NEC

MOS INTEGRATED CIRCUIT

μPD64A, 65

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

DESCRIPTION

The μPD64A and 65 feature low-voltage 2.0 V operation, and incorporate a carrier generator for infrared remote control transmission, a standby release function through key entry, and a programmable timer, making them ideal for infrared remote control transmitters.

A one-time PROM product, the μPD6P5 is also available for program evaluation or small-quantity production.

FEATURES

- Program memory (ROM)
  - μPD64A: 1002 × 10 bits
  - μPD65: 2026 × 10 bits
- Data memory (RAM): 32 × 4 bits
- On-chip carrier generator for infrared remote control
- 9-bit programmable timer: 1 channel
- Instruction execution time: 16 μs (when operating at f_X = 4 MHz: ceramic oscillation)
- Stack levels: 1 (stack RAM is also used for data memory RF)
- I/O pins ($K_{I/O}$): 8
- Input pins (K_I): 4
- Sense input pin (S_0, S_2): 2
- $S_1/\overline{LED}$ pin (I/O): 1 (when in output mode, this is the remote control transmission display pin.)
- Power supply voltage: V_{DD} = 2.0 to 3.6 V
- Operating ambient temperature: T_A = –40 to +85°C
- Oscillation frequency: f_X = 2.4 to 8 MHz
- On-chip POC circuit

APPLICATIONS

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

Unless otherwise specified, the μPD65 is treated as the representative model throughout this document.

Document No. U14380EJ3V0DS00 (3rd edition)
Date Published December 2000 N CP(K)
Printed in Japan

The mark ★ shows major revised points.

ORDERING INFORMATION

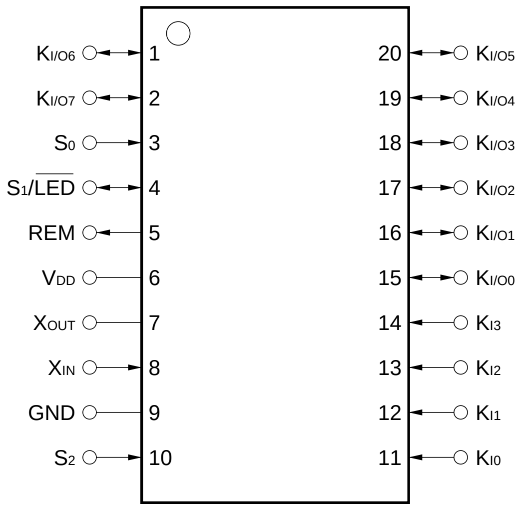
Part Number	Package
μPD64AMC-×××-5A4	20-pin plastic SSOP (7.62 mm (300))
μPD65MC-×××-5A4	20-pin plastic SSOP (7.62 mm (300))

Remark ××× indicates ROM code suffix.

PIN CONFIGURATION (TOP VIEW)

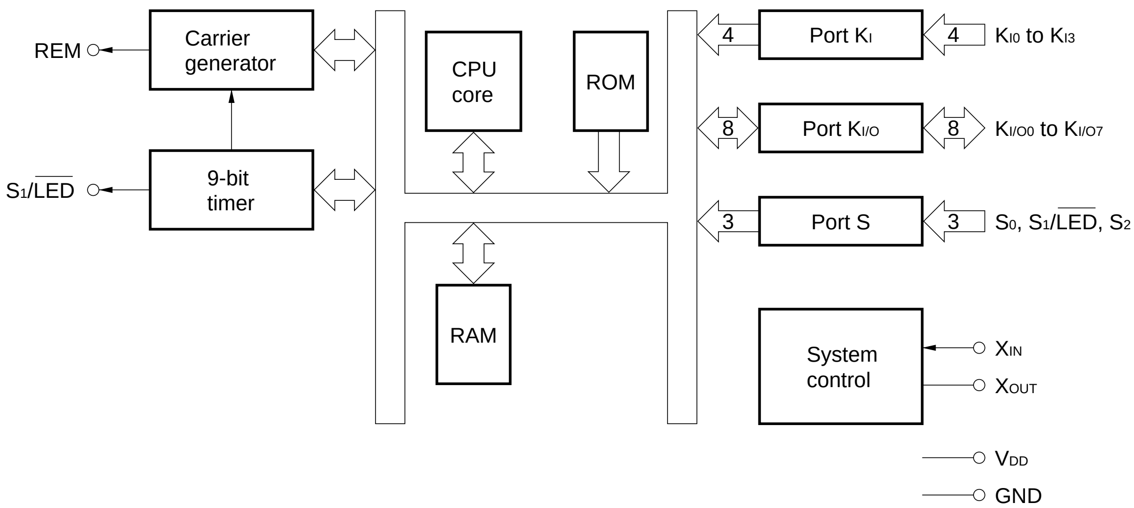
20-Pin Plastic SSOP (7.62 mm (300))

- μPD64AMC-×××-5A4
- μPD65MC-×××-5A4



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

BLOCK DIAGRAM



LIST OF FUNCTIONS

Item	μPD64A	μPD65	μPD6P5
ROM capacity	1002 × 10 bits	2026 × 10 bits	
	Mask ROM		One-time PROM
RAM capacity	32 × 4 bits		
Stack	1 level (multiplexed with RF of RAM)		
I/O pins	• 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 (alternate function as S_1 pin)		
Number of keys	• 32 • 56 (when extended by key extension input)		
★ Clock frequency	Ceramic oscillation • f_X = 2.4 to 8 MHz		Ceramic oscillation • f_X = 2.4 to 4.8 MHz
Instruction execution time	16 μs (f_X = 4 MHz)		
Carrier frequency	f_X/8, f_X/16, f_X/64, f_X/96, f_X/128, f_X/192, no carrier (high level)		
Timer	9-bit programmable timer: 1 channel		
POC circuit	On chip		
★ 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))		

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. I/O can be switched in 8-bit units. In input mode, a pull-down resistor is added. In output mode, these pins can be used as key scan outputs from the key matrix.	CMOS push-pull[Note 1]	High-level output
3	S_0	Input port. Can also be used as a key return input from key matrix. In input mode, use of a pull-down resistor for the S_0 and S_1 ports can be specified by software in 2-bit units. If input mode is canceled by software, this pin is placed in the OFF mode and enters a high-impedance state.	—	High impedance (OFF mode)
4	$S_1/\overline{\text{LED}}$	I/O port. In input mode (S_1), this pin can also be used as a key return input from key matrix. Use of a pull-down resistor for the S_0 and S_1 ports can be specified by software in 2-bit units. In output mode ($\overline{\text{LED}}$), 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 $\overline{\text{LED}}$ output synchronously with the REM signal.	CMOS push-pull	High-level output ($\overline{\text{LED}}$)
5	REM	Infrared remote control transmission output. The output is active high. Carrier frequency: $f_X/8$, $f_X/64$, $f_X/96$, high-level, $f_X/16$, $f_X/128$, $f_X/192$ (usable on software)	CMOS push-pull	Low-level output
6	V_{DD}	Power supply	—	—
7 8	X_{OUT} X_{IN}	Connecting ceramic resonators for system clock.	—	Low level (oscillation stopped)
9	GND	Ground.	—	—
10	S_2	Input port. The use of a STOP mode release for the S_2 port can be specified by software. When using this pin as a key input from a key matrix, enable the use of the STOP mode release (at this time, a pull-down resistor is connected internally.) When the STOP mode release is disabled, this pin can be used as an input port that does not release the STOP mode even if the release condition is established (at this time, a pull-down resistor is not connected internally.)	—	Input (high impedance, STOP mode release cannot be used)
11 to 14	K_{I0} to K_{I3}[Note 2]	4-bit input port. These pins can be used as key return inputs from the key matrix. Use of a pull-down resistor can be specified by software in 4-bit units.	—	Input (low level)

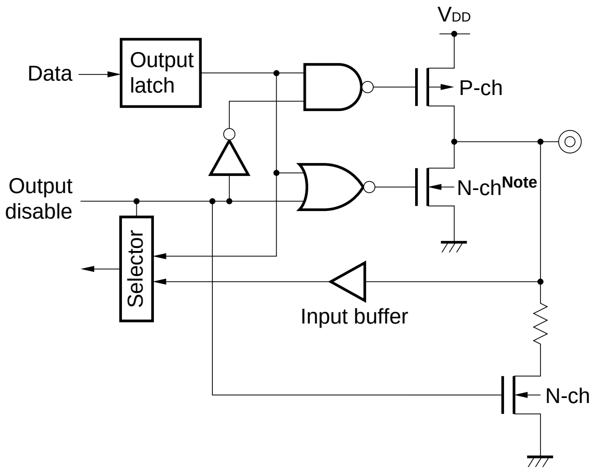
Notes 1. Be aware that the drive capability of the low-level output side is held low.

★ 2. In order to prevent malfunction, do not input a high level to all the K_{I0} to K_{I3} pins when POC is released due to supply voltage startup (these pins can be left open. When leaving open, keep pull-down resistors connected).

1.2 Pin I/O Circuits

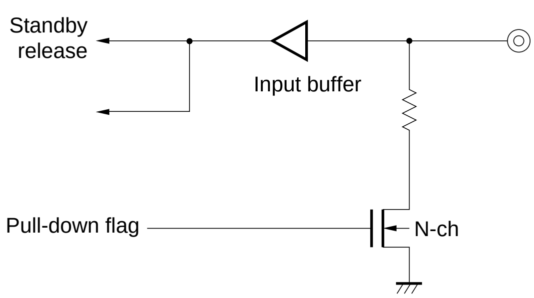
The I/O circuits of the μPD64A and 65 pins are shown in partially simplified forms below.

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

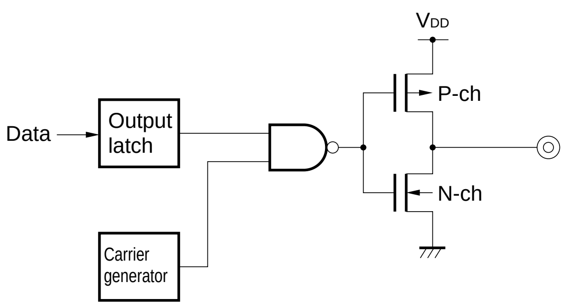


Note The drive capability is held low.

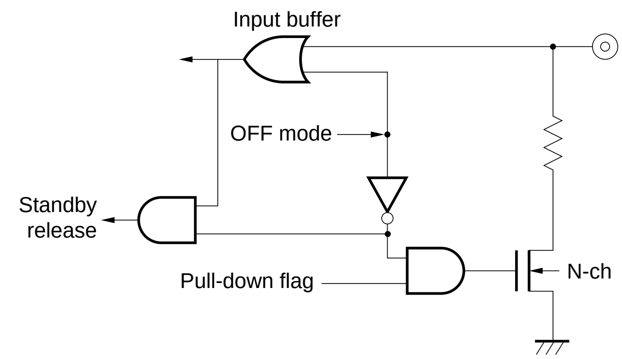
(2) K_{I0} to K_{I3}



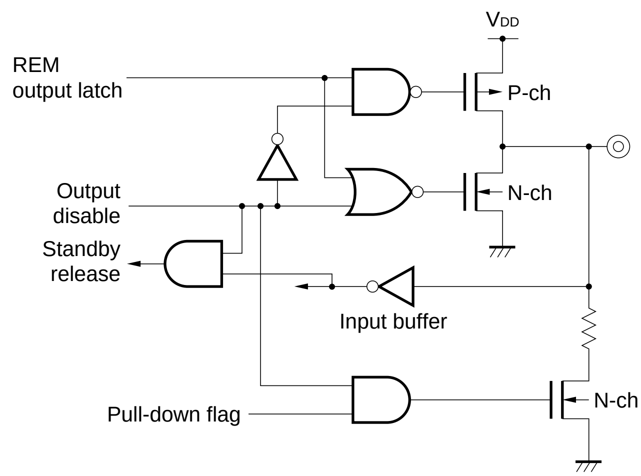
(3) REM



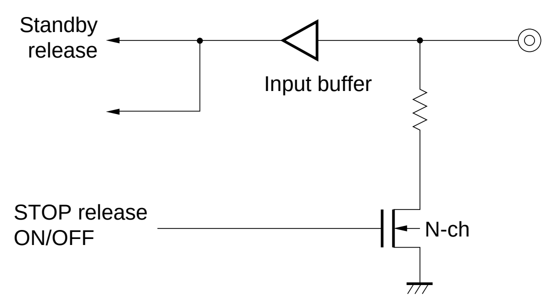
(4) S_0



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



(6) S_2



1.3 Recommended Connection of Unused Pins

The following connections are recommended for unused pins.

Table 1-1. Recommended Connection of Unused Pins

Pin		Connection	
		Inside Microcontroller	Outside Microcontroller
$K_{I/O}$	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_I		—	

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

2. INTERNAL CPU FUNCTIONS

2.1 Program Counter (PC): 11 Bits

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

Figure 2-1. Program Counter Configuration

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

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

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

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

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

2.2 Stack Pointer (SP): 1 Bit

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

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

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

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

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

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

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

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

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

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

Figure 2-2. Address Stack Register Configuration

				$\overbrace{}^{\text{RF}}$							
ASR	ASR10	ASR9	ASR8	ASR7	ASR6	ASR5	ASR4	ASR3	ASR2	ASR1	ASR0

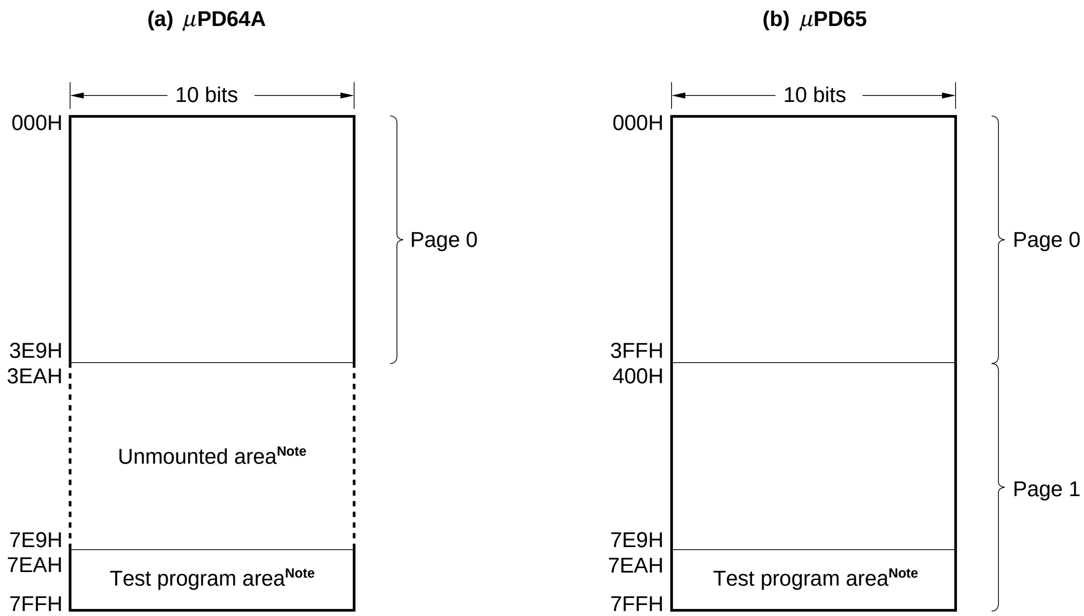
2.4 Program Memory (ROM): 1002 steps × 10 bits (μPD64A)
2026 steps × 10 bits (μPD65)

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

The program memory stores programs and table data, etc.

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

Figure 2-3. Program Memory Map



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

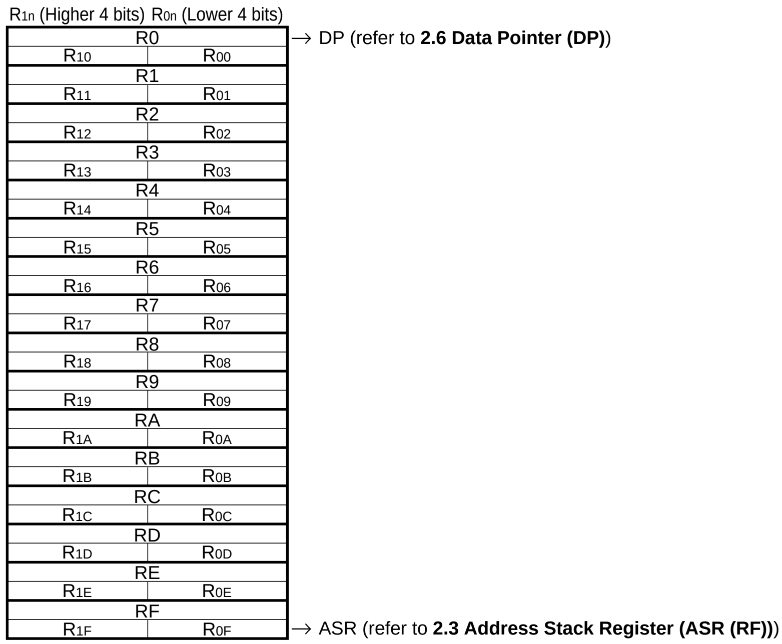
2.5 Data Memory (RAM): 32 × 4 Bits

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

RF is also used as the ASR.

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

Figure 2-4. Data Memory Configuration



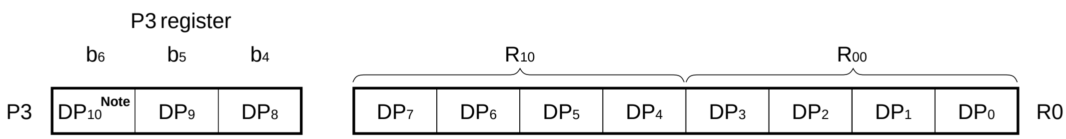
2.6 Data Pointer (DP): 11 Bits

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

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

After reset, the pointer contents become "000H".

Figure 2-5. Data Pointer Configuration



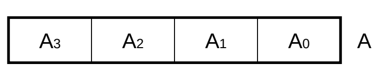
Note Set DP_{10} of the μPD64A to 0.

2.7 Accumulator (A): 4 Bits

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

After reset, the accumulator contents are left undefined.

Figure 2-6. Accumulator Configuration



2.8 Arithmetic and Logic Unit (ALU): 4 Bits

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

2.9 Flags

2.9.1 Status flag (F)

Pin and timer statuses can be checked by executing the STTS instruction to check the status flag.
The status flag is set (1) in the following cases.

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

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

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

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

Operand Value of STTS Instruction				Condition for Status Flag (F) to Be Set
b_3	b_2	b_1	b_0	
0	0	0	0	High level is input to at least one of the K_I pins.
	0	1	1	High level is input to at least one of the K_I pins.
	1	1	0	High level is input to at least one of the K_I pins.
	1	0	1	The down counter of the timer is 0.
1	Any combination 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 the S_0**Note 1**, S_1**Note 1**, and S_2**Note 2** pins.

★ **Notes 1.** The S_0 and S_1 pins must be set to input mode (bit 2 of the P4 register is set to 0 and bit 0 to 1).
2. The use of STOP mode release for the S_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 (1) in the following cases.

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

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

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

3. PORT REGISTERS (PX)

The $K_{I/O}$ 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. Port register values after reset are shown below.

Figure 3-1. Port Register Configuration

<table>
<tr><th colspan="8">Port register</th><th>After reset</th></tr>
<tr><td colspan="8">P0</td><td rowspan="3">FFH</td></tr>
<tr><td colspan="4">P_{10}</td><td colspan="4">P_{00}</td></tr>
<tr><td>$K_{I/O7}$</td><td>$K_{I/O6}$</td><td>$K_{I/O5}$</td><td>$K_{I/O4}$</td><td>$K_{I/O3}$</td><td>$K_{I/O2}$</td><td>$K_{I/O1}$</td><td>$K_{I/O0}$</td></tr>
<tr><td colspan="8">P1</td><td rowspan="3">×FH[Note]</td></tr>
<tr><td colspan="4">P_{11}</td><td colspan="4">P_{01}</td></tr>
<tr><td>K_{I3}</td><td>K_{I2}</td><td>K_{I1}</td><td>K_{I0}</td><td>$S_1/\overline{LED}$</td><td>S_0</td><td>S_2</td><td>1</td></tr>
<tr><td colspan="8">P3 (control register 0)</td><td rowspan="3">03H</td></tr>
<tr><td colspan="4">P_{13}</td><td colspan="4">P_{03}</td></tr>
<tr><td>0</td><td>DP_{10}</td><td>DP_9</td><td>DP_8</td><td>TCTL</td><td>CARY</td><td>MOD_1</td><td>MOD_0</td></tr>
<tr><td colspan="8">P4 (control register 1)</td><td rowspan="3">26H</td></tr>
<tr><td colspan="4">P_{14}</td><td colspan="4">P_{04}</td></tr>
<tr><td>0</td><td>0</td><td>K_I pull-down</td><td>S_0/S_1 pull-down</td><td>S_2 STOP release</td><td>$S_1/\overline{LED}$ mode</td><td>$K_{I/O}$ mode</td><td>S_0 mode</td></tr>
</table>

Note ×: Value based on the K_I pin state

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

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.

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

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

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

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

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

Caution Because during double pressing of a key, a high-level output and a low-level output may conflict with each other at the $K_{I/O}$ port, 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 may occur.

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, state of the $K_{I/O}$ pin 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 Port (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 entry.

The pin state can be read.

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, the 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, do not input a high level to all the K_{I0} to K_{I3} pins when POC is released due to supply voltage startup (these pins can be left open. When leaving open, keep pull-down resistors connected).**

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 with bit 0 of the P4 register.

In input mode, 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.

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

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

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

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

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

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

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

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

3.2.4 S_2 port (bit 1 of P1)

The S_2 port is an input port.

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

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

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

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

3.3 Control Register 0 (P3)

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

After reset, the register becomes 0000 0011B.

Table 3-4. Control Register 0 (P3)

Bit		b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
Name		—	DP (Data Pointer)			TCTL	CARY	MOD_1	MOD_0
			DP_{10} Note	DP_9	DP_8				
Set value	0	Fixed to “0”	0	0	0	1/1	ON	Refer to **Table 3-5**.	
	1		1	1	1	1/2	OFF		
After reset		0	0	0	0	0	0	1	1

b_0, b_1: These bits specify the carrier frequency and duty ratio of the REM output.

b_2: This bit specifies the availability of the carrier of the frequency specified by b_0 and b_1.
“0” = ON (with carrier); “1” = OFF (without carrier; high level)

b_3: This bit changes the carrier frequency and the timer clock’s frequency division ratio.
“0” = 1/1 (carrier frequency: The specified value of b_0 and b_1; timer clock: $f_X/64$)
“1” = 1/2 (carrier frequency: Half of the specified value of b_0 and b_1; timer clock: $f_X/128$)

Table 3-5. Timer Clock and Carrier Frequency Settings

b_3	b_2	b_1	b_0	Timer Clock	Carrier Frequency (Duty Ratio)
0	0	0	0	$f_X/64$	$f_X/8$ (Duty 1/2)
		0	1		$f_X/64$ (Duty 1/2)
		1	0		$f_X/96$ (Duty 1/2)
		1	1		$f_X/96$ (Duty 1/3)
	1	×	×		Without carrier (high level)
1	0	0	0	$f_X/128$	$f_X/16$ (Duty 1/2)
		0	1		$f_X/128$ (Duty 1/2)
		1	0		$f_X/192$ (Duty 1/2)
		1	1		$f_X/192$ (Duty 1/3)
	1	×	×		Without carrier (high level)

b_4, b_5, b_6: These bits specify the higher 3 bits (DP_8, DP_9 and DP_{10}) of ROM’s data pointer.

Note Set DP_{10} of the μPD64A to 0.

Remark ×: don’t care

3.4 Control Register 1 (P4)

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

Table 3-6. Control Register 1 (P4)

Bit		b_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
Set value	0	Fixed to "0"	Fixed to "0"	OFF	OFF	Disabled	S_1	IN	OFF
	1			ON	ON	Enabled	$\overline{LED}$	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: Specifies the use of the STOP mode release for the S_2 port (with/without pull-down resistor). "0" = Disabled (pull-down unavailable); "1" = Enabled (pull-down available).
b_4: Specifies the availability of the pull-down resistor for the S_0/S_1 port input mode. "0" = OFF (unavailable); "1" = ON (available).
b_5: Specifies the availability of the pull-down resistor for the K_I port. "0" = OFF (unavailable); "1" = ON (available).

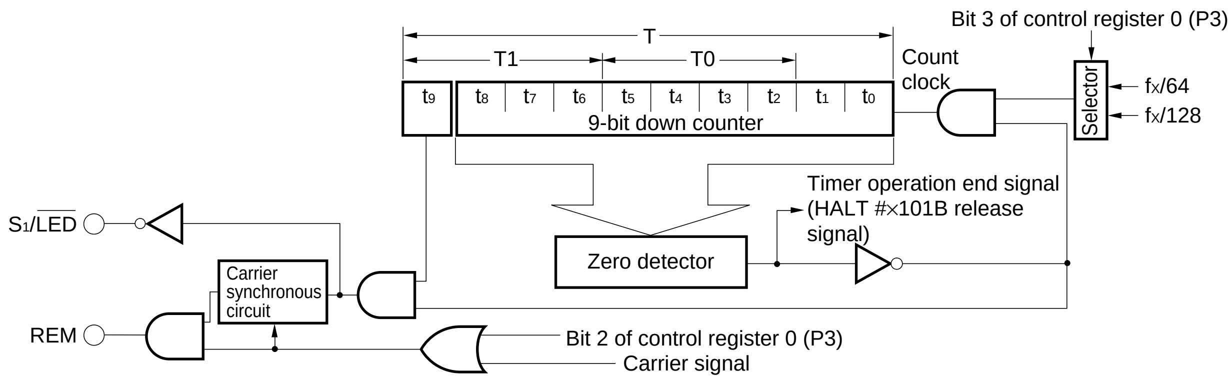
Remark In output mode or in OFF mode, all the pull-down resistors are automatically disconnected.

4. TIMER

4.1 Timer Configuration

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

Figure 4-1. Timer Configuration



4.2 Timer Operation

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

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

The down counter is decremented (–1) in the cycle of $64/f_X$ or $128/f_X$ **Note**. 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 operation, the HALT mode is released and the instruction following the HALT instruction is executed. The output of the timer operation end signal is continued while the down counter is 0 and the timer is stopped. The following expression indicates the relationship between the timer's time and the down counter's set value.

Timer time = (Set value + 1) × $64/f_X$ (or $128/f_X$ **Note**)

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

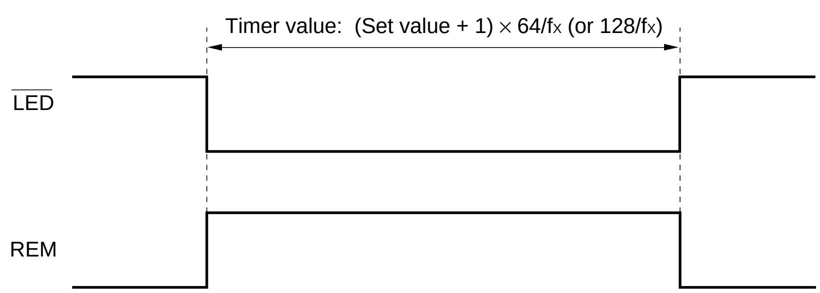
By setting the flag (t_9) that enables the timer output to 1, the timer can output its operation status from the $S_1/\overline{LED}$ and REM pins. The REM pin can also output the carrier while the timer is in operation.

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

	$S_1/\overline{LED}$ Pin	REM Pin
Timer operating	L	H (or carrier output **Note**)
Timer halted	H	L

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

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

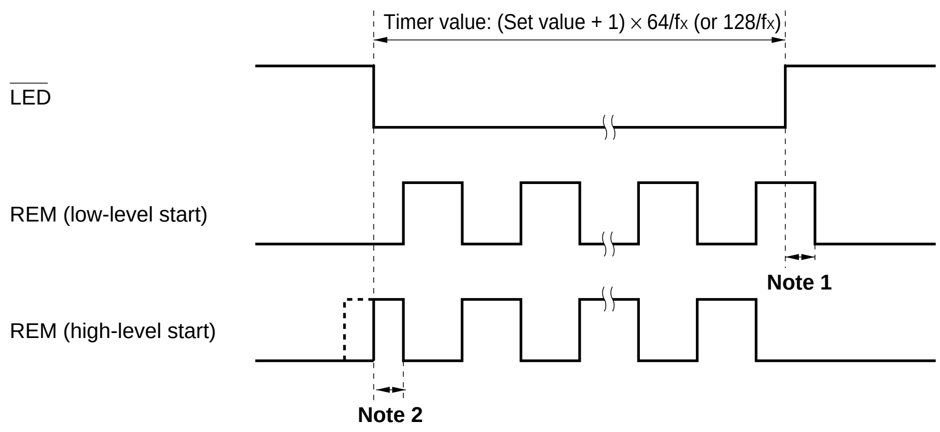


4.3 Carrier Output

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

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

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



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

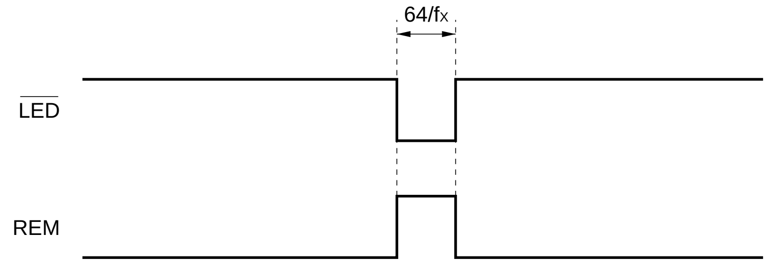
2. Error of carrier high-level width: 0 to “carrier high-level width”

4.4 Software Control of Timer Output

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

Figure 4-4. Pulse Output 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
      ⋮
```



5. STANDBY FUNCTION

5.1 Outline of Standby Function

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

In STOP mode, the system clock stops oscillation. At this time, the X_{IN} and X_{OUT} pins are fixed at a low level.

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

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

Table 5-1. Statuses in Standby Mode

<table>
<tr><th colspan="3"></th><th>STOP Mode</th><th>HALT Mode</th></tr>
<tr><td colspan="3">Setting instruction</td><td colspan="2">HALT instruction</td></tr>
<tr><td colspan="3">Clock oscillator</td><td>Oscillation stopped</td><td>Oscillation continued</td></tr>
<tr><td rowspan="7">Operation statuses</td><td colspan="2">CPU</td><td colspan="2">• Operation halted</td></tr>
<tr><td colspan="2">Data memory</td><td colspan="2">• Immediately preceding status retained</td></tr>
<tr><td colspan="2">Accumulator</td><td colspan="2">• Immediately preceding status retained</td></tr>
<tr><td rowspan="2">Flag</td><td>F</td><td colspan="2">• 0 (when 1, the flag does not enter the standby mode.)</td></tr>
<tr><td>CY</td><td colspan="2">• Immediately preceding status retained</td></tr>
<tr><td colspan="2">Port register</td><td colspan="2">• Immediately preceding status retained</td></tr>
<tr><td colspan="2">Timer</td><td>• Operation halted
(The count value is reset to “0”)</td><td>• Operable</td></tr>
</table>

Cautions 1. Write the NOP instruction as the first instruction after STOP mode is released.

2. When standby mode is released, the status flag (F) is set (1).

3. If, at the point the standby mode has been set, its release condition is met, then the system does not enter the standby mode. However, the status flag (F) is set (1).

5.2 Standby Mode Setting and Release

The standby mode is set with the HALT #$b_3b_2b_1b_0$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 (0).

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

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

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

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

Example

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

Table 5-2. Addresses Executed After Standby Mode Release

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

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

<table>
<tr><th colspan="4">Operand Value of HALT Instruction</th><th rowspan="2">Setting Mode</th><th rowspan="2">Precondition for Setting</th><th rowspan="2">Release Conditions</th></tr>
<tr><th>b_3</th><th>b_2</th><th>b_1</th><th>b_0</th></tr>
<tr><td rowspan="3">0</td><td>0</td><td>0</td><td>0</td><td>STOP</td><td>All $K_{I/O}$ pins are high-level output.</td><td>High level is input to at least one of K_I pins.</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>STOP</td><td>All $K_{I/O}$ pins are high-level output.</td><td>High level is input to at least one of K_I pins.</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>STOP[Note 1]</td><td>The $K_{I/O0}$ pin is high-level output.</td><td>High level is input to at least one of K_I pins.</td></tr>
<tr><td rowspan="2">1</td><td colspan="3" rowspan="2">Any combination of $b_2b_1b_0$ above</td><td rowspan="2">STOP</td><td colspan="2">[The following conditions are added in addition to the above.]</td></tr>
<tr><td>—</td><td>High level is input to at least one of S_0, S_1 and S_2 pins[Note 2].</td></tr>
<tr><td>0/1</td><td>1</td><td>0</td><td>1</td><td>HALT</td><td>—</td><td>When the timer's down counter is 0</td></tr>
</table>

Notes 1. When setting HALT #×110B, configure a key matrix by using the $K_{I/O0}$ pin and the K_I pin so that an internal reset takes effect at the time of program runaway.

2. At least one of the S_0, S_1 and S_2 pins (the pin used for releasing the standby) 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.

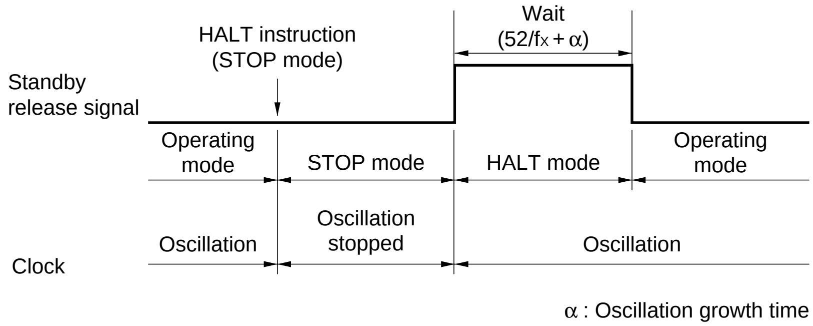
2. If STOP mode is set when the timer's down counter is not 0 (timer operating), the system enters STOP mode only after all the 10 bits of the timer's down counter and the timer output enable flag are cleared to 0.

3. Write the NOP instruction as the first instruction after STOP mode is released.

5.3 Standby Mode Release Timing

(1) STOP mode release timing

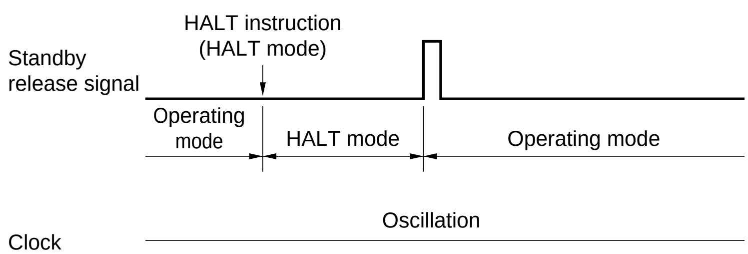
Figure 5-1. STOP Mode Release by Release Condition



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

(2) HALT mode release timing

Figure 5-2. HALT Mode Release by Release Condition



6. RESET

The system is reset by the following occurrences.

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

Table 6-1. Hardware Statuses After Reset

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

Notes 1. The following resets are available.
- Reset when executing the HALT instruction (when the operand value is illegal or does not satisfy the precondition)
- Reset when executing the RLZ instruction (when A = 0)
- Reset by stack pointer overflow or underflow

2. Value according to the K_I or S_2 pin status.
★ In order to prevent malfunction, do not input a high level to all the K_{I0} to K_{I3} pins when POC is released due to supply voltage startup (these pins can be left open. When leaving open, keep pull-down resistors connected).

7. POC CIRCUIT

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

Cautions 1. There are cases in which the POC circuit cannot detect a low power supply voltage of less than 1 ms. Therefore, if the power supply voltage has become low for a period of less than 1 ms, the POC circuit may malfunction because it does not generate an internal reset signal.

2. Clock oscillation is stopped by the resonator due to low power supply voltage before the POC circuit generates the internal reset signal. In this case, malfunction may result, for example when the power supply voltage is recovered after the oscillation is stopped. This type of phenomenon takes place because the POC circuit does not generate an internal reset signal (because the power supply voltage recovers before the low power supply voltage is detected) even though the clock has stopped. If, by any chance, a malfunction has taken place, remove the battery for a short time and put it back. In most cases, normal operation will be resumed.

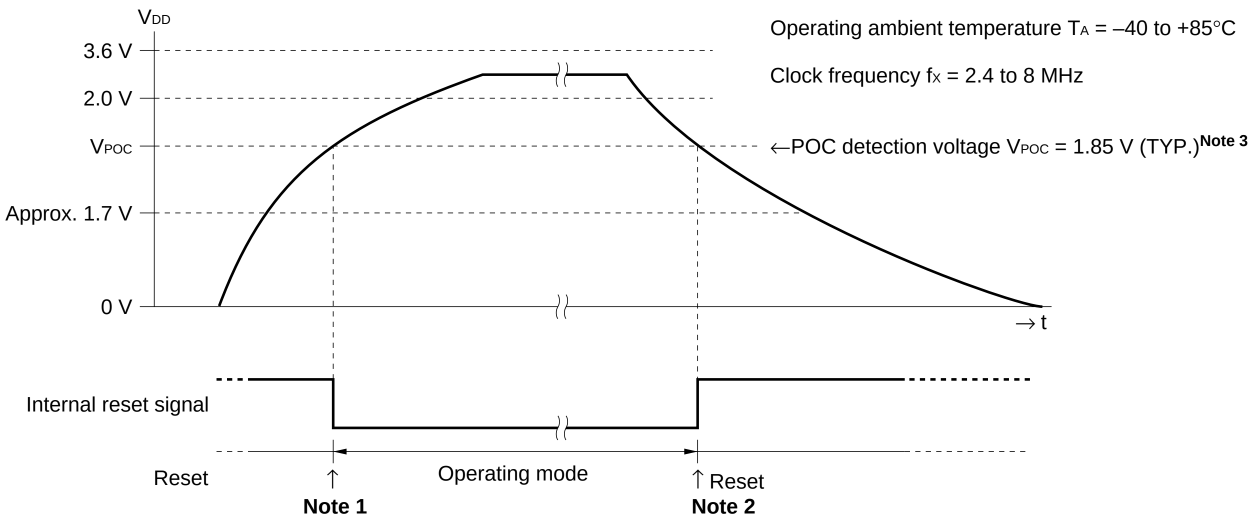
★ **3. In order to prevent malfunction, do not input a high level to all the K_{I0} to K_{I3} pins when POC is released due to supply voltage startup (these pins can be left open. When leaving open, keep pull-down resistors connected).**

7.1 Functions of POC Circuit

The POC circuit has the following functions.

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

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



★ **Notes 1.** In reality, an oscillation stabilization wait time must elapse before the circuit is switched to operating mode. The oscillation stabilization wait time is about $246/f_X$ to $694/f_X$ (about 70 to 190 μs, at f_X = 3.64 MHz).

2. For the POC circuit to generate an internal reset signal when the power supply voltage has fallen, it is necessary for the power supply voltage to be kept less than the V_{POC} for a period of 1 ms or more. Therefore, in reality, there is a time lag of up to 1 ms until the reset takes effect.

3. The POC detection voltage (V_{POC}) varies between approximately 1.7 to 2.0 V; thus, the reset may be canceled at a power supply voltage smaller than the assured range (V_{DD} = 2.0 to 3.6 V). However, as long as the conditions for operating the POC circuit are met, the actual lowest operating power supply voltage is lower than the POC detection voltage. Therefore, no malfunction occurs due to the shortage of power supply voltage. However, malfunction for such reasons as the clock not oscillating due to low power supply voltage may occur (refer to **Cautions 3**. in **7. POC CIRCUIT**).

7.2 Oscillation Check at Low Supply Voltage

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

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

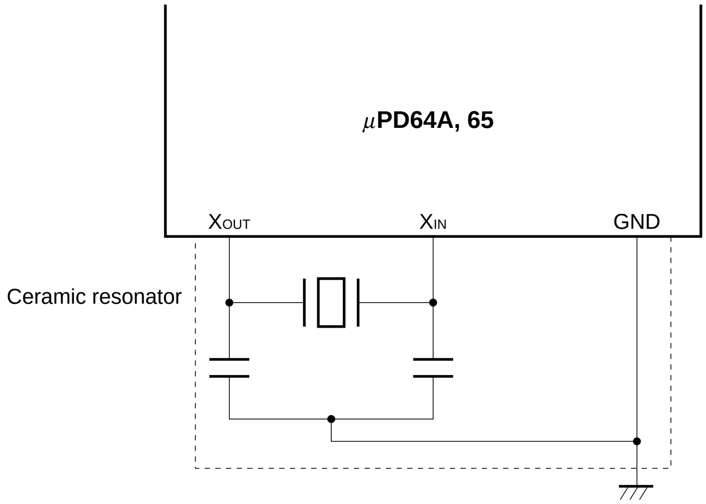
<2> Connect a power supply whose output voltage can be varied and then gradually raise the power supply voltage V_{DD} from 0 V (making sure to avoid V_{DD} > 3.6 V).

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

8. SYSTEM CLOCK OSCILLATOR

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

Figure 8-1. System Clock



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

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

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

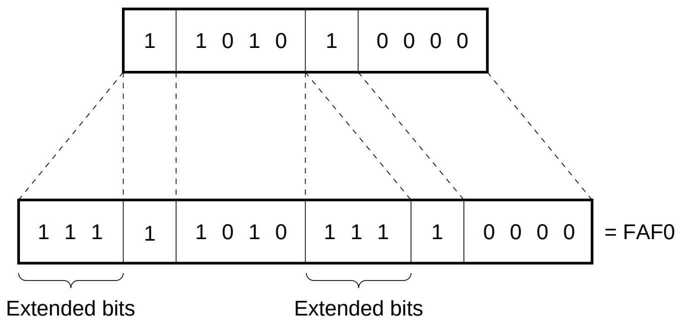
9. INSTRUCTION SET

9.1 Machine Language Output by Assembler

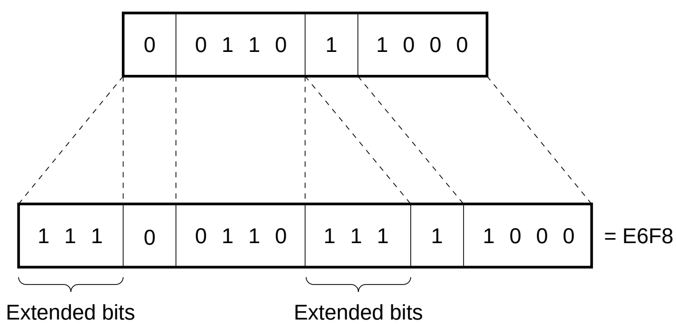
The bit length of the machine language of this product is 10 bits per word. However, the machine language output by the assembler is extended to 16 bits per word. As shown in the example below, the extension is made by inserting 3 extended bits (111) in two locations.

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

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



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



9.2 Circuit Symbol Description

A:	Accumulator
ASR:	Address stack register
addr:	Program memory address
CY:	Carry flag
data4:	4-bit immediate data
data8:	8-bit immediate data
data10:	10-bit immediate data
F:	Status flag
PC:	Program counter
Pn:	Port register pair (n = 0, 1, 3, 4)
P0n:	Port register (lower 4 bits)
P1n:	Port register (higher 4 bits)
ROMn:	Bit n of the program memory (n = 0 to 9)
Rn:	Register pair
R0n:	Data memory (general-purpose register; n = 0 to F)
R1n:	Data memory (general-purpose register; n = 0 to F)
SP:	Stack pointer
T:	Timer register
T0:	Timer register (lower 4 bits)
T1:	Timer register (higher 4 bits)
(×):	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) ← (A) ∧ (Rmn) m = 0, 1 n = 0 to F	1	1
	A, R1n	FAEn			CY ← A_3 • Rmn_3		
	A, @R0H	FAF0			(A) ← (A) ∧ $((P13), (R0))_{7\text{-}4}$ CY ← A_3 • ROM_7		
	A, @R0L	FBF0			(A) ← (A) ∧ $((P13), (R0))_{3\text{-}0}$ CY ← A_3 • ROM_3		
	A, #data4	FBF1	data4		(A) ← (A) ∧ data4 CY ← A_3 • $data4_3$	2	
ORL	A, R0n	FDEn			(A) ← (A) ∨ (Rmn) m = 0, 1 n = 0 to F	1	
	A, R1n	FCEn			CY ← 0		
	A, @R0H	FCF0			(A) ← (A) ∨ $((P13), (R0))_{7\text{-}4}$ CY ← 0		
	A, @R0L	FDF0			(A) ← (A) ∨ $((P13), (R0))_{3\text{-}0}$ CY ← 0		
	A, #data4	FDF1	data4		(A) ← (A) ∨ data4 CY ← 0	2	
XRL	A, R0n	F5En			(A) ← (A) $\veebar$ (Rmn) m = 0, 1 n = 0 to F	1	
	A, R1n	F4En			CY ← A_3 • Rmn_3		
	A, @R0H	F4F0			(A) ← (A) $\veebar$ $((P13), (R0))_{7\text{-}4}$ CY ← A_3 • ROM_7		
	A, @R0L	F5F0			(A) ← (A) $\veebar$ $((P13), (R0))_{3\text{-}0}$ CY ← A_3 • ROM_3		
	A, #data4	F5F1	data4		(A) ← (A) $\veebar$ data4 CY ← A_3 • $data4_3$	2	
INC	A	F4F3			(A) ← (A) + 1 if (A) = 0 CY ← 1 else CY ← 0	1	
RL	A	FCF3			(A_{n+1}) ← (A_n), (A_0) ← (A_3) CY ← A_3		
RLZ	A	FEF3			if A = 0 reset else (A_{n+1}) ← (A_n), (A_0) ← (A_3) CY ← A_3		

Input/output instructions

Mnemonic	Operand	Instruction Code			Operation	Instruction Length	Instruction Cycle
		1st Word	2nd Word	3rd Word			
IN	A, P0n	FFF8 + n	—	—	(A) ← (Pmn) m = 0, 1 n = 0, 1, 3, 4	1	1
	A, P1n	FEF8 + n	—	—	CY ← 0		
OUT	P0n, A	E5F8 + n	—	—	(Pmn) ← (A) m = 0, 1 n = 0, 1, 3, 4		
	P1n, A	E4F8 + n	—	—			
ANL	A, P0n	FBF8 + n	—	—	(A) ← (A) ∧ (Pmn) m = 0, 1 n = 0, 1, 3, 4		
	A, P1n	FAF8 + n	—	—	CY ← A_3 • 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) $\veebar$ (Pmn) m = 0, 1 n = 0, 1, 3, 4		
	A, P1n	F4F8 + n	—	—	CY ← A_3 • 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-P0n are handled in pairs.

Data transfer instructions

Mnemonic	Operand	Instruction Code			Operation	Instruction Length	Instruction Cycle
		1st Word	2nd Word	3rd Word			
MOV	A, R0n	FFEn			(A) ← (Rmn) m = 0, 1 n = 0 to F	1	1
	A, R1n	FEEn			CY ← 0		
	A, @R0H	FEF0			(A) ← $((P13), (R0))_{7-4}$ CY ← 0		
	A, @R0L	FFF0			(A) ← $((P13), (R0))_{3-0}$ 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-R0n) ← data8 n = 0 to F	2	1
	Rn, @R0	E7En	—	—	(R1n-R0n) ← ((P13), (R0)) n = 1 to F	1	

Remark Rn: R1n-R0n are handled in pairs.

Branch instructions

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

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

Subroutine instructions

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

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

Timer operation instructions

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

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

Other instructions

Mnemonic	Operand	Instruction Code			Operation	Instruction Length	Instruction Cycle
		1st Word	2nd Word	3rd Word			
HALT	#data4	E2F1	data4		Standby mode	2	1
STTS	#data4	E3F1	data4		If statuses match F ← 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 Operation 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: $(A) \leftarrow (A) \wedge (Rmn)$ m = 0, 1   n = 0 to F

$CY \leftarrow A_3 \bullet Rmn_3$

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: $(A) \leftarrow (A) \wedge ((P13), (R0))_{7\text{-}4}$ (in the case of ANL A, @R0H)

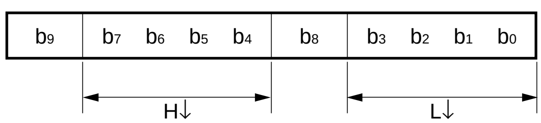
$CY \leftarrow A_3 \bullet ROM_7$

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

$CY \leftarrow A_3 \bullet ROM_3$

The accumulator contents and the program memory contents specified with the control register P13 and register pair R_{10}-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**



Valid bits at the time of accumulator operation

ANL A, #data4

<1> Instruction code:

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

<2> Cycle count: 1

<3> Function: $(A) \leftarrow (A) \wedge data4$

$CY \leftarrow A_3 \bullet data4_3$

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) $\vee$ (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) $\vee$ (P13), $(R0))_{7\text{-}4}$ (in the case of ORL A, @R0H)

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

CY ← 0

The accumulator contents and the program memory contents specified with the control register P13 and register pair 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) $\vee$ data4

CY ← 0

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

XRL A, R0n

XRL A, R1n

<1> Instruction code:

1	0 1 0 R_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 ← $A_3 \bullet Rmn_3$

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

XRL A, @R0H

XRL A, @R0L

<1> Instruction code:

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

<2> Cycle count: 1

<3> Function: $(A) \leftarrow (A) \veebar ((P13), (R0))_{7\text{-}4}$ (in the case of XRL A, @R0H)

$CY \leftarrow A_3 \bullet ROM_7$

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

$CY \leftarrow A_3 \bullet ROM_3$

The accumulator contents and the program memory contents specified with 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: $(A) \leftarrow (A) \veebar data4$

$CY \leftarrow A_3 \bullet data4_3$

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

INC A

<1> Instruction code:

1	0 1 0 0	1	0 0 1 1

<2> Cycle count: 1

<3> Function: $(A) \leftarrow (A) + 1$

If $A = 0$ $CY \leftarrow 1$

else $CY \leftarrow 0$

The accumulator contents are incremented (+1).

RL A

<1> Instruction code:

1	1 1 0 0	1	0 0 1 1

<2> Cycle count: 1

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

$CY \leftarrow A_3$

The accumulator contents are rotated counterclockwise bit by bit.

RLZ A

<1> Instruction code:

1	1 1 1 0	1	0 0 1 1

<2> Cycle count: 1

<3> Function: If $A = 0$ reset

else $(A_{n+1}) \leftarrow (A_n), (A_0) \leftarrow (A_3)$

$CY \leftarrow A_3$

The accumulator contents are rotated counterclockwise bit by bit.
If A = 0H at the time of instruction execution, an internal reset takes effect.

9.5 I/O Instructions

IN A, P0n
IN A, P1n

<1> Instruction code:

1	1	1	1	P_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) into 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) $\wedge$ (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) $\vee$ (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) $\veebar$ (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}-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) ← $((P13), (R0))_{7-4}$

CY ← 0

The higher 4 bits (b_7 b_6 b_5 b_4) of the program memory specified with 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) ← $((P13), (R0))_{3-0}$

CY ← 0

The lower 4 bits (b_3 b_2 b_1 b_0) of the program memory specified with control register P13 and register pair R_{10}-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}

Lower column

Higher column

MOV Rn, @R0

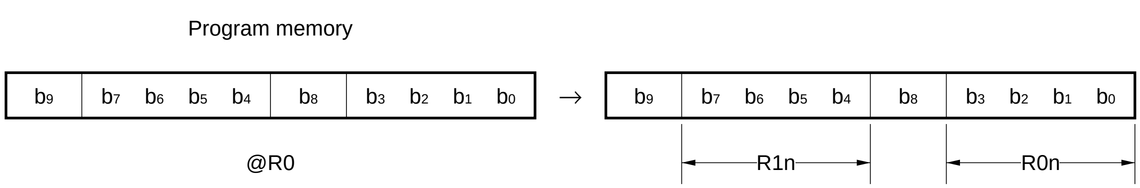
<1> Instruction code:

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



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

9.7 Branch Instructions

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

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

JMP addr

<1> Instruction code:

page 0	0	1 0 0 0	1	0 0 0 1
	a_9	a_7 a_6 a_5 a_4	a_8	a_3 a_2 a_1 a_0
page 1	0	1 0 0 1	1	0 0 0 1

<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
	a_9	a_7 a_6 a_5 a_4	a_8	a_3 a_2 a_1 a_0
page 1	0	1 0 1 0	1	0 0 0 1

<2> Cycle count: 1

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

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

JNC addr

<1> Instruction code:

page 0	0	1 1 0 1	1	0 0 0 1
	a_9	a_7 a_6 a_5 a_4	a_8	a_3 a_2 a_1 a_0
page 1	0	1 0 1 1	1	0 0 0 1

<2> Cycle count: 1

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

If the carry flag CY is cleared (0), a jump is made to the address specified with addr (a_9 to a_0).

JF addr

<1> Instruction code:

page 0	0	1 1 1 0	1	0 0 0 1
	a_9	a_7 a_6 a_5 a_4	a_8	a_3 a_2 a_1 a_0
page 1	1	0 0 0 0	1	0 0 0 1

<2> Cycle count: 1

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

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

JNF addr

<1> Instruction code:

page 0	0	1 1 1 1	1	0 0 0 1
	a_9	a_7 a_6 a_5 a_4	a_8	a_3 a_2 a_1 a_0
page 1	1	0 0 0 1	1	0 0 0 1

<2> Cycle count: 1

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

If the status flag F is cleared (0), a jump is made to the address specified with addr (a_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.

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

CALL addr

<1> Instruction code:

0	0 1 1 0	1	0 0 1 0

page 0 | 0 | 1 0 0 0 | 1 | 0 0 0 1 | , page 1 | 0 | 1 0 0 1 | 1 | 0 0 0 1 |

a_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

The stack pointer value is incremented (+1) and the program counter value is saved to the address stack register. The address specified with the operand addr (a_9 to a_0) is then entered in the program counter. If a carry is generated when the stack pointer value is incremented (+1), an internal reset takes effect.

RET

<1> Instruction code:

0	1 0 0 0	1	0 0 1 0

<2> Cycle count: 1

<3> Function: PC ← ASR
SP ← SP – 1

The value saved in the address stack register is restored to the program counter. The stack pointer is then decremented (–1).
If a borrow is generated when the stack pointer value is decremented (–1), an internal reset takes effect.

9.9 Timer Operation Instructions

MOV A, T0
MOV A, T1

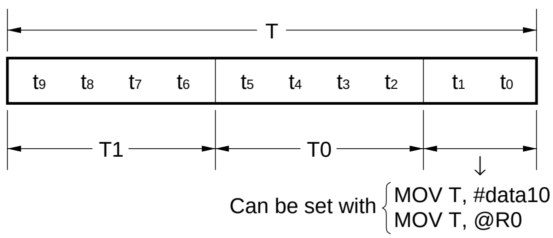
<1> Instruction code:

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

<2> Cycle count: 1

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

The timer 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 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 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 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 timer register T (t_9-t_0).

Remark The timer time is set with (set value + 1) × 64/f_X (or 128/f_X).

MOV T, @R0

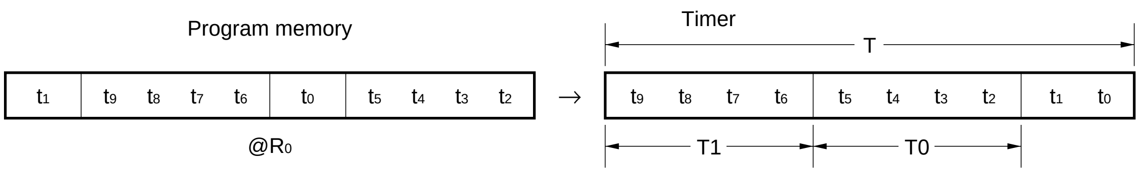
<1> Instruction code: 0 | 0 1 1 1 | 1 | 1 1 1 1

<2> Cycle count: 1

<3> Function: (T) ← ((P13), (R0))

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

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



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

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

9.10 Other Instructions

HALT #data4

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

0 | 0 0 0 0 | 0 | d_3 d_2 d_1 d_0

<2> Cycle count: 1

<3> Function: Standby mode

Places the CPU in standby mode.

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

STTS R0n

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

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

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

STTS #data4

<1> Instruction code:

0	0 0 1 1	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

The S_0, S_1, $K_{I/O}$, K_I, and TIMER statuses are compared with the immediate data contents. If at least one of the statuses matches the bits that have been set, status flag F is set (1).
If none of them match, status flag F is cleared (0).

SCAF (Set Carry If 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

The carry flag CY is set (1) if the accumulator contents are F_H.
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 Quasi Directives

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

10.1.1 OPTION and ENDOP quasi directives

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

Format

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

10.1.2 Mask option definition quasi directive

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

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

Example

Symbol field	Mnemonic field	Operand field	Comment field
	OPTION		
	USEPOC		; POC circuit incorporated
	ENDOP		

Table 10-1. Mask Option Definition Quasi Directive

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

11. ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings (T_A = +25°C)

Parameter	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
		$\overline{\text{LED}}$	Peak value	–7.5	mA
			rms value	–5	mA
		Per $K_{I/O}$ pin	Peak value	–13.5	mA
			rms value	–9	mA
		Total of $\overline{\text{LED}}$ 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
		$\overline{\text{LED}}$	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)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Power supply voltage	V_{DD}	f_X = 2.4 to 8 MHz	2.0	3.0	3.6	V

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

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	S_2		$0.8V_{DD}$		V_{DD}	V
	V_{IH2}	$K_{I/O}$		$0.7V_{DD}$		V_{DD}	V
	V_{IH3}	K_I, S_0, S_1		$0.65V_{DD}$		V_{DD}	V
Input voltage, low	V_{IL1}	S_2		0		$0.2V_{DD}$	V
	V_{IL2}	$K_{I/O}$		0		$0.3V_{DD}$	V
	V_{IL3}	K_I, S_0, S_1		0		$0.15V_{DD}$	V
Input leakage current, high	I_{LIH1}	K_I $V_I = V_{DD}$, pull-down resistor not incorporated				3	μA
	I_{LIH2}	S_0, S_1, S_2 $V_I = V_{DD}$, pull-down resistor not incorporated				3	μA
Input leakage current, low	I_{LIL1}	K_I	V_I = 0 V			–3	μA
	I_{LIL2}	$K_{I/O}$	V_I = 0 V			–3	μA
	I_{LIL3}	S_0, S_1, S_2	V_I = 0 V			–3	μA
Output voltage, high	V_{OH1}	REM, $\overline{\text{LED}}$, $K_{I/O}$	I_{OH} = –0.3 mA	$0.8V_{DD}$			V
Output voltage, low	V_{OL1}	REM, $\overline{\text{LED}}$	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	30	70		μA
			V_{DD} = 3.0 V, V_{OL} = 2.2 V	100	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_{DDDR}	In STOP mode		0.9		3.6	V
Supply current	I_{DD1}	Operating mode	f_X = 8.0 MHz, V_{DD} = 3 V ±10%		0.8	1.6	mA
			f_X = 4.0 MHz, V_{DD} = 3 V ±10%		0.7	1.4	mA
	I_{DD2}	HALT mode	f_X = 8.0 MHz, V_{DD} = 3 V ±10%		0.75	1.5	mA
			f_X = 4.0 MHz, V_{DD} = 3 V ±10%		0.65	1.3	mA
	I_{DD3}	STOP mode	V_{DD} = 3 V ±10%		1.9	9.0	μA
			V_{DD} = 3 V ±10%, T_A = 25°C		1.9	5.0	μA

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

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Instruction execution time	t_{CY}			7.9		27	μs
K_I, S_0, S_1, S_2 high-level width	t_H			10			μs
		When standby mode is released	In HALT mode	10			μs
			In STOP mode	**Note**			μs

Note 10 + 52/f_X + oscillation growth time

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

POC Circuit (T_A = –40 to +85°C)

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

Note The voltage with which the POC circuit cancels an internal reset. If $V_{POC} < V_{DD}$, the internal reset is canceled.
From the time of $V_{POC} \geq V_{DD}$ until the internal reset takes effect, a lag 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.

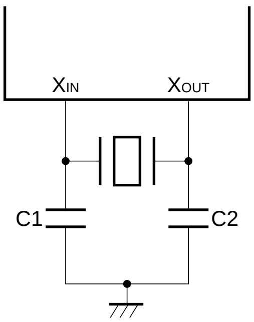
System Clock Oscillator Characteristics (T_A = –40 to +85°C, V_{DD} = 2.0 to 3.6 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Oscillation frequency (ceramic resonator)	f_X		2.4	3.64	8.0	MHz

Recommended Ceramic Resonator (T_A = –40 to +85°C)

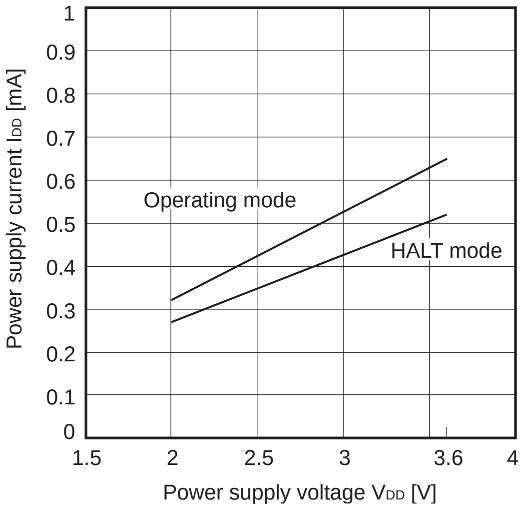
Manufacturer	Part Number	Frequency (MHz)	Recommended Constant		Power Supply Voltage [V]		Remark
			C1 [pF]	C2 [pF]	MIN.	MAX.	
TDK Corp.	FCR3.52MC5	3.52	Unnecessary (On-chip C type)		2.0	3.6	
	FCR3.58MC5	3.58					
	FCR3.64MC5	3.64					
	FCR3.84MC5	3.84					
	FCR4.0MC5	4.0					
	FCR6.0MC5	6.0					
	FCR8.0MC5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

External circuit example

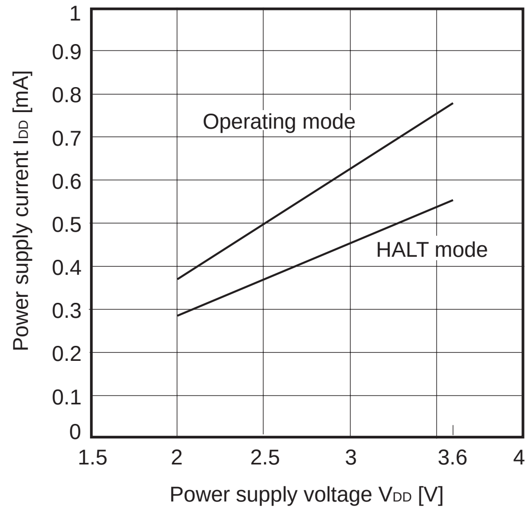


12. CHARACTERISTIC CURVES (REFERENCE VALUES)

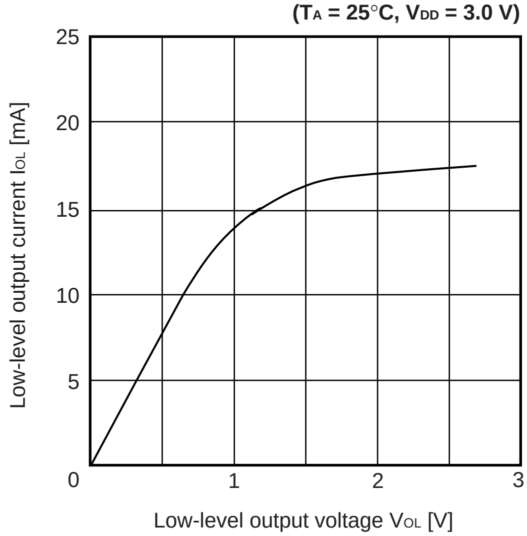
I_{DD} vs. V_{DD} (f_X = 4 MHz)
(T_A = 25°C)



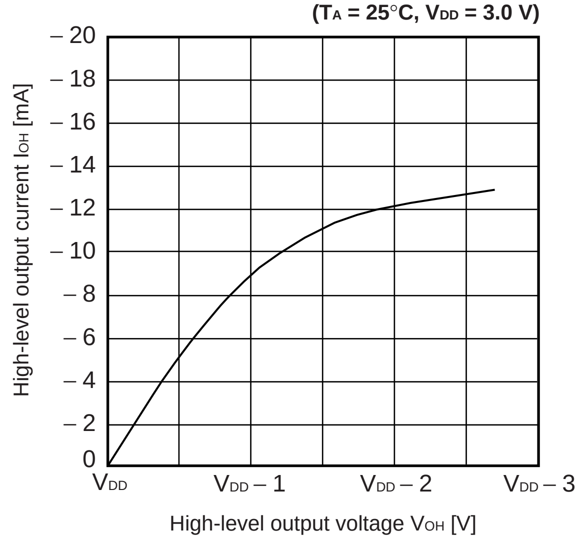
I_{DD} vs. V_{DD} (f_X = 8 MHz)
(T_A = 25°C)



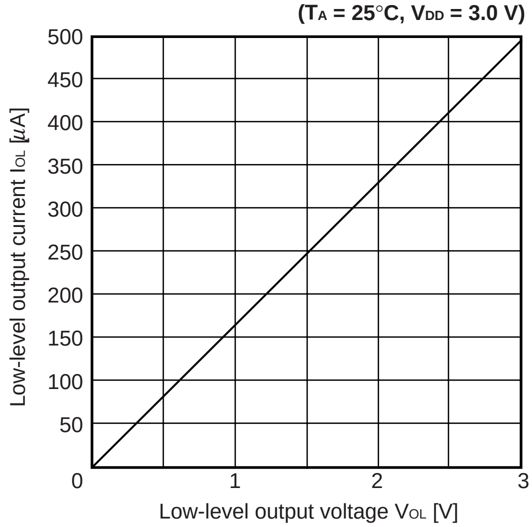
I_{OL} vs. V_{OL} (REM, $\overline{LED}$)
(T_A = 25°C, V_{DD} = 3.0 V)



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



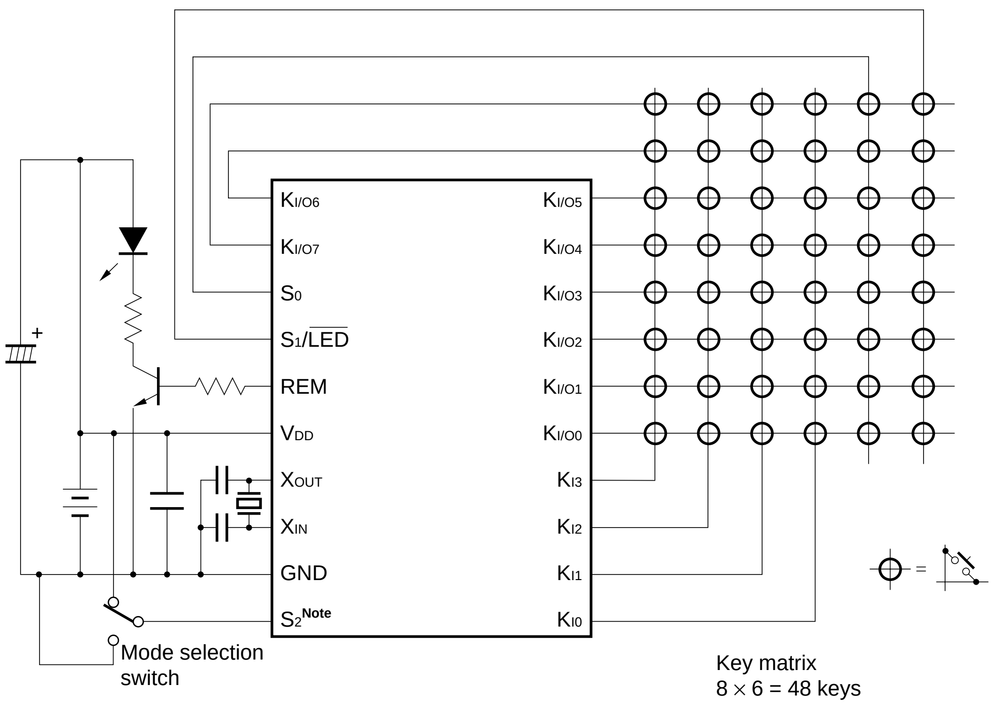
I_{OL} vs. V_{OL} ($K_{I/O}$)
(T_A = 25°C, V_{DD} = 3.0 V)



13. APPLICATION CIRCUIT EXAMPLE

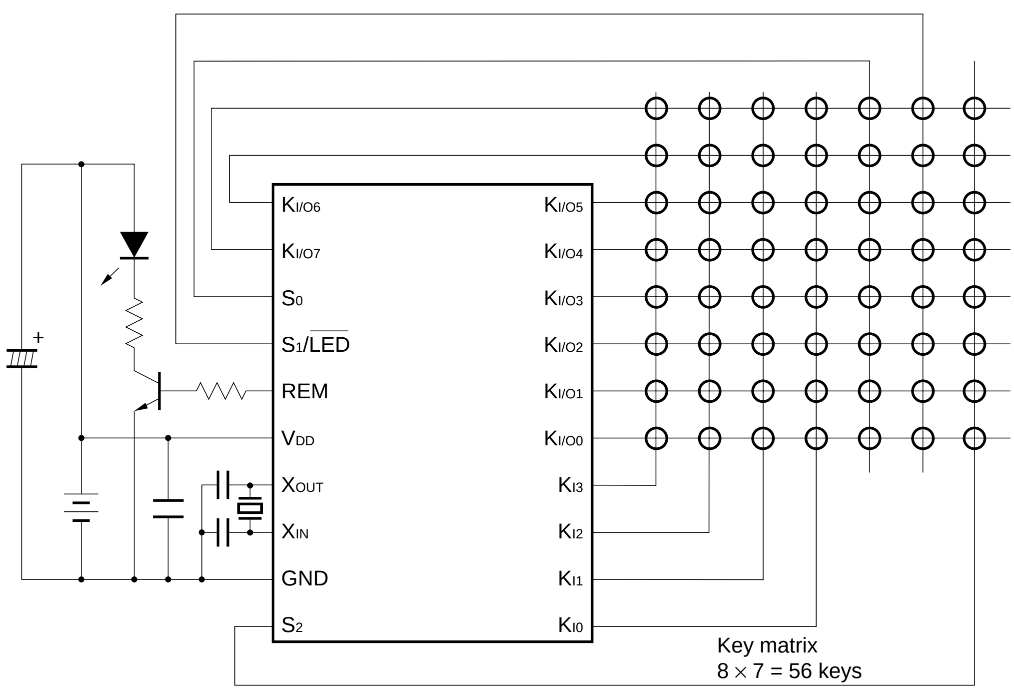
Example of application to system

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



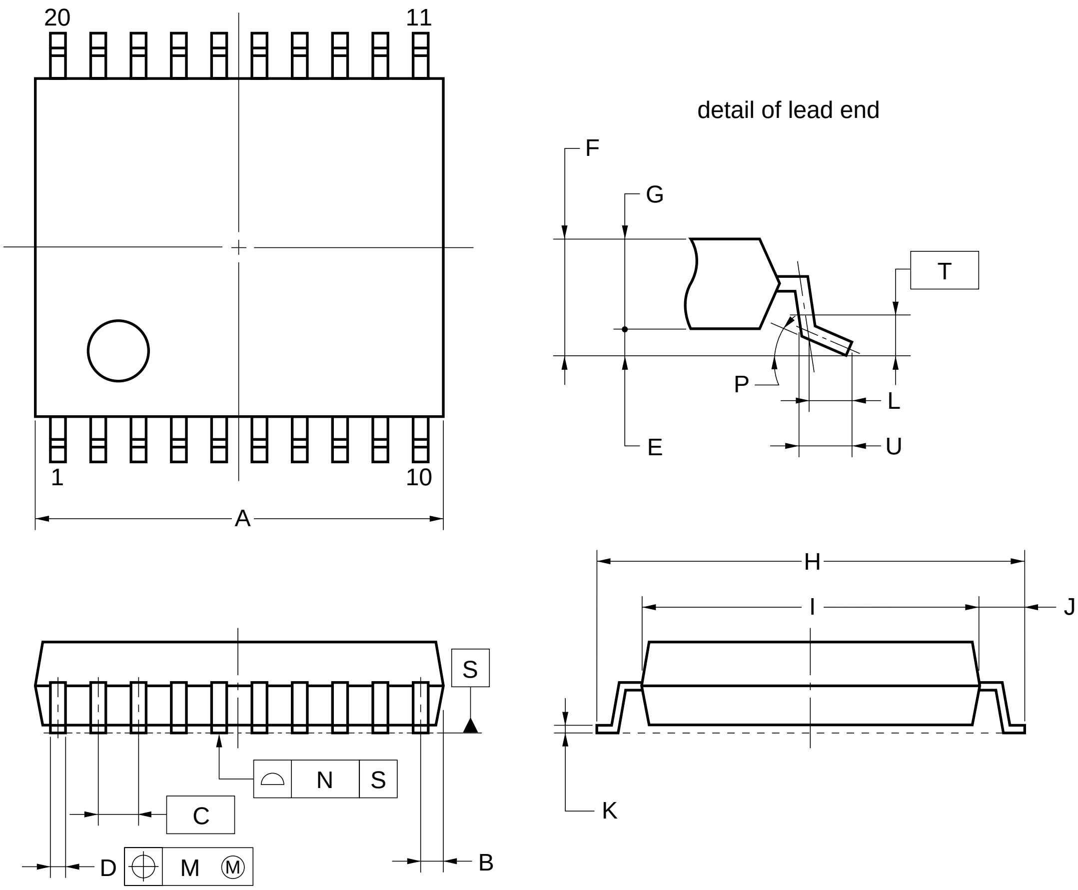
Note S_2: Set this pin to disabled when releasing STOP mode.

★ ▪ **Remote-control transmitter (56 keys accommodated)**



14. PACKAGE DRAWING

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



NOTE

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

ITEM	MILLIMETERS
A	6.65±0.15
B	0.475 MAX.
C	0.65 (T.P.)
D	$0.24^{+0.08}_{-0.07}$
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	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
T	0.25
U	0.6±0.15

S20MC-65-5A4-2

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

15. RECOMMENDED SOLDERING CONDITIONS

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

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

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

Table 15-1. Surface Mounting Type Soldering Conditions

μPD64AMC-×××-5A4: 20-pin plastic SSOP (7.62 mm (300))
μPD65MC-×××-5A4: 20-pin plastic SSOP (7.62 mm (300))

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

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

APPENDIX A. DEVELOPMENT TOOLS

Emulators are provided for the μPD64A and 65 emulation tools.

Hardware

★ • **Emulators (EB-65, EB-69[Note])**

These are tools used to emulate the μPD64A and 65.

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

Software

• **Assembler (AS6133)**

• This is a development tool for remote control transmitter software.

Part Number List of AS6133

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

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

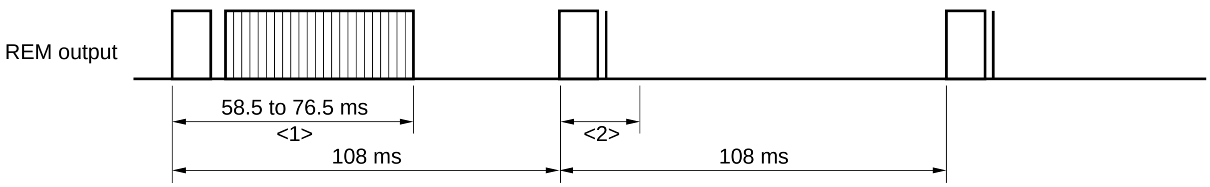
APPENDIX B. FUNCTIONAL COMPARISON BETWEEN μPD64A, 65 AND OTHER PRODUCTS

Item		μPD62	μPD62A	μPD64	μPD64A	μPD65
ROM capacity		512 × 10 bits	512 × 10 bits	1002 × 10 bits	1002 × 10 bits	2026 × 10 bits
RAM capacity		32 × 4 bits				
Stack		1 level (multiplexed with RF of RAM)				
Key matrix		8 × 6 = 48 keys			8 × 7 = 56 keys	
Key extended input		S_0, S_1			S_0, S_1, 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 = 2.4 to 8 MHz • f_X = 2.4 to 4 MHz (with POC circuit)	Ceramic oscillation • f_X = 2.4 to 8 MHz	
Timer	Clock	f_X/64, f_X/128				
	Count start	Writing count value				
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				
	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)				
Relationship 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			• Provided • Generates internal reset signal on detection	
	POC detection voltage	V_{POC} = 1.6 V (TYP.)	V_{POC} = 1.85 V (TYP.)	V_{POC} = 1.6 V (TYP.)	V_{POC} = 1.85 V (TYP.)	
Mask option		POC circuit only			None	
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	• 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 ambient temperature		• T_A = –40 to +85°C • T_A = –20 to +70°C (with POC circuit)	• T_A = –40 to +85°C	• T_A = –40 to +85°C • T_A = –20 to +70°C (with POC circuit)	T_A = –40 to +85°C	
★ Electrical specifications, recommended soldering conditions		Refer to the data sheet for each product.				
Package		20-pin plastic SSOP		• 20-pin plastic SOP • 20-pin plastic SSOP	20-pin plastic SSOP	
One-time PROM version		μPD6P4B			μPD6P5	

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

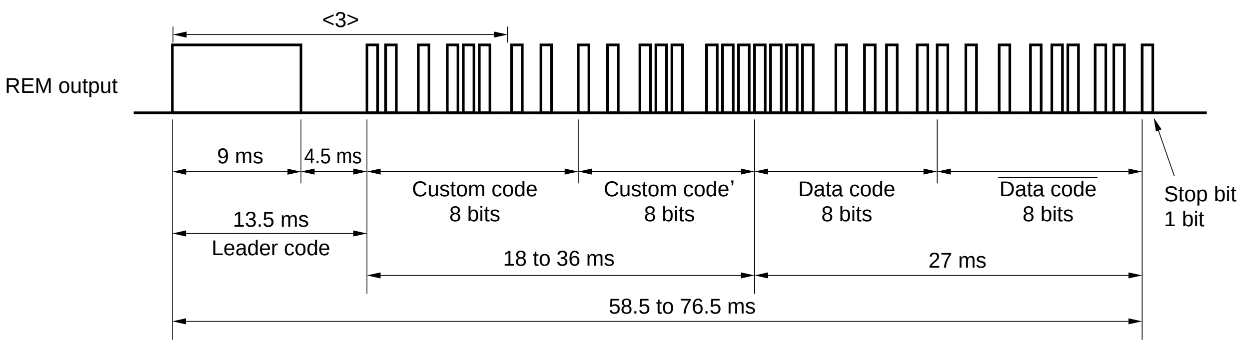
Caution **When using the NEC transmission format, please apply for a custom code at NEC.**

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

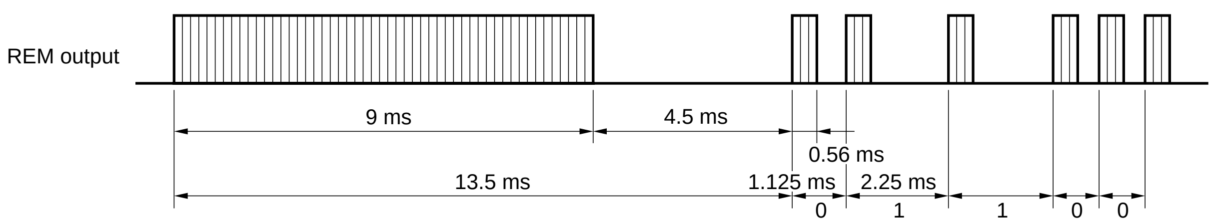


Remark If the key is repeatedly pressed, the power consumption of the infrared light-emitting diode (LED) can be reduced by sending the reader code and the stop bit from the second time.

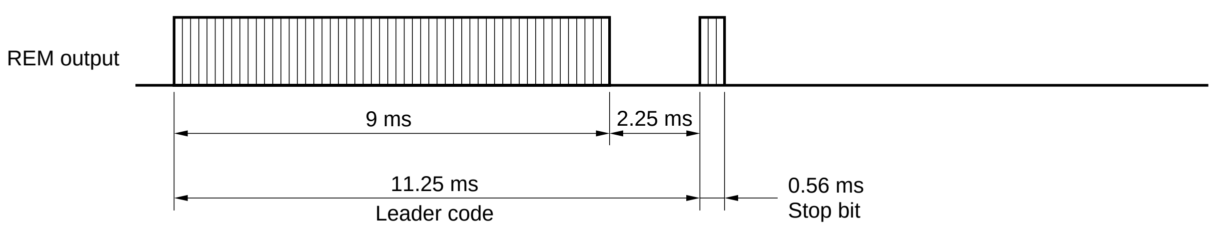
(2) Enlarged waveform of <1>



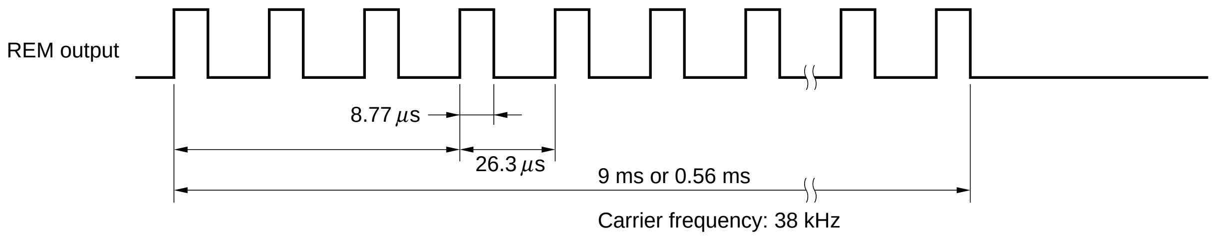
(3) Enlarged waveform of <3>



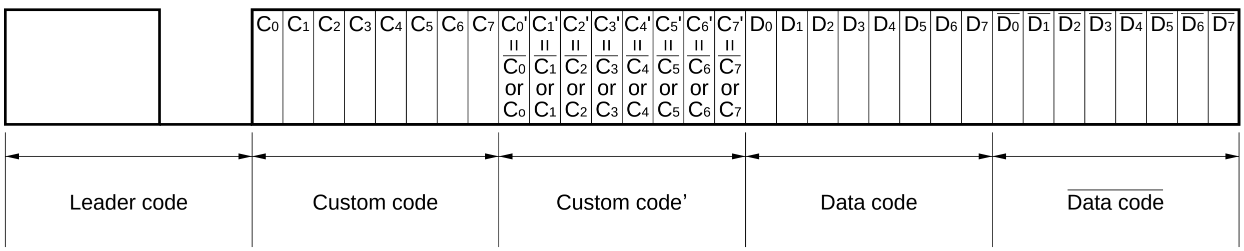
(4) Enlarged waveform of <2>



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



(6) Bit array of each code



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

[MEMO]

[MEMO]

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note:

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

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note:

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

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note:

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

Regional Information

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

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

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

NEC Electronics Inc. (U.S.)
Santa Clara, California
Tel: 408-588-6000
800-366-9782
Fax: 408-588-6130
800-729-9288

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

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

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

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

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

NEC Electronics (France) S.A.
Madrid Office
Madrid, Spain
Tel: 91-504-2787
Fax: 91-504-2860

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

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

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

NEC Electronics Singapore Pte. Ltd.
United Square, Singapore
Tel: 65-253-8311
Fax: 65-250-3583

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

NEC do Brasil S.A.
Electron Devices Division
Guarulhos-SP Brasil
Tel: 55-11-6462-6810
Fax: 55-11-6462-6829

J00.7

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