

To our customers,

---

## Old Company Name in Catalogs and Other Documents

---

On April 1<sup>st</sup>, 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 1<sup>st</sup>, 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.

## 4-BIT SINGLE-CHIP MICROCONTROLLERS

### FOR INFRARED REMOTE CONTROL TRANSMISSION

#### DESCRIPTION

The  $\mu$ PD64A and 65 feature low-voltage 2.0 V operation, and incorporate a carrier generator for infrared remote control transmission, a standby release function through key entry, and a programmable timer, making them ideal for infrared remote control transmitters.

A one-time PROM product, the  $\mu$ PD6P5 is also available for program evaluation or small-quantity production.

#### FEATURES

- Program memory (ROM)
  - $\mu$ PD64A: 1002  $\times$  10 bits
  - $\mu$ PD65: 2026  $\times$  10 bits
- Data memory (RAM): 32  $\times$  4 bits
- On-chip carrier generator for infrared remote control
- 9-bit programmable timer: 1 channel
- Instruction execution time: 16  $\mu$ s (when operating at  $f_x = 4$  MHz: ceramic oscillation)
- Stack levels: 1 (stack RAM is also used for data memory RF)
- I/O pins ( $K_{I/O}$ ): 8
- Input pins ( $K_I$ ): 4
- Sense input pin ( $S_0, S_2$ ): 2
- $S_1/\overline{LED}$  pin (I/O): 1 (when in output mode, this is the remote control transmission display pin.)
- Power supply voltage:  $V_{DD} = 2.0$  to 3.6 V
- Operating ambient temperature:  $T_A = -40$  to  $+85^\circ\text{C}$
- Oscillation frequency:  $f_x = 2.4$  to 8 MHz
- On-chip POC circuit

#### APPLICATIONS

Infrared remote control transmitters (for AV and household electric appliances)

Unless otherwise specified, the  $\mu$ PD65 is treated as the representative model throughout this document.

The information in this document is subject to change without notice. Before using this document, please confirm that this is the latest version.

Not all devices/types available in every country. Please check with local NEC representative for availability and additional information.

**ORDERING INFORMATION**

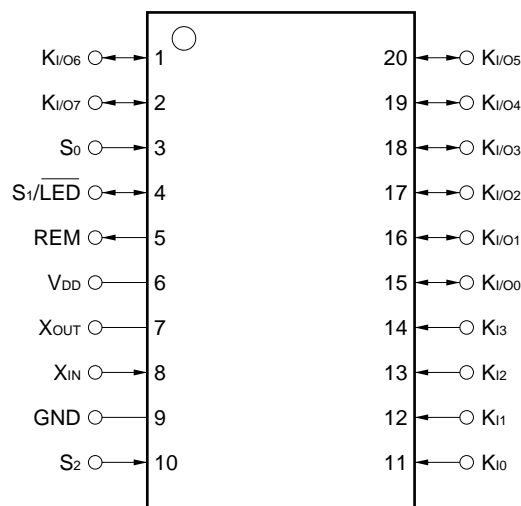
Part Number	Package
μPD64AMC-xxx-5A4	20-pin plastic SSOP (7.62 mm (300))
μPD65MC-xxx-5A4	20-pin plastic SSOP (7.62 mm (300))

**Remark** xxx indicates ROM code suffix.

**PIN CONFIGURATION (TOP VIEW)**

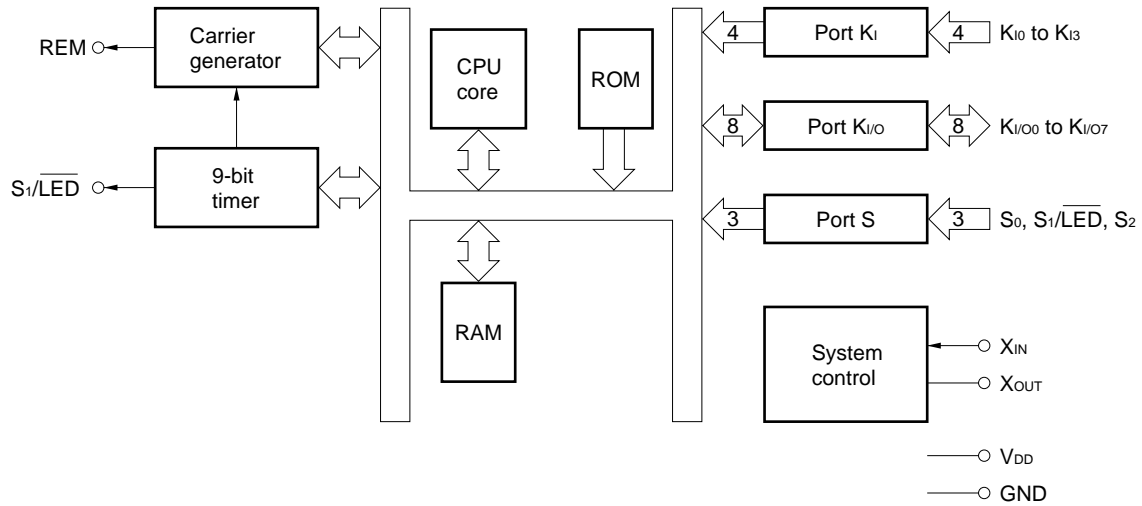
**20-Pin Plastic SSOP (7.62 mm (300))**

- μPD64AMC-xxx-5A4
- μPD65MC-xxx-5A4



**Caution** The pin numbers of K<sub>1</sub> and K<sub>1/O</sub> are in the reverse order of the μPD6600A and 6124A.

**BLOCK DIAGRAM**



**LIST OF FUNCTIONS**

Item	μPD64A	μPD65	μPD6P5
ROM capacity	1002 × 10 bits	2026 × 10 bits	
	Mask ROM		One-time PROM
RAM capacity	32 × 4 bits		
Stack	1 level (multiplexed with RF of RAM)		
I/O pins	<ul style="list-style-type: none"> <li>• Key input (K<sub>i</sub>): 4</li> <li>• Key I/O (K<sub>i/O</sub>): 8</li> <li>• Key extended input (S<sub>0</sub>, S<sub>1</sub>, S<sub>2</sub>): 3</li> <li>• Remote control transmission display output (LED): 1 (alternate function as S<sub>1</sub> pin)</li> </ul>		
Number of keys	<ul style="list-style-type: none"> <li>• 32</li> <li>• 56 (when extended by key extension input)</li> </ul>		
Clock frequency	Ceramic oscillation • f <sub>x</sub> = 2.4 to 8 MHz		Ceramic oscillation • f <sub>x</sub> = 2.4 to 4.8 MHz
Instruction execution time	16 μs (f <sub>x</sub> = 4 MHz)		
Carrier frequency	f <sub>x</sub> /8, f <sub>x</sub> /16, f <sub>x</sub> /64, f <sub>x</sub> /96, f <sub>x</sub> /128, f <sub>x</sub> /192, no carrier (high level)		
Timer	9-bit programmable timer: 1 channel		
POC circuit	On chip		
Supply voltage	V <sub>DD</sub> = 2.0 to 3.6 V		V <sub>DD</sub> = 2.2 to 3.6 V
Operating ambient temperature	T <sub>A</sub> = -40 to +85°C		
Package	20-pin plastic SSOP (7.62 mm (300))		

CONTENTS

- 1. PIN FUNCTIONS ..... 6
  - 1.1 List of Pin Functions ..... 6
  - 1.2 Pin I/O Circuits ..... 7
  - 1.3 Recommended Connection of Unused Pins ..... 8
  
- 2. INTERNAL CPU FUNCTIONS ..... 9
  - 2.1 Program Counter (PC) ..... 9
  - 2.2 Stack Pointer (SP) ..... 9
  - 2.3 Address Stack Register (ASR (RF)) ..... 9
  - 2.4 Program Memory (ROM) ..... 10
  - 2.5 Data Memory (RAM) ..... 10
  - 2.6 Data Pointer (DP) ..... 11
  - 2.7 Accumulator (A) ..... 11
  - 2.8 Arithmetic and Logic Unit (ALU) ..... 12
  - 2.9 Flags ..... 12
    - 2.9.1 Status flag (F) ..... 12
    - 2.9.2 Carry flag (CY) ..... 13
  
- 3. PORT REGISTERS (PX) ..... 14
  - 3.1 K<sub>I/O</sub> Port (P0) ..... 15
  - 3.2 K<sub>I</sub> Port/Special Port (P1) ..... 16
    - 3.2.1 K<sub>I</sub> port (P<sub>11</sub>: bits 4 to 7 of P1) ..... 16
    - 3.2.2 S<sub>0</sub> port (bit 2 of P1) ..... 16
    - 3.2.3 S<sub>1</sub>/LED port (bit 3 of P1) ..... 16
    - 3.2.4 S<sub>2</sub> port (bit 1 of P1) ..... 17
  - 3.3 Control Register 0 (P3) ..... 17
  - 3.4 Control Register 1 (P4) ..... 18
  
- 4. TIMER ..... 19
  - 4.1 Timer Configuration ..... 19
  - 4.2 Timer Operation ..... 20
  - 4.3 Carrier Output ..... 21
  - 4.4 Software Control of Timer Output ..... 21
  
- 5. STANDBY FUNCTION ..... 22
  - 5.1 Outline of Standby Function ..... 22
  - 5.2 Standby Mode Setting and Release ..... 23
  - 5.3 Standby Mode Release Timing ..... 25
  
- 6. RESET ..... 26
  
- 7. POC CIRCUIT ..... 27
  - 7.1 Functions of POC Circuit ..... 28
  - 7.2 Oscillation Check at Low Supply Voltage ..... 28
  
- 8. SYSTEM CLOCK OSCILLATOR ..... 29

9. INSTRUCTION SET ..... 30

    9.1 Machine Language Output by Assembler ..... 30

    9.2 Circuit Symbol Description ..... 31

    9.3 Mnemonic to/from Machine Language (Assembler Output) Contrast Table ..... 32

    9.4 Accumulator Operation Instructions ..... 36

    9.5 I/O Instructions ..... 39

    9.6 Data Transfer Instructions ..... 40

    9.7 Branch Instructions ..... 42

    9.8 Subroutine Instructions ..... 43

    9.9 Timer Operation Instructions ..... 44

    9.10 Other Instructions ..... 45

10. ASSEMBLER RESERVED WORDS ..... 47

    10.1 Mask Option Quasi Directives ..... 47

        10.1.1 OPTION and ENDOP quasi directives ..... 47

        10.1.2 Mask option definition quasi directive ..... 47

11. ELECTRICAL SPECIFICATIONS ..... 48

12. CHARACTERISTIC CURVES (REFERENCE VALUES) ..... 52

13. APPLICATION CIRCUIT EXAMPLE ..... 53

14. PACKAGE DRAWING ..... 54

15. RECOMMENDED SOLDERING CONDITIONS ..... 55

APPENDIX A. DEVELOPMENT TOOLS ..... 56

APPENDIX B. FUNCTIONAL COMPARISON BETWEEN μPD64A, 65 AND OTHER PRODUCTS .... 57

APPENDIX C. EXAMPLE OF REMOTE-CONTROL TRANSMISSION FORMAT ..... 58

## 1. PIN FUNCTIONS

### 1.1 List of Pin Functions

Pin No.	Symbol	Function	Output Format	After Reset
1 2 15 to 20	$K_{i/00}$ to $K_{i/07}$	8-bit I/O port. I/O can be switched in 8-bit units. In input mode, a pull-down resistor is added. In output mode, these pins can be used as key scan outputs from the key matrix.	CMOS push-pull <sup>Note 1</sup>	High-level output
3	$S_0$	Input port. Can also be used as a key return input from key matrix. In input mode, use of a pull-down resistor for the $S_0$ and $S_1$ ports can be specified by software in 2-bit units. If input mode is canceled by software, this pin is placed in the OFF mode and enters a high-impedance state.	—	High impedance (OFF mode)
4	$S_1/\overline{LED}$	I/O port. In input mode ( $S_1$ ), this pin can also be used as a key return input from key matrix. Use of a pull-down resistor for the $S_0$ and $S_1$ ports can be specified by software in 2-bit units. In output mode ( $\overline{LED}$ ), this pin becomes the remote control transmission display output (active low). When the remote control carrier is output from the $\overline{REM}$ output, this pin outputs a low level from the $\overline{LED}$ output synchronously with the $\overline{REM}$ signal.	CMOS push-pull	High-level output ( $\overline{LED}$ )
5	REM	Infrared remote control transmission output. The output is active high. Carrier frequency: $f_x/8$ , $f_x/64$ , $f_x/96$ , high-level, $f_x/16$ , $f_x/128$ , $f_x/192$ (usable on software)	CMOS push-pull	Low-level output
6	$V_{DD}$	Power supply	—	—
7 8	$X_{OUT}$ $X_{IN}$	Connecting ceramic resonators for system clock.	—	Low level (oscillation stopped)
9	GND	Ground.	—	—
10	$S_2$	Input port. The use of a STOP mode release for the $S_2$ port can be specified by software. When using this pin as a key input from a key matrix, enable the use of the STOP mode release (at this time, a pull-down resistor is connected internally.) When the STOP mode release is disabled, this pin can be used as an input port that does not release the STOP mode even if the release condition is established (at this time, a pull-down resistor is not connected internally.)	—	Input (high impedance, STOP mode release cannot be used)
11 to 14	$K_{i0}$ to $K_{i3}$ <sup>Note 2</sup>	4-bit input port. These pins can be used as key return inputs from the key matrix. Use of a pull-down resistor can be specified by software in 4-bit units.	—	Input (low level)

**Notes** 1. Be aware that the drive capability of the low-level output side is held low.

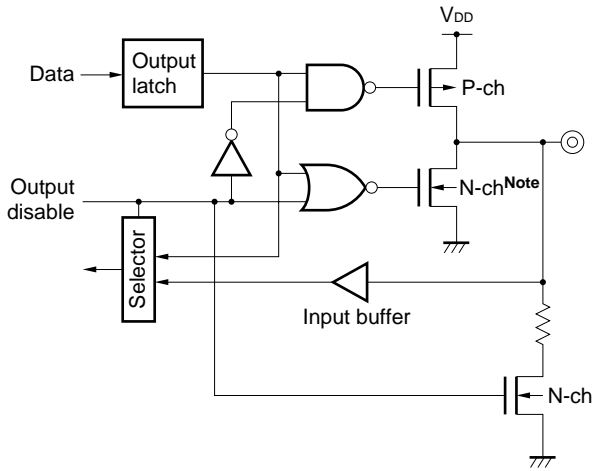
- ★ 2. In order to prevent malfunction, do not input a high level to all the  $K_{i0}$  to  $K_{i3}$  pins when POC is released due to supply voltage startup (these pins can be left open. When leaving open, keep pull-down resistors connected).



1.2 Pin I/O Circuits

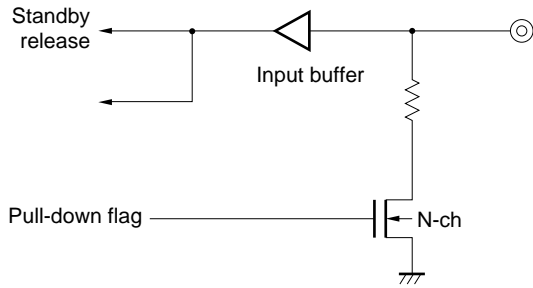
The I/O circuits of the  $\mu$ PD64A and 65 pins are shown in partially simplified forms below.

(1)  $K_{I/O0}$  to  $K_{I/O7}$

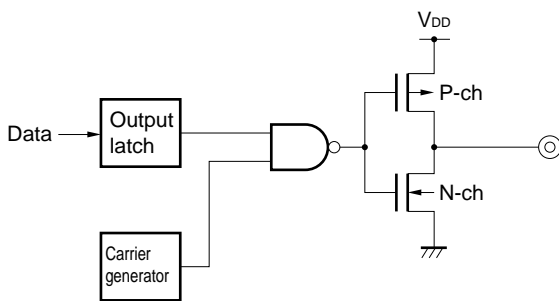


**Note** The drive capability is held low.

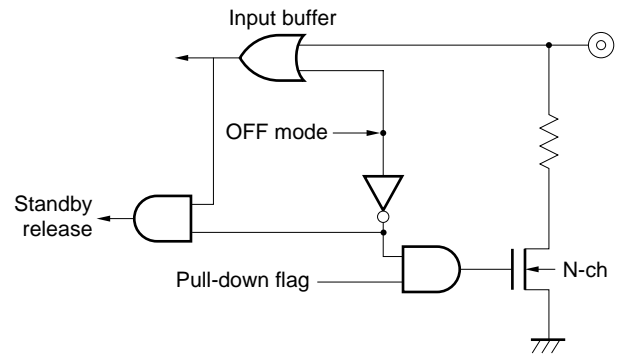
(2)  $K_{I0}$  to  $K_{I3}$



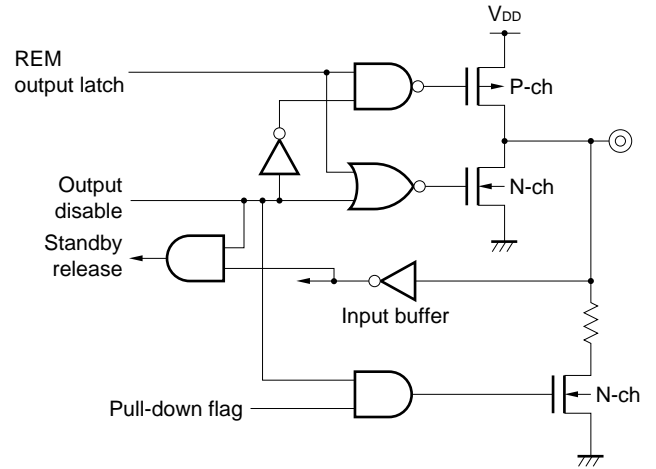
(3) REM



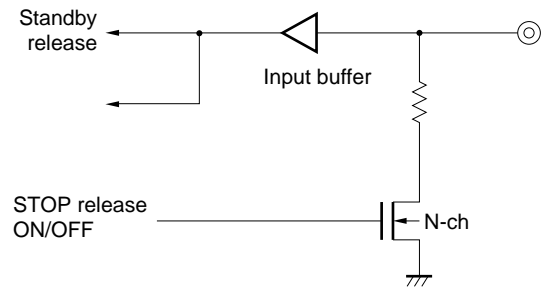
(4)  $S_0$



(5)  $S_1/\overline{LED}$



(6)  $S_2$



### 1.3 Recommended Connection of Unused Pins

The following connections are recommended for unused pins.

**Table 1-1. Recommended Connection of Unused Pins**

Pin		Connection	
		Inside Microcontroller	Outside Microcontroller
K <sub>I/O</sub>	Input mode	—	Leave open
	Output mode	High-level output	
REM		—	
S <sub>1</sub> /LED		Output mode (LED) setting	
S <sub>0</sub>		OFF mode setting	Directly connect to GND
S <sub>2</sub>		—	
K <sub>I</sub>		—	

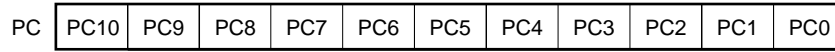
**Caution** The I/O mode and the pin output level are recommended to be fixed by setting them repeatedly in each loop of the program.

## 2. INTERNAL CPU FUNCTIONS

### 2.1 Program Counter (PC): 11 Bits

This is a binary counter that holds the address information of the program memory.

Figure 2-1. Program Counter Configuration



The program counter contains the address of the instruction that should be executed next. Normally, the counter contents are automatically incremented in accordance with the instruction length (byte count) each time an instruction is executed.

However, when executing jump instructions (JMP, JC, JNC, JF, JNF), the program counter contains the jump destination address written in the operand.

When executing the subroutine call instruction (CALL), the call destination address written in the operand is entered in the PC after the PC contents at the time are saved to the address stack register (ASR). If the return instruction (RET) is executed after the CALL instruction is executed, the address saved to the ASR is restored to the PC.

After reset, the value of the program counter becomes “000H”.

### 2.2 Stack Pointer (SP): 1 Bit

This is a 1-bit register that holds the status of the address stack register.

The stack pointer contents are incremented when the call instruction (CALL) is executed; they are decremented when the return instruction (RET) is executed.

After reset, the stack pointer contents are cleared to “0”.

When the stack pointer overflows (stack level 2 or more) or underflows, the CPU is hung up, causing a system reset signal to be generated and the PC to be cleared to “000H”.

As no instruction is available to set a value directly for the stack pointer, it is not possible to operate the pointer by means of a program.

### 2.3 Address Stack Register (ASR (RF)): 11 Bits

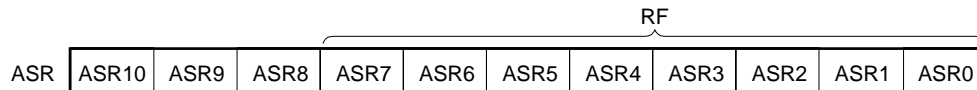
The address stack register saves the return address of the program after a subroutine call instruction is executed.

The lower 8 bits are allocated to the RF of the data memory as a dual-function RAM. The register holds the ASR value even after the RET is executed.

After reset, it holds the previous data (undefined when turning on the power).

**Caution** If the RF is accessed as the data memory, the higher 3 bits of the ASR become undefined.

Figure 2-2. Address Stack Register Configuration



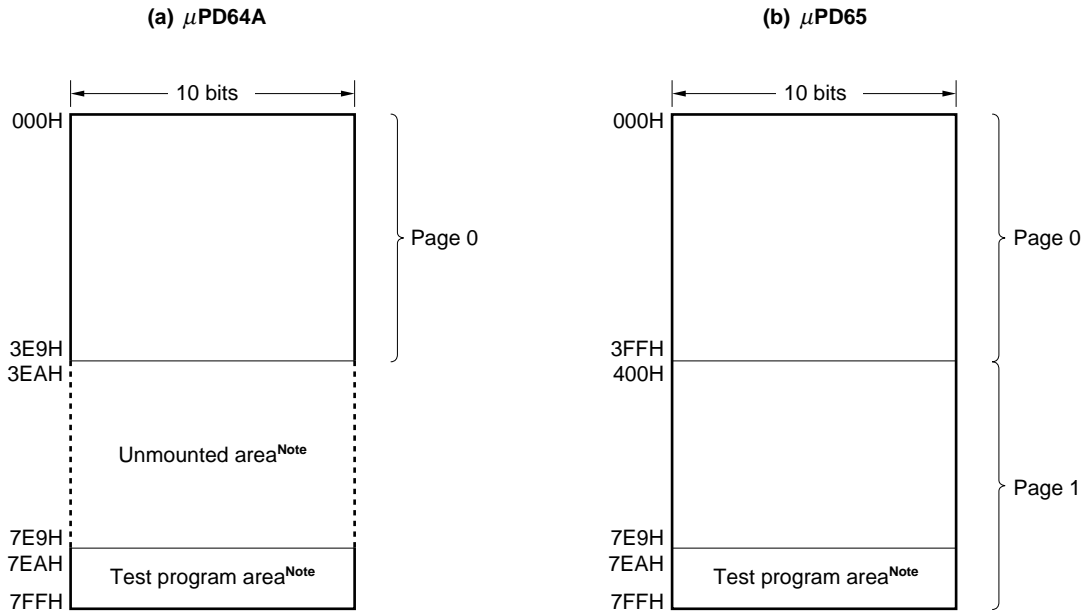
**2.4 Program Memory (ROM): 1002 steps × 10 bits (μPD64A)  
2026 steps × 10 bits (μPD65)**

The ROM consists of 10 bits per step, and is addressed by the program counter.

The program memory stores programs and table data, etc.

The 22 steps from 7EAH to 7FFH cannot be used in the test program area.

**Figure 2-3. Program Memory Map**



**Note** The unmounted area and test program area are designed so that a program or data placed in either of them by mistake is returned to the 000H address.

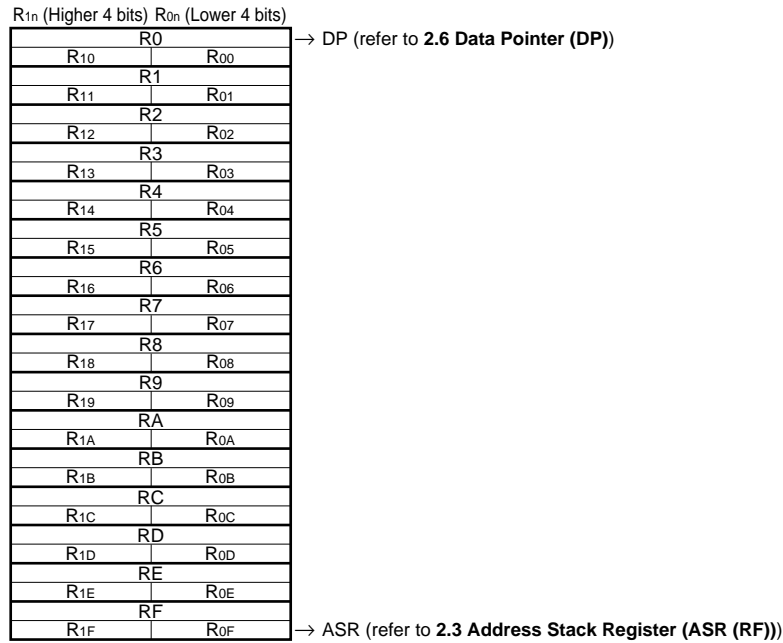
**2.5 Data Memory (RAM): 32 × 4 Bits**

The data memory, which is a static RAM consisting of 32 × 4 bits, is used to hold processed data. The data memory is sometimes processed in 8-bit units. R0 can be used as the ROM data pointer.

RF is also used as the ASR.

After reset, R0 is cleared to "00H" and R1 to RF hold the previous data (undefined when turning on the power).

Figure 2-4. Data Memory Configuration



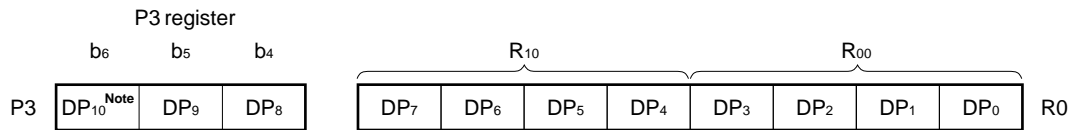
**2.6 Data Pointer (DP): 11 Bits**

The ROM data table can be referenced by setting the ROM address in the data pointer to call the ROM contents.

The lower 8 bits of the ROM address are specified by R0 of the data memory; and the higher 3 bits by bits 4, 5, and 6 of the P3 register (CR0).

After reset, the pointer contents become “000H”.

Figure 2-5. Data Pointer Configuration



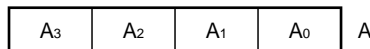
**Note** Set DP<sub>10</sub> of the μPD64A to 0.

**2.7 Accumulator (A): 4 Bits**

The accumulator, which is a register consisting of 4 bits, plays a leading role in performing various operations.

After reset, the accumulator contents are left undefined.

Figure 2-6. Accumulator Configuration



### 2.8 Arithmetic and Logic Unit (ALU): 4 Bits

The arithmetic and logic unit (ALU), which is an arithmetic circuit consisting of 4 bits, executes simple manipulations with priority given to logical operations.

### 2.9 Flags

#### 2.9.1 Status flag (F)

Pin and timer statuses can be checked by executing the STTS instruction to check the status flag.

The status flag is set (1) in the following cases.

- If the condition specified with the operand is met when the STTS instruction has been executed.
- When standby mode is released.
- When the release condition is met at the point of executing the HALT instruction. (In this case, the system does not enter standby mode.)

Conversely, the status flag is cleared (0) in the following cases.

- If the condition specified with the operand is not met when the STTS instruction has been executed.
- While the status flag is set (1), the HALT instruction is executed, but the release condition is not met at the point of executing the HALT instruction. (In this case, the system does not enter standby mode.)

**Table 2-1. Conditions for Status Flag (F) to Be Set by STTS Instruction**

Operand Value of STTS Instruction				Condition for Status Flag (F) to Be Set
b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>	
0	0	0	0	High level is input to at least one of the K <sub>i</sub> pins.
	0	1	1	High level is input to at least one of the K <sub>i</sub> pins.
	1	1	0	High level is input to at least one of the K <sub>i</sub> pins.
	1	0	1	The down counter of the timer is 0.
1	Any combination of b <sub>2</sub> , b <sub>1</sub> , and b <sub>0</sub> above			[The following condition is added in addition to the above.] High level is input to at least one of the S <sub>0</sub> <sup>Note 1</sup> , S <sub>1</sub> <sup>Note 1</sup> , and S <sub>2</sub> <sup>Note 2</sup> pins.

- ★ **Notes**
1. The S<sub>0</sub> and S<sub>1</sub> pins must be set to input mode (bit 2 of the P4 register is set to 0 and bit 0 to 1).
  2. The use of STOP mode release for the S<sub>2</sub> pin must be enabled (bit 3 of the P4 register is set to 1).

### 2.9.2 Carry flag (CY)

The carry flag is set (1) in the following cases.

- If the ANL instruction or the XRL instruction is executed when bit 3 of the accumulator is “1” and bit 3 of the operand is “1”.
- If the RL instruction or the RLZ instruction is executed when bit 3 of the accumulator is “1”.
- If the INC instruction or the SCAF instruction is executed when the value of the accumulator is 0FH.

The carry flag is cleared (0) in the following cases.

- If the ANL instruction or the XRL instruction is executed when either bit 3 of the accumulator or bit 3 of the operand (or both) is “0”.
- If the RL instruction or the RLZ instruction is executed when bit 3 of the accumulator is “0”.
- If the INC instruction or the SCAF instruction is executed when the value of the accumulator is other than 0FH.
- If the ORL instruction is executed.
- When data is written to the accumulator by the MOV instruction or the IN instruction.

### 3. PORT REGISTERS (PX)

The K<sub>I/O</sub> port, the K<sub>I</sub> port, the special ports (S<sub>0</sub>, S<sub>1</sub>/ $\overline{\text{LED}}$ , S<sub>2</sub>), and the control registers are treated as port registers. Port register values after reset are shown below.

Figure 3-1. Port Register Configuration

Port register								After reset
P0								FFH
P <sub>10</sub>				P <sub>00</sub>				
K <sub>I/O7</sub>	K <sub>I/O6</sub>	K <sub>I/O5</sub>	K <sub>I/O4</sub>	K <sub>I/O3</sub>	K <sub>I/O2</sub>	K <sub>I/O1</sub>	K <sub>I/O0</sub>	
P1								×FH <sup>Note</sup>
P <sub>11</sub>				P <sub>01</sub>				
K <sub>I3</sub>	K <sub>I2</sub>	K <sub>I1</sub>	K <sub>I0</sub>	S <sub>1</sub> / $\overline{\text{LED}}$	S <sub>0</sub>	S <sub>2</sub>	1	
P3 (control register 0)								03H
P <sub>13</sub>				P <sub>03</sub>				
0	DP <sub>10</sub>	DP <sub>9</sub>	DP <sub>8</sub>	TCTL	CARY	MOD <sub>1</sub>	MOD <sub>0</sub>	
P4 (control register 1)								26H
P <sub>14</sub>				P <sub>04</sub>				
0	0	K <sub>I</sub> pull-down	S <sub>0</sub> /S <sub>1</sub> pull-down	S <sub>2</sub> STOP release	S <sub>1</sub> / $\overline{\text{LED}}$ mode	K <sub>I/O</sub> mode	S <sub>0</sub> mode	

**Note** ×: Value based on the K<sub>I</sub> pin state

Table 3-1. Relationship Between Ports and Read/Write

Port Name	Input Mode		Output Mode	
	Read	Write	Read	Write
K <sub>I/O</sub>	Pin state	Output latch	Output latch	Output latch
K <sub>I</sub>	Pin state	—	—	—
S <sub>0</sub>	Pin state	—	<b>Note</b>	—
S <sub>1</sub> / $\overline{\text{LED}}$	Pin state	—	Pin state	—
S <sub>2</sub>	Pin state	—	—	—

**Note** When in OFF mode, “1” is always read.



### 3.1 K<sub>I/O</sub> Port (P0)

The K<sub>I/O</sub> port is an 8-bit I/O port for key scan output.

Input/output mode is set by bit 1 of the P4 register.

If a read instruction is executed, the pin state can be read in input mode, whereas the output latch contents can be read in output mode.

If a write instruction is executed, data can be written to the output latch regardless of input or output mode.

After reset, the port enters output mode, and the value of the output latch (P0) becomes 1111 1111B.

The K<sub>I/O</sub> port includes a pull-down resistor, which functions in input mode only.

**Caution** Because during double pressing of a key, a high-level output and a low-level output may conflict with each other at the K<sub>I/O</sub> port, the low-level output current of the K<sub>I/O</sub> port is held low. Therefore, be careful when using the K<sub>I/O</sub> port for purposes other than key scan output.

The K<sub>I/O</sub> port is designed so that, even when connected directly to V<sub>DD</sub> within the normal supply voltage range (V<sub>DD</sub> = 2.0 to 3.6 V), no problem may occur.

Table 3-2. K<sub>I/O</sub> Port (P0)

Bit	b <sub>7</sub>	b <sub>6</sub>	b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>
Name	K <sub>I/O7</sub>	K <sub>I/O6</sub>	K <sub>I/O5</sub>	K <sub>I/O4</sub>	K <sub>I/O3</sub>	K <sub>I/O2</sub>	K <sub>I/O1</sub>	K <sub>I/O0</sub>

b<sub>0</sub> to b<sub>7</sub>: When reading: In input mode, state of the K<sub>I/O</sub> pin is read.

In output mode, the K<sub>I/O</sub> pin's output latch contents are read.

When writing: Data is written to the K<sub>I/O</sub> pin's output latch regardless of input or output mode.

### 3.2 K<sub>I</sub> Port/Special Port (P1)

#### 3.2.1 K<sub>I</sub> port (P<sub>11</sub>: bits 4 to 7 of P1)

The K<sub>I</sub> port is a 4-bit input port for key entry.

The pin state can be read.

Use of a pull-down resistor for the K<sub>I</sub> port can be specified in 4-bit units by software using bit 5 of the P4 register.

After reset, a pull-down resistor is connected.

**Table 3-3. K<sub>I</sub>/Special Port Register (P1)**

Bit	b <sub>7</sub>	b <sub>6</sub>	b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>
Name	K <sub>13</sub>	K <sub>12</sub>	K <sub>11</sub>	K <sub>10</sub>	S <sub>1</sub> / $\overline{\text{LED}}$	S <sub>0</sub>	S <sub>2</sub>	Fixed to "1"

b<sub>1</sub>: The state of the S<sub>2</sub> pin is read (read only).

b<sub>2</sub>: In input mode, the state of the S<sub>0</sub> pin is read (read only).

In OFF mode, this bit is fixed to 1.

b<sub>3</sub>: The state of the S<sub>1</sub>/ $\overline{\text{LED}}$  pin is read regardless of input/output mode (read only).

b<sub>4</sub> to b<sub>7</sub>: The state of the K<sub>I</sub> pin is read (read only).

★ **Caution** In order to prevent malfunction, do not input a high level to all the K<sub>10</sub> to K<sub>13</sub> pins when POC is released due to supply voltage startup (these pins can be left open. When leaving open, keep pull-down resistors connected).

#### 3.2.2 S<sub>0</sub> port (bit 2 of P1)

The S<sub>0</sub> port is an input/OFF mode port.

The pin state can be read by setting this port to input mode with bit 0 of the P4 register.

In input mode, use of a pull-down resistor for the S<sub>0</sub> and S<sub>1</sub>/ $\overline{\text{LED}}$  ports can be specified in 2-bit units by software using bit 4 of the P4 register.

If input mode is released (thus set to OFF mode), the pin becomes high impedance but through current does not flow internally. In OFF mode, "1" can be read regardless of the pin state.

After reset, this port is set to OFF mode, thus becoming high impedance.

#### 3.2.3 S<sub>1</sub>/ $\overline{\text{LED}}$ port (bit 3 of P1)

The S<sub>1</sub>/ $\overline{\text{LED}}$  port is an I/O port.

Bit 2 of the P4 register can be used to set input or output mode. The pin state can be read in both input mode and output mode.

In input mode, use of a pull-down resistor for the S<sub>0</sub> and S<sub>1</sub>/ $\overline{\text{LED}}$  ports can be specified in 2-bit units by software using bit 4 of the P4 register.

In output mode, the pull-down resistor is automatically disconnected, and this pin becomes the remote control transmission display pin (refer to 4. TIMER).

After reset, this port enters output mode, and a high level is output.

**3.2.4 S<sub>2</sub> port (bit 1 of P1)**

The S<sub>2</sub> port is an input port.

Use of a STOP mode release for the S<sub>2</sub> port can be specified by bit 3 of the P4 register.

When using this port as a key input from a key matrix, enable the use of the STOP mode release (bit 3 of P4 register is set to 1) (at this time, a pull-down resistor is connected internally). When the STOP mode release is disabled (bit 3 of P4 register is set to 0), this port can be used as an input port that does not release the STOP mode even if the release condition is established (at this time, a pull-down resistor is not connected internally).

The state of the pin can be read in both cases.

After reset, the pin is set to input mode in which the STOP mode release is disabled, and enters a high-impedance state.

**3.3 Control Register 0 (P3)**

Control register 0 consists of 8 bits. The contents that can be controlled are as shown below.

After reset, the register becomes 0000 0011B.

**Table 3-4. Control Register 0 (P3)**

Bit	b <sub>7</sub>	b <sub>6</sub>	b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>
Name	—	DP (Data Pointer)			TCTL	CARY	MOD <sub>1</sub>	MOD <sub>0</sub>
		DP <sub>10</sub> Note	DP <sub>9</sub>	DP <sub>8</sub>				
Set value	0	Fixed to "0"	0	0	0	1/1	ON	Refer to <b>Table 3-5</b> .
	1		1	1	1	1/2	OFF	
After reset	0	0	0	0	0	0	1	1

b<sub>0</sub>, b<sub>1</sub>: These bits specify the carrier frequency and duty ratio of the REM output.

b<sub>2</sub>: This bit specifies the availability of the carrier of the frequency specified by b<sub>0</sub> and b<sub>1</sub>.  
 "0" = ON (with carrier); "1" = OFF (without carrier; high level)

b<sub>3</sub>: This bit changes the carrier frequency and the timer clock's frequency division ratio.  
 "0" = 1/1 (carrier frequency: The specified value of b<sub>0</sub> and b<sub>1</sub>; timer clock: f<sub>x</sub>/64)  
 "1" = 1/2 (carrier frequency: Half of the specified value of b<sub>0</sub> and b<sub>1</sub>; timer clock: f<sub>x</sub>/128)

**Table 3-5. Timer Clock and Carrier Frequency Settings**

b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>	Timer Clock	Carrier Frequency (Duty Ratio)
0	0	0	0	f <sub>x</sub> /64	f <sub>x</sub> /8 (Duty 1/2)
		0	1		f <sub>x</sub> /64 (Duty 1/2)
		1	0		f <sub>x</sub> /96 (Duty 1/2)
		1	1		f <sub>x</sub> /96 (Duty 1/3)
	1	×	×	Without carrier (high level)	
1	0	0	0	f <sub>x</sub> /128	f <sub>x</sub> /16 (Duty 1/2)
		0	1		f <sub>x</sub> /128 (Duty 1/2)
		1	0		f <sub>x</sub> /192 (Duty 1/2)
		1	1		f <sub>x</sub> /192 (Duty 1/3)
	1	×	×	Without carrier (high level)	

b<sub>4</sub>, b<sub>5</sub>, b<sub>6</sub>: These bits specify the higher 3 bits (DP<sub>8</sub>, DP<sub>9</sub> and DP<sub>10</sub>) of ROM's data pointer.

**Note** Set DP<sub>10</sub> of the μPD64A to 0.

**Remark** ×: don't care

### 3.4 Control Register 1 (P4)

Control register 1 consists of 8 bits. The contents that can be controlled are as shown below. After reset, the register becomes 0010 0110B.

**Table 3-6. Control Register 1 (P4)**

Bit	b <sub>7</sub>	b <sub>6</sub>	b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>
Name	—	—	K <sub>1</sub> Pull-down	S <sub>0</sub> /S <sub>1</sub> Pull-down	S <sub>2</sub> STOP Release	S <sub>1</sub> / $\overline{\text{LED}}$ Mode	K <sub>I/O</sub> Mode	S <sub>0</sub> Mode
Set value	0	Fixed to "0"	Fixed to "0"	OFF	OFF	Disabled	S <sub>1</sub>	IN
	1			ON	ON	Enabled	$\overline{\text{LED}}$	OUT
After reset	0	0	1	0	0	1	1	0

- b<sub>0</sub>: Specifies the input mode of the S<sub>0</sub> port. "0" = OFF mode (high impedance); "1" = IN (input mode).
- b<sub>1</sub>: Specifies the I/O mode of the K<sub>I/O</sub> port. "0" = IN (input mode); "1" = OUT (output mode).
- b<sub>2</sub>: Specifies the I/O mode of the S<sub>1</sub>/ $\overline{\text{LED}}$  port. "0" = S<sub>1</sub> (input mode); "1" =  $\overline{\text{LED}}$  (output mode).
- b<sub>3</sub>: Specifies the use of the STOP mode release for the S<sub>2</sub> port (with/without pull-down resistor). "0" = Disabled (pull-down unavailable); "1" = Enabled (pull-down available).
- b<sub>4</sub>: Specifies the availability of the pull-down resistor for the S<sub>0</sub>/S<sub>1</sub> port input mode. "0" = OFF (unavailable); "1" = ON (available).
- b<sub>5</sub>: Specifies the availability of the pull-down resistor for the K<sub>1</sub> port. "0" = OFF (unavailable); "1" = ON (available).

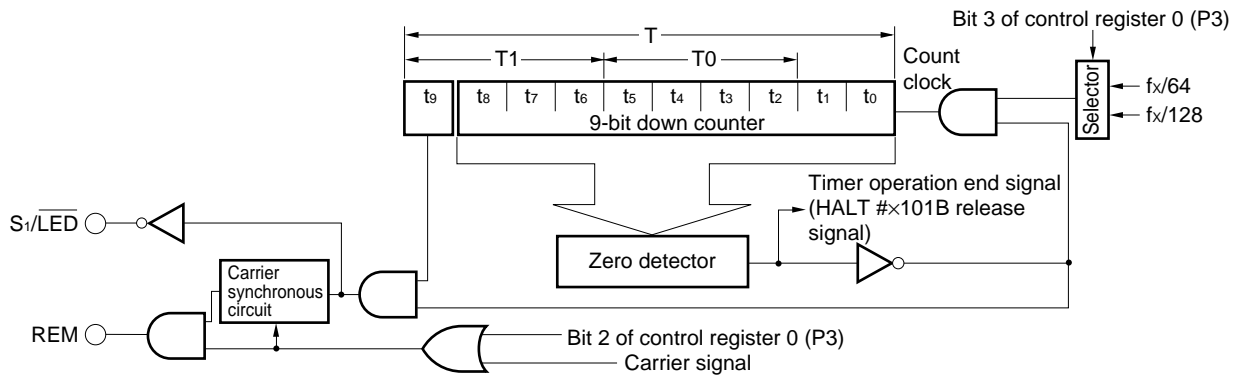
**Remark** In output mode or in OFF mode, all the pull-down resistors are automatically disconnected.

4. TIMER

4.1 Timer Configuration

The timer is the block used for creating a remote control transmission pattern. As shown in Figure 4-1, it consists of a 9-bit down counter ( $t_8$  to  $t_0$ ), a flag ( $t_9$ ) to enable the 1-bit timer output, and a zero detector.

Figure 4-1. Timer Configuration



### 4.2 Timer Operation

The timer starts (counting down) when a value other than 0 is set for the down counter with a timer operation instruction. The timer operation instructions for making the timer start operation are shown below.

```
MOV T0, A
MOV T1, A
MOV T, #data10
MOV T, @R0
```

The down counter is decremented (-1) in the cycle of 64/fx or 128/fx<sup>Note</sup>. If the value of the down counter becomes 0, the zero detector generates the timer operation end signal to stop the timer operation. At this time, if the timer is in HALT mode (HALT #x101B) waiting for the timer to stop operation, the HALT mode is released and the instruction following the HALT instruction is executed. The output of the timer operation end signal is continued while the down counter is 0 and the timer is stopped. The following expression indicates the relationship between the timer's time and the down counter's set value.

$$\text{Timer time} = (\text{Set value} + 1) \times 64/\text{fx} \text{ (or } 128/\text{fx}^{\text{Note}})$$

**Note** This becomes 128/fx if bit 3 of control register 0 is set (1).

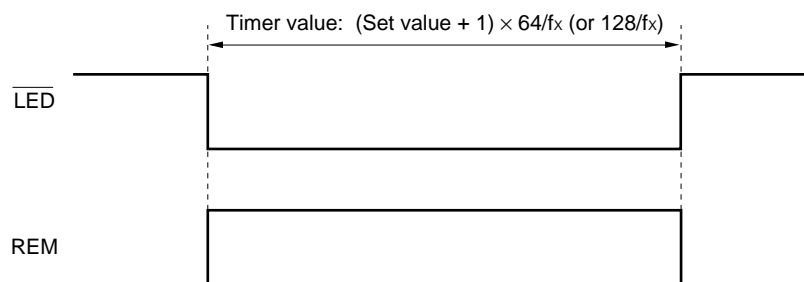
By setting the flag (t<sub>9</sub>) that enables the timer output to 1, the timer can output its operation status from the S<sub>1</sub>/LED and REM pins. The REM pin can also output the carrier while the timer is in operation.

**Table 4-1. Timer Output (When t<sub>9</sub> = 1)**

	S <sub>1</sub> /LED Pin	REM Pin
Timer operating	L	H (or carrier output <sup>Note</sup> )
Timer halted	H	L

**Note** The carrier is output if bit 2 of control register 0 is cleared (0).

**Figure 4-2. Timer Output (When Carrier Is Not Output)**

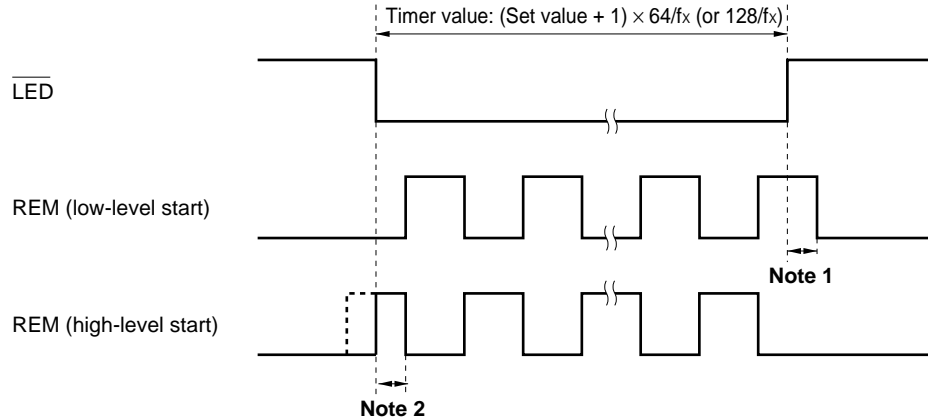


### 4.3 Carrier Output

The carrier for remote-controlled transmission can be output from the REM pin by clearing (0) bit 2 of control register 0.

As shown in Figure 4-3, in the case where the timer stops when the carrier is at a high level, the carrier continues to be output until its next fall and then stops due to the function of the carrier synchronous circuit. When the timer starts operation, however, the high-level width of the first carrier may become shorter than the specified width.

Figure 4-3. Timer Output (When Carrier Is Output)



- Notes**
1. Error when the REM output ends: Lead by “carrier low-level width” to lag by “carrier high-level width”
  2. Error of carrier high-level width: 0 to “carrier high-level width”

### 4.4 Software Control of Timer Output

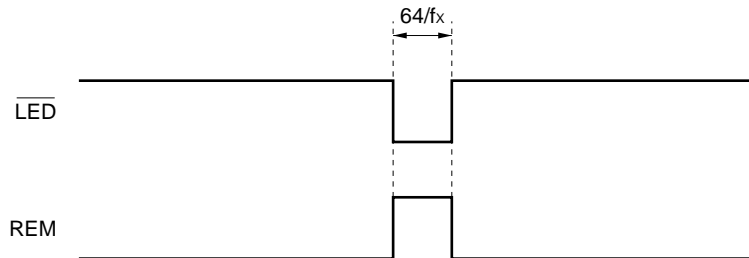
The timer output can be controlled by software. As shown in Figure 4-4, a pulse with a minimum width of 1 instruction cycle ( $64/f_x$ ) can be output.

Figure 4-4. Pulse Output of 1 Instruction Cycle Width

```

:
MOV T, #000000000B; low-level output from the REM pin
:
MOV T, #100000000B; high-level output from the REM pin
MOV T, #000000000B; low-level output from the REM pin
:

```



## 5. STANDBY FUNCTION

### 5.1 Outline of Standby Function

To reduce current consumption, two types of standby modes, i.e., HALT mode and STOP mode, are available.

In STOP mode, the system clock stops oscillation. At this time, the X<sub>IN</sub> and X<sub>OUT</sub> pins are fixed at a low level.

In HALT mode, CPU operation halts, while the system clock continues oscillation. When in HALT mode, the timer (including REM output and  $\overline{\text{LED}}$  output) operates.

In either STOP mode or HALT mode, the statuses of the data memory, accumulator, and port register, etc. immediately before the standby mode is set are retained. Therefore, make sure to set the port status for the system so that the current consumption of the whole system is suppressed before the standby mode is set.

**Table 5-1. Statuses in Standby Mode**

		STOP Mode	HALT Mode	
Setting instruction		HALT instruction		
Clock oscillator		Oscillation stopped	Oscillation continued	
Operation statuses	CPU	• Operation halted		
	Data memory	• Immediately preceding status retained		
	Accumulator	• Immediately preceding status retained		
	Flag	F	• 0 (when 1, the flag does not enter the standby mode.)	
		CY	• Immediately preceding status retained	
	Port register	• Immediately preceding status retained		
Timer	• Operation halted (The count value is reset to "0")	• Operable		

- Cautions**
1. Write the NOP instruction as the first instruction after STOP mode is released.
  2. When standby mode is released, the status flag (F) is set (1).
  3. If, at the point the standby mode has been set, its release condition is met, then the system does not enter the standby mode. However, the status flag (F) is set (1).



### 5.2 Standby Mode Setting and Release

The standby mode is set with the HALT #b3b2b1b0B instruction for both STOP mode and HALT mode. For the standby mode to be set, the status flag (F) is required to have been cleared (0).

The standby mode is released by the release condition specified with the reset (POC) or the operand of the HALT instruction. If the standby mode is released, the status flag (F) is set (1).

Even when the HALT instruction is executed in a state in which the status flag (F) is set (1), the standby mode is not set. If the release condition is not met at this time, the status flag is cleared (0). If the release condition is met, the status flag remains set (1).

Even in the case when the release condition has already been met at the point that the HALT instruction is executed, the standby mode is not set. Here, also, the status flag (F) is set (1).

**Caution** Note that depending on the status of the status flag (F), the HALT instruction may not be executed. For example, when setting HALT mode after checking the key status with the STTS instruction, the system does not enter HALT mode as long as the status flag (F) remains set (1), sometimes resulting in an unintended operation. In this case, the intended operation can be realized by executing the STTS instruction immediately after setting the timer to clear (0) the status flag.

```

Example  STTS    #03H    ;To check the Ki pin status
           :
           MOV     T, #0xxH ;To set the timer
           STTS    #05H    ;To clear the status flag
           : (During this time, be sure not to execute an instruction that may set the status flag.)
           HALT   #05H    ;To set HALT mode
    
```

**Table 5-2. Addresses Executed After Standby Mode Release**

Release Condition	Address Executed After Release
Reset	0 address
Release condition shown in <b>Table 5-3</b>	The address following the HALT instruction

Table 5-3. Standby Mode Setting (HALT #b<sub>3</sub>b<sub>2</sub>b<sub>1</sub>b<sub>0</sub>B) and Release Conditions

Operand Value of HALT Instruction				Setting Mode	Precondition for Setting	Release Conditions
b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>			
0	0	0	0	STOP	All K <sub>I/O</sub> pins are high-level output.	High level is input to at least one of K <sub>I</sub> pins.
	0	1	1	STOP	All K <sub>I/O</sub> pins are high-level output.	High level is input to at least one of K <sub>I</sub> pins.
	1	1	0	STOP <sup>Note 1</sup>	The K <sub>I/O0</sub> pin is high-level output.	High level is input to at least one of K <sub>I</sub> pins.
1	Any combination of b <sub>2</sub> b <sub>1</sub> b <sub>0</sub> above			STOP	[The following conditions are added in addition to the above.]	
					—	High level is input to at least one of S <sub>0</sub> , S <sub>1</sub> and S <sub>2</sub> pins <sup>Note 2</sup> .
0/1	1	0	1	HALT	—	When the timer's down counter is 0

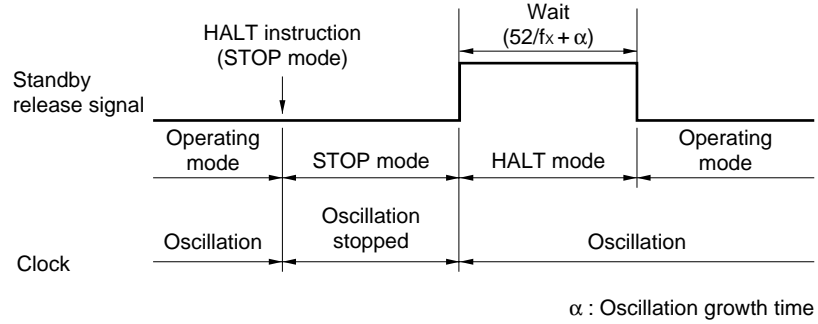
- Notes**
1. When setting HALT #×110B, configure a key matrix by using the K<sub>I/O0</sub> pin and the K<sub>I</sub> pin so that an internal reset takes effect at the time of program runaway.
  2. At least one of the S<sub>0</sub>, S<sub>1</sub> and S<sub>2</sub> pins (the pin used for releasing the standby) must be specified as follows.
    - S<sub>0</sub>, S<sub>1</sub> pins: Input mode (specified by bits 0 and 2 of the P4 register)
    - S<sub>2</sub> pin: Use of STOP mode release enabled (specified by bit 3 of the P4 register)

- Cautions**
1. The internal reset takes effect when the HALT instruction is executed with an operand value other than that above or when the precondition has not been satisfied when executing the HALT instruction.
  2. If STOP mode is set when the timer's down counter is not 0 (timer operating), the system enters STOP mode only after all the 10 bits of the timer's down counter and the timer output enable flag are cleared to 0.
  3. Write the NOP instruction as the first instruction after STOP mode is released.

### 5.3 Standby Mode Release Timing

#### (1) STOP mode release timing

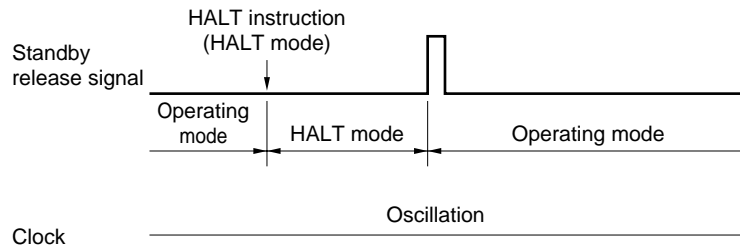
Figure 5-1. STOP Mode Release by Release Condition



**Caution** When a release condition is established in the STOP mode, the device is released from the STOP mode, and goes into a wait state. At this time, if the release condition is not held, the device goes into STOP mode again after the wait time has elapsed. Therefore, when releasing the STOP mode, it is necessary to hold the release condition longer than the wait time.

#### (2) HALT mode release timing

Figure 5-2. HALT Mode Release by Release Condition



6. RESET

The system is reset by the following occurrences.

- When the POC circuit has detected low-power voltage
- When the operand value is illegal or does not satisfy the precondition when the HALT instruction is executed
- When the accumulator is 0H when the RLZ instruction is executed
- When the stack pointer overflows or underflows

Table 6-1. Hardware Statuses After Reset

Hardware		• Reset by Internal POC Circuit in Operation • Reset by Other Factor <sup>Note 1</sup>	• Reset by Internal POC Circuit in Standby Mode
PC (11 bits)		000H	
SP (1 bit)		0B	
Data memory	R0 = DP	000H	
	R1 to RF	Undefined	Previous status retained
Accumulator (A)		Undefined	
Status flag (F)		0B	
Carry flag (CY)		0B	
Timer (10 bits)		000H	
Port register	P0	FFH	
	P1	xxxx 11x1B <sup>Note 2</sup>	
Control register	P3	03H	
	P4	26H	

**Notes 1.** The following resets are available.

- Reset when executing the HALT instruction (when the operand value is illegal or does not satisfy the precondition)
- Reset when executing the RLZ instruction (when A = 0)
- Reset by stack pointer overflow or underflow

**2.** Value according to the K<sub>1</sub> or S<sub>2</sub> pin status.



In order to prevent malfunction, do not input a high level to all the K<sub>10</sub> to K<sub>13</sub> pins when POC is released due to supply voltage startup (these pins can be left open. When leaving open, keep pull-down resistors connected).

## 7. POC CIRCUIT

The POC circuit monitors the power supply voltage and applies an internal reset in the microcontroller when the battery is replaced.

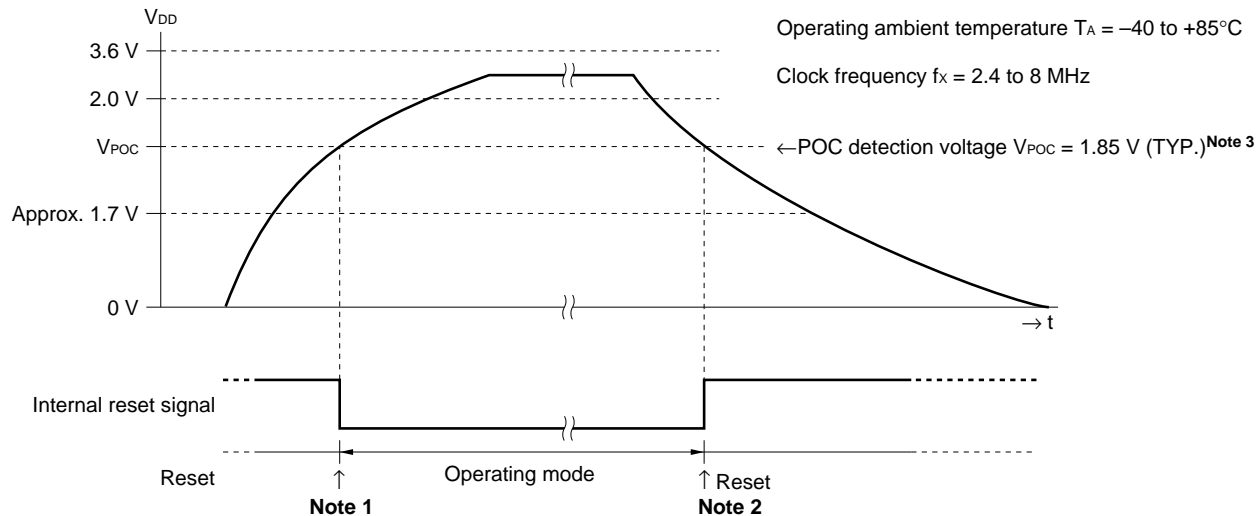
- Cautions**
1. There are cases in which the POC circuit cannot detect a low power supply voltage of less than 1 ms. Therefore, if the power supply voltage has become low for a period of less than 1 ms, the POC circuit may malfunction because it does not generate an internal reset signal.
  2. Clock oscillation is stopped by the resonator due to low power supply voltage before the POC circuit generates the internal reset signal. In this case, malfunction may result, for example when the power supply voltage is recovered after the oscillation is stopped. This type of phenomenon takes place because the POC circuit does not generate an internal reset signal (because the power supply voltage recovers before the low power supply voltage is detected) even though the clock has stopped. If, by any chance, a malfunction has taken place, remove the battery for a short time and put it back. In most cases, normal operation will be resumed.
  - ★ 3. In order to prevent malfunction, do not input a high level to all the K<sub>10</sub> to K<sub>13</sub> pins when POC is released due to supply voltage startup (these pins can be left open. When leaving open, keep pull-down resistors connected).

### 7.1 Functions of POC Circuit

The POC circuit has the following functions.

- Generates an internal reset signal when  $V_{DD} \leq V_{POC}$ .
- Cancels an internal reset signal when  $V_{DD} > V_{POC}$ .

Here,  $V_{DD}$ : Power supply voltage,  $V_{POC}$ : POC detection voltage.



- ★ **Notes**
  1. In reality, an oscillation stabilization wait time must elapse before the circuit is switched to operating mode. The oscillation stabilization wait time is about  $246/f_x$  to  $694/f_x$  (about 70 to 190  $\mu\text{s}$ , at  $f_x = 3.64$  MHz).
  2. For the POC circuit to generate an internal reset signal when the power supply voltage has fallen, it is necessary for the power supply voltage to be kept less than the  $V_{POC}$  for a period of 1 ms or more. Therefore, in reality, there is a time lag of up to 1 ms until the reset takes effect.
  3. The POC detection voltage ( $V_{POC}$ ) varies between approximately 1.7 to 2.0 V; thus, the reset may be canceled at a power supply voltage smaller than the assured range ( $V_{DD} = 2.0$  to  $3.6$  V). However, as long as the conditions for operating the POC circuit are met, the actual lowest operating power supply voltage is lower than the POC detection voltage. Therefore, no malfunction occurs due to the shortage of power supply voltage. However, malfunction for such reasons as the clock not oscillating due to low power supply voltage may occur (refer to **Cautions 3.** in **7. POC CIRCUIT**).

### 7.2 Oscillation Check at Low Supply Voltage

A reliable reset operation can be expected from the POC circuit if it satisfies the condition that the clock can oscillate even at a low power supply voltage (the oscillation start voltage of the resonator being even lower than the POC detection voltage). Whether this condition is met or not can be checked by measuring the oscillation status on a product which actually contains a POC circuit, as follows.

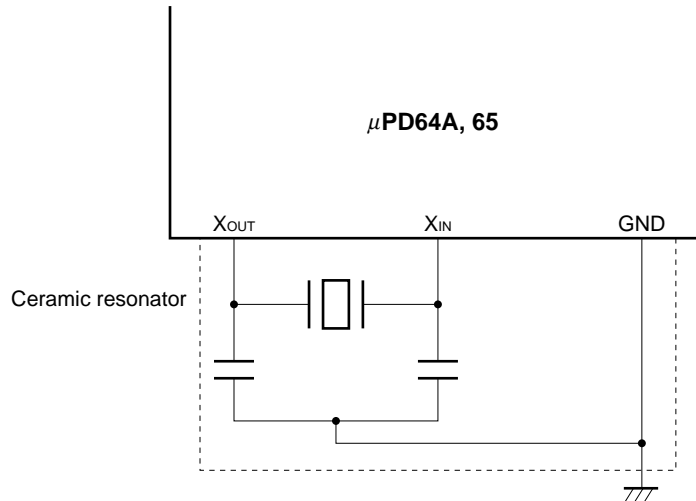
- <1> Connect a storage oscilloscope to the  $X_{OUT}$  pin so that the oscillation status can be measured.
- <2> Connect a power supply whose output voltage can be varied and then gradually raise the power supply voltage  $V_{DD}$  from 0 V (making sure to avoid  $V_{DD} > 3.6$  V).

At first (when  $V_{DD} < \text{approx. } 1.7$  V), the  $X_{OUT}$  pin is 0 V regardless of the  $V_{DD}$ . However, at the point when  $V_{DD}$  reaches the POC detection voltage ( $V_{POC} = 1.85$  V (TYP.)), the voltage of the  $X_{OUT}$  pin jumps to about  $0.5V_{DD}$ . Maintain this power supply voltage for a while to measure the waveform of the  $X_{OUT}$  pin. If, by any chance, the oscillation start voltage of the resonator is lower than the POC detection voltage, the growing oscillation of the  $X_{OUT}$  pin can be confirmed within several ms after  $V_{DD}$  has reached  $V_{POC}$ .

8. SYSTEM CLOCK OSCILLATOR

The system clock oscillator is configured by an oscillator circuit for a ceramic resonator ( $f_x = 2.4$  to 8 MHz).

Figure 8-1. System Clock



The system clock oscillator stops its oscillation after reset or in STOP mode.

**Caution** When using the system clock oscillator, wire as follows in the area enclosed by the broken lines in the above figure to avoid an adverse effect from wiring capacitance.

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

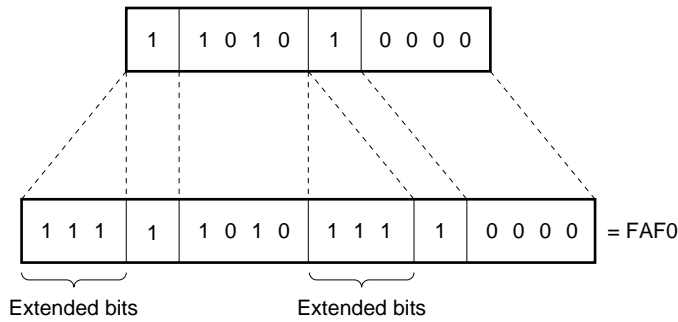
9. INSTRUCTION SET

9.1 Machine Language Output by Assembler

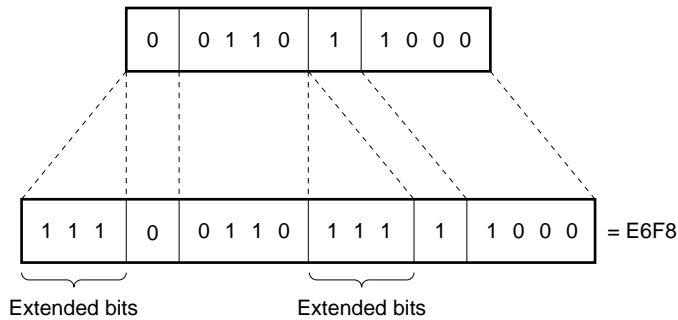
The bit length of the machine language of this product is 10 bits per word. However, the machine language output by the assembler is extended to 16 bits per word. As shown in the example below, the extension is made by inserting 3 extended bits (111) in two locations.

Figure 9-1. Example of Assembler Output (10 Bits Extended to 16 Bits)

<1> In the case of "ANL A, @R0H"



<2> In the case of "OUT P0, #data8"





## 9.2 Circuit Symbol Description

A:	Accumulator
ASR:	Address stack register
addr:	Program memory address
CY:	Carry flag
data4:	4-bit immediate data
data8:	8-bit immediate data
data10:	10-bit immediate data
F:	Status flag
PC:	Program counter
Pn:	Port register pair (n = 0, 1, 3, 4)
P0n:	Port register (lower 4 bits)
P1n:	Port register (higher 4 bits)
ROMn:	Bit n of the program memory (n = 0 to 9)
Rn:	Register pair
R0n:	Data memory (general-purpose register; n = 0 to F)
R1n:	Data memory (general-purpose register; n = 0 to F)
SP:	Stack pointer
T:	Timer register
T0:	Timer register (lower 4 bits)
T1:	Timer register (higher 4 bits)
(x):	Content addressed with x

9.3 Mnemonic to/from Machine Language (Assembler Output) Contrast Table

Accumulator operation instructions

Mnemonic	Operand	Instruction Code			Operation	Instruction Length	Instruction Cycle
		1st Word	2nd Word	3rd Word			
ANL	A, R0n	FBE <sub>n</sub>			$(A) \leftarrow (A) \wedge (Rm_n) \quad m = 0, 1 \quad n = 0 \text{ to } F$	1	1
	A, R1n	FAE <sub>n</sub>			$CY \leftarrow A_3 \cdot Rm_n_3$		
	A, @R0H	FAF0			$(A) \leftarrow (A) \wedge ((P13), (R0))_{7-4}$ $CY \leftarrow A_3 \cdot ROM_7$		
	A, @R0L	FBF0			$(A) \leftarrow (A) \wedge ((P13), (R0))_{3-0}$ $CY \leftarrow A_3 \cdot ROM_3$		
	A, #data4	FBF1	data4		$(A) \leftarrow (A) \wedge \text{data4}$ $CY \leftarrow A_3 \cdot \text{data4}_3$	2	
ORL	A, R0n	FDE <sub>n</sub>			$(A) \leftarrow (A) \vee (Rm_n) \quad m = 0, 1 \quad n = 0 \text{ to } F$	1	
	A, R1n	FCE <sub>n</sub>			$CY \leftarrow 0$		
	A, @R0H	FCF0			$(A) \leftarrow (A) \vee ((P13), (R0))_{7-4}$ $CY \leftarrow 0$		
	A, @R0L	FDF0			$(A) \leftarrow (A) \vee ((P13), (R0))_{3-0}$ $CY \leftarrow 0$		
	A, #data4	FDF1	data4		$(A) \leftarrow (A) \vee \text{data4}$ $CY \leftarrow 0$	2	
XRL	A, R0n	F5E <sub>n</sub>			$(A) \leftarrow (A) \nabla (Rm_n) \quad m = 0, 1 \quad n = 0 \text{ to } F$	1	
	A, R1n	F4E <sub>n</sub>			$CY \leftarrow A_3 \cdot Rm_n_3$		
	A, @R0H	F4F0			$(A) \leftarrow (A) \nabla ((P13), (R0))_{7-4}$ $CY \leftarrow A_3 \cdot ROM_7$		
	A, @R0L	F5F0			$(A) \leftarrow (A) \nabla ((P13), (R0))_{3-0}$ $CY \leftarrow A_3 \cdot ROM_3$		
	A, #data4	F5F1	data4		$(A) \leftarrow (A) \nabla \text{data4}$ $CY \leftarrow A_3 \cdot \text{data4}_3$	2	
INC	A	F4F3			$(A) \leftarrow (A) + 1$ if (A) = 0 CY ← 1 else CY ← 0	1	
RL	A	FCF3			$(A_{n+1}) \leftarrow (A_n), (A_0) \leftarrow (A_3)$ $CY \leftarrow A_3$		
RLZ	A	FEF3			if A = 0 reset else $(A_{n+1}) \leftarrow (A_n), (A_0) \leftarrow (A_3)$ $CY \leftarrow A_3$		

**Input/output instructions**

Mnemonic	Operand	Instruction Code			Operation	Instruction Length	Instruction Cycle
		1st Word	2nd Word	3rd Word			
IN	A, P0n	FFF8 + n	—	—	(A) ← (Pmn) m = 0, 1 n = 0, 1, 3, 4	1	1
	A, P1n	FEF8 + n	—	—	CY ← 0		
OUT	P0n, A	E5F8 + n	—	—	(Pmn) ← (A) m = 0, 1 n = 0, 1, 3, 4		
	P1n, A	E4F8 + n	—	—			
ANL	A, P0n	FBF8 + n	—	—	(A) ← (A) ∧ (Pmn) m = 0, 1 n = 0, 1, 3, 4		
	A, P1n	FAF8 + n	—	—	CY ← A <sub>3</sub> • Pmn <sub>3</sub>		
ORL	A, P0n	fdf8 + n	—	—	(A) ← (A) ∨ (Pmn) m = 0, 1 n = 0, 1, 3, 4		
	A, P1n	FCF8 + n	—	—	CY ← 0		
XRL	A, P0n	F5F8 + n	—	—	(A) ← (A) ⊕ (Pmn) m = 0, 1 n = 0, 1, 3, 4		
	A, P1n	F4F8 + n	—	—	CY ← A <sub>3</sub> • Pmn <sub>3</sub>		

Mnemonic	Operand	Instruction Code			Operation	Instruction Length	Instruction Cycle
		1st Word	2nd Word	3rd Word			
OUT	Pn, #data8	E6F8 + n	data8		(Pn) ← data8 n = 0, 1, 3, 4	2	1

**Remark** Pn: P1n-P0n are handled in pairs.

**Data transfer instructions**

Mnemonic	Operand	Instruction Code			Operation	Instruction Length	Instruction Cycle
		1st Word	2nd Word	3rd Word			
MOV	A, R0n	FFEn			(A) ← (Rmn) m = 0, 1 n = 0 to F	1	1
	A, R1n	FEEEn			CY ← 0		
	A, @R0H	FEF0			(A) ← ((P13), (R0)) <sub>7-4</sub> CY ← 0		
	A, @R0L	FFF0			(A) ← ((P13), (R0)) <sub>3-0</sub> CY ← 0	2	
	A, #data4	FFF1	data4		(A) ← data4 CY ← 0		
	R0n, A	E5En			(Rmn) ← (A) m = 0, 1 n = 0 to F	1	
	R1n, A	E4En					

Mnemonic	Operand	Instruction Code			Operation	Instruction Length	Instruction Cycle
		1st Word	2nd Word	3rd Word			
MOV	Rn, #data8	E6En	data8	—	(R1n-R0n) ← data8 n = 0 to F	2	1
	Rn, @R0	E7En	—	—	(R1n-R0n) ← ((P13), (R0)) n = 1 to F		

**Remark** Rn: R1n-R0n are handled in pairs.

**Branch instructions**

Mnemonic	Operand	Instruction Code			Operation	Instruction Length	Instruction Cycle
		1st Word	2nd Word	3rd Word			
JMP	addr (Page 0)	E8F1	addr		PC ← addr	2	1
	addr (Page 1)	E9F1	addr				
JC	addr (Page 0)	ECF1	addr		If CY = 1 PC ← addr	2	1
	addr (Page 1)	EAF1	addr		else PC ← PC + 2		
JNC	addr (Page 0)	EDF1	addr		If CY = 0 PC ← addr	2	1
	addr (Page 1)	EBF1	addr		else PC ← PC + 2		
JF	addr (Page 0)	EEF1	addr		If F = 1 PC ← addr	2	1
	addr (Page 1)	F0F1	addr		else PC ← PC + 2		
JNF	addr (Page 0)	EFF1	addr		If F = 0 PC ← addr	2	1
	addr (Page 1)	F1F1	addr		else PC ← PC + 2		

**Caution** 0 and 1, which refer to PAGE0 and 1, are not written when describing mnemonics.

**Subroutine instructions**

Mnemonic	Operand	Instruction Code			Operation	Instruction Length	Instruction Cycle
		1st Word	2nd Word	3rd Word			
CALL	addr (Page 0)	E6F2	E8F1	addr	SP ← SP + 1, ASR ← PC, PC ← addr	3	2
	addr (Page 1)	E6F2	E9F1	addr			
RET		E8F2			PC ← ASR, SP ← SP - 1	1	1

**Caution** 0 and 1, which refer to PAGE0 and 1, are not written when describing mnemonics.

**Timer operation instructions**

Mnemonic	Operand	Instruction Code			Operation	Instruction Length	Instruction Cycle
		1st Word	2nd Word	3rd Word			
MOV	A, T0	FFFF			(A) ← (Tn) n = 0, 1	1	1
	A, T1	FEFF			CY ← 0		
	T0, A	E5FF			(Tn) ← (A) n = 0, 1		
	T1, A	F4FF			(T) n ← 0		

Mnemonic	Operand	Instruction Code			Operation	Instruction Length	Instruction Cycle
		1st Word	2nd Word	3rd Word			
MOV	T, #data10	E6FF	data10		(T) ← data10	1	1
	T, @R0	F4FF			(T) ← ((P13), (R0))		

Other instructions

Mnemonic	Operand	Instruction Code			Operation	Instruction Length	Instruction Cycle
		1st Word	2nd Word	3rd Word			
HALT	#data4	E2F1	data4		Standby mode	2	1
STTS	#data4	E3F1	data4		If statuses match $F \leftarrow 1$ else $F \leftarrow 0$		
	R0n	E3En			If statuses match $F \leftarrow 1$ else $F \leftarrow 0$ <span style="float: right;">n = 0 to F</span>	1	
SCAF		FAF3			If A = 0FH $CY \leftarrow 1$ else $CY \leftarrow 0$		
NOP		E0E0			PC $\leftarrow$ PC + 1		

### 9.4 Accumulator Operation Instructions

#### ANL A, R0n

#### ANL A, R1n

<1> Instruction code: 

1	1	0	1	R <sub>4</sub>	0	R <sub>3</sub>	R <sub>2</sub>	R <sub>1</sub>	R <sub>0</sub>
---	---	---	---	----------------	---	----------------	----------------	----------------	----------------

<2> Cycle count: 1

<3> Function:  $(A) \leftarrow (A) \wedge (R_{mn}) \quad m = 0, 1 \quad n = 0 \text{ to } F$   
 $CY \leftarrow A_3 \cdot R_{mn3}$

The accumulator contents and the register R<sub>mn</sub> contents are ANDed and the results are entered in the accumulator.

#### ANL A, @R0H

#### ANL A, @R0L

<1> Instruction code: 

1	1	0	1	0	1	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---

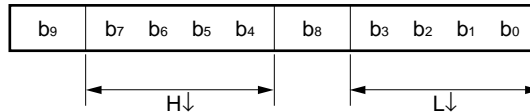
<2> Cycle count: 1

<3> Function:  $(A) \leftarrow (A) \wedge ((P13), (R0))_{7-4}$  (in the case of ANL A, @R0H)  
 $CY \leftarrow A_3 \cdot ROM_7$   
 $(A) \leftarrow (A) \wedge ((P13), (R0))_{3-0}$  (in the case of ANL A, @R0L)  
 $CY \leftarrow A_3 \cdot ROM_3$

The accumulator contents and the program memory contents specified with the control register P13 and register pair R<sub>10</sub>-R<sub>00</sub> are ANDed and the results are entered in the accumulator.

If H is specified, b<sub>7</sub>, b<sub>6</sub>, b<sub>5</sub>, and b<sub>4</sub> take effect. If L is specified, b<sub>3</sub>, b<sub>2</sub>, b<sub>1</sub>, and b<sub>0</sub> take effect.

- Program memory (ROM) organization



Valid bits at the time of accumulator operation

#### ANL A, #data4

<1> Instruction code: 

1	1	0	1	1	1	0	0	0	1
---	---	---	---	---	---	---	---	---	---

0	0	0	0	0	0	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>
---	---	---	---	---	---	----------------	----------------	----------------	----------------

<2> Cycle count: 1

<3> Function:  $(A) \leftarrow (A) \wedge \text{data4}$   
 $CY \leftarrow A_3 \cdot \text{data4}_3$

The accumulator contents and the immediate data are ANDed and the results are entered in the accumulator.

**ORL A, R0n**

**ORL A, R1n**

<1> Instruction code: 

1	1	1	0	R <sub>4</sub>	0	R <sub>3</sub>	R <sub>2</sub>	R <sub>1</sub>	R <sub>0</sub>
---	---	---	---	----------------	---	----------------	----------------	----------------	----------------

<2> Cycle count: 1

<3> Function:  $(A) \leftarrow (A) \vee (R_{mn}) \quad m = 0, 1 \quad n = 0 \text{ to } F$   
CY ← 0

The accumulator contents and the register Rmn contents are ORed and the results are entered in the accumulator.

**ORL A, @R0H**

**ORL A, @R0L**

<1> Instruction code: 

1	1	1	0	0	1	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---

<2> Cycle count: 1

<3> Function:  $(A) \leftarrow (A) \vee (P13), (R0))_{7-4}$  (in the case of ORL A, @R0H)  
 $(A) \leftarrow (A) \vee (P13), (R0))_{3-0}$  (in the case of ORL A, @R0L)  
CY ← 0

The accumulator contents and the program memory contents specified with the control register P13 and register pair R10-R00 are ORed and the results are entered in the accumulator.

If H is specified, b7, b6, b5, and b4 take effect. If L is specified, b3, b2, b1, and b0 take effect.

**ORL A, #data4**

<1> Instruction code: 

1	1	1	0	1	1	0	0	0	1
0	0	0	0	0	0	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>

<2> Cycle count: 1

<3> Function:  $(A) \leftarrow (A) \vee \text{data4}$   
CY ← 0

The accumulator contents and the immediate data are ORed and the results are entered in the accumulator.

**XRL A, R0n**

**XRL A, R1n**

<1> Instruction code: 

1	0	1	0	R <sub>4</sub>	0	R <sub>3</sub>	R <sub>2</sub>	R <sub>1</sub>	R <sub>0</sub>
---	---	---	---	----------------	---	----------------	----------------	----------------	----------------

<2> Cycle count: 1

<3> Function:  $(A) \leftarrow (A) \nabla (R_{mn}) \quad m = 0, 1 \quad n = 0 \text{ to } F$   
CY ← A<sub>3</sub> • Rmn<sub>3</sub>

The accumulator contents and the register Rmn contents are exclusive-ORed and the results are entered in the accumulator.

**XRL A, @R0H**

**XRL A, @R0L**

<1> Instruction code: 

1	0	1	0	0	1	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---

<2> Cycle count: 1

<3> Function:  $(A) \leftarrow (A) \vee (P13), (R0))_{7-4}$  (in the case of XRL A, @R0H)  
 $CY \leftarrow A_3 \cdot ROM_7$   
 $(A) \leftarrow (A) \vee (P13), (R0))_{3-0}$  (in the case of XRL A, @R0L)  
 $CY \leftarrow A_3 \cdot ROM_3$

The accumulator contents and the program memory contents specified with the control register P13 and register pair R10-R00 are exclusive-ORed and the results are entered in the accumulator. If H is specified, b7, b6, b5, and b4 take effect. If L is specified, b3, b2, b1, and b0 take effect.

**XRL A, #data4**

<1> Instruction code: 

1	0	1	0	1	1	0	0	0	1
0	0	0	0	0	0	$d_3$	$d_2$	$d_1$	$d_0$

<2> Cycle count: 1

<3> Function:  $(A) \leftarrow (A) \vee data4$   
 $CY \leftarrow A_3 \cdot data4_3$

The accumulator contents and the immediate data are exclusive-ORed and the results are entered in the accumulator.

**INC A**

<1> Instruction code: 

1	0	1	0	0	1	0	0	1	1
---	---	---	---	---	---	---	---	---	---

<2> Cycle count: 1

<3> Function:  $(A) \leftarrow (A) + 1$   
 If  $A = 0$   $CY \leftarrow 1$   
 else  $CY \leftarrow 0$

The accumulator contents are incremented (+1).

**RL A**

<1> Instruction code: 

1	1	1	0	0	1	0	0	1	1
---	---	---	---	---	---	---	---	---	---

<2> Cycle count: 1

<3> Function:  $(A_{n+1}) \leftarrow (A_n), (A_0) \leftarrow (A_3)$   
 $CY \leftarrow A_3$

The accumulator contents are rotated counterclockwise bit by bit.

**RLZ A**

<1> Instruction code: 

1	1	1	1	0	1	0	0	1	1
---	---	---	---	---	---	---	---	---	---

<2> Cycle count: 1

<3> Function: If  $A = 0$  reset  
 else  $(A_{n+1}) \leftarrow (A_n), (A_0) \leftarrow (A_3)$   
 $CY \leftarrow A_3$

The accumulator contents are rotated counterclockwise bit by bit.

**If A = 0H at the time of instruction execution, an internal reset takes effect.**



### 9.5 I/O Instructions

#### IN A, P0n

#### IN A, P1n

- <1> Instruction code: 

1	1	1	1	P <sub>4</sub>	1	1	P <sub>2</sub>	P <sub>1</sub>	P <sub>0</sub>
---	---	---	---	----------------	---	---	----------------	----------------	----------------
- <2> Cycle count: 1
- <3> Function:  $(A) \leftarrow (P_{mn}) \quad m = 0, 1 \quad n = 0, 1, 3, 4$   
CY ← 0

The port P<sub>mn</sub> data is loaded (read) into the accumulator.

#### OUT P0n, A

#### OUT P1n, A

- <1> Instruction code: 

0	0	1	0	P <sub>4</sub>	1	1	P <sub>2</sub>	P <sub>1</sub>	P <sub>0</sub>
---	---	---	---	----------------	---	---	----------------	----------------	----------------
- <2> Cycle count: 1
- <3> Function:  $(P_{mn}) \leftarrow (A) \quad m = 0, 1 \quad n = 0, 1, 3, 4$

The accumulator contents are transferred to port P<sub>mn</sub> to be latched.

#### ANL A, P0n

#### ANL A, P1n

- <1> Instruction code: 

1	1	0	1	P <sub>4</sub>	1	1	P <sub>2</sub>	P <sub>1</sub>	P <sub>0</sub>
---	---	---	---	----------------	---	---	----------------	----------------	----------------
- <2> Cycle count: 1
- <3> Function:  $(A) \leftarrow (A) \wedge (P_{mn}) \quad m = 0, 1 \quad n = 0, 1, 3, 4$   
CY ← A<sub>3</sub> • P<sub>mn</sub>

The accumulator contents and the port P<sub>mn</sub> contents are ANDed and the results are entered in the accumulator.

#### ORL A, P0n

#### ORL A, P1n

- <1> Instruction code: 

1	1	1	0	P <sub>4</sub>	1	1	P <sub>2</sub>	P <sub>1</sub>	P <sub>0</sub>
---	---	---	---	----------------	---	---	----------------	----------------	----------------
- <2> Cycle count: 1
- <3> Function:  $(A) \leftarrow (A) \vee (P_{mn}) \quad m = 0, 1 \quad n = 0, 1, 3, 4$   
CY ← 0

The accumulator contents and the port P<sub>mn</sub> contents are ORed and the results are entered in the accumulator.

#### XRL A, P0n

#### XRL A, P1n

- <1> Instruction code: 

1	0	1	0	P <sub>4</sub>	1	1	P <sub>2</sub>	P <sub>1</sub>	P <sub>0</sub>
---	---	---	---	----------------	---	---	----------------	----------------	----------------
- <2> Cycle count: 1
- <3> Function:  $(A) \leftarrow (A) \nabla (P_{mn}) \quad m = 0, 1 \quad n = 0, 1, 3, 4$   
CY ← A<sub>3</sub> • P<sub>mn</sub>

The accumulator contents and the port P<sub>mn</sub> contents are exclusive-ORed and the results are entered in the accumulator.

**OUT Pn, #data8**

<1> Instruction code: 

0	0	1	1	0	1	1	P <sub>2</sub>	P <sub>1</sub>	P <sub>0</sub>
0	d <sub>7</sub>	d <sub>6</sub>	d <sub>5</sub>	d <sub>4</sub>	0	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>

<2> Cycle count: 1

<3> Function: (Pn) ← data8 n = 0, 1, 3, 4

The immediate data is transferred to port Pn. In this case, port Pn refers to P<sub>1n</sub>-P<sub>0n</sub> operating in pairs.

**9.6 Data Transfer Instructions**

**MOV A, R0n**

**MOV A, R1n**

<1> Instruction code: 

1	1	1	1	R <sub>4</sub>	0	R <sub>3</sub>	R <sub>2</sub>	R <sub>1</sub>	R <sub>0</sub>
---	---	---	---	----------------	---	----------------	----------------	----------------	----------------

<2> Cycle count: 1

<3> Function: (A) ← (Rmn) m = 0, 1 n = 0 to F  
CY ← 0

The register Rmn contents are transferred to the accumulator.

**MOV A, @R0H**

<1> Instruction code: 

1	1	1	1	0	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---

<2> Cycle count: 1

<3> Function: (A) ← ((P13), (R0))<sub>7-4</sub>  
CY ← 0

The higher 4 bits (b<sub>7</sub> b<sub>6</sub> b<sub>5</sub> b<sub>4</sub>) of the program memory specified with control register P13 and register pair R<sub>10</sub>-R<sub>00</sub> are transferred to the accumulator. b<sub>9</sub> is ignored.

**MOV A, @R0L**

<1> Instruction code: 

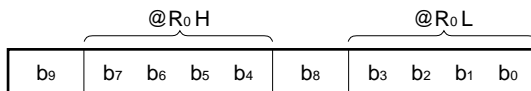
1	1	1	1	1	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---

<2> Cycle count: 1

<3> Function: (A) ← ((P13), (R0))<sub>3-0</sub>  
CY ← 0

The lower 4 bits (b<sub>3</sub> b<sub>2</sub> b<sub>1</sub> b<sub>0</sub>) of the program memory specified with control register P13 and register pair R<sub>10</sub>-R<sub>00</sub> are transferred to the accumulator. b<sub>8</sub> is ignored.

• **Program memory (ROM) contents**



**MOV A, #data4**

<1> Instruction code: 

1	1	1	1	1	1	0	0	0	1
0	0	0	0	0	0	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>

<2> Cycle count: 1

<3> Function: (A) ← data4  
CY ← 0

The immediate data is transferred to the accumulator.

**MOV R0n, A**

**MOV R1n, A**

<1> Instruction code: 

0	0	1	0	R <sub>4</sub>	0	R <sub>3</sub>	R <sub>2</sub>	R <sub>1</sub>	R <sub>0</sub>
---	---	---	---	----------------	---	----------------	----------------	----------------	----------------

<2> Cycle count: 1

<3> Function: (R<sub>m</sub>n) ← (A) m = 0, 1 n = 0 to F  
The accumulator contents are transferred to register R<sub>m</sub>n.

**MOV Rn, #data8**

<1> Instruction code: 

0	0	1	0	0	R <sub>3</sub>	R <sub>2</sub>	R <sub>1</sub>	R <sub>0</sub>	
0	d <sub>7</sub>	d <sub>6</sub>	d <sub>5</sub>	d <sub>4</sub>	0	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>

<2> Cycle count: 1

<3> Function: (R<sub>1</sub>n-R<sub>0</sub>n) ← data8 n = 0 to F

The immediate data is transferred to the register. Using this instruction, registers operate as register pairs.

The pair combinations are as follows.

- R0: R<sub>10</sub> - R<sub>00</sub>
  - R1: R<sub>11</sub> - R<sub>01</sub>
  - ⋮
  - RE: R<sub>1E</sub> - R<sub>0E</sub>
  - RF: R<sub>1F</sub> - R<sub>0F</sub>
- └──┬── Lower column  
└──┬── Higher column

**MOV Rn, @R0**

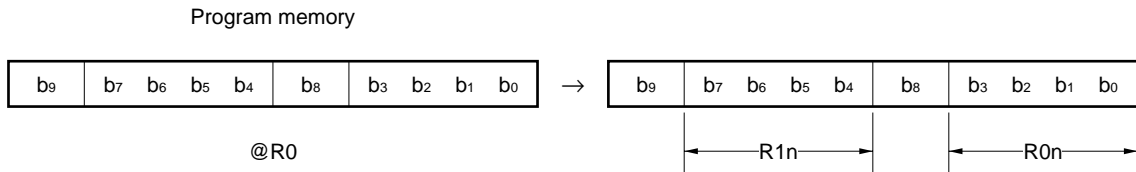
<1> Instruction code: 

0	0	1	1	1	0	R <sub>3</sub>	R <sub>2</sub>	R <sub>1</sub>	R <sub>0</sub>
---	---	---	---	---	---	----------------	----------------	----------------	----------------

<2> Cycle count: 1

<3> Function: (R<sub>1</sub>n-R<sub>0</sub>n) ← ((P13), R0) n = 1 to F

The program memory contents specified with control register P13 and register pair R<sub>10</sub>-R<sub>00</sub> are transferred to register pair R<sub>1</sub>n-R<sub>0</sub>n. The program memory consists of 10 bits and has the following state after the transfer to the register.



★ The higher 3 bits of the program memory address is specified with the control register (P13).

### 9.7 Branch Instructions

The program memory consists of pages in steps of 1K (000H to 3FFH). However, as the assembler automatically performs page optimization, it is unnecessary to designate pages. The pages allowed for each product are as follows.

- μPD64A (ROM: 1K steps): Page 0
- μPD65 (ROM: 2K steps): Pages 0, 1
- μPD6P5 (PROM: 2K steps): Pages 0, 1

#### JMP addr

<1> Instruction code: page 0 

0	1	0	0	0	1	0	0	0	1
---	---	---	---	---	---	---	---	---	---

, page 1 

0	1	0	0	1	1	0	0	0	1
---	---	---	---	---	---	---	---	---	---

a <sub>9</sub>	a <sub>7</sub>	a <sub>6</sub>	a <sub>5</sub>	a <sub>4</sub>	a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

<2> Cycle count: 1

<3> Function: PC ← addr

The 10 bits (PC<sub>9:0</sub>) of the program counter are replaced directly by the specified address addr (a<sub>9</sub> to a<sub>0</sub>).

#### JC addr

<1> Instruction code: page 0 

0	1	1	0	0	1	0	0	0	1
---	---	---	---	---	---	---	---	---	---

, page 1 

0	1	0	1	0	1	0	0	0	1
---	---	---	---	---	---	---	---	---	---

a <sub>9</sub>	a <sub>7</sub>	a <sub>6</sub>	a <sub>5</sub>	a <sub>4</sub>	a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

<2> Cycle count: 1

<3> Function: If CY = 1 PC ← addr  
 else PC ← PC + 2

If the carry flag CY is set (1), a jump is made to the address specified with addr (a<sub>9</sub> to a<sub>0</sub>).

#### JNC addr

<1> Instruction code: page 0 

0	1	1	0	1	1	0	0	0	1
---	---	---	---	---	---	---	---	---	---

, page 1 

0	1	0	1	1	1	0	0	0	1
---	---	---	---	---	---	---	---	---	---

a <sub>9</sub>	a <sub>7</sub>	a <sub>6</sub>	a <sub>5</sub>	a <sub>4</sub>	a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

<2> Cycle count: 1

<3> Function: If CY = 0 PC ← addr  
 else PC ← PC + 2

If the carry flag CY is cleared (0), a jump is made to the address specified with addr (a<sub>9</sub> to a<sub>0</sub>).

#### JF addr

<1> Instruction code: page 0 

0	1	1	1	0	1	0	0	0	1
---	---	---	---	---	---	---	---	---	---

, page 1 

1	0	0	0	0	1	0	0	0	1
---	---	---	---	---	---	---	---	---	---

a <sub>9</sub>	a <sub>7</sub>	a <sub>6</sub>	a <sub>5</sub>	a <sub>4</sub>	a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

<2> Cycle count: 1

<3> Function: If F = 1 PC ← addr  
 else PC ← PC + 2

If the status flag F is set (1), a jump is made to the address specified with addr (a<sub>9</sub> to a<sub>0</sub>).

#### JNF addr

<1> Instruction code: page 0 

0	1	1	1	1	1	0	0	0	1
---	---	---	---	---	---	---	---	---	---

, page 1 

1	0	0	0	1	1	0	0	0	1
---	---	---	---	---	---	---	---	---	---

a <sub>9</sub>	a <sub>7</sub>	a <sub>6</sub>	a <sub>5</sub>	a <sub>4</sub>	a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

<2> Cycle count: 1

<3> Function: If F = 0 PC ← addr  
 else PC ← PC + 2

If the status flag F is cleared (0), a jump is made to the address specified with addr (a<sub>9</sub> to a<sub>0</sub>).

### 9.8 Subroutine Instructions

The program memory consists of pages in steps of 1K (000H to 3FFH). However, as the assembler automatically performs page optimization, it is unnecessary to designate pages. The pages allowed for each product are as follows.

- μPD64A (ROM: 1K steps): Page 0
- μPD65 (ROM: 2K steps): Pages 0, 1
- μPD6P5 (PROM: 2K steps): Pages 0, 1

#### CALL addr

- <1> Instruction code: 

0	0	1	1	0	1	0	0	1	0
---	---	---	---	---	---	---	---	---	---

page 0	0	1	0	0	0	1	0	0	0	1
--------	---	---	---	---	---	---	---	---	---	---

, page 1 

0	1	0	0	1	1	0	0	0	1
---	---	---	---	---	---	---	---	---	---

a <sub>9</sub>	a <sub>7</sub>	a <sub>6</sub>	a <sub>5</sub>	a <sub>4</sub>	a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------
- <2> Cycle count: 2
- <3> Function: SP ← SP + 1  
ASR ← PC  
PC ← addr

The stack pointer value is incremented (+1) and the program counter value is saved to the address stack register. The address specified with the operand addr (a<sub>9</sub> to a<sub>0</sub>) is then entered in the program counter. If a carry is generated when the stack pointer value is incremented (+1), an internal reset takes effect.

#### RET

- <1> Instruction code: 

0	1	0	0	0	1	0	0	1	0
---	---	---	---	---	---	---	---	---	---
- <2> Cycle count: 1
- <3> Function: PC ← ASR  
SP ← SP - 1

The value saved in the address stack register is restored to the program counter. The stack pointer is then decremented (-1).

If a borrow is generated when the stack pointer value is decremented (-1), an internal reset takes effect.

### 9.9 Timer Operation Instructions

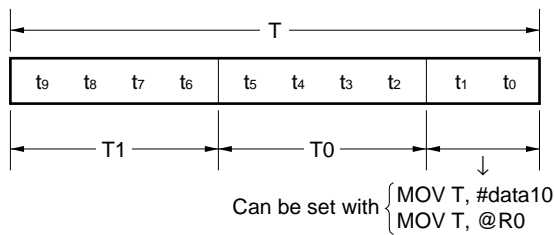
**MOV A, T0**

**MOV A, T1**

- <1> Instruction code: 

1	1	1	1	0/1	1	1	1	1	1
---	---	---	---	-----	---	---	---	---	---
- <2> Cycle count: 1
- <3> Function:  $(A) \leftarrow (T_n) \quad n = 0, 1$   
 $CY \leftarrow 0$

The timer T<sub>n</sub> contents are transferred to the accumulator. T1 corresponds to (t<sub>9</sub>, t<sub>8</sub>, t<sub>7</sub>, t<sub>6</sub>); T0 corresponds to (t<sub>5</sub>, t<sub>4</sub>, t<sub>3</sub>, t<sub>2</sub>).



**MOV T0, A**

**MOV T1, A**

- <1> Instruction code: 

0	0	1	0/1	1	1	1	1	1	1
---	---	---	-----	---	---	---	---	---	---
- <2> Cycle count: 1
- <3> Function:  $(T_n) \leftarrow (A) \quad n = 0, 1$

The accumulator contents are transferred to timer register T<sub>n</sub>. T1 corresponds to (t<sub>9</sub>, t<sub>8</sub>, t<sub>7</sub>, t<sub>6</sub>); T0 corresponds to (t<sub>5</sub>, t<sub>4</sub>, t<sub>3</sub>, t<sub>2</sub>). **After executing this instruction, if data is transferred to T1, t<sub>1</sub> becomes 0; if data is transferred to T0, t<sub>0</sub> becomes 0.**

**MOV T, #data10**

- <1> Instruction code: 

0	0	1	1	0	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---

t <sub>9</sub>	t <sub>8</sub>	t <sub>7</sub>	t <sub>6</sub>	t <sub>5</sub>	t <sub>4</sub>	t <sub>3</sub>	t <sub>2</sub>
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------
- <2> Cycle count: 1
- <3> Function:  $(T) \leftarrow \text{data10}$

The immediate data is transferred to timer register T (t<sub>9</sub>-t<sub>0</sub>).

**Remark** The timer time is set with (set value + 1) × 64/f<sub>x</sub> (or 128/f<sub>x</sub>).

**MOV T, @R0**

<1> Instruction code: 

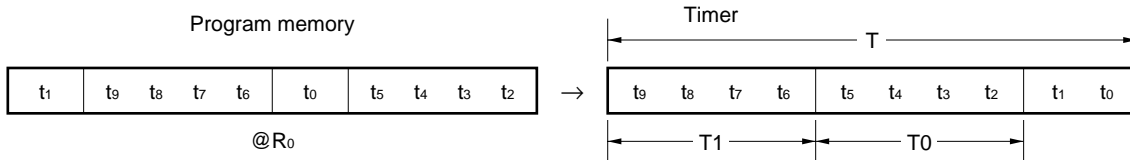
0	0	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---

<2> Cycle count: 1

<3> Function:  $(T) \leftarrow ((P13), (R0))$

The program memory contents specified by the control register P13 and the register pair R<sub>10</sub>-R<sub>00</sub> are transferred to timer register T (t<sub>9</sub> to t<sub>0</sub>).

The program memory, which consists of 10 bits, is placed in the following state after the transfer to the register.



★ The higher 3 bits of the program memory address are specified by the control register (P13).

**Caution** When setting a timer value in the program memory, be sure to use the DT quasi directive.

**9.10 Other Instructions**

**HALT #data4**

<1> Instruction code: 

0	0	0	1	0	1	0	0	0	1
---	---	---	---	---	---	---	---	---	---

0	0	0	0	0	0	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>
---	---	---	---	---	---	----------------	----------------	----------------	----------------

<2> Cycle count: 1

<3> Function: Standby mode

Places the CPU in standby mode.

The condition to cancel the standby mode (HALT/STOP mode) is specified by the immediate data.

**STTS R0n**

<1> Instruction code: 

0	0	0	1	1	0	R <sub>3</sub>	R <sub>2</sub>	R <sub>1</sub>	R <sub>0</sub>
---	---	---	---	---	---	----------------	----------------	----------------	----------------

<2> Cycle count: 1

<3> Function: If statuses match  $F \leftarrow 1$   
 else  $F \leftarrow 0$  n = 0 to F

The S<sub>0</sub>, S<sub>1</sub>, K<sub>I/O</sub>, K<sub>I</sub>, and TIMER statuses are compared with the register R<sub>0n</sub> contents. If at least one of the statuses matches the bits that have been set, status flag F is set (1).

If none of them match, status flag F is cleared (0).

**STTS #data4**

<1> Instruction code: 

0	0	0	1	1	0	0	0	1	
0	0	0	0	0	0	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>

<2> Cycle count: 1

<3> Function: If statuses match  $F \leftarrow 1$   
else  $F \leftarrow 0$

The S<sub>0</sub>, S<sub>1</sub>, K<sub>I/O</sub>, K<sub>I</sub>, and TIMER statuses are compared with the immediate data contents. If at least one of the statuses matches the bits that have been set, status flag F is set (1).

If none of them match, status flag F is cleared (0).

**SCAF (Set Carry If Acc = FH)**

<1> Instruction code: 

1	1	0	1	0	1	0	0	1	1
---	---	---	---	---	---	---	---	---	---

<2> Cycle count: 1

<3> Function: If  $A = 0FH$   $CY \leftarrow 1$   
else  $CY \leftarrow 0$

The carry flag CY is set (1) if the accumulator contents are FH.

The accumulator values after executing the SCAF instruction are as follows.

Accumulator Value		Carry Flag
Before Execution	After Execution	
xxx0	0000	0 (clear)
xx01	0001	0 (clear)
x011	0011	0 (clear)
0111	0111	0 (clear)
1111	1111	1 (set)

**Remark** x: don't care

**NOP**

<1> Instruction code: 

0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---

<2> Cycle count: 1

<3> Function:  $PC \leftarrow PC + 1$

No operation



**10. ASSEMBLER RESERVED WORDS**

**10.1 Mask Option Quasi Directives**

When creating the μPD64A and 65 program, it is necessary to use a mask option quasi directive in the assembler's source program.

**10.1.1 OPTION and ENDOP quasi directives**

The OPTION and subsequent quasi directives down to the ENDOP quasi directive are called the mask option definition block. The format of the mask option definition block is as follows.

**Format**

<u>Symbol field</u> [Label:]	<u>Mnemonic field</u> OPTION	<u>Operand field</u>	<u>Comment field</u> [; Comment]
	:		
	:		
	ENDOP		

**10.1.2 Mask option definition quasi directive**

The quasi directive that can be used in the mask option definition block is shown in Table 10-1.

The mask option definition can only be specified as follows. Be sure to specify the following quasi directive.

**Example**

<u>Symbol field</u>	<u>Mnemonic field</u>	<u>Operand field</u>	<u>Comment field</u>
	OPTION		
	USEPOC		; POC circuit incorporated
	ENDOP		

**Table 10-1. Mask Option Definition Quasi Directive**

Name	Mask Option Definition Quasi Directive	PRO File	
		Address Value	Data Value
POC	USEPOC (POC circuit incorporated)	2044H	01

11. ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings (T<sub>A</sub> = +25°C)

Parameter	Symbol	Conditions		Ratings	Unit
Power supply voltage	V <sub>DD</sub>			-0.3 to +3.8	V
Input voltage	V <sub>I</sub>	K <sub>I/O</sub> , K <sub>I</sub> , S <sub>0</sub> , S <sub>1</sub> , S <sub>2</sub>		-0.3 to V <sub>DD</sub> + 0.3	V
Output voltage	V <sub>O</sub>			-0.3 to V <sub>DD</sub> + 0.3	V
Output current, high	I <sub>OH</sub> Note	REM	Peak value	-30	mA
			rms value	-20	mA
		LED	Peak value	-7.5	mA
			rms value	-5	mA
		Per K <sub>I/O</sub> pin	Peak value	-13.5	mA
			rms value	-9	mA
		Total of LED and K <sub>I/O</sub> pins	Peak value	-18	mA
			rms value	-12	mA
Output current, low	I <sub>OL</sub> Note	REM	Peak value	7.5	mA
			rms value	5	mA
		LED	Peak value	7.5	mA
			rms value	5	mA
Operating ambient temperature	T <sub>A</sub>			-40 to +85	°C
Storage temperature	T <sub>stg</sub>			-65 to +150	°C

**Note** The rms value should be calculated as follows: [rms value] = [Peak value] × √Duty

**Caution** Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Recommended Power Supply Voltage Range (T<sub>A</sub> = -40 to +85°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Power supply voltage	V <sub>DD</sub>	f <sub>x</sub> = 2.4 to 8 MHz	2.0	3.0	3.6	V

DC Characteristics (T<sub>A</sub> = -40 to +85°C, V<sub>DD</sub> = 2.0 to 3.6 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Input voltage, high	V <sub>IH1</sub>	S <sub>2</sub>	0.8V <sub>DD</sub>		V <sub>DD</sub>	V	
	V <sub>IH2</sub>	K <sub>I/O</sub>	0.7V <sub>DD</sub>		V <sub>DD</sub>	V	
	V <sub>IH3</sub>	K <sub>I</sub> , S <sub>0</sub> , S <sub>1</sub>	0.65V <sub>DD</sub>		V <sub>DD</sub>	V	
Input voltage, low	V <sub>IL1</sub>	S <sub>2</sub>	0		0.2V <sub>DD</sub>	V	
	V <sub>IL2</sub>	K <sub>I/O</sub>	0		0.3V <sub>DD</sub>	V	
	V <sub>IL3</sub>	K <sub>I</sub> , S <sub>0</sub> , S <sub>1</sub>	0		0.15V <sub>DD</sub>	V	
Input leakage current, high	I <sub>LIH1</sub>	K <sub>I</sub> V <sub>I</sub> = V <sub>DD</sub> , pull-down resistor not incorporated			3	μA	
	I <sub>LIH2</sub>	S <sub>0</sub> , S <sub>1</sub> , S <sub>2</sub> V <sub>I</sub> = V <sub>DD</sub> , pull-down resistor not incorporated			3	μA	
Input leakage current, low	I <sub>LIL1</sub>	K <sub>I</sub> V <sub>I</sub> = 0 V			-3	μA	
	I <sub>LIL2</sub>	K <sub>I/O</sub> V <sub>I</sub> = 0 V			-3	μA	
	I <sub>LIL3</sub>	S <sub>0</sub> , S <sub>1</sub> , S <sub>2</sub> V <sub>I</sub> = 0 V			-3	μA	
Output voltage, high	V <sub>OH1</sub>	REM, $\overline{\text{LED}}$ , K <sub>I/O</sub>   I <sub>OH</sub> = -0.3 mA	0.8V <sub>DD</sub>			V	
Output voltage, low	V <sub>OL1</sub>	REM, $\overline{\text{LED}}$   I <sub>OL</sub> = 0.3 mA			0.3	V	
	V <sub>OL2</sub>	K <sub>I/O</sub>   I <sub>OL</sub> = 15 μA			0.4	V	
Output current, high	I <sub>OH1</sub>	REM   V <sub>DD</sub> = 3.0 V, V <sub>OH</sub> = 1.0 V	-5	-12		mA	
	I <sub>OH2</sub>	K <sub>I/O</sub>   V <sub>DD</sub> = 3.0 V, V <sub>OH</sub> = 2.2 V	-2.5	-7		mA	
Output current, low	I <sub>OL1</sub>	K <sub>I/O</sub>   V <sub>DD</sub> = 3.0 V, V <sub>OL</sub> = 0.4 V	30	70		μA	
		K <sub>I/O</sub>   V <sub>DD</sub> = 3.0 V, V <sub>OL</sub> = 2.2 V	100	390		μA	
On-chip pull-down resistor	R <sub>1</sub>	K <sub>I</sub> , S <sub>0</sub> , S <sub>1</sub> , S <sub>2</sub>	75	150	300	kΩ	
	R <sub>2</sub>	K <sub>I/O</sub>	130	250	500	kΩ	
Data retention power supply voltage	V <sub>DDDR</sub>	In STOP mode	0.9		3.6	V	
Supply current	I <sub>DD1</sub>	Operating mode	f <sub>X</sub> = 8.0 MHz, V <sub>DD</sub> = 3 V ±10%		0.8	1.6	mA
			f <sub>X</sub> = 4.0 MHz, V <sub>DD</sub> = 3 V ±10%		0.7	1.4	mA
	I <sub>DD2</sub>	HALT mode	f <sub>X</sub> = 8.0 MHz, V <sub>DD</sub> = 3 V ±10%		0.75	1.5	mA
			f <sub>X</sub> = 4.0 MHz, V <sub>DD</sub> = 3 V ±10%		0.65	1.3	mA
	I <sub>DD3</sub>	STOP mode	V <sub>DD</sub> = 3 V ±10%		1.9	9.0	μA
			V <sub>DD</sub> = 3 V ±10%, T <sub>A</sub> = 25°C		1.9	5.0	μA

**AC Characteristics (T<sub>A</sub> = -40 to +85°C, V<sub>DD</sub> = 2.0 to 3.6 V)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Instruction execution time	t <sub>cy</sub>		7.9		27	μs	
K <sub>I</sub> , S <sub>0</sub> , S <sub>1</sub> , S <sub>2</sub> high-level width	t <sub>H</sub>		10			μs	
		When standby mode is released	In HALT mode	10			μs
			In STOP mode	<b>Note</b>			μs

**Note** 10 + 52/f<sub>x</sub> + oscillation growth time

**Remark** t<sub>cy</sub> = 64/f<sub>x</sub> (f<sub>x</sub>: System clock oscillation frequency)

**POC Circuit (T<sub>A</sub> = -40 to +85°C)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
POC detection voltage <sup>Note</sup>	V <sub>POC</sub>			1.85	2.0	V

**Note** The voltage with which the POC circuit cancels an internal reset. If V<sub>POC</sub> < V<sub>DD</sub>, the internal reset is canceled.

From the time of V<sub>POC</sub> ≥ V<sub>DD</sub> until the internal reset takes effect, a lag of up to 1 ms occurs. When the period of V<sub>POC</sub> ≥ V<sub>DD</sub> lasts less than 1 ms, the internal reset may not take effect.

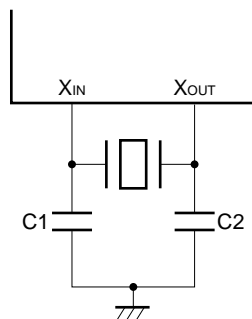
**System Clock Oscillator Characteristics (T<sub>A</sub> = -40 to +85°C, V<sub>DD</sub> = 2.0 to 3.6 V)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Oscillation frequency (ceramic resonator)	f <sub>x</sub>		2.4	3.64	8.0	MHz

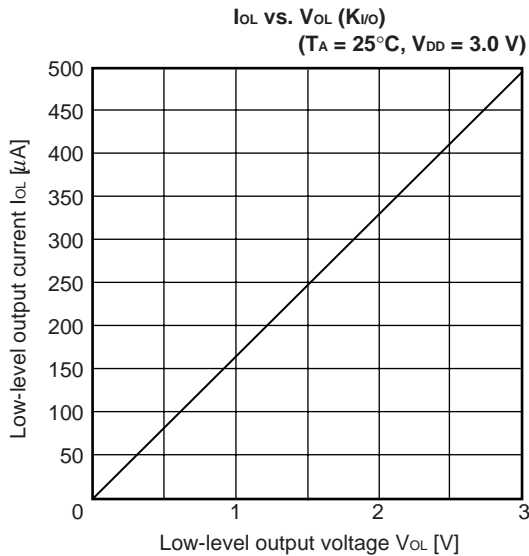
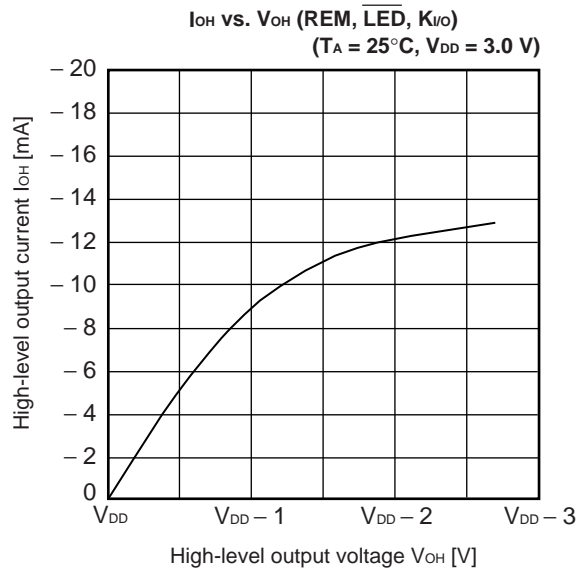
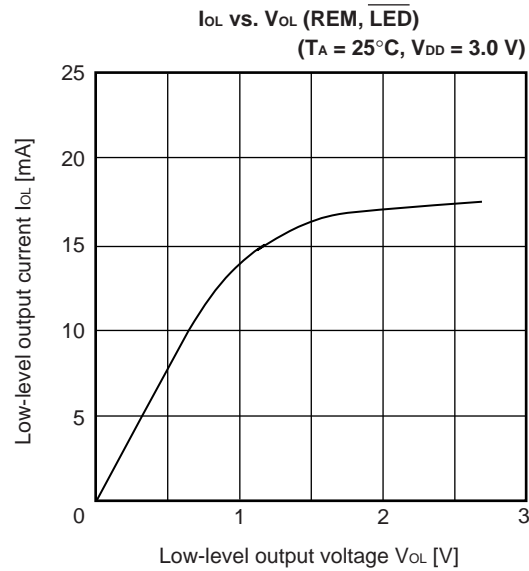
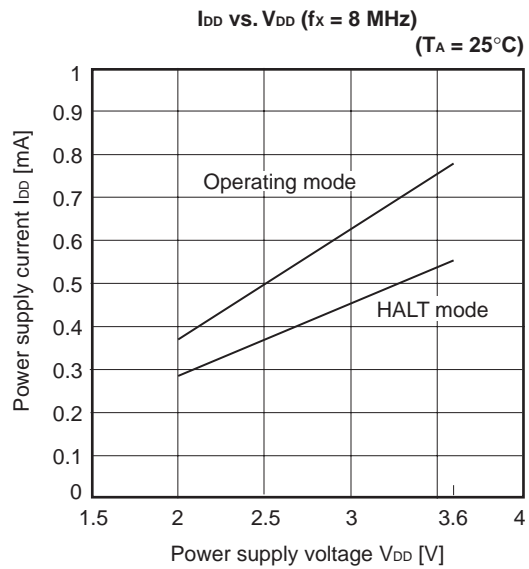
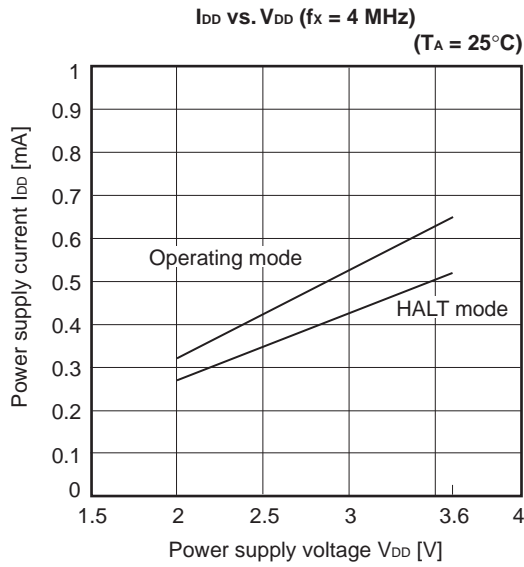
Recommended Ceramic Resonator (T<sub>A</sub> = -40 to +85°C)

Manufacturer	Part Number	Frequency (MHz)	Recommended Constant		Power Supply Voltage [V]		Remark
			C1 [pF]	C2 [pF]	MIN.	MAX.	
TDK Corp.	FCR3.52MC5	3.52	Unnecessary (On-chip C type)		2.0	3.6	
	FCR3.58MC5	3.58					
	FCR3.64MC5	3.64					
	FCR3.84MC5	3.84					
	FCR4.0MC5	4.0					
	FCR6.0MC5	6.0					
	FCR8.0MC5	8.0					
Murata Mfg. Co., Ltd	CSA2.50MG040	2.5	100	100			
	CST2.50MG040		Unnecessary (On-chip C type)				
	CSA3.52MG	3.52	30	30			
	CST3.52MGW		Unnecessary (On-chip C type)				
	CSTS0352MG03		Unnecessary (On-chip C type)				
	CSA3.58MG	3.58	30	30			
	CST3.58MGW		Unnecessary (On-chip C type)				
	CST0358MG03		Unnecessary (On-chip C type)				
	CSA3.64MG	3.64	30	30			
	CST3.64MGW		Unnecessary (On-chip C type)				
	CSTS0364MG03		Unnecessary (On-chip C type)				
	CSA3.84MG	3.84	30	30			
	CST3.84MGW		Unnecessary (On-chip C type)				
	CST0384MG03		Unnecessary (On-chip C type)				
	CSA4.00MG	4.0	30	30			
	CST4.00MGW		Unnecessary (On-chip C type)				
	CSTS0400MG03		Unnecessary (On-chip C type)				
	CSA6.00MG	6.0	30	30			
	CST6.00MGW		Unnecessary (On-chip C type)				
	CSTS0600MG03		Unnecessary (On-chip C type)				
CSA8.00MTZ	8.0	30	30				
CST8.00MTW		Unnecessary (On-chip C type)					
CSTS0800MG03		Unnecessary (On-chip C type)					

External circuit example



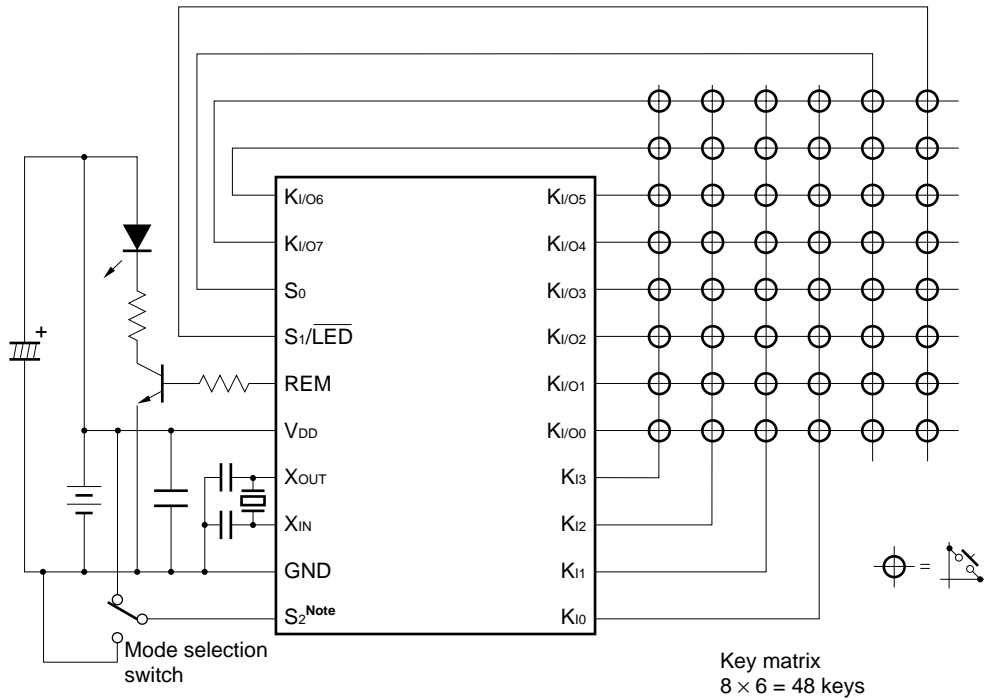
12. CHARACTERISTIC CURVES (REFERENCE VALUES)



13. APPLICATION CIRCUIT EXAMPLE

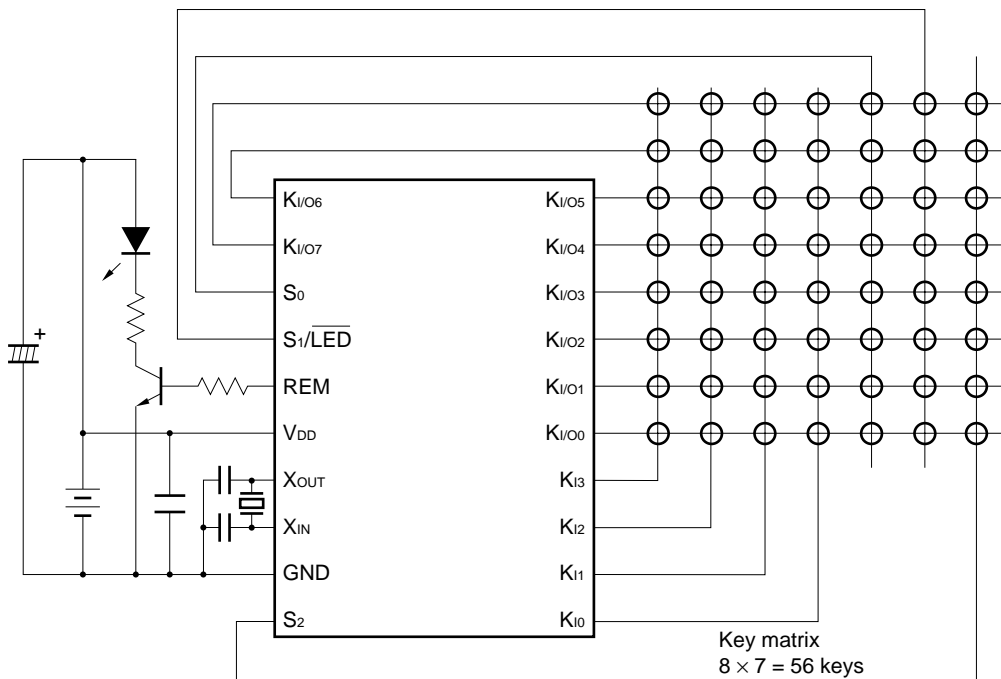
Example of application to system

- ★ ■ Remote-control transmitter (48 keys; mode selection switch accommodated)



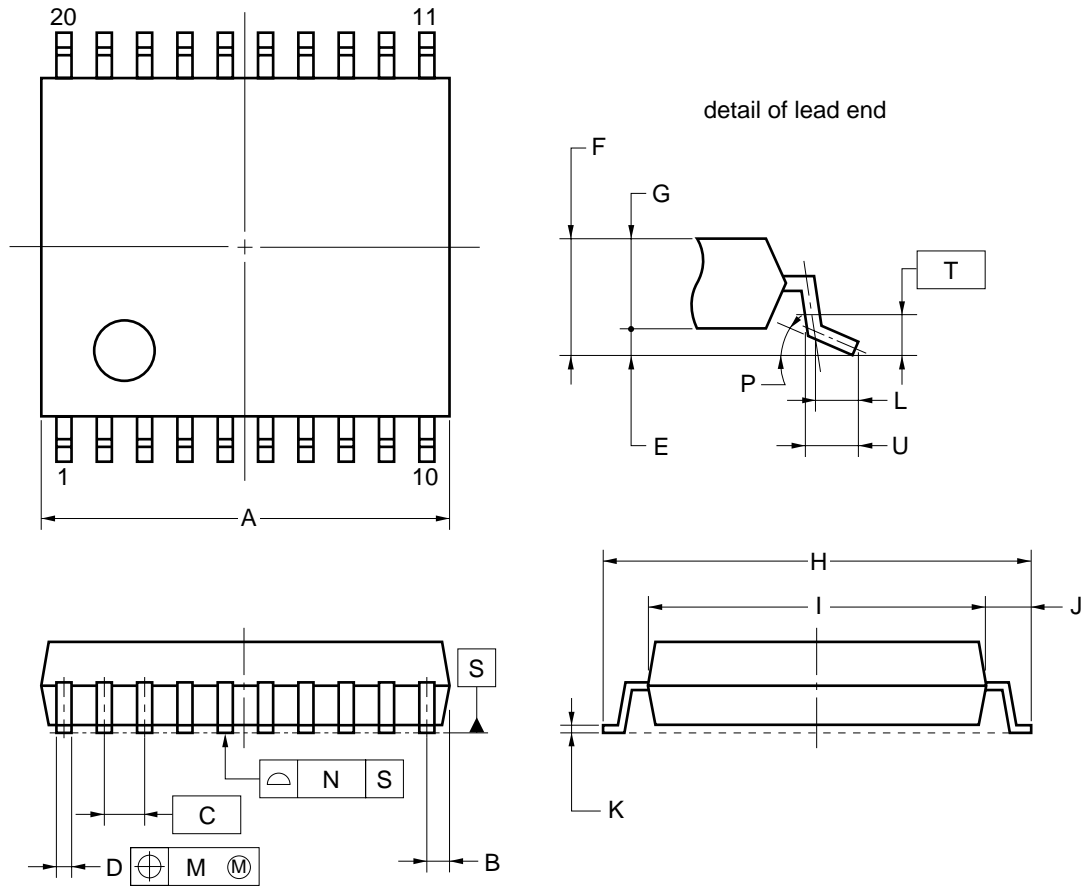
**Note** S2: Set this pin to disabled when releasing STOP mode.

- ★ ■ Remote-control transmitter (56 keys accommodated)



14. PACKAGE DRAWING

20-PIN PLASTIC SSOP (7.62 mm (300))



**NOTE**

Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	6.65±0.15
B	0.475 MAX.
C	0.65 (T.P.)
D	0.24 <sup>+0.08</sup> <sub>-0.07</sub>
E	0.1±0.05
F	1.3±0.1
G	1.2
H	8.1±0.2
I	6.1±0.2
J	1.0±0.2
K	0.17±0.03
L	0.5
M	0.13
N	0.10
P	3° <sup>+5°</sup> <sub>-3°</sub>
T	0.25
U	0.6±0.15

S20MC-65-5A4-2

**Remark** The external dimensions and material of the ES version are the same as those of the mass-produced version.



**15. RECOMMENDED SOLDERING CONDITIONS**

The μPD64A and 65 should be soldered and mounted under the following recommended conditions.

For the details of the recommended soldering conditions, refer to the document **Semiconductor Device Mounting Technology Manual (C10535E)**.

For soldering methods and conditions other than those recommended below, contact your NEC sales representative.

**Table 15-1. Surface Mounting Type Soldering Conditions**

**μPD64AMC-xxx-5A4: 20-pin plastic SSOP (7.62 mm (300))**

**μPD65MC-xxx-5A4: 20-pin plastic SSOP (7.62 mm (300))**

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C, time: 30 seconds max. (210°C or higher), count: three times or less	IR35-00-3
VPS	Package peak temperature: 215°C, time: 40 seconds max. (200°C or higher), count: three times or less	VP15-00-3
Wave soldering	Solder bath temperature: 260°C max., time: 10 seconds max, count: once, Preheating temperature: 120°C max. (package surface temperature)	WS60-00-1
Partial heating	Pin temperature: 300°C max., time: 3 seconds max. (per pin row)	—

**Caution Do not use different soldering methods together (except for partial heating).**

**APPENDIX A. DEVELOPMENT TOOLS**

Emulators are provided for the μPD64A and 65 emulation tools.

**Hardware**

- ★ • **Emulators (EB-65, EB-69<sup>Note</sup>)**

These are tools used to emulate the μPD64A and 65.

**Note** Products of Naito Densai Machida Mfg. Co., Ltd. For details, consult Naito Densai Machida Mfg. Co., Ltd. (+81-44-822-3813).

**Software**

- **Assembler (AS6133)**
  - This is a development tool for remote control transmitter software.

**Part Number List of AS6133**

Host Machine	OS	Supply Medium	Part Number
PC-9800 series (CPU: 80386 or more)	MS-DOS™ (Ver. 5.0 to Ver. 6.2)	3.5-inch 2HD	μS5A13AS6133
IBM PC/AT™ compatible	MS-DOS (Ver. 6.0 to Ver. 6.22)	3.5-inch 2HC	μS7B13AS6133
	PC DOS™ (Ver. 6.1 to Ver. 6.3)		

**Caution** Although Ver.5.0 or later has a task swap function, this function cannot be used with this software.

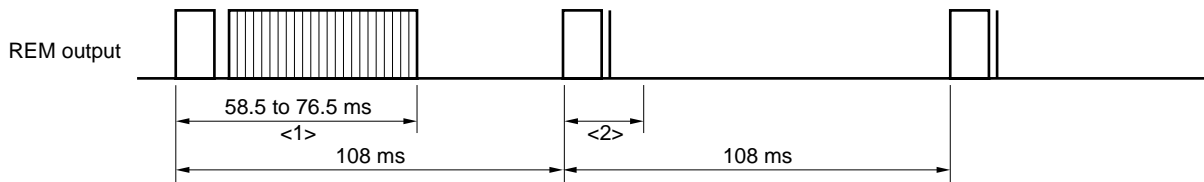
APPENDIX B. FUNCTIONAL COMPARISON BETWEEN μPD64A, 65 AND OTHER PRODUCTS

Item		μPD62	μPD62A	μPD64	μPD64A	μPD65
ROM capacity		512 × 10 bits	512 × 10 bits	1002 × 10 bits	1002 × 10 bits	2026 × 10 bits
RAM capacity		32 × 4 bits				
Stack		1 level (multiplexed with RF of RAM)				
Key matrix		8 × 6 = 48 keys			8 × 7 = 56 keys	
Key extended input		S <sub>0</sub> , S <sub>1</sub>			S <sub>0</sub> , S <sub>1</sub> , S <sub>2</sub>	
Clock frequency		Ceramic oscillation • f <sub>x</sub> = 2.4 to 8 MHz • f <sub>x</sub> = 2.4 to 4 MHz (with POC circuit)	Ceramic oscillation • f <sub>x</sub> = 2.4 to 8 MHz	Ceramic oscillation • f <sub>x</sub> = 2.4 to 8 MHz • f <sub>x</sub> = 2.4 to 4 MHz (with POC circuit)	Ceramic oscillation • f <sub>x</sub> = 2.4 to 8 MHz	
Timer	Clock	f <sub>x</sub> /64, f <sub>x</sub> /128				
	Count start	Writing count value				
Carrier	Frequency	<ul style="list-style-type: none"> <li>f<sub>x</sub>/8, f<sub>x</sub>/64, f<sub>x</sub>/96 (timer clock: f<sub>x</sub>/64)</li> <li>f<sub>x</sub>/16, f<sub>x</sub>/128, f<sub>x</sub>/192 (timer clock: f<sub>x</sub>/128)</li> <li>No carrier</li> </ul>				
	Output start	Synchronized with timer				
Instruction execution time		16 μs (f <sub>x</sub> = 4 MHz)				
"MOV Rn, @R0" instruction		n = 1 to F				
Standby mode	Reset	RESET input, POC			POC	
	Release condition (HALT instruction)	<ul style="list-style-type: none"> <li>HALT mode for timer only.</li> <li>STOP mode for only releasing K<sub>i</sub> (K<sub>I/O</sub> high-level output or K<sub>I/O0</sub> high-level output)</li> </ul>				
Relationship between HALT instruction execution and status flag (F)		HALT instruction not executed when F = 1				
POC circuit		<ul style="list-style-type: none"> <li>Mask option</li> <li>Low level output to RESET pin on detection</li> </ul>			<ul style="list-style-type: none"> <li>Provided</li> <li>Generates internal reset signal on detection</li> </ul>	
	POC detection voltage	V <sub>POC</sub> = 1.6 V (TYP.)	V <sub>POC</sub> = 1.85 V (TYP.)	V <sub>POC</sub> = 1.6 V (TYP.)	V <sub>POC</sub> = 1.85 V (TYP.)	
Mask option		POC circuit only			None	
Supply voltage		<ul style="list-style-type: none"> <li>V<sub>DD</sub> = 1.8 to 3.6 V</li> <li>V<sub>DD</sub> = 2.2 to 3.6 V (with POC circuit)</li> </ul>	V <sub>DD</sub> = 2.0 to 3.6 V	<ul style="list-style-type: none"> <li>V<sub>DD</sub> = 1.8 to 3.6 V</li> <li>V<sub>DD</sub> = 2.2 to 3.6 V (with POC circuit)</li> </ul>	V <sub>DD</sub> = 2.0 to 3.6 V	
Operating ambient temperature		<ul style="list-style-type: none"> <li>T<sub>A</sub> = -40 to +85°C</li> <li>T<sub>A</sub> = -20 to +70°C (with POC circuit)</li> </ul>	T <sub>A</sub> = -40 to +85°C	<ul style="list-style-type: none"> <li>T<sub>A</sub> = -40 to +85°C</li> <li>T<sub>A</sub> = -20 to +70°C (with POC circuit)</li> </ul>	T <sub>A</sub> = -40 to +85°C	
★ Electrical specifications, recommended soldering conditions		Refer to the data sheet for each product.				
Package		20-pin plastic SSOP		<ul style="list-style-type: none"> <li>20-pin plastic SOP</li> <li>20-pin plastic SSOP</li> </ul>	20-pin plastic SSOP	
One-time PROM version		μPD6P4B			μPD6P5	

**APPENDIX C. EXAMPLE OF REMOTE-CONTROL TRANSMISSION FORMAT**  
**(in the case of NEC transmission format in one-shot command transmission mode)**

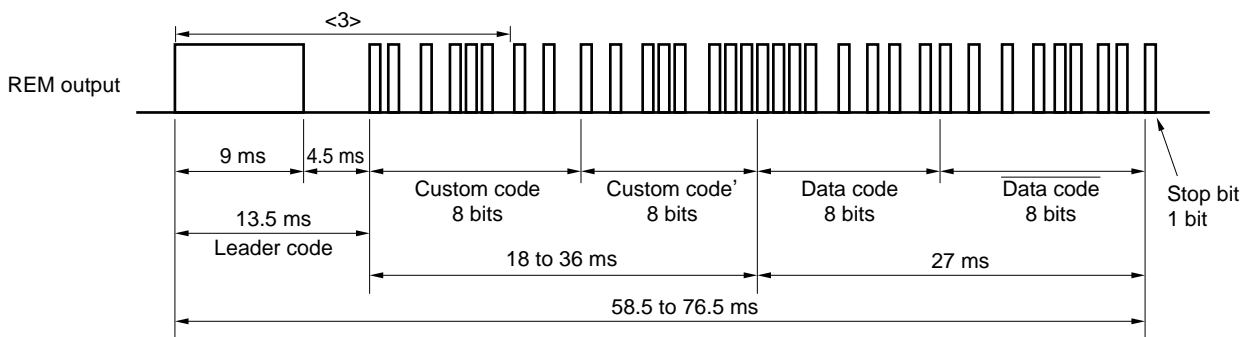
**Caution** When using the NEC transmission format, please apply for a custom code at NEC.

(1) REM output waveform (from <2> on, the output is made only when the key is kept pressed)

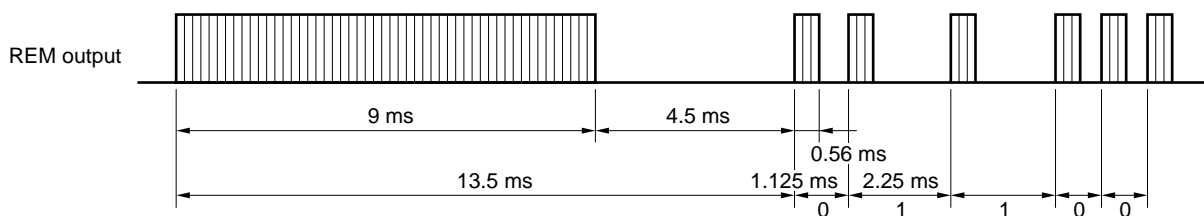


**Remark** If the key is repeatedly pressed, the power consumption of the infrared light-emitting diode (LED) can be reduced by sending the reader code and the stop bit from the second time.

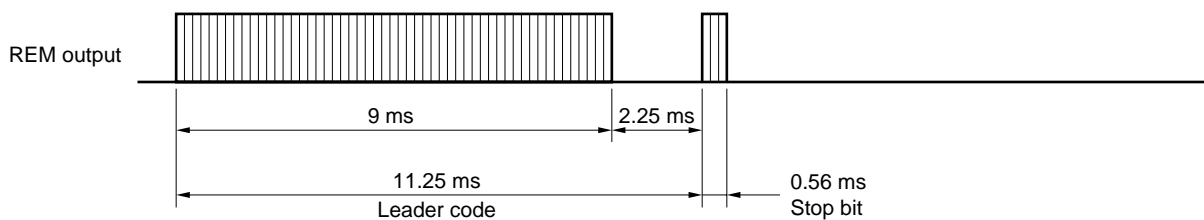
(2) Enlarged waveform of <1>



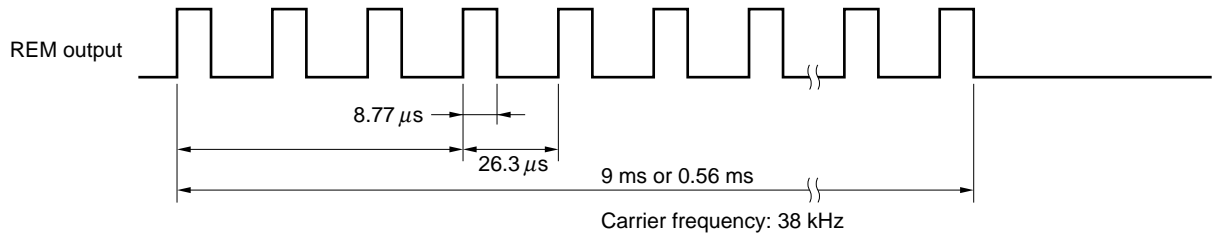
(3) Enlarged waveform of <3>



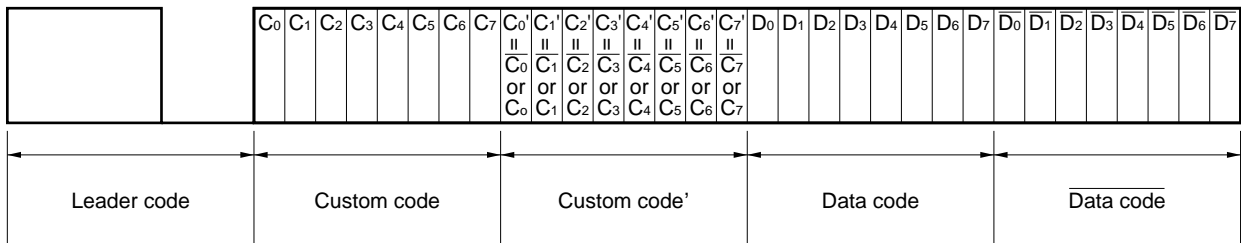
(4) Enlarged waveform of <2>



(5) Carrier waveform (enlarged waveform of each code's high period)



(6) Bit array of each code



**Caution** To prevent malfunction with other systems when receiving data in the NEC transmission format, the total 32 bits of the 16-bit custom codes (Custom code, Custom code') and the 16-bit data codes (Data code, Data code) must not only be fully decoded (make sure to check Data code as well) but also checked to make sure that no signals are present.

[MEMO]

[MEMO]

**NOTES FOR CMOS DEVICES****① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS**

Note:

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

**② HANDLING OF UNUSED INPUT PINS FOR CMOS**

Note:

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

**③ STATUS BEFORE INITIALIZATION OF MOS DEVICES**

Note:

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



## Regional Information

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

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

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

### **NEC Electronics Inc. (U.S.)**

Santa Clara, California  
Tel: 408-588-6000  
800-366-9782  
Fax: 408-588-6130  
800-729-9288

### **NEC Electronics (Germany) GmbH**

Duesseldorf, Germany  
Tel: 0211-65 03 02  
Fax: 0211-65 03 490

### **NEC Electronics (UK) Ltd.**

Milton Keynes, UK  
Tel: 01908-691-133  
Fax: 01908-670-290

### **NEC Electronics Italiana s.r.l.**

Milano, Italy  
Tel: 02-66 75 41  
Fax: 02-66 75 42 99

### **NEC Electronics (Germany) GmbH**

Benelux Office  
Eindhoven, The Netherlands  
Tel: 040-2445845  
Fax: 040-2444580

### **NEC Electronics (France) S.A.**

Velizy-Villacoublay, France  
Tel: 01-30-67 58 00  
Fax: 01-30-67 58 99

### **NEC Electronics (France) S.A.**

Madrid Office  
Madrid, Spain  
Tel: 91-504-2787  
Fax: 91-504-2860

### **NEC Electronics (Germany) GmbH**

Scandinavia Office  
Taeby, Sweden  
Tel: 08-63 80 820  
Fax: 08-63 80 388

### **NEC Electronics Hong Kong Ltd.**

Hong Kong  
Tel: 2886-9318  
Fax: 2886-9022/9044

### **NEC Electronics Hong Kong Ltd.**

Seoul Branch  
Seoul, Korea  
Tel: 02-528-0303  
Fax: 02-528-4411

### **NEC Electronics Singapore Pte. Ltd.**

United Square, Singapore  
Tel: 65-253-8311  
Fax: 65-250-3583

### **NEC Electronics Taiwan Ltd.**

Taipei, Taiwan  
Tel: 02-2719-2377  
Fax: 02-2719-5951

### **NEC do Brasil S.A.**

Electron Devices Division  
Guarulhos-SP Brasil  
Tel: 55-11-6462-6810  
Fax: 55-11-6462-6829

**MS-DOS is either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries.**

**PC/AT and PC DOS are trademarks of International Business Machines 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.

**• The information in this document is current as of September, 2000. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC's data sheets or data books, etc., for the most up-to-date specifications of NEC semiconductor products. Not all products and/or types are available in every country. Please check with an NEC sales representative for availability and additional information.**

• No part of this document may be copied or reproduced in any form or by any means without prior written consent of NEC. NEC assumes no responsibility for any errors that may appear in this document.

• NEC does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC semiconductor products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC or others.

• Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of customer's equipment shall be done under the full responsibility of customer. NEC assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.

• While NEC endeavours to enhance the quality, reliability and safety of NEC semiconductor products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC semiconductor products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment, and anti-failure features.

• NEC semiconductor products are classified into the following three quality grades:

"Standard", "Special" and "Specific". The "Specific" quality grade applies only to semiconductor products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of a semiconductor product depend on its quality grade, as indicated below. Customers must check the quality grade of each semiconductor product before using it in a particular application.

"Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots

"Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support)

"Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

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

(Note)

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

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