mc |
| $m_R$ | : | Data memory row address (high-order) |
| mc | : | Data memory column address (low-order) |
| n | : | Bit position (four bits) |
| n4 | : | Immediate data (four bits) |
| PAGE | : | Page (bit 11 of the program counter) |
| PC | : | Program counter |
| p | : | Peripheral address |
| $p_H$ | : | Peripheral address (three high-order bits) |
| $p_L$ | : | Peripheral address (four low-order bits) |
| r | : | General register column address |
| rf | : | Register file address |
| $rf_R$ | : | Register file row address (three high-order bits) |
| rfc | : | Register file column address (four low-order bits) |
| SP | : | Stack pointer |
| s | : | Stop release condition |
| WR | : | Window register |
| (×) | : | Contents of × |

## 20.3 LIST OF THE INSTRUCTION SET

| Instruction set | Mnemonic | Operand | Operation | Instruction code | | | |
|---|---|---|---|---|---|---|---|
| | | | | Op code | Operand | | |
| Add | ADD | r, m | (r) ← (r) + (m) | 00000 | $m_R$ | mc | r |
| | | m, #n4 | (m) ← (m) + n4 | 10000 | $m_R$ | mc | n4 |
| | ADDC | r, m | (r) ← (r) + (m) + CY | 00010 | $m_R$ | mc | r |
| | | m, #n4 | (m) ← (m) + n4 + CY | 10010 | $m_R$ | mc | n4 |
| | INC | AR | AR ← AR + 1 | 00111 | 000 | 1001 | 0000 |
| | | IX | IX ← IX + 1 | 00111 | 000 | 1000 | 0000 |
| Subtract | SUB | r, m | (r) ← (r) − (m) | 00001 | $m_R$ | mc | r |
| | | m, #n4 | (m) ← (m) − n4 | 10001 | $m_R$ | mc | n4 |
| | SUBC | r, m | (r) ← (r) − (m) − CY | 00011 | $m_R$ | mc | r |
| | | m, #n4 | (m) ← (m) − n4 − CY | 10011 | $m_R$ | mc | n4 |
| Logical operation | OR | r, m | (r) ← (r) ∨ (m) | 00110 | $m_R$ | mc | r |
| | | m, #n4 | (m) ← (m) ∨ n4 | 10110 | $m_R$ | mc | n4 |
| | AND | r, m | (r) ← (r) ∧ (m) | 00100 | $m_R$ | mc | r |
| | | m, #n4 | (m) ← (m) ∧ n4 | 10100 | $m_R$ | mc | n4 |
| | XOR | r, m | (r) ← (r) ⊻ (m) | 00101 | $m_R$ | mc | r |
| | | m, #n4 | (m) ← (m) ⊻ n4 | 10101 | $m_R$ | mc | n4 |
| Test | SKT | m, #n | CMP ← 0, if (m) ∧ n = n, then skip | 11110 | $m_R$ | mc | n |
| | SKF | m, #n | CMP ← 0, if (m) ∧ n = 0, then skip | 11111 | $m_R$ | mc | n |
| Compare | SKE | m, #n4 | (m) − n4, skip if zero | 01001 | $m_R$ | mc | n4 |
| | SKNE | m, #n4 | (m) − n4, skip if not zero | 01011 | $m_R$ | mc | n4 |
| | SKGE | m, #n4 | (m) − n4, skip if not borrow | 11001 | $m_R$ | mc | n4 |
| | SKLT | m, #n4 | (m) − n4, skip if borrow | 11011 | $m_R$ | mc | n4 |
| Rotation | RORC | r | ┌→ CY → (r)$b_3$ → (r)$b_2$ → (r)$b_1$ → (r)$b_0$ ┐ | 00111 | 000 | 0111 | r |
| Transfer | LD | r, m | (r) ← (m) | 01000 | $m_R$ | mc | r |
| | ST | m, r | (m) ← (r) | 11000 | $m_R$ | mc | r |
| | MOV | @r, m | if MPE = 1: (MP, (r)) ← (m)<br>if MPE = 0: (BANK, $m_R$, (r)) ← (m) | 01010 | $m_R$ | mc | r |
| | | m, @r | if MPE = 1: (m) ← (MP, (r))<br>if MPE = 0: (m) ← (BANK, $m_R$, (r)) | 11010 | $m_R$ | mc | r |
| | | m, #n4 | (m) ← n4 | 11101 | $m_R$ | mc | n4 |
| | MOVT | DBF, @AR | SP ← SP − 1, ASR ← PC, PC ← AR,<br>DBF ← (PC), PC ← ASR, SP ← SP + 1 | 00111 | 000 | 0001 | 0000 |
| | PUSH | AR | SP ← SP − 1, ASR ← AR | 00111 | 000 | 1101 | 0000 |
| | POP | AR | AR ← ASR, SP ← SP + 1 | 00111 | 000 | 1100 | 0000 |
| | PEEK | WR, rf | WR ← (rf) | 00111 | $rf_R$ | 0011 | rfc |
| | POKE | rf, WR | (rf) ← WR | 00111 | $rf_R$ | 0010 | rfc |
| | GET | DBF, p | DBF ← (p) | 00111 | $p_H$ | 1011 | $p_L$ |
| | PUT | p, DBF | (p) ← DBF | 00111 | $p_H$ | 1010 | $p_L$ |

| Instruction set | Mnemonic | Operand | Operation | Instruction code | | | |
|---|---|---|---|---|---|---|---|
| | | | | Op code | Operand | | |
| Branch | BR | addr | PC ← addr, PAGE ← 0 | 01100 | addr | | |
| | | @AR | PC ← AR | 00111 | 000 | 0100 | 0000 |
| Sub-routine | CALL | addr | SP ← SP – 1, ASR ← PC, PC ← addr | 11100 | addr | | |
| | | @AR | SP ← SP – 1, ASR ← PC, PC ← AR | 00111 | 000 | 0101 | 0000 |
| | RET | | PC ← ASR, SP ← SP + 1 | 00111 | 000 | 1110 | 0000 |
| | RETSK | | PC ← ASR, SP ← SP + 1 and skip | 00111 | 001 | 1110 | 0000 |
| | RETI | | PC ← ASR, INTR ← INTSK, SP ← SP + 1 | 00111 | 100 | 1110 | 0000 |
| Interrupt | EI | | INTEF ← 1 | 00111 | 000 | 1111 | 0000 |
| | DI | | INTEF ← 0 | 00111 | 001 | 1111 | 0000 |
| Others | STOP | s | STOP | 00111 | 010 | 1111 | s |
| | HALT | h | HALT | 00111 | 011 | 1111 | h |
| | NOP | | No operation | 00111 | 100 | 1111 | 0000 |

## 20.4 ASSEMBLER (AS17K) BUILT-IN MACRO INSTRUCTIONS

**Legend**

flag n : Flag type symbol

< > : Data < > is omissible.

| | Mnemonic | Operand | Operation | n |
|---|---|---|---|---|
| Built-in macro | SKTn | flag 1, ···flag n | if (flag 1) to (flag n) = all "1", then skip | 1 ≤ n ≤ 4 |
| | SKFn | flag 1, ···flag n | if (flag 1) to (flag n) = all "0", then skip | 1 ≤ n ≤ 4 |
| | SETn | flag 1, ···flag n | (flag 1) to (flag n) ← 1 | 1 ≤ n ≤ 4 |
| | CLRn | flag 1, ···flag n | (flag 1) to (flag n) ← 0 | 1 ≤ n ≤ 4 |
| | NOTn | flag 1, ···flag n | if (flag n) = "0", then (flag n) ← 1<br>if (flag n) = "1", then (flag n) ← 0 | 1 ≤ n ≤ 4 |
| | INITFLG | <NOT> flag 1,<br>···<<NOT> flag n> | if description = NOT flag n, then (flag n) ← 0<br>if description = flag n, then (flag n) ← 1 | 1 ≤ n ≤ 4 |
| | BANKn | | (BANK) ← n | n = 0 |

# 21. ASSEMBLER RESERVED WORDS

## 21.1 MASK OPTION PSEUDO INSTRUCTIONS

The μPD17147GS has the following mask options.

- Built-in pull-up resistor for pin $\overline{\text{RESET}}$
- Built-in pull-up resistors for pins $P0F_1$ and $P0F_0$
- Built-in pull-up resistor for pin INT
- Incorporated POC circuit

Specify whether these mask options are used in the source program using mask-option-definition pseudo instructions.

### 21.1.1 Specifying Mask Options

Mask options are coded in the assembler source program with the following pseudo instructions.

- OPTION and ENDOP pseudo instructions
- Mask-option-definition pseudo instructions

#### (1) OPTION and ENDOP pseudo instructions

These pseudo instructions specify the block in which mask options are coded (mask option definition block).

To specify a mask option, code the corresponding mask-option-definition pseudo instruction in the block enclosed with the OPTION pseudo instruction and ENDOP pseudo instruction.

Format:

| Symbol | Mnemonic | Operand | Comment |
|---|---|---|---|
| [label:] | OPTION | | [;comment] |
| | • | | |
| | • | | |
| | • | | |
| | • | | |
| | • | | |
| | ENDOP | | |

### (2) Mask option definition pseudo instructions

**Table 21-1 Mask Option Definition Pseudo Instructions**

| Option | Definition pseudo instruction and format | Operand | Meaning |
|---|---|---|---|
| Built-in pull-up resistor for pin $\overline{\text{RESET}}$ | OPTRES <operand> | OPEN | Without pull-up resistor |
| | | PULLUP | With pull-up resistor |
| Built-in pull-up resistors for pins $P0F_1$ and $P0F_0$ | OPT0F <operand-1>, <operand-2>[Note] | OPEN | Without pull-up resistor |
| | | PULLUP | With pull-up resistor |
| Built-in pull-up resistor for pin INT | OPTINT <operand> | OPEN | Without pull-up resistor |
| | | PULLUP | With pull-up resistor |
| Incorporated POC circuit | OPTPOC <operand> | NOUSE | Without POC circuit |
| | | USE | With POC circuit |

**Note** <operand-1> and <operand-2> specify the mask options for the $P0F_1$ and $P0F_0$ pins, respectively.

### (3) Example of specifying mask options

```
; Example of specifying mask options in μPD17147GS

MASK_OPTION:

    OPTION                  ; Beginning of mask option definition block
    OPTRES   PULLUP         ; RESET pin has the built-in pull-up resistor.
    OPTP0F   PULLUP, OPEN   ; P0F1 pin has the built-in pull-up resistor. P0F0 pin leaves open (external pull-
                              up).
    OPTINT   PULLUP         ; INT pin has the built-in pull-up resistor.
    OPTPOC   NOUSE          ; Internal POC circuit is not used.
    ENDOP                   ; End of mask option definition block
```

## 21.2 RESERVED SYMBOLS

The reserved symbols defined in the device file (AS17145) are listed below.

### System register (SYSREG)

| Symbolic name | Attribute | Value | Read/ write | Description |
|---|---|---|---|---|
| AR3 | MEM | 0.74H | R | Bits 15 to 12 of the address register |
| AR2 | MEM | 0.75H | R/W | Bits 11 to 8 of the address register |
| AR1 | MEM | 0.76H | R/W | Bits 7 to 4 of the address register |
| AR0 | MEM | 0.77H | R/W | Bits 3 to 0 of the address register |
| WR | MEM | 0.78H | R/W | Window register |
| BANK | MEM | 0.79H | R/W | Bank register |
| IXH | MEM | 0.7AH | R/W | Index register high |
| MPH | MEM | 0.7AH | R/W | Data memory row address pointer high |
| MPE | FLG | 0.7AH.3 | R/W | Memory pointer enable flag |
| IXM | MEM | 0.7BH | R/W | Index register middle |
| MPL | MEM | 0.7BH | R/W | Data memory row address pointer low |
| IXL | MEM | 0.7CH | R/W | Index register low |
| RPH | MEM | 0.7DH | R/W | General register pointer high |
| RPL | MEM | 0.7EH | R/W | General register pointer low |
| PSW | MEM | 0.7FH | R/W | Program status word |
| BCD | FLG | 0.7EH.0 | R/W | BCD flag |
| CMP | FLG | 0.7FH.3 | R/W | Compare flag |
| CY | FLG | 0.7FH.2 | R/W | Carry flag |
| Z | FLG | 0.7FH.1 | R/W | Zero flag |
| IXE | FLG | 0.7FH.0 | R/W | Index enable flag |

**Fig. 21-1 System Register Configuration**

| Address | 74H | | | | 75H | | | | 76H | | | | 77H | | | | 78H | | | | 79H | | | | 7AH | | | | 7BH | | | | 7CH | | | | 7DH | | | | 7EH | | | | 7FH | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Address register (AR) | | | | | | | | | | | | | | | | Window register (WR) | | | | Bank register (BANK) | | | | Index register (IX) Data memory row address pointer (MP) | | | | | | | | | | | | General register pointer (RP) | | | | | | | | Program status word (PSWORD) | | | |
| Symbol | AR3 | | | | AR2 | | | | AR1 | | | | AR0 | | | | WR | | | | BANK | | | | IXH MPH | | | | IXM MPL | | | | IXL | | | | RPH | | | | RPL | | | | PSW | | | |
| Bit | $b_3$ | $b_2$ | $b_1$ | $b_0$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ |
| Data[Note] | 0 | 0 | 0 | 0 | 0 | (AR) | | | | | | | | | | | ← | | | → | 0 | 0 | 0 | 0 | MPE | 0 | 0 | 0 | 0 | (MP) | | | (IX) | | | | → | 0 | 0 | 0 | 0 | (RP) | | | | BCD | CMP | CY | Z | IXE |
| Initial value when reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Not defined | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Note** A bit indicating zero is fixed to zero.

### Data buffer (DBF)

| Symbolic name | Attribute | Value | Read/ write | Description |
|---|---|---|---|---|
| DBF3 | MEM | 0.0CH | R/W | DBF bits 15 to 12 |
| DBF2 | MEM | 0.0DH | R/W | DBF bits 11 to 8 |
| DBF1 | MEM | 0.0EH | R/W | DBF bits 7 to 4 |
| DBF0 | MEM | 0.0FH | R/W | DBF bits 3 to 0 |

### Port register

| Symbolic name | Attribute | Value | Read/ write | Description |
|---|---|---|---|---|
| P0A3 | FLG | 0.70H.3 | R/W | Port 0A bit 3 |
| P0A2 | FLG | 0.70H.2 | R/W | Port 0A bit 2 |
| P0A1 | FLG | 0.70H.1 | R/W | Port 0A bit 1 |
| P0A0 | FLG | 0.70H.0 | R/W | Port 0A bit 0 |
| P0B3 | FLG | 0.71H.3 | R/W | Port 0B bit 3 |
| P0B2 | FLG | 0.71H.2 | R/W | Port 0B bit 2 |
| P0B1 | FLG | 0.71H.1 | R/W | Port 0B bit 1 |
| P0B0 | FLG | 0.71H.0 | R/W | Port 0B bit 0 |
| P0C3 | FLG | 0.72H.3 | R/W | Port 0C bit 3 |
| P0C2 | FLG | 0.72H.2 | R/W | Port 0C bit 2 |
| P0C1 | FLG | 0.72H.1 | R/W | Port 0C bit 1 |
| P0C0 | FLG | 0.72H.0 | R/W | Port 0C bit 0 |
| P0D3 | FLG | 0.73H.3 | R/W | Port 0D bit 3 |
| P0D2 | FLG | 0.73H.2 | R/W | Port 0D bit 2 |
| P0D1 | FLG | 0.73H.1 | R/W | Port 0D bit 1 |
| P0D0 | FLG | 0.73H.0 | R/W | Port 0D bit 0 |
| P0E3 | FLG | 0.6EH.3 | R/W | Port 0E bit 3 |
| P0E2 | FLG | 0.6EH.2 | R/W | Port 0E bit 2 |
| P0E1 | FLG | 0.6EH.1 | R/W | Port 0E bit 1 |
| P0E0 | FLG | 0.6EH.0 | R/W | Port 0E bit 0 |
| P0F1 | FLG | 0.6FH.1 | R | Port 0F bit 1 |
| P0F0 | FLG | 0.6FH.0 | R | Port 0F bit 0 |

**Register file (control register)**

(1/2)

| Symbolic name | At-tribute | Value | Read/ write | Description |
|---|---|---|---|---|
| SP | MEM | 0.81H | R/W | Stack pointer |
| SIOTS | FLG | 0.82H.3 | R/W | Serial interface start flag |
| SIOHIZ | FLG | 0.82H.2 | R/W | Function selection flag for the $P0D_1$/SO pin |
| SIOCK1 | FLG | 0.82H.1 | R/W | Bit 1 of serial clock selection flag |
| SIOCK0 | FLG | 0.82H.0 | R/W | Bit 0 of serial clock selection flag |
| WDTRES | FLG | 0.83H.3 | R/W | Watchdog timer reset flag |
| WDTEN | FLG | 0.83H.0 | R/W | Watchdog timer enable flag |
| TM1OSEL | FLG | 0.8BH.3 | R/W | Function selection flag for the $P0D_3$/$\overline{TM1OUT}$ pin |
| SIOEN | FLG | 0.8BH.0 | R/W | Serial interface enable flag |
| P0EGPU | FLG | 0.8CH.2 | R/W | P0E group pull-up selection flag (1: Pull-up) |
| P0BGPU | FLG | 0.8CH.1 | R/W | P0B group pull-up selection flag (1: Pull-up) |
| P0AGPU | FLG | 0.8CH.0 | R/W | P0A group pull-up selection flag (1: Pull-up) |
| P0DBPU3 | FLG | 0.8DH.3 | R/W | $P0D_3$ pull-up selection flag (1: Pull-up) |
| P0DBPU2 | FLG | 0.8DH.2 | R/W | $P0D_2$ pull-up selection flag (1: Pull-up) |
| P0DBPU1 | FLG | 0.8DH.1 | R/W | $P0D_1$ pull-up selection flag (1: Pull-up) |
| P0DBPU0 | FLG | 0.8DH.0 | R/W | $P0D_0$ pull-up selection flag (1: Pull-up) |
| INT | FLG | 0.8FH.0 | R | INT pin status flag |
| TM0EN | FLG | 0.91H.3 | R/W | Timer 0 enable flag |
| TM0RES | FLG | 0.91H.2 | R/W | Timer 0 reset flag |
| TM0CK1 | FLG | 0.91H.1 | R/W | Bit 1 of count pulse selection flag for timer 0 |
| TM0CK0 | FLG | 0.91H.0 | R/W | Bit 0 of count pulse selection flag for timer 0 |
| TM1EN | FLG | 0.92H.3 | R/W | Timer 1 enable flag |
| TM1RES | FLG | 0.92H.2 | R/W | Timer 1 reset flag |
| TM1CK1 | FLG | 0.92H.1 | R/W | Bit 1 of count pulse selection flag for timer 1 |
| TM1CK0 | FLG | 0.92H.0 | R/W | Bit 0 of count pulse selection flag for timer 1 |
| BTMISEL | FLG | 0.93H.3 | R/W | Interrupt request clock selection flag for the basic interval timer |
| BTMRES | FLG | 0.93H.2 | R/W | Basic interval timer reset flag |
| BTMCK1 | FLG | 0.93H.1 | R/W | Bit 1 of count pulse selection flag for the basic interval timer |
| BTMCK0 | FLG | 0.93H.0 | R/W | Bit 0 of count pulse selection flag for the basic interval timer |
| P0C3IDI | FLG | 0.9BH.3 | R/W | $ADC_3$/$P0C_3$ pin function selection flag |
| P0C2IDI | FLG | 0.9BH.2 | R/W | $ADC_2$/$P0C_2$ pin function selection flag |
| P0C1IDI | FLG | 0.9BH.1 | R/W | $ADC_1$/$P0C_1$ pin function selection flag |
| P0C0IDI | FLG | 0.9BH.0 | R/W | $ADC_0$/$P0C_0$ pin function selection flag |
| P0CBIO3 | FLG | 0.9CH.3 | R/W | Input/output selection flag for $P0C_3$ (1: Output port) |
| P0CBIO2 | FLG | 0.9CH.2 | R/W | Input/output selection flag for $P0C_2$ (1: Output port) |
| P0CBIO1 | FLG | 0.9CH.1 | R/W | Input/output selection flag for $P0C_1$ (1: Output port) |
| P0CBIO0 | FLG | 0.9CH.0 | R/W | Input/output selection flag for $P0C_0$ (1: Output port) |
| IEGMD1 | FLG | 0.9FH.1 | R/W | Bit 1 of detection edge selection flag for the INT pin |
| IEGMD0 | FLG | 0.9FH.0 | R/W | Bit 0 of detection edge selection flag for the INT pin |

**Register file (control register)**

(2/2)

| Symbolic name | At-tribute | Value | Read/ write | Description |
|---|---|---|---|---|
| ADCSTRT | FLG | 0.0A0H.0 | R/W | A/D converter start flag (always 0 when read) |
| ADCSOFT | FLG | 0.0A1H.3 | R/W | A/D converter operation mode selection flag (1: Single mode) |
| ADCCMP | FLG | 0.0A1H.1 | R | A/D converter comparison result flag (valid only in the single mode) |
| ADCEND | FLG | 0.0A1H.0 | R | A/D converter conversion end flag |
| ADCCH3 | FLG | 0.0A2H.3 | R/W | Dummy flag |
| ADCCH2 | FLG | 0.0A2H.2 | R/W | Dummy flag |
| ADCCH1 | FLG | 0.0A2H.1 | R/W | Bit 1 of channel selection flag for the A/D converter |
| ADCCH0 | FLG | 0.0A2H.0 | R/W | Bit 0 of channel selection flag for the A/D converter |
| P0DBIO3 | FLG | 0.0ABH.3 | R/W | Input/output selection flag for $P0D_3$ (1: Output port) |
| P0DBIO2 | FLG | 0.0ABH.2 | R/W | Input/output selection flag for $P0D_2$ (1: Output port) |
| P0DBIO1 | FLG | 0.0ABH.1 | R/W | Input/output selection flag for $P0D_1$ (1: Output port) |
| P0DBIO0 | FLG | 0.0ABH.0 | R/W | Input/output selection flag for $P0D_0$ (1: Output port) |
| P0EGIO | FLG | 0.0ACH.2 | R/W | Group input/output selection flag for P0E (1: All P0E pins are output ports.) |
| P0BGIO | FLG | 0.0ACH.1 | R/W | Group input/output selection flag for P0B (1: All P0B pins are output ports.) |
| P0AGIO | FLG | 0.0ACH.0 | R/W | Group input/output selection flag for P0A (1: All P0A pins are output ports.) |
| IPSIO | FLG | 0.0AEH.0 | R/W | Interrupt enable flag for the serial interface |
| IPBTM | FLG | 0.0AFH.3 | R/W | Interrupt enable flag for the basic interval timer |
| IPTM1 | FLG | 0.0AFH.2 | R/W | Interrupt enable flag for timer 1 |
| IPTM0 | FLG | 0.0AFH.1 | R/W | Interrupt enable flag for timer 0 |
| IP | FLG | 0.0AFH.0 | R/W | Interrupt enable flag for the INT pin |
| IRQSIO | FLG | 0.0BBH.0 | R/W | Interrupt request flag for the serial interface |
| IRQBTM | FLG | 0.0BCH.0 | R/W | Interrupt request flag for the basic interval timer |
| IRQTM1 | FLG | 0.0BDH.0 | R/W | Interrupt request flag for timer 1 |
| IRQTM0 | FLG | 0.0BEH.0 | R/W | Interrupt request flag for timer 0 |
| IRQ | FLG | 0.0BFH.0 | R/W | Interrupt request flag for the INT pin |

### Peripheral hardware register

| Symbolic name | At-tribute | Value | Read/ write | Description |
|---|---|---|---|---|
| SIOSFR | DAT | 01H | R/W | Peripheral address of the shift register |
| TM0M | DAT | 02H | W | Peripheral address of the timer 0 modulo register |
| TM1M | DAT | 03H | W | Peripheral address of the timer 1 modulo register |
| ADCR | DAT | 04H | R/W | Peripheral address of A/D converter data register |
| TM0TM1C | DAT | 45H | R | Peripheral address of timer 0 timer 1 count register |
| AR | DAT | 40H | R/W | Peripheral address of the address register for GET, PUT, PUSH, CALL, BR, MOVT, and INC instructions |

### Others

| Symbolic name | At-tribute | Value | Description |
|---|---|---|---|
| DBF | DAT | 0FH | Fixed operand value for a GET/PUT/MOVT instruction |
| IX | DAT | 01H | Fixed operand value for an INC instruction |

Fig. 21-2 Control Register Configuration (1/2)

| Row address | Item | Column address 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|---|
| 0 (8) | Symbol | | 0 SP | SIOTS SIOHIZ SIOCK1 SIOCK0 | WDTRES 0 0 WDTEN | | | | |
| | When reset | | 0 1 0 1 | 0 0 0 0 | 0 0 0 0 | | | | |
| | Read/ Write | | R/W | R/W | R/W | | | | |
| 1 (9) | Symbol | | TM0EN TM0RES TM0CK1 TM0CK0 | TM1EN TM1RES TM1CK1 TM1CK0 | BTMISEL BTMRES BTMCK1 BTMCK0 | | | | |
| | When reset | | 0 0 0 0 | 1 0 0 0 | 0 0 0 0 | | | | |
| | Read/ Write | | R/W | R/W | R/W | | | | |
| 2 (A) | Symbol | 0 0 0 ADCSTRT | ADCSOFT 0 ADCCMP ADCEND | ADCCH3 ADCCH2 ADCCH1 ADCCH0 | | | | | |
| | When reset | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | | | | | |
| | Read/ Write | R/W | R/W R | R/W | | | | | |
| 3 (B) | Symbol | | | | | | | | |
| | When reset | | | | | | | | |
| | Read/ Write | | | | | | | | |

**Remark** The address enclosed in parentheses apply when the assembler is used.
The names of all the flags in the control registers are assembler reserved words saved in the device file. Using these reserved words is useful in programming.

Fig. 21-2 Control Register Configuration (2/2)

| 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|
| | | | TM1OSEL 0 0 SIOEN | 0 P0EGPU P0BGPU P0AGPU | P0DBPU3 P0DBPU2 P0DBPU1 P0DBPU0 | | 0 0 0 INT |
| | | | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | | 0 0 0 Note |
| | | | R/W | R/W | R/W | | R |
| | | | P0C3IDI P0C2IDI P0C1IDI P0C0IDI | P0CBIO3 P0CBIO2 P0CBIO1 P0CBIO0 | | | 0 0 IEGMD1 IEGMD0 |
| | | | 0 0 0 0 | 0 0 0 0 | | | 0 0 0 0 |
| | | | R/W | R/W | | | R/W |
| | | | P0DBIO3 P0DBIO2 P0DBIO1 P0DBIO0 | 0 P0EGIO P0BGIO P0AGIO | | 0 0 0 IPSIO | IPBTM IPTM1 IPTM0 IP |
| | | | 0 0 0 0 | 0 0 0 0 | | 0 0 0 0 | 0 0 0 0 |
| | | | R/W | R/W | | R/W | R/W |
| | | | 0 0 0 IRQSIO | 0 0 0 IRQBTM | 0 0 0 IRQTM1 | 0 0 0 IRQTM0 | 0 0 0 IRQ |
| | | | 0 0 0 0 | 0 0 0 1 | 0 0 0 1 | 0 0 0 0 | 0 0 0 0 |
| | | | R/W | R/W | R/W | R/W | R/W |

**Note** The INT flag depends on the status of the INT pin.

## 22. ELECTRICAL CHARACTERISTICS

### ABSOLUTE MAXIMUM RATINGS ($T_A$ = 25 °C)

| Parameter | Symbol | Conditions | | Rated value | Unit |
|---|---|---|---|---|---|
| Supply voltage | $V_{DD}$ | | | −0.3 to +7.0 | V |
| A/D converter reference voltage | $V_{REF}$ | | | −0.3 to $V_{DD}$ + 0.3 | V |
| Input voltage | $V_I$ | P0A, P0B, P0C, P0D, P0E, P0F, INT, $\overline{RESET}$, and $X_{IN}$ | | −0.3 to $V_{DD}$ + 0.3 | V |
| Output voltage | $V_O$ | | | −0.3 to $V_{DD}$ + 0.3 | V |
| High-level output current | $I_{OH}$ Note | Each of P0A, P0B, and P0C pins | Peak value | −15 | mA |
| | | | rms | −7.5 | mA |
| | | Total of P0A, P0B, and P0C pins | Peak value | −30 | mA |
| | | | rms | −15 | mA |
| Low-level output current | $I_{OL}$ Note | Each of P0A, P0B, and P0C | Peak value | 15 | mA |
| | | | rms | 7.5 | mA |
| | | Each of P0D and P0E | Peak value | 30 | mA |
| | | | rms | 15 | mA |
| | | Total of P0A, P0B, P0C, P0D, and P0E pins | Peak value | 100 | mA |
| | | | rms | 50 | mA |
| Operating temperature | $T_{opt}$ | | | −40 to +85 | °C |
| Storage temperature | $T_{stg}$ | | | −65 to +150 | °C |
| Allowable dissipation | $P_d$ | $T_A$ = 85 °C | 30-pin plastic shrink SOP | 50 | mW |

**Note** Calculate a root-mean-square value as follows: [rms value] = [peak value] × $\sqrt{duty}$.

**Caution Absolute maximum ratings are rated values beyond which some physical damages may be caused to the product; if any of the parameters in the table above exceeds its rated value even for a moment, the quality of the product may deteriorate. Be sure to use the product within the rated values.**

### RECOMMENDED POWER VOLTAGE RANGE ($T_A$ = −40 to +85 °C)

| Parameter | Symbol | Conditions | | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|---|
| Supply voltage | $V_{DD}$ | CPU (except A/D converter and POC circuit) | fx = 400 kHz to 2 MHz | 2.7 | | 5.5 | V |
| | | | fx = 400 kHz to 4 MHz | 3.6 | | 5.5 | V |
| | | | fx = 400 kHz to 8 MHz | 4.5 | | 5.5 | V |
| | | A/D converter | Absolute accuracy: ±1.5 LSB, 2.5 V ≤ $V_{REF}$ ≤ $V_{DD}$ | 4.0 | | 5.5 | V |
| | | POC circuit (mask option) | fx = 400 kHz to 4 MHz | 4.5 | | 5.5 | V |

**DC CHARACTERISTICS ($V_{DD}$ = 2.7 to 5.5 V, $T_A$ = −40 to +85 °C)**

| Parameter | Symbol | Conditions | | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|---|
| High-level input voltage | $V_{IH1}$ | P0A, P0B, P0C, P0D, P0E, and P0F | | 0.7$V_{DD}$ | | $V_{DD}$ | V |
| | $V_{IH2}$ | $\overline{RESET}$, SCK, SI, and INT | | 0.8$V_{DD}$ | | $V_{DD}$ | V |
| | $V_{IH3}$ | $X_{IN}$ | | $V_{DD}$−0.5 | | $V_{DD}$ | V |
| Low-level input voltage | $V_{IL1}$ | P0A, P0B, P0C, P0D, P0E, and P0F | | 0 | | 0.3$V_{DD}$ | V |
| | $V_{IL2}$ | $\overline{RESET}$, SCK, SI, and INT | | 0 | | 0.2$V_{DD}$ | V |
| | $V_{IL3}$ | $X_{IN}$ | | 0 | | 0.4 | V |
| High-level output voltage | $V_{OH}$ | P0A, P0B, and P0C | 4.5 ≤ $V_{DD}$ ≤ 5.5, $I_{OH}$ = −1.0 mA | $V_{DD}$ − 0.3 | | | V |
| | | | 2.7 ≤ $V_{DD}$ < 4.5, $I_{OH}$ = −0.5 mA | $V_{DD}$ − 0.3 | | | V |
| Low-level output voltage | $V_{OL1}$ | P0A, P0B, P0C, P0D, and P0E | 4.5 ≤ $V_{DD}$ ≤ 5.5, $I_{OL}$ = 1.0 mA | | | 0.3 | V |
| | | | 2.7 ≤ $V_{DD}$ < 4.5, $I_{OL}$ = 0.5 mA | | | 0.3 | V |
| | $V_{OL2}$ | P0D and P0E, $I_{OL}$ = 15 mA | 4.5 ≤ $V_{DD}$ ≤ 5.5 | | | 1.0 | V |
| | | | 2.7 ≤ $V_{DD}$ < 4.5 | | | 2.0 | V |
| High-level input leakage current | $I_{LIH}$ | P0A, P0B, P0C, P0D, P0E, P0F, $\overline{RESET}$, and INT, $V_{IN}$ = $V_{DD}$ | | | | 3 | μA |
| Low-level input leakage current | $I_{LIL}$ | P0A, P0B, P0C, P0D, P0E, P0F, $\overline{RESET}$, and INT, $V_{IN}$ = 0 V | | | | −3 | μA |
| High-level output leakage current | $I_{LOH}$ | P0A, P0B, P0C, P0D, and P0E, $V_{OUT}$ = $V_{DD}$ | | | | 3 | μA |
| Low-level output leakage current | $I_{LOL}$ | P0A, P0B, P0C, P0D, and P0E, $V_{OUT}$ = 0 V | | | | −3 | μA |
| Built-in pull-up resistance[Note 1] | $R_{PULL}$ | P0A, P0B, P0E, P0F, $\overline{RESET}$, and INT | | 50 | 100 | 200 | kΩ |
| | | P0D | | 3 | 10 | 30 | kΩ |
| Power supply current[Note 2] | $I_{DD1}$ | Normal operation mode | $f_X$ = 8.0 MHz, $V_{DD}$ = 5 V ±10 % | | 2.0 | 4.5 | mA |
| | | | $f_X$ = 4.0 MHz, $V_{DD}$ = 5 V ±10 % | | 1.4 | 3.3 | mA |
| | | | $f_X$ = 2.0 MHz, $V_{DD}$ = 3 V ±10 % | | 0.5 | 1.5 | mA |
| | | | $f_X$ = 400 kHz, $V_{DD}$ = 5 V ±10 % | | 0.9 | 1.7 | mA |
| | | | $f_X$ = 400 kHz, $V_{DD}$ = 3 V ±10 % | | 0.3 | 1.0 | mA |
| | $I_{DD2}$ | HALT mode | $f_X$ = 8.0 MHz, $V_{DD}$ = 5 V ±10 % | | 1.0 | 2.0 | mA |
| | | | $f_X$ = 4.0 MHz, $V_{DD}$ = 5 V ±10 % | | 0.9 | 1.9 | mA |
| | | | $f_X$ = 2.0 MHz, $V_{DD}$ = 3 V ±10 % | | 0.3 | 1.0 | mA |
| | | | $f_X$ = 400 kHz, $V_{DD}$ = 5 V ±10 % | | 0.7 | 1.5 | mA |
| | | | $f_X$ = 400 kHz, $V_{DD}$ = 3 V ±10 % | | 0.3 | 0.9 | mA |
| | $I_{DD3}$ | STOP mode | $V_{DD}$ = 5 V ±10 % | | 3.0 | 10 | μA |
| | | | $V_{DD}$ = 3 V ±10 % | | 2.0 | 10 | μA |

**Notes 1.** The pull-up resistors for the P0F, $\overline{RESET}$, and INT pins are specified with mask options.

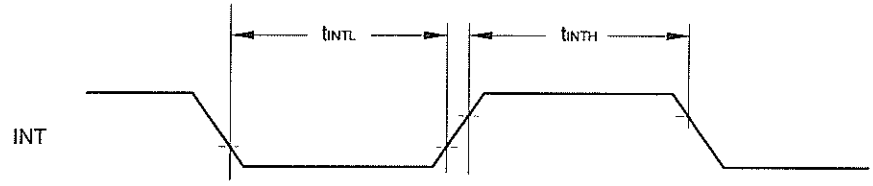
**2.** This current excludes the current which flows through the A/D converter, POC circuit, and built-in pull-up resistors.

**AC CHARACTERISTICS ($V_{DD}$ = 2.7 to 5.5 V, $T_A$ = −40 to +85 °C)**

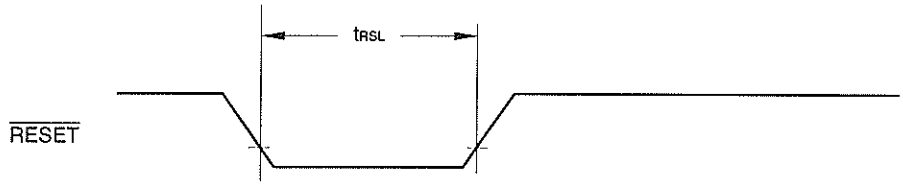
| Parameter | Symbol | Conditions | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| CPU clock cycle time (instruction execution time) | $t_{CY}$ | $V_{DD}$ = 4.5 to 5.5 V | 1.9 | | 41 | μs |
| | | $V_{DD}$ = 3.6 to 5.5 V | 3.9 | | 41 | μs |
| | | | 7.9 | | 41 | μs |
| INT input frequency (TM0 count clock input) | $f_{INT}$ | | 0 | | 400 | kHz |
| INT high/low level width (external interrupt input) | $t_{INTH}$, $t_{INTL}$ | $V_{DD}$ = 4.5 to 5.5 V | 10 | | | μs |
| | | | 50 | | | μs |
| $\overline{RESET}$ low-level width | $t_{RSL}$ | $V_{DD}$ = 4.5 to 5.5 V | 10 | | | μs |
| | | | 50 | | | μs |
| $\overline{RLS}$ low-level width | $t_{RLSL}$ | $V_{DD}$ = 4.5 to 5.5 V | 10 | | | μs |
| | | | 50 | | | μs |

**Remark** $t_{CY}$ = 16/fx (fx: oscillation frequency system clock)

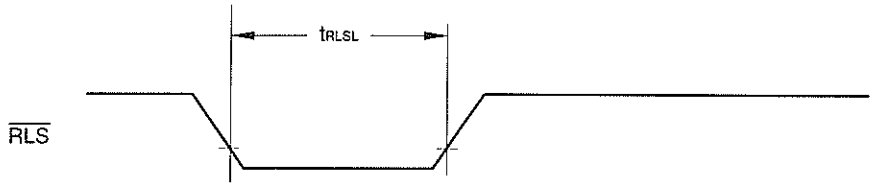
**Interrupt input timing**




**$\overline{RESET}$ input timing**



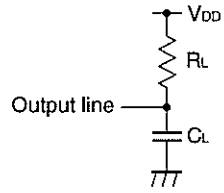

**$\overline{RLS}$ input timing**



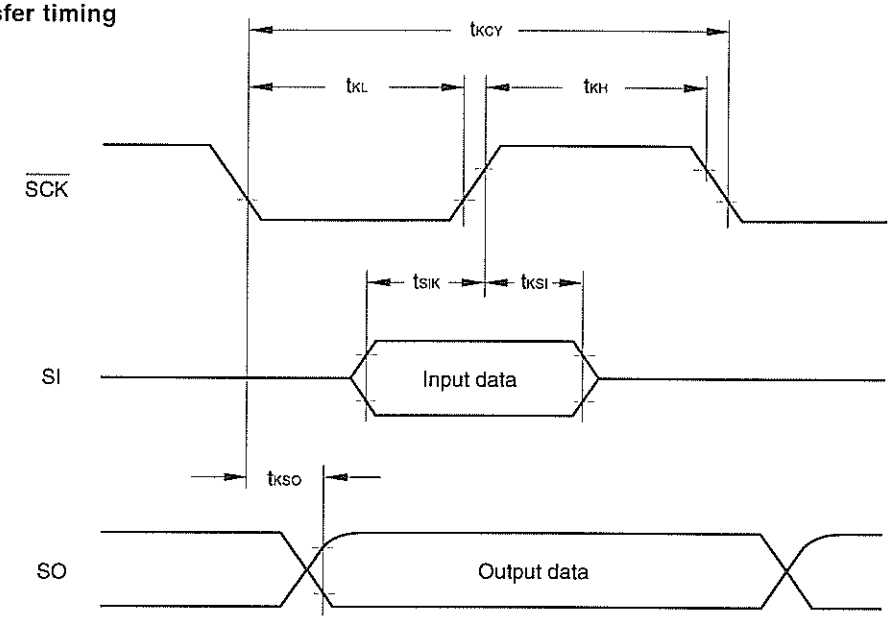
**SERIAL TRANSFER OPERATION ($V_{DD}$ = 2.7 to 5.5 V, $T_A$ = −40 to +85 °C)**

| Parameter | Symbol | | Conditions | | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|---|---|
| $\overline{SCK}$ cycle time | $t_{KCY}$ | Input | $V_{DD}$ = 4.5 to 5.5 V | | 2.0 | | | μs |
| | | | | | 10 | | | μs |
| | | Output | $R_L$ = 1 kΩ, $C_L$ = 100 pF | $V_{DD}$ = 4.5 to 5.5 V | 2.0 | | | μs |
| | | | | | 8 | | | μs |
| | | | Built-in pull-up resistor, $C_L$ = 100 pF | $V_{DD}$ = 4.5 to 5.5 V | 32 | | | μs |
| | | | | | 64 | | | μs |
| $\overline{SCK}$ high/low level width | $t_{KH}$, $t_{KL}$ | Input | $V_{DD}$ = 4.5 to 5.5 V | | 1.0 | | | μs |
| | | | | | 5.0 | | | μs |
| | | Output | $R_L$ = 1 kΩ, $C_L$ = 100 pF | $V_{DD}$ = 4.5 to 5.5 V | $t_{KCY}/2 - 0.6$ | | | μs |
| | | | | | $t_{KCY}/2 - 1.2$ | | | μs |
| | | | Built-in pull-up resistor, $C_L$ = 100 pF | $V_{DD}$ = 4.5 to 5.5 V | $t_{KCY}/2 - 1.2$ | | | μs |
| | | | | | $t_{KCY}/2 - 24$ | | | μs |
| SI setup time (with respect to $\overline{SCK}$↑) | $t_{SIK}$ | | | | 100 | | | ns |
| SI hold time (with respect to $\overline{SCK}$↑) | $t_{KSI}$ | | | | 100 | | | ns |
| Delay from $\overline{SCK}$↓ to SO | $t_{KSO}$ | | $R_L$ = 1 kΩ, $C_L$ = 100 pF | $V_{DD}$ = 4.5 to 5.5 V | | | 0.8 | μs |
| | | | | | | | 1.4 | μs |
| | | | Built-in pull-up resistor, $C_L$ = 100 pF | $V_{DD}$ = 4.5 to 5.5 V | | | 14 | μs |
| | | | | | | | 26 | μs |

**Remark** $R_L$: a resistive load for the output line
$C_L$: a capacitive load for the output line



Serial transfer timing




**A/D CONVERTER ($V_{DD}$ = 4.0 to 5.5 V, $T_A$ = −40 to +85 °C)**

| Parameter | Symbol | Conditions | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| Resolution | | | 8 | 8 | 8 | bit |
| Absolute accuracy[Note 1] | | 2.5 V ≤ $V_{REF}$ ≤ $V_{DD}$ | | | ±1.5 | LSB |
| Conversion time[Note 2] | $t_{CONV}$ | | | | 25 $t_{CY}$ | μs |
| Analog signal input voltage | $V_{ADIN}$ | | 0 | | $V_{REF}$ | V |
| Reference input voltage | $V_{REF}$ | | 2.5 | | $V_{DD}$ | V |
| A/D converter circuit current | $I_{ADC}$ | When A/D converter is operating | | 1.0 | 2.0 | mA |
| $V_{REF}$ pin current | $I_{REF}$ | | | 0.1 | 0.3 | mA |

**Notes 1.** Absolute accuracy excluding quantization error (±0.5 LSB)

**2.** Time from conversion start instruction execution to conversion end (ADCEND = 1) (50 μs at fx = 8 MHz)

**Remark:** $t_{CY}$ = 16/fx (fx: system clock oscillation frequency)

**POC CIRCUIT (MASK OPTION[Note 1]) ($V_{DD}$ = 2.7 to 5.5 V, $T_A$ = −40 to +85 °C)**

| Parameter | Symbol | Conditions | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| POC detection voltage[Note 2] | $V_{POC}$ | | 3.6 | 4.0 | 4.45 | V |
| Supply voltage fall speed | $t_{POCS}$ | | | | 0.08 | V/ms |
| Reset detection pulse width | $t_{SAMP}$ | | 1 | | | ms |
| POC circuit current | $I_{POC}$ | | | 3.0 | 10 | μA |

**Notes 1.** This circuit can be used in an application circuit where $V_{DD}$ is 4.5 to 5.5 V and the clock frequency (fx) ranges from 400 kHz to 4 MHz.

**2.** This is the voltage at which the POC circuit releases the internal reset. When $V_{POC}$ becomes less than $V_{DD}$, the internal reset is released.

## CHARACTERISTICS OF THE SYSTEM CLOCK OSCILLATOR ($V_{DD}$ = 2.7 to 5.5 V, $T_A$ = −40 to +85 °C)

| Resonator[Note] | Parameter | Conditions | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| Ceramic resonator | Oscillation frequency | | 0.39 | | 2.04 | MHz |
| | | $V_{DD}$ = 3.6 to 5.5 V | 0.39 | | 4.08 | MHz |
| | | $V_{DD}$ = 4.5 to 5.5 V | 0.39 | | 8.16 | MHz |

**Note** Do not use a resonator having an oscillation generation time of 2 ms or more.

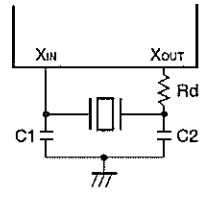
## RECOMMENDED CERAMIC RESONATORS ($T_A$ = −40 to +85 °C)

| Manufacturer | Part number | Recommended constants | | | Oscillation supply voltage [V][Note 1] | | Remarks |
|---|---|---|---|---|---|---|---|
| | | C1 [pF] | C2 [pF] | Rd [kΩ] | Min. | Max. | |
| Murata Mfg. | CSB400P | 220 | 220 | 5.6 | 2.7 | 5.5 | |
| | CSA2. 00MG040 | 100 | 100 | 0 | 2.7 | 5.5 | |
| | CST2. 00MG040 | Not required[Note 2] | | 0 | 2.7 | 5.5 | |
| | CSA4. 00MG | 30 | 30 | 0 | 3.6 | 5.5 | |
| | CST4. 00MGW | Not required[Note 2] | | 0 | 3.6 | 5.5 | |
| | CSA8. 00MG | 30 | 30 | 0 | 4.5 | 5.5 | |
| | CST8. 00MGW | Not required[Note 2] | | 0 | 4.5 | 5.5 | |
| | CSB400JA | 220 | 220 | 5.6 | 2.7 | 5.5 | Electrical equipment for automobiles |
| | CSA2. 00MGA040 | 100 | 100 | 0 | 2.7 | 5.5 | |
| | CST2. 00MGA040 | Not required[Note 2] | | 0 | 2.7 | 5.5 | |
| | CSA4. 00MGA | 30 | 30 | 0 | 3.6 | 5.5 | |
| | CST4. 00MGWA | Not required[Note 2] | | 0 | 3.6 | 5.5 | |
| | CSA8. 00MTZA | 30 | 30 | 0 | 4.5 | 5.5 | |
| | CST8. 00MTWA | Not required[Note 2] | | 0 | 4.5 | 5.5 | |
| Kyocera | KBR-400BK | 470 | 470 | 0 | 2.7 | 5.5 | |
| | KBR-2. 0MS | 47 | 47 | 0 | 2.7 | 5.5 | |
| | KBR-4. 0MSA | 33 | 33 | 0 | 3.6 | 5.5 | |
| | KBR-8. 0M | 33 | 33 | 0 | 4.5 | 5.5 | |
| TDK | FCR400K3 | 330 | 330 | 0 | 2.7 | 5.5 | |
| | FCR2. 0M3 | 100 | 100 | 0 | 2.7 | 5.5 | |
| | FCR2. 0MC3 | Not required[Note 2] | | 1.5 | 2.7 | 5.5 | |
| | FCR4. 0M5 | 33 | 33 | 0 | 3.6 | 5.5 | |
| | CCR4. 0MC3 | Not required[Note 2] | | 0 | 3.6 | 5.5 | |
| | FCR4. 0MC5 | Not required[Note 2] | | 0 | 3.6 | 5.5 | |
| | FCR8. 0M2S | 33 | 33 | 0 | 4.5 | 5.5 | |
| | FCR8. 0MC | Not required[Note 2] | | 0 | 4.5 | 5.5 | |

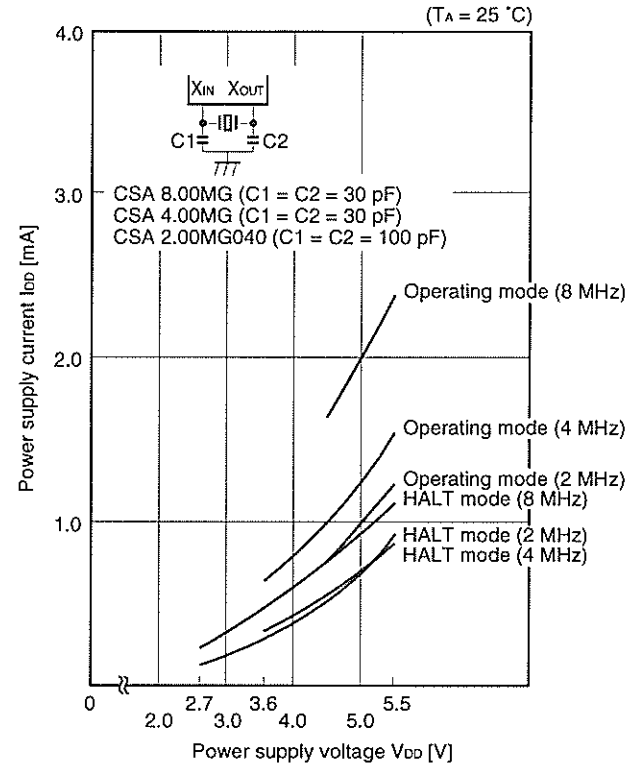
**Notes 1.** When the POC circuit (mask option) is not used

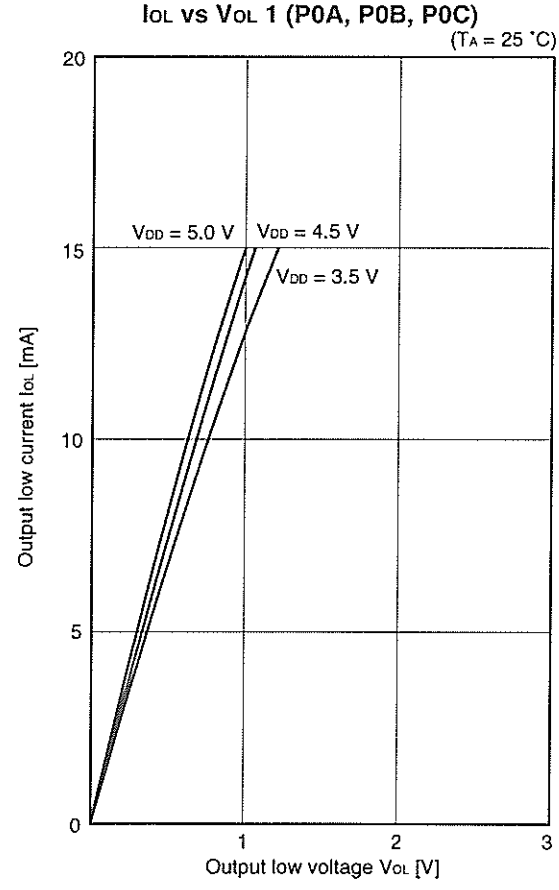
**2.** Capacitors are contained.

Example of an external circuit

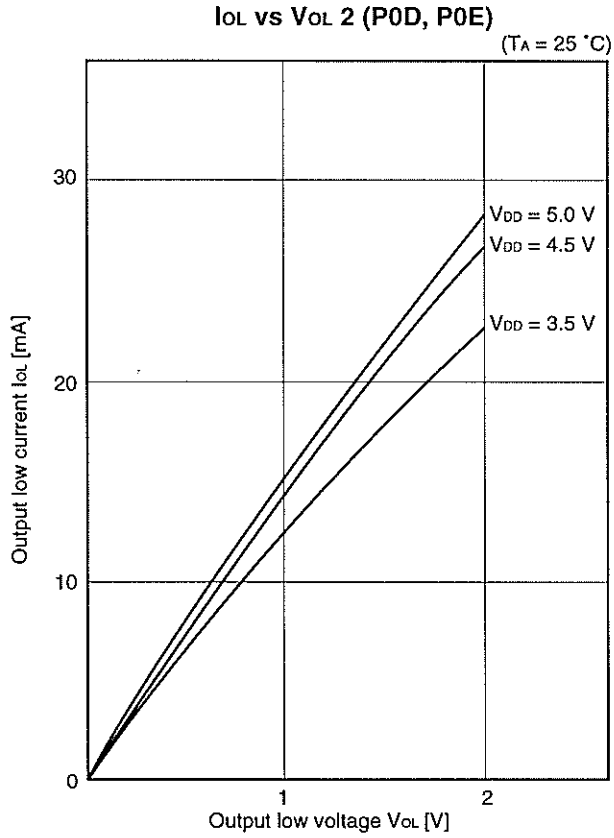


## 23. CHARACTERISTIC CURVES (FOR REFERENCE)

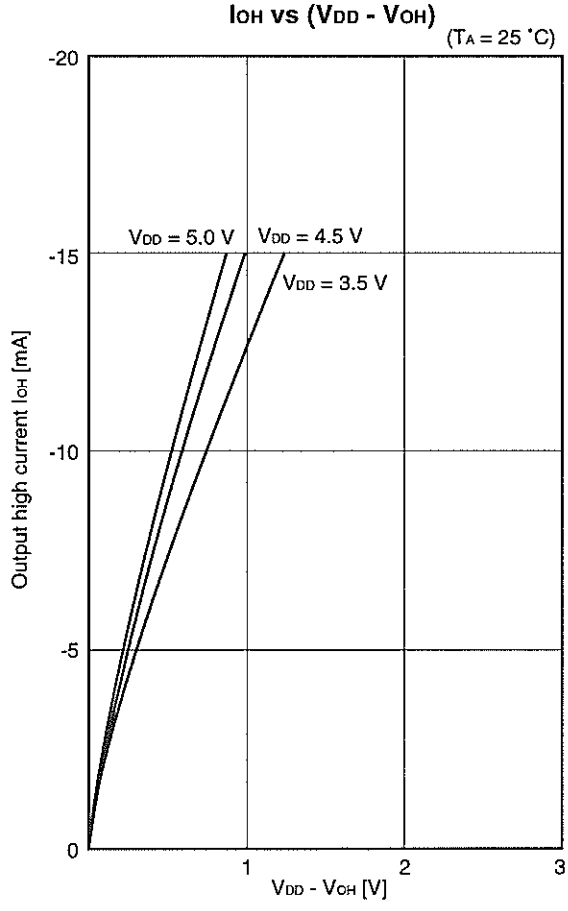



**Caution Absolute maximum rating of output current is 15 mA (peak value) per pin.**



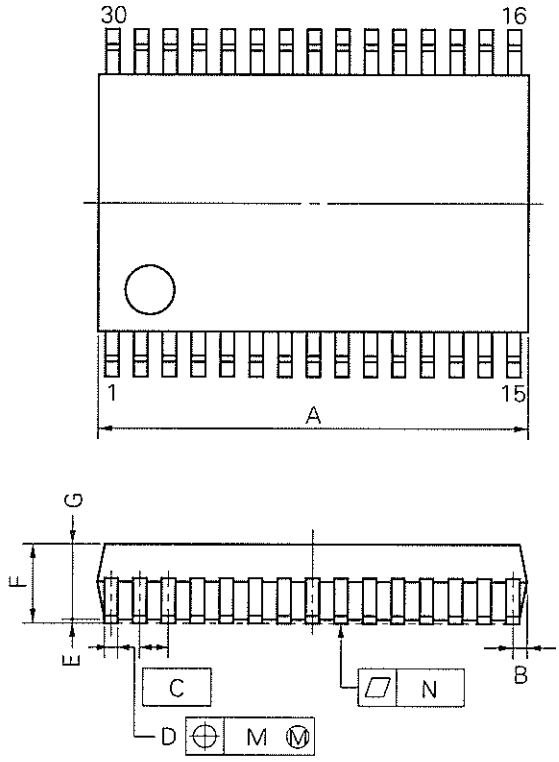

**Caution Absolute maximum rating of output current is 30 mA (peak value) per pin.**



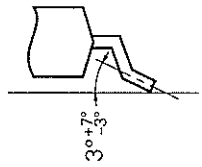

**Caution Absolute maximum rating of output current is –15 mA (peak value) per pin.**

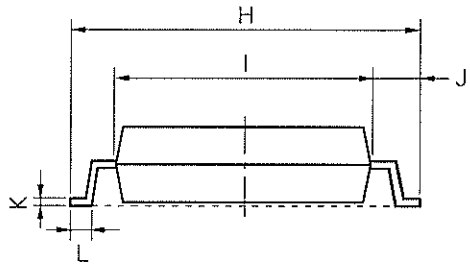
## 24. PACKAGE DRAWING

### 30 PIN PLASTIC SHRINK SOP (300 mil)




detail of lead end




P30GS-65-300B-1

| ITEM | MILLIMETERS | INCHES |
|---|---|---|
| A | 10.11 MAX. | 0.398 MAX. |
| B | 0.51 MAX. | 0.020 MAX. |
| C | 0.65 (T.P.) | 0.026 (T.P.) |
| D | $0.30^{+0.10}_{-0.05}$ | $0.012^{+0.004}_{-0.003}$ |
| E | 0.125±0.075 | 0.005±0.003 |
| F | 2.0 MAX. | 0.079 MAX. |
| G | 1.7±0.1 | 0.067±0.004 |
| H | 8.1±0.2 | 0.319±0.008 |
| I | 6.1±0.2 | 0.240±0.008 |
| J | 1.0±0.2 | $0.039^{+0.009}_{-0.008}$ |
| K | $0.15^{+0.10}_{-0.05}$ | $0.006^{+0.004}_{-0.002}$ |
| L | 0.5±0.2 | $0.020^{+0.008}_{-0.009}$ |
| M | 0.10 | 0.004 |
| N | 0.10 | 0.004 |

**NOTE**

Each lead centerline is located within 0.10 mm (0.004 inch) of its true position (T.P.) at maximum material condition.

## 25. RECOMMENDED SOLDERING CONDITIONS

The conditions listed below shall be met when soldering this product.

For details of the recommended soldering conditions, refer to our document *SMD Surface Mount Technology Manual* (IEI-1207).

Please consult with our sales offices in case any other soldering process is used, or in case soldering is done under different conditions.

**Table 25-1 Soldering Conditions for Surface-Mount Devices**

**μPD17147GS-×××-GJG: 30-pin plastic shrink SOP (300 mil)**

| Soldering process | Soldering conditions | Symbol |
|---|---|---|
| Infrared ray reflow | Peak package's surface temperature: 235 °C<br>Reflow time: 30 seconds or less (at 210 °C or more)<br>Maximum allowable number of reflow processes: 2<br>Exposure limit [Note]: 7 days (10 hours of pre-baking is required at 125 °C afterward.)<br><Cautions><br>(1) Do not start reflow-soldering the device if its temperature is higher than the room temperature because of a previous reflow soldering.<br>(2) Do not use water for flux cleaning before a second reflow soldering. | IR35-107-2 |
| VPS | Peak package's surface temperature: 215 °C<br>Reflow time: 40 seconds or less (at 200 °C or more)<br>Maximum allowable number of reflow processes: 2<br>Exposure limit [Note]: 7 days (10 hours of pre-baking is required at 125 °C afterward.)<br><Cautions><br>(1) Do not start reflow-soldering the device if its temperature is higher than the room temperature because of a previous reflow soldering.<br>(2) Do not use water for flux cleaning before a second reflow soldering. | VP15-107-2 |
| Partial heating method | Terminal temperature: 300 °C or less<br>Heat time: 3 seconds or less (for each side of device) | – |

**Note** Exposure limit before soldering after dry-pack package is opened.
Storage conditions: Temperature of 25 °C and maximum relative humidity at 65 % or less

**Caution Do not apply more than a single process at once, except for "Partial heating method."**

## APPENDIX DEVELOPMENT TOOLS

The following support tools are available for developing programs for the μPD17147GS.

### Hardware

| Name | Description |
|---|---|
| In-circuit emulator<br>[IE-17K<br>IE-17K-ET[Note 1]<br>EMU-17K[Note 2]] | The IE-17K, IE-17K-ET, and EMU-17K are in-circuit emulators applicable to the 17K series.<br>The IE-17K and IE-17K-ET are connected to the PC-9800 series (host machine) or IBM PC/AT™ through the RS-232-C interface. The EMU-17K is inserted into the extension slot of the PC-9800 series (host machine).<br>Use the system evaluation board (SE board) corresponding to each product together with one of these in-circuit emulators. *SIMPLEHOST*®, a man machine interface, implements an advanced debug environment.<br>The EMU-17K also enables user to check the contents of the data memory in real time. |
| SE board<br>(SE-17145) | The SE-17145 is an SE board for the μPD17145 sub-series. It is used solely for evaluating the system. It is also used for debugging in combination with the in-circuit emulator. |
| Emulation probe<br>(EP-17K30GT) | The EP-17K30GT is an emulation probe for the 17K series 30-pin SOP (300 mil). Use this probe together with the conversion adapter EV-9500GT-30[Note 3], to check the target system with the corresponding SE board. |
| Conversion adapter<br>(EV-9500GT-30[Note 3]) | The EV-9500GT-30 is an adapter for the 30-pin SOP (300 mil). Use this conversion adapter to connect the emulation probe, EP-17K30GT, to the target system. |
| PROM Programmer[Note 4]<br>[AF-9703, AF-9704, AF-9705<br>or AF-9706] | The AF-9703, AF-9704, AF-9705 and AF-9706 are PROM writers for the μPD17P149. Use one of these PROM writers with the program adapter, AF-9808M, to program the μPD17P149. |
| Programmer adapter[Note 4]<br>(AF-9808M) | The AF-9808M is a socket unit for the μPD17P149. It is used with the AF-9703, AF-9704, AF-9705 or AF-9706. |

**Notes 1.** Low-end model, operating on an external power supply

**2.** The EMU-17K is a product of I.C Corporation. Contact I.C Corporation (Tokyo, 03-3447-3793) for details.

**3.** An EP-17K30GT is supplied together with two EV-9500GT-30s. A set of five EV-9500GT-30s is also available.

**4.** The AF-9703, AF-9704, AF-9705, AF-9706, and AF-9808M are products of Ando Electric Co., Ltd. Contact Ando Electric Co., Ltd. (Tokyo, 03-3733-1163) for details.

**Software**

| Name | Description | Host machine | OS | | Distribution media | Part number |
|---|---|---|---|---|---|---|
| 17K series assembler (AS17K) | AS17K is an assembler applicable to the 17K series. In developing μPD17147GS program, AS17K is used in combination with a device file (AS17145). | PC-9800 series | MS-DOS™ | | 5.25-inch, 2HD | μS5A10AS17K |
| | | | | | 3.5-inch, 2HD | μS5A13AS17K |
| | | IBM PC/AT | PC DOS™ | | 5.25-inch, 2HC | μS7B10AS17K |
| | | | | | 3.5-inch, 2HC | μS7B13AS17K |
| Device file (AS17145) | AS17145 is a device file for the μPD17145, μPD17147, and μPD17149. They are used together with the assembler (AS17K) which is applicable to the 17K series. | PC-9800 series | MS-DOS | | 5.25-inch, 2HD | μS5A10AS17145 |
| | | | | | 3.5-inch, 2HD | μS5A13AS17145 |
| | | IBM PC/AT | PC DOS | | 5.25-inch, 2HC | μS7B10AS17145 |
| | | | | | 3.5-inch, 2HC | μS7B13AS17145 |
| Support software (*SIMPLEHOST*) | *SIMPLEHOST*, running on the Windows™, provides man-machine-interface in developing programs by using a personal computer and the in-circuit emulator. | PC-9800 series | MS-DOS | Windows | 5.25-inch, 2HD | μS5A10IE17K |
| | | | | | 3.5-inch, 2HD | μS5A13IE17K |
| | | IBM PC/AT | PC DOS | | 5.25-inch, 2HC | μS7B10IE17K |
| | | | | | 3.5-inch, 2HC | μS7B13IE17K |

**Remark** The following table lists the versions of the operating systems described in the above table.

| OS | Versions |
|---|---|
| MS-DOS | Ver. 3.30 to Ver. 5.00A[Note] |
| PC DOS | Ver. 3.1 to Ver. 5.0[Note] |
| Windows | Ver. 3.0 to Ver. 3.1 |

**Note** MS-DOS versions 5.00 and 5.00A and PC DOS Ver. 5.0 are provided with a task swap function. This function, however, cannot be used in these software packages.

## Cautions on CMOS Devices

① **Countermeasures against static electricity for all MOSs**

**Caution When handling MOS devices, take care so that they are not electrostatically charged.**

Strong static electricity may cause dielectric breakdown in gates. When transporting or storing MOS devices, use conductive trays, magazine cases, shock absorbers, or metal cases that NEC uses for packaging and shipping. Be sure to ground MOS devices during assembling. Do not allow MOS devices to stand on plastic plates or do not touch pins.

Also handle boards on which MOS devices are mounted in the same way.

② **CMOS-specific handling of unused input pins**

**Caution Hold CMOS devices at a fixed input level.**

Unlike bipolar or NMOS devices, if a CMOS device is operated with no input, an intermediate-level input may be caused by noise. This allows current to flow in the CMOS device, resulting in a malfunction. Use a pull-up or pull-down resistor to hold a fixed input level. Since unused pins may function as output pins at unexpected times, each unused pin should be separately connected to the $V_{DD}$ or GND pin through a resistor.

If handling of unused pins is documented, follow the instructions in the document.

③ **Statuses of all MOS devices at initialization**

**Caution The initial status of a MOS device is unpredictable when power is turned on.**

Since characteristics of a MOS device are determined by the amount of ions implanted in molecules, the initial status cannot be determined in the manufacture process. NEC has no responsibility for the output statuses of pins, input and output settings, and the contents of registers at power on. However, NEC assures operation after reset and items for mode setting if they are defined.

When you turn on a device having a reset function, be sure to reset the device first.

***SIMPLEHOST* is a registered trademark of NEC Corporation.**

**MS-DOS and Windows are trademarks of Microsoft Corporation.**

**PC/AT and PC DOS are trademarks of IBM Corporation.**

The export of this product from Japan is regulated by the Japanese government. To export this product may be prohibited without governmental license, the need for which must be judged by the customer. The export or re-export of this product from a country other than Japan may also be prohibited without a license from that country. Please call an NEC sales representative.



M4 94.11