Welcome to E-XFL.COM

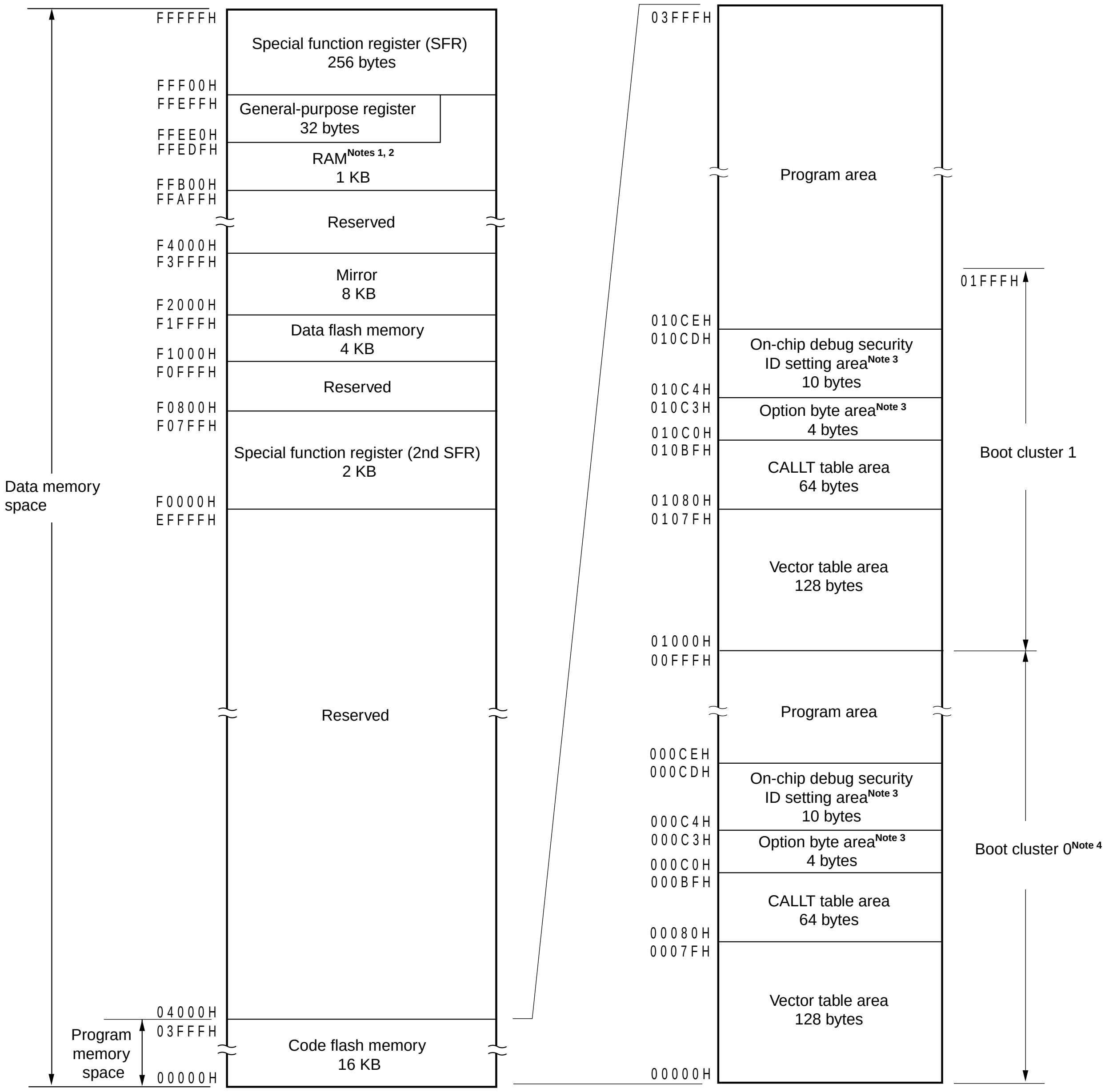
What is "Embedded - Microcontrollers"?

"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	RL78
Core Size	16-Bit
Speed	24MHz
Connectivity	CSI, I²C, LINbus, UART/USART
Peripherals	DMA, LCD, LVD, POR, PWM, WDT
Number of I/O	58
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	1.5K x 8
Voltage - Supply (Vcc/Vdd)	1.6V ~ 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	80-LQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/r5f10wmcafb-30

Figure 3-1. Memory Map (R5F10WLA, R5F10WMA)



Notes 1. Do not allocate RAM addresses which are used as a stack area, a data buffer, a branch destination of vector interrupt processing, and a DMA transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash 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 **28.6 Security Setting**).

Caution While RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed and the RAM area + 10 bytes when instructions are fetched from RAM areas, respectively.

Reset signal generation sets RAM parity error resets to enabled (RPERDIS = 0). For details, see 25.3.3 RAM parity error detection function.

3.2.4 Special function registers (SFRs)

Unlike a general-purpose register, each SFR has a special function.

SFRs are allocated to the FFF00H to FFFFFH area.

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 as follows for the 1-bit manipulation instruction operand (sfr.bit).
  When the bit name is defined: <Bit name>
  When the bit name is not defined: <Register name>.<Bit number> or <Address>.<Bit number>
- 8-bit manipulation
  Describe the symbol reserved by the assembler for the 8-bit manipulation instruction operand (sfr). 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 (sfrp). When specifying an address, describe an even address.

Table 3-5 gives a list of the SFRs. The meanings of items in the table are as follows.

- Symbol
  Symbol indicating the address of a special function register. It is a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler. When using the assembler, debugger, and simulator, symbols can be written as an instruction operand.
- R/W
  Indicates whether the corresponding 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 extended SFRs are not assigned.

Remark For extended SFRs (2nd SFRs), see **3.3.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

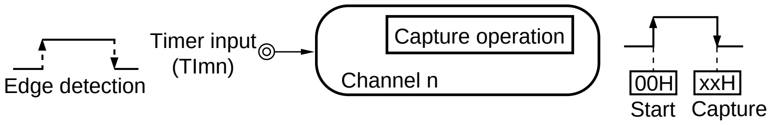
Table 4-6. Setting Examples of Registers and Output Latches When Using Alternate Function (10/15)

Pin Name	Used Function		PIOR×	POM××	PMC××	PM××	P××	PFSEGxx (PFDEG, ISCVL3, ISCCAP)[Note]	Alternate Function Output		80-pin	64-pin
	Function Name	I/O							SAU Output Function	Other than SAU		
P53	P53	Input	–	×	–	1	×	0	–	–	√	√
		Output		0		0	0/1					
		N-ch open drain output		1								
	INTP2	Input		×		1	×					
	SEG7	Output		0		0	0	1				
P54	P54	Input	–	×	–	1	×	0	–	–	√	√
		Output	×	0		0	0/1			TO02 = 0		
		N-ch open drain output		1								
	TI02	Input	PIOR0 = 0	×		1	×			–		
	TO02	Output	PIOR0 = 0	0		0	0			–		
	SEG8	Output	×	0		0	0	1		–		
P55	P55	Input	–	×	–	1	×	0	–	–	√	×
		Output	–	0		0	0/1					
		N-ch open drain output		1								
	INTP5	Input	PIOR4 = 0	×		1	×					
	SEG9	Output	×	0		0	0	1				
P56	P56	Input	–	×	–	1	×	0	–	–	√	×
		Output	×	0		0	0/1			TO06 = 0		
		N-ch open drain output		1								
	TI06	Input	PIOR0 = 0	×		1	×			–		
	TO06	Output	PIOR0 = 0	0		0	0			–		
	SEG10	Output	×	0		0	0	1		–		
P57	P57	Input	–	–	–	1	×	0	–	–	√	√
		Output				1	0/1					
	INTP6	Input				1	×					
	SEG11	Output				0	0	1				

Note PFDEG, ISCVL3, and ISCCAP are registers that correspond to P45, P125, and P126 and P127, respectively.

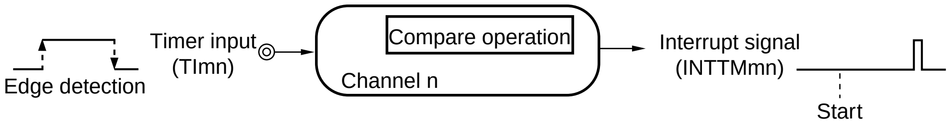
(5) Measurement of high-/low-level width of input signal

Counting is started by a single edge of the signal input to the timer input pin (TImn), and the count value is captured at the other edge. In this way, the high-level or low-level width of the input signal can be measured.



(6) Delay counter

Counting is started at the valid edge of the signal input to the timer input pin (TImn), and an interrupt is generated after any delay period.



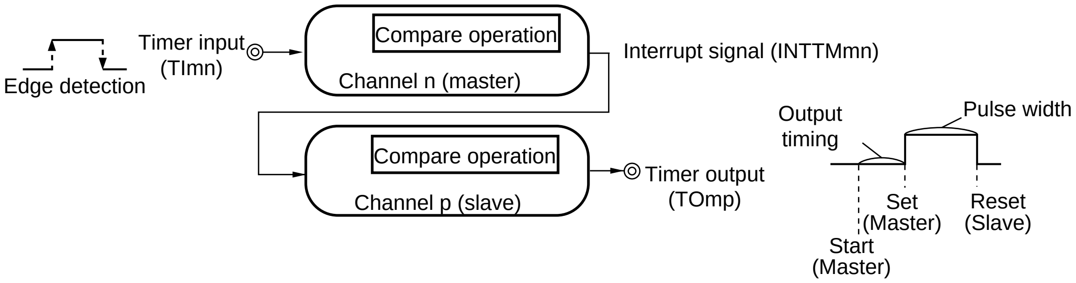
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.1.2 Simultaneous channel operation function

By using the combination of a master channel (a reference timer mainly controlling the cycle) and slave channels (timers operating according to the master channel), channels can be used for the following purposes.

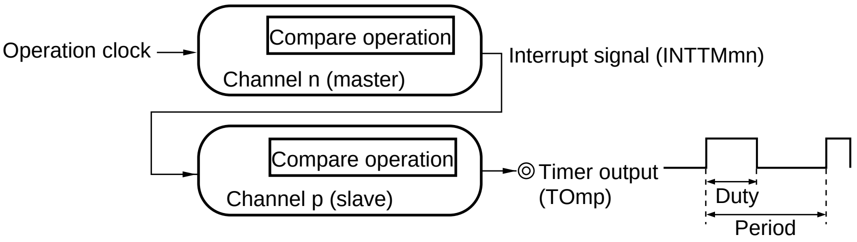
(1) One-shot pulse output

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



(2) PWM (Pulse Width Modulation) output

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



(**Caution** and **Remark** are listed on the next page.)

Figure 6-45. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (2/2)

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (master channel output mode)

(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets master channel output mode.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

8.3.11 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 is a decimal counter that count ups when the date counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks of f_{RTC} 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.

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

Reset signal generation does not clear this register to its default value.

Figure 8-12. Format of Month Count Register (MONTH)

Address: FFF97H After reset: Undefined R/W

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

Caution When reading or writing to MONTH while the clock counter operates (RTCE = 1), be sure to use the flows shown in 8.4.3 Reading real-time clock 2 counter and 8.4.4 Writing to real-time clock 2 counter.

8.3.12 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 is a decimal counter that 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 of f_{RTC} 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.

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

Reset signal generation does not clear this register to its default value.

Figure 8-13. Format of Year Count Register (YEAR)

Address: FFF98H After reset: Undefined R/W

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

Caution When reading or writing to YEAR while the clock counter operates (RTCE = 1), be sure to use the flows shown in 8.4.3 Reading real-time clock 2 counter and 8.4.4 Writing to real-time clock 2 counter.

Examples 2. When target correction value = –18.3 [ppm]

SUBCUD[8:0] = $(-18.3 \times 2^{15} / 10^{6})_{\text{2's complement (9 bit fixed-point format)}}$ + 0001.00000B
= $(-0.59965)_{\text{2's complement (9 bit fixed-point format)}}$ + 0001.00000B
= 1111.01101B + 0001.00000B
= 0000.01101B

CHAPTER 9 12-BIT INTERVAL TIMER

9.1 Functions of 12-bit Interval Timer

An interrupt (INTIT) is generated at any previously specified time interval. It can be used for waking the system up from STOP mode and triggering an A/D converter's SNOOZE mode.

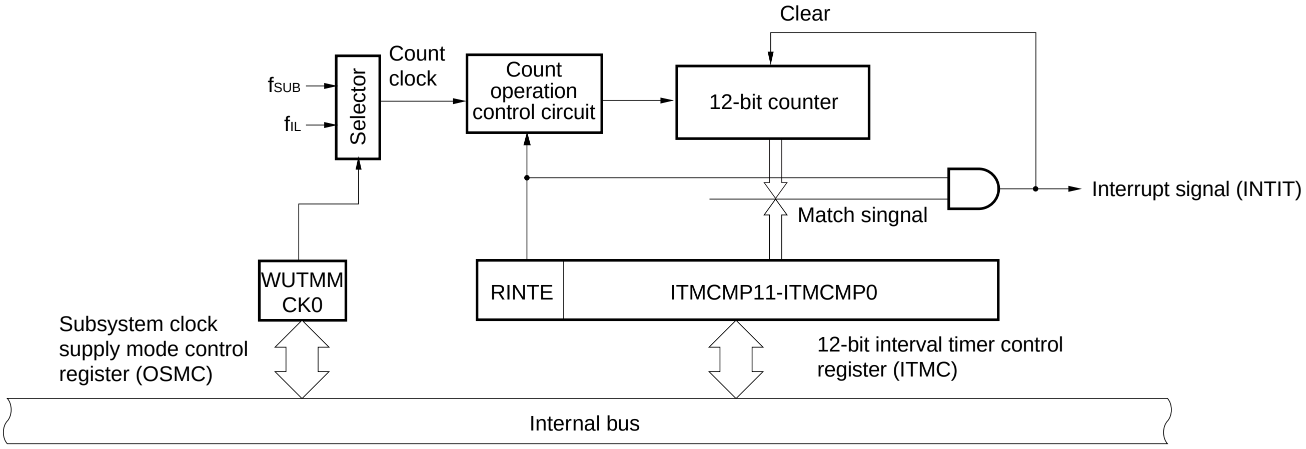
9.2 Configuration of 12-bit Interval Timer

The 12-bit interval timer includes the following hardware.

Table 9-1. Configuration of 12-bit Interval Timer

Item	Configuration
Counter	12-bit counter
Control registers	Peripheral enable register 1 (PER1)
	Subsystem clock supply mode control register (OSMC)
	12-bit interval timer control register (ITMC)

Figure 9-1. Block Diagram of 12-bit Interval Timer



14.1.2 UART (UART0 to UART3)

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel). The LIN-bus can be implemented by using timer array unit with an external interrupt (INTP0).

For details about the settings, see **14.6 Operation of UART (UART0 to UART3) Communication.**

[Data transmission/reception]

- Data length of 7, 8, or 9 bits[Note]
- Select the MSB/LSB first
- Select of reverse
- Parity bit appending and parity check functions
- Stop bit appending

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

In addition, UART0 and UART2 reception support the SNOOZE mode. When RxD input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. Only UART0, UART2 can be specified for asynchronous reception.

The LIN-bus is accepted in UART0 (0 and 1 channels of unit 0).

[LIN-bus functions]

- Wakeup signal detection
- Break field (BF) detection
- Sync field measurement, baud rate calculation

} Using the external interrupt (INTP0) and timer array unit

Note Only UART0, UART2 can be specified for the 9-bit data length.

(2) Operation procedure

Figure 14-26. Initial Setting Procedure for Master Transmission

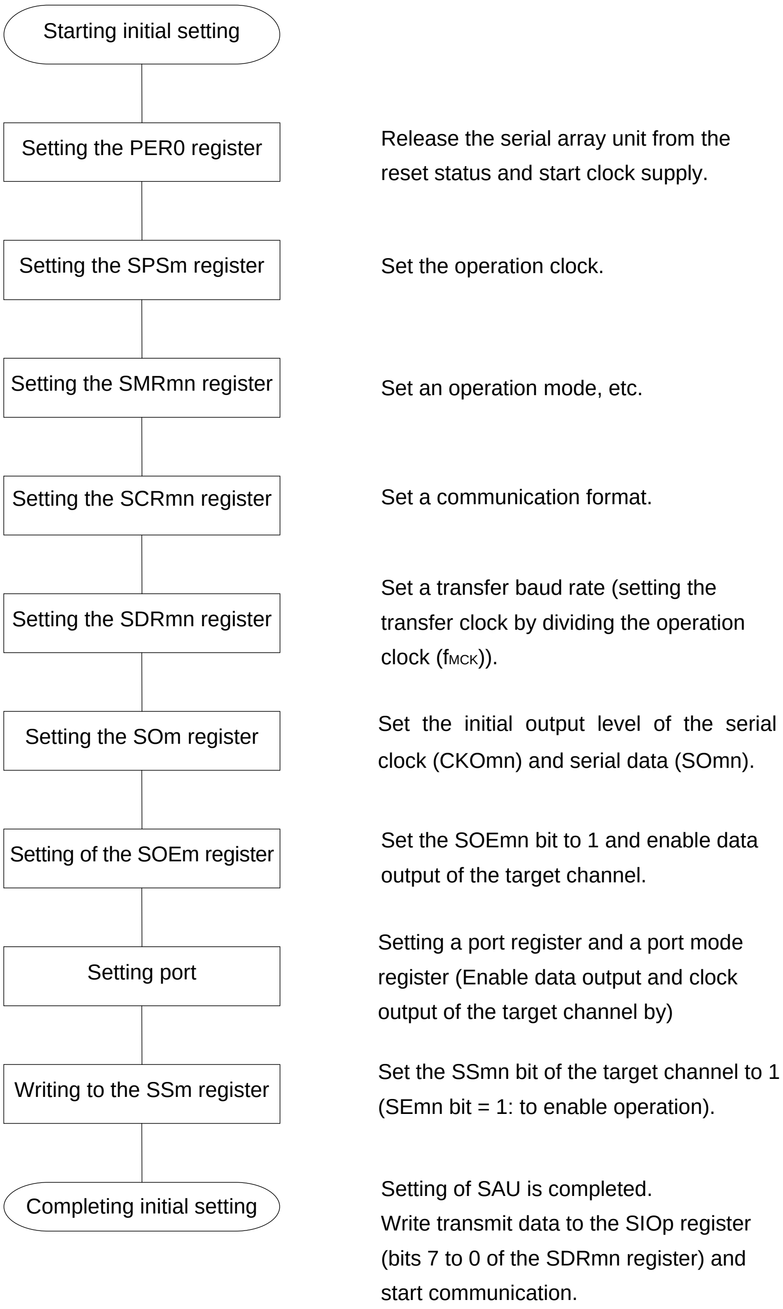
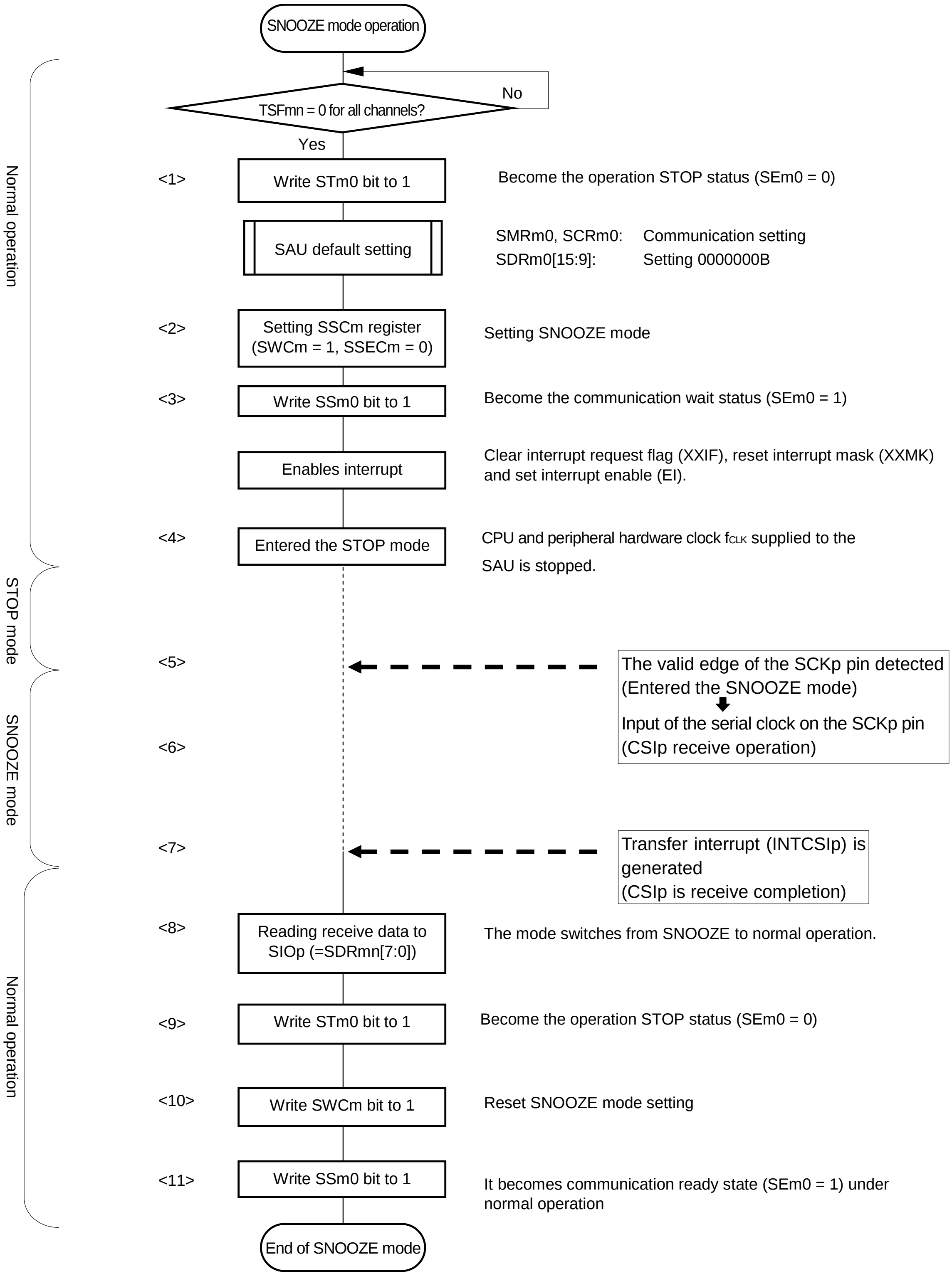

Starting initial setting
Setting the PER0 register
Release the serial array unit from the reset status and start clock supply.
Setting the SPSm register
Set the operation clock.
Setting the SMRmn register
Set an operation mode, etc.
Setting the SCRmn register
Set a communication format.
Setting the SDRmn register
Set a transfer baud rate (setting the transfer clock by dividing the operation clock (fMCK)).
Setting the SOm register
Set the initial output level of the serial clock (CKOmn) and serial data (SOmn).
Setting of the SOEm register
Set the SOEmn bit to 1 and enable data output of the target channel.
Setting port
Setting a port register and a port mode register (Enable data output and clock output of the target channel by)
Writing to the SSm register
Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation).
Completing initial setting
Setting of SAU is completed.
Write transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and start communication.

Figure 14-72. Flowchart of SNOOZE Mode Operation (Once Startup)

<R>

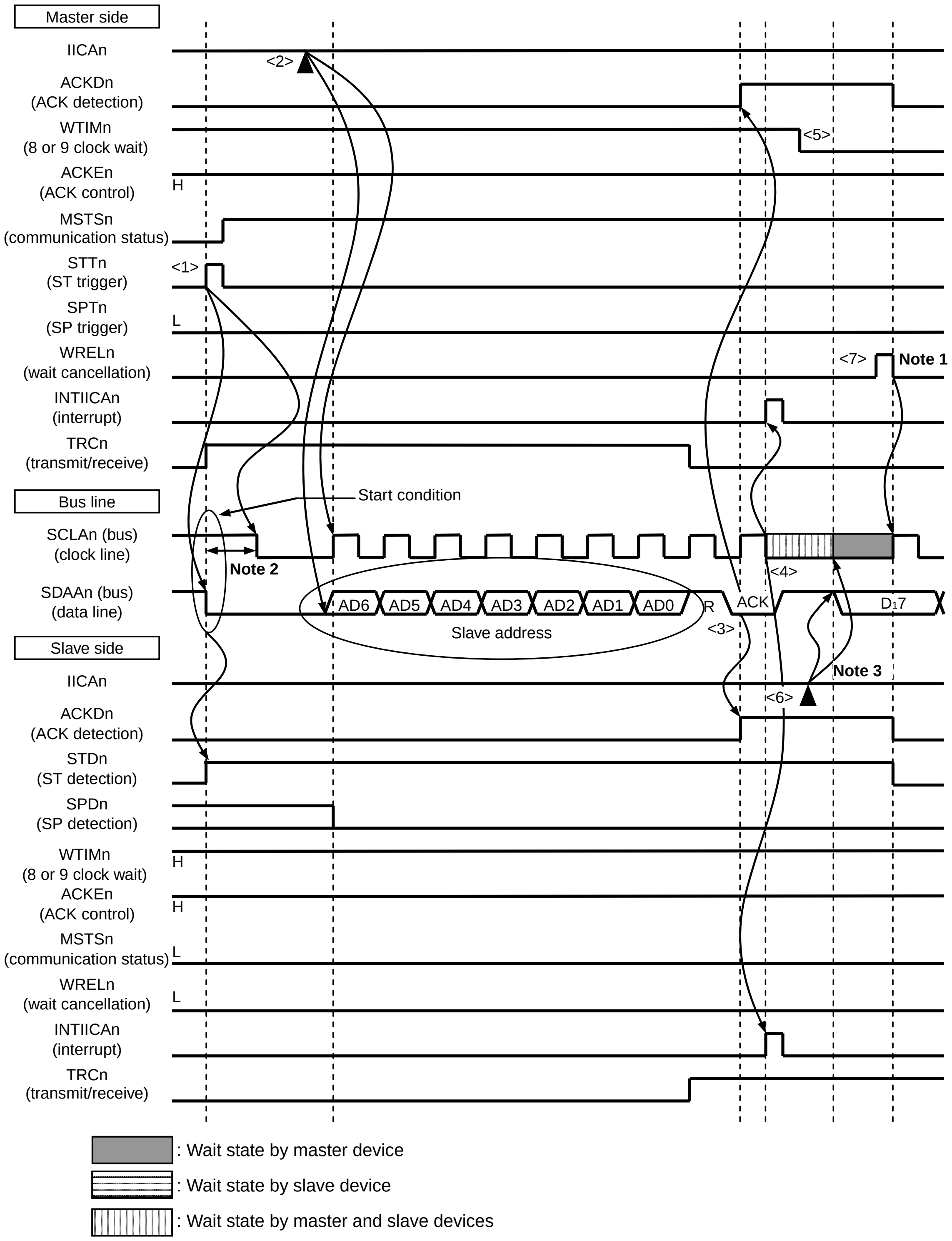

SNOOZE mode operation
TSFmn = 0 for all channels?
No
Yes
<1>
Write STm0 bit to 1
Become the operation STOP status (SEm0 = 0)
SAU default setting
SMRm0, SCRm0: Communication setting
SDRm0[15:9]: Setting 0000000B
<2>
Setting SSCm register (SWCm = 1, SSECm = 0)
Setting SNOOZE mode
<3>
Write SSm0 bit to 1
Become the communication wait status (SEm0 = 1)
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI).
<4>
Entered the STOP mode
CPU and peripheral hardware clock fCLK supplied to the SAU is stopped.
<5>
The valid edge of the SCKp pin detected (Entered the SNOOZE mode)
<6>
Input of the serial clock on the SCKp pin (CSIp receive operation)
<7>
Transfer interrupt (INTCSIp) is generated (CSIp is receive completion)
<8>
Reading receive data to SIOp (=SDRmn[7:0])
The mode switches from SNOOZE to normal operation.
<9>
Write STm0 bit to 1
Become the operation STOP status (SEm0 = 0)
<10>
Write SWCm bit to 1
Reset SNOOZE mode setting
<11>
Write SSm0 bit to 1
It becomes communication ready state (SEm0 = 1) under normal operation
End of SNOOZE mode
Normal operation
STOP mode
SNOOZE mode
Normal operation

Remarks 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 14-71 Timing Chart of SNOOZE Mode Operation (once startup)**.

2. m = 0; p = 00

Figure 15-33. Example of Slave to Master Communication (8-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (1/3)

(1) Start condition ~ address ~ data



Notes 1. For releasing wait state during reception of a master device, write “FFH” to IICAn or set the WRELn bit.

2. Make sure that the time between the fall of the SDAAn pin signal and the fall of the SCLAn pin signal is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

3. Write data to IICAn, not setting the WRELn bit, in order to cancel a wait state during transmission by a slave device.

Remark n = 0

Figure 16-1. Block Diagram of LCD Controller/Driver

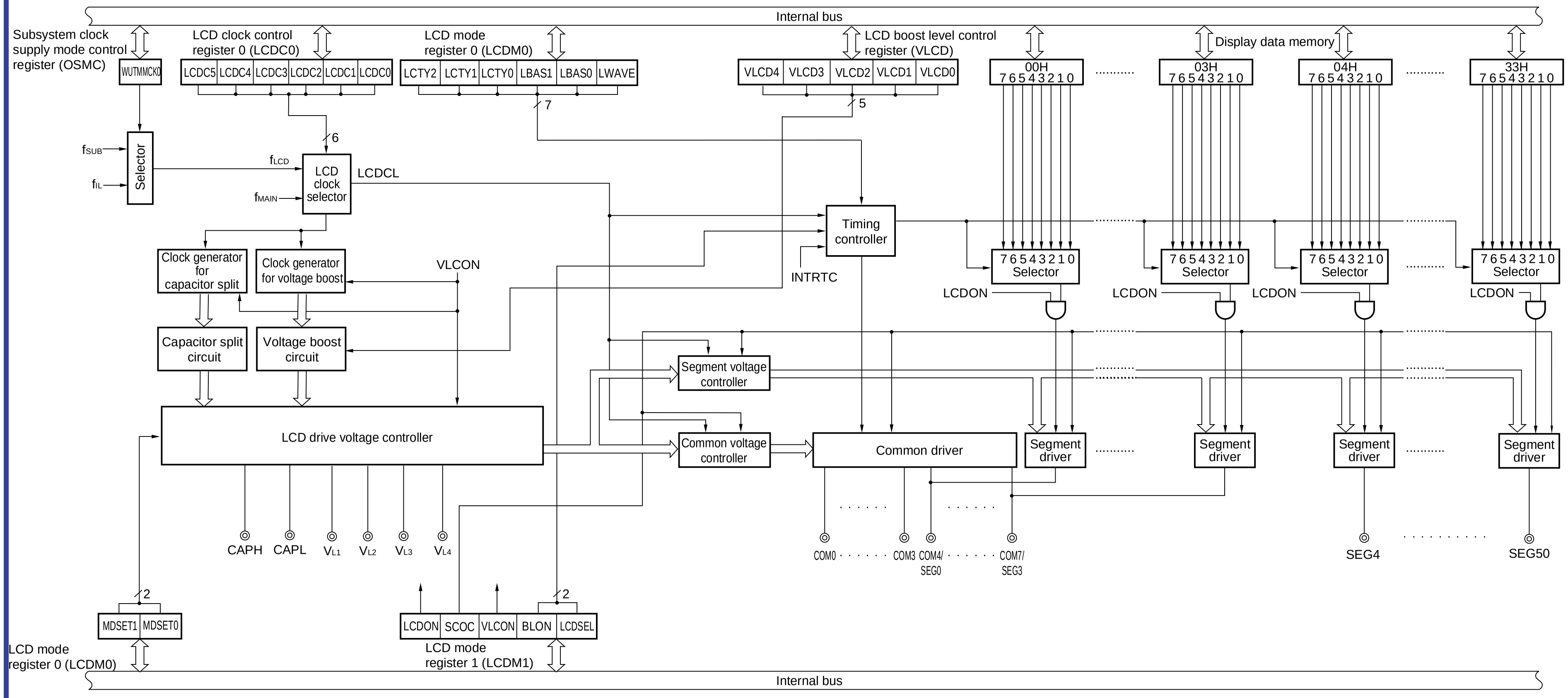

Internal bus
Subsystem clock supply mode control register (OSMC)
WUTMMCK0
LCD clock control register 0 (LCDC0)
LCDC5 LCDC4 LCDC3 LCDC2 LCDC1 LCDC0
LCD mode register 0 (LCDM0)
LCTY2 LCTY1 LCTY0 LBAS1 LBAS0 LWAVE
7
LCD boost level control register (VLCD)
VLCD4 VLCD3 VLCD2 VLCD1 VLCD0
5
Display data memory
00H 7 6 5 4 3 2 1 0
03H 7 6 5 4 3 2 1 0
04H 7 6 5 4 3 2 1 0
33H 7 6 5 4 3 2 1 0
6
fSUB
fIL
Selector
fLCD
fMAIN
LCD clock selector
LCDCL
Timing controller
INTRTC
7 6 5 4 3 2 1 0 Selector
LCDON
Clock generator for capacitor split
Clock generator for voltage boost
VLCON
Capacitor split circuit
Voltage boost circuit
Segment voltage controller
Common voltage controller
LCD drive voltage controller
Common driver
Segment driver
CAPH
CAPL
VL1
VL2
VL3
VL4
COM0
COM3
COM4/ SEG0
COM7/ SEG3
SEG4
SEG50
2
MDSET1 MDSET0
LCD mode register 0 (LCDM0)
LCDON SCOC VLCON BLON LCDSEL
LCD mode register 1 (LCDM1)

Table 16-10. Relationship Between LCD Display Data Register Contents and Segment/Common Outputs (2/4)

(a) Other than 8-time-slice (static, 2-time slice, 3-time slice, and 4-time slice) (2/2)

Register Name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	80-pin	64-pin
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0		
SEG33	F0422H	SEG33 (B-pattern area)				SEG33 (A-pattern area)				√	√
SEG34	F0423H	SEG34 (B-pattern area)				SEG34 (A-pattern area)				√	√
SEG35	F0424H	SEG35 (B-pattern area)				SEG35 (A-pattern area)				√	√
SEG36	F0425H	SEG36 (B-pattern area)				SEG36 (A-pattern area)				√	√
SEG37	F0426H	SEG37 (B-pattern area)				SEG37 (A-pattern area)				√	√
SEG38	F0427H	SEG38 (B-pattern area)				SEG38 (A-pattern area)				√	√
SEG39	F0428H	SEG39 (B-pattern area)				SEG39 (A-pattern area)				√	√
SEG40	F0429H	SEG40 (B-pattern area)				SEG40 (A-pattern area)				√	√
SEG41	F042AH	SEG41 (B-pattern area)				SEG41 (A-pattern area)				√	√
SEG42	F042BH	SEG42 (B-pattern area)				SEG42 (A-pattern area)				√	√
SEG43	F042CH	SEG43 (B-pattern area)				SEG43 (A-pattern area)				√	√
SEG44	F042DH	SEG44 (B-pattern area)				SEG44 (A-pattern area)				√	√
SEG45	F042EH	SEG45 (B-pattern area)				SEG45 (A-pattern area)				√	√
SEG46	F042FH	SEG46 (B-pattern area)				SEG46 (A-pattern area)				√	√
SEG47	F0430H	SEG47 (B-pattern area)				SEG47 (A-pattern area)				√	√
SEG48	F0431H	SEG48 (B-pattern area)				SEG48 (A-pattern area)				√	√
SEG49	F0432H	SEG49 (B-pattern area)				SEG49 (A-pattern area)				√	√
SEG50	F0433H	SEG50 (B-pattern area)				SEG50 (A-pattern area)				√	√

Remark √: Supported, –: Not supported

17.2.2 Multiplication/division data register B (MDBL, MDBH)

The MDBH and MDBL registers set the values that are used for multiplication or division operation and store the operation result. They store the operation result (product) in the multiplication mode and multiply-accumulator mode, and set the divisor data in the division mode.

The MDBH and MDBL registers can be set by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 17-3. Format of Multiplication/Division Data Register B (MDBH, MDBL)

Address: FFFF4H, FFFF5H, FFFF6H, FFFF7H After reset: 0000H, 0000H R/W

Symbol FFFF7H FFFF6H

Symbol																
MDBH	MDBH 15	MDBH 14	MDBH 13	MDBH 12	MDBH 11	MDBH 10	MDBH 9	MDBH 8	MDBH 7	MDBH 6	MDBH 5	MDBH 4	MDBH 3	MDBH 2	MDBH 1	MDBH 0

Symbol FFFF5H FFFF4H

Symbol																
MDBL	MDBL 15	MDBL 14	MDBL 13	MDBL 12	MDBL 11	MDBL 10	MDBL 9	MDBL 8	MDBL 7	MDBL 6	MDBL 5	MDBL 4	MDBHL 3	MDBL 2	MDBL 1	MDBL 0

Cautions 1. Do not rewrite the MDBH and MDBL registers values during division operation processing (when the multiplication/division control register (MDUC) value is 81H or C1H) or multiply-accumulation operation processing. The operation result will be an undefined value.

2. Do not set the MDBH and MDBL registers to 0000H in the division mode. If they are set, the operation result will be an undefined value.

3. The data is in the two's complement format in either the multiplication mode (signed) or multiply-accumulator mode (signed).

The following table shows the functions of the MDBH and MDBL registers during operation execution.

Table 17-3. Functions of MDBH and MDBL Registers During Operation Execution

Operation Mode	Setting	Operation Result
Multiplication mode (unsigned) Multiply-accumulator mode (unsigned)	–	MDBH: Multiplication result (product) (unsigned) Higher 16 bits MDBL: Multiplication result (product) (unsigned) Lower 16 bits
Multiplication mode (signed) Multiply-accumulator mode (signed)	–	MDBH: Multiplication result (product) (signed) Higher 16 bits MDBL: Multiplication result (product) (signed) Lower 16 bits
Division mode (unsigned)	MDBH: Divisor (unsigned) (higher 16 bits) MDBL: Divisor (unsigned) (lower 16 bits)	–

18.2 Configuration of DMA Controller

The DMA controller includes the following hardware.

Table 18-1. Configuration of DMA Controller

Item	Configuration
Address registers	• DMA SFR address registers 0 to 3 (DSA0 to DSA3) • DMA RAM address registers 0 to 3 (DRA0 to DRA3)
Count register	• DMA byte count registers 0 to 3 (DBC0 to DBC3)
Control registers	• DMA mode control registers 0 to 3 (DMC0 to DMC3) • DMA operation control registers 0 to 3 (DRC0 to DRC3)

18.2.1 DMA SFR address register n (DSAn)

This is an 8-bit register that is used to set an SFR address that is the transfer source or destination of DMA channel n.

Set the lower 8 bits of the SFR addresses FFF00H to FFFFFH.

This register is not automatically incremented but fixed to a specific value.

In the 16-bit transfer mode, the least significant bit is ignored and is treated as an even address.

The DSAn register can be read or written in 8-bit units. However, it cannot be written during DMA transfer.

Reset signal generation clears this register to 00H.

Figure 18-1. Format of DMA SFR Address Register n (DSAn)

Address: FFFB0H (DSA0), FFFB1H (DSA1), F0200H (DSA2), F0201H (DSA3) After reset: 00H R/W

	7	6	5	4	3	2	1	0
DSAn								

Remark n: DMA channel number (n = 0 to 3)

There are two flash memory programming modes: wide voltage mode and full speed mode. The supply voltage value applied to the microcontroller during write operations and the setting information of the user option byte for setting of the flash memory programming mode determine which mode is selected.

When a dedicated flash memory programmer is used for serial programming, setting the voltage on GUI selects the mode automatically.

Table 28-5. Programming Modes and Voltages at Which Data Can Be Written, Erased, or Verified

Power Supply Voltage (V_{DD})	User Option Byte Setting for Switching to Flash Memory Programming Mode		Flash Programming Mode
	Flash Operation Mode	Operating Frequency	
2.7 V ≤ V_{DD} ≤ 5.5 V	Blank state		Full speed mode
	HS (high speed main) mode	1 MHz to 24 MHz	Full speed mode
	LS (low speed main) mode	1 MHz to 8 MHz	Wide voltage mode
	LV (low voltage main) mode	1 MHz to 4 MHz	Wide voltage mode
2.4 V ≤ V_{DD} < 2.7 V	Blank state		Full speed mode
	HS (high speed main) mode	1 MHz to 16 MHz	Full speed mode
	LS (low speed main) mode	1 MHz to 8 MHz	Wide voltage mode
	LV (low voltage main) mode	1 MHz to 4 MHz	Wide voltage mode
1.8 V ≤ V_{DD} < 2.4 V	Blank state		Wide voltage mode
	LS (low speed main) mode	1 MHz to 8 MHz	Wide voltage mode
	LV (low voltage main) mode	1 MHz to 4 MHz	Wide voltage mode

Remarks 1. Using both the wide voltage mode and full speed mode imposes no restrictions on writing, deletion, or verification.

2. For details about communication commands, see **28.4.4 Communication commands**.

28.4.3 Selecting communication mode

Communication mode of the RL78 microcontroller as follows.

Table 28-6. Communication Modes

Communication Mode	Standard Setting[Note 1]				Pins Used
	Port	Speed[Note 2]	Frequency	Multiply Rate	
1-line mode (when flash memory programmer is used, or when external device is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	–	–	TOOL0
Dedicated UART (when external device is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	–	–	TOOLTxD, TOOLRxD

Notes 1. Selection items for Standard settings on GUI of the flash memory programmer.

2. Because factors other than the baud rate error, such as the signal waveform slew, also affect UART communication, thoroughly evaluate the slew as well as the baud rate error.

28.5.2 Boot swap function

If rewriting the boot area failed by temporary power failure or other reasons, restarting a program by resetting or overwriting is disabled due to data destruction in the boot area.

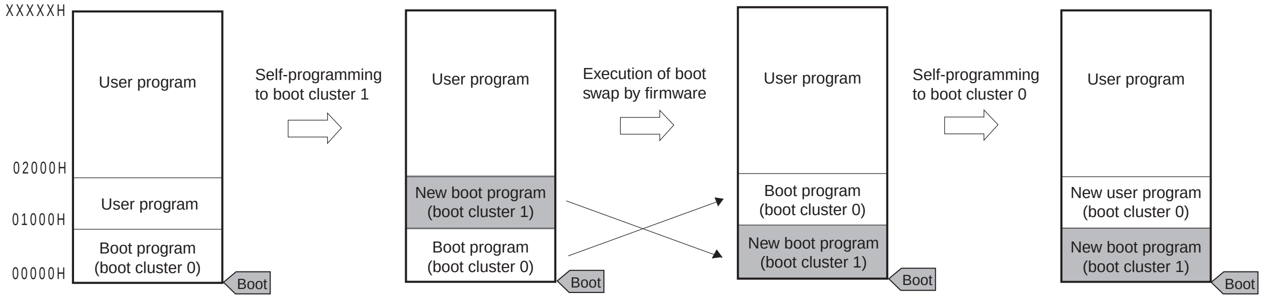
The boot swap function is used to avoid this problem.

Before erasing boot cluster 0[Note], which is a boot area, by self-programming, write a new boot program to boot cluster 1 in advance. When the program has been correctly written to boot cluster 1, swap this boot cluster 1 and boot cluster 0 by using the set information function of the firmware of the RL78 microcontroller, so that boot cluster 1 is used as a boot area. After that, erase or write the original area, boot cluster 0.

As a result, even if a power failure occurs while the area is being rewritten, the program is executed correctly because it is booted from boot cluster 1 to be swapped when the program is reset and started next.

Note A boot cluster is a 4 KB area and boot clusters 0 and 1 are swapped by the boot swap function.

Figure 28-9. Boot Swap Function



In an example of above figure, it is as follows.

Boot cluster 0: Boot area before boot swap

Boot cluster 1: Boot area after boot swap

(T_A = –40 to +85°C, 1.6 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

<R>

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low[Note 1]	I_{OL1}	Per pin for P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P70 to P77, P125 to P127, P130				20.0[Note 2]	mA
		Per pin for P60 and P61				15.0[Note 2]	mA
		Total of P40 to P47, P130 (When duty = 70%[Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			70.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			15.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			9.0	mA
			1.6 V ≤ V_{DD} < 1.8 V			4.5	mA
		Total of P00 to P07, P10 to P17, P22 to P27, P30 to P35, P50 to P57, P70 to P77, P125 to P127 (When duty = 70%[Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			90.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			35.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			20.0	mA
			1.6 V ≤ V_{DD} < 1.8 V			10.0	mA
		Total of all pins (When duty = 70%[Note 3])				160.0	mA
	I_{OL2}	Per pin for P20 and P21				0.4[Note 2]	mA
		Total of all pins (When duty = 70%[Note 3])	1.6 V ≤ V_{DD} ≤ 5.5 V			0.8	mA

Notes 1. Value of the current at which the device operation is guaranteed even if the current flows from an output pin to the V_{SS} pin

2. Do not exceed the total current value.

3. Output current value under conditions where the duty factor ≤ 70%.

The output current value that has changed to the duty factor > 70% the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n%).

- Total output current of pins = (I_{OL} × 0.7)/(n × 0.01)

  <Example> Where n = 80% and I_{OL} = 70.0 mA

  Total output current of pins = (70.0 × 0.7)/(80 × 0.01) ≅ 61.25 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

Notes 1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The values below the MAX. column include the peripheral operation current. However, not including the current flowing into the LCD controller/driver, A/D converter, LVD circuit, comparator, I/O port, on-chip pull-up/pull-down resistors, and the current flowing during data flash rewrite.

2. During HALT instruction execution by flash memory.

3. When high-speed on-chip oscillator and subsystem clock are stopped.

4. When high-speed system clock and subsystem clock are stopped.

5. When high-speed on-chip oscillator and high-speed system clock are stopped.
When RTCLPC = 1 and setting ultra-low current consumption (AMPHS1 = 1). The current flowing into the real-time clock 2 is included. However, not including the current flowing into the clock output/buzzer output, 12-bit interval timer, and watchdog timer.

6. Not including the current flowing into the real-time clock 2, clock output/buzzer output, 12-bit interval timer, and watchdog timer.

7. Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.

HS (high-speed main) mode: 2.7 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 24 MHz
2.4 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 16 MHz
LS (low-speed main) mode: 1.8 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 8 MHz
LV (low-voltage main) mode: 1.6 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 4 MHz

8. Regarding the value for current to operate the subsystem clock in STOP mode, refer to that in HALT mode.

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{HOCO}: High-speed on-chip oscillator clock frequency (48 MHz max.)

3. f_{IH}: High-speed on-chip oscillator clock frequency (24 MHz max.)

4. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

5. Except subsystem clock operation and STOP mode, temperature condition of the TYP. value is T_A = 25°C