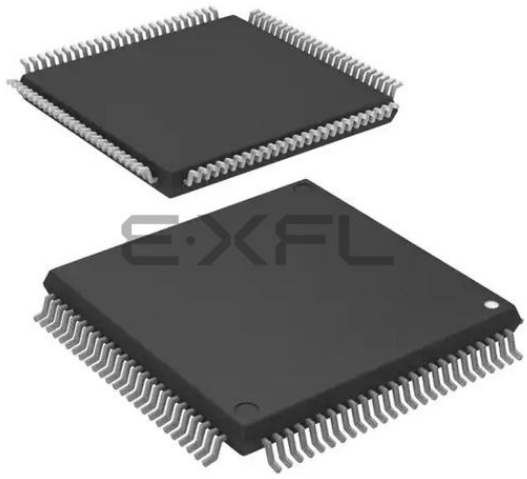
Renesas Electronics America Inc - UPD78F1168AGF-GAS-AX Datasheet



Welcome to E-XFL.COM

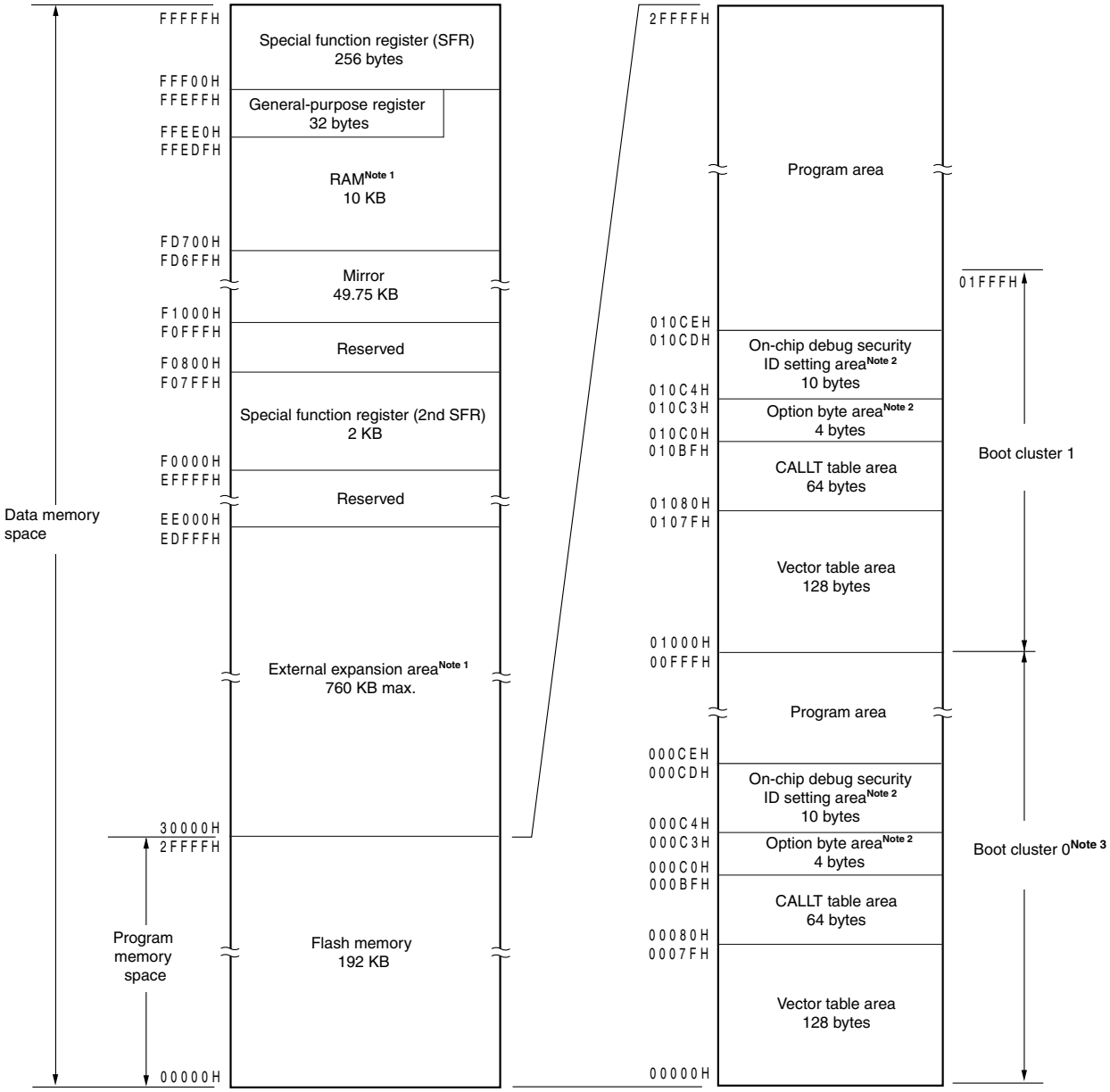
What is "Embedded - Microcontrollers"?

"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	83
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	30K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 16x10b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd78f1168agf-gas-ax

Figure 3-4. Memory Map (μPD78F1165, 78F1165A)



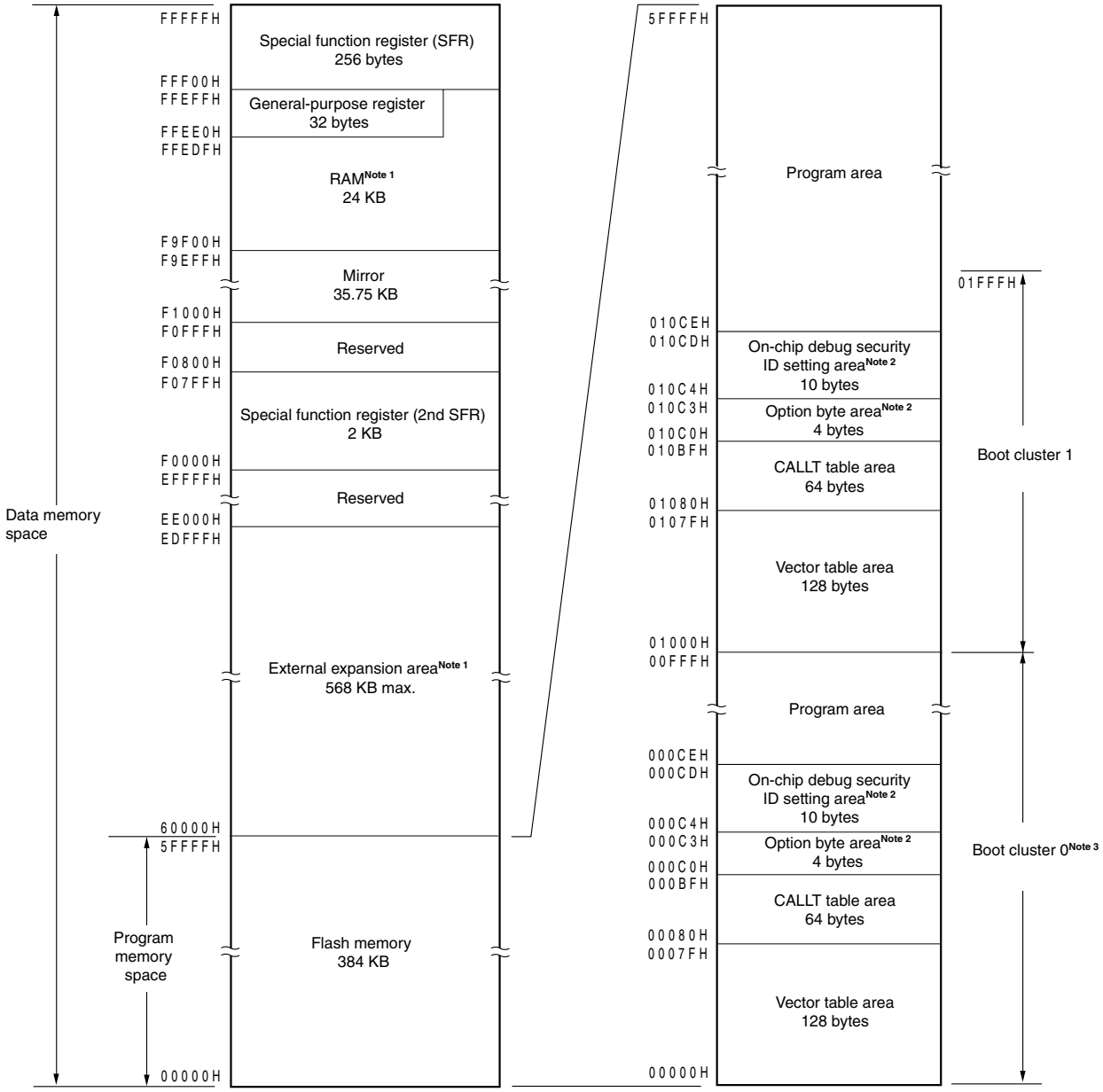
Notes 1. Instructions can be executed from the RAM area excluding the general-purpose register area, and from the external expansion area.

2. 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.

3. Writing boot cluster 0 can be prohibited depending on the setting of security (see **25.7 Security Setting**).

Figure 3-6. Memory Map (μPD78F1167, 78F1167A)



Notes 1. Instructions can be executed from the RAM area excluding the general-purpose register area, and from the external expansion area.

2. 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.

3. Writing boot cluster 0 can be prohibited depending on the setting of security (see **25.7 Security Setting**).

4.2.5 Port 4

Port 4 is an 8-bit I/O port with an output latch. Port 4 can be set to the input mode or output mode in 1-bit units using port mode register 4 (PM4). When the P40 to P47 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 4 (PU4)[Note].

Input to the P43 and P44 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 4 (PIM4).

Output from the P43 and P45 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 4 (POM4).

This port can also be used for external interrupt request input, serial interface data I/O, clock I/O, flash memory programmer/debugger data I/O, clock output, and timer I/O.

Reset signal generation sets port 4 to input mode.

Figures 4-15 to 4-22 show block diagrams of port 4.

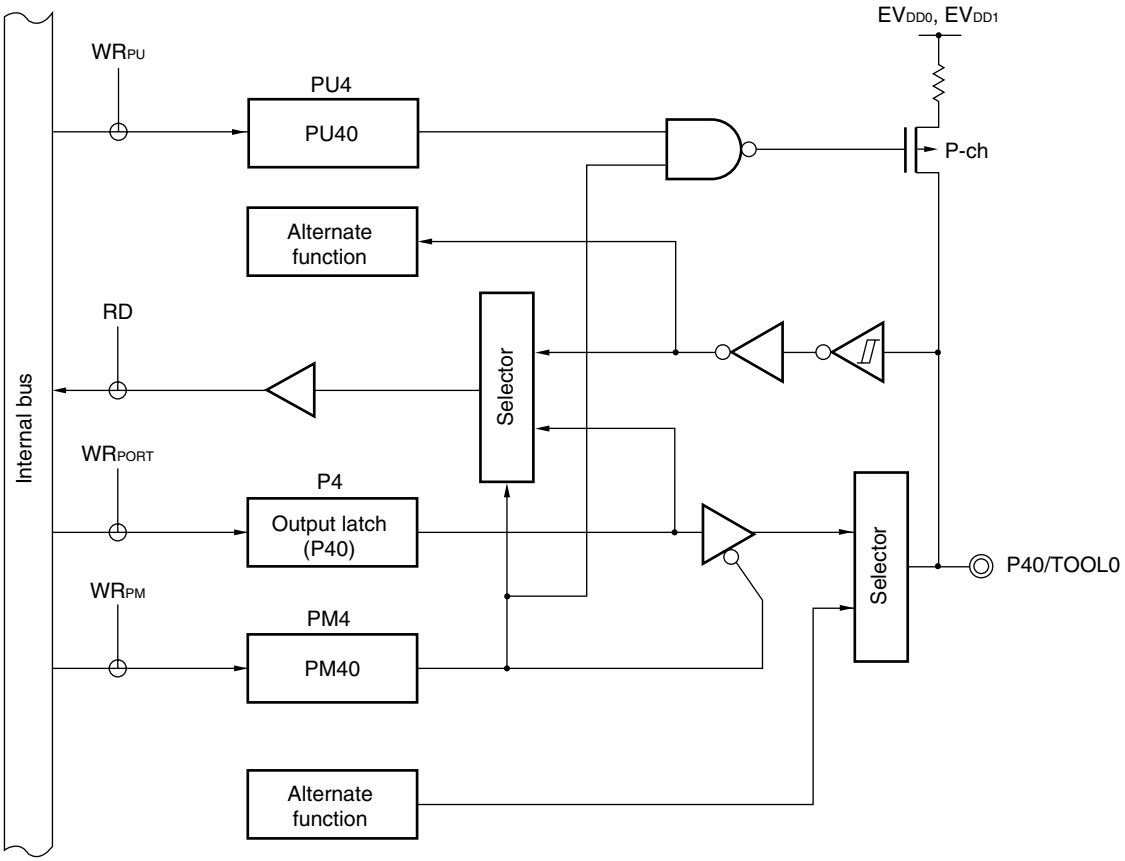
Note When a tool is connected, the P40 and P41 pins cannot be connected to a pull-up resistor.

Cautions 1. When a tool is connected, the P40 pin cannot be used as a port pin.
When the on-chip debug function is used, P41 pin can be used as follows by the mode setting on the debugger.
1-line mode: can be used as a port (P41).
2-line mode: used as a TOOL1 pin and cannot be used as a port (P41).

2. To use P43/$\overline{\text{SCK01}}$, P44/SI01, or P45/SO01 as a general-purpose port, note the serial array unit 0 setting. For details, refer to Table 13-6 Relationship Between Register Settings and Pins (Channel 1 of Unit 0: CSI01, UART0 reception).

3. To use P42/TI04/TO04 or P46/INTP1/TI05/TO05 as a general-purpose port, set bits 4 and 5 (TO04, TO05) of timer output register 0 (TO0) and bits 4 and 5 (TOE04, TOE05) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.

Figure 4-15. Block Diagram of P40



P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
RD: Read signal
WR××: Write signal

5.2 Registers Controlling External Bus Interface Functions

The external bus interface function is controlled by the following registers.

- Peripheral enable register 1 (PER1)
- Memory extension mode control register (MEM)
- Port mode registers 0, 1, 5, 6, 7, 8 (PM0, PM1, PM5, PM6, PM7, PM8)
- Port registers 0, 1, 5, 6, 7, 8 (P0, P1, P5, P6, P7, P8)

(1) Peripheral enable register 1 (PER1)

PER1 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.

When the external bus interface is used, be sure to set bit 0 (EXBEN) of this register to 1.

PER1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 5-2. Format of Peripheral Enable Register 1 (PER1)

Address: F00F1H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
PER1	0	0	0	0	0	0	0	EXBEN

EXBEN	Control of external bus interface input clock
0	Stops supply of input clock. • SFR used by external bus interface cannot be written. • External bus interface is in the reset status.
1	Supplies input clock. • SFR used by external bus interface can be read/written.

Caution When setting the external bus interface, be sure to set EXBEN to 1 first. If EXBEN = 0, writing to a control register of the external bus interface is ignored, and, even if the register is read, only the default value is read (except for port mode registers 0, 1, 5, 6, 7, 8 (PM0, PM1, PM5, PM6, PM7, PM8) and port registers 0, 1, 5, 6, 7, 8 (P0, P1, P5, P6, P7, P8)).

(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.

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

Reset signal generation clears this register to 01H.

Figure 8-12. 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 counter 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 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 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.

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

Reset signal generation clears this register to 00H.

Figure 8-13. 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

11.7 Cautions for A/D Converter

(1) Operating current in STOP mode

Shift to STOP mode after clearing the A/D converter (by clearing bit 7 (ADCS) of the A/D converter mode register (ADM) to 0). The operating current can be reduced by clearing bit 0 (ADCE) of the A/D converter mode register (ADM) to 0 at the same time.

To restart from the standby status, clear bit 0 (ADIF) of interrupt request flag register 1L (IF1L) to 0 and start operation.

(2) Reducing current when A/D converter is stopped

Be sure that the voltage to be applied to AV_{REF0} normally satisfies the conditions stated in Table 11-1.

If bit 7 (ADCS) and bit 0 (ADCE) of the A/D converter mode register (ADM) are set to 0, the current will not be increased by the A/D converter even if a voltage is applied to AV_{REF0}, while the A/D converter is stopped. If a current flows from the power supply that supplies a voltage to AV_{REF0} to an external circuit of the microcontroller as shown in Figure 11-25, AV_{REF0} = 0 V = AV_{SS} can be achieved and the external current can be reduced by satisfying the following conditions.

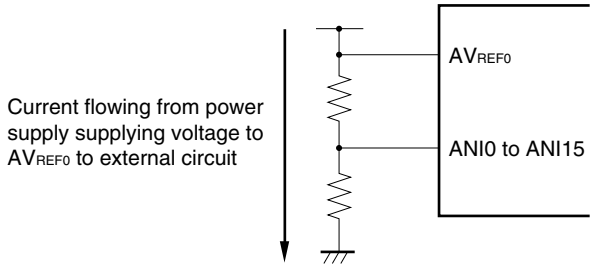
Set the following states before setting AV_{REF0} = 0 V.

- Set ADCS and ADCE of the A/D converter mode register (ADM) to 0.
- Set the port mode registers (PM20 to PM27 and PM150 to PM157) of the digital I/O pins to 1 to set to input mode, or set the digital I/O pins to low-level output (high-level output disabled) by setting the port mode registers (PM20 to PM27 and PM150 to PM157) and port registers (P20 to P27 and P150 to P157) to 0 to set to output mode.
- Make sure that no voltage is applied to all any of the analog or digital pins (P20/ANI0 to P27/ANI7 and P150/ANI8 to P157/ANI15) (set to 0 V).

Do not perform the following operation when AV_{REF0} = 0 V.

- Do not access the port registers (P20 to P27 and P150 to P157) or port mode registers (PM20 to PM27 and PM150 to PM157) by using instructions or via DMA transfer.

Figure 11-25. Example of Circuit Where Current Flows to External Circuit



When restarting the A/D converter, operate it after the AV_{REF0} voltage rises and stabilizes and setting ADCE = 1 (see **11.4.1 Basic operations of A/D converter** for the procedure for setting the A/D converter operation). Access digital ports after the AV_{REF0} voltage has risen and stabilized.

<R> Stop the conversion performed by the D/A converter when the AV_{REF0} voltage is rising or falling.

12.4.2 Operation in real-time output mode

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

The setting method is described below.

<1> Set the DAMDn bit of the DAM register to 1 (real-time output mode).
<2> Set the analog voltage value to be output to the ANOn pin to the DACSn register.
<3> Set the DACEn bit of the DAM register to 1 (D/A conversion enable).
Steps <1> to <3> above constitute the initial settings.
<4> Operate timer channel 4 and timer channel 5.
<5> D/A conversion starts and the analog voltage set in <2> is output to the ANOn pin when the INTTM04 and INTTM05 signals are generated.
The output level, however, is determined when the settling time elapses after D/A conversion starts.
<6> After that, the value set in the DACSn register is output every time the INTTM04 and INTTM05 signals are generated.
Set the analog voltage value to be output to the ANOn pin to the DACSn register before the next D/A conversion is started (INTTM04 and INTTM05 signals are generated).
When the DACEn bit of the DAM register is set to 0 (D/A conversion operation stop), analog voltage output is stopped, and the P110/ANO0 and P111/ANO1 pins can be used in port mode. At this time, the P110/ANO0 and P111/ANO1 pins are at high impedance because the PM11n bit of the PM11 register is 1 (input mode). The set value of the P11 register is output by setting the PM11n bit to 0 (output mode).

D/A conversion starts by setting the DACEn bit, as described in <3>, and an analog voltage is output to the ANOn pin, but the output value of the ANOn pin up to <5> is undefined. An arbitrary value, however, can be output in <3> by performing the following settings before performing the setting in <1>.

i. Set the DAMDn bit of the DAM register to 0 (normal mode).
ii. Set the voltage value output from the ANOn pin in <3> to the DACSn register.
iii. Afterward, perform <1> to <3>.
Consequently, the value set in ii can be output in <3>. The output level, however, is determined when the settling time elapses after D/A conversion starts.

Cautions 1. Make the interval for generating a start trigger to the same channel by one clock longer than f_{CLK}. If a start trigger is successively generated for every f_{CLK}, D/A conversion will be performed only at the first trigger.

2. Note the following points in the procedure (i to iii) for outputting an arbitrary value in <3>.

- **Do not generate the start trigger of the real-time output mode before enabling D/A conversion operation in <3> after the value is set to the DACSn register in ii.**
- **An arbitrary value cannot be output in <3> if the DACEN bit of the PER0 register is cleared once after the value is set to the DACSn register in ii.**

Remarks 1. For the output values of the ANO0 and ANO1 pins in the HALT and STOP modes, see **CHAPTER 19 STANDBY FUNCTION**.

2. n = 0, 1

3. f_{CLK}: CPU/peripheral hardware clock

(18) Port mode registers 0, 1, 4, 14 (PM0, PM1, PM4, PM14)

These registers set input/output of ports 0, 1, 4 and 14 in 1-bit units.

When using the P02/SO10/TxD1, P03/SI10/RxD1/SDA10, P04/$\overline{\text{SCK10}}$/SCL10, P10/$\overline{\text{SCK00}}$/EX24, P12/SO00/TxD0/EX26, P13/TxD3/EX27, P43/$\overline{\text{SCK01}}$, P45/SO01, P142/$\overline{\text{SCK20}}$/SCL20, P143/SI20/RxD2/SDA20, and P144/SO20/TxD2 pins for serial data output or serial clock output, clear the PM02, PM03, PM04, PM10, PM12, PM13, PM43, PM45, PM142, PM143, and PM144 bits to 0, and set the output latches of P02, P03, P04, P10, P12, P13, P43, P45, P142, P143, and P144 to 1.

When using the P03/SI10/RxD1/SDA10, P04/$\overline{\text{SCK10}}$/SCL10, P10/$\overline{\text{SCK00}}$/EX24, P11/SI00/RxD0/EX25, P14/RxD3/EX28, P43/$\overline{\text{SCK01}}$, P44/SI01, P142/$\overline{\text{SCK20}}$/SCL20, and P143/SI20/RxD2/SDA20 pins for serial data input or serial clock input, set the PM03, PM04, PM10, PM11, PM14, PM43, PM44, PM142, and PM143 bits to 1. At this time, the output latches of P03, P04, P10, P11, P14, P43, P44, P142, and P143 may be 0 or 1.

PM0, PM1, PM4, and PM14 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 13-21. Format of Port Mode Registers 0, 1, 4, and 14 (PM0, PM1, PM4, PM14)

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	1	PM06	PM05	PM04	PM03	PM02	PM01	PM00

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

Address: FFF24H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40

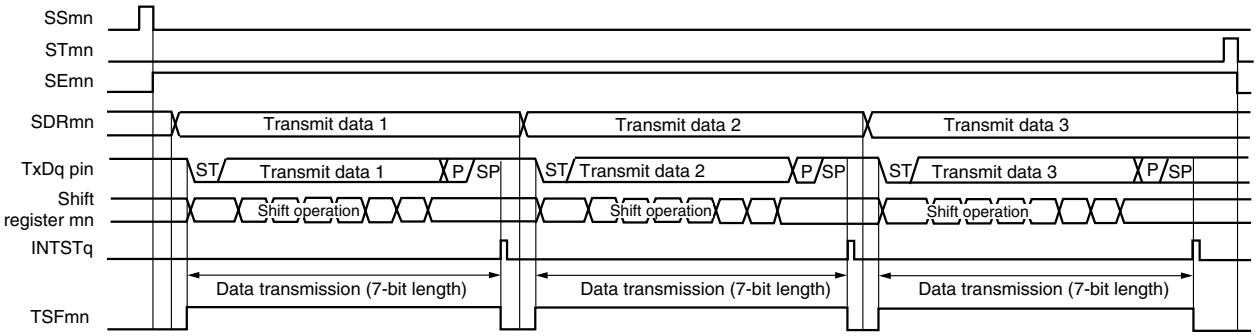
Address: FFF2EH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM14	1	1	PM145	PM144	PM143	PM142	PM141	PM140

PMmn	Pmn pin I/O mode selection (m = 0, 1, 4, 14; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

(3) Processing flow (in single-transmission mode)

Figure 13-75. Timing Chart of UART Transmission (in Single-Transmission Mode)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 3)

13.7 Operation of Simplified I²C (IIC10, IIC20) Communication

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This communication function is designed to execute single communication with devices such as EEPROM, flash memory, and A/D converter, and therefore, can be used only by the master and does not have a wait detection function. Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed.

[Data transmission/reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function[Note] and ACK detection function
- Data length of 8 bits
  (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Manual generation of start condition and stop condition

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

<R>
- Overrun error
- Parity error (ACK error)

* [Functions not supported by simplified I²C]
  - Slave transmission, slave reception
  - Arbitration loss detection function
  - Wait detection function

Note An ACK is not output when the last data is being received by writing 0 to the SOEmn (SOEm register) bit and stopping the output of serial communication data. See **13.7.3 (2) Processing flow** for details.

Remarks 1. To use the full-function I²C bus, see **CHAPTER 14 SERIAL INTERFACE IIC0**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

The channels supporting simplified I²C (IIC10, IIC20) are channel 2 of SAU0 and channel 0 of SAU1.

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	–
	1	CSI01		–
	2	CSI10	UART1	IIC10
	3	–		–
1	0	CSI20	UART2	IIC20
	1	–		–
	2	–	UART3 (supporting LIN-bus)	–
	3	–		–

Simplified I²C (IIC10, IIC20) performs the following four types of communication operations.

- Address field transmission (See **13.7.1**.)
- Data transmission (See **13.7.2**.)
- Data reception (See **13.7.3**.)
- Stop condition generation (See **13.7.4**.)

Table 13-9. Relationship Between Register Settings and Pins (Channel 0 of Unit 1: CSI20, UART2 Transmission, IIC20)

SE10 Note 1	MD102	MD101	SOE10	SO10	CKO10	TXE10	RXE10	PM142	P142	PM143 Note 2	P143 Note 2	PM144	P144	Operation Mode	Pin Function		
															$\overline{\text{SCK20}}$/SCL20/P142	SI20/SDA20/RxD2/P143 Note 2	SO20/TxD2/P144
0	0	0	0	1	1	0	0	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	Operation stop mode	P142	P143	P144
	0	1														P143/RxD2	
	1	0														P143	
1	0	0	0	1	1	0	1	1	×	1	×	× Note 3	× Note 3	Slave CSI20 reception	$\overline{\text{SCK20}}$ (input)	SI20	P144
			1	0/1 Note 4	1	1	0	1	×	× Note 3	× Note 3	0	1	Slave CSI20 transmission	$\overline{\text{SCK20}}$ (input)	P143	SO20
			1	0/1 Note 4	1	1	1	1	×	1	×	0	1	Slave CSI20 transmission/reception	$\overline{\text{SCK20}}$ (input)	SI20	SO20
			0	1	0/1 Note 4	0	1	0	1	1	×	× Note 3	× Note 3	Master CSI20 reception	$\overline{\text{SCK20}}$ (output)	SI20	P144
			1	0/1 Note 4	0/1 Note 4	1	0	0	1	× Note 3	× Note 3	0	1	Master CSI20 transmission	$\overline{\text{SCK20}}$ (output)	P143	SO20
			1	0/1 Note 4	0/1 Note 4	1	1	0	1	1	×	0	1	Master CSI20 transmission/reception	$\overline{\text{SCK20}}$ (output)	SI20	SO20
	0	1	1	0/1 Note 4	1	1	0	× Note 3	× Note 3	× Note 3	× Note 3	0	1	UART2 transmission Note 5	P142	P143/RxD2	TxD2
0	1	0	0	0/1 Note 6	0/1 Note 6	0	0	0	1	0	1	× Note 3	× Note 3	IIC20 start condition	SCL20	SDA20	P144
						1	0										
						0	1										
1			1	0/1 Note 4	0/1 Note 4	1	0	0	1	0	1	× Note 3	× Note 3	IIC20 address field transmission	SCL20	SDA20	P144
			1	0/1 Note 4	0/1 Note 4	1	0	0	1	0	1	× Note 3	× Note 3	IIC20 data transmission	SCL20	SDA20	P144
			1	0/1 Note 4	0/1 Note 4	0	1	0	1	0	1	× Note 3	× Note 3	IIC20 data reception	SCL20	SDA20	P144
0			0	0/1 Note 7	0/1 Note 7	0	0	0	1	0	1	× Note 3	× Note 3	IIC20 stop condition	SCL20	SDA20	P144
						1	0										
						0	1										

Notes 1. The SE1 register is a read-only status register which is set using the SS1 and ST1 registers.

2. When channel 1 of unit 1 is set to UART2 reception, this pin becomes an RxD2 function pin (refer to **Table 13-10**). In this case, operation stop mode or UART2 transmission must be selected for channel 0 of unit 1.

3. This pin can be set as a port function pin.

4. This is 0 or 1, depending on the communication operation. For details, refer to **13.3 (12) Serial output register m (SOm)**.

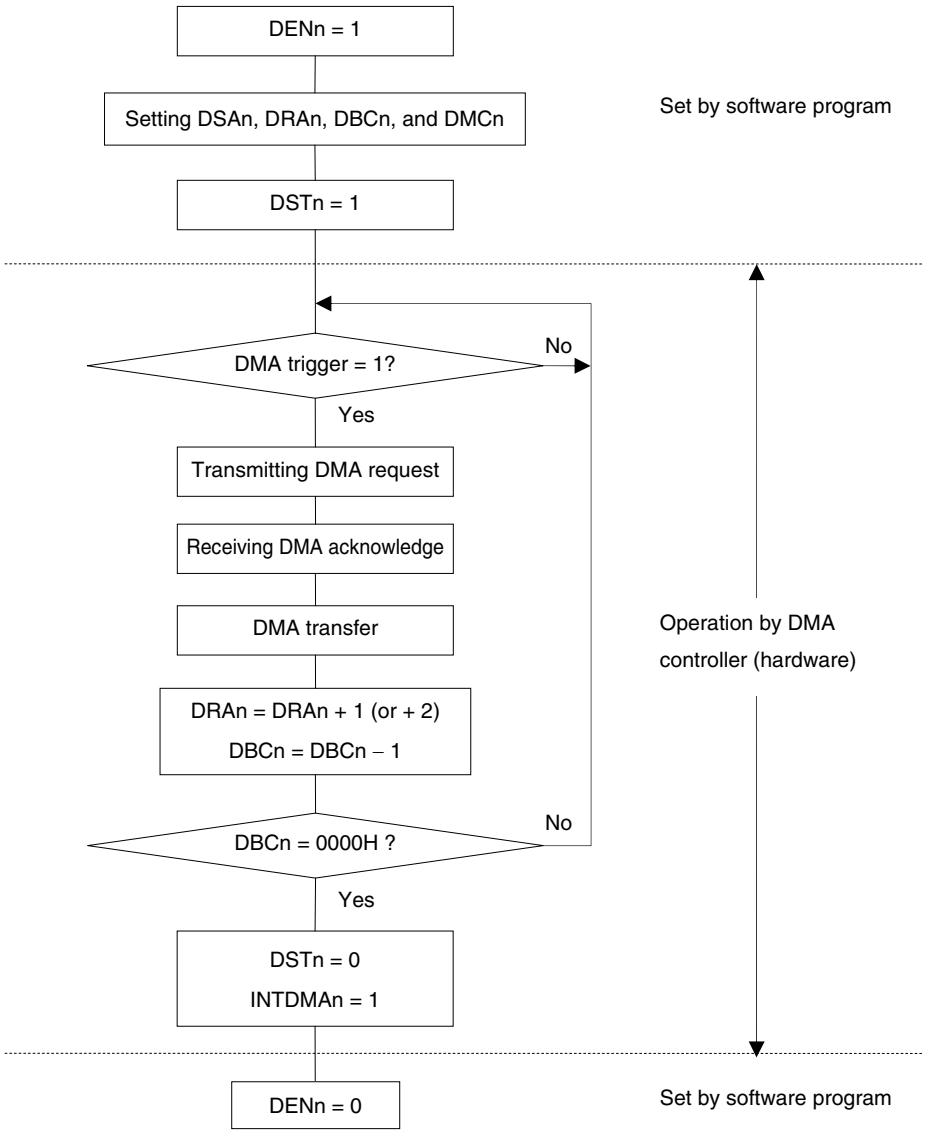
5. When using UART2 transmission and reception in a pair, set channel 1 of unit 1 to UART2 reception (refer to **Table 13-10**).

6. Set the CKO10 bit to 1 before a start condition is generated. Clear the SO10 bit from 1 to 0 when the start condition is generated.

7. Set the CKO10 bit to 1 before a stop condition is generated. Clear the SO10 bit from 0 to 1 when the stop condition is generated.

Remark X: Don't care

Figure 16-6. Operation Procedure



Remark n: DMA channel number (n = 0, 1)

CHAPTER 19 STANDBY FUNCTION

19.1 Standby Function and Configuration

19.1.1 Standby function

The standby function reduces the operating current of the system, and the following two modes are available.

(1) HALT mode

HALT instruction execution sets the HALT mode. In the HALT mode, the CPU operation clock is stopped. If the high-speed system clock oscillator, internal high-speed oscillator, or subsystem clock oscillator is operating before the HALT mode is set, oscillation of each clock continues. In this mode, the operating current is not decreased as much as in the STOP mode, but the HALT mode is effective for restarting operation immediately upon interrupt request generation and carrying out intermittent operations frequently.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the high-speed system clock oscillator and internal high-speed oscillator stop, stopping the whole system, thereby considerably reducing the CPU operating current.

Because this mode can be cleared by an interrupt request, it enables intermittent operations to be carried out. However, because a wait time is required to secure the oscillation stabilization time after the STOP mode is released when the X1 clock is selected, select the HALT mode if it is necessary to start processing immediately upon interrupt request generation.

In either of these two modes, all the contents of registers, flags and data memory just before the standby mode is set are held. The I/O port output latches and output buffer statuses are also held.

Cautions 1. The STOP mode can be used only when the CPU is operating on the main system clock. The STOP mode cannot be set while the CPU operates with the subsystem clock. The HALT mode can be used when the CPU is operating on either the main system clock or the subsystem clock.

2. When shifting to the STOP mode, be sure to stop the peripheral hardware operation operating with main system clock before executing STOP instruction.

3. The following sequence is recommended for operating current reduction of the A/D converter when the standby function is used: First clear bit 7 (ADCS) and bit 0 (ADCE) of the A/D converter mode register (ADM) to 0 to stop the A/D conversion operation, and then execute the STOP instruction.

4. It can be selected by the option byte whether the internal low-speed oscillator continues oscillating or stops in the HALT or STOP mode. For details, see CHAPTER 24 OPTION BYTE.

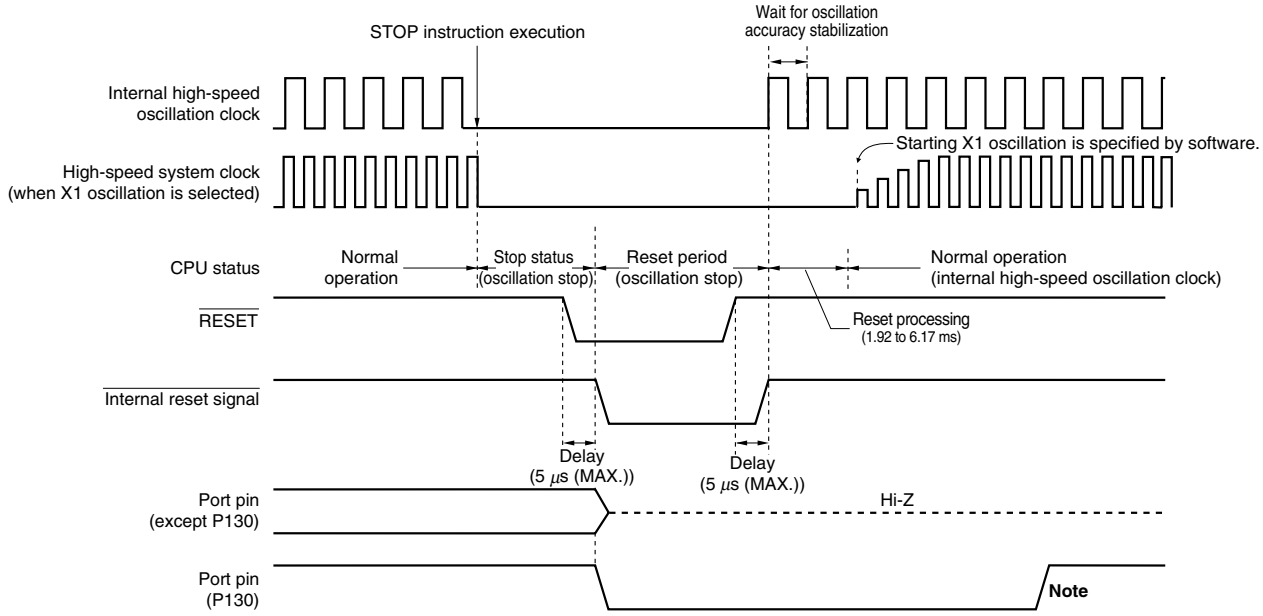
19.1.2 Registers controlling standby function

The standby function is controlled by the following two registers.

- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)

Remark For the registers that start, stop, or select the clock, see **CHAPTER 6 CLOCK GENERATOR**.

Figure 20-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input



Note Set P130 to high-level output by software.

Remarks 1. When reset is effected, P130 outputs a low level. If P130 is set to output a high level before reset is effected, the output signal of P130 can be dummy-output as the CPU reset signal.

2. For the reset timing of the power-on-clear circuit and low-voltage detector, see **CHAPTER 21 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 22 LOW-VOLTAGE DETECTOR**.

Table 23-1. Regulator Output Voltage Conditions

Mode	Output Voltage	Condition
Low consumption current mode	1.8 V	During system reset
		In STOP mode (except during OCD mode)
		When both the high-speed system clock (f_{MX}) and the internal high-speed oscillation clock (f_{IH}) are stopped during CPU operation with the subsystem clock (f_{XT})
		When both the high-speed system clock (f_{MX}) and the internal high-speed oscillation clock (f_{IH}) are stopped during the HALT mode when the CPU operation with the subsystem clock (f_{XT}) has been set
Normal current mode	2.5 V	Other than above

Table 26-1. Differences Between 1-Line Mode and 2-Line Mode

Communication Mode	Flash Memory Programming Function	Debugging Function
1-line mode	Available	• Pseudo real-time RAM monitor (RRM) function not supported
2-line mode	None	• Pseudo real-time RAM monitor (RRM) function supported

Remark 2-line mode is not used for flash programming, however, even if TOOL1 pin is connected with CLK_IN of QB-MINI2, writing is performed normally with no problem.

26.2 On-Chip Debug Security ID

The 78K0R/KG3 has an on-chip debug operation control bit in the flash memory at 000C3H (see **CHAPTER 24 OPTION BYTE**) and an on-chip debug security ID setting area at 000C4H to 000CDH, to prevent third parties from reading memory content.

When the boot swap function is used, also set a value that is the same as that of 010C3H and 010C4H to 010CDH in advance, because 000C3H, 000C4H to 000CDH and 010C3H, and 010C4H to 010CDH are switched.

For details on the on-chip debug security ID, refer to the **QB-MINI2 On-Chip Debug Emulator with Programming Function User's Manual (U18371)**.

Table 26-2. On-Chip Debug Security ID

Address	On-Chip Debug Security ID
000C4H to 000CDH	Any ID code of 10 bytes
010C4H to 010CDH	

26.3 Securing of User Resources

To perform communication between the 78K0R/KG3 and QB-MINI2, as well as each debug function, the securing of memory space must be done beforehand.

If NEC Electronics assembler RA78K0R or compiler CC78K0R is used, the items can be set by using linker options.

(1) Securing of memory space

The shaded portions in Figure 26-2 are the areas reserved for placing the debug monitor program, so user programs or data cannot be allocated in these spaces. When using the on-chip debug function, these spaces must be secured so as not to be used by the user program. Moreover, this area must not be rewritten by the user program.

Table 28-5. Operation List (2/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit data transfer	MOV	A, [HL + byte]	2	1	4	A ← (HL + byte)	
		[HL + byte], A	2	1	–	(HL + byte) ← A	
		A, [HL + B]	2	1	4	A ← (HL + B)	
		[HL + B], A	2	1	–	(HL + B) ← A	
		A, [HL + C]	2	1	4	A ← (HL + C)	
		[HL + C], A	2	1	–	(HL + C) ← A	
		word[B], #byte	4	1	–	(B + word) ← byte	
		A, word[B]	3	1	4	A ← (B + word)	
		word[B], A	3	1	–	(B + word) ← A	
		word[C], #byte	4	1	–	(C + word) ← byte	
		A, word[C]	3	1	4	A ← (C + word)	
		word[C], A	3	1	–	(C + word) ← A	
		word[BC], #byte	4	1	–	(BC + word) ← byte	
		A, word[BC]	3	1	4	A ← (BC + word)	
		word[BC], A	3	1	–	(BC + word) ← A	
		[SP + byte], #byte	3	1	–	(SP + byte) ← byte	
		A, [SP + byte]	2	1	–	A ← (SP + byte)	
		[SP + byte], A	2	1	–	(SP + byte) ← A	
		B, saddr	2	1	–	B ← (saddr)	
		B, !addr16	3	1	4	B ← (addr16)	
		C, saddr	2	1	–	C ← (saddr)	
		C, !addr16	3	1	4	C ← (addr16)	
		X, saddr	2	1	–	X ← (saddr)	
		X, !addr16	3	1	4	X ← (addr16)	
		ES:!addr16, #byte	5	2	–	(ES, addr16) ← byte	
		A, ES:!addr16	4	2	5	A ← (ES, addr16)	
		ES:!addr16, A	4	2	–	(ES, addr16) ← A	
		A, ES:[DE]	2	2	5	A ← (ES, DE)	
		ES:[DE], A	2	2	–	(ES, DE) ← A	
		ES:[DE + byte],#byte	4	2	–	((ES, DE) + byte) ← byte	
		A, ES:[DE + byte]	3	2	5	A ← ((ES, DE) + byte)	
		ES:[DE + byte], A	3	2	–	((ES, DE) + byte) ← A	

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum (except when branching to the external memory area).

3. In products where the external memory area is adjacent to the internal flash area, the number of waits is added to the number of instruction execution clocks placed in the last address (16-byte max.) in the flash memory, in order to use the external bus interface function. This should be done because, during pre-reading of the instruction code, an external memory wait being inserted due to an external memory area exceeding the flash space is accessed. For the number of waits, refer to **5.4 Number of Instruction Wait Clocks for Data Access**.

Table 28-5. Operation List (16/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Stack manipulate	PUSH	PSW	2	1	–	(SP – 1) ← PSW, (SP – 2) ← 00H, SP ← SP – 2			
		rp	1	1	–	(SP – 1) ← rp_H, (SP – 2)← rp_L, SP ← SP – 2			
	POP	PSW	2	3	–	PSW ← (SP + 1), SP ← SP + 2	R	R	R
		rp	1	1	–	rp_L ← (SP), rp_H ← (SP + 1), SP ← SP + 2			
	MOVW	SP, #word	4	1	–	SP ← word			
		SP, AX	2	1	–	SP ← AX			
		AX, SP	2	1	–	AX ← SP			
		HL, SP	3	1	–	HL ← SP			
		BC, SP	3	1	–	BC ← SP			
		DE, SP	3	1	–	DE ← SP			
	ADDW	SP, #byte	2	1	–	SP ← SP + byte			
	SUBW	SP, #byte	2	1	–	SP ← SP – byte			
Unconditional branch	BR	AX	2	3	–	PC ← CS, AX			
		$addr20	2	3	–	PC ← PC + 2 + jdisp8			
		$!addr20	3	3	–	PC ← PC + 3 + jdisp16			
		!addr16	3	3	–	PC ← 0000, addr16			
		!!addr20	4	3	–	PC ← addr20			
Conditional branch	BC	$addr20	2	2/4[Note 3]	–	PC ← PC + 2 + jdisp8 if CY = 1			
	BNC	$addr20	2	2/4[Note 3]	–	PC ← PC + 2 + jdisp8 if CY = 0			
	BZ	$addr20	2	2/4[Note 3]	–	PC ← PC + 2 + jdisp8 if Z = 1			
	BNZ	$addr20	2	2/4[Note 3]	–	PC ← PC + 2 + jdisp8 if Z = 0			
	BH	$addr20	3	2/4[Note 3]	–	PC ← PC+3+jdisp8 if (Z ∨ CY)=0			
	BNH	$addr20	3	2/4[Note 3]	–	PC ← PC+3+jdisp8 if (Z ∨ CY)=1			
	BT	saddr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 1			
		sfr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
		A.bit, $addr20	3	3/5[Note 3]	–	PC ← PC + 3 + jdisp8 if A.bit = 1			
		PSW.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if PSW.bit = 1			
<R>		[HL].bit, $addr20	3	3/5[Note 3]	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 1			
<R>		ES:[HL].bit, $addr20	4	4/6[Note 3]	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1			

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. This indicates the number of clocks "when condition is not met/when condition is met".

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum (except when branching to the external memory area).

3. In products where the external memory area is adjacent to the internal flash area, the number of waits is added to the number of instruction execution clocks placed in the last address (16-byte max.) in the flash memory, in order to use the external bus interface function. This should be done because, during pre-reading of the instruction code, an external memory wait being inserted due to an external memory area exceeding the flash space is accessed. For the number of waits, refer to **5.4 Number of Instruction Wait Clocks for Data Access**.

(5) Serial interface: On-chip debug (UART)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

(a) On-chip debug (UART)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate			$f_{CLK}/2^{12}$		$f_{CLK}/6$	bps
		Flash memory programming mode			2.66	Mbps
TOOL1 output frequency	f_{TOOL1}	2.7 V ≤ V_{DD} ≤ 5.5 V			10	MHz
		1.8 V ≤ V_{DD} < 2.7 V			2.5	MHz