# Renesas Electronics America Inc - UPD78F1011GC-GAD-AX Datasheet

**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Active |
| Core Processor | 78K/0R |
| Core Size | 16-Bit |
| Speed | 20MHz |
| Connectivity | 3-Wire SIO, I²C, LINbus, UART/USART |
| Peripherals | DMA, LVD, POR, PWM, WDT |
| Number of I/O | 66 |
| Program Memory Size | 96KB (96K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 6K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.8V ~ 5.5V |
| Data Converters | A/D 12x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 80-LQFP |
| Supplier Device Package | - |
| Purchase URL | https://www.e-xfl.com/product-detail/renesas-electronics-america/upd78f1011gc-gad-ax |

# How to Use This Manual

**Readers**

This manual is intended for user engineers who wish to understand the functions of the 78K0R/Kx3-L and design and develop application systems and programs for these devices.
The target products are as follows.

- 78K0R/KC3-L: $\mu$PD78F1000, 78F1001, 78F1002, 78F1003
- 78K0R/KD3-L: $\mu$PD78F1004, 78F1005, 78F1006
- 78K0R/KE3-L: $\mu$PD78F1007, 78F1008, 78F1009
- 78K0R/KF3-L: $\mu$PD78F1010, 78F1011, 78F1012, 78F1027, 78F1028
- 78K0R/KG3-L: $\mu$PD78F1013, 78F1014, 78F1029, 78F1030

**Purpose**

This manual is intended to give users an understanding of the functions described in the **Organization** below.

**Organization**

The 78K0R/Kx3-L manual is separated into two parts: this manual and the instructions edition (common to the 78K0R Microcontroller).

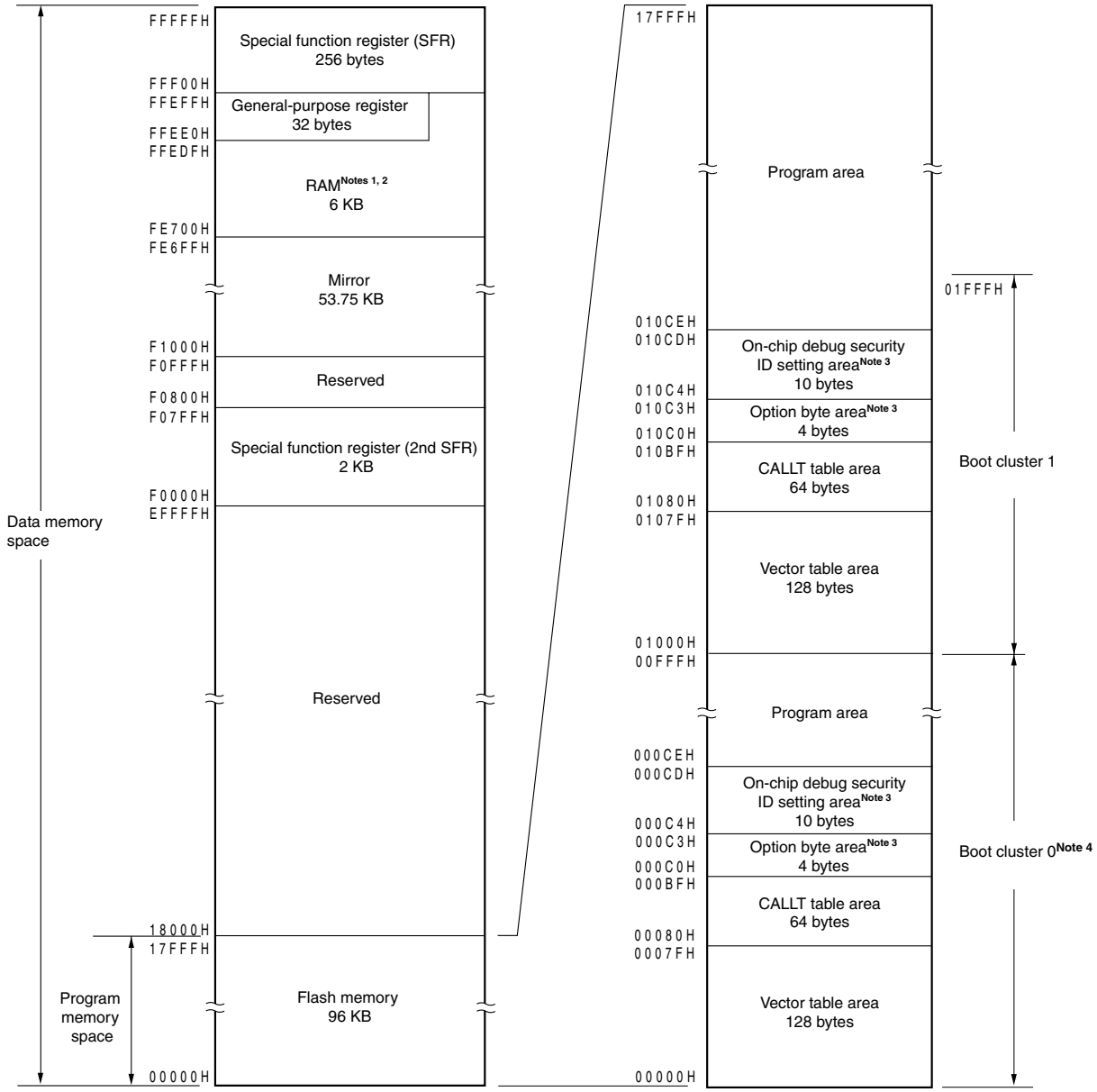
| **78K0R/Kx3-L User's Manual (This Manual)** | **78K0R Microcontroller User's Manual Instructions** |
|---|---|
| • Pin functions<br>• Internal block functions<br>• Interrupts<br>• Other on-chip peripheral functions<br>• Electrical specifications | • CPU functions<br>• Instruction set<br>• Explanation of each instruction |

**How to Read This Manual**

It is assumed that the readers of this manual have general knowledge of electrical engineering, logic circuits, and microcontrollers.

- To gain a general understanding of functions:
  → Read this manual in the order of the **CONTENTS**. The mark "<R>" shows major revised points. The revised points can be easily searched by copying an "<R>" in the PDF file and specifying it in the "Find what:" field.
- How to interpret the register format:
  → For a bit number enclosed in angle brackets, the bit name is defined as a reserved word in the RA78K0R, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0R.
- To know details of the 78K0R Microcontroller instructions:
  → Refer to the separate document **78K0R Microcontroller Instructions User's Manual (U17792E)**.

**Figure 4-6. Memory Map (μPD78F1011, 78F1013)**




**Notes 1.** While using the self-programming function, the area FFE20H to FFEFFH cannot be used as stack memory.

**2.** Instructions can be executed from the RAM area excluding the general-purpose register area.

**3.** When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

When boot swap is used: Set the option bytes to 000C0H to 000C3H and 010C0H to 010C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 010C4H to 010CDH.

**4.** Writing boot cluster 0 can be prohibited depending on the setting of security (see **26.7 Security Setting**).

### 4.4.8 Based indexed addressing

**[Function]**

Based indexed addressing uses the contents of a register pair specified with the instruction word as the base address, and the content of the B register or C register similarly specified with the instruction word as offset address. The sum of these values is used to specify the target address.

**[Operand format]**

| Identifier | Description |
|---|---|
| – | [HL+B], [HL+C] (only the space from F0000H to FFFFFH is specifiable) |
| – | ES:[HL+B], ES:[HL+C] (higher 4-bit addresses are specified by the ES register) |

**Figure 4-46. Example of [HL+B], [HL+C]**

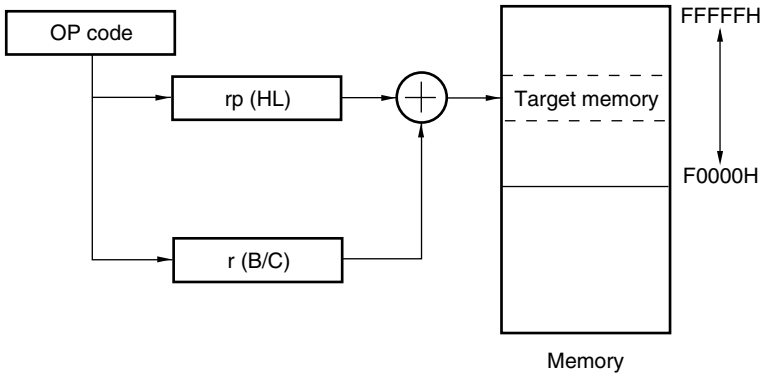



**Figure 4-47. Example of ES:[HL+B], ES:[HL+C]**

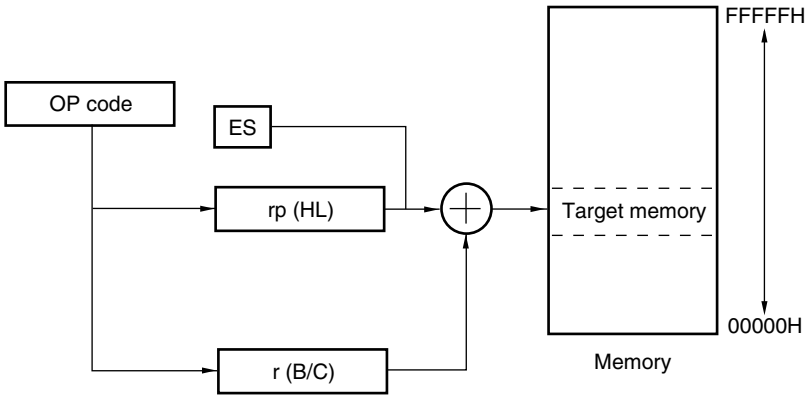
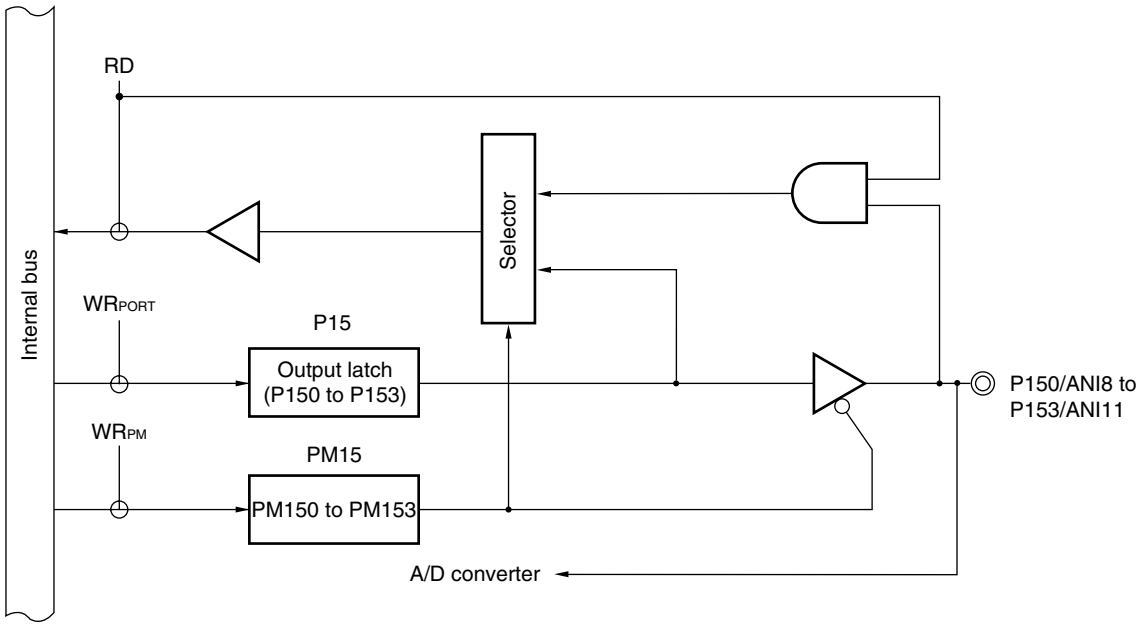
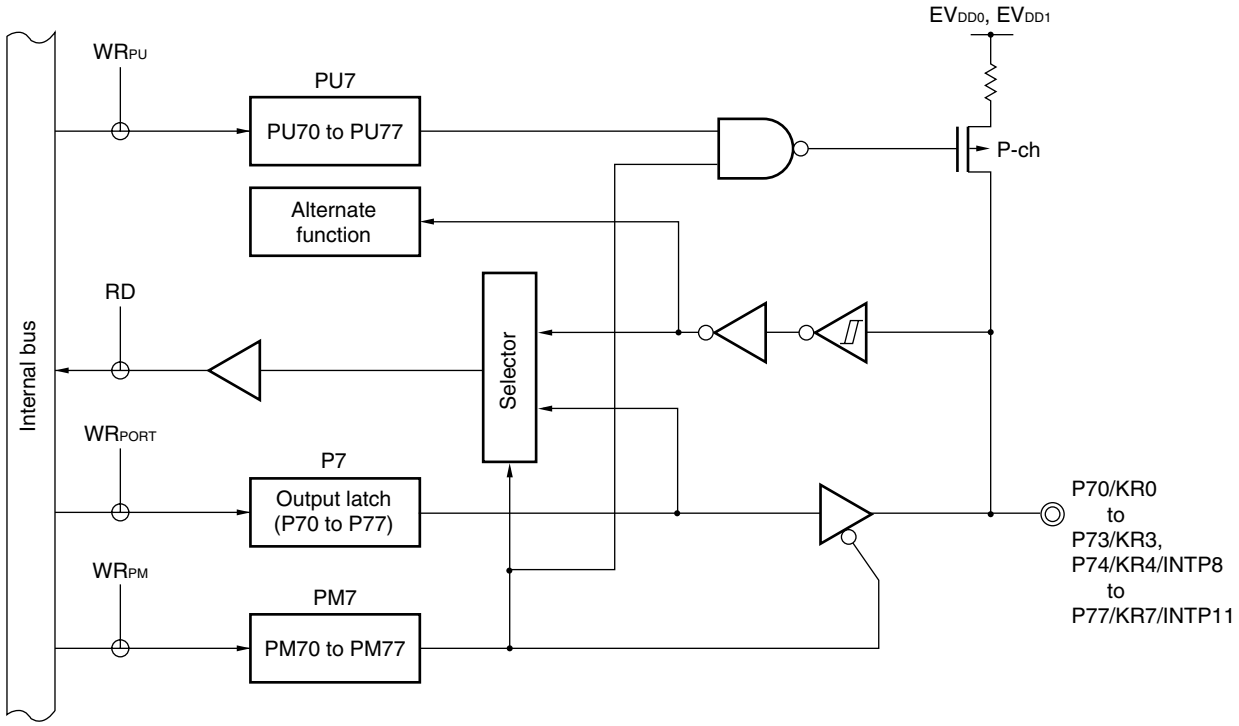


**Figure 5-27. Block Diagram of P150 to P153**




P15: Port register 15
PM15: Port mode register 15
RD: Read signal
WR××: Write signal

**Figure 6-37. Block Diagram of P70 to P77**




P7: Port register 7
PU7: Pull-up resistor option register 7
PM7: Port mode register 7
RD: Read signal
WR××: Write signal

### 6.2.16 Port 15

| | 78K0R/KF3-L ($\mu$PD78F10xx: xx = 10 to 12, 27,28) | 78K0R/KG3-L ($\mu$PD78F10xx: xx = 13, 14, 29, 30) |
|---|---|---|
| P150/ANI8 | √ | √ |
| P151/ANI9 | √ | √ |
| P152/ANI10 | √ | √ |
| P153/ANI11 | √ | √ |
| P154/ANI12 | – | √ |
| P155/ANI13 | – | √ |
| P156/ANI14 | – | √ |
| P157/ANI15 | – | √ |

Port 15 is an I/O port with an output latch. Port 15 can be set to the input mode or output mode in 1-bit units using port mode register 15 (PM15).

This port can also be used for A/D converter analog input.

To use P150/ANI8 to P157/ANI15 as digital input pins, set them in the digital I/O mode by using the A/D port configuration register (ADPC) and in the input mode by using the PM15 register. Use these pins starting from the lower bit.

To use P150/ANI8 to P157/ANI15 as digital output pins, set them in the digital I/O mode by using the ADPC register and in the output mode by using the PM15 register.

**Table 6-6. Setting Functions of P150/ANI8 to P157/ANI15 Pins**

| ADPC Register | PM15 Register | ADS Register | P150/ANI8 to P157/ANI15 Pins |
|---|---|---|---|
| Digital I/O selection | Input mode | – | Digital input |
| | Output mode | – | Digital output |
| Analog input selection | Input mode | Selects ANI. | Analog input (to be converted) |
| | | Does not select ANI. | Analog input (not to be converted) |
| | Output mode | Selects ANI. | Setting prohibited |
| | | Does not select ANI. | |

All P150/ANI8 to P157/ANI15 are set in the digital input mode when the reset signal is generated.

Figure 6-51 shows a block diagram of port 15.

**Caution See 3.2.16 $AV_{REF}$, $AV_{SS}$, $V_{DD}$, $EV_{DD0}$, $EV_{DD1}$, $V_{SS}$, $EV_{SS0}$, $EV_{SS1}$ for the voltage to be applied to the $AV_{REF}$ pin when using port 15 as a digital I/O.**

**Figure 6-54. Format of Port Register (78K0R/KF3-L)**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | After reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| P0 | 0 | P06 | P05 | P04 | P03 | P02 | 0 | 0 | FFF00H | 00H (output latch) | R/W |
| P1 | P17 | P16 | P15 | P14 | P13 | P12 | P11 | P10 | FFF01H | 00H (output latch) | R/W |
| P2 | P27 | P26 | P25 | P24 | P23 | P22 | P21 | P20 | FFF02H | 00H (output latch) | R/W |
| P3 | 0 | 0 | 0 | 0 | 0 | 0 | P31 | P30 | FFF03H | 00H (output latch) | R/W |
| P4 | P47 | P46 | P45 | P44 | P43 | P42 | P41 | P40 | FFF04H | 00H (output latch) | R/W |
| P5 | 0 | 0 | P55 | P54 | P53 | P52 | P51 | P50 | FFF05H | 00H (output latch) | R/W |
| P6 | P67 | P66 | P65 | P64 | P63 | P62 | P61 | P60 | FFF06H | 00H (output latch) | R/W |
| P7 | P77 | P76 | P75 | P74 | P73 | P72 | P71 | P70 | FFF07H | 00H (output latch) | R/W |
| P9 | 0 | 0 | 0 | 0 | 0 | 0 | P91 | P90 | FFF09H | 00H (output latch) | R/W |
| P11 | 0 | 0 | 0 | 0 | 0 | 0 | P111 | P110 | FFF0BH | 00H (output latch) | R/W |
| P12 | 0 | 0 | 0 | P124 | P123 | P122 | P121 | P120 | FFF0CH | Undefined | R/W[Note] |
| P13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | P130 | FFF0DH | 00H (output latch) | R/W |
| P14 | 0 | 0 | 0 | P144 | P143 | P142 | 0 | P140 | FFF0EH | 00H (output latch) | R/W |
| P15 | 0 | 0 | 0 | 0 | P153 | P152 | P151 | P150 | FFF0FH | 00H (output latch) | R/W |

| Pmn | m = 0 to 7, 9, 11 to 15; n = 0 to 7 | |
|---|---|---|
| | Output data control (in output mode) | Input data read (in input mode) |
| 0 | Output 0 | Input low level |
| 1 | Output 1 | Input high level |

**Note** P121 to P124 are read-only.

**(8) Operation speed mode control register (OSMC)**

This register is used to reduce power consumption by stopping as many unnecessary clock functions as possible.

The FLPC and FSEL bits can be used to control the step-up circuit of the flash memory for high-speed operation.

If the microcontroller operates on a system clock of 10 MHz or more, set this register to 01B.

If the microcontroller operates at low speed on a system clock of 10 MHz or less, power consumption can be reduced, because the voltage booster can be stopped by setting this register to its initial value, 00B. Furthermore, when CPU operates with the system clock of 1 MHz, the power consumption can be further reduced by setting the FLPC bit to 1.

If the RTCLPC bit is set to 1 and real-time counter is operating, current consumption can be reduced, because the circuit that synchronizes the clock to the peripheral functions, except the real-time counter[Note 1], is stopped in STOP mode and in HALT mode while subsystem clock[Note 2] is selected as CPU clock.

The OSMC register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Notes 1.** 40-pin product of the 78K0R/KC3-L does not have real-time counter.

**2.** 40-pin product of the 78K0R/KC3-L does not have subsystem clock.

**Figure 7-12. Format of Operation Speed Mode Control Register (OSMC)**

Address: F00F3H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| OSMC | RTCLPC | 0 | 0 | 0 | 0 | 0 | FLPC | FSEL |

| FLPC | FSEL | $f_{CLK}$ frequency selection |
|---|---|---|
| 0 | 0 | Operates at a frequency of 10 MHz or less (default). |
| 0 | 1 | Operates at a frequency higher than 10 MHz. |
| 1 | 0 | Operates at a frequency of 1 MHz. |
| 1 | 1 | Setting prohibited |

| RTCLPC[Note] | Setting in STOP mode and in HALT mode while subsystem clock is selected as CPU clock |
|---|---|
| 0 | Enables supply of subsystem clock to peripheral functions<br>(See Table 20-1 for peripheral functions whose operations are enabled.) |
| 1 | Stops supply of subsystem clock to peripheral functions other than real-time counter |

**Note** 40-pin product of the 78K0R/KC3-L does not have RTCLPC bit. Be sure to clear RTCLPC bit to 0.

**Cautions 1. Write "1" to the FSEL bit before the following two operations.**
- **Changing the clock prior to dividing $f_{CLK}$ to a clock other than $f_{IH}$.**
- **Operating the DMA controller.**

**2. The CPU waits (140.5 clock ($f_{CLK}$)) when "1" is written to the FSEL bit. Interrupt requests issued during a wait will be suspended. However, counting the oscillation stabilization time of $f_X$ can continue even while the CPU is waiting.**

**3. To increase $f_{CLK}$ to 10 MHz or higher, set the FSEL bit to "1", then change $f_{CLK}$ after three or more clocks have elapsed.**

**4. To set the FSEL bit to 0, set $f_{CLK}$ to 10 MHz or less in advance.**

**5. Set FSEL = 0 to shift to STOP mode while $V_{DD} \leq 2.7$ V.**

(Cautions are given on the next page.)

**(2) Timer clock select register m (TPSm)**

The TPSm register is a 16-bit register that is used to select two types of operation clocks (CKm0, CKm1) that are commonly supplied to each channel. CKm1 is selected by bits 7 to 4 of the TPSm register, and CKm0 is selected by bits 3 to 0.

Rewriting of the TPSm register during timer operation is possible only in the following cases.

If the PRSm00 to PRSm03 bits can be rewritten:

All channels for which CKm0 is selected as the operation clock (CKSmn = 0) are stopped (TEmn = 0).

If the PRSm10 to PRSm13 bits can be rewritten:

All channels for which CKm1 is selected as the operation clock (CKSmn = 1) are stopped (TEmn = 0).

The TPSm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TPSm register can be set with an 8-bit memory manipulation instruction with TPSmL.

Reset signal generation clears this register to 0000H.

**Table 8-4. Operations from Count Operation Enabled State to Timer/counter Register mn (TCRmn) Count Start**

| Timer operation mode | Operation when TSmn = 1 is set |
|---|---|
| • Interval timer mode | No operation is carried out from start trigger detection (TSmn=1) until count clock generation.<br>The first count clock loads the value of the TDRmn register to the TCRmn register and the subsequent count clock performs count down operation (see **8.3 (6) (a) Start timing in interval timer mode**). |
| • Event counter mode | Writing 1 to the TSmn bit loads the value of the TDRmn register to the TCRmn register.<br>The subsequent count clock performs count down operation.<br>The external trigger detection selected by the STSmn2 to STSmn0 bits in the TMRmn register does not start count operation (see **8.3 (6) (b) Start timing in event counter mode**). |
| • Capture mode | No operation is carried out from start trigger detection until count clock generation.<br>The first count clock loads 0000H to the TCRmn register and the subsequent count clock performs count up operation (see **8.3 (6) (c) Start timing in capture mode**). |
| • One-count mode | The waiting-for-start-trigger state is entered by writing 1 to the TSmn bit while the timer is stopped (TEmn = 0).<br>No operation is carried out from start trigger detection until count clock generation.<br>The first count clock loads the value of the TDRmn register to the TCRmn register and the subsequent count clock performs count down operation (see **8.3 (6) (d) Start timing in one-count mode**). |
| • Capture & one-count mode | The waiting-for-start-trigger state is entered by writing 1 to the TSmn bit while the timer is stopped (TEmn = 0).<br>No operation is carried out from start trigger detection until count clock generation.<br>The first count clock loads 0000H to the TCRmn register and the subsequent count clock performs count up operation (see **8.3 (6) (e) Start timing in capture & one-count mode**). |

**(11) Month count register (MONTH)**

The MONTH register is an 8-bit register that takes a value of 1 to 12 (decimal) and indicates the count value of months.

It counts up when the day counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Even if the day count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 01 to 12 to this register in BCD code. If a value outside the range is set, the register value returns to the normal value after 1 period.

The MONTH register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 01H.

**Figure 9-12. Format of Month Count Register (MONTH)**

Address: FFF97H After reset: 01H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| MONTH | 0 | 0 | 0 | MONTH10 | MONTH8 | MONTH4 | MONTH2 | MONTH1 |

**(12) Year count register (YEAR)**

The YEAR register is an 8-bit register that takes a value of 0 to 99 (decimal) and indicates the count value of years.

It counts up when the month count register (MONTH) overflows.

Values 00, 04, 08, …, 92, and 96 indicate a leap year.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Even if the MONTH register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 00 to 99 to this register in BCD code. If a value outside the range is set, the register value returns to the normal value after 1 period.

The YEAR register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 9-13. Format of Year Count Register (YEAR)**

Address: FFF98H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| YEAR | YEAR80 | YEAR40 | YEAR20 | YEAR10 | YEAR8 | YEAR4 | YEAR2 | YEAR1 |

**(14) Alarm minute register (ALARMWM)**

This register is used to set minutes of alarm.

The ALARMWM register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Caution Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, the alarm is not detected.**

**Figure 9-15. Format of Alarm Minute Register (ALARMWM)**

Address: FFF9AH After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ALARMWM | 0 | WM40 | WM20 | WM10 | WM8 | WM4 | WM2 | WM1 |

**(15) Alarm hour register (ALARMWH)**

This register is used to set hours of alarm.

The ALARMWH register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 12H.

However, the value of this register is 00H if the AMPM bit (bit 3 of the RTCC0 register) is set to 1 after reset.

**Caution Set a decimal value of 00 to 23, 01 to 12, or 21 to 32 to this register in BCD code. If a value outside the range is set, the alarm is not detected.**

**Figure 9-16. Format of Alarm Hour Register (ALARMWH)**

Address: FFF9BH After reset: 12H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ALARMWH | 0 | 0 | WH20 | WH10 | WH8 | WH4 | WH2 | WH1 |

**Caution Bit 5 (WH20) of the ALARMWH register indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).**

**(16) Alarm week register (ALARMWW)**

This register is used to set date of alarm.

The ALARMWW register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

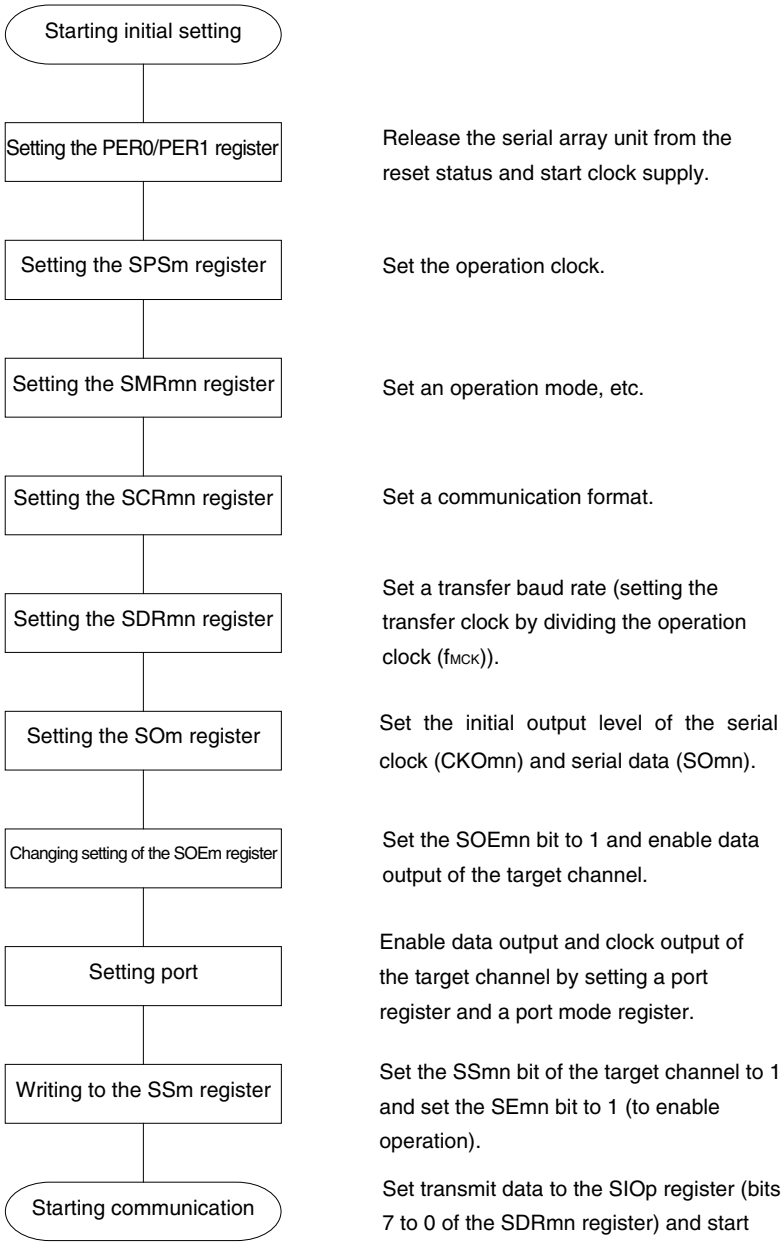
**Figure 9-17. Format of Alarm Week Register (ALARMWW)**

Address: FFF9CH After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ALARMWW | 0 | WW6 | WW5 | WW4 | WW3 | WW2 | WW1 | WW0 |

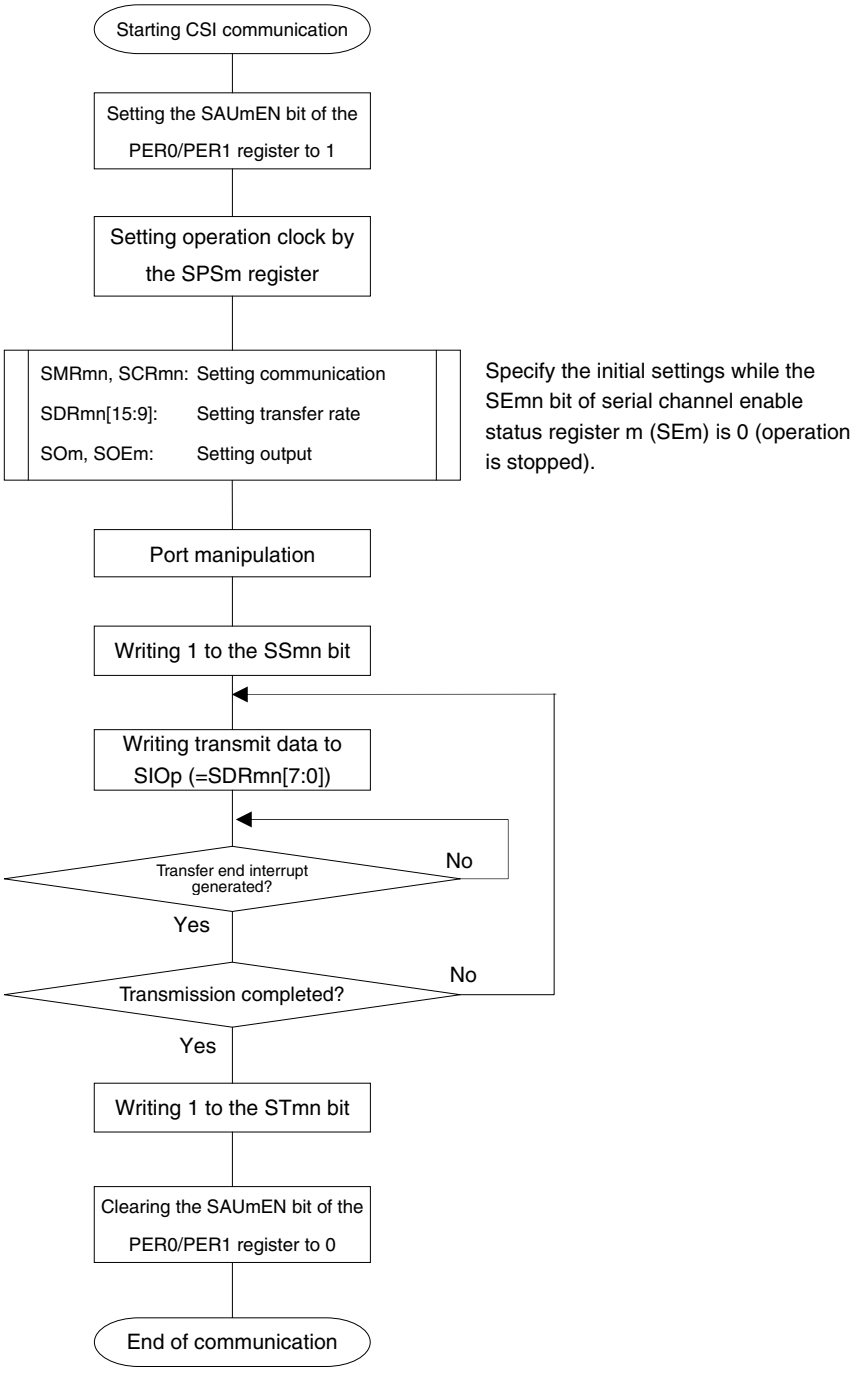
### (2) Operation procedure

**Figure 14-29. Initial Setting Procedure for Master Transmission**




**Caution After setting the SAUmEN bit of peripheral enable register 0/1 (PER0/PER1) to 1, be sure to set serial clock select register m (SPSm) after 4 or more $f_{CLK}$ clocks have elapsed.**

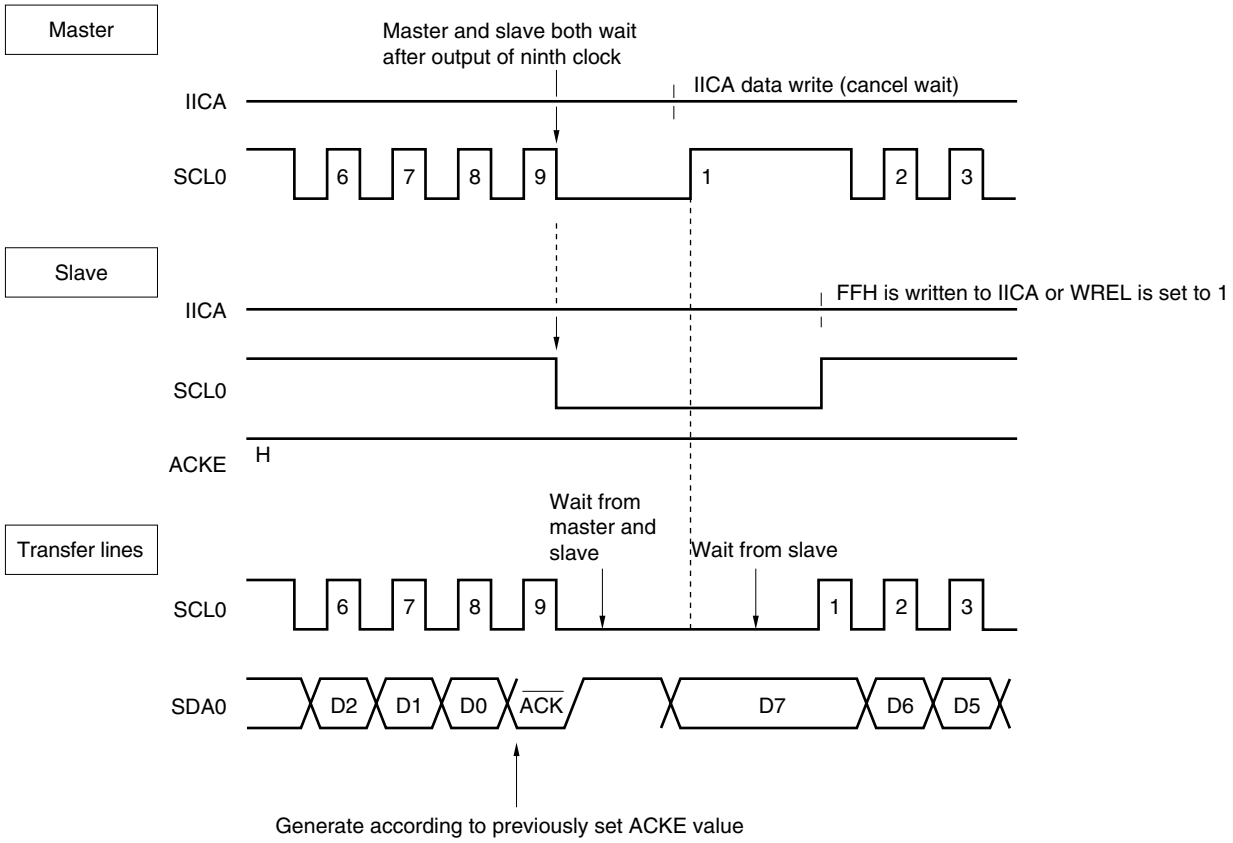
**Figure 14-33. Flowchart of Master Transmission (in Single-Transmission Mode)**




**Caution After setting the SAUmEN bit of peripheral enable register 0/1 (PER0/PER1) to 1, be sure to set serial clock select register m (SPSm) after 4 or more $f_{CLK}$ clocks have elapsed.**

**Figure 15-20. Wait (2/2)**

**(2) When master and slave devices both have a nine-clock wait (master transmits, slave receives, and ACKE = 1)**




**Remark** ACKE: Bit 2 of IICA control register 0 (IICCTL0)
WREL: Bit 5 of IICA control register 0 (IICCTL0)

A wait may be automatically generated depending on the setting of bit 3 (WTIM) of IICA control register 0 (IICCTL0).

Normally, the receiving side cancels the wait state when bit 5 (WREL) of the IICCTL0 register is set to 1 or when FFH is written to the IICA shift register (IICA), and the transmitting side cancels the wait state when data is written to the IICA register.

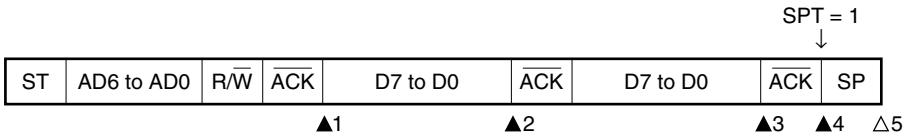
The master device can also cancel the wait state via either of the following methods.

- By setting bit 1 (STT) of the IICCTL0 register to 1
- By setting bit 0 (SPT) of the IICCTL0 register to 1

### (1) Master device operation

#### (a) Start ~ Address ~ Data ~ Data ~ Stop (transmission/reception)

##### (i) When WTIM = 0

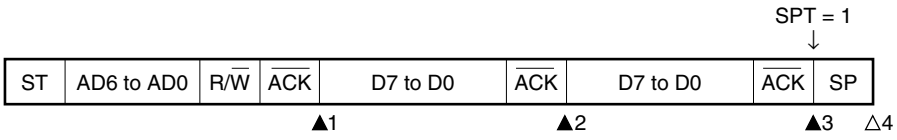



▲1: IICS = 1000×110B
▲2: IICS = 1000×000B
▲3: IICS = 1000×000B (Sets the WTIM bit to 1)Note
▲4: IICS = 1000××00B (Sets the SPT bit to 1)Note
△5: IICS = 00000001B

**Note** To generate a stop condition, set the WTIM bit to 1 and change the timing for generating the INTIICA interrupt request signal.

**Remark** ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

##### (ii) When WTIM = 1




▲1: IICS = 1000×110B
▲2: IICS = 1000×100B
▲3: IICS = 1000××00B (Sets the SPT bit to 1)
△4: IICS = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

### 16.4.2 Division operation

- Initial setting
  - <1> Set bit 7 (DIVMODE) of the multiplication/division control register (MDUC) to 1.
  - <2> Set the dividend (higher 16 bits) to multiplication/division data register A (H) (MDAH).
  - <3> Set the dividend (lower 16 bits) to multiplication/division data register A (L) (MDAL).
  - <4> Set the divisor (higher 16 bits) to multiplication/division data register B (H) (MDBH).
  - <5> Set the divisor (lower 16 bits) to multiplication/division data register B (L) (MDBL).
  - <6> Set bit 0 (DIVST) of the MDUC register to 1.

    (There is no preference in the order of executing steps <2> to <5>.)
- During operation processing
  - <7> The operation will end when one of the following processing is completed.
    - A wait of at least 16 clocks (The operation will end when 16 clocks have been issued.)
    - A check whether the DIVST bit has been cleared
    - Generation of a division completion interrupt (INTMD)

    (The read values of the MDBL, MDBH, MDCL, and MDCH registers during operation processing are not guaranteed.)
- Operation end
  - <8> The DIVST bit is cleared (0) and an interrupt request signal (INTMD) is generated (end of operation).
  - <9> Read the quotient (lower 16 bits) from the MDAL register.
  - <10> Read the quotient (higher 16 bits) from the MDAH register.
  - <11> Read the remainder (lower 16 bits) from multiplication/division data register C (L) (MDCL).
  - <12> Read the remainder (higher 16 bits) from multiplication/division data register C (H) (MDCH).

    (There is no preference in the order of executing steps <9> to <12>.)
- Next operation
  - <13> To execute multiplication operation next, start from the “Initial setting” in **16.4.1 Multiplication operation**.
  - <14> To execute division operation next, start from the “Initial setting” for division operation.

**Remark** Steps <1> to <12> correspond to <1> to <12> in Figure 16-7.

## 26.7 Security Settings

The 78K0R/Kx3-L supports a security function that prohibits rewriting the user program written to the internal flash memory, so that the program cannot be changed by an unauthorized person.

The operations shown below can be performed using the Security Set command. The security setting is valid when the programming mode is set next.

- Disabling batch erase (chip erase)
  Execution of the block erase and batch erase (chip erase) commands for entire blocks in the flash memory is prohibited by this setting during on-board/off-board programming. Once execution of the batch erase (chip erase) command is prohibited, all of the prohibition settings (including prohibition of batch erase (chip erase)) can no longer be cancelled.

  **Caution After the security setting for the batch erase is set, erasure cannot be performed for the device. In addition, even if a write command is executed, data different from that which has already been written to the flash memory cannot be written, because the erase command is disabled.**

- Disabling block erase
  Execution of the block erase command for a specific block in the flash memory is prohibited during on-board/off-board programming. However, blocks can be erased by means of self programming.

- Disabling write
  Execution of the write and block erase commands for entire blocks in the flash memory is prohibited during on-board/off-board programming. However, blocks can be written by means of self programming.

- Disabling rewriting boot cluster 0
  Execution of the batch erase (chip erase) command, block erase command, and write command on boot cluster 0 (00000H to 00FFFH) in the flash memory is prohibited by this setting.

The batch erase (chip erase), block erase, write commands, and rewriting boot cluster 0 are enabled by the default setting when the flash memory is shipped. Security can be set by on-board/off-board programming and self programming. Each security setting can be used in combination.

All the security settings are cleared by executing the batch erase (chip erase) command.

Table 26-8 shows the relationship between the erase and write commands when the 78K0R/Kx3-L security function is enabled.

**Remark** To prohibit writing and erasing during self-programming, use the flash sealed window function (see **26.9.2** for detail).

### 29.1.3 Description of flag operation column

The change of the flag value when the instruction is executed is shown in the “Flag” column using the following symbols.

**Table 29-3. Symbols in “Flag” Column**

| Symbol | Change of Flag Value |
|---|---|
| (Blank) | Unchanged |
| 0 | Cleared to 0 |
| 1 | Set to 1 |
| × | Set/cleared according to the result |
| R | Previously saved value is restored |

### 29.1.4 PREFIX instruction

Instructions with “ES:” have a PREFIX operation code as a prefix to extend the accessible data area to the 1 MB space (00000H to FFFFFH), by adding the ES register value to the 64 KB space from F0000H to FFFFFH. When a PREFIX operation code is attached as a prefix to the target instruction, only one instruction immediately after the PREFIX operation code is executed as the addresses with the ES register value added.

A interrupt and DMA transfer are not acknowledged between a PREFIX instruction code and the instruction immediately after.

**Table 29-4. Use Example of PREFIX Operation Code**

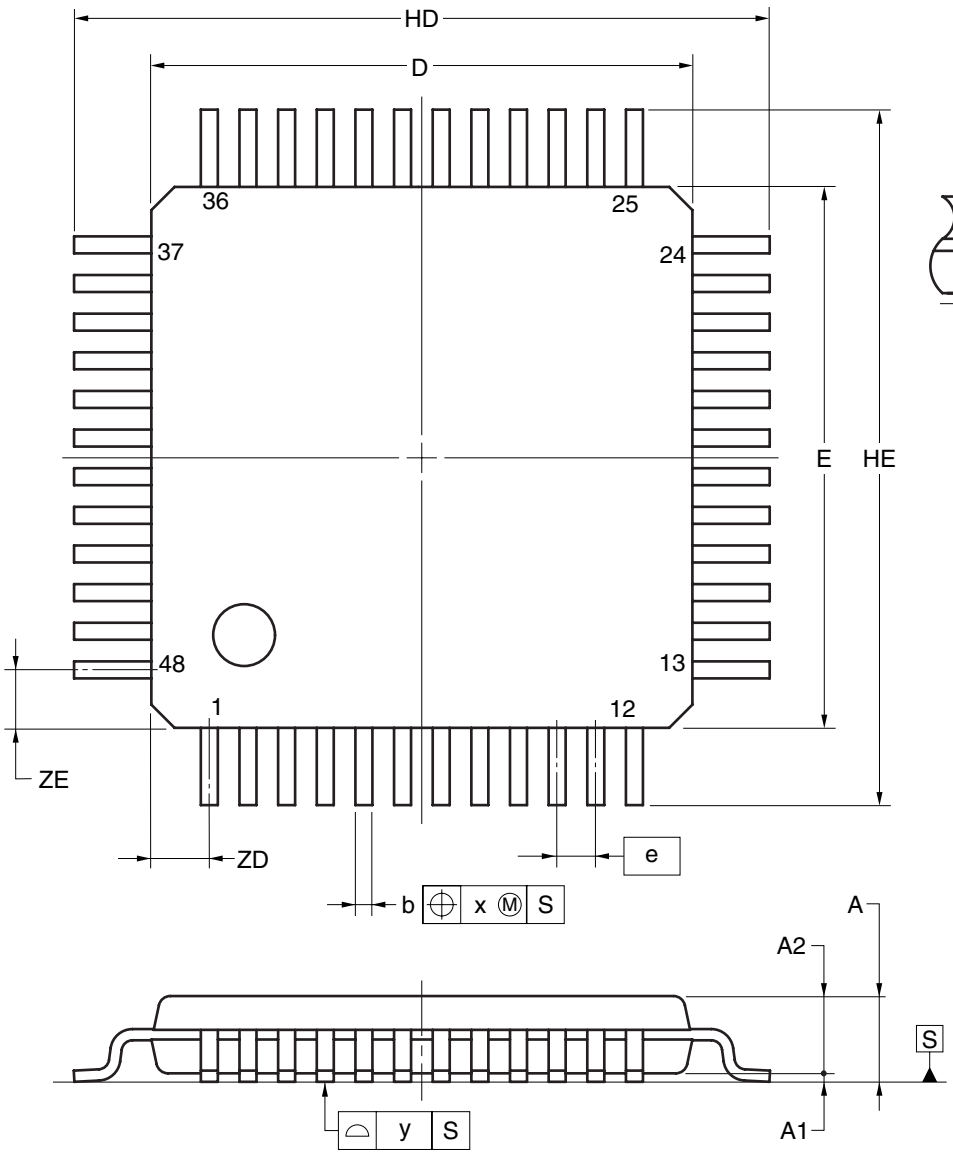
| Instruction | Opcode | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| MOV !addr16, #byte | CFH | !addr16 | | #byte | – |
| MOV ES:!addr16, #byte | 11H | CFH | !addr16 | | #byte |
| MOV A, [HL] | 8BH | – | – | – | – |
| MOV A, ES:[HL] | 11H | 8BH | – | – | – |

**Caution Set the ES register value with MOV ES, A, etc., before executing the PREFIX instruction.**

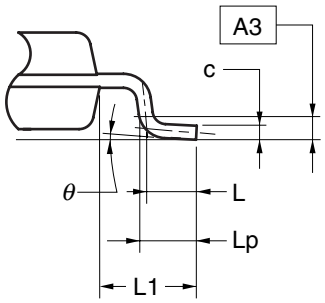
### 32.3 78K0R/KC3-L (48-pin products)

$\mu$PD78F1001GA-HAA-AX, 78F1002GA-HAA-AX, 78F1003GA-HAA-AX

## 48-PIN PLASTIC TQFP (FINE PITCH) (7x7)




detail of lead end




(UNIT:mm)

| ITEM | DIMENSIONS |
|---|---|
| D | 7.00±0.20 |
| E | 7.00±0.20 |
| HD | 9.00±0.20 |
| HE | 9.00±0.20 |
| A | 1.20 MAX. |
| A1 | 0.10±0.05 |
| A2 | 1.00±0.05 |
| A3 | 0.25 |
| b | $0.20^{+0.07}_{-0.03}$ |
| c | $0.125^{+0.075}_{-0.025}$ |
| L | 0.50 |
| Lp | 0.60±0.15 |
| L1 | 1.00±0.20 |
| θ | $3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$ |
| e | 0.50 |
| x | 0.08 |
| y | 0.08 |
| ZD | 0.75 |
| ZE | 0.75 |

P48GA-50-HAA

**NOTE**
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.