**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 | CSI, I²C, LINbus, UART/USART |
| Peripherals | DMA, LCD, LVD, POR, PWM, WDT |
| Number of I/O | 46 |
| Program Memory Size | 128KB (128K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 7K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.8V ~ 5.5V |
| Data Converters | A/D 8x12b; D/A 2x12b |
| 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/upd78f1502agc-gad-ax |

**(2) Non-port functions (4/5) : 78K0R/LH3**

| Function Name | I/O | Function | After Reset | Alternate Function |
|---|---|---|---|---|
| SI00 | Input | Serial data input to CSI00 | Input port | P81/RxD0/INTP9 |
| SI01 | | Serial data input to CSI01 | | P76/KR6 |
| SI10 | | Serial data input to CSI10 | | P14/RxD1/SDA10/ INTP4 |
| SI20 | | Serial data input to CSI20 | | P11/RxD2/SDA20/ INTP6 |
| SO00 | Output | Serial data output from CSI00 | Input port | P82/TxD0 |
| SO01 | | Serial data output from CSI01 | | P77/KR7 |
| SO10 | | Serial data output from CSI10 | | P13/TxD1/TO04 |
| SO20 | | Serial data output from CSI20 | | P12/TxD2/TO02 |
| TI00 | Input | External count clock input to 16-bit timer 00 | Input port | P31/TO03/RTCDIV/ RTCCL/PCLBUZ1/ INTP2 |
| TI01 | | External count clock input to 16-bit timer 01 | | P32/TO01/INTP5/ PCLBUZ0 |
| TI02 | | External count clock input to 16-bit timer 02 | | P52/SEG51 |
| TI03 | | External count clock input to 16-bit timer 03 | | P30/TO00/RTC1HZ/ INTP1 |
| TI04 | | External count clock input to 16-bit timer 04 | | P53/SEG50 |
| TI05 | | External count clock input to 16-bit timer 05 | | P16/TO05/INTP10 |
| TI06 | | External count clock input to 16-bit timer 06 | | P34/TO06/INTP8 |
| TI07 | | External count clock input to 16-bit timer 07 | | P33/TO07/INTP3 |
| TI10 | | External count clock input to 16-bit timer 10 | | P84/TO10 |
| TI11 | | External count clock input to 16-bit timer 11 | | P85/TO11 |
| TI12 | | External count clock input to 16-bit timer 12 | | P86/TO12 |
| TI13 | | External count clock input to 16-bit timer 13 | | P87/TO13 |
| TO00 | Output | 16-bit timer 00 output | Input port | P30/TI03/RTC1HZ/ INTP1 |
| TO01 | | 16-bit timer 01 output | | P32/TI01/INTP5/ PCLBUZ0 |
| TO02 | | 16-bit timer 02 output | | P12/SO20/TxD2 |
| TO03 | | 16-bit timer 03 output | | P31/TI00/RTCDIV/ RTCCL/PCLBUZ1/ INTP2 |
| TO04 | | 16-bit timer 04 output | | P13/SO10/TxD1 |
| TO05 | | 16-bit timer 05 output | | P16/TI05/INTP10 |
| TO06 | | 16-bit timer 06 output | | P34/TI06/INTP8 |
| TO07 | | 16-bit timer 07 output | | P33/TI07/INTP3 |

### 3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)

Unlike a general-purpose register, each extended SFR (2nd SFR) has a special function.

Extended SFRs are allocated to the F0000H to F07FFH area. SFRs other than those in the SFR area (FFF00H to FFFFFH) are allocated to this area. An instruction that accesses the extended SFR area, however, is 1 byte longer than an instruction that accesses the SFR area.

Extended SFRs can be manipulated like general-purpose registers, using operation, transfer, and bit manipulation instructions. The manipulable bit units, 1, 8, and 16, depend on the SFR type.

Each manipulation bit unit can be specified as follows.

1-bit manipulation

Describe the symbol reserved by the assembler for the 1-bit manipulation instruction operand (!addr16.bit). This manipulation can also be specified with an address.

8-bit manipulation

Describe the symbol reserved by the assembler for the 8-bit manipulation instruction operand (!addr16). This manipulation can also be specified with an address.

16-bit manipulation

Describe the symbol reserved by the assembler for the 16-bit manipulation instruction operand (!addr16). When specifying an address, describe an even address.

Table 3-6 gives a list of the extended SFRs. The meanings of items in the table are as follows.

Symbol

Symbol indicating the address of an extended SFR. It is a reserved word in the RA78K0R, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0R. When using the RA78K0R, ID78K0R-QB, and SM+ for 78K0R, symbols can be written as an instruction operand.

R/W

Indicates whether the corresponding extended SFR can be read or written.

R/W: Read/write enable
R: Read only
W: Write only

Manipulable bit units

“ ” indicates the manipulable bit unit (1, 8, or 16). “ ” indicates a bit unit for which manipulation is not possible.

After reset

Indicates each register status upon reset signal generation.

**Caution Do not access addresses to which 2nd SFRs are not assigned.**

**Remark** For SFRs in the SFR area, see **3.2.4 Special function registers (SFRs)**.

## 3.4 Addressing for Processing Data Addresses

### 3.4.1 Implied addressing

**[Function]**

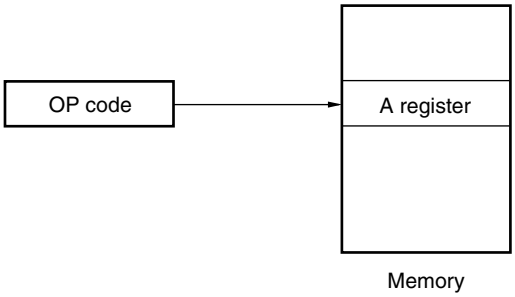
Instructions for accessing registers (such as accumulators) that have special functions are directly specified with the instruction word, without using any register specification field in the instruction word.

**[Operand format]**

Because implied addressing can be automatically employed with an instruction, no particular operand format is necessary.

Implied addressing can be applied only to MULU X.

**Figure 3-19. Outline of Implied Addressing**




### 3.4.2 Register addressing

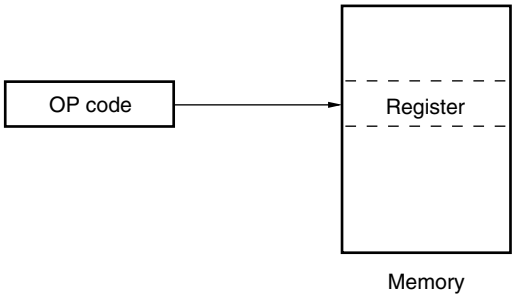
**[Function]**

Register addressing accesses a general-purpose register as an operand. The instruction word of 3-bit long is used to select an 8-bit register and the instruction word of 2-bit long is used to select a 16-bit register.

**[Operand format]**

| Identifier | Description |
|---|---|
| r | X, A, C, B, E, D, L, H |
| rp | AX, BC, DE, HL |

**Figure 3-20. Outline of Register Addressing**



### 3.4.3 Direct addressing

**[Function]**

Direct addressing uses immediate data in the instruction word as an operand address to directly specify the target address.

**[Operand format]**

| Identifier | Description |
|---|---|
| ADDR16 | Label or 16-bit immediate data (only the space from F0000H to FFFFFH is specifiable) |
| ES: ADDR16 | Label or 16-bit immediate data (higher 4-bit addresses are specified by the ES register) |

**Figure 3-21. Example of ADDR16**

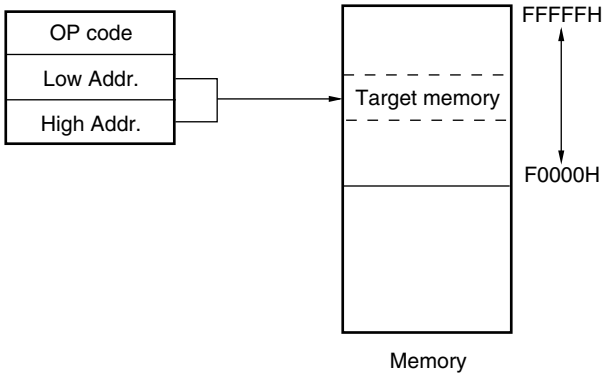



**Figure 3-22. Example of ES:ADDR16**

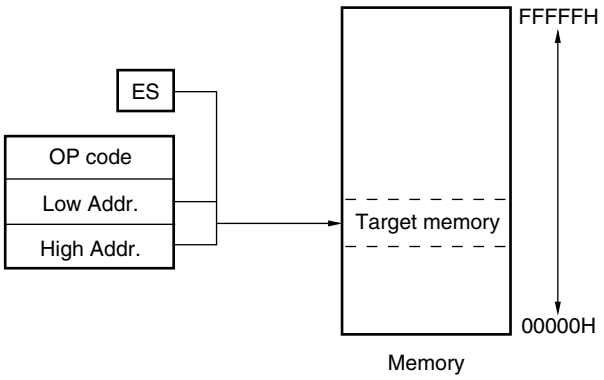


**Figure 5-8. Format of Peripheral Enable Register 0 (PER0) (2/2)**

| SAU1EN | Control of serial array unit 1 input clock |
|---|---|
| 0 | Stops input clock supply.<br>SFR used by the serial array unit 1 cannot be written.<br>The serial array unit is in the reset status. |
| 1 | Supplies input clock.<br>SFR used by the serial array unit 1 can be read and written. |

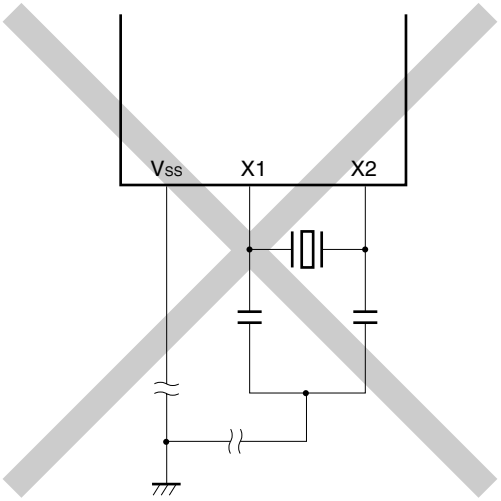
| SAU0EN | Control of serial array unit 0 input clock |
|---|---|
| 0 | Stops input clock supply.<br>SFR used by the serial array unit 0 cannot be written.<br>The serial array unit 0 is in the reset status. |
| 1 | Supplies input clock.<br>SFR used by the serial array unit 0 can be read and written. |

| TAU1EN | Control of timer array unit 1 input clock |
|---|---|
| 0 | Stops input clock supply.<br>SFR used by timer array unit 1 cannot be written.<br>Timer array unit 1 is in the reset status. |
| 1 | Supplies input clock.<br>SFR used by timer array unit 1 can be read and written. |

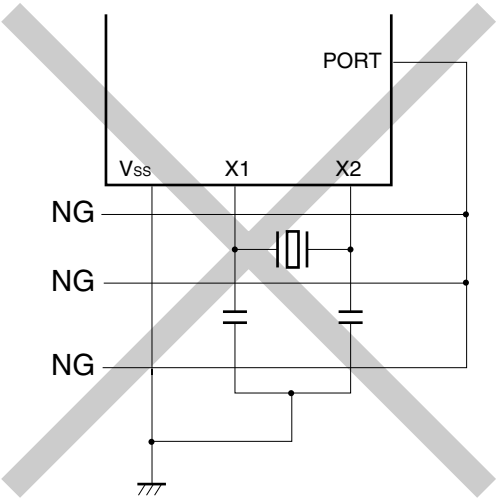
| TAU0EN | Control of timer array unit 0 input clock |
|---|---|
| 0 | Stops input clock supply.<br>SFR used by timer array unit 0 cannot be written.<br>Timer array unit 0 is in the reset status. |
| 1 | Supplies input clock.<br>SFR used by timer array unit 0 can be read and written. |

**Figure 5-12. Examples of Incorrect Resonator Connection (1/2)**

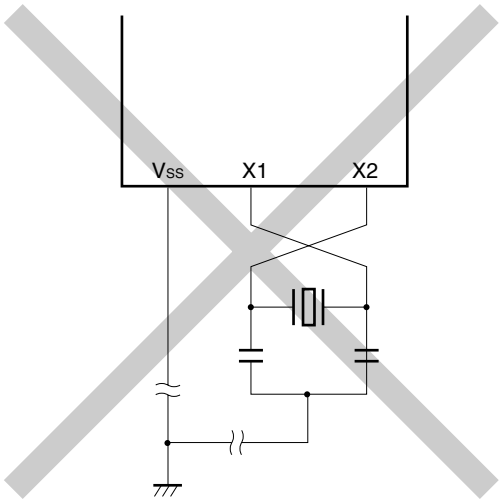
**(a) Too long wiring**



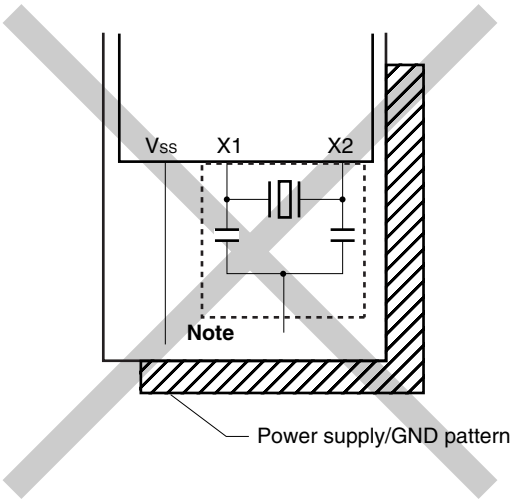

**(b) Crossed signal line**




**(c) Signal lines of X1 and X2 cross**




**(d) Power supply/GND pattern exists underneath X1 and X2 wiring**




**Note** Do not place a power supply/GND pattern underneath the wiring section (in broken lines above) of the X1 and X2 pins and resonator in the multilayer board and double-sided board.
Do not configure a layout that may cause capacitance elements and affect the oscillation characteristics.

**Remark** When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

### 6.1.2 Functions of each channel when it operates with another channel

Combination operation functions are those functions that are attained by using the master channel (mostly the reference timer that controls cycles) and the slave channels (timers that operate following the master channel) in combination (for details, refer to **6.6.1 Overview of single-operation function and combination operation function**).

**(1) PWM (Pulse Width Modulator) output**

Two channels are used as a set to generate a pulse with a specified period and a specified duty factor.

**(2) One-shot pulse output**

Two channels are used as a set to generate a one-shot pulse with a specified delay time and a specified pulse width.

**(3) Multiple PWM (Pulse Width Modulator) output**

By extending the PWM function and using one master channel and two or more slave channels, up to seven types of PWM signals that have a specific period and a specified duty factor can be generated.

### 6.1.3 LIN-bus supporting function (channel 7 of timer array unit 0 only)

**(1) Detection of wakeup signal**

The timer starts counting at the falling edge of a signal input to the serial data input pin (RxD3) of UART3 and the count value of the timer is captured at the rising edge. In this way, a low-level width can be measured. If the low-level width is greater than a specific value, it is recognized as a wakeup signal.

**(2) Detection of sync break field**

The timer starts counting at the falling edge of a signal input to the serial data input pin (RxD3) of UART3 after a wakeup signal is detected, and the count value of the timer is captured at the rising edge. In this way, a low-level width is measured. If the low-level width is greater than a specific value, it is recognized as a sync break field.

**(3) Measurement of pulse width of sync field**

After a sync break field is detected, the low-level width and high-level width of the signal input to the serial data input pin (RxD3) of UART3 are measured. From the bit interval of the sync field measured in this way, a baud rate is calculated.

### 7.4.2 Shifting to STOP mode after starting operation

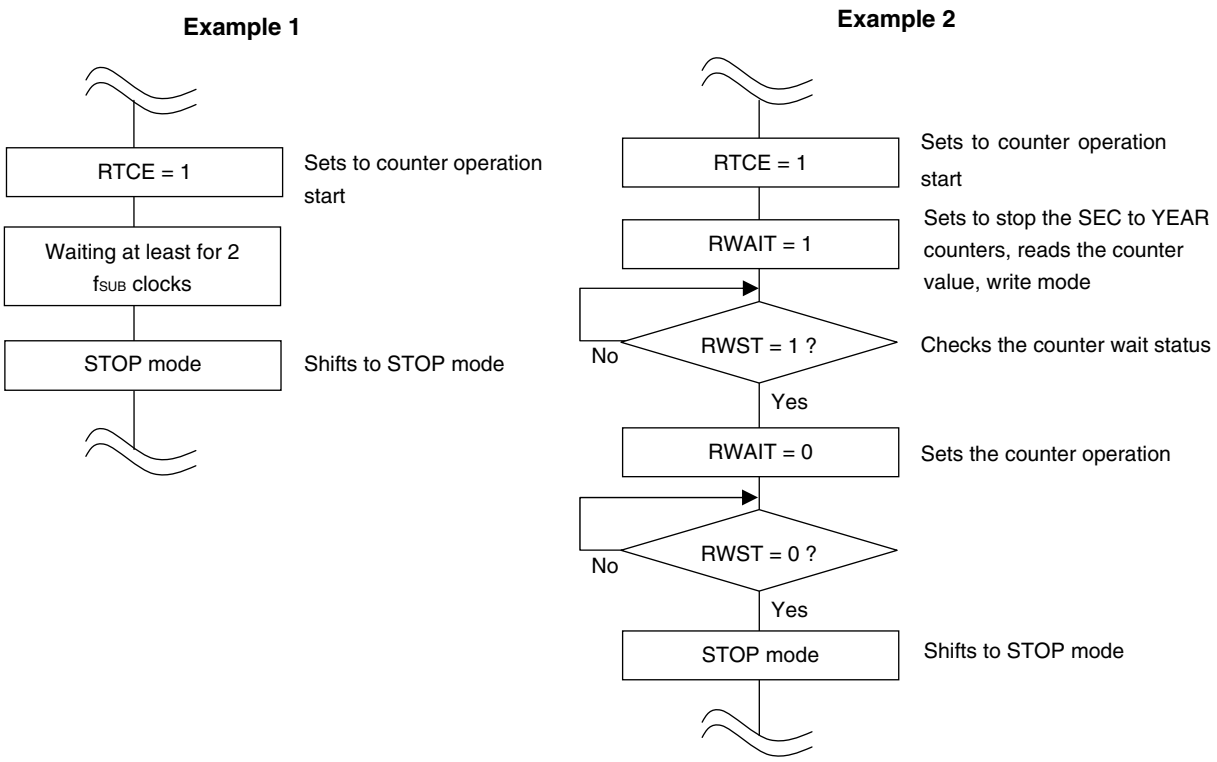
Perform one of the following processing when shifting to STOP mode immediately after setting RTCE to 1.

However, after setting RTCE to 1, this processing is not required when shifting to STOP mode after the first INTRTC interrupt has occurred.

Shifting to STOP mode when at least two subsystem clocks ($f_{SUB}$) (about 62 s) have elapsed after setting RTCE to 1 (see **Figure 7-20**, **Example 1**).

Checking by polling RWST to become 1, after setting RTCE to 1 and then setting RWAIT to 1. Afterward, setting RWAIT to 0 and shifting to STOP mode after checking again by polling that RWST has become 0 (see **Figure 7-20**, **Example 2**).

**Figure 7-20. Procedure for Shifting to STOP Mode After Setting RTCE to 1**



## 11.4 Operation of D/A Converter

### 11.4.1 Operation in normal mode

D/A conversion is performed using write operation to the DACSn register as the trigger. The setting method is described below.

<1> Set bit 6 (DACEN) of peripheral enable register 0 (PER0) to 1 to start the supply of the input clock to the D/A converter.
<2> Set the DAMDn bit of the D/A converter mode register (DAM) to 0 (normal mode).
<3> Use the bit 6 (DAREF) of the DAM register to select the D/A converter reference voltage supply on the positive side.
<4> Use the DARESn bit of the DAM register to select the resolution of the D/A converter.
<5> Set the analog voltage value to be output to the ANOn pin to the D/A conversion value setting register Wn (DACSWn) or D/A conversion value setting register n (DACSn).
Steps <1> and <5> above constitute the initial settings.
<6> Set the DACEn bit of the DAM register to 1 (D/A conversion enable).
After the wait time (20 s or more) elapses, D/A conversion starts, and then, after the settling time (18 s (max.)) elapses, the D/A converted analog voltage value is output from the ANOn pin.
<7> To perform subsequent D/A conversions, write to the DACSWn or DACSn register.
The previous D/A conversion result is held until the next D/A conversion is performed.
When the DACEn bit of the DAM register is set to 0 (D/A conversion operation stop), D/A conversion stops, the ANOn pin goes into a high-impedance state when the PM11n bit of the PM11 register = 1 (input mode), and the ANOn pin outputs the set value of the P11 register when the PM11n bit = 0 (output mode).

**Cautions 1. Even if 1, 0, and then 1 is set to the DACEn bit, there is a wait after 1 is set for the last time.**
**2. If the DACSWn or DACSn register is rewritten during the settling time, D/A conversion is aborted and reconversion by using the rewritten values starts.**

**Remark** n = 0, 1

### 11.4.2 Operation in real-time output mode

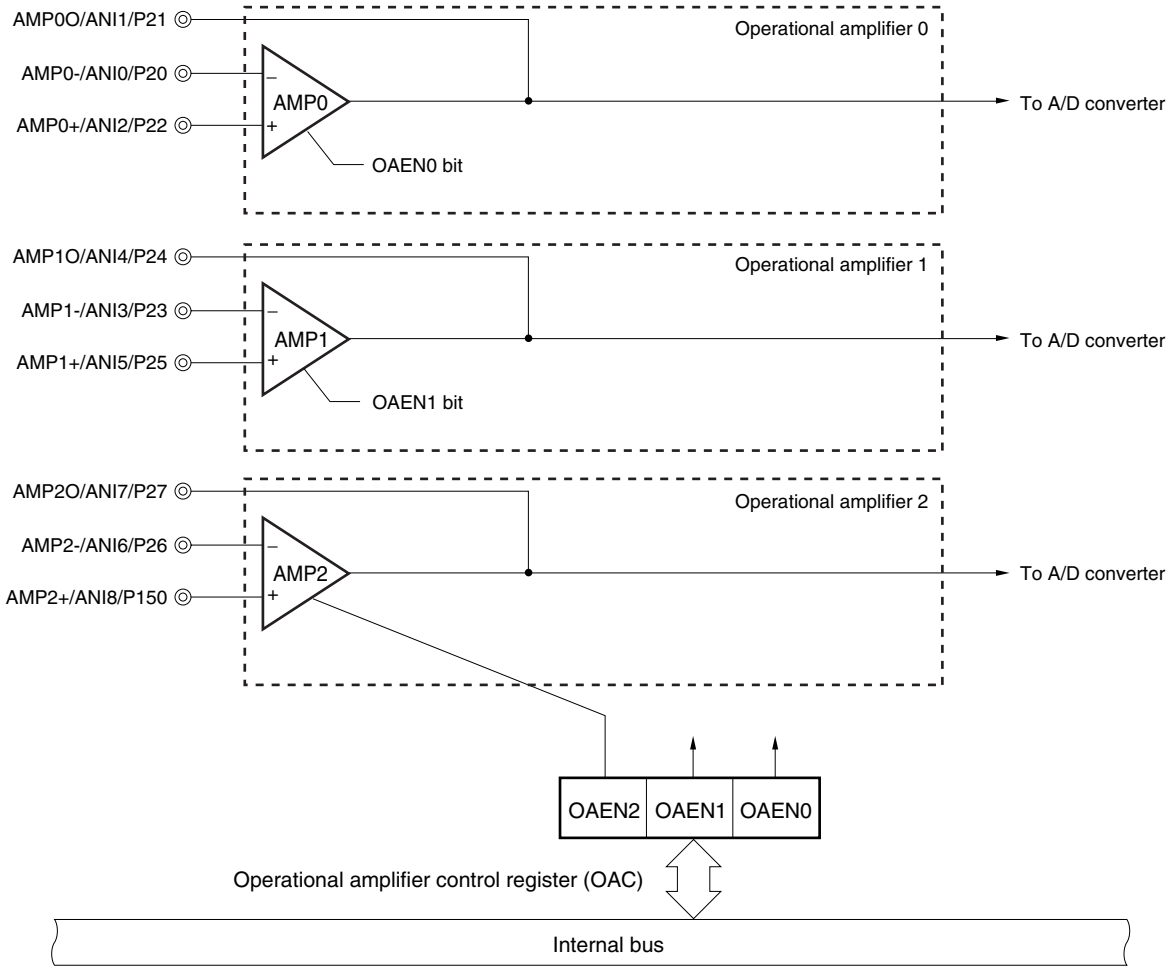
D/A conversion is performed using the interrupt request signals (INTTM04 and INTTM05) [Note] of timer channels 4 and 5 as triggers.

The setting method is described below.

**Note** Channel 0 of the D/A converter: INTTM04
Channel 1 of the D/A converter: INTTM05

<1> Set bit 6 (DACEN) of peripheral enable register 0 (PER0) to 1 to start the supply of the input clock to the D/A converter.
<2> Set the DAMDn bit of the D/A converter mode register (DAM) to 0 (normal mode).
<3> Use the bit 6 (DAREF) of the DAM register to select the D/A converter reference voltage supply on the positive side.
<4> Use the DARESn bit of the DAM register to select the resolution of the D/A converter.
<5> Set the analog voltage value to be output to the ANOn pin to the D/A conversion value setting register Wn (DACSWn) or D/A conversion value setting register n (DACSn).

**Figure 12-1. Block Diagram of Operational Amplifier**




**Remark** 78K0R/LF3: Operational amplifiers 0, 1
78K0R/LG3, 78K0R/LH3: Operational amplifiers 0 to 2

**(7) Serial flag clear trigger register mn (SIRmn)**

SIRmn is a trigger register that is used to clear each error flag of channel n.

When each bit (FECTmn, PECTmn, OVCTmn) of this register is set to 1, the corresponding bit (FEFmn, PEFmn, OVFmn) of serial status register mn is cleared to 0. Because SIRmn is a trigger register, it is cleared immediately when the corresponding bit of SSRmn is cleared.

SIRmn can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of SIRmn can be set with an 8-bit memory manipulation instruction with SIRmnL.

Reset signal generation clears this register to 0000H.

**Figure 14-10. Format of Serial Flag Clear Trigger Register mn (SIRmn)**

Address: F0108H, F0109H (SIR00) to F010EH, F010FH (SIR03), F0148H, F0149H (SIR10), F014AH, F014BH (SIR11), F014EH, F014FH (SIR13) After reset: 0000H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SIRmn | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | FECTmn | PECTmn | OVCTmn |

| FECTmn | Clear trigger of framing error of channel n |
|---|---|
| 0 | No trigger operation |
| 1 | Clears the FEFmn bit of the SSRmn register to 0. |

| PECTmn | Clear trigger of parity error flag of channel n |
|---|---|
| 0 | No trigger operation |
| 1 | Clears the PEFmn bit of the SSRmn register to 0. |

| OVCTmn | Clear trigger of overrun error flag of channel n |
|---|---|
| 0 | No trigger operation |
| 1 | Clears the OVFmn bit of the SSRmn register to 0. |

**Caution Be sure to clear bits 15 to 3 to "0".**

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

**2.** When the SIRmn register is read, 0000H is always read.

## 14.4 Operation stop mode

Each serial interface of serial array unit has the operation stop mode.

In this mode, serial communication cannot be executed, thus reducing the power consumption.

In addition, the P10/SCK20/SCL20, P11/SI20/SDA20/RxD2/INTP6, P12/SO20/TxD2/TO02, P13/SO10/TxD1/TO04, P14/SI10/SDA10/RxD1/INTP4, P15/SCK10/SCL10/INTP7, P50/RxD3/SEGx (78K0R/LF3: x = 30, 78K0R/LG3: x = 39, 78K0R/LH3: x = 53), P51/TxD3/SEGx (78K0R/LF3: x = 29, 78K0R/LG3: x = 38, 78K0R/LH3: x = 52), P75/SCK01/KR5, P76/SI01/KR6, P77/SO01/KR7, P80/SCK00/INTP11, P81/SI00/RxD0/INTP9, and P82/SO00/TxD0 pins can be used as ordinary port pins in this mode.

### 14.4.1 Stopping the operation by units

The stopping of the operation by units is set by using peripheral enable register 0 (PER0).

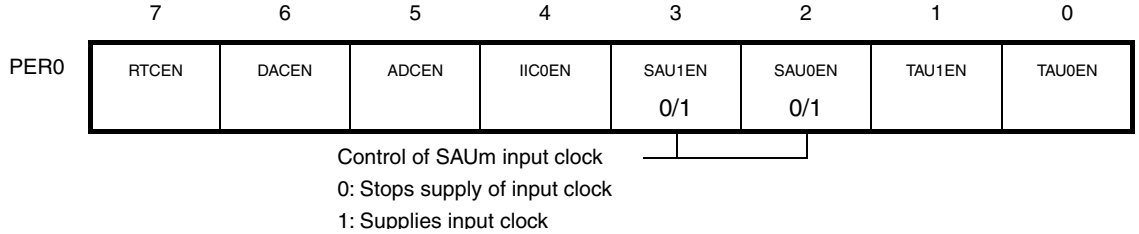
PER0 is used to enable or disable use of each peripheral hardware macro. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

To stop the operation of serial array unit 0, set bit 2 (SAU0EN) to 0.

To stop the operation of serial array unit 1, set bit 3 (SAU1EN) to 0.

**Figure 14-22. Peripheral Enable Register 0 (PER0) Setting When Stopping the Operation by Units**

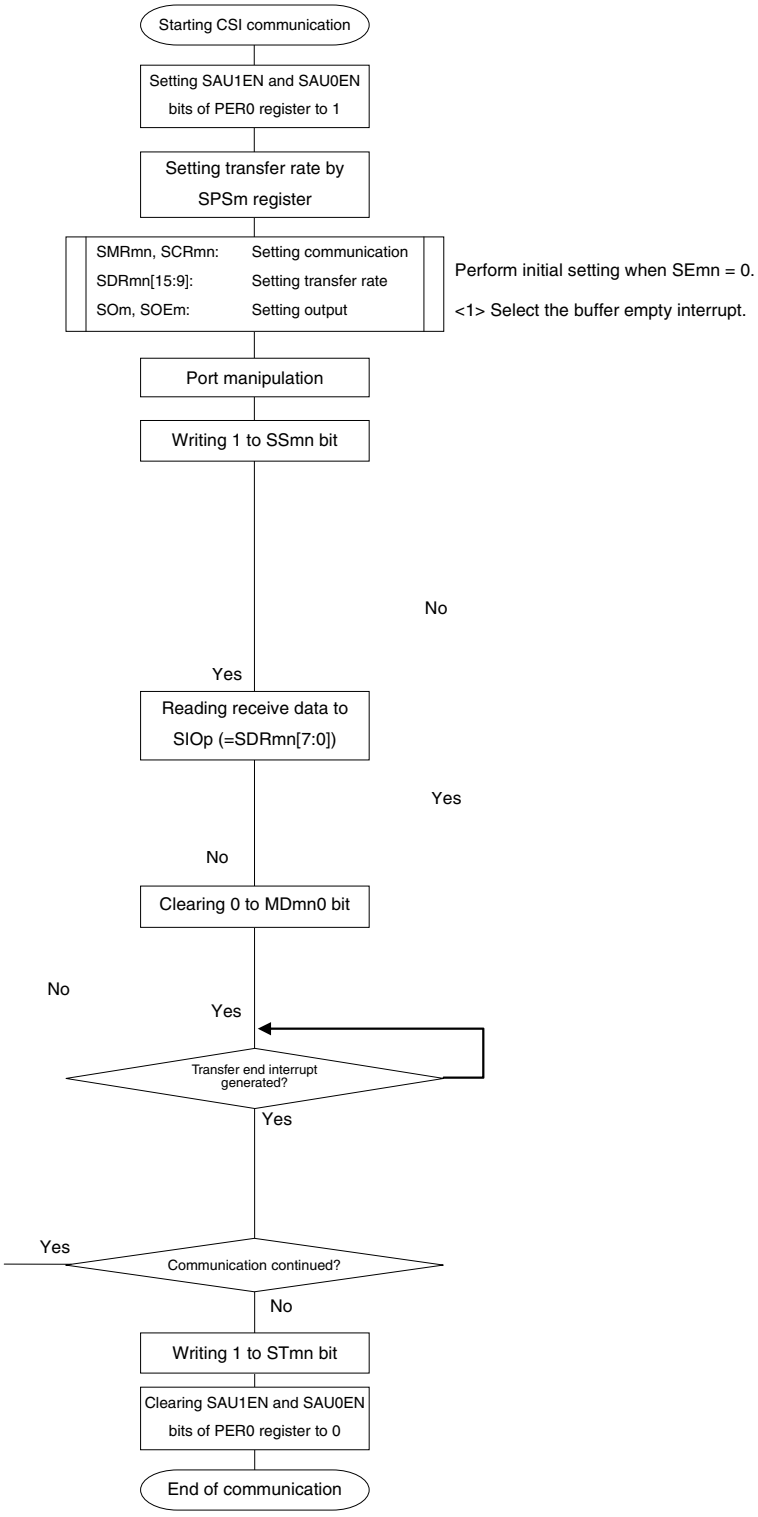
**(a) Peripheral enable register 0 (PER0) … Set only the bit of SAUm to be stopped to 0.**




**Caution If SAUmEN = 0, writing to a control register of serial array unit m is ignored, and, even if the register is read, only the default value is read (except for input switch control register (ISC), noise filter enable register (NFEN0), port input mode registers (PIM1, PIM7), port output mode registers (POM1, POM7, POM8), port mode registers (PM1, PM5, PM7, PM8), and port registers (P1, P5, P7, P8)).**

**Remark** m: Unit number (m = 0, 1)
×: Bits not used with serial array units (depending on the settings of other peripheral functions)
0/1: Set to 0 or 1 depending on the usage of the user

**Figure 14-67. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**



The meanings of <3> to <10> in (2) Address ~ data ~ data in Figure 15-32 are explained below.

<3> If the address received matches the address of a slave device[Note], that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA: end of address transmission) at the falling edge of the 9th clock, and the slave device whose address matched the transmitted slave address also issues an interrupt (INTIICA: address match). The master device and slave device also set a wait status (SCL0 = 0)[Note] when the addresses match.

<5> The master device writes the data to transmit to the IICA shift register (IICA) and releases the wait status that it set by the master device.

<6> If the slave device releases the wait status (WREL = 1), the master device starts transferring data to the slave device.

<7> When data transfer is complete, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA: end of transfer).

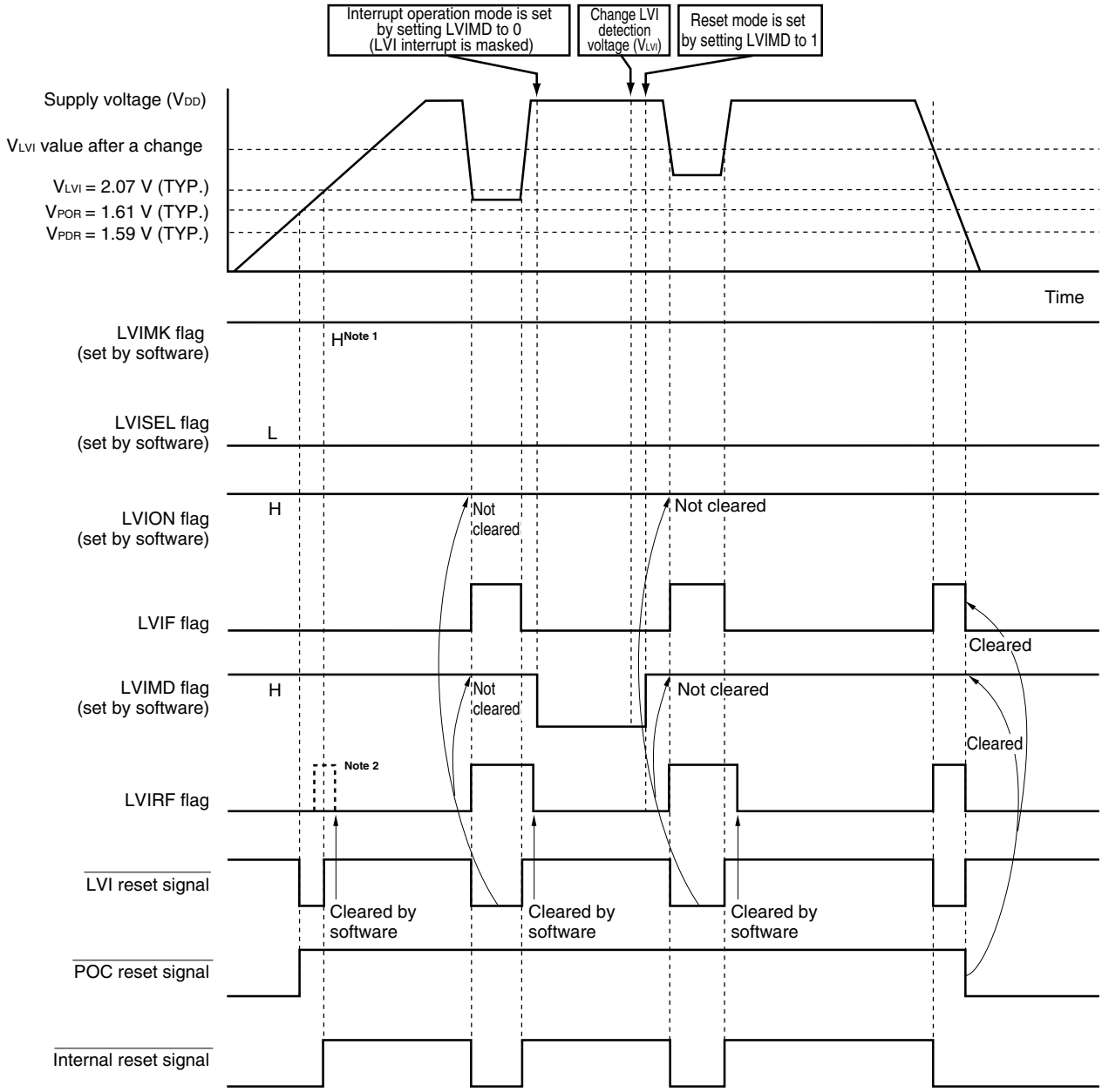
<9> The master device writes the data to transmit to the IICA register and releases the wait status that it set by the master device.

<10> The slave device reads the received data and releases the wait status (WREL = 1). The master device then starts transferring data to the slave device.

**Note** If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDA0 = 1). The slave device also does not issue the INTIICA interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

**Remark** <1> to <15> in Figure 15-32 represent the entire procedure for communicating data using the $I^2C$ bus. Figure 15-32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 15-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 15-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

**Figure 24-6. Timing of Low-Voltage Detector Internal Reset Signal Generation (Bit: LVISEL = 0, Option Byte: LVIOFF = 0)**




**Notes 1.** The LVIMK flag is set to “1” by reset signal generation.

**2.** LVIRF is bit 0 of the reset control flag register (RESF).
When the LVI default start function (bit 0 (LVIOFF) of 000C1H = 0) is used, the LVIRF flag may become 1 from the beginning due to the power-on waveform.
For details of RESF, see **CHAPTER 22 RESET FUNCTION**.

**Remark** $V_{POR}$: POC power supply rise detection voltage
$V_{PDR}$: POC power supply fall detection voltage

**Table 28-1. Lists the Differences Between 1-line Mode and 2-line Mode.**

| Com | Flas | Debug |
|---|---|---|
| | | |