DATA SHEET

NEC

MOS INTEGRATED CIRCUIT

μPD67, 67A, 68, 68A, 69

4-BIT SINGLE-CHIP MICROCONTROLLER FOR INFRARED REMOTE CONTROL TRANSMISSION

DESCRIPTION

With their 2.0 V low-voltage operation, carrier generator for infrared remote control transmission, standby release ★ function through key input, and programmable timer, the μPD67, 67A, 68, 68A, and 69 are ideal for infrared remote control transmitters.

A one-time PROM product, the μPD6P9, has also been provided for the μPD67, 67A, 68, 68A, and 69 for program evaluation or small-quantity production.

FEATURES

- Program memory (ROM)
  - μPD67, 67A: 1,002 × 10 bits
  - μPD68, 68A: 2,026 × 10 bits
  - μPD69: 4,074 × 10 bits
- Data memory (RAM)
  - μPD67, 67A, 68, 68A: 32 × 4 bits
  - μPD69: 128 × 4 bits
- On-chip carrier generator for infrared remote control: Each high-/low-level width can be set from 250 ns to 64 μs (@ f_X = 4 MHz operation) via modulo registers
- 9-bit programmable timer: 1 channel
- Instruction execution time: 16 μs (@ f_X = 4 MHz operation: ceramic oscillation)
- Stack level: 1 level (Stack RAM is multiplexed with data memory RF.)
- I/O pins ($K_{I/O}$): 8
- Input pins (K_I): 4
- Sense input pins (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°C
- Oscillator frequency: f_X = 3.5 to 4.5 MHz
- On-chip POC circuit and RAM retention detector
- Capacitor for oscillator: 15 pF (mask option)

APPLICATIONS

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

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.

Document No. U14935EJ2V0DS00 (2nd edition)
Date Published May 2002 N CP(K)
Printed in Japan

The mark ★ shows major revised points.

ORDERING INFORMATION

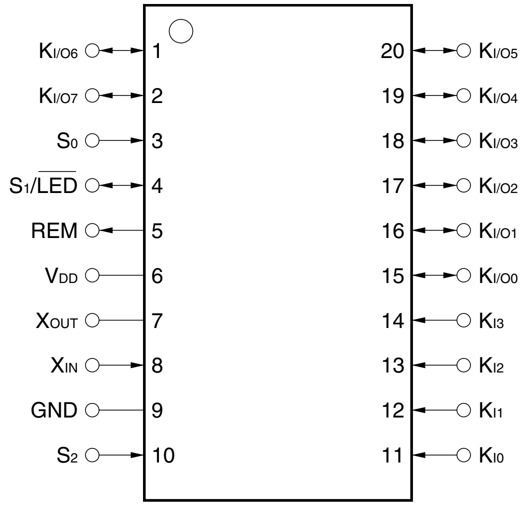
	Part Number	Package
	μPD67MC-×××-5A4	20-pin plastic SSOP (7.62 mm (300))
★	μPD67AMC-×××-5A4	20-pin plastic SSOP (7.62 mm (300))
	μPD68MC-×××-5A4	20-pin plastic SSOP (7.62 mm (300))
★	μPD68AMC-×××-5A4	20-pin plastic SSOP (7.62 mm (300))
	μPD69MC-×××-5A4	20-pin plastic SSOP (7.62 mm (300))

Remark ××× indicates ROM code suffix.

PIN CONFIGURATION (TOP VIEW)

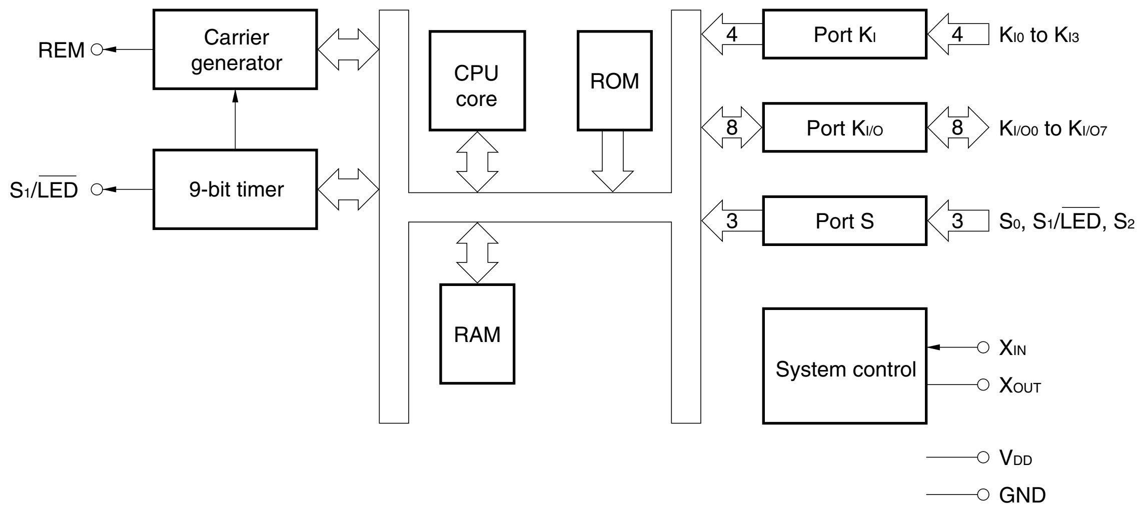
20-pin Plastic SSOP (7.62 mm (300))

- μPD67MC-×××-5A4
- ★ μPD67AMC-×××-5A4
- μPD68MC-×××-5A4
- ★ μPD68AMC-×××-5A4
- μPD69MC-×××-5A4



Caution The pin numbers of K_I and $K_{I/O}$ are in the reverse order of those in the μPD6600A, and 6124A.

BLOCK DIAGRAM



LIST OF FUNCTIONS

★

Item	μPD67, 67A	μPD68, 68A	μPD69	μPD6P9
ROM capacity	1,002 × 10 bits	2,026 × 10 bits	4,074 × 10 bits	
	Mask ROM			One-time PROM
RAM capacity	32 × 4 bits		128 × 4 bits	
Stack	1 level (multiplexed with RF of RAM)			
I/O pins	• Key input (K_I): 4 • Key I/O ($K_{I/O}$): 8 • Key extended input (S_0, S_1, S_2): 3 • Remote control transmission display output ($\overline{LED}$): 1 (multiplexed with S_1 pin)			
Number of keys	• 32 • 56 (when extended by key extension input)			
Clock frequency	Ceramic oscillation • f_X = 3.5 to 4.5 MHz			
Instruction execution time	16 μs (@ f_X = 4 MHz)			
Carrier frequency	Each high-/low-level width can be set from 250 ns to 64 μs (@ f_X = 4 MHz operation) via modulo registers			
Timer Note	9-bit programmable timer: 1 channel, timer clock: f_X/64			
POC circuit	On-chip			
RAM retention detector	On-chip			
Capacitor for oscillation (15 pF)	Mask option			Set to be used/ not used in device
Supply voltage	V_{DD} = 2.0 to 3.6 V			V_{DD} = 2.2 to 3.6 V
Operating ambient temperature	T_A = –40 to +85°C			
Package	20-pin plastic SSOP (7.62 mm (300))			

★ **Note** The timer output time differs between the μPD67, 68, and 69 and the μPD67A and 68A. For details, refer to **4 TIMER**.

CONTENTS

1. PIN FUNCTIONS

1.1 List of Pin Functions

Pin No.	Symbol	Function	Output Format	After Reset
1 2 15 to 20	$K_{I/O0}$ to $K_{I/O7}$	8-bit I/O port. Input/output can be specified in 8-bit units. In input mode, the use of a pull-down resistor can be specified. In output mode, these pins can be used as key scan outputs from a key matrix.	CMOS push-pull[Note 1]	High-level output
3	S_0	Input port. Can also be used as a key return input from a key matrix. In input mode, the 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 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 a key matrix. The 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 REM output, this pin outputs a low level from the $\overline{LED}$ output in synchronization with the REM signal.	CMOS push-pull	High-level output ($\overline{LED}$)
5	REM	Infrared remote control transmission output. This output is active high. Each carrier high-/low-level width can be freely set in a range of 250 ns to 64 μs (@ f_X = 4 MHz) by software.	CMOS push-pull	Low-level output
6	V_{DD}	Power supply	—	—
7 8	X_{OUT} X_{IN}	Pins for connecting ceramic resonators for the system clock. A capacitor (15 pF) for the oscillator can be specified by a mask option.	—	Low level (oscillation stopped)
9	GND	GND	—	—
10	S_2	Input port. The use of STOP mode release for the S_2 port can be specified by software. When used as a key input from a key matrix, enable the use of STOP mode release (at this time, a pull-down resistor is connected internally.) When 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}[Note 2]	4-bit input port. These pins can also be used as a key return inputs from a key matrix. The use of a pull-down resistor can be specified by software in 4-bit units.	—	Input (low-level)

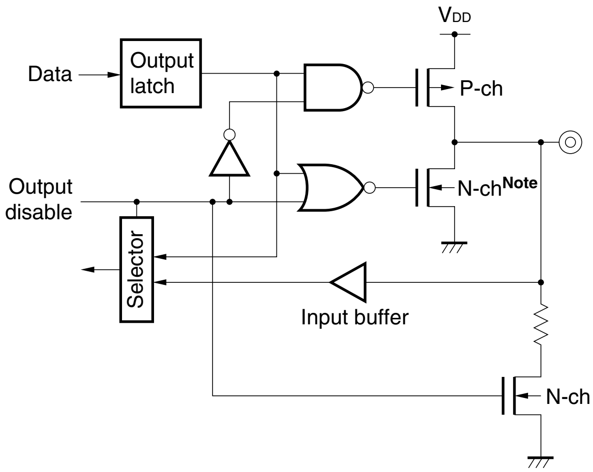
Notes 1. Be careful about this because the drive capacity of the low-level output side is held low.

2. In order to prevent malfunction, be sure to input a low level to one or more of pins K_{I0} to K_{I3} when POC is released by supply voltage rising (Can be left open. When open, leave the pull-down resistor connected).

1.2 Pin I/O Circuits

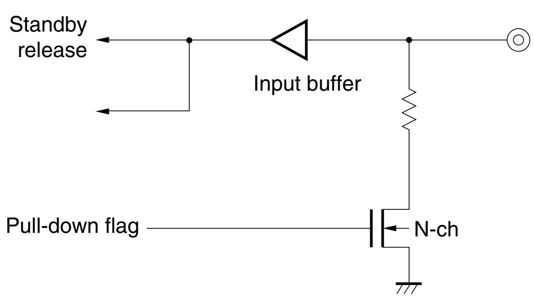
The I/O circuits of pins of the μPD67, 67A, 68, 68A, and 69 are shown in partially simplified forms below.

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

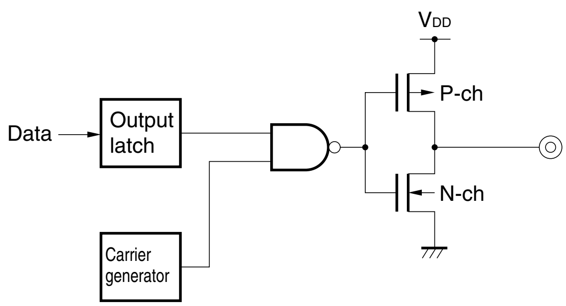


Note The drive capacity is held low.

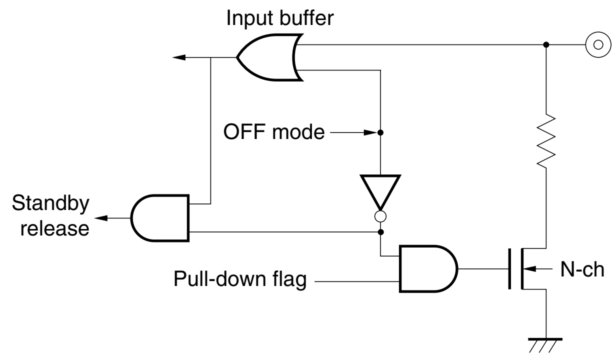
(2) K_{I0} to K_{I3}



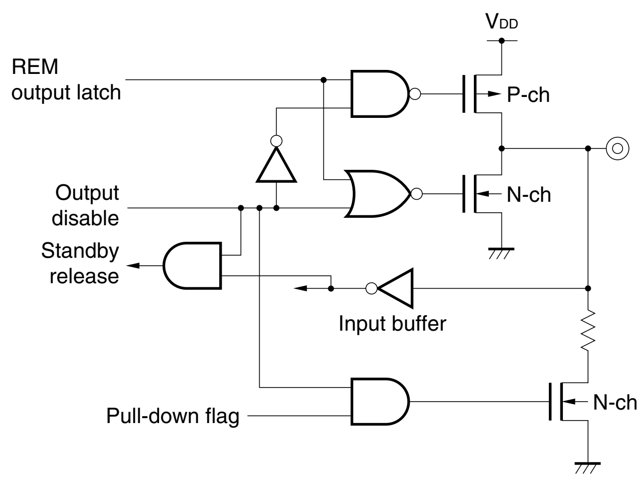
(3) REM



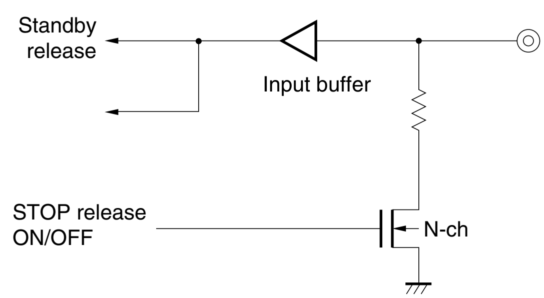
(4) S_0



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



(6) S_2



1.3 Connection of Unused Pins

The following connections are recommended for unused pins.

Table 1-1. Connection of Unused Pins

Pin		Connection: Inside the Microcontroller	Connection: Outside the Microcontroller
$K_{I/O}$	Input mode	—	Leave open.
	Output mode	High-level output	
REM		—	
$S_1/\overline{LED}$		Output mode ($\overline{LED}$) setting	
S_0		OFF mode setting	Directly connect to GND.
S_2		—	
K_1		—	

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 (μPD67, 67A, 68, 68A)
12 Bits (μPD69)

The program counter (PC) is a binary counter that holds the address information of the program memory.

Figure 2-1. Program Counter Configuration

μPD69: PC11 to PC0
μPD67, 67A, 68, 68A: PC10 to PC0

PC	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0

The PC 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 PC 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 in the address stack register (ASR). If the return instruction (RET) is executed after the CALL instruction is executed, the address saved in the ASR is restored to the PC.

After reset, the value of the PC 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 and decremented when the return instruction (RET) is executed.

When reset, the stack pointer contents are cleared to 0.

When the stack pointer overflows (stack level 2 or more) or underflows, the CPU is defined as hung up, a system reset signal is generated, and the PC becomes 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 (μPD67, 67A, 68, 68A)
12 Bits (μPD69)

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

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

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

Caution If RF is accessed as the data memory, the higher 3 bits of the μPD67, 67A, 68, and 68A, and higher 4 bits of the μPD69 become undefined.

Figure 2-2. Address Stack Register Configuration

RF: ASR7 to ASR0

ASR	ASR11	ASR10	ASR9	ASR8	ASR7	ASR6	ASR5	ASR4	ASR3	ASR2	ASR1	ASR0

μPD67, 67A, 68, 68A: ASR10 to ASR0
μPD69: ASR11 to ASR0

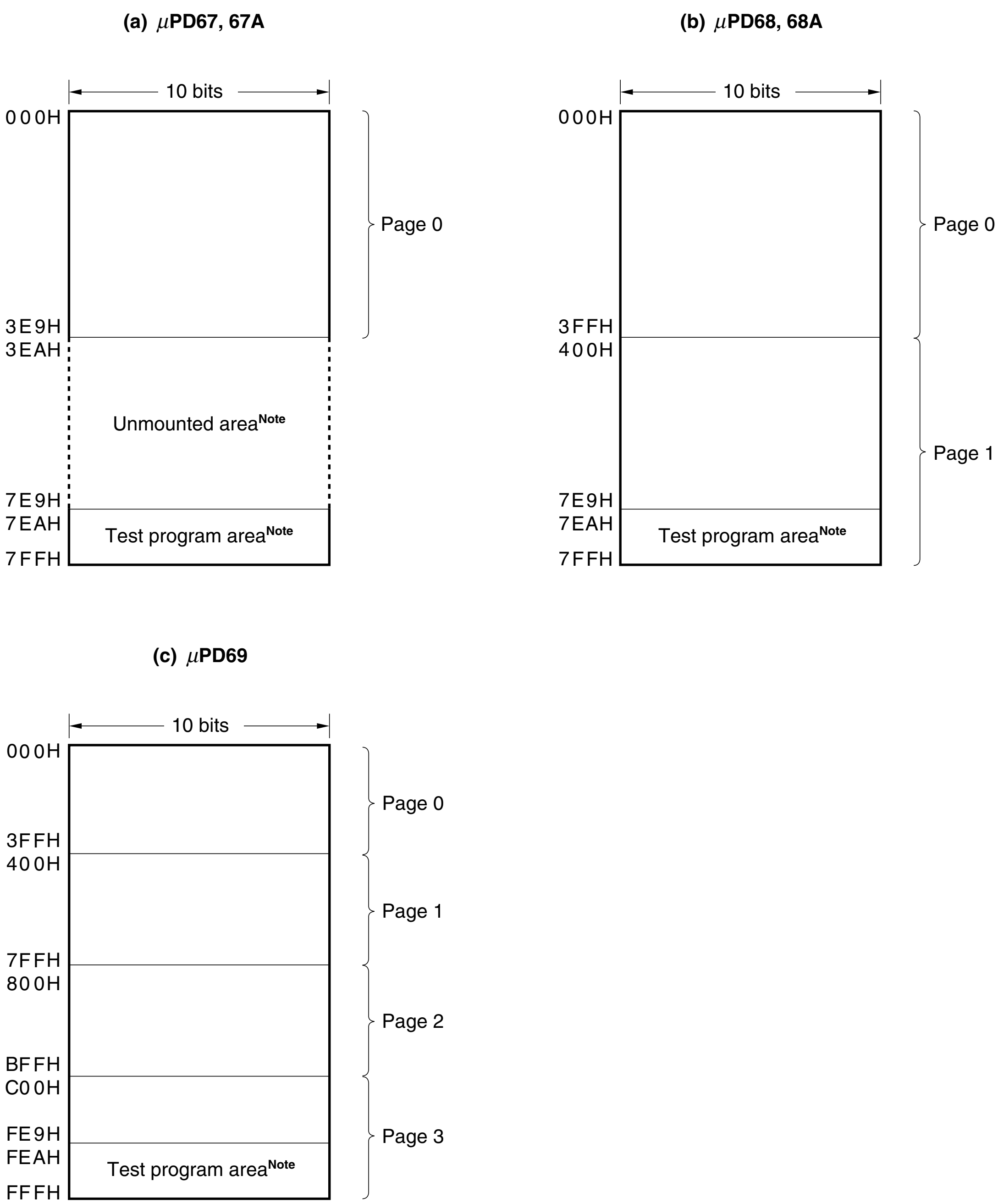
2.4 Program Memory (ROM): 1,002 Steps × 10 Bits (μPD67, 67A)
2,026 Steps × 10 Bits (μPD68, 68A)
4,074 Steps × 10 Bits (μPD69)

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 of the μPD67, 67A, 68, and 68A, and FEAH to FFFH of the μPD69 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.

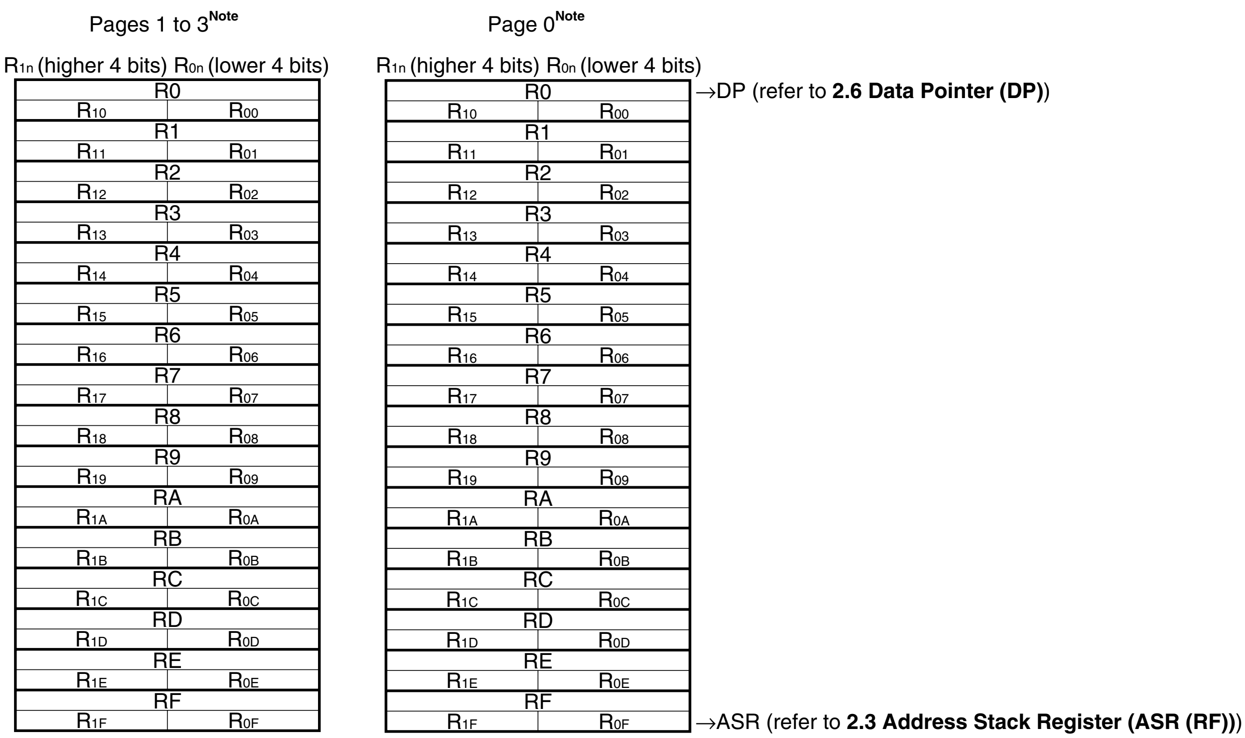
2.5 Data Memory (RAM): 32 × 4 Bits (μPD67, 67A, 68, 68A)
128 × 4 Bits (μPD69)

The data memory, which is a static RAM consisting of 32 × 4 bits, is used to retain 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 retain the previous data (undefined when turning on the power).

Figure 2-4. Data Memory Configuration



Note μPD67, 67A, 68, 68A: Page 0
μPD69: Pages 0 to 3 (pages can be switched using bits 0 and 1 of control register 0)

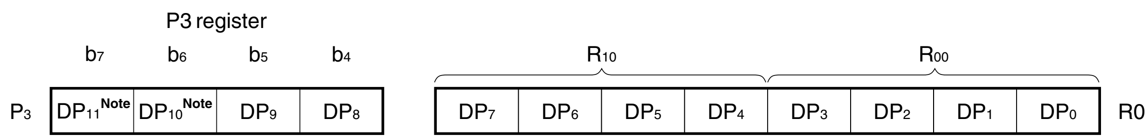
2.6 Data Pointer (DP): 12 Bits

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 4 bits by bits 4 to 7 of the P3 register (CR0).

After reset, the pointer contents become 000H.

Figure 2-5. Data Pointer Configuration



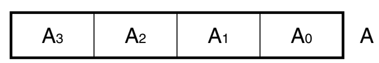
Note Set DP_{10} and DP_{11} to 0 in the case of the μPD67 and 67A, and set DP_{10} to 0 in the case of the μPD68 and 68A.

2.7 Accumulator (A): 4 Bits

The accumulator, which refers to 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 refers to an arithmetic circuit consisting of 4 bits, executes simple (mainly 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 (to 1) in the following cases.

- If the condition specified with the operand is met when the STTS instruction is 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 the standby mode.)

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

- If the condition specified with the operand is not met when the STTS instruction is executed.
- When the status flag has been set (to 1), the HALT instruction executed, but the release condition is not met at the point of executing the HALT instruction. (In this case, the system does not enter the 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_3	b_2	b_1	b_0	
0	0	0	0	High level is input to at least one of K_I pins.
	0	1	1	High level is input to at least one of K_I pins.
	1	1	0	High level is input to at least one of K_I pins.
	1	0	1	The down counter of the timer is 0.
1	Either of the combinations of b_2, b_1, and b_0 above.			[The following condition is added in addition to the above.] High level is input to at least one of S_0**Note 1**, S_1**Note 1**, or S_2**Note 2** pins.

Notes 1. The S_0 and S_1 pins must be set to input mode (bit 2 and bit 0 of the P4 register are set to 0 and 1, respectively).

2. The use of STOP mode release for the S_2 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 (to 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 (to 0) in the following cases:

- If the ANL instruction or the XRL instruction is executed when at least either bit 3 of the accumulator or bit 3 of the operand 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 $K_{I/O}$ port, the K_I port, the special ports (S_0, $S_1/\overline{LED}$, S_2), and the control registers are treated as port registers. After reset, the port register values are as shown below.

Figure 3-1. Port Register Configuration

Port register								After reset
P0								FFH
P_{10}				P_{00}				
$K_{I/O7}$	$K_{I/O6}$	$K_{I/O5}$	$K_{I/O4}$	$K_{I/O3}$	$K_{I/O2}$	$K_{I/O1}$	$K_{I/O0}$	
P1								××××11×1B[Note 1]
P_{11}				P_{01}				
K_{I3}	K_{I2}	K_{I1}	K_{I0}	$S_1/\overline{LED}$	S_0	S_2	1	
P3 (control register 0)								0000×000B[Note 2]
P_{13}				P_{03}				
DP_{11}	DP_{10}	DP_9	DP_8	RAM retention flag	–	ID1	ID0	
P4 (control register 1)								26H
P_{14}				P_{04}				
0	0	K_I Pull-down	S_0/S_1 Pull-down	S_2 STOP release	$S_1/\overline{LED}$ mode	$K_{I/O}$ mode	S_0 mode	

Notes 1. ×: Refers to the value based on the K_I and S_2 pin state.

2. ×: Refers to the value based on decrease of power supply voltage (0 when $V_{DD} \leq V_{ID}$)

Remark V_{ID}: RAM retention detection voltage

Table 3-1. Relationship Between Ports and Reading/Writing

Port Name	Input Mode		Output Mode	
	Read	Write	Read	Write
$K_{I/O}$	Pin state	Output latch	Output latch	Output latch
K_I	Pin state	—	—	—
S_0	Pin state	—	**Note**	—
$S_1/\overline{LED}$	Pin state	—	Pin state	—
S_2	Pin state	—	—	—

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

3.1 $K_{I/O}$ Port (P0)

The $K_{I/O}$ port is an 8-bit I/O port for key scan output.

I/O 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 is placed in output mode and the value of the output latch (P0) becomes 1111 1111B.

The $K_{I/O}$ port incorporates a pull-down resistor, allowing pull-down in input mode only.

Caution When a key is double-pressed, a high-level output and a low-level output may conflict at the $K_{I/O}$ port. To avoid this, the low-level output current of the $K_{I/O}$ port is held low. Therefore, be careful when using the $K_{I/O}$ port for purposes other than key scan output.
The $K_{I/O}$ port is designed so that even when connected directly to V_{DD} within the normal supply voltage range (V_{DD} = 2.0 to 3.6 V), no problem occurs.

Table 3-2. $K_{I/O}$ Port (P0)

Bit	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
Name	$K_{I/O7}$	$K_{I/O6}$	$K_{I/O5}$	$K_{I/O4}$	$K_{I/O3}$	$K_{I/O2}$	$K_{I/O1}$	$K_{I/O0}$

b_0 to b_7: When reading: In input mode, the $K_{I/O}$ pin's state is read.
In output mode, the $K_{I/O}$ pin's output latch contents are read.
When writing: Data is written to the $K_{I/O}$ pin's output latch regardless of input or output mode.

3.2 K_I Port/Special Ports (P1)

3.2.1 K_I port (P_{11}: bits 4 to 7 of P1)

The K_I port is a 4-bit input port for key input. The pin state can be read.

The use of a pull-down resistor for the K_I 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_I/Special Port Register (P1)

Bit	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
Name	K_{I3}	K_{I2}	K_{I1}	K_{I0}	$S_1/\overline{LED}$	S_0	S_2	Fixed to "1"

b_1: The state of the S_2 pin is read (read only).
b_2: In input mode, state of the S_0 pin is read (read only).
In OFF mode, this bit is fixed to 1.
b_3: The state of the $S_1/\overline{LED}$ pin is read regardless of input/output mode (read only).
b_4 to b_7: The state of the K_I pin is read (read only).

Caution In order to prevent malfunction, be sure to input a low level to one or more of pins K_{I0} to K_{I3} when POC is released by supply voltage rising (Can be left open. When open, leave the pull-down resistor connected).

3.2.2 S_0 port (bit 2 of P1)

The S_0 port is an input/OFF mode port.

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

In input mode, the use of a pull-down resistor for the S_0 and $S_1/\overline{LED}$ port 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 is configured so that through current does not flow internally. In OFF mode, 1 can be read regardless of the pin state.

After reset, S_0 is set to OFF mode, thus becoming high-impedance.

3.2.3 $S_1/\overline{LED}$ port (bit 3 of P1)

The $S_1/\overline{LED}$ port is an I/O port.

Input or output mode can be set using bit 2 of the P4 resister. The pin state can be read in both input mode and output mode.

When in input mode, the use of a pull-down resistor for the S_0 and $S_1/\overline{LED}$ ports can be specified in 2-bit units by software using bit 4 of the P4 register.

When 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, $S_1/\overline{LED}$ is placed in output mode, and a high level is output.

3.2.4 S_2 port (bit 1 of P1)

The S_2 port is an input port.

Use of STOP mode release for the S_2 port can be specified by bit 3 of the P4 register.

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

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

After reset, S_2 is set to input mode where 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 ×000B[Note].

Note ×: Refers to the value based on a decrease of power supply voltage (0 when $V_{DD} \le V_{ID}$)

Remark V_{ID}: RAM retention detection voltage

Table 3-4. Control Register 0 (P3)

(1) μPD67, 67A, 68, 68A

Bit		b_7[Note]	b_6[Note]	b_5	b_4	b_3	b_2	b_1	b_0
Name		DP (Data Pointer)				RAM retention flag	—		ID0
		DP_{11}	DP_{10}	DP_9	DP_8				
Setting	0	0	0	0	0	Not retainable	Fixed to 0		
	1	1	1	1	1	Retainable			
After reset		0	0	0	0	×	0	0	0

(2) μPD69

Bit		b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
Name		DP (Data Pointer)				RAM retention flag	—	ID1	ID0
		DP_{11}	DP_{10}	DP_9	DP_8				
Setting	0	0	0	0	0	Not retainable	Fixed to 0	Specification of PAGE0 to PAGE3	
	1	1	1	1	1	Retainable			
After reset		0	0	0	0	×	0	0	0

b_0, b_1: Specify RAM pages 0 to 3 (μPD69 only). Fixed to 0 in the μPD67, 67A, 68, and 68A.

ID1	ID0	RAM
0	0	Page 0
0	1	Page 1
1	0	Page 2
1	1	Page 3

b_3: RAM retention flag. For function details, refer to **3.3.1 RAM retention flag (bit 3 of P3)**.

b_4 to b_7: Specify the higher bits of the ROM data pointer (DP_8 to DP_{11}).

Note Set b_7 and b_6 to 0 in the case of the μPD67 and 67A, and set b_7 to 0 in the case of the μPD68 and 68A.

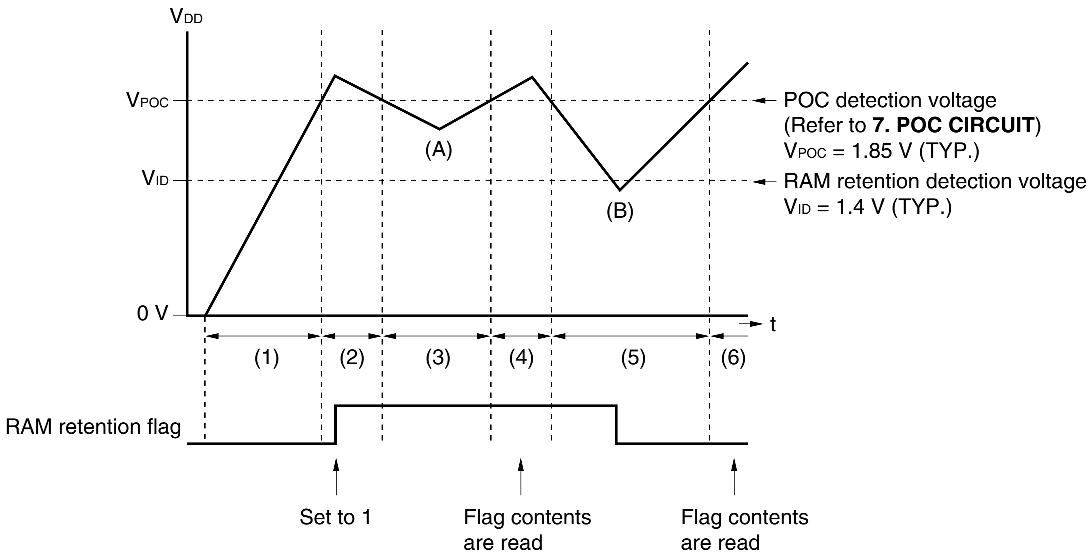
3.3.1 RAM retention flag (bit 3 of P3)

The RAM retention flag indicates whether the supply voltage has fallen below the level at which the contents of the RAM are lost while the battery is being exchanged or when the battery voltage has dropped.

This flag is at bit 3 of control register 0 (P3).

It is cleared to 0 if the supply voltage drops below the RAM retention detection voltage (approx. 1.4 V TYP.). If this flag is 0, it can be judged that the RAM contents have been lost or that power has just been applied. This flag can be used to initialize the RAM via software. After initializing the RAM and writing the necessary data to it, set this RAM retention flag to 1 by software. At this time, 1 means that data has been set to the RAM.

Figure 3-2. Supply Voltage Transition and Detection Voltage



(1) If the supply voltage rises after the battery has been set, and exceeds V_{POC} (POC detection voltage), reset is cleared. Because the supply voltage rises from 0 V, which is lower than V_{ID} (RAM retention detection voltage), the RAM retention flag remains in the initial status 0.

(2) The supply voltage has now risen to the level at which the device can operate. Write the necessary data to the RAM and set the RAM retention flag to 1.

(3) The device is reset if the supply voltage drops below V_{POC}. At point (A) in the above figure, the RAM retention flag remains 1 because the supply voltage is higher than V_{ID} at this point.

(4) If the RAM retention flag is checked by software after reset has been cleared, it is 1. This means that the contents of the RAM have not been lost. It is therefore not necessary to initialize the RAM by software.

(5) The device is reset if the supply voltage drops below V_{POC}. At point (B) in the figure, the voltage is lower than V_{ID}. Consequently, the RAM retention flag is cleared to 0.

(6) If the RAM retention flag is checked by software after reset has been cleared, it is 0. This means that the contents of the RAM may have been lost. If this case, initialize the RAM by software.

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 3-5. Control Register 1 (P4)

Bit		b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
Name		—	—	K_I Pull-down	S_0/S_1 Pull-down	S_2 STOP release	$S_1/\overline{LED}$ mode	$K_{I/O}$ mode	S_0 mode
Setting	0	Fixed to 0	Fixed to 0	OFF	OFF	Disable	S_1	IN	OFF
	1			ON	ON	Enable	$\overline{LED}$	OUT	IN
After reset		0	0	1	0	0	1	1	0

b_0: Specifies the input mode of the S_0 port. 0 = OFF mode (high impedance); 1 = IN (input mode).

b_1: Specifies the I/O mode of the $K_{I/O}$ port.
0 = IN (input mode); 1 = OUT (output mode).

b_2: Specifies the I/O mode of the $S_1/\overline{LED}$ port. 0 = S_1 (input mode); 1 = $\overline{LED}$ (output mode).

b_3: Specified the use of STOP mode release by S_2 port (with/without pull-down resistor). 0 = disable (without pull-down); 1 = enable (with pull-down).

b_4: Specifies the use of a pull-down resistor in S_0/S_1 port input mode. 0 = OFF (not used);
1 = ON (used)

b_5: Specifies the use of a pull-down resistor for the K_I port. 0 = OFF (not used);
1 = ON (used).

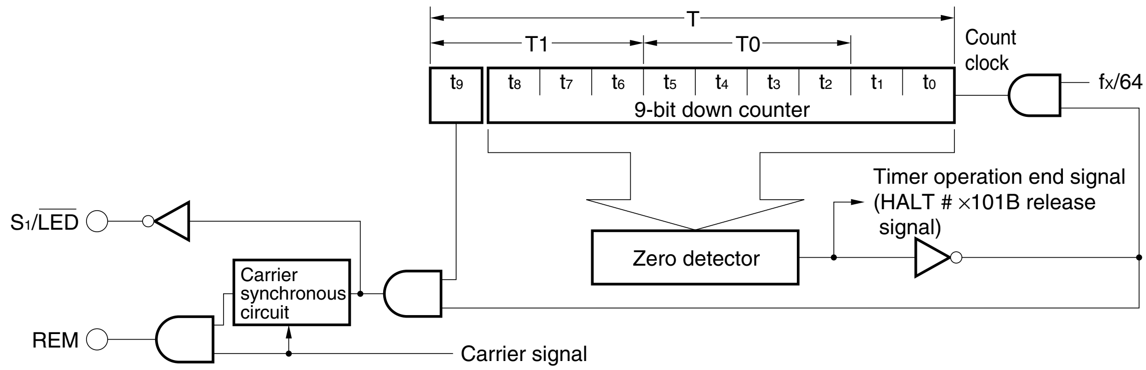
Remark In output mode or in OFF mode, all the pull-down resistors are automatically disconnected.

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) permitting 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 manipulation instruction. The timer manipulation 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. 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 #×101B) waiting for the timer to stop its 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 relational expression applies between the timer's output time and the down counter's set value.

★ (a) μPD67, 68, and 69
Timer output time = (Set value + 1) × 64/fx

(b) μPD67A and 68A
Timer output time = (Set value + 1) × 64/fx – 4/fx

In addition, when the timer is set successively, in the μPD67A and 68A, the timer output time is also 4/fx shorter than the total time. An example is shown below.

Example When fx = 4 MHz

```
MOV T, #3FFH
STTS #05H
HALT #05H
MOV T, #232H
STTS #05H
HALT #05H
```

In the case above, the timer output time is as follows.

(a) μPD67, 68, and 69
(Set value + 1) × 64/fx + (Set value + 1) × 64/fx
= (511 + 1) × 64/4 + (50 + 1) × 64/4
= 9.008 ms

(b) μPD67A and 68A
(Set value + 1) × 64/fx + (Set value + 1) × 64/fx – 4/fx
= (511 +1) × 64/4 + (50 +1) × 64/4 – 4/4
= 9.007 ms

By setting the flag (t_9) that enables the timer output to 1, the timer can output its operation status from the $S_1/\overline{LED}$ pin and the REM pin. The REM pin can also output the carrier while the timer is in operation.

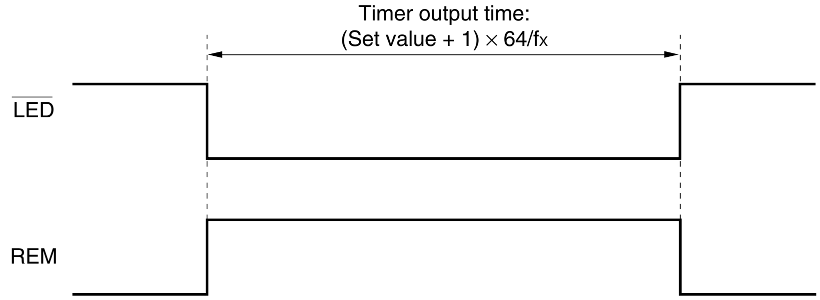
Table 4-1. Timer Output (at t_9 = 1)

	$S_1/\overline{LED}$ Pin	REM Pin
Timer operating	Low level	High level (or carrier output[Note])
Timer halting	High level	Low level

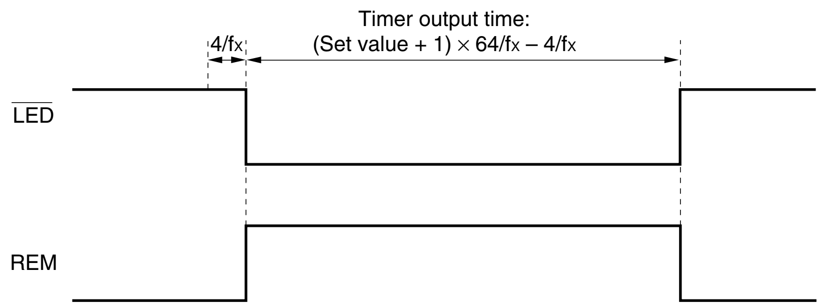
Note The carrier output results if bit 9 (CARY) of the high-level period setting modulo register (MOD1) is cleared (to 0).

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

(a) μPD67, 68, and 69



(b) μPD67A and 68A

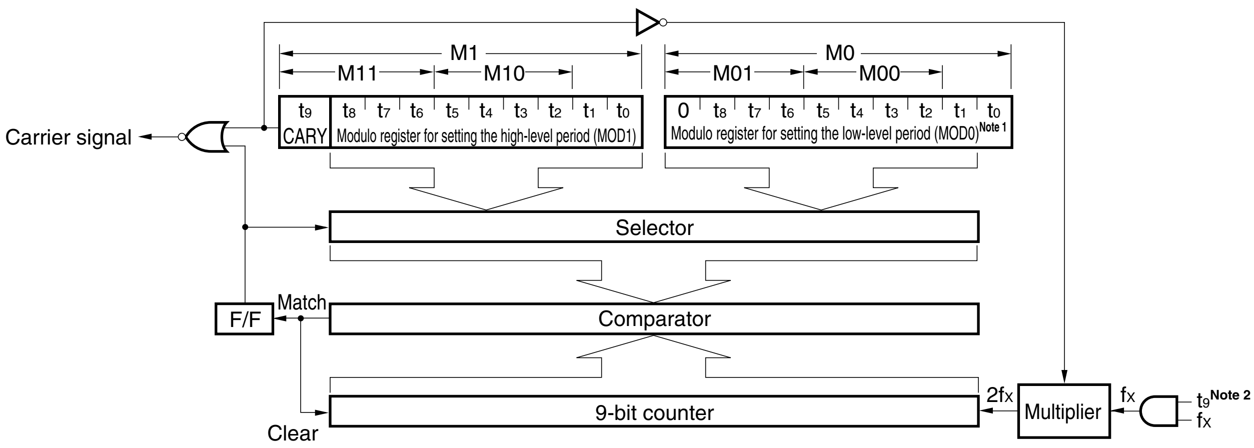


4.3 Carrier Output

4.3.1 Carrier output generator

The carrier generator consists of a 9-bit counter and two modulo registers for setting the high- and low-level periods (MOD1 and MOD0 respectively).

Figure 4-3. Configuration of Remote Controller Carrier Generator



Notes 1. Bit 9 of the modulo register for setting the low-level period (MOD0) is fixed to 0.

2. t9: Flag that enables timer output (timer block) (see **Figure 4-1 Timer Configuration**)

The carrier duty ratio and carrier frequency can be determined by setting the high- and low-level widths using the respective modulo registers. Each of these widths can be set in a range of 250 ns to 64 μs (@ f_X = 4 MHz).

The system clock multiplied by 2 is used for the 9-bit counter input (8 MHz when f_X = 4 MHz). MOD0 and MOD1 are read and written using timer manipulation instructions.

MOV A, M00	MOV M00, A	MOV M0, #data10
MOV A, M01	MOV M01, A	MOV M1, #data10
MOV A, M10	MOV M10, A	MOV M0, @R0
MOV A, M11	MOV M11, A	MOV M1, @R0

The values of MOD0 and MOD1 can be calculated from the following expressions.

$$MOD0 = (2 \times f_X \times (1 - D) \times T) - 1$$
$$MOD1 = (2 \times f_X \times D \times T) - 1$$

Caution Be sure to input values in range of 001H to 1FFH to MOD0 and MOD1.

Remark D: Carrier duty ratio (0 < D < 1)
f_X: Input clock (MHz)
T: Carrier cycle (μs)

4.3.2 Carrier output control

Remote controller carrier can be output from the REM pin by clearing (0) bit 9 (CARY) of the modulo register for setting the high-level period (MOD1).

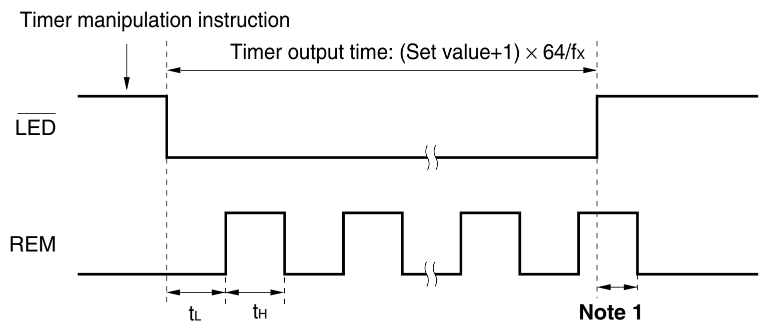
When performing carrier output, be sure to set the timer operation after setting the MOD0 and MOD1 values. Note that a malfunction may occur if the values of MOD0 and MOD1 are changed while carrier is being output from the REM pin.

Executing the timer manipulation instruction starts the carrier output from the low level.

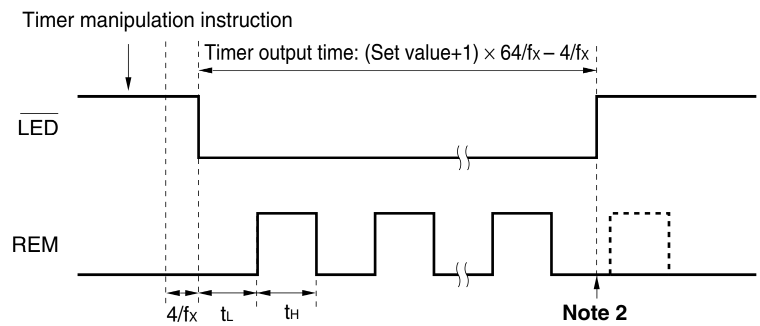
If the timer's down counter reaches 0 during carrier output, carrier output is stopped and the REM pin becomes low level. If the down counter reaches 0 while the carrier output is high level, carrier output will stop after first becoming low level following the set period of high level.

★ **Figure 4-4. Timer Output (When Carrier Is Output)**

(a) μPD67, 68, and 69



(b) μPD67A and 68A



Notes 1. If the down counter reaches 0 while the carrier output is high level, carrier output will stop after becoming low level.

2. As shown in figure (b) above, in the μPD67A and 68A, because the timer output time is 4/fx shorter (1 μs: fx = 4 MHz) than in the μPD67, 68, and 69, the down counter reaches 0 while the carrier output is low level, so the carrier may be one clock shorter than in the μPD67, 68, and 69.

Output from the REM pin is as follows, in accordance with the values set to bit 9 (CARY) of MOD1 and the timer output enable flag (t_9), and the value of the timer block's 9-bit down counter (t_0 to t_8).

Table 4-2. REM Pin Output

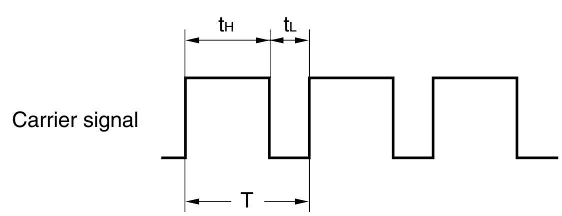
MOD1 Bit 9 (CARY)	Timer Output Enable Flag (Timer Block t_9)	9-Bit Down Counter (Timer Block t_0 to t_8)	REM Pin
—	—	0	Low-level output
—	0	Other than 0	
0	1		Carrier output[Note]
1			High-level output

Note Input values in the range of 001H to 1FFH to MOD0 and MOD1.

Caution MOD0 and MOD1 must be set while the REM pin is low level (t_9 = 0 or t_0 to t_8 = 0).

Table 4-3. Example of Carrier Frequency Settings (f_X = 4 MHz)

Setting Value		t_H (μs)	t_L (μs)	T (μs)	f_C (kHz)	Duty
MOD1	MOD0					
01H	01H	0.25	0.25	0.5	2,000	1/2
07H	0BH	1.0	1.5	2.5	400	2/5
13H	13H	2.5	2.5	5.0	200	1/2
27H	27H	5.0	5.0	10	100	1/2
41H	41H	8.25	8.25	16.5	60.6	1/2
41H	85H	8.25	16.75	25	40	1/3
45H	89H	8.75	17.25	26.0	38.5	1/3
45H	8BH	8.75	17.5	26.25	38.10	1/3
45H	8CH	8.75	17.625	26.375	37.9	1/3
47H	91H	9.0	18.25	27.25	36.7	1/3
48H	94H	9.125	18.625	27.75	36.0	1/3
69H	D5H	13.25	26.75	40.0	25	1/3
77H	77H	15.0	15.0	30.0	33.3	1/2
C7H	C7H	25.0	25.0	50.0	20	1/2
FFH	FFH	32.0	32.0	64.0	15.6	1/2



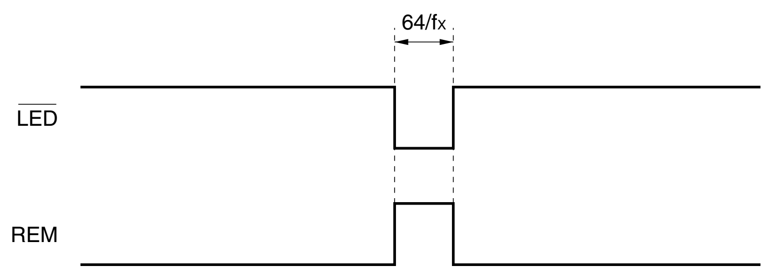
★ 4.4 Software Control of Timer Output

The timer output can be controlled by software. As shown in Figure 4-5, a pulse with a minimum width of 1 instruction cycle (64/fx) can be output in the μPD67, 68, and 69, and a pulse with a minimum width of 64/fx – 4/fx can be output in the μPD67A and 68A.

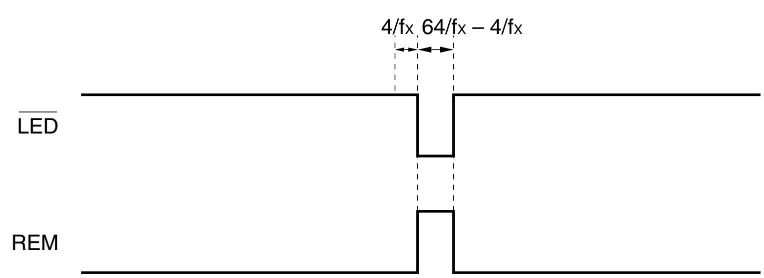
Figure 4-5. Output of Pulse of 1-Instruction Cycle Width

```
        ⋮
MOV T, #0000000000B; low-level output from the REM pin
        ⋮
MOV T, #1000000000B; high-level output from the REM pin
MOV T, #0000000000B; low-level output from the REM pin
        ⋮
```

(a) μPD67, 68, and 69



(b) μPD67A and 68A



5. STANDBY FUNCTION

5.1 Outline of Standby Function

To save current consumption, two types of standby modes, i.e., HALT mode and STOP mode, have been provided available.

In STOP mode, the system clock stops oscillation. At this time, the X_{IN} and X_{OUT} pins are fixed to 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 $\overline{LED}$ output) operates.

In either STOP mode or HALT mode, the statuses of the data memory, accumulator, and port registers, 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 During 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 is not placed in 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 (to 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 #$b_3b_2b_1b_0$B 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 (to 0).

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

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

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

Caution Depending on the status of the status flag (F), the HALT instruction may not be executed. Be careful about this. 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 (to 1) and thus sometimes performs an unintended operation. In this case, the intended operation can be realized by executing the STTS instruction immediately after setting the timer to clear (to 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	Address 0
Release condition shown in Table 5-3	The address following the HALT instruction

Table 5-3. Standby Mode Setup (HALT #$b_3b_2b_1b_0$B) and Release Conditions

Operand Value of HALT Instruction				Setting Mode	Precondition for Setup	Release Condition
b_3	b_2	b_1	b_0			
0	0	0	0	STOP	All $K_{I/O}$ pins are high-level output.	High level is input to at least one of K_I pins.
	0	1	1	STOP	All $K_{I/O}$ pins are high-level output.	High level is input to at least one of K_I pins.
	1	1	0	STOP[Note 1]	The $K_{I/O0}$ pin is high-level output.	High level is input to at least one of K_I pins.
1	Any of the combinations of $b_2b_1b_0$ above			STOP	[The following condition is added in addition to the above.]	
					—	High level is input to at least one of S_0, S_1 and S_2 pins[Note 2].
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 $K_{I/O0}$ pin and the K_I pin so that the standby mode can be released.

2. At least one of the S_0, S_1 and S_2 pins (the pin used for releasing the standby mode) must be specified as follows:

S_0, S_1 pins: Input mode (specified by bits 0 and 2 of the P4 register)

S_2 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.

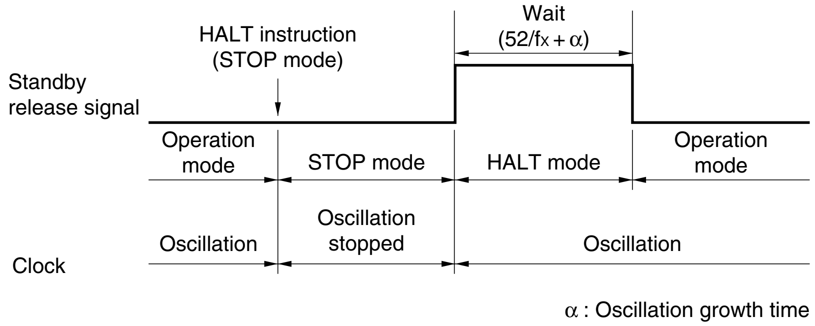
2. If STOP mode is set when the timer's down counter is not 0 (timer operating), the system is placed in STOP mode only after all the 10 bits of the timer's down counter and the timer output permit flag are cleared to 0.

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

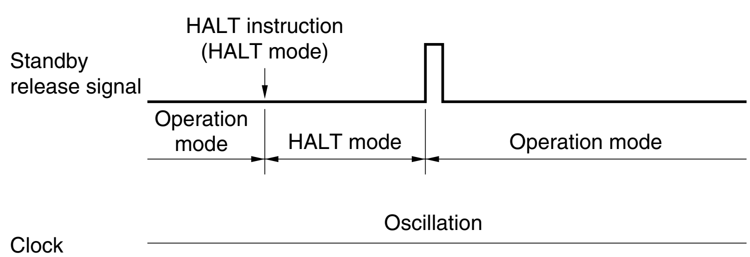
Figure 5-1. STOP Mode Release by Release Condition



Caution When a release condition is met 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

A system reset is effected by the following causes:

- When the POC circuit has detected low power-supply 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 stack pointer overflows or underflows

Table 6-1. Hardware Statuses After Reset

Hardware		• Reset by On-Chip POC Circuit During Operation • Reset by Other Factors[Note 1]	• Reset by the On-Chip POC Circuit During Standby Mode
PC	11 bits: μPD67, 67A, 68, 68A 12 bits: μPD69	000H	
SP (1 bit)		0B	
Data memory	R0 = DP	000H	
	R1 to RF	Undefined	
Accumulator (A)		Undefined	
Status flag (F)		0B	
Carry flag (CY)		0B	
Timer (10 bits)		000H	
Port register	P0	FFH	
	P1	×××× 11×1B[Note 2]	
Control register	P3	0000×000B[Note 3]	
	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's overflow or underflow

2. ×: Refers to the value by the K_I or S_2 pin status.
In order to prevent malfunction, be sure to input a low level to one or more of pins K_{I0} to K_{I3} when POC is released by supply voltage rising (Can be left open. When open, leave the pull-down resistor connected).

3. ×: Refers to the value based on a decrease of power supply voltage (0 when $V_{DD} \leq V_{ID}$).

Remark V_{ID}: RAM retention detection voltage

7. POC CIRCUIT

The POC circuit monitors the power supply voltage and applies an internal reset to 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 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.

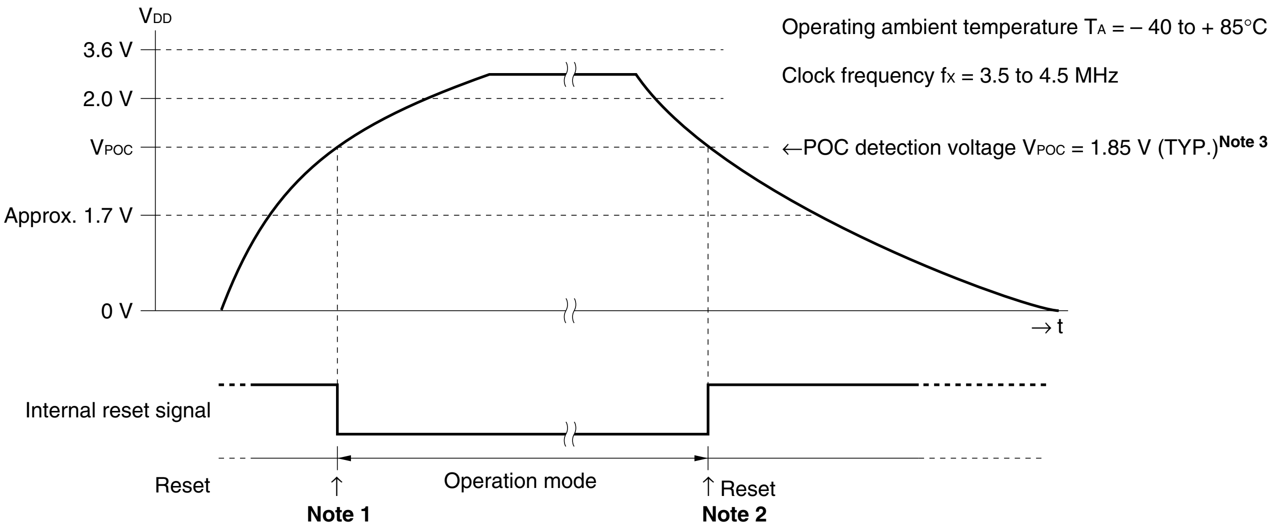
3. In order to prevent malfunction, be sure to input a low level to one or more of pins K_{I0} to K_{I3} when POC is released due to supply voltage rising (Can be left open. When open, leave the pull-down resistor connected).

7.1 Functions of POC Circuit

The POC circuit has the following functions:

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

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



Notes 1. Actually, oscillation stabilization wait time must elapse before the circuit is switched to operation mode. The oscillation stabilization wait time is about $534/f_X$ to $918/f_X$ (when about 134 to 230 μs; @ f_X = 4 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 the period of 1 ms or more. Therefore, in reality, there is the 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 guaranteed 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 becomes lower than the POC detection voltage. Therefore, there is no malfunction occurring due to a 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 of the POC circuit if it satisfies the condition that the clock can oscillate even at 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 in a product that actually includes a POC circuit, as follows.

<1> Connect a storage oscilloscope to the X_{OUT} pin so that the oscillation status can be measured.

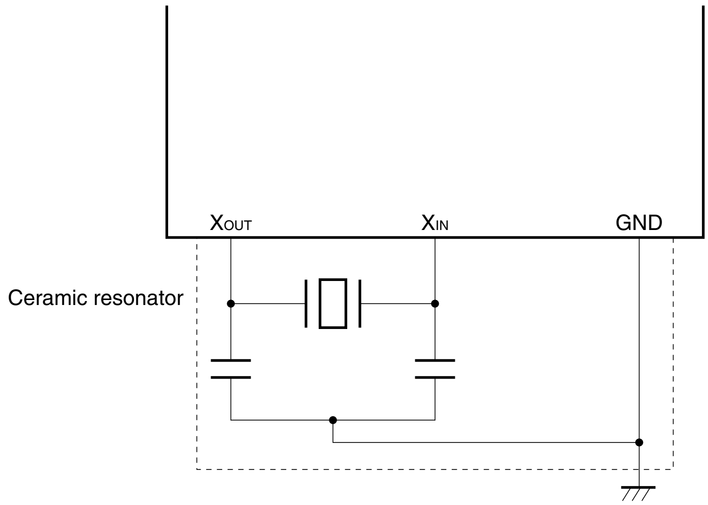
<2> Connect a power supply whose output voltage can be varied and then gradually raise the power supply voltage V_{DD} from 0 V (making sure to avoid V_{DD} > 3.6V).

At first (during V_{DD} < approx. 1.7 V), the X_{OUT} pin is 0 V regardless of the V_{DD}. However, at the point that 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 the V_{DD} has reached the V_{POC}.

8. SYSTEM CLOCK OSCILLATOR

The system clock oscillator consists of oscillators for ceramic resonators (f_X = 3.5 to 4.5 MHz).

Figure 8-1. System Clock



The system clock oscillator stops oscillating when a reset is applied 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 GND. Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

A capacitor (15 pF) for the oscillator can be incorporated via a mask option.

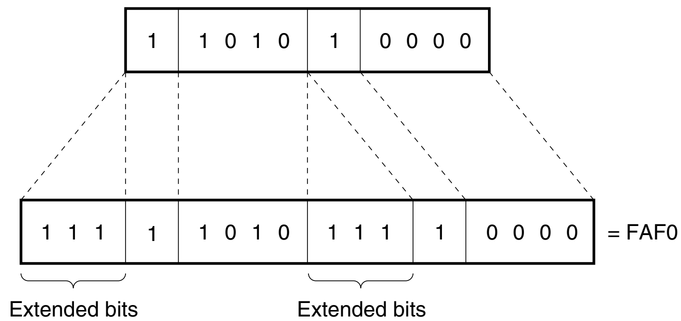
9. INSTRUCTION SET

9.1 Machine Language Output by Assembler

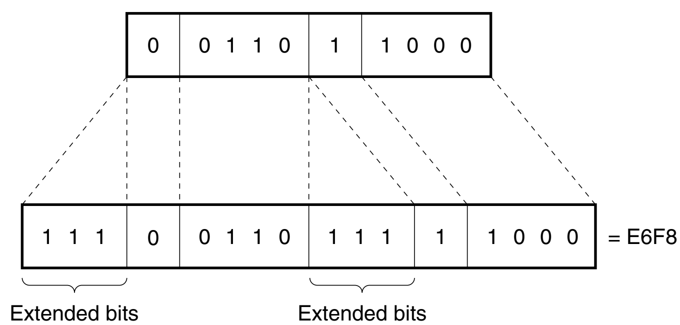
The bit length of the machine language of this product is 10 bits per word. However, the machine language that is output by the assembler is extended to 16 bits per word. As shown in the example below, the extension is made by inserting 3-bit extended bits (111) in two locations.

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
M0:	Modulo register for setting the low-level period
M00:	Modulo register for setting the low-level period (lower 4 bits)
M01:	Modulo register for setting the low-level period (higher 4 bits)
M1:	Modulo register for setting the high-level period
M10:	Modulo register for setting the high-level period (lower 4 bits)
M11:	Modulo register for setting the high-level period (higher 4 bits)
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's (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)
(×):	Content addressed with ×

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	FBEn			$(A) \leftarrow (A) \wedge (Rmn)$ m = 0, 1 n = 0 to F $CY \leftarrow A_3 \bullet Rmn_3$	1	1
	A, R1n	FAEn					
	A, @R0H	FAF0			$(A) \leftarrow (A) \wedge ((P13), (R0))_{7-4}$ $CY \leftarrow A_3 \bullet ROM_7$		
	A, @R0L	FBF0			$(A) \leftarrow (A) \wedge ((P13), (R0))_{3-0}$ $CY \leftarrow A_3 \bullet ROM_3$		
	A, #data4	FBF1	data4		$(A) \leftarrow (A) \wedge$ data4 $CY \leftarrow A_3 \bullet data4_3$	2	
ORL	A, R0n	FDEn			$(A) \leftarrow (A) \vee (Rmn)$ m = 0, 1 n = 0 to F $CY \leftarrow 0$	1	
	A, R1n	FCEn					
	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		$(A) \leftarrow (A) \vee$ data4 $CY \leftarrow 0$	2	
XRL	A, R0n	F5En			$(A) \leftarrow (A) \veebar (Rmn)$ m = 0, 1 n = 0 to F $CY \leftarrow A_3 \bullet Rmn_3$	1	
	A, R1n	F4En					
	A, @R0H	F4F0			$(A) \leftarrow (A) \veebar ((P13), (R0))_{7-4}$ $CY \leftarrow A_3 \bullet ROM_7$		
	A, @R0L	F5F0			$(A) \leftarrow (A) \veebar ((P13), (R0))_{3-0}$ $CY \leftarrow A_3 \bullet ROM_3$		
	A, #data4	F5F1	data4		$(A) \leftarrow (A) \veebar$ data4 $CY \leftarrow A_3 \bullet data4_3$	2	
INC	A	F4F3			$(A) \leftarrow (A) + 1$ if (A) = 0 $CY \leftarrow 1$ else $CY \leftarrow 1$	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 $(A_{n+1}) \leftarrow (A_n)$, $(A_0) \leftarrow (A_3)$ $CY \leftarrow A_3$		

I/O 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 ← $A_3 \bullet Pmn_3$		
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 ← $A_3 \bullet Pmn_3$		

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 to P0n are dealt with in pairs.

Data Transfer Instruction

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	FEEn			CY ← 0		
	A, @R0H	FEF0			(A) ← $((P13), (R0))_{7\text{-}4}$ CY ← 0		
	A, @R0L	FFF0			(A) ← $((P13), (R0))_{3\text{-}0}$ CY ← 0		
	A, #data4	FFF1	data4		(A) ← data4 CY ← 0	2	
	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 to R0n) ← data8 n = 0 to F	2	1
	Rn, @R0	E7En	—	—	(R1n to R0n) ← ((P13), (R0))n = 1 to F	1	

Remark Rn: R1n to 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				
	addr (Page 2)	E8F4	addr				
	addr (Page 3)	E9F4	addr				
JC	addr (Page 0)	ECF1	addr		if CY = 1 PC ← addr		
	addr (Page 1)	EAF1	addr		else PC ← PC + 2		
	addr (Page 2)	ECF4	addr				
	addr (Page 3)	EAF4	addr				
JNC	addr (Page 0)	EDF1	addr		if CY = 0 PC ← addr		
	addr (Page 1)	EBF1	addr		else PC ← PC + 2		
	addr (Page 2)	EDF4	addr				
	addr (Page 3)	EBF4	addr				
JF	addr (Page 0)	EEF1	addr		if F = 1 PC ← addr		
	addr (Page 1)	F0F1	addr		else PC ← PC + 2		
	addr (Page 2)	EEF4	addr				
	addr (Page 3)	F0F4	addr				
JNF	addr (Page 0)	EFF1	addr		if F = 0 PC ← addr		
	addr (Page 1)	F1F1	addr		else PC ← PC + 2		
	addr (Page 2)	EFF4	addr				
	addr (Page 3)	F1F4	addr				

Caution 0 and 4, which refer to PAGE0 and 4, 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			
	addr (Page 2)	E6F2	E8F4	addr			
	addr (Page 3)	E6F2	E9F4	addr			
RET		E8F2			PC ← ASR, SP ← SP – 1	1	1

Caution 0 and 4, which refer to PAGE0 and 4, 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		
	A, M00	FFF6			(A) ← (M0n) n = 0, 1		
	A, M01	FEF6			CY ← 0		
	A, M10	FFF7			(A) → (M1n) n = 0, 1		
	A, M11	FEF7			CY → 0		
	T0, A	E5FF			(Tn) ← (A) n = 0, 1		
	T1, A	F4FF			(T) n ← 0		
	M00, A	E5F6			(M0n) ← (A) n = 0, 1		
	M01, A	E4F6			CY ← 0		
	M10, A	E5F7			(M1n) ← (A) n = 0, 1		
	M11, A	E4F7			CY ← 0		

Mnemonic	Operand	Instruction Code			Operation	Instruction Length	Instruction Cycle
		1st Word	2nd Word	3rd Word			
MOV	T, #data10	E6FF	data10		(T) ← data10	2	1
	M0, #data10	E6F6	data10		(M0) ← data10		
	M1, #data10	E6F7	data10		(M1) ← data10		
	T, @R0	F4FF			(T) ← ((P13), (R0))	1	
	M0, @R0	E7F6			(M0) ← ((P13), (R0))		
	M1, @R0	E7F7			(M1) ← ((P13), (R0))		

Others

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 ← 1 else F ← 0		
	R0n	E3En			if statuses match F ← 1 else F ← 0 n = 0 to F	1	
SCAF		FAF3			if A = 0FH CY ← 1 else CY ← 0		
NOP		E0E0			PC ← PC + 1		

9.4 Accumulator Manipulation Instructions

ANL A, R0n

ANL A, R1n

<1> Instruction code:

1	1	0	1	R_4	0	R_3	R_2	R_1	R_0

<2> Cycle count: 1

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

CY ← A_3 • Rmn_3

The accumulator contents and the register Rmn 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) ← (A) $\wedge$ ((P13), $(R0))_{7\text{-}4}$ (in the case of ANL A, @R0H)

CY ← A_3 • ROM_7

(A) ← (A) $\wedge$ ((P13), $(R0))_{3\text{-}0}$ (in the case of ANL A, @R0L)

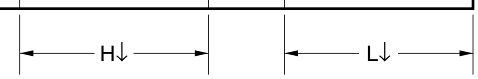
CY ← A_3 • ROM_3

The accumulator contents and the program memory contents specified by the control register P13 and register pair R_{10} to R_{00} are ANDed and the results are entered in the accumulator.

If H is specified, b_7, b_6, b_5 and b_4 take effect. If L is specified, b_3, b_2, b_1 and b_0 take effect.

- **Program memory (ROM) organization**

b_9	b_7	b_6	b_5	b_4	b_8	b_3	b_2	b_1	b_0



Valid bits at the time of accumulator manipulation

ANL A, #data4

<1> Instruction code:

1	1	0	1	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) ← (A) $\wedge$ data4

CY ← A_3 • $data4_3$

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_4	0	R_3 R_2 R_1 R_0

<2> Cycle count: 1

<3> Function: (A) ← (A) ∨ (Rmn)   m = 0, 1   n = 0 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) ← (A) ∨ (P13), $(R0))_{7\text{-}4}$ (in the case of ORL A, @R0H)

(A) ← (A) ∨ (P13), $(R0))_{3\text{-}0}$ (in the case of ORL A, @R0L)

CY ← 0

The accumulator contents and the program memory contents specified by the control register P13 and register pair R_{10}-R_{00} are ORed and the results are entered in the accumulator.

If H is specified, b_7, b_6, b_5 and b_4 take effect. If L is specified, b_3, b_2, b_1 and b_0 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_3 d_2 d_1 d_0

<2> Cycle count: 1

<3> Function: (A) ← (A) ∨ data4

CY ← 0

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

XRL A, R0n

XRL A, R1n

<1> Instruction code:

1	0 1 0 R_4	0	R_3 R_2 R_1 R_0

<2> Cycle count: 1

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

CY ← A_3 • Rmn_3

The accumulator contents and the register Rmn contents are 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) ← (A) ∀ (P13), $(R0))_{7\text{-}4}$ (in the case of XRL A, @R0H)

CY ← A_3 • ROM_7

(A) ← (A) ∀ (P13), $(R0))_{3\text{-}0}$ (in the case of XRL A, @R0L)

CY ← A_3 • ROM_3

The accumulator contents and the program memory contents specified by the control register P13 and register pair R_{10}-R_{00} are exclusive-ORed and the results are entered in the accumulator.

If H is specified, b_7, b_6, b_5, and b_4 take effect. If L is specified, b_3, b_2, b_1, and b_0 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) ← (A) ∀ data4

CY ← A_3 • $data4_3$

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) ← (A) + 1

if A = 0 CY ← 1

else CY ← 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}) ← (An), (A_0) ← (A_3)

CY ← A_3

The accumulator contents are rotated anticlockwise 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}) ← (An), (A_0) ← (A_3)

CY ← A_3

The accumulator contents are rotated anticlockwise bit by bit.

If A = 0H at the time of command 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_4	1	1 P_2 P_1 P_0

<2> Cycle count: 1

<3> Function: (A) ← (Pmn) m = 0, 1 n = 0, 1, 3, 4
CY ← 0

The port Pmn data is loaded (read) onto the accumulator.

OUT P0n, A
OUT P1n, A

<1> Instruction code:

0	0 1 0 P_4	1	1 P_2 P_1 P_0

<2> Cycle count: 1

<3> Function: (Pmn) ← (A) m = 0, 1 n = 0, 1, 3, 4

The accumulator contents are transferred to port Pmn to be latched.

ANL A, P0n
ANL A, P1n

<1> Instruction code:

1	1 0 1 P_4	1	1 P_2 P_1 P_0

<2> Cycle count: 1

<3> Function: (A) ← (A) ∧ (Pmn) m = 0, 1 n = 0, 1, 3, 4
CY ← A_3 • Pmn

The accumulator contents and the port Pmn 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_4	1	1 P_2 P_1 P_0

<2> Cycle count: 1

<3> Function: (A) ← (A) ∨ (Pmn) m = 0, 1 n = 0, 1, 3, 4
CY ← 0

The accumulator contents and the port Pmn 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_4	1	1 P_2 P_1 P_0

<2> Cycle count: 1

<3> Function: (A) ← (A) ⊻ (Pmn) m = 0, 1 n = 0, 1, 3, 4
CY ← A_3 • Pmn

The accumulator contents and the port Pmn 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_2 P_1 P_0
0	d_7 d_6 d_5 d_4	0	d_3 d_2 d_1 d_0

<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_{1n} to P_{0n} operating in pairs.

9.6 Data Transfer Instructions

MOV A, R0n

MOV A, R1n

<1> Instruction code:

1	1 1 1 R_4	0	R_3 R_2 R_1 R_0

<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) ← $((P13), (R0))_{7\text{-}4}$

CY ← 0

The higher 4 bits (b_7 b_6 b_5 b_4) of the program memory specified by control register P13 and register pair R_{10}-R_{00} are transferred to the accumulator. b_9 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) ← $((P13), (R0))_{3\text{-}0}$

CY ← 0

The lower 4 bits (b_3 b_2 b_1 b_0) of the program memory specified by control register P13 and register pair R_{10} to R_{00} are transferred to the accumulator. b_8 is ignored.

- **Program memory (ROM) contents**

	@R_0H		@R_0L
b_9	b_7 b_6 b_5 b_4	b_8	b_3 b_2 b_1 b_0

MOV A, #data4

<1> Instruction code:

1	1 1 1 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) ← 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_4	0	R_3	R_2	R_1	R_0

<2> Cycle count: 1

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

The accumulator contents are transferred to register Rmn.

MOV Rn, #data8

<1> Instruction code:

0	0	1	1	0	0	R_3	R_2	R_1	R_0
0	d_7	d_6	d_5	d_4	0	d_3	d_2	d_1	d_0

<2> Cycle count: 1

<3> Function: (R1n-R0n) ← 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:

R_0: R_{10} - R_{00}

R_1: R_{11} - R_{01}

:

R_E: R_{1E} - R_{0E}

R_F: R_{1F} - R_{0F}

(R_{0F} → Lower column; R_{1F} → Higher column)

MOV Rn, @R0

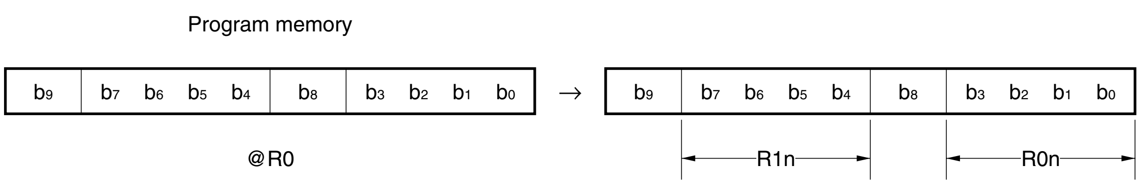
<1> Instruction code:

0	0	1	1	1	0	R_3	R_2	R_1	R_0

<2> Cycle count: 1

<3> Function: (R1n-R0n) ← ((P13), R0)) n = 1 to F

The program memory contents specified by control register P13 and register pair R_{10} to R_{00} are transferred to register pair R1n to R0n. The program memory consists of 10 bits and has the following state after the transfer to the register.



The higher 2 to 4 bits of the program memory address are specified by 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.

μPD67, 67A (ROM: 1K steps): Page 0
μPD68, 68A (ROM: 2K steps): Pages 0, 1
μPD69 (ROM: 4K steps): Pages 0 to 3
μPD6P9 (PROM: 4K steps): Pages 0 to 3

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]
Page 2 [0 | 1 0 0 0 | 1 | 0 1 0 0] ; page 3 [0 | 1 0 0 1 | 1 | 0 1 0 0]
[a_9 | a_7 a_6 a_5 a_4 | a_8 | a_3 a_2 a_1 a_0]

<2> Cycle count: 1

<3> Function: PC ← addr

The 10 bits (PC_{9-0}) of the program counter are replaced directly by the specified address addr (a_9 to a_0).

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]
Page 2 [0 | 1 1 0 0 | 1 | 0 1 0 0] ; page 3 [0 | 1 0 1 0 | 1 | 0 1 0 0]
[a_9 | a_7 a_6 a_5 a_4 | a_8 | a_3 a_2 a_1 a_0]

<2> Cycle count: 1

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

If the carry flag CY is set (to 1), a jump is made to the address specified by addr (a_9 to a_0).

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]
Page 2 [0 | 1 1 0 1 | 1 | 0 1 0 0] ; page 3 [0 | 1 0 1 1 | 1 | 0 1 0 0]
[a_9 | a_7 a_6 a_5 a_4 | a_8 | a_3 a_2 a_1 a_0]

<2> Cycle count: 1

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

If the carry flag CY is cleared (to 0), a jump is made to the address specified by addr (a9 to a0).

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]
Page 2 [0 | 1 1 1 0 | 1 | 0 1 0 0] ; page 3 [1 | 0 0 0 0 | 1 | 0 1 0 0]
[a_9 | a_7 a_6 a_5 a_4 | a_8 | a_3 a_2 a_1 a_0]

<2> Cycle count: 1

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

If the status flag F is set (to 1), a jump is made to the address specified by addr (a_9 to a_0).

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]

Page 2 [0 | 1 1 1 1 | 1 | 0 1 0 0] ; page 3 [1 | 0 0 0 1 | 1 | 0 1 0 0]

[a_9 | a_7 a_6 a_5 a_4 | a_8 | a_3 a_2 a_1 a_0]

<2> Cycle count: 1

<3> Function: if F = 0 PC ← addr

else PC ← PC + 2

If the status flag F is cleared (to 0), a jump is made to the address specified by addr (a_9 to a_0).

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.

μPD67, 67A (ROM: 1K steps): Page 0
μPD68, 68A (ROM: 2K steps): Pages 0, 1
μPD69 (ROM: 4K steps): Pages 0 to 3
μPD6P9 (PROM: 4K steps): Pages 0 to 3

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]

Page 2 [0 | 1 0 0 0 | 1 | 0 1 0 0] ; page 3 [0 | 1 0 0 1 | 1 | 0 1 0 0]

[a_9 | a_7 a_6 a_5 a_4 | a_8 | a_3 a_2 a_1 a_0]

<2> Cycle count: 2

<3> Function: SP ← SP + 1

ASR ← PC

PC ← addr

Increments (+1) the stack pointer value and saves the program counter value in the address stack register. Then, enters the address specified by the operand addr (a_9 to a_0) into 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

Restores the value saved in the address stack register to the program counter. Then, decrements (–1) the stack pointer.

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

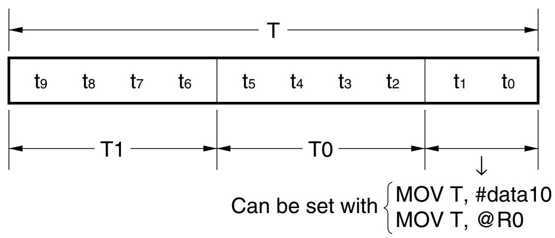
<1> Instruction code:

1	1 1 1 0/1	1	1 1 1 1

<2> Cycle count: 1

<3> Function: (A) ← (Tn)   n = 0, 1
CY ← 0

The timer register Tn contents are transferred to the accumulator. T1 corresponds to (t_9, t_8, t_7, t_6); T0 corresponds to (t_5, t_4, t_3, t_2).



MOV A, M00
MOV A, M01

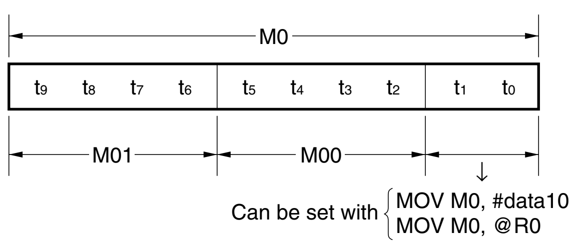
<1> Instruction code:

1	1 1 1 0/1	1	0 1 1 0

<2> Cycle count: 1

<3> Function: (A) ← (M0n)   n = 0, 1
CY ← 0

The modulo register M0n contents are transferred to the accumulator. M01 corresponds to (t_9, t_8, t_7, t_6); M00 corresponds to (t_5, t_4, t_3, t_2).



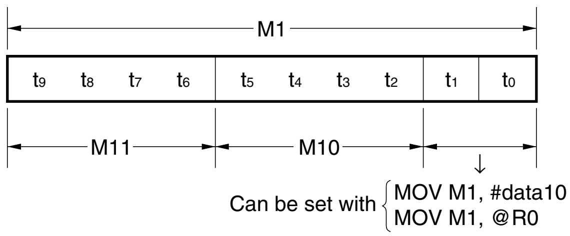
MOV A, M10
MOV A, M11

<1> Instruction code: | 1 | 1 1 1 0/1 | 1 | 0 1 1 1 |

<2> Cycle count: 1

<3> Function: (A) ← (M1n) n = 0, 1
CY ← 0

The modulo register M1n contents are transferred to the accumulator. M11 corresponds to (t_9, t_8, t_7, t_6); M10 corresponds to (t_5, t_4, t_3, t_2).



MOV T0, A
MOV T1, A

<1> Instruction code: | 0 | 0 1 0 0/1 | 1 | 1 1 1 1 |

<2> Cycle count: 1

<3> Function: (Tn) ← (A) n = 0, 1

The accumulator contents are transferred to the timer register Tn. T1 corresponds to (t_9, t_8, t_7, t_6); T0 corresponds to (t_5, t_4, t_3, t_2). **After executing this instruction, if data is transferred to T1, t_1 becomes 0; if data is transferred to T0, t_0 becomes 0.**

MOV M00, A
MOV M01, A

<1> Instruction code: | 0 | 0 1 0 0/1 | 1 | 0 1 1 0 |

<2> Cycle count: 1

<3> Function: (M0n) ← (A) n = 0, 1
CY ← 0

The accumulator contents are transferred to the modulo register M0n. M01 corresponds to (t_9, t_8, t_7, t_6); M00 corresponds to (t_5, t_4, t_3, t_2). **After executing this instruction, if data is transferred to M01, t_1 becomes 0; if data is transferred to M00, t_0 becomes 0.**

MOV M10, A
MOV M11, A

<1> Instruction code: | 0 | 0 1 0 0/1 | 1 | 0 1 1 1 |

<2> Cycle count: 1

<3> Function: (M1n) ← (A) n = 0, 1
CY ← 0

The accumulator contents are transferred to the modulo register M1n. M11 corresponds to (t_9, t_8, t_7, t_6); M10 corresponds to (t_5, t_4, t_3, t_2). **After executing this instruction, if data is transferred to M11, t_1 becomes 0; if data is transferred to M10, t_0 becomes 0.**

MOV T, #data10

<1> Instruction code:

0	0 1 1 0	1	1 1 1 1
t_1	t_9 t_8 t_7 t_6	t_0	t_5 t_4 t_3 t_2

<2> Cycle count: 1

<3> Function: (T) ← data10

The immediate data is transferred to the timer register T (t_9 to t_0).

★ **Remark** The timer time is set as follows.

(a) μPD67, 68, and 69
(Set value + 1) × 64/fx

(b) μPD67A and 68A
(Set value + 1) × 64/fx – 4/fx

MOV M0, #data10

<1> Instruction code:

0	0 1 1 0	1	0 1 1 0
t_1	t_9 t_8 t_7 t_6	t_0	t_5 t_4 t_3 t_2

<2> Cycle count: 1

<3> Function: (M0) ← data10

The immediate data is transferred to the modulo register M0 (t_9 to t_0).

MOV M1, #data10

<1> Instruction code:

0	0 1 1 0	1	0 1 1 1
t_1	t_9 t_8 t_7 t_6	t_0	t_5 t_4 t_3 t_2

<2> Cycle count: 1

<3> Function: (M1) ← data10

The immediate data is transferred to the modulo register M1 (t_9 to t_0).

MOV T, @R0

<1> Instruction code:

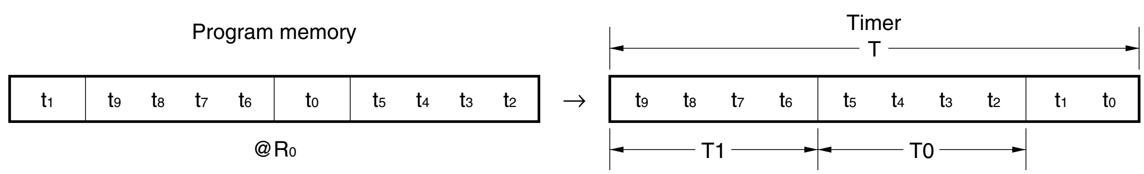
0	0 1 1 1	1	1 1 1 1

<2> Cycle count: 1

<3> Function: (T) ← ((P13), (R0))

Transfers the program memory contents to the timer register T (t_9 to t_0) specified by the control register P13 and the register pair R_{10} to R_{00}.

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



The higher 2 to 4 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.**

MOV M0, @R0

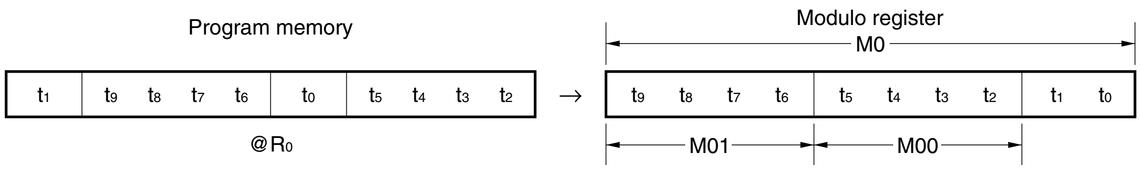
<1> Instruction code: | 0 | 0 1 1 1 | 1 | 0 1 1 0 |

<2> Cycle count: 1

<3> Function: (M0) ← ((P13), (R0))

Transfers the program memory contents to the modulo register M0 (t_9 to t_0) specified by the control register P13 and the register pair R_{10} to R_{00}.

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



The higher 2 to 4 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.

MOV M1, @R0

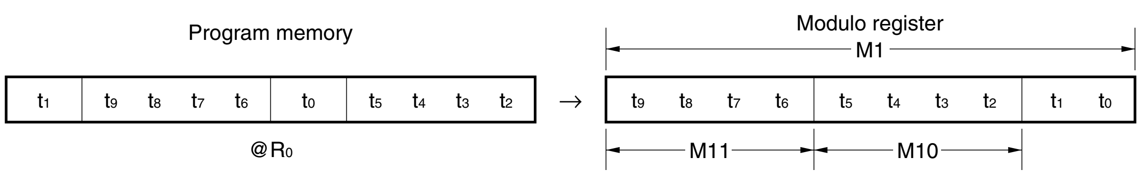
<1> Instruction code: | 0 | 0 1 1 1 | 1 | 0 1 1 1 |

<2> Cycle count: 1

<3> Function: (M1) ← ((P13), (R0))

Transfers the program memory contents to the modulo register M1 (t_9 to t_0) specified by the control register P13 and the register pair R_{10} to R_{00}.

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



The higher 2 to 4 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 Others

HALT #data4

<1> Instruction code: | 0 | 0 0 1 0 | 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: Standby mode

Places the CPU in standby mode.

The condition for having the standby mode (HALT/STOP mode) canceled is specified by the immediate data.

STTS R0n

<1> Instruction code: | 0 | 0 0 1 1 | 0 | R_3 R_2 R_1 R_0 |

<2> Cycle count: 1

<3> Function: if statuses match F ← 1

else F ← 0 n = 0 to F

Compares the S_0, S_1, $K_{I/O}$, K_I, and TIMER statuses with the register R_{0n} contents. If at least one of the statuses matches the bits that have been set, the status flag F is set (to 1).

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

STTS #data4

<1> Instruction code: | 0 | 0 0 1 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: if statuses match F ← 1

else F ← 0

Compares the S_0, S_1, S_2, $K_{I/O}$, K_I, and TIMER statuses with the immediate data contents. If at least one of the statuses matches the bits that have been set, the status flag F is set (to 1).

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

SCAF (Set Carry If A_{CC} = F_H)

<1> Instruction code: | 1 | 1 0 1 0 | 1 | 0 0 1 1 |

<2> Cycle count: 1

<3> Function: if A = 0FH CY ← 1

else CY ← 0

Sets the carry flag CY (to 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	
×××0	0000	0 (clear)
××01	0001	0 (clear)
×011	0011	0 (clear)
0111	0111	0 (clear)
1111	1111	1 (set)

Remark ×: don’t care

NOP

<1> Instruction code: | 0 | 0 0 0 0 | 0 | 0 0 0 0 |

<2> Cycle count: 1

<3> Function: PC ← PC + 1

No operation

10. ASSEMBLER RESERVED WORDS

10.1 Mask Option Directives

When creating a program in the μPD67, 67A, 68, 68A, and 69, 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 quasi-directives from the OPTION quasi-directive 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

Symbol field	Mnemonic field	Operand field	Comment field
[Label:]	OPTION		[; Comment]
	:		
	:		
	ENDOP		

10.1.2 Mask option definition quasi-directives

The quasi-directives that can be used in the mask option definition block are listed in Table 10-1.

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

Example

Symbol field	Mnemonic field	Operand field	Comment field
	OPTION		
	USECAP		; Capacitor for oscillation incorporated
	ENDOP		

Table 10-1. Mask Option Definition Directives

Name	Mask Option Definition Quasi-Directive	PRO File	
		Address Value	Data Value
CAP	USECAP (Capacitor for oscillation incorporated)	2043H	01
	NOUSECAP (Capacitor for oscillation not incorporated)		00

11. ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings (T_A = +25°C)

Item	Symbol	Conditions		Ratings	Unit
Power supply voltage	V_{DD}			–0.3 to +3.8	V
Input voltage	V_I	$K_{I/O}$, K_I, S_0, S_1, S_2		–0.3 to V_{DD} + 0.3	V
Output voltage	V_O			–0.3 to V_{DD} + 0.3	V
Output current, high	I_{OH} Note	REM	Peak value	–30	mA
			rms value	–20	mA
		$\overline{LED}$	Peak value	–7.5	mA
			rms value	–5	mA
		One $K_{I/O}$ pin	Peak value	–13.5	mA
			rms value	–9	mA
		Total for $\overline{LED}$ and $K_{I/O}$ pins	Peak value	–18	mA
			rms value	–12	mA
Output current, low	I_{OL} Note	REM	Peak value	7.5	mA
			rms value	5	mA
		$\overline{LED}$	Peak value	7.5	mA
			rms value	5	mA
Operating ambient temperature	T_A			–40 to +85	°C
Storage temperature	T_{stg}			–65 to +150	°C

Note The rms value should be calculated as follows: [rms value] = [Peak value] $\times \sqrt{\text{Duty}}$.

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_A = –40 to +85°C)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Power supply voltage	V_{DD}	f_X = 3.5 to 4.5 MHz	2.0	3.0	3.6	V

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 2.0 to 3.6 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	$K_{I/O}$		$0.7V_{DD}$		V_{DD}	V
	V_{IH2}	K_I, S_0, S_1, S_2		$0.65V_{DD}$		V_{DD}	V
Input voltage, low	V_{IL1}	$K_{I/O}$		0		$0.3V_{DD}$	V
	V_{IL2}	K_I, S_0, S_1, S_2		0		$0.15V_{DD}$	V
Input leakage current, high	I_{LH1}	K_I $V_I = V_{DD}$, pull-down resistor not incorporated				3	μA
	I_{LH2}	S_0, S_1, S_2 $V_I = V_{DD}$, pull-down resistor not incorporated				3	μA
Input leakage current, low	I_{UL1}	K_I V_I = 0 V				–3	μA
	I_{UL2}	$K_{I/O}$ V_I = 0 V				–3	μA
	I_{UL3}	S_0, S_1, S_2 V_I = 0 V				–3	μA
Output voltage, high	V_{OH1}	REM, $\overline{LED}$, $K_{I/O}$	I_{OH} = –0.3 mA	$0.8V_{DD}$			V
Output voltage, low	V_{OL1}	REM, $\overline{LED}$	I_{OL} = 0.3 mA			0.3	V
	V_{OL2}	$K_{I/O}$	I_{OL} = 15 μA			0.4	V
Output current, high	I_{OH1}	REM	V_{DD} = 3.0 V, V_{OH} = 1.0 V	–5	–12		mA
	I_{OH2}	$K_{I/O}$	V_{DD} = 3.0 V, V_{OH} = 2.2 V	–2.5	–7		mA
Output current, low	I_{OL1}	$K_{I/O}$	V_{DD} = 3.0 V, V_{OL} = 0.4 V	47	70		μA
			V_{DD} = 3.0 V, V_{OL} = 2.2 V	260	390		μA
On-chip pull-down resistor	R_1	K_I, S_0, S_1, S_2		75	150	300	kΩ
	R_2	$K_{I/O}$		130	250	500	kΩ
Data retention power supply voltage	V_{DDOR}	In STOP mode		0.9		3.6	V
RAM retention detection voltage	V_{ID}				1.4	1.5	V
Supply current	I_{DD1}	Operation mode	f_X = 4.0 MHz, V_{DD} = 3 V ±10%		0.7	1.4	mA
	I_{DD2}	HALT mode	f_X = 4.0 MHz, V_{DD} = 3 V ±10%		0.65	1.3	mA
	I_{DD3}	STOP mode	V_{DD} = 3 V ±10%		2.0	9.0	μA
			V_{DD} = 3 V ±10%, T_A = 25°C		1.8	3.0	μA

AC Characteristics (T_A = –40 to +85°C, V_{DD} = 2.0 to 3.6 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Command execution time	t_{CY}			14	16	18.5	μs
K_I, S_0, S_1, S_2 high-level width	t_H			10			μs
		When releasing standby mode	In HALT mode	10			μs
			In STOP mode	**Note**			μs

Note 10 + 278/f_X + oscillation growth time

Remark t_{CY} = 64/f_X (f_X: System clock oscillator frequency)

POC Circuit (T_A = –40 to +85°C)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
POC detection voltage**Note**	V_{POC}			1.85	2.0	V

Note Refers to the voltage with which the POC circuit releases an internal reset. If $V_{POC} < V_{DD}$, the internal reset is released.
From the time of $V_{POC} \geq V_{DD}$ until the internal reset takes effect, a delay of up to 1 ms occurs. When the period of $V_{POC} \geq V_{DD}$ lasts less than 1 ms, the internal reset may not take effect.

System Clock Oscillator Characteristics (T_A = –40 to +85°C, V_{DD} = 2.0 to 3.6 V)

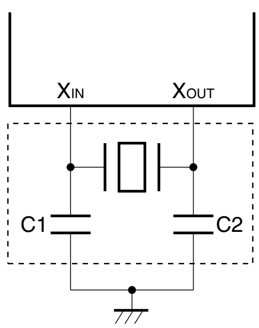
Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Oscillator frequency (ceramic resonator)	f_X		3.5	4.0	4.5	MHz

RECOMMENDED OSCILLATOR CONSTANT

★ **Ceramic Resonator (T_A = –40 to +85°C) (Without On-Chip Capacitor for Oscillator Specified by Mask Option)**

Manufacturer	Part Number	Frequency	Recommended Constant (pF)		Oscillation Voltage Range (V_{DD})		Remark
		(MHz)	C1	C2	MIN.	MAX.	
Murata Mfg. Co., Ltd.	CSTLS3M50G53-B0	3.5	Unnecessary (on-chip C type)		2.0	3.6	–
	CSTLS3M50G56-B0						
	CSALA4M00G55-B0	4.0	30	30			
	CSTLS4M00G53-B0		Unnecessary (on-chip C type)				
	CSTLS4M00G56-B0						
	CSTLS4M50G53-B0	4.5					
	CSTLS4M50G56-B0						
TDK	FCR3.52MC5	3.52	Unnecessary (on-chip C type)				
	FCR4.0MC5	4.0					
Kyocera	KBR-3.64MKE	3.64	Unnecessary (on-chip C type)				
	KBR-3.64MSE		33	33			
	KBR-4.0MKE	4.0	Unnecessary (on-chip C type)				
	KBR-4.0MSE		33	33			

External circuit example

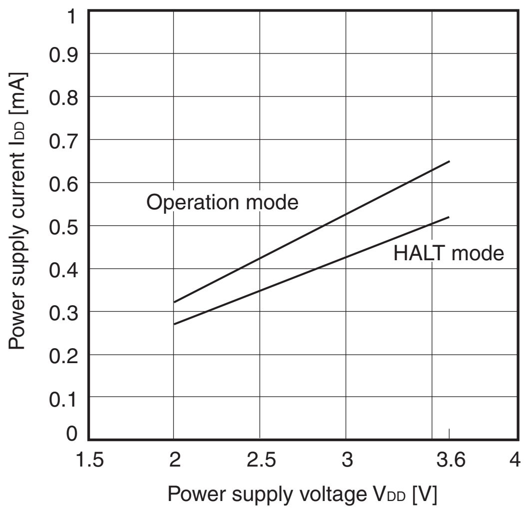


Caution These oscillator constants are reference values based on evaluation by the manufacturer of the resonator under a specific environment .
If optimization of the oscillator characteristics is required for the actual application, apply to the resonator manufacturer for evaluation on the mounting circuit.
The oscillation voltage and oscillation frequency only indicate the oscillator characteristics; the oscillator must be used within the ratings of the DC and AC characteristics specified under the internal operation conditions of the μPD67, 67A, 68, 68A, and 69 .

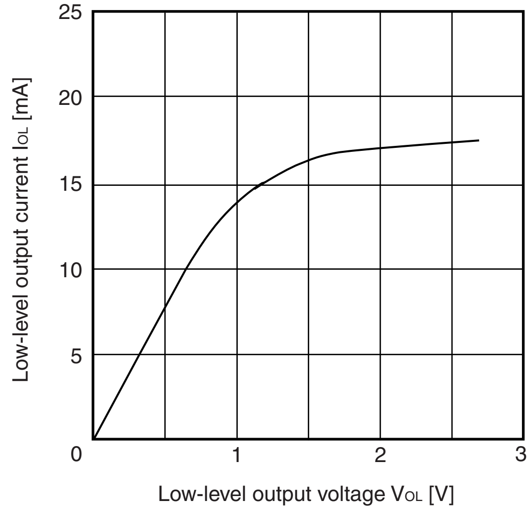
Remark The incorporation of the oscillation capacitor by a mask option is under evaluation.

12. CHARACTERISTIC CURVES (REFERENCE VALUES)

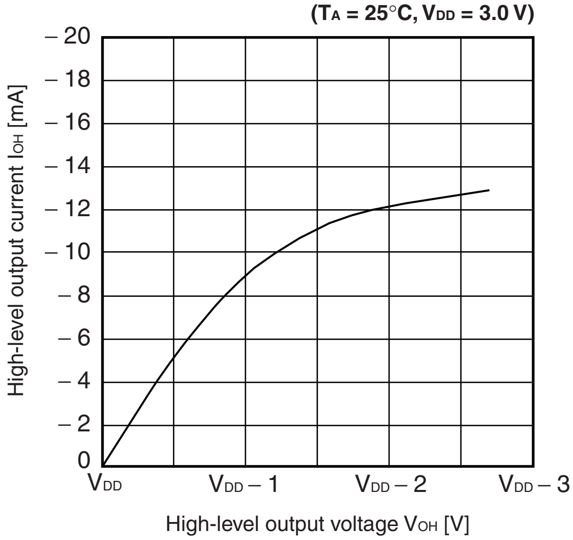
I_{DD} vs. V_{DD} (f_X = 4 MHz)
(T_A = 25°C)



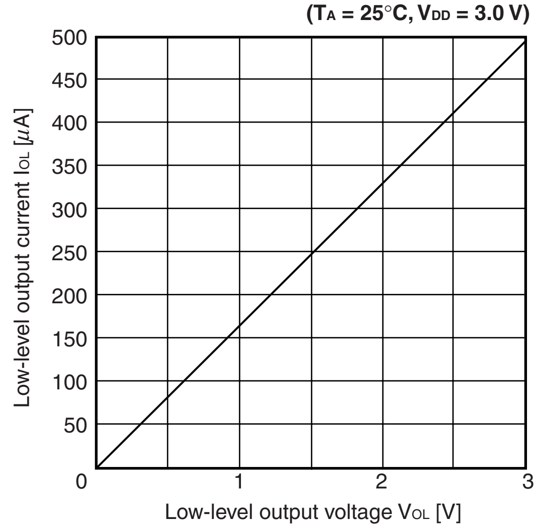
I_{OL} vs. V_{OL} (REM, $\overline{LED}$)
(T_A = 25°C, V_{DD} = 3.0 V)



I_{OH} vs. V_{OH} (REM, $\overline{LED}$, $K_{I/O}$)
(T_A = 25°C, V_{DD} = 3.0 V)



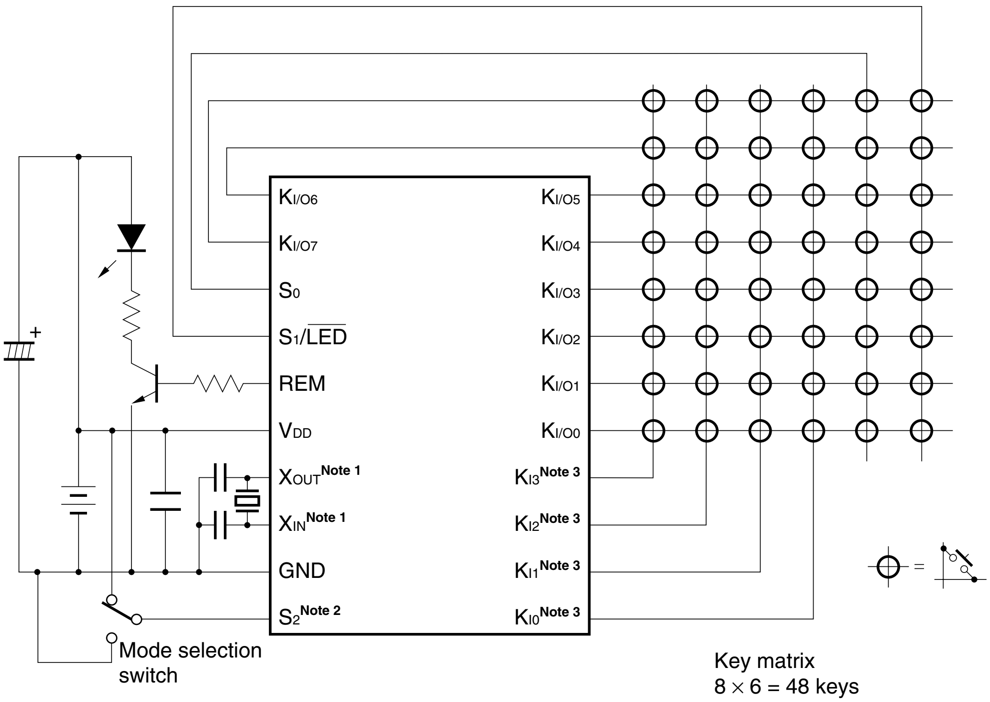
I_{OL} vs. V_{OL} ($K_{I/O}$)
(T_A = 25°C, V_{DD} = 3.0 V)



13. APPLICATION CIRCUIT EXAMPLE

Example of Application in System

- **Remote-control transmitter (48 keys; mode selection switch supported)**

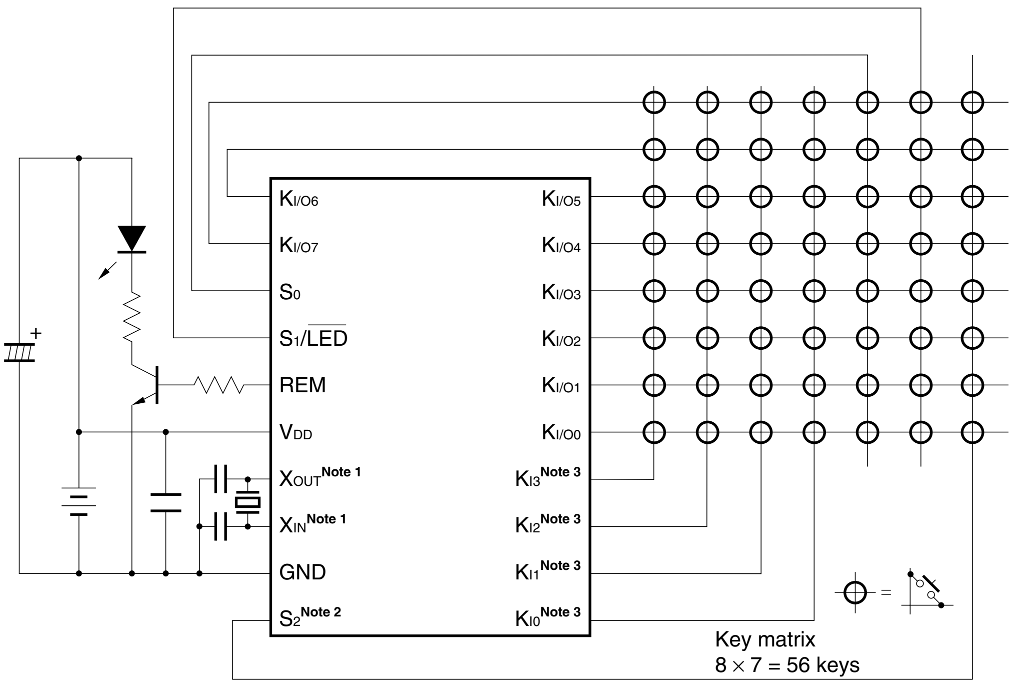


Notes 1. When incorporation of a capacitor for oscillation has not been specified by a mask option.

2. S_2: Set to disable for STOP mode release.

3. Set pins K_{I0} to K_{I3} to "with pull-down resistors".

• **Remote-control transmitter (56 keys accommodated)**



Notes 1. When incorporation of a capacitor for oscillation has not been specified by a mask option.

2. S_2: Set to enable for STOP mode release.

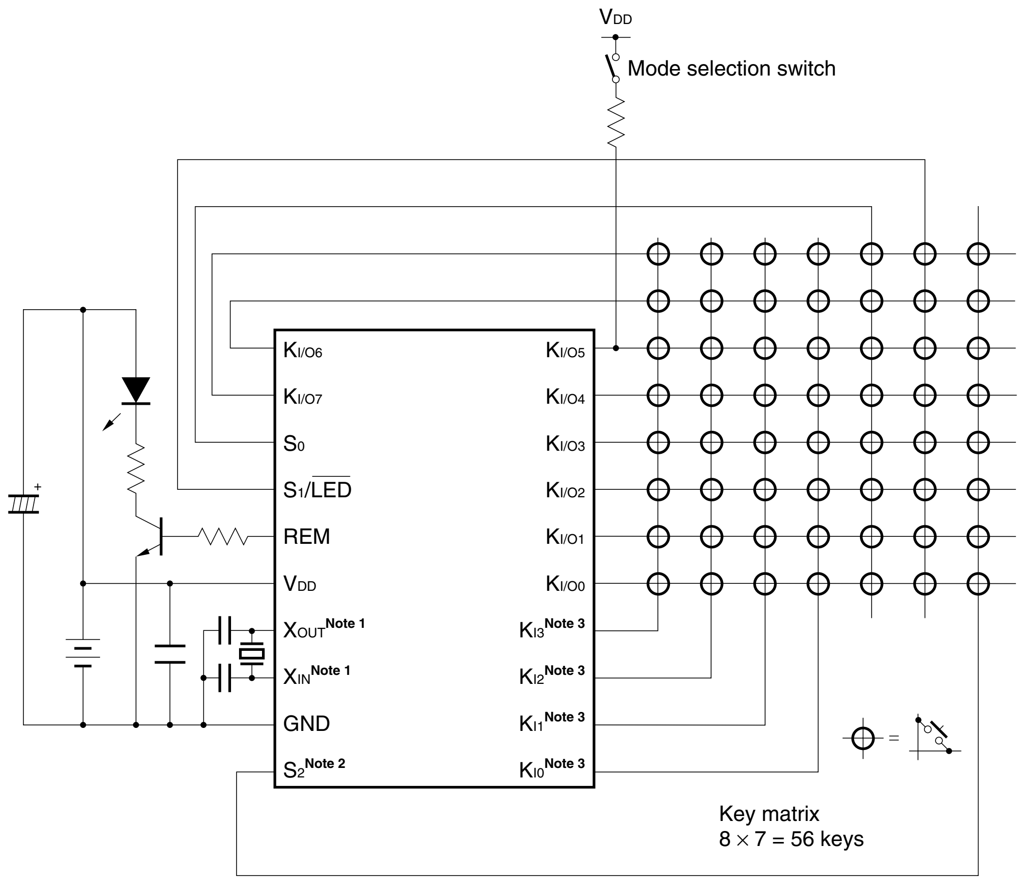
3. Set pins K_{I0} to K_{I3} to "with pull-down resistors".

- **Remote-control transmitter (56 keys supported, mode selection switch supported)**

  Data can be read from the $K_{I/O0}$ to $K_{I/O7}$ pins by connecting a pull-up resistor of 50 kΩ and a switch to these pins (which then become high level when the switch is on and low level when off). Set the $K_{I/O0}$ to $K_{I/O7}$ pins to input mode at this time. Reading data from these pins enables multiple output data to be obtained for the same key input.

  A pull-up resistor can be connected to any of pins $K_{I/O0}$ to $K_{I/O7}$ (the figure below shows an example of when a pull-up resistor is connected to the $K_{I/O5}$ pin).

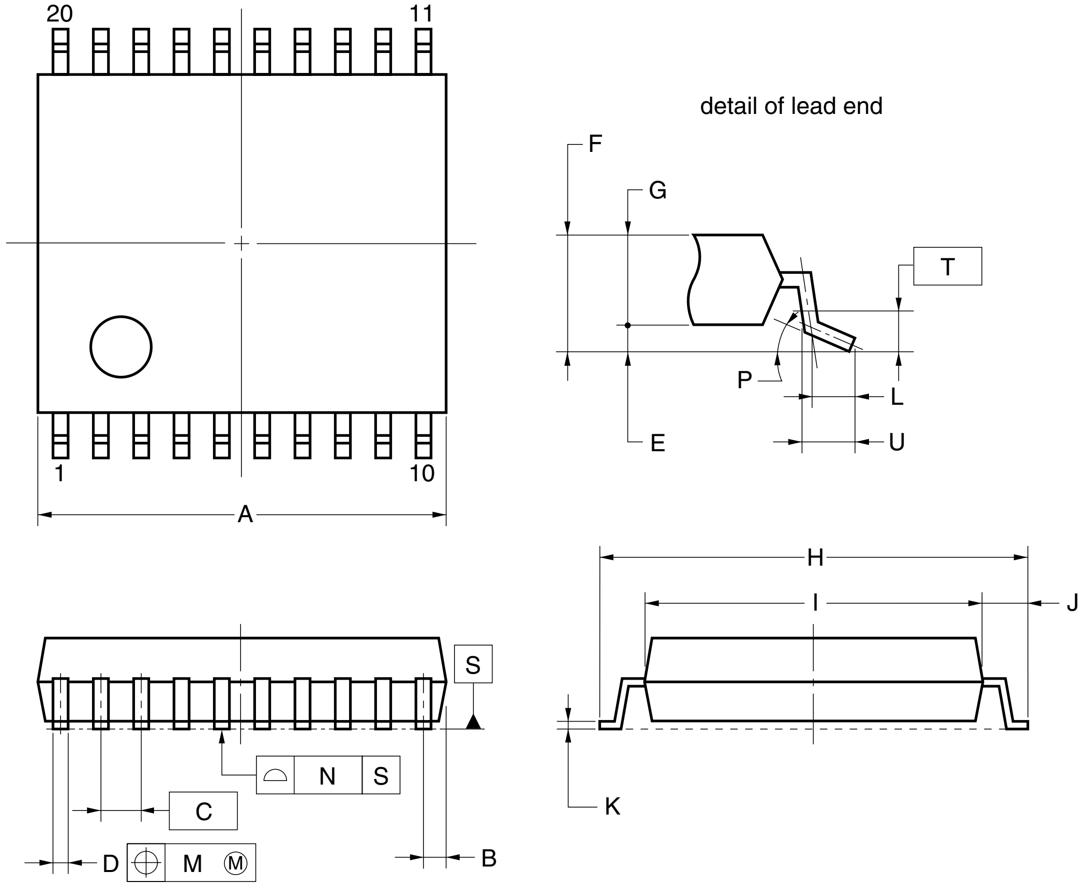
  The mode may not be correctly read while a key is being pressed.



Notes
1. When incorporation of a capacitor for oscillation has not been specified by a mask option.
2. S_2: Set to enable for STOP mode release.
3. Set pins K_{I0} to K_{I3} to "with pull-down resistors".

14. PACKAGE DRAWINGS

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	$0.24^{+0.08}_{-0.07}$
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	$3°^{+5°}_{-3°}$
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 μPD67, 67A, 68, 68A, and 69 should be soldered and mounted under the following recommended conditions.

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

For soldering methods and conditions other than those recommended below, contact an NEC sales representatives.

Table 15-1. Surface Mounting Type Soldering Conditions

μPD67MC-×××-5A4: 20-pin plastic SSOP (7.62 mm (300))
μPD67AMC-×××-5A4: 20-pin plastic SSOP (7.62 mm (300))
μPD68MC-×××-5A4: 20-pin plastic SSOP (7.62 mm (300))
μPD68AMC-×××-5A4: 20-pin plastic SSOP (7.62 mm (300))
μPD69MC-×××-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. (at 210°C or higher), Count: Three times or less	IR35-00-3
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (at 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, Preliminary heat temperature: 120°C max. (package surface temperature)	WS60-00-1
Partial heating	Pin temperature: 300°C max., Time: 3 sec. max. (per pin row)	—

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

★ ## APPENDIX A. DEVELOPMENT TOOLS

An emulator is provided as an emulation tool and a PROM programmer and program adapter are provided as writing tools for the PROM product, the μPD6P9.

Hardware

- Emulator (EB-69[Note 1])
  Tool to emulate the μPD67, 67A, 68, 68A, 69, and 6P9.

- Emulation probe (NP-20GS[Note 1])
  Probe for 20-pin SOP/SSOP to connect the emulator to the target system.

- Flexible board (EV-9500GS-20)
  20-pin flexible board to facilitate the connection between the emulation probe and the target system.

- PROM programmer (AF-9706[Note 2], AF-9708[Note 2], AF-9709[Note 2])
  PROM programmer supporting the μPD6P9.
  The μPD6P9 can be programmed by connecting the program adapter.

- Program adapter (PA-61P34BMC)
  Adapter to program the μPD6P9. Use in combination with the AF-9706, AF-9708, and AF-9709.

Notes 1. This is a product of Naito Densei Machida Mfg. Co., Ltd.
For details, contact Naito Densei Machida Mfg. Co., Ltd. (TEL: +81-45-475-4191).
2. This is a product of Ando Electric Co., Ltd.
For details, contact Ando Electric Co., Ltd. (TEL: +81-3-3733-1151).

Software

- **Assembler (AS6133 Ver. 2.22 or later)**
  Development tool for remote control transmitter software.

Ordering Number List of AS6133

Host Machine	OS	Supply Medium	Ordering Number
PC-9800 series (CPU: 80,386 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.

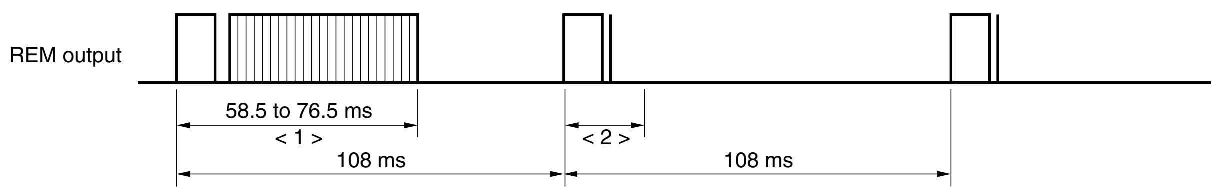
★ APPENDIX B. FUNCTIONAL COMPARISON BETWEEN μPD67A, 68A, 69, AND OTHER PRODUCTS

Item		μPD64	μPD65	μPD67A	μPD68A	μPD69
ROM capacity		1,002 × 10 bits	2,026 × 10 bits	1,002 × 10 bits	2,026 × 10 bits	4,074 × 10 bits
RAM capacity		32 × 4 bits				128 × 4 bits (32 × 4 bits × 4 pages)
Stack		1 level (multiplexed with RF of RAM)				
Key matrix		8 × 6 = 48 keys	8 × 7 = 56 keys			
Key extended input		S_0, S_1	S_0 to S_2			
Clock frequency		Ceramic oscillation • f_X = 2.4 to 8 MHz • f_X = 2.4 to 4 MHz (with POC circuit)	Ceramic oscillation • f_X = 2.4 to 8 MHz	Ceramic oscillation • f_X = 3.5 to 4.5 MHz		
Timer	Clock	f_X/64, f_X/128		f_X/64		
	Count start	Writing count value				
	Output value	(Set value + 1) × 64/f_X (or 128/f_X)		(Set value + 1) × 64/f_X – 4/f_X		(Set value + 1) × 64/f_X
Carrier	Frequency	• f_X/8, f_X/64, f_X/96 (timer clock: f_X/64) • f_X/16, f_X/128, f_X/192 (timer clock: f_X/128) • No carrier		Each high-/low-level width can be set from 250 ns to 64 μs (@ f_X = 4 MHz operation) via modulo registers (2 channels).		
	Output start	Synchronized with timer				
Instruction execution time		16 μs (f_X = 4 MHz)				
"MOV Rn, @R0" instruction		n = 1 to F				
Standby mode	Reset	$\overline{\text{RESET}}$ input, POC	POC			
	Release condition (HALT instruction)	• HALT mode for timer only. • STOP mode for only releasing K_I ($K_{I/O}$ high-level output or $K_{I/O0}$ high-level output)				
Relation between HALT instruction execution and status flag (F)		HALT instruction not executed when F = 1				
POC circuit		• Mask option • Low level output to $\overline{\text{RESET}}$ pin on detection • V_{POC} = 1.6 V (TYP.)	• Provided • Generates internal reset signal on detection • V_{POC} = 1.85 V (TYP.)			
RAM retention detector		None		• Provided • V_{ID} = 1.4 V (TYP.)		
Mask option		POC circuit	None	Capacitor for oscillator (15 pF)		
Supply voltage		• V_{DD}=1.8 to 3.6 V • V_{DD}=2.2 to 3.6 V (with POC circuit)	V_{DD} = 2.0 to 3.6 V			
Operating temperature		• T_A=–40 to +85°C • T_A=–20 to +70°C (with POC circuit)	T_A=–40 to +85°C			
Package		• 20-pin plastic SOP • 20-pin plastic SSOP	20-pin plastic SSOP			
One-time PROM model		μPD6P4B	μPD6P5	μPD6P9		

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

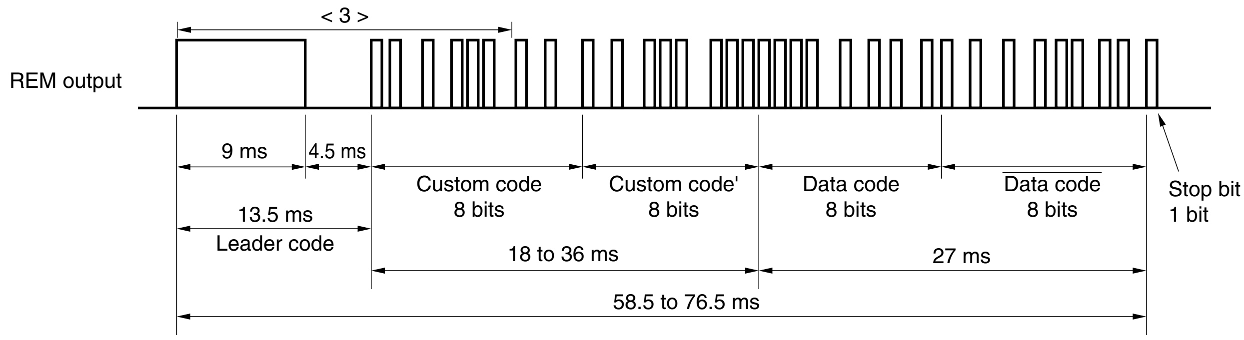
Caution When using the NEC transmission format, please apply to NEC for a custom code.

(1) REM output waveform (From <2> on, the output is made only when the key is held down)

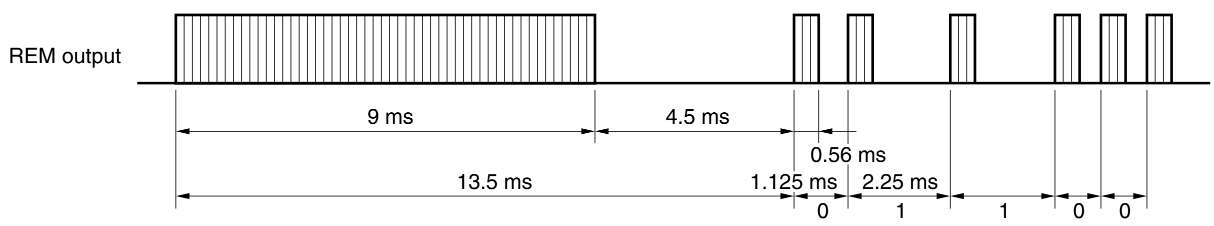


Remark If the key is repeatedly pressed, the power consumption of the infrared light-emitting diode (LED) can be reduced by sending the reader code and the stop bit from the second time.

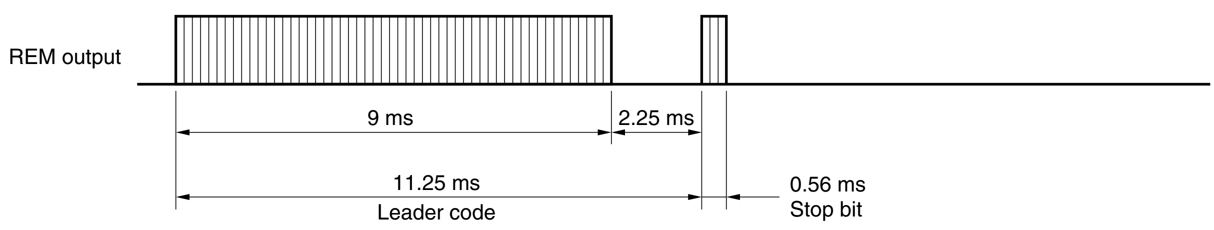
(2) Enlarged waveform of <1>



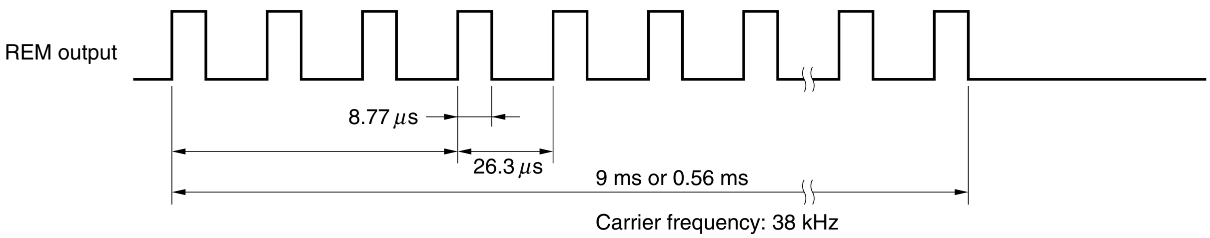
(3) Enlarged waveform of <3>



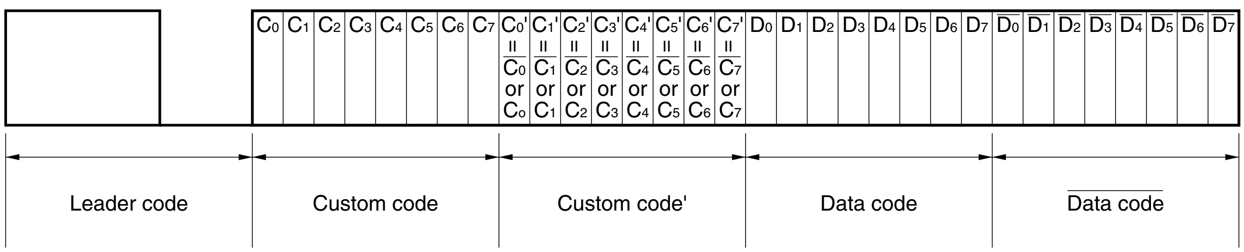
(4) Enlarged waveform of <2>



(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, not only fully decode (make sure to check $\overline{\text{Data code}}$ as well) the total 32 bits of the 16-bit custom codes (Custom code, Custom code') and the 16-bit data codes (Data code, $\overline{\text{Data code}}$), but also check to make sure that no signals are present.**

[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 do Brasil S.A.
Electron Devices Division
Guarulhos-SP, Brasil
Tel: 11-6462-6810
Fax: 11-6462-6829

NEC Electronics (Europe) GmbH
Duesseldorf, Germany
Tel: 0211-65 03 01
Fax: 0211-65 03 327

- **Sucursal en España**
  Madrid, Spain
  Tel: 091-504 27 87
  Fax: 091-504 28 60
- **Succursale Française**
  Vélizy-Villacoublay, France
  Tel: 01-30-67 58 00
  Fax: 01-30-67 58 99
- **Filiale Italiana**
  Milano, Italy
  Tel: 02-66 75 41
  Fax: 02-66 75 42 99
- **Branch The Netherlands**
  Eindhoven, The Netherlands
  Tel: 040-244 58 45
  Fax: 040-244 45 80
- **Branch Sweden**
  Taeby, Sweden
  Tel: 08-63 80 820
  Fax: 08-63 80 388
- **United Kingdom Branch**
  Milton Keynes, UK
  Tel: 01908-691-133
  Fax: 01908-670-290

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 Shanghai, Ltd.
Shanghai, P.R. China
Tel: 021-6841-1138
Fax: 021-6841-1137

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-2719-2377
Fax: 02-2719-5951

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

J02.4