Date: Aug. 30, 2016

RENESAS TECHNICAL UPDATE

TOYOSU FORESIA, 3-2-24, Toyosu, Koto-ku, Tokyo 135-0061, Japan
Renesas Electronics Corporation

Product Category	MPU/MCU		Document No.	TN-RL*-A063A/E	Rev.	1.00
Title	Correction for Incorrect Description Notice Descriptions in the RL78/I1B User's Manual: Hardware Rev. 2.00 Changed		Information Category	Technical Notification		
Applicable Product	RL78/I1B Group	Lot No. All lots	Reference Document	RL78/I1B User's Manual: Hardware Rev. 2.00 R01UH0407EJ0200 (Mar. 2014)		

This document describes misstatements found in the RL78/I1B User's Manual: Hardware Rev. 2.00 (R01UH0407EJ0200).

Corrections

(1/2)

Applicable Item	Applicable Page	Contents
1.6 Outline of Functions System Main Clock	p.10	Incorrect descriptions revised
3.1 Memory Space Figure 3-2	p.40	Incorrect descriptions revised
4.3.9 LCD port function registers 0 to 5 Figure 4-9	p.102	Caution added
4.3.9 LCD port function registers 0 to 5 Figure 4-9 Caution	p.102	Caution added
5.3.3 Clock operation status control register(CSC)Caution 6	p.133	Incorrect descriptions revised
5.6.5 Conditions before changing the CPU clock and processing after changing CPU clock Table 5-4	p.160,161	Incorrect descriptions revised
5.6.7 Conditions before stopping clock oscillation	p.163	Explanation added
7.3.3 Timer mode register mn Figure 7-12	p.193	Incorrect descriptions revised
7.3.8 Timer input select register 0	p.201	Incorrect descriptions revised
7.6.4 Collective manipulation of TOmn bit Figure7-38	p.228	Caution deleted
8.2 Configuration of Real-time Clock 2 Figure 8-1	p.281	Incorrect descriptions revised
8.3.6 Real-time clock control register 1 Note 1, 2	p.289	Caution added
12.5 Cautions of Clock Output/Buzzer Output Controller	p.347	Incorrect descriptions revised
14.3.2 A/D converter mode register 0 Figure 14-4	p.364	Incorrect descriptions revised
18.2 Configuration of Serial Array Unit Figure 18-1	p.455	Incorrect descriptions revised
18.2 Configuration of Serial Array Unit Figure 18-2	p.456	Incorrect descriptions revised
18.3.5 Serial data register mn	p.466	Incorrect descriptions revised
18.3.12 Serial output register m	p.475	Incorrect descriptions revised
18.3.13 Serial output level register m Figure 18-18	p.477	Incorrect descriptions revised
18.5.7 SNOOZE mode function	p.543	Incorrect descriptions revised
18.5.7 SNOOZE mode function Figure 18-71	p.543	Incorrect descriptions revised
18.5.7 SNOOZE mode function Note	p.543	Incorrect descriptions revised
18.5.7 SNOOZE mode function Figure 18-72	p.544	Incorrect descriptions revised
18.5.7 SNOOZE mode function Figure 18-73	p.545	Incorrect descriptions revised
18.5.7 SNOOZE mode function Note	p.545	Incorrect descriptions revised
18.5.7 SNOOZE mode function Figure 18-74	P546	Incorrect descriptions revised
18.6.3 SNOOZE mode function Description	p.569	Incorrect descriptions revised
18.6.3 SNOOZE mode function Note 5	p.569	Caution added

(2/2)

Applicable Item	Applicable Page	Contents
18.6.3 SNOOZE mode function Figure 18-90	p.571	Incorrect descriptions revised
18.6.3 SNOOZE mode function Figure 18-91	p.572	Incorrect descriptions revised
18.6.3 SNOOZE mode function Figure 18-92	p.573	Incorrect descriptions revised
18.6.3 SNOOZE mode function Figure 18-93	p.574	Incorrect descriptions revised
18.6.3 SNOOZE mode function Figure 18-94	p.575	Incorrect descriptions revised
18. 7. 1 LIN transmission Figure 18-99	p.584	Incorrect descriptions revised
18. 7. 2 LIN reception Figure 18-100	p.586	Incorrect descriptions revised
19.3.6 IICA low-level width setting register n	p.629	Explanation added
19.5.14 Communication reservation Calculation formula	P.648	Incorrect descriptions revised
19.5.14 Communication reservation Figure 19-27	p.650	Incorrect descriptions revised
19.5.16 Communication operations Figure 19-28	p.654	Incorrect descriptions revised
19.5.16 Communication operations Figure 19-29 (1/3)	p.655	Incorrect descriptions revised
19.5.16 Communication operations Figure 19-29 (2/3)	p.656	Incorrect descriptions revised
19.5.16 Communication operations Figure 19-29 Note	p.656	Incorrect descriptions revised
19.5.16 Communication operations Figure 19-30	p.659	Incorrect descriptions revised
21.3.7 LCD port function registers 0 to 5	p.718	Incorrect descriptions revised
22.1 Functions of DTC Table 22-1	p.765	Incorrect descriptions revised
22.4.2 Normal mode	p.780	Incorrect descriptions revised
22.5.3 DTC pending instruction	p.787	Incorrect descriptions revised
23.4.4 Interrupt request hold	p.813	Section explanation added
23.4.5 Interrupt request hold	p.813	Explanation added
25.1 Timing of Reset Operation	p.835	Caution deleted
26.3 Operation of Power-on-reset Circuit Figure26-2 Note3	p.845	Incorrect descriptions revised
27.1 Functions of Voltage Detector	p.847	Incorrect descriptions revised
27.1 Functions of Voltage Detector Table	p.847	Incorrect descriptions revised
33.1.2 Communication mode Figure 33-2	p.910	Incorrect descriptions revised
33.5 Self-Programming Remark 1	p.920	Incorrect descriptions revised
36. 2 Operation List	p.947	Caution added
37.3.1 Pin characteristics On-chip pull-up resistance	p.964	Incorrect descriptions revised
37.3.2 Supply current characteristics Supply current	p.965	Incorrect descriptions revised
37.6.2 24-bit ΔΣ A/D converter characteristics	p.1005	Incorrect descriptions revised
37.9 RAM Data Retention Characteristics Title	p.1014	Incorrect descriptions revised
37.9 RAM Data Retention Characteristics Note	p.1014	Incorrect descriptions revised
37.9 RAM Data Retention Characteristics Figure	p.1014	Incorrect descriptions revised

1.6 Outline of Functions

Incorrect:

(1/2)

Item		80-pin		100-pin	
		R5F10MMEDFB	R5F10MMGDFB	R5F10MPEDFB	R5F10MPGDFB
Code flash memory (KB)		64	128	64	128
Data flash memory (KB)		–			
RAM (KB)		6	8 Note 1	6	8 Note 1
Address space		1 MB			
Main system clock	High-speed system clock	X1 (crystal/ceramic) oscillation, external main system clock input (EXCLK) 1 to 20 MHz: V_{DD} = 2.7 to 5.5 V, 1 to 8 MHz: V_{DD} = 1.9 to 2.7 V			
	High-speed on-chip oscillator clock	HS (High-speed main) mode: 24/12/6/3 MHz (V_{DD} = 2.7 to 5.5 V), HS (High-speed main) mode: 12/6/3 MHz (V_{DD} = 2.4 to 5.5 V), LS (Low-speed main) mode: 6/3 MHz (V_{DD} = 1.9 to 5.5 V)			
Subsystem clock		XT1 (crystal) oscillation, external subsystem clock input (EXCLKS) 32.768 kHz (TYP.): V_{DD} = 1.9 to 5.5 V			
High-speed on-chip oscillator clock frequency correction function		Correct the frequency of the high-speed on-chip oscillator clock by the subsystem clock.			
Low-speed on-chip oscillator		15 kHz (TYP.): V_{DD} = 1.9 to 5.5 V			
General-purpose register		8 bits × 8 registers × 4 banks			
Minimum instruction execution time		0.04167 μs (High-speed on-chip oscillator: f_{IH} = 24 MHz operation)			
		0.05 μs (High-speed system clock: f_{MX} = 20 MHz operation)			
		30.5 μs (Subsystem clock: f_{SUB} = 32.768 kHz operation)			
Instruction set		• Data transfer (8/16 bits) • Adder and subtractor/logical operation (8/16 bits) • Multiplication (16 bits × 16 bits), division (32 bits ÷ 32 bits) • Multiplication and accumulation (16 bits × 16 bits + 32 bits) • Rotate, barrel shift, and bit manipulation (set, reset, test, and boolean operation), etc.			
I/O port	Total	53		69	
	CMOS I/O	44		60	
	CMOS input	5		5	
	CMOS output	1		1	
	N-ch O.D I/O (6 V tolerance)	3		3	
Timer	16-bit timer TAU	8 channels			
	Watchdog timer	1 channel			
	12-bit interval timer	1 channel			
	8-bit interval timer	4 channels			
	Real-time clock 2	1 channel			
	Oscillation stop detection circuit	1 channel			
	Timer output	Timer outputs: 8 channels PWM outputs: 7 Note 2			
	RTC output	1 channel • 1 Hz (subsystem clock: f_{SUB} = 32.768 kHz)			

Notes 1. In the case of the 8 KB, this is about 7 KB when the self-programming function is used.

2. The number of outputs varies, depending on the setting of channels in use and the number of the master (see **7.9.3 Operation as multiple PWM output function**).

1.6 Outline of Functions

Correct:

(1/2)

<R>

Item		80-pin		100-pin	
		R5F10MMEDFB	R5F10MMGDFB	R5F10MPEDFB	R5F10MPGDFB
Code flash memory (KB)		64	128	64	128
Data flash memory (KB)		–			
RAM (KB)		6	8 Note 1	6	8 Note 1
Address space		1 MB			
Main system clock	High-speed system clock	X1 (crystal/ceramic) oscillation, external main system clock input (EXCLK) HS (High-speed main) mode: 1 to 20 MHz (V_{DD} = 2.7 to 5.5 V), HS (High-speed main) mode: 1 to 16 MHz (V_{DD} = 2.4 to 5.5 V), LS (Low-speed main) mode: 1 to 8 MHz (V_{DD} = 1.9 to 5.5 V)			
	High-speed on-chip oscillator clock	HS (High-speed main) mode: 24/12/6/3 MHz (V_{DD} = 2.7 to 5.5 V), HS (High-speed main) mode: 12/6/3 MHz (V_{DD} = 2.4 to 5.5 V), LS (Low-speed main) mode: 6/3 MHz (V_{DD} = 1.9 to 5.5 V)			
Subsystem clock		XT1 (crystal) oscillation, external subsystem clock input (EXCLKS) 32.768 kHz (TYP.): V_{DD} = 1.9 to 5.5 V			
High-speed on-chip oscillator clock frequency correction function		Correct the frequency of the high-speed on-chip oscillator clock by the subsystem clock.			
Low-speed on-chip oscillator		15 kHz (TYP.): V_{DD} = 1.9 to 5.5 V			
General-purpose register		8 bits × 8 registers × 4 banks			
Minimum instruction execution time		0.04167 μs (High-speed on-chip oscillator: f_{IH} = 24 MHz operation)			
		0.05 μs (High-speed system clock: f_{MX} = 20 MHz operation)			
		30.5 μs (Subsystem clock: f_{SUB} = 32.768 kHz operation)			
Instruction set		• Data transfer (8/16 bits) • Adder and subtractor/logical operation (8/16 bits) • Multiplication (16 bits × 16 bits), division (32 bits ÷ 32 bits) • Multiplication and accumulation (16 bits × 16 bits + 32 bits) • Rotate, barrel shift, and bit manipulation (set, reset, test, and boolean operation), etc.			
I/O port	Total	53		69	
	CMOS I/O	44		60	
	CMOS input	5		5	
	CMOS output	1		1	
	N-ch O.D I/O (6 V tolerance)	3		3	
Timer	16-bit timer TAU	8 channels			
	Watchdog timer	1 channel			
	12-bit interval timer	1 channel			
	8-bit interval timer	4 channels			
	Real-time clock 2	1 channel			
	Oscillation stop detection circuit	1 channel			
	Timer output	Timer outputs: 8 channels PWM outputs: 7 Note 2			
	RTC output	1 channel • 1 Hz (subsystem clock: f_{SUB} = 32.768 kHz)			

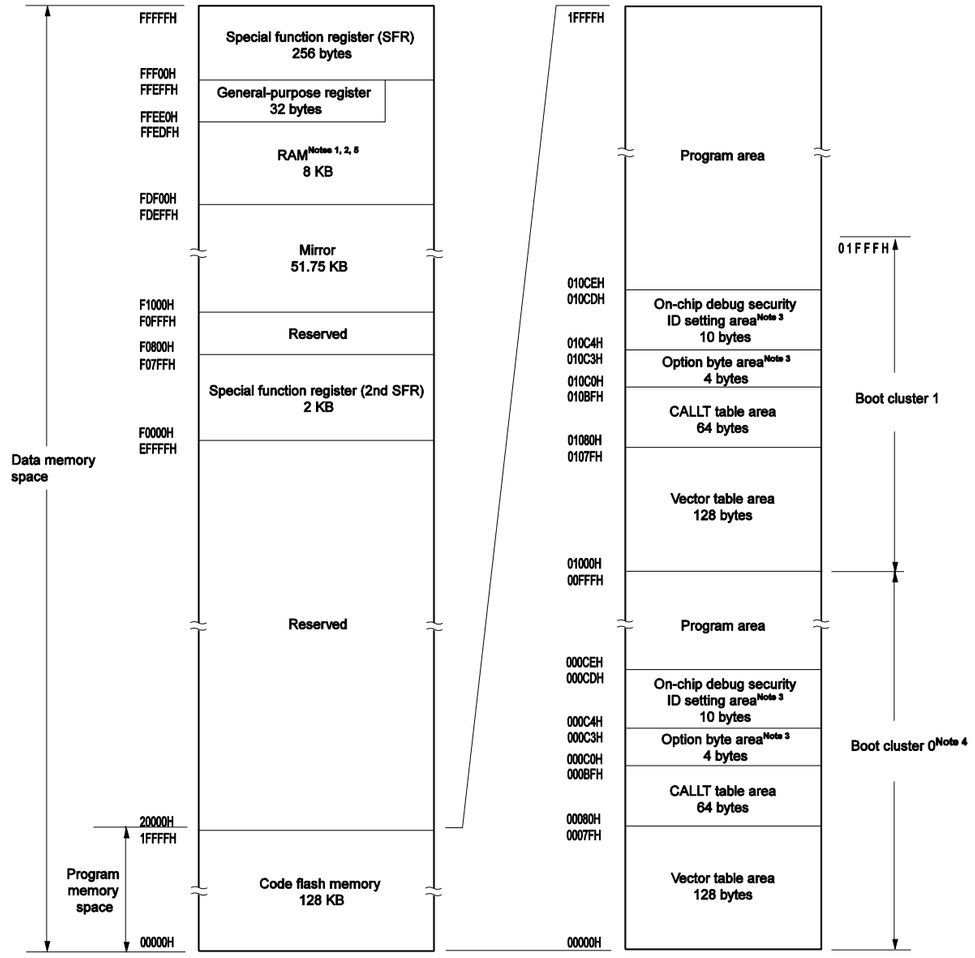
Notes 1. In the case of the 8 KB, this is about 7 KB when the self-programming function is used.

2. The number of outputs varies, depending on the setting of channels in use and the number of the master (see **7.9.3 Operation as multiple PWM output function**).

3.1 Memory Space

Incorrect:

Figure 3-2. Memory Map (R5F10MMG, R5F10MPG)



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 DTC transfer destination/transfer source to the areas FFE20H to FFEDFH and FDF00H to FE309H when performing self-programming.

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.

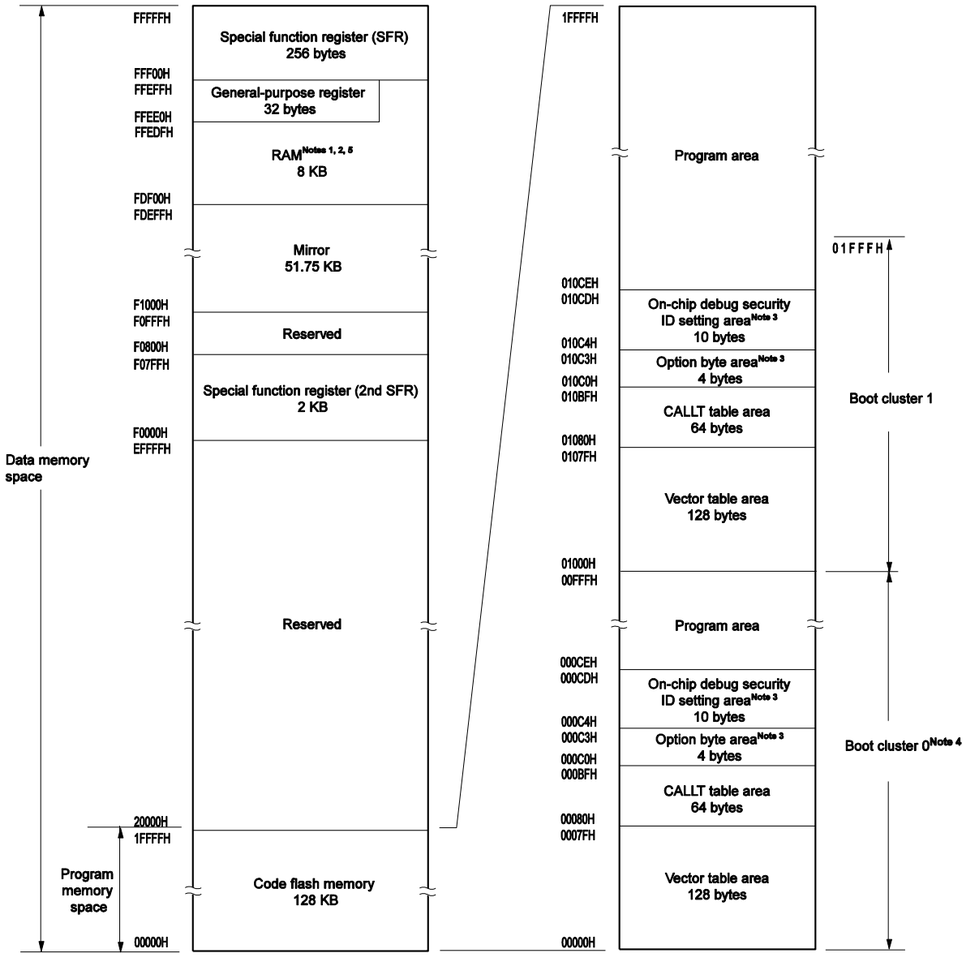
4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **33.6 Security Settings**).

5. When using the trace function of on-chip debugging, area FE300H to FE6FFH is disabled.

3.1 Memory Space

Correct:

Figure 3-2. Memory Map (R5F10MMG, R5F10MPG)



<R> **Notes 1.** Do not allocate the stack area, data buffers for use by the flash library, arguments of library functions, branch destinations in the processing of vectored interrupts, or destinations or sources for DTC transfer to the area from FFE20H to FFEDFH when performing self-programming. The RAM area used by the flash library starts at FDF00H. For the RAM areas used by the flash library, see Self RAM list of Flash Self-Programming Library for RL78 Family (R20UT2944).

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 **33.6 Security Settings**).

5. When using the trace function of on-chip debugging, area FE300H to FE6FFH is disabled.

4.3.9 LCD port function registers 0 to 5 (PFSEG0 to PFSEG5)

Incorrect:

These registers set whether to use pins P10 to P17, P30 to P37, P50 to P57, P70 to P77, P80 to P85 as port pins (other than segment output pins) or segment output pins.

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

Reset signal generation sets these registers to FFH (PFSEG0 is set to F0H, and PFSEG5 is set to 03H).

Remark The correspondence between the segment output pins (SEGxx) and the PFSEG register (PFSEGxx bits) and the existence of SEGxx pins in each product are shown in Table 4-4 Segment Output Pins in Each Product and Correspondence with PFSEG Register (PFSEG Bits).

Figure 4-9. Format of **LCD port function registers 0 to 5 (PFSEG0 to PFSEG5)**

Address: F0300H After reset: F0H R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG0	PFSEG07	PFSEG06	PFSEG05	PFSEG04	0	0	0	0

Address: F0301H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG1	PFSEG15	PFSEG14	PFSEG13	PFSEG12	PFSEG11	PFSEG10	PFSEG09	PFSEG08

Address: F0302H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG2	PFSEG23	PFSEG22	PFSEG21	PFSEG20	PFSEG19	PFSEG18	PFSEG17	PFSEG16

Address: F0303H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG3	PFSEG31	PFSEG30	PFSEG29	PFSEG28	PFSEG27	PFSEG26	PFSEG25	PFSEG24

Address: F0304H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG4	PFSEG39	PFSEG38	PFSEG37	PFSEG36	PFSEG35	PFSEG34	PFSEG33	PFSEG32

Address: F0305H After reset: 03H R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG5	0	0	0	0	0	0	PFSEG41	PFSEG40

4.3.9 LCD port function registers 0 to 5 (PFSEG0 to PFSEG5)

Correct:

These registers set whether to use pins P10 to P17, P30 to P37, P50 to P57, P70 to P77, P80 to P85 as port pins (other than segment output pins) or segment output pins.

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

Reset signal generation sets these registers to FFH (PFSEG0 is set to F0H, and PFSEG5 is set to 03H).

Remark The correspondence between the segment output pins (SEGxx) and the PFSEG register (PFSEGxx bits) and the existence of SEGxx pins in each product are shown in Table 4-4 Segment Output Pins in Each Product and Correspondence with PFSEG Register (PFSEG Bits).

Figure 4-9. Format of **LCD port function registers 0 to 5 (PFSEG0 to PFSEG5)**

Address: F0300H After reset: F0H R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG0	PFSEG07	PFSEG06	PFSEG05	PFSEG04	0	0	0	0

Address: F0301H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG1	PFSEG15	PFSEG14	PFSEG13	PFSEG12	PFSEG11	PFSEG10	PFSEG09	PFSEG08

Address: F0302H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG2	PFSEG23	PFSEG22	PFSEG21	PFSEG20	PFSEG19	PFSEG18	PFSEG17	PFSEG16

Address: F0303H After reset: FFH R/W

	Symbol	7	6	5	4	3	2	1	0
<R>	PFSEG3	PFSEG31 Note	PFSEG30 Note	PFSEG29 Note	PFSEG28 Note	PFSEG27	PFSEG26	PFSEG25	PFSEG24

Address: F0304H After reset: FFH R/W

	Symbol	7	6	5	4	3	2	1	0
<R>	PFSEG4	PFSEG39 Note	PFSEG38 Note	PFSEG37	PFSEG36	PFSEG35	PFSEG34	PFSEG33	PFSEG32

Address: F0305H After reset: 03H R/W

	Symbol	7	6	5	4	3	2	1	0
<R>	PFSEG5	0	0	0	0	0	0	PFSEG41 Note	PFSEG40 Note

<R> **Note** Be sure to set "1" for 80-pin products

5.3.3 Clock operation status control register (CSC)

Incorrect:

Cautions 5. Do not stop the clock selected for the CPU/peripheral hardware clock (f_{CLK}) by using the OSC register.

6. The setting of the flags of the register to stop clock oscillation (disabling the external clock input) and the condition before clock oscillation is stopped are shown in Table 5-2.

Table 5-2. Stopping the Clock

Clock	Condition Before Stopping Clock (Disabling External Clock Input)	Setting of CSC Register Flags
X1 oscillator clock External main system clock	The CPU/peripheral hardware clock is a clock other than the high-speed system clock. (CLS = 0 and MCS = 0, or CLS = 1)	MSTOP = 1
XT1 oscillator clock External subsystem clock	The CPU/peripheral hardware clock is a clock other than the subsystem clock. (CLS = 0)	XTSTOP = 1
High-speed on-chip oscillator clock	The CPU/peripheral hardware clock is a clock other than the high-speed on-chip oscillator clock. (CLS = 0 and MCS = 1, or CLS = 1)	HIOSTOP = 1

5.3.3 Clock operation status control register (CSC)

Correct:

Cautions 5. Do not stop the clock selected for the CPU/peripheral hardware clock (f_{CLK}) by using the OSC register.

<R> **6. The setting of the flags of the register to stop clock oscillation (disabling the external clock input) and the condition before clock oscillation is stopped are shown in Table 5-2. When stopping the clock, confirm the condition before stopping clock.**

Table 5-2. Stopping the Clock

Clock	Condition Before Stopping Clock (Disabling External Clock Input)	Setting of CSC Register Flags
X1 oscillator clock External main system clock	The CPU/peripheral hardware clock is a clock other than the high-speed system clock. (CLS = 0 and MCS = 0, or CLS = 1)	MSTOP = 1
XT1 oscillator clock External subsystem clock	The CPU/peripheral hardware clock is a clock other than the subsystem clock. (CLS = 0)	XTSTOP = 1
High-speed on-chip oscillator clock	The CPU/peripheral hardware clock is a clock other than the high-speed on-chip oscillator clock. (CLS = 0 and MCS = 1, or CLS = 1)	HIOSTOP = 1

5.6.5 Conditions before changing the CPU clock and processing after changing CPU clock

Incorrect:

The conditions before changing the CPU clock and processing after changing the CPU clock are shown below.

Table 5-4. Changing CPU Clock (1/2)

CPU Clock		Conditions Before Change	Processing After Change
Before Change	After Change		
High-speed on-chip oscillator clock	X1 clock	X1 oscillation is stable • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • The oscillation stabilization time has elapsed	Operating current can be reduced by stopping high-speed on-chip oscillator (HIOSTOP = 1).
	External main system clock	Inputting the external clock from the EXCLK pin is enabled • OSCSEL = 1, EXCLK = 1, MSTOP = 0	
	XT1 clock	XT1 oscillation is stable • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • The oscillation stabilization time has elapsed	
	External subsystem clock	Inputting the external clock from the EXCLKS pin is enabled • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	
X1 clock	High-speed on-chip oscillator clock	Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • The oscillation accuracy stabilization time has elapsed	X1 oscillation can be stopped (MSTOP = 1).
	External main system clock	Transition impossible	–
	XT1 clock	XT1 oscillation is stable • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • The oscillation stabilization time has elapsed	X1 oscillation can be stopped (MSTOP = 1).
	External subsystem clock	Inputting the external clock from the EXCLKS pin is enabled • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	X1 oscillation can be stopped (MSTOP = 1).
External main system clock	High-speed on-chip oscillator clock	Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • The oscillation accuracy stabilization time has elapsed	Inputting the external main system clock can be disabled (MSTOP = 1).
	X1 clock	Transition impossible	–
	XT1 clock	XT1 oscillation is stable • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • The oscillation stabilization time has elapsed	Inputting the external main system clock can be disabled (MSTOP = 1).
	External subsystem clock	Inputting the external clock from the EXCLKS pin is enabled • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	Inputting the external main system clock can be disabled (MSTOP = 1).

5.6.5 Conditions before changing the CPU clock and processing after changing CPU clock

Correct:

The conditions before changing the CPU clock and processing after changing the CPU clock are shown below.

Table 5-4. Changing CPU Clock (1/2)

	CPU Clock		Conditions Before Change	Processing After Change
	Before Change	After Change		
<R>	High-speed on-chip oscillator clock	X1 clock	X1 oscillation is stable • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • The oscillation stabilization time has elapsed	After confirming that the CPU clock has changed from the high-speed on-chip oscillator clock to the X1 clock, external main system clock, XT1 clock, or external subsystem clock, operating current can be reduced by stopping the high-speed on-chip oscillator (HIOSTOP = 1).
		External main system clock	Inputting the external clock from the EXCLK pin is enabled • OSCSEL = 1, EXCLK = 1, MSTOP = 0	
		XT1 clock	XT1 oscillation is stable • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • The oscillation stabilization time has elapsed	
		External subsystem clock	Inputting the external clock from the EXCLKS pin is enabled • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	
<R>	X1 clock	High-speed on-chip oscillator clock	Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • The oscillation accuracy stabilization time has elapsed	After confirming that the CPU clock has changed from the X1 clock to the high-speed on-chip oscillator clock, the X1 oscillation can be stopped (MSTOP = 1).
		External main system clock	Transition impossible	–
<R>		XT1 clock	XT1 oscillation is stable • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • The oscillation stabilization time has elapsed	After confirming that the CPU clock has changed from the X1 clock to the XT1 clock, the X1 oscillation can be stopped (MSTOP = 1).
<R>		External subsystem clock	Inputting the external clock from the EXCLKS pin is enabled • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	After confirming that the CPU clock has changed from the X1 clock to the external subsystem clock, the X1 oscillation can be stopped (MSTOP = 1).
<R>	External main system clock	High-speed on-chip oscillator clock	Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • The oscillation accuracy stabilization time has elapsed	After confirming that the CPU clock has changed from the external main system clock to the high-speed on-chip oscillator clock, inputting the external main system clock can be disabled (MSTOP = 1).
		X1 clock	Transition impossible	–
<R>		XT1 clock	XT1 oscillation is stable • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • The oscillation stabilization time has elapsed	After confirming that the CPU clock has changed from the external main system clock to the XT1 clock, inputting the external main system clock can be disabled (MSTOP = 1).
<R>		External subsystem clock	Inputting the external clock from the EXCLKS pin is enabled • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	After confirming that the CPU clock has changed from the external main system clock to the external subsystem clock, inputting the external main system clock can be disabled (MSTOP = 1).

Incorrect:

Table 5-4. Changing CPU Clock (2/2)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
XT1 clock	High-speed on-chip oscillator clock	The high-speed on-chip oscillator is oscillating and the high-speed on-chip oscillator clock is selected as the main system clock • HIOSTOP = 0, MCS = 0	XT1 oscillation can be stopped (XTSTOP = 1)
	X1 clock	X1 oscillation is stable and the high-speed system clock is selected as the main system clock • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • The oscillation stabilization time has elapsed • MCS = 1	
	External main system clock	Inputting the external clock from the EXCLK pin is enabled and the high-speed system clock is selected as the main system clock • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	
	External subsystem clock	Transition impossible	–
External subsystem clock	High-speed on-chip oscillator clock	The high-speed on-chip oscillator is oscillating and the high-speed on-chip oscillator clock is selected as the main system clock • HIOSTOP = 0, MCS = 0	Inputting external subsystem clock can be disabled (XTSTOP = 1).
	X1 clock	X1 oscillation is stable and the high-speed system clock is selected as the main system clock • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • The oscillation stabilization time has elapsed • MCS = 1	
	External main system clock	Inputting the external clock from the EXCLK pin is enabled and the high-speed system clock is selected as the main system clock • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	
	XT1 clock	Transition impossible	_

Correct:

Table 5-4. Changing CPU Clock (2/2)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
<R> XT1 clock	High-speed on-chip oscillator clock	The high-speed on-chip oscillator is oscillating and the high-speed on-chip oscillator clock is selected as the main system clock • HIOSTOP = 0, MCS = 0	After confirming that the CPU clock has changed from the XT1 clock to the high-speed on-chip oscillator clock, X1 clock, or external main system clock, the XT1 oscillation can be stopped (XTSTOP = 1).
	X1 clock	X1 oscillation is stable and the high-speed system clock is selected as the main system clock • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • The oscillation stabilization time has elapsed • MCS = 1	
	External main system clock	Inputting the external clock from the EXCLK pin is enabled and the high-speed system clock is selected as the main system clock • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	
	External subsystem clock	Transition impossible	–
<R> External subsystem clock	High-speed on-chip oscillator clock	The high-speed on-chip oscillator is oscillating and the high-speed on-chip oscillator clock is selected as the main system clock • HIOSTOP = 0, MCS = 0	After confirming that the CPU clock has changed from the external subsystem clock to the high-speed on-chip oscillator clock, X1 clock, or external main system clock, inputting the external subsystem clock can be disabled (XTSTOP = 1).
	X1 clock	X1 oscillation is stable and the high-speed system clock is selected as the main system clock • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • The oscillation stabilization time has elapsed • MCS = 1	
	External main system clock	Inputting the external clock from the EXCLK pin is enabled and the high-speed system clock is selected as the main system clock • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	
	XT1 clock	Transition impossible	_

5.6.7 Conditions before stopping clock oscillation

Incorrect:

The following lists the register flag settings for stopping the clock oscillation (disabling external clock input) and conditions before the clock oscillation is stopped.

Table 5-8. Conditions Before Stopping the Clock Oscillation and Flag Settings

Clock	Conditions Before Stopping Clock Oscillation (Disabling External Clock Input)	SFR Flag Settings
High-speed on-chip oscillator clock	MCS = 1 or CLS = 1 (The CPU is operating on a clock other than the high-speed on-chip oscillator clock.)	HIOSTOP = 1
X1 oscillator clock External main system clock	MCS = 0 or CLS = 1 (The CPU is operating on a clock other than the high-speed system clock.)	MSTOP = 1
XT1 oscillator clock External subsystem clock	CLS = 0 (The CPU is operating on a clock other than the subsystem clock.)	XTSTOP = 1

5.6.7 Conditions before stopping clock oscillation

Correct:

The following lists the register flag settings for stopping the clock oscillation (disabling external clock input) and conditions before the clock oscillation is stopped.

<R> When stopping the clock, confirm the condition before stopping clock.

Table 5-8. Conditions Before Stopping the Clock Oscillation and Flag Settings

Clock	Conditions Before Stopping Clock Oscillation (Disabling External Clock Input)	SFR Flag Settings
High-speed on-chip oscillator clock	MCS = 1 or CLS = 1 (The CPU is operating on a clock other than the high-speed on-chip oscillator clock.)	HIOSTOP = 1
X1 oscillator clock External main system clock	MCS = 0 or CLS = 1 (The CPU is operating on a clock other than the high-speed system clock.)	MSTOP = 1
XT1 oscillator clock External subsystem clock	CLS = 0 (The CPU is operating on a clock other than the subsystem clock.)	XTSTOP = 1

7.3.3 Timer mode register mn

Incorrect:

Figure 7-12. Format of Timer Mode Register mn (TMRmn) (1/4)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2, 4, 6)	CKS mn1	CKS mn0	0	CCS mn	MAST ERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKS mn1	CKS mn0	0	CCS mn	SPLIT mn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0, 5, 7)	CKS mn1	CKS mn0	0	CCS mn	0Note	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

CKS mn1	CKS mn0	Selection of operation clock (f_{MCK}) of channel n
0	0	Operation clock CKm0 set by timer clock select register m (TPSm)
0	1	Operation clock CKm2 set by timer clock select register m (TPSm)
1	0	Operation clock CKm1 set by timer clock select register m (TPSm)
1	1	Operation clock CKm3 set by timer clock select register m (TPSm)
Operation clock (f_{MCK}) is used by the edge detector. A count clock (f_{TCLK}) and a sampling clock are generated depending on the setting of the CCSmn bit. The operation clocks CKm2 and CKm3 can only be selected for channels 1 and 3.		

CCS mn	Selection of count clock (f_{TCLK}) of channel n
0	Operation clock (f_{MCK}) specified by the CKSmn0 and CKSmn1 bits
1	Valid edge of input signal input from the TImn pin In channel 1, Valid edge of input signal selected by TIS0
Count clock (f_{TCLK}) is used for the counter, output controller, and interrupt controller.	

Note Bit 11 is fixed at 0 of read only, write is ignored.

Cautions 1. Be sure to clear bits 13, 5, and 4 to “0”.

2. The timer array unit must be stopped (TTm = 00FFH) if the clock selected for f_{CLK} is changed (by changing the value of the system clock control register (CKC)), even if the operating clock specified by using the CKSmn0 and CKSmn1 bits (f_{MCK}) or the valid edge of the signal input from the TImn pin is selected as the count clock (f_{TCLK}).

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

7.3.3 Timer mode register mn

Correct:

Figure 7-12. Format of Timer Mode Register mn (TMRmn) (1/4)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2, 4, 6)	CKS mn1	CKS mn0	0	CCS mn	MAST ERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKS mn1	CKS mn0	0	CCS mn	SPLIT mn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0, 5, 7)	CKS mn1	CKS mn0	0	CCS mn	0Note	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

CKS mn1	CKS mn0	Selection of operation clock (f_{MCK}) of channel n
0	0	Operation clock CKm0 set by timer clock select register m (TPSm)
0	1	Operation clock CKm2 set by timer clock select register m (TPSm)
1	0	Operation clock CKm1 set by timer clock select register m (TPSm)
1	1	Operation clock CKm3 set by timer clock select register m (TPSm)
Operation clock (f_{MCK}) is used by the edge detector. A count clock (f_{TCLK}) and a sampling clock are generated depending on the setting of the CCSmn bit. The operation clocks CKm2 and CKm3 can only be selected for channels 1 and 3.		

<R>

CCS mn	Selection of count clock (f_{TCLK}) of channel n
0	Operation clock (f_{MCK}) specified by the CKSmn0 and CKSmn1 bits
1	Valid edge of input signal input from the TImn pin In channel 5, Valid edge of input signal selected by TIS0 In channel 7, Valid edge of input signal selected by ISC
Count clock (f_{TCLK}) is used for the counter, output controller, and interrupt controller.	

Note Bit 11 is fixed at 0 of read only, write is ignored.

Cautions 1. Be sure to clear bits 13, 5, and 4 to “0”.

2. The timer array unit must be stopped (TTm = 00FFH) if the clock selected for f_{CLK} is changed (by changing the value of the system clock control register (CKC)), even if the operating clock specified by using the CKSmn0 and CKSmn1 bits (f_{MCK}) or the valid edge of the signal input from the TImn pin is selected as the count clock (f_{TCLK}).

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

7.3.8 Timer input select register 0 (TIS0)

Incorrect:

The TIS0 register is used to select the channel 5 timer input.

The TIS0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-17. Format of Timer Input Select register 0 (TIS0)

Address: F0074H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	0	0	0	TIS02	TIS01	TIS00

TIS02	TIS01	TIS00	Selection of timer input used with channel 5
0	0	0	Input signal of timer input pin (TI05)
0	0	1	
0	1	0	
0	1	1	
1	0	0	Low-speed on-chip oscillator clock (f_{IL})
1	0	1	Subsystem clock (f_{SUB})
Other than above			Setting prohibited

Caution High-level width, low-level width of timer input is selected, will require more than 1/f_{MCK} +10 ns.

Therefore, when selecting f_{SUB} to f_{CLK} (CSS bit of CKS register = 1), can not TIS02 bit set to 1.

7.3.8 Timer input select register 0 (TIS0)

Correct:

The TIS0 register is used to select the channel 5 timer input.

The TIS0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-17. Format of Timer Input Select register 0 (TIS0)

Address: F0074H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	0	0	0	TIS02	TIS01	TIS00

TIS02	TIS01	TIS00	Selection of timer input used with channel 5
0	0	0	Input signal of timer input pin (TI05)
0	0	1	
0	1	0	
0	1	1	
1	0	0	Low-speed on-chip oscillator clock (f_{IL})
1	0	1	Subsystem clock (f_{SUB})
Other than above			Setting prohibited

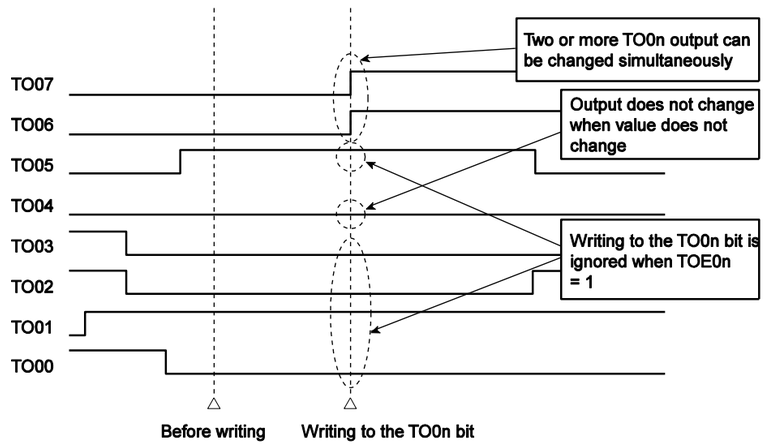
<R> **Caution High-level width, low-level width of timer input is selected, will require more than 1/f_{MCK} +10 ns.**

Therefore, when selecting f_{SUB} to f_{CLK} (CSS bit of CKC register = 1), can not TIS02 bit set to 1.

7.6.4 Collective manipulation of TOmn bit

Incorrect:

Figure 7-38. TO0n Pin Statuses by Collective Manipulation of TO0n Bit



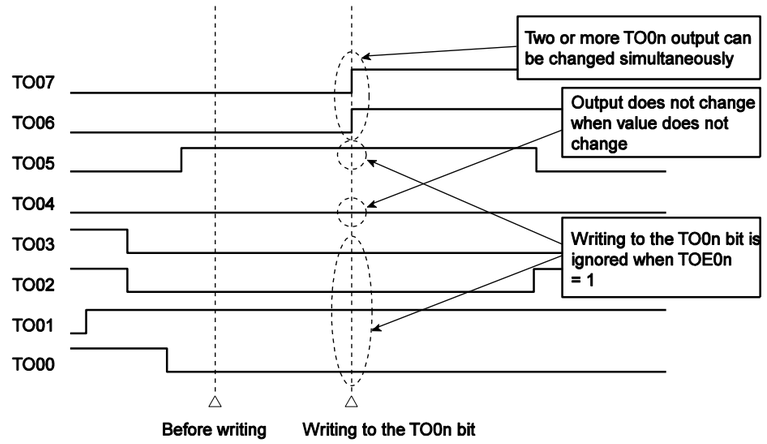
Caution While timer output is enabled (TOEmn = 1), even if the output by timer interrupt of each timer (INTTMmn) contends with writing to the TOmn bit, output is normally done to the TOmn pin.

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

7.6.4 Collective manipulation of TOmn bit

Correct:

Figure 7-38. TO0n Pin Statuses by Collective Manipulation of TO0n Bit

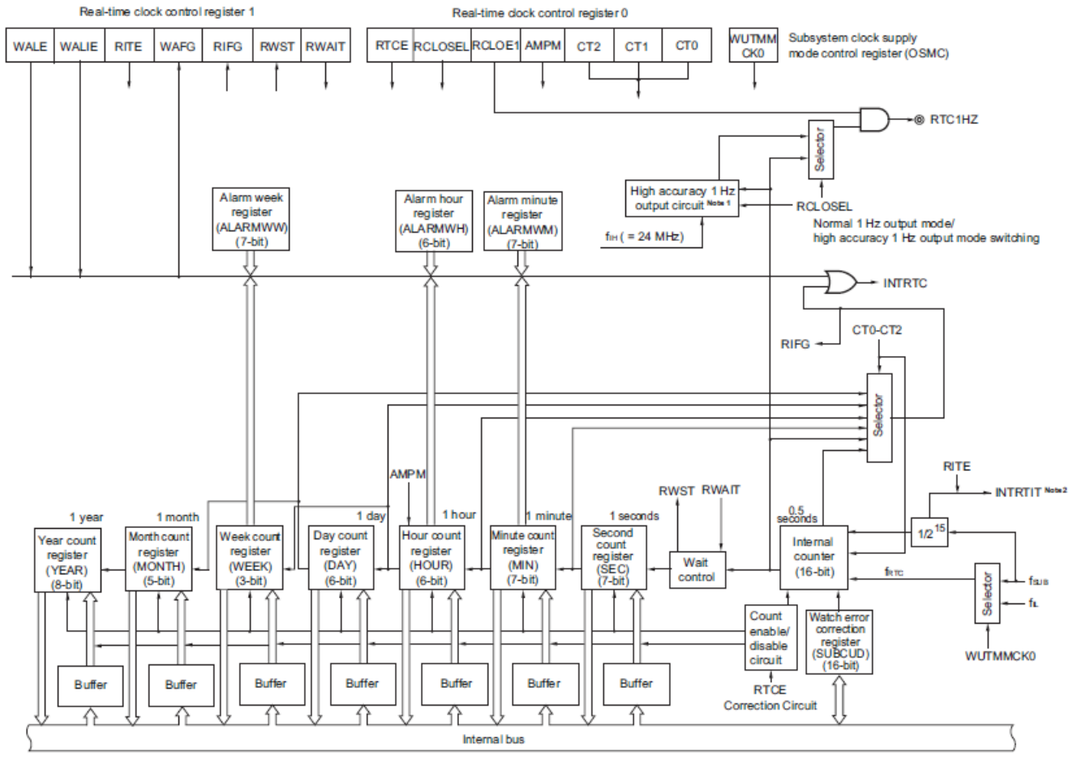


Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

8.2 Configuration of Real-time Clock 2

Incorrect:

Figure 8-1. Real-time Clock 2 Diagram



Notes 1. A high-speed on-chip oscillator (HOCO: 24 MHz) can be used for high precision 1 Hz output. HOCO must be set to ON in order to run in high precision 1 Hz output mode. To run in normal 1 Hz mode, there is no need to set HOCO to ON.

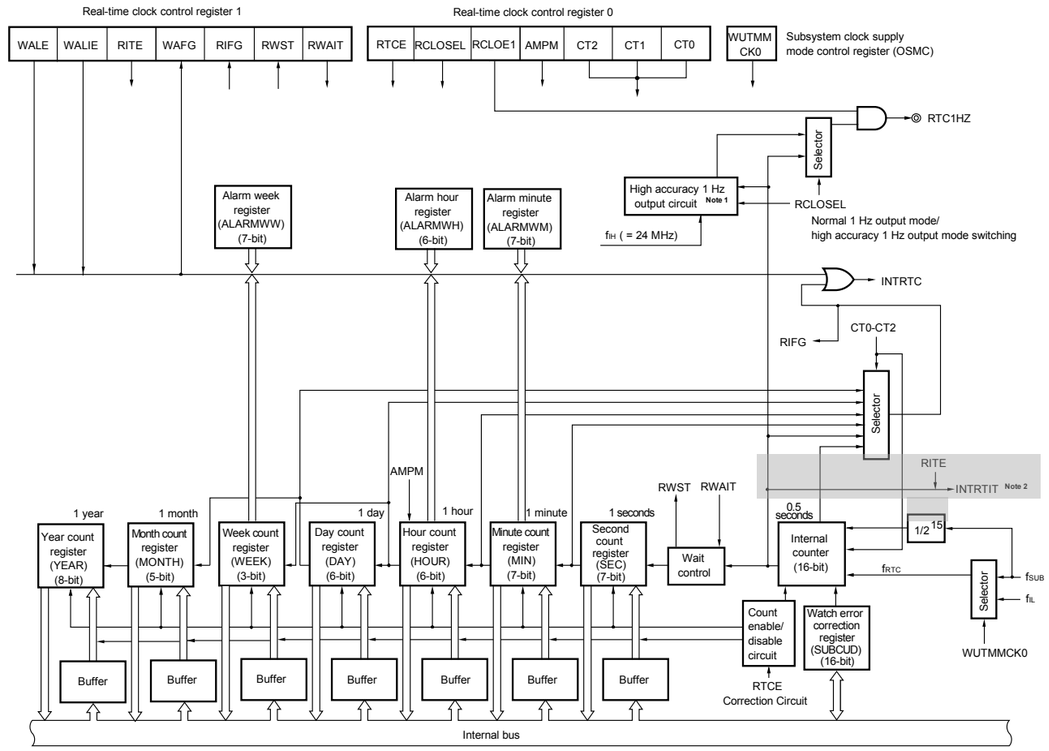
2. An interrupt that indicates the timing to get the correction value from the clock error correction register (SUBCUD). The fetch timing is 1 second (f_{SUB} base) interval.

8.2 Configuration of Real-time Clock 2

Correct:

Figure 8-1. Real-time Clock 2 Diagram

<R>



Notes 1. A high-speed on-chip oscillator (HOCO: 24 MHz) can be used for high precision 1 Hz output. HOCO must be set to ON in order to run in high precision 1 Hz output mode. To run in normal 1 Hz mode, there is no need to set HOCO to ON.

2. An interrupt that indicates the timing to get the correction value from the clock error correction register (SUBCUD). The fetch timing is 1 second (f_{SUB} base) interval.

8.3.6 Real-time clock control register 1 (RTCC1)

Incorrect:

Figure 8-7. Format of Real-time Clock Control Register 1 (RTCC1) (3/3)

Address: FFF9EH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	<1>	<0>
RTCC1	WALE	WALIE	RITE	WAFG	RIFG	0	RWST	RWAIT

RWST	Wait status flag of real-time clock 2
0	Counter is operating.
1	Mode to read or write counter value.
This status flag indicates whether the setting of the RWAIT bit is valid. Before reading or writing the counter value, confirm that the value of this flag is 1. Even if the RWAIT bit is set to 0, the RWST bit is not set to 0 while writing to the counter. After writing is completed, the RWST bit is set to 0.	

RWAIT	Wait control of real-time clock 2
0	Sets counter operation.
1	Stops SEC to YEAR counters. Mode to read or write counter value.
This bit controls the operation of the counter. Be sure to write “1” to it to read or write the counter value. As the counter (16-bit) is continuing to run, complete reading or writing within one second and turn back to 0. When RWAIT = 1, it takes up to 1 clock of f_{RTC} until the counter value can be read or written (RWST = 1). When the internal counter (16-bit) overflowed while RWAIT = 1, it keeps the event of overflow until RWAIT = 0, then counts up. However, when it wrote a value to second count register, it will not keep the overflow event.	

Caution If writing is performed to RTCC1 with a 1-bit manipulation instruction, the RIFG and WAFG flags may be cleared. Therefore, to perform writing to RTCC1, be sure to use an 8-bit manipulation instruction.

To prevent the RIFG and WAFG flags from being cleared during writing, set 1 (writing disabled) to the corresponding bit. If the RIFG and WAFG flags are not used and the value may be changed, RTCC1 may be written by using a 1-bit manipulation instruction.

Remarks 1. Constant-period interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the constant-period interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

2. The internal counter (16 bits) is cleared when the second count register (SEC) is written.

8.3.6 Real-time clock control register 1 (RTCC1)

Correct:

Figure 8-7. Format of Real-time Clock Control Register 1 (RTCC1) (3/3)

Address: FFF9EH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	<1>	<0>
RTCC1	WALE	WALIE	RITE	WAFG	RIFG	0	RWST	RWAIT

RWST	Wait status flag of real-time clock 2
0	Counter is operating.
1	Mode to read or write counter value.
This status flag indicates whether the setting of the RWAIT bit is valid. Before reading or writing the counter value, confirm that the value of this flag is 1. Even if the RWAIT bit is set to 0, the RWST bit is not set to 0 while writing to the counter. After writing is completed, the RWST bit is set to 0.	

<R>

RWAIT	Wait control of real-time clock 2
0	Sets counter operation.
1	Stops SEC to YEAR counters. Mode to read or write counter value.
This bit controls the operation of the counter. Be sure to write “1” to it to read or write the counter value. As the counter (16-bit) is continuing to run, complete reading or writing within one second and turn back to 0. When RWAIT = 1, it takes up to 1 clock of f_{RTC} until the counter value can be read or written (RWST = 1)Notes1, 2. When the internal counter (16-bit) overflowed while RWAIT = 1, it keeps the event of overflow until RWAIT = 0, then counts up. However, when it wrote a value to second count register, it will not keep the overflow event.	

<R> **Notes 1.** When the RWAIT bit is set to 1 within one cycle of f_{RTC} clock after setting the RTCE bit to 1, the RWST bit being set to 1 may take up to two cycles of the operating clock (f_{RTC}).

<R> **2.** When the RWAIT bit is set to 1 within one cycle of f_{RTC} clock after release from the standby mode (HALT mode, STOP mode, or SNOOZE mode), the RWST bit being set to 1 may take up to two cycles of the operating clock (f_{RTC}).

Caution If writing is performed to RTCC1 with a 1-bit manipulation instruction, the RIFG and WAFG flags may be cleared. Therefore, to perform writing to RTCC1, be sure to use an 8-bit manipulation instruction.

To prevent the RIFG and WAFG flags from being cleared during writing, set 1 (writing disabled) to the corresponding bit. If the RIFG and WAFG flags are not used and the value may be changed, RTCC1 may be written by using a 1-bit manipulation instruction.

Remarks 1. Constant-period interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the constant-period interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

2. The internal counter (16 bits) is cleared when the second count register (SEC) is written.

12.5 Cautions of Clock Output/Buzzer Output Controller

Incorrect:

12.5 Cautions of Clock Output/Buzzer Output Controller

When the main system clock is selected for the PCLBUZn output (CSEL = 0), if HALT or STOP mode is entered within 1.5 clock cycles output from the PCLBUZn pin after the output is disabled (PCLOEn = 0), the PCLBUZn output width becomes shorter.

12.5 Cautions of Clock Output/Buzzer Output Controller

Correct:

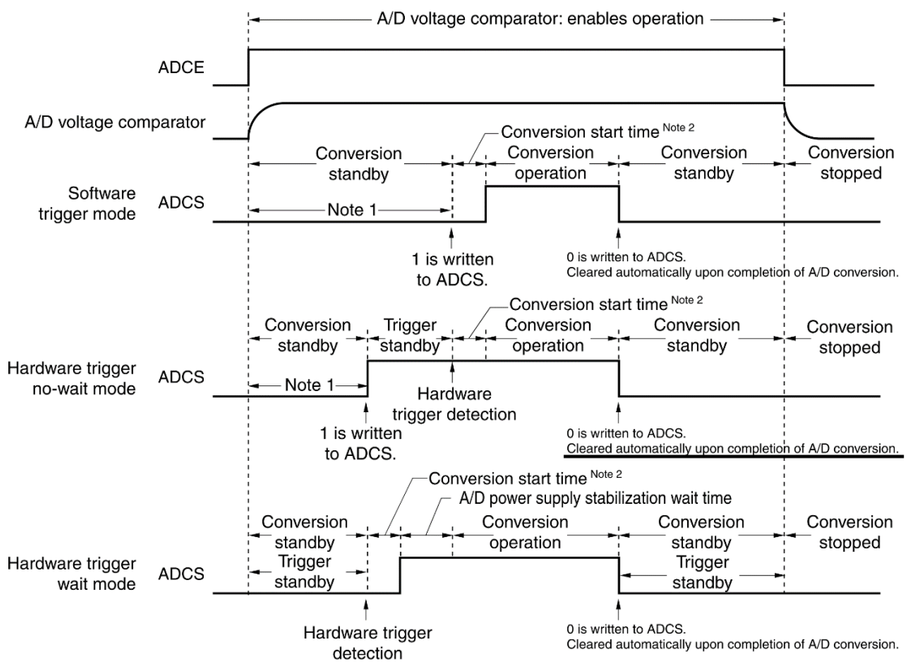
12.5 Cautions of Clock Output/Buzzer Output Controller

<R> When the main system clock is selected for the PCLBUZn output (CSEL = 0), if STOP mode is entered within 1.5 clock cycles output from the PCLBUZn pin after the output is disabled (PCLOEn = 0), the PCLBUZn output width becomes shorter.

14.3.2 A/D converter mode register 0 (ADM0)

Incorrect:

Figure 14-4. Timing Chart When A/D Voltage Comparator Is Used

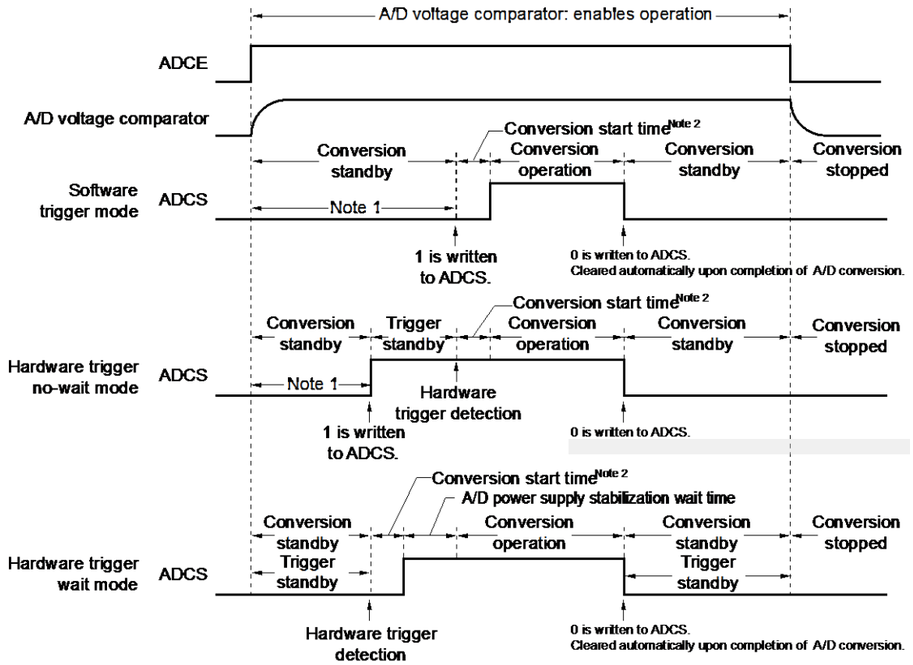


14.3.2 A/D converter mode register 0 (ADM0)

Correct:

Figure 14-4. Timing Chart When A/D Voltage Comparator Is Used

<R>

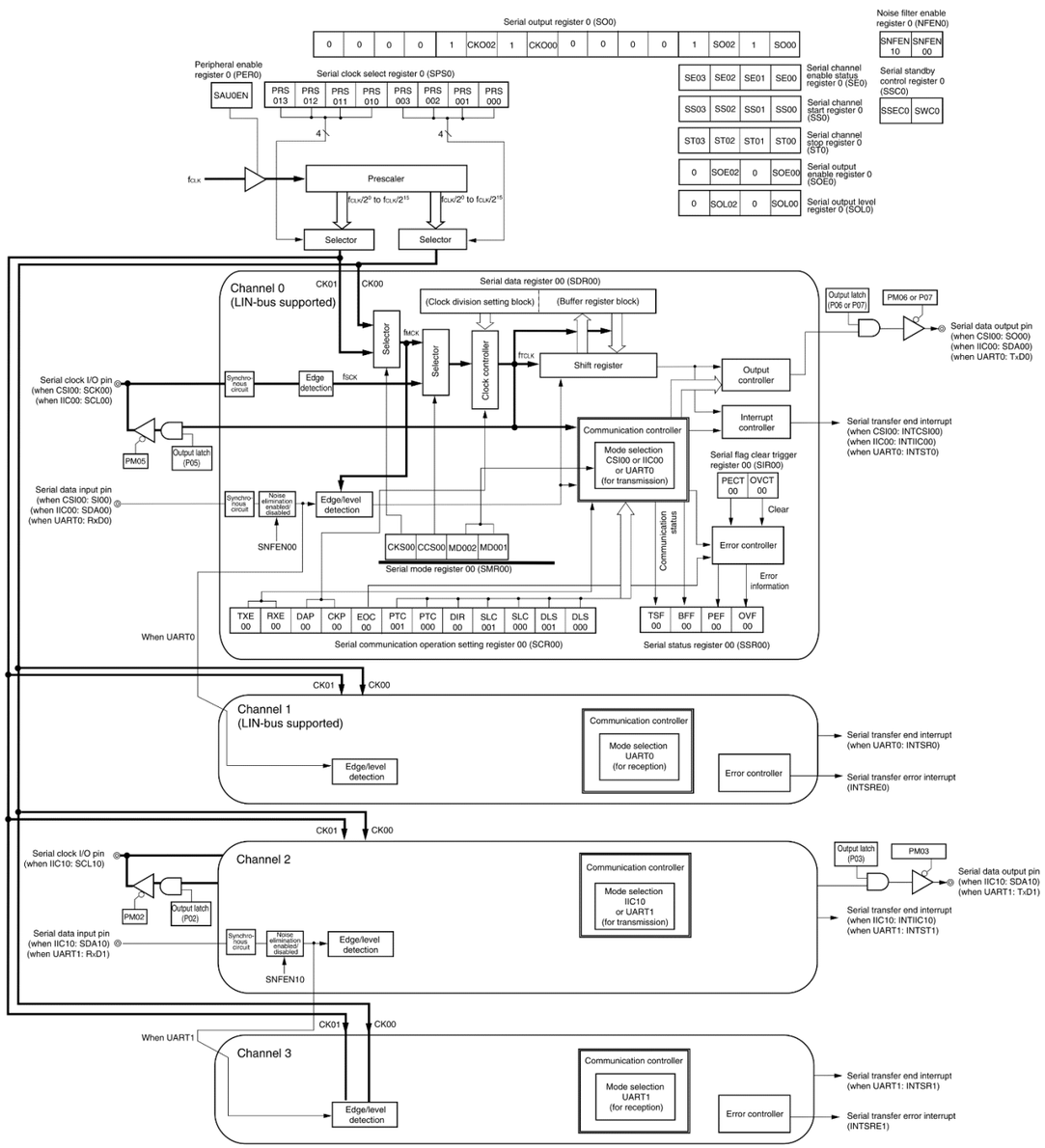


18.2 Configuration of Serial Array Unit

Incorrect:

Figure 18-1 shows the block diagram of the serial array unit 0.

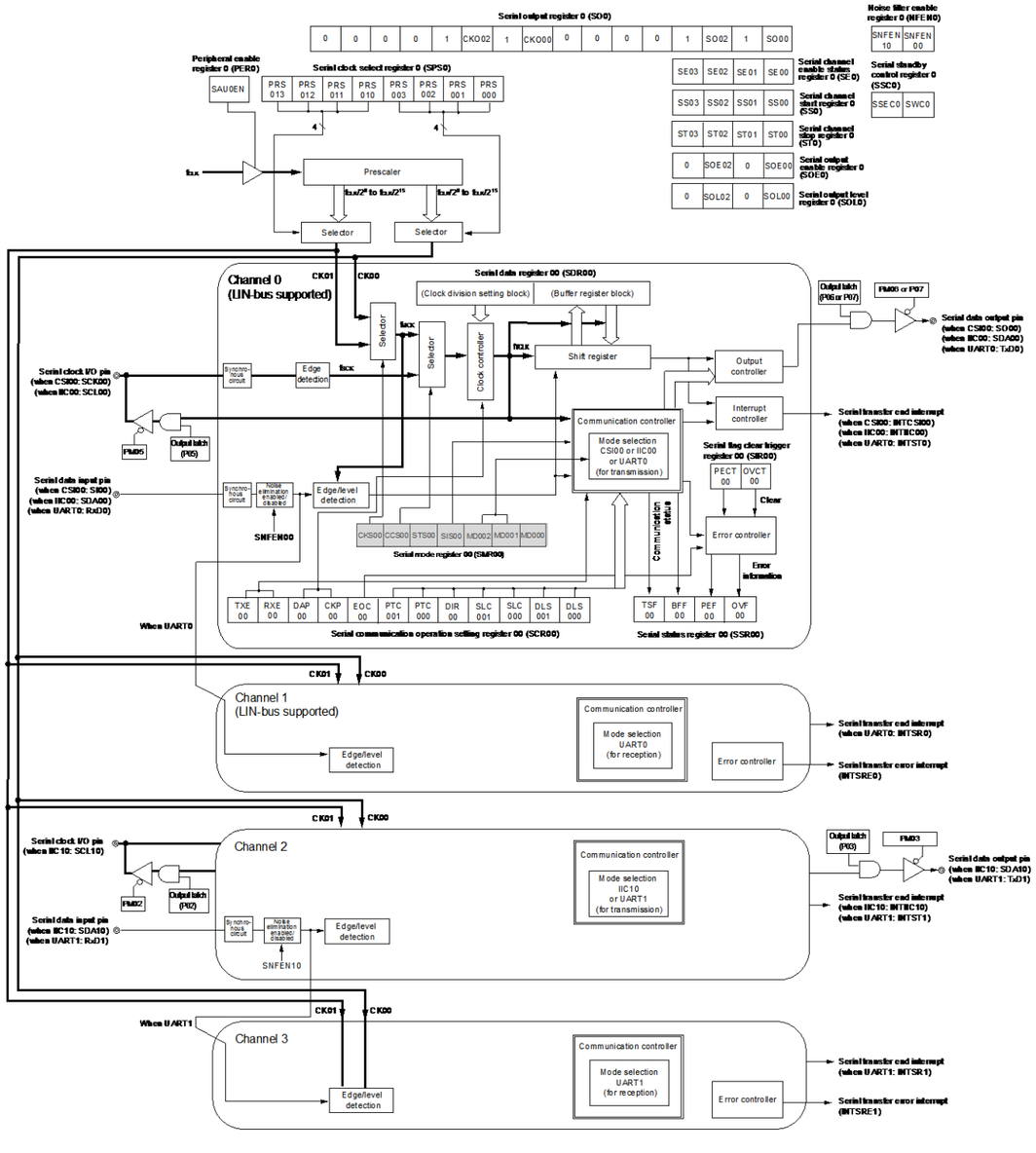
Figure 18-1. Block Diagram of Serial Array Unit 0



Correct:

Figure 18-1 shows the block diagram of the serial array unit 0.

Figure 18-1. Block Diagram of Serial Array Unit 0

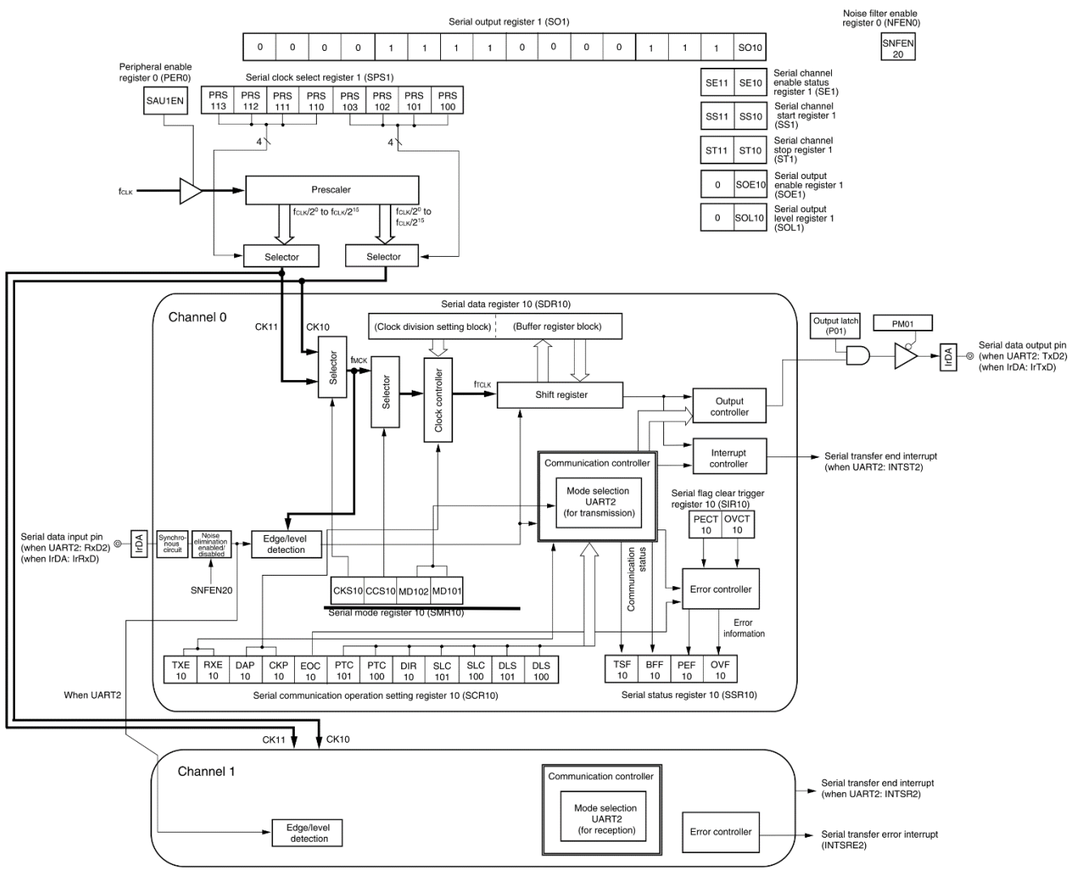


18.2 Configuration of Serial Array Unit

Incorrect:

Figure 18-2 shows the block diagram of the serial array unit 1.

Figure 18-2. Block Diagram of Serial Array Unit 1

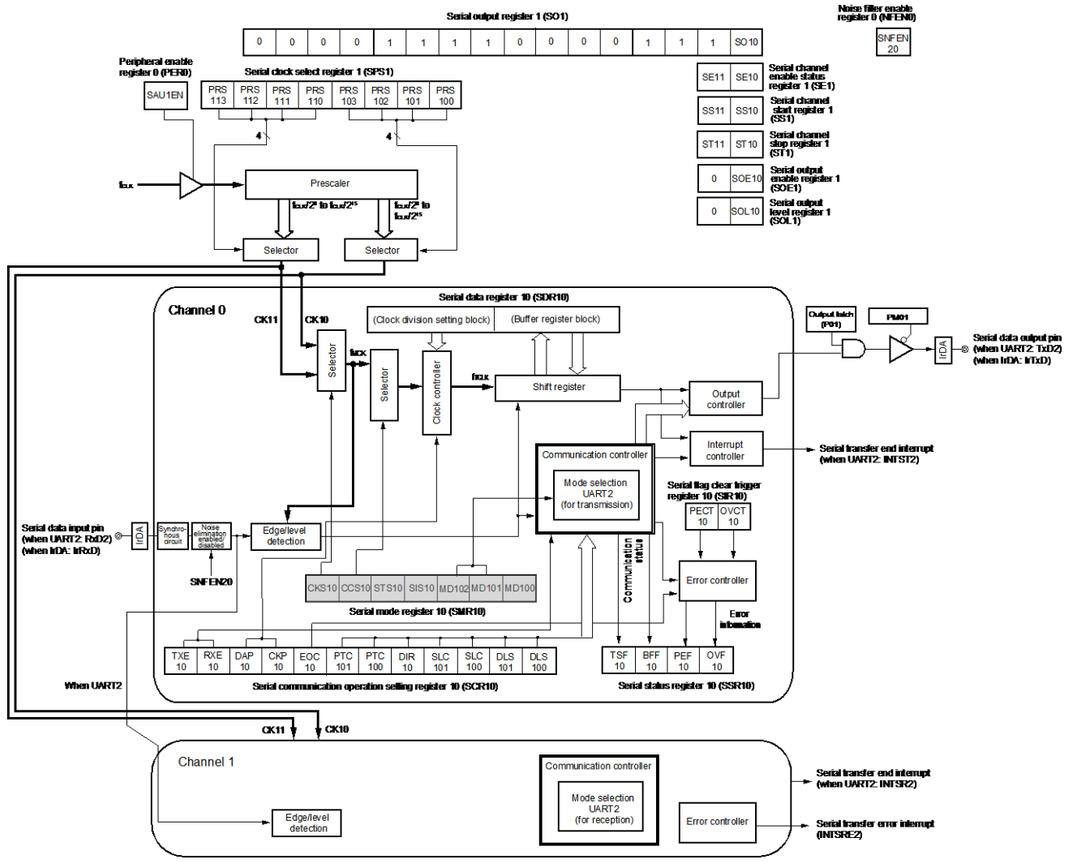


18.2 Configuration of Serial Array Unit

Correct:

Figure 18-2 shows the block diagram of the serial array unit 1.

Figure 18-2. Block Diagram of Serial Array Unit 1



18.3.5 Serial data register mn (SDRmn)

Incorrect:

The SDRmn register is the transmit/receive data register (16 bits) of channel n. Bits 8 to 0 (lower 9 bits) of SDR00 and SDR01 or bits 7 to 0 (lower 8 bits) of SDR02, SDR03, SDR10 and SDR11 function as a transmit/receive buffer register, and bits 15 to 9 (higher 7 bits) are used as a register that sets the division ratio of the operation clock (f_{MCK}).

If the CCSmn bit of serial mode register mn (SMRmn) is cleared to 0, the clock set by dividing the operating clock by bits 15 to 9 (higher 7 bits) of the SDRmn register is used as the transfer clock.

If the CCSmn bit of serial mode register mn (SMRmn) is set to 1, set bits 15 to 9 (upper 7 bits) of SDR00 to 0000000B. The input clock f_{SCK} (slave transfer in CSI mode) from the SCKp pin is used as the transfer clock.

The lower 8/9 bits of the SDRmn register function as a transmit/receive buffer register. During reception, the parallel data converted by the shift register is stored in the lower 8/9 bits, and during transmission, the data to be transmitted to the shift register is set to the lower 8/9 bits.

The SDRmn register can be read or written in 16-bit units.

However, the higher 7 bits can be written or read only when the operation is stopped (SEmn = 0). During operation (SEmn = 1), a value is written only to the lower 8/9 bits of the SDRmn register. When the SDRmn register is read during operation, the higher 7 bits are always read as 0.

Reset signal generation clears the SDRmn register to 0000H.

18.3.5 Serial data register mn (SDRmn)

Correct:

<R> The SDRmn register is the transmit/receive data register (16 bits) of channel n. Bits 8 to 0 (lower 9 bits) of SDR00, SDR01, SDR10 and SDR11 or bits 7 to 0 (lower 8 bits) of SDR02 and SDR03 function as a transmit/receive buffer register, and bits 15 to 9 (higher 7 bits) are used as a register that sets the division ratio of the operation clock (f_{MCK}).

If the CCSmn bit of serial mode register mn (SMRmn) is cleared to 0, the clock set by dividing the operating clock by bits 15 to 9 (higher 7 bits) of the SDRmn register is used as the transfer clock.

If the CCSmn bit of serial mode register mn (SMRmn) is set to 1, set bits 15 to 9 (upper 7 bits) of SDR00 to 0000000B. The input clock f_{SCK} (slave transfer in CSI mode) from the SCKp pin is used as the transfer clock.

The lower 8/9 bits of the SDRmn register function as a transmit/receive buffer register. During reception, the parallel data converted by the shift register is stored in the lower 8/9 bits, and during transmission, the data to be transmitted to the shift register is set to the lower 8/9 bits.

The SDRmn register can be read or written in 16-bit units.

However, the higher 7 bits can be written or read only when the operation is stopped (SEmn = 0). During operation (SEmn = 1), a value is written only to the lower 8/9 bits of the SDRmn register. When the SDRmn register is read during operation, the higher 7 bits are always read as 0.

Reset signal generation clears the SDRmn register to 0000H.

18.3.12 Serial output register m (SOm)

Incorrect:

The SOm register is a buffer register for serial output of each channel.

The value of the SOmn bit of this register is output from the serial data output pin of channel n.

The value of the CKOmn bit of this register is output from the serial clock output pin of channel n.

The SOmn bit of this register can be rewritten by software only when serial output is disabled (SOEmn = 0). When serial output is enabled (SOEmn = 1), rewriting by software is ignored, and the value of the register can be changed only by a serial communication operation.

The CKOmn bit of this register can be rewritten by software only when the channel operation is stopped (SEmn = 0). While channel operation is enabled (SEmn = 1), rewriting by software is ignored, and the value of the CKOmn bit can be changed only by a serial communication operation.

To use the pin for serial interface as a port function pin, set the corresponding CKOmn and SOmn bits to “1”.

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

Reset signal generation clears the SOm register to 0F0FH.

18.3.12 Serial output register m (SOm)

Correct:

The SOm register is a buffer register for serial output of each channel.

The value of the SOmn bit of this register is output from the serial data output pin of channel n.

The value of the CKOmn bit of this register is output from the serial clock output pin of channel n.

The SOmn bit of this register can be rewritten by software only when serial output is disabled (SOEmn = 0). When serial output is enabled (SOEmn = 1), rewriting by software is ignored, and the value of the register can be changed only by a serial communication operation.

The CKOmn bit of this register can be rewritten by software only when the channel operation is stopped (SEmn = 0). While channel operation is enabled (SEmn = 1), rewriting by software is ignored, and the value of the CKOmn bit can be changed only by a serial communication operation.

To use the pin for serial interface as a port function pin, set the corresponding CKOmn and SOmn bits to “1”.

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

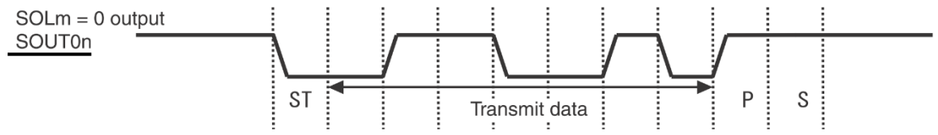
<R> Reset signal generation clears the SO0 register to 0F0FH, the SO1 register to 0303H.

18.3.13 Serial output level register m (SOLm)

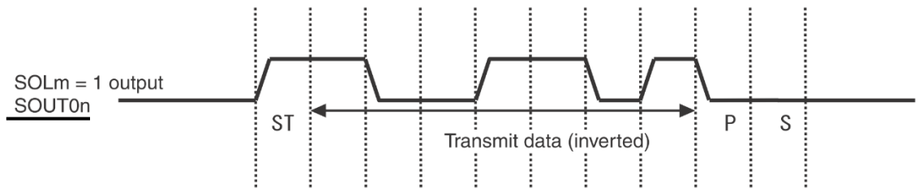
Incorrect:

Figure 18-18. Examples of Reverse Transmit Data

(1) Non-reverse Output (SOLmn = 0)



(2) Reverse Output (SOLmn = 1)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00 to 03, 10, 11

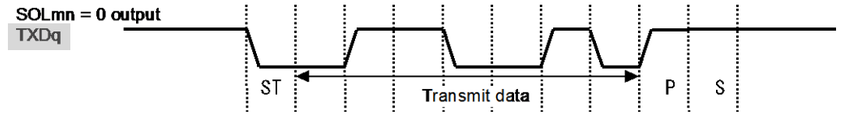
18.3.13 Serial output level register m (SOLm)

Correct:

Figure 18-18. Examples of Reverse Transmit Data

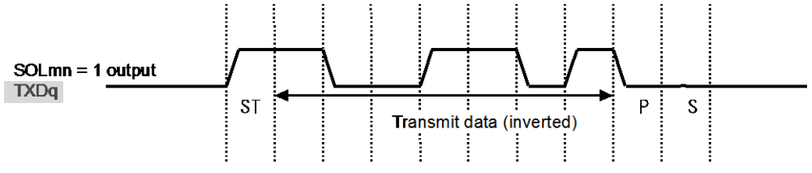
(1) Non-reverse Output (SOLmn = 0)

<R>



(2) Reverse Output (SOLmn = 1)

<R>



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00 to 03, 10, 11

18.5.7 SNOOZE mode function

Incorrect:

SNOOZE mode makes CSI operate reception by SCKp pin input detection while the STOP mode. Normally CSI stops communication in the STOP mode. But, using the SNOOZE mode makes reception CSI operate unless the CPU operation by detecting SCKp pin input.

When using the CSI in SNOOZE mode, make the following setting before switching to the STOP mode (see **Figure 18-72 Flowchart of SNOOZE Mode Operation (Once Startup)** and **Figure 18-74 Flowchart of SNOOZE Mode Operation (Continuous Startup)**).

- When using the SNOOZE mode function, set the SWCm bit of serial standby control register m (SSCm) to 1 just before switching to the STOP mode. After the initial setting has been completed, set the SSm1 bit of serial channel start register m (SSm) to 1.

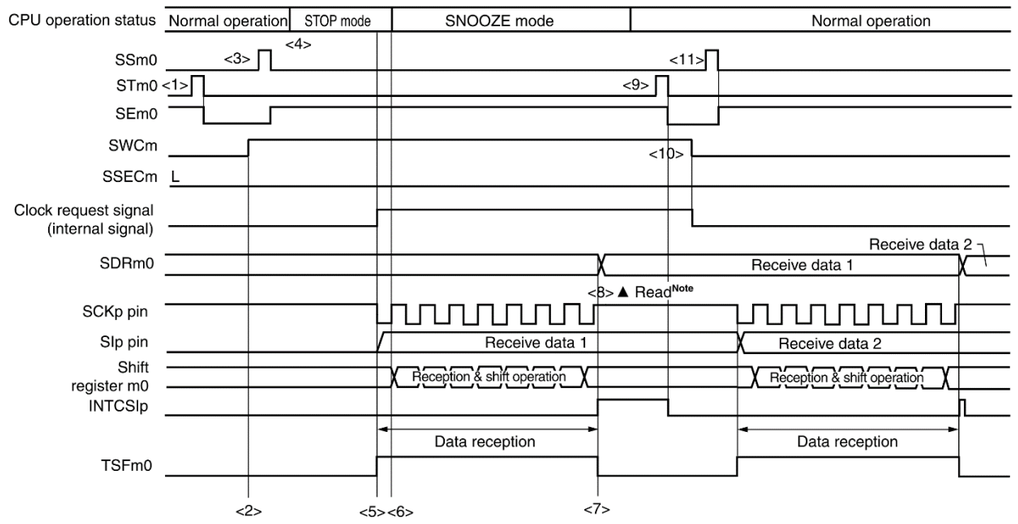
After a transition to the STOP mode, the CSI starts reception operations upon detection of an edge of the SCKp pin.

Cautions 1. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected for f_{CLK}.

2. The maximum transfer rate when using CSIp in the SNOOZE mode is 1 Mbps.

(1) SNOOZE mode operation (once startup)

Figure 18-71. Timing Chart of SNOOZE Mode Operation (Once Startup) (Type 1: DAPm0 = 0, CKPm0 = 0)



Note Only read received data while SWCm = 1 and before the next edge of the SCKp pin input is detected.

Cautions 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (clear the SEm0 bit, and stop the operation).

And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

2. When SWCm = 1, the BFFm1 and OVFm1 flags will not change.

18.5.7 SNOOZE mode function

Correct:

SNOOZE mode makes CSI operate reception by SCKp pin input detection while the STOP mode. Normally CSI stops communication in the STOP mode. But, using the SNOOZE mode makes reception CSI operate unless the CPU operation by detecting SCKp pin input.

When using the CSI in SNOOZE mode, make the following setting before switching to the STOP mode (see **Figure 18-72 Flowchart of SNOOZE Mode Operation (Once Startup)** and **Figure 18-74 Flowchart of SNOOZE Mode Operation (Continuous Startup)**).

- When using the SNOOZE mode function, set the SWCm bit of serial standby control register m (SSCm) to 1 just
<R> before switching to the STOP mode. After the initial setting has been completed, set the SSm0 bit of serial channel start register m (SSm) to 1.
<R> • The CPU shifts to the SNOOZE mode on detecting the valid edge of the SCKp signal following a transition to the STOP mode. A CSIp starts reception on detecting input of the serial clock on the SCKp pin.

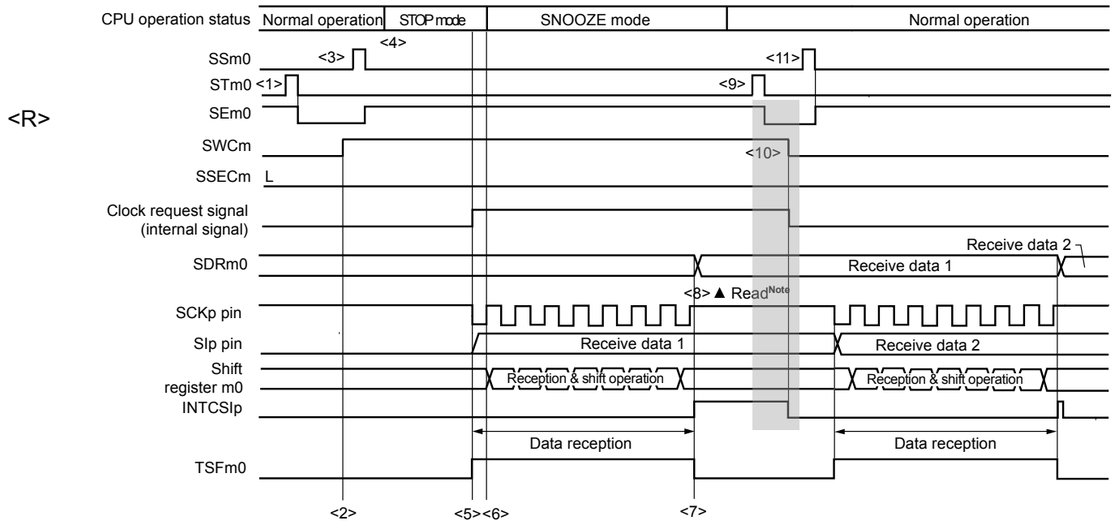
Cautions 1. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected for f_{CLK}.

2. The maximum transfer rate when using CSIp in the SNOOZE mode is 1 Mbps.

(1) SNOOZE mode operation (once startup)

Figure 18-71. Timing Chart of SNOOZE Mode Operation (Once Startup) (Type 1: DAPm0 = 0, CKPm0 = 0)

<R>



<R> **Note** Only read received data while SWCm = 1 and before the next valid edge of the SCKp pin input is detected.

Cautions 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (clear the SEm0 bit, and stop the operation).

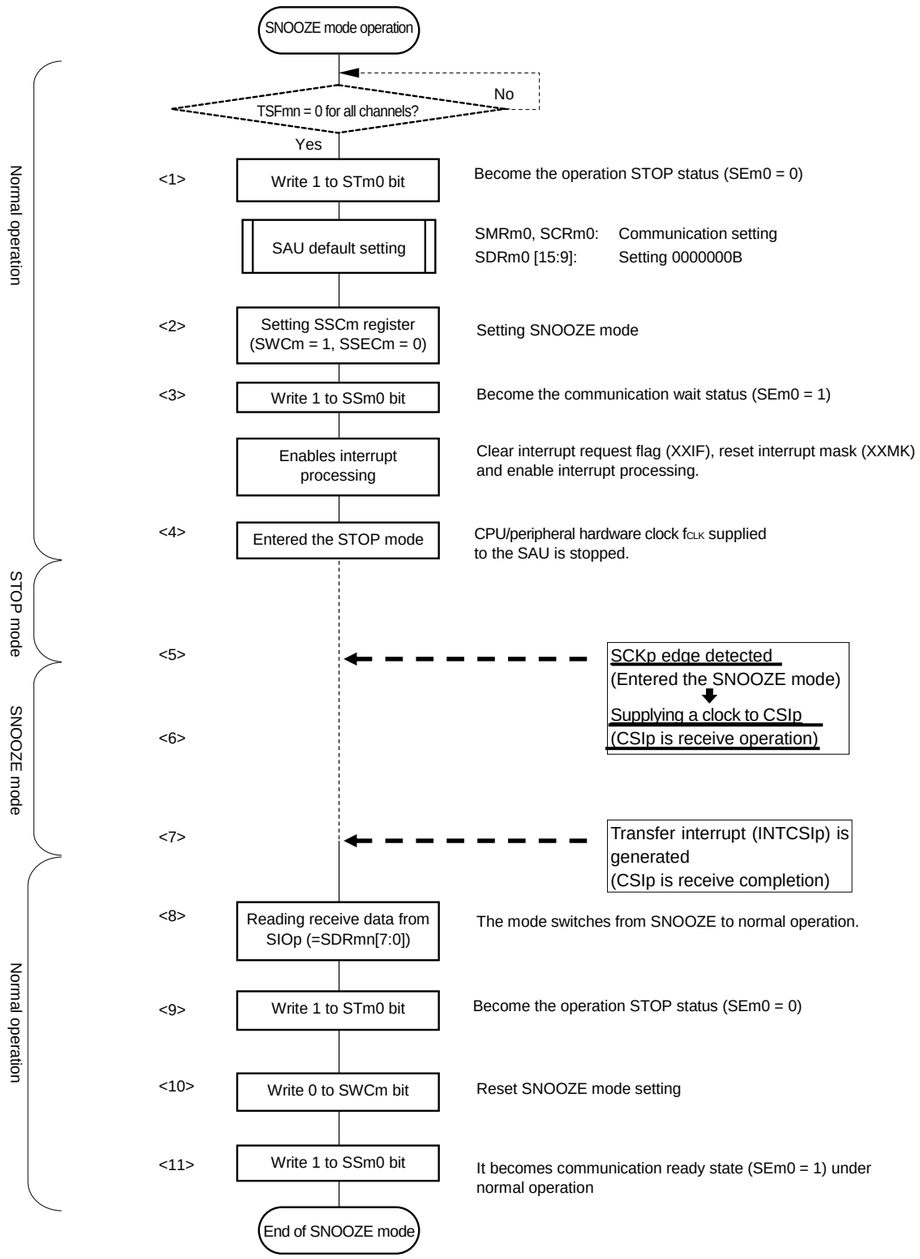
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

2. When SWCm = 1, the BFFm1 and OVFm1 flags will not change.

18.5.7 SNOOZE mode function

Incorrect:

Figure 18-72. Flowchart of SNOOZE Mode Operation (Once Startup)



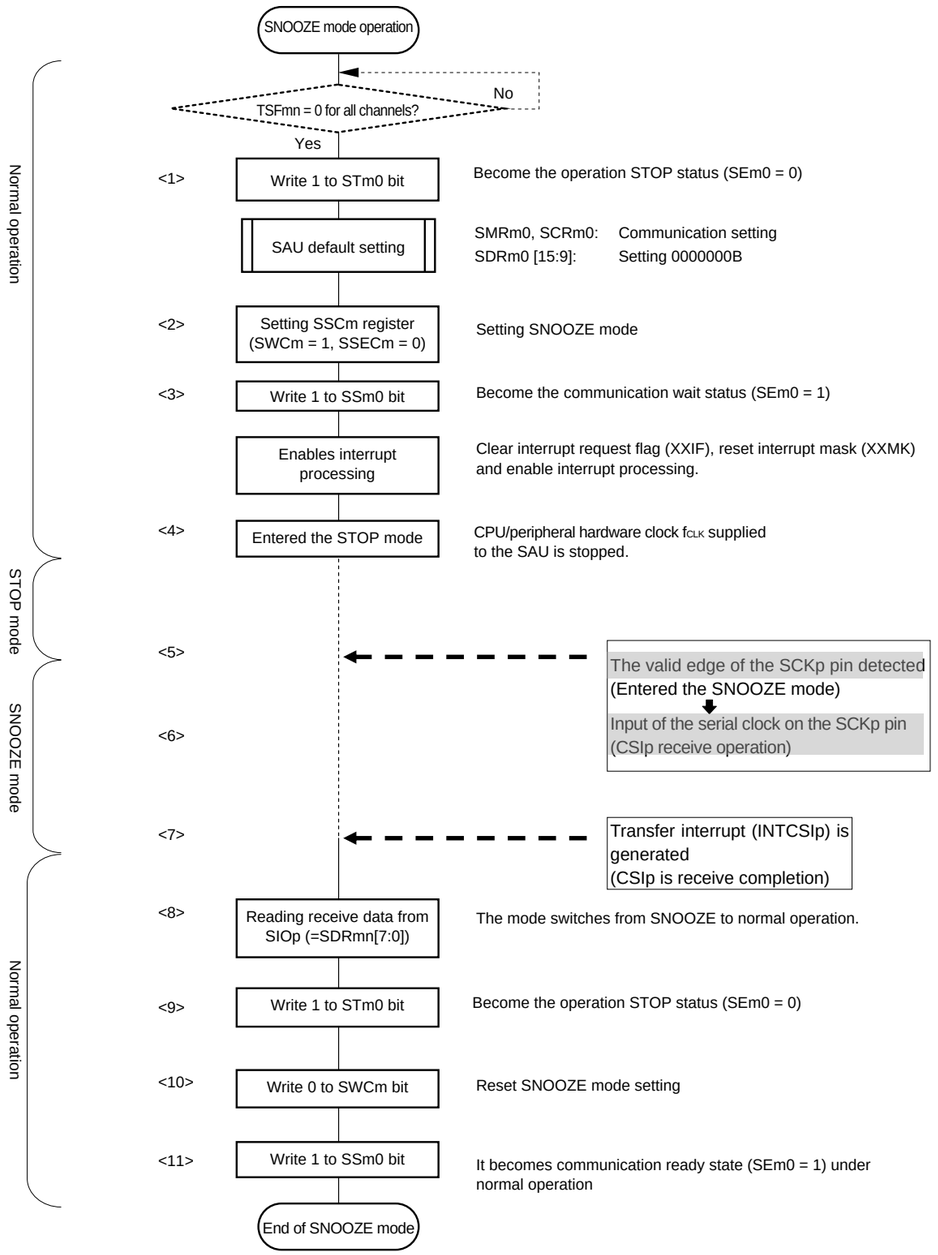
Remarks 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 18-71 Timing Chart of SNOOZE Mode Operation (Once Startup)**.

2. m = 0; p = 00

18.5.7 SNOOZE mode function

Correct:

Figure 18-72. Flowchart of SNOOZE Mode Operation (Once Startup)



Remarks 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 18-71 Timing Chart of SNOOZE Mode Operation (Once Startup)**.

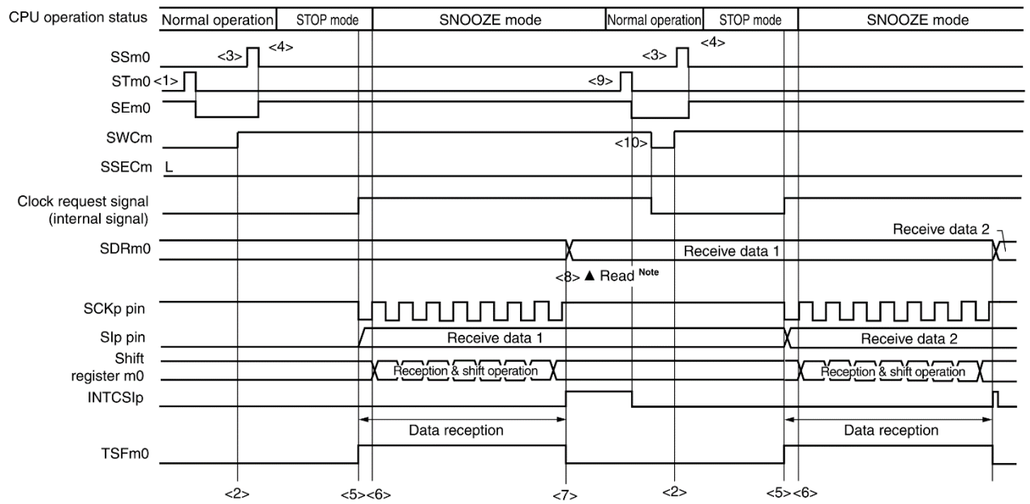
2. m = 0; p = 00

18. 5. 7 SNOOZE モード機能

Incorrect:

(2) SNOOZE mode operation (continuous startup)

Figure 18-73. Timing Chart of SNOOZE Mode Operation (Continuous Startup) (Type 1: DAPm0 = 0, CKPm0 = 0)



Note Only read received data while SWCm = 1 and before the next edge of the SCKp pin input is detected.

Cautions 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (clear the SEm0 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE release).
2. When SWCm = 1, the BFFm1 and OVFm1 flags will not change.

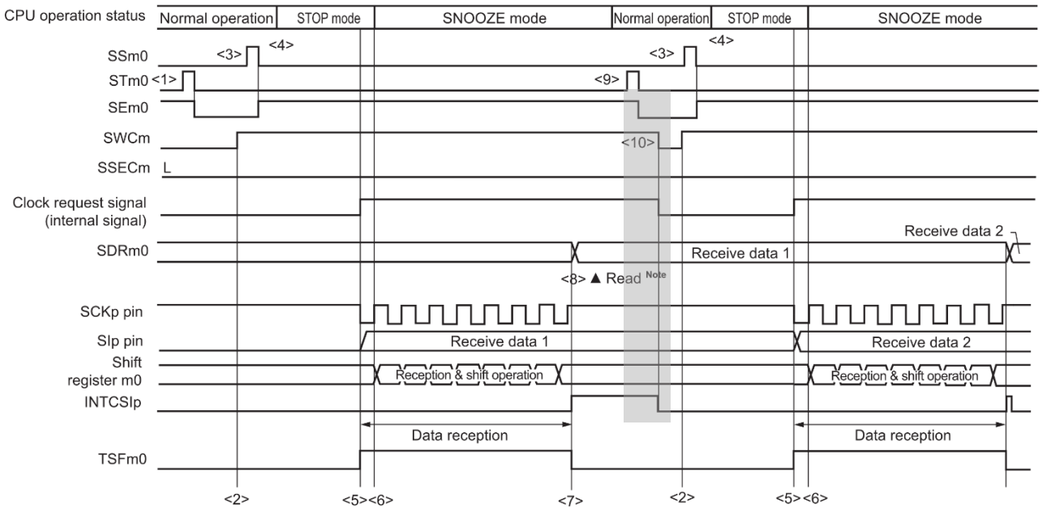
Remarks 1. <1> to <10> in the figure correspond to <1> to <10> in **Figure 18-74 Flowchart of SNOOZE Mode Operation (Continuous Startup)**.
2. m = 0; p = 00

Correct:

(2) SNOOZE mode operation (continuous startup)

Figure 18-73. Timing Chart of SNOOZE Mode Operation (Continuous Startup) (Type 1: DAPm0 = 0, CKPm0 = 0)

<R>



<R> **Note** Only read received data while SWCm = 1 and before the next valid edge of the SCKp pin input is detected.

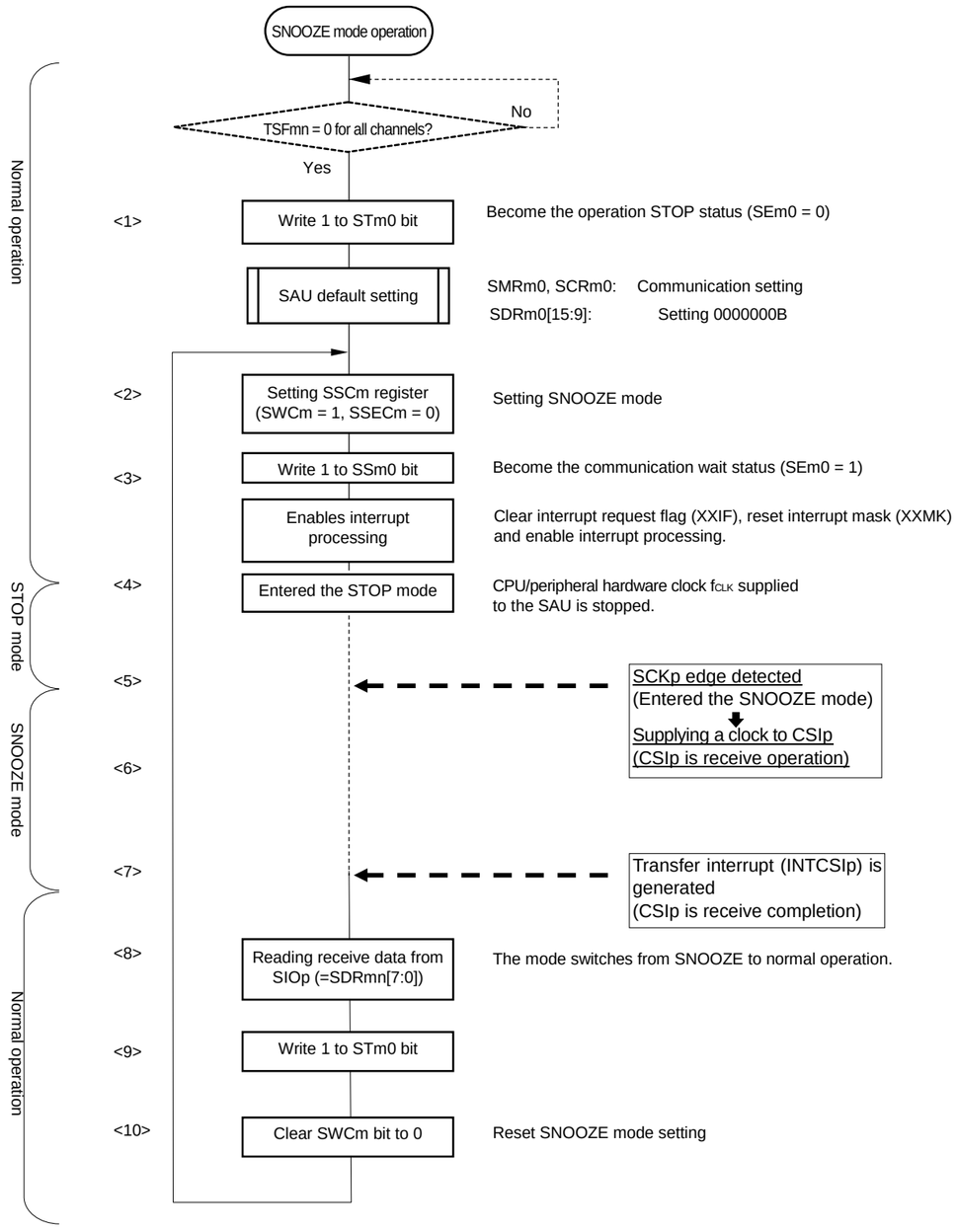
Cautions 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (clear the SEm0 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE release).
2. When SWCm = 1, the BFFm1 and OVFm1 flags will not change.

Remarks 1. <1> to <10> in the figure correspond to <1> to <10> in **Figure 18-74 Flowchart of SNOOZE Mode Operation (Continuous Startup)**.
2. m = 0; p = 00

18. 5. 7 SNOOZE モード機能

Incorrect:

Figure 18-74. Flowchart of SNOOZE Mode Operation (Continuous Startup)

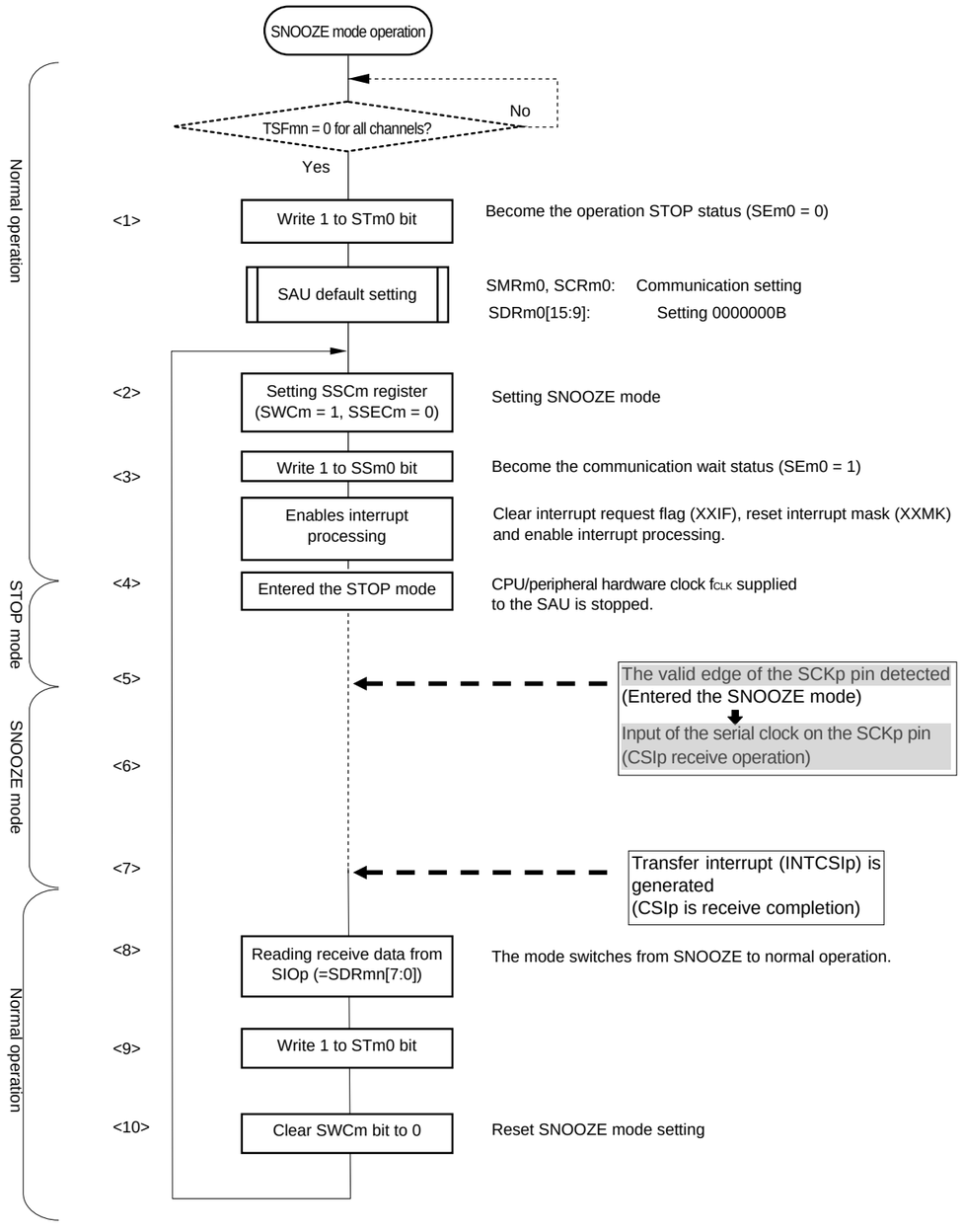


Remarks 1. <1> to <10> in the figure correspond to <1> to <10> in **Figure 18-73 Timing Chart of SNOOZE Mode Operation (Continuous Startup)**.

2. m = 0; p = 00

Correct:

Figure 18-74. Flowchart of SNOOZE Mode Operation (Continuous Startup)



<R>

Remarks 1. <1> to <10> in the figure correspond to <1> to <10> in **Figure 18-73 Timing Chart of SNOOZE Mode Operation (Continuous Startup)**.

2. m = 0; p = 00

18.6.3 SNOOZE mode function

Incorrect:

The SNOOZE mode makes the UART perform reception operations upon RxDq pin input detection while in the STOP mode. Normally the UART stops communication in the STOP mode. However, using the SNOOZE mode enables the UART to perform reception operations without CPU operation. Only UART0 can be set to the SNOOZE mode.

When using UARTq in the SNOOZE mode, make the following settings before entering the STOP mode. (See **Figure 18-92** and **Figure 18-94 Flowchart of SNOOZE Mode Operation**.)

- In the SNOOZE mode, the baud rate setting for UART reception needs to be changed to a value different from that in normal operation. Set the SPSm register and bits 15 to 9 of the SDRmn register with reference to Table 18-3.
- Set the EOCmn and SSECmn bits. This is for enabling or stopping generation of an error interrupt (INTSRE0) when a communication error occurs.
- When using the SNOOZE mode function, set the SWCm bit of serial standby control register m (SSCm) to 1 just before switching to the STOP mode. After the initial setting has completed, set the SSm1 bit of serial channel start register m (SSm) to 1.

Upon detecting the edge of RxDq (start bit input) after a transition was made to the STOP mode, UART reception is started.

Cautions 1. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected for f_{CLK}.

2. The maximum transfer rate when using UARTq in the SNOOZE mode is 4800 bps.

3. When SWCm = 1, UARTq can be used only when the reception operation is started in the STOP mode. When used simultaneously with another SNOOZE mode function or interrupt, if the reception operation is started in a state other than the STOP mode, such as those given below, data may not be received correctly and a framing error or parity error may be generated.

- **When after the SWCm bit has been set to 1, the reception operation is started before the STOP mode is entered**
- **When the reception operation is started while another function is in the SNOOZE mode**
- **When after returning from the STOP mode to normal operation due to an interrupt or other cause, the reception operation is started before the SWCm bit is returned to 0**

4. If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFmn, FEFmn, or OVFmn flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFmn, FEFmn, or OVFmn flag before setting the SWC0 bit to 1 and read the value in bits 7 to 0 (RxDq register) of the SDRm1 register.

18.6.3 SNOOZE mode function

Correct:

The SNOOZE mode makes the UART perform reception operations upon RxDq pin input detection while in the STOP mode. Normally the UART stops communication in the STOP mode. However, using the SNOOZE mode enables the UART to perform reception operations without CPU operation. Only UART0 can be set to the SNOOZE mode.

When using UARTq in the SNOOZE mode, make the following settings before entering the STOP mode. (See **Figure 18-92** and **Figure 18-94 Flowchart of SNOOZE Mode Operation**.)

- In the SNOOZE mode, the baud rate setting for UART reception needs to be changed to a value different from that in normal operation. Set the SPSm register and bits 15 to 9 of the SDRmn register with reference to Table 18-3.
- Set the EOCmn and SSECmn bits. This is for enabling or stopping generation of an error interrupt (INTSRE0) when a communication error occurs.
- When using the SNOOZE mode function, set the SWCm bit of serial standby control register m (SSCm) to 1 just before switching to the STOP mode. After the initial setting has completed, set the SSm1 bit of serial channel start register m (SSm) to 1.

<R> • A UARTq starts reception in SNOOZE mode on detecting input of the start bit on the RxDq pin following a transition of the CPU to the STOP mode.

Cautions 1. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected for f_{CLK}.

2. The maximum transfer rate when using UARTq in the SNOOZE mode is 4800 bps.

3. When SWCm = 1, UARTq can be used only when the reception operation is started in the STOP mode. When used simultaneously with another SNOOZE mode function or interrupt, if the reception operation is started in a state other than the STOP mode, such as those given below, data may not be received correctly and a framing error or parity error may be generated.

- **When after the SWCm bit has been set to 1, the reception operation is started before the STOP mode is entered**
- **When the reception operation is started while another function is in the SNOOZE mode**
- **When after returning from the STOP mode to normal operation due to an interrupt or other cause, the reception operation is started before the SWCm bit is returned to 0**

4. If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFmn, FEFmn, or OVFmn flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFmn, FEFmn, or OVFmn flag before setting the SWC0 bit to 1 and read the value in bits 7 to 0 (RxDq register) of the SDRm1 register.

<R> **5. The CPU shifts from the STOP mode to the SNOOZE mode on detecting the valid edge of the RxDq signal. Note, however, that transfer through the UART channel may not start and the CPU may remain in the SNOOZE mode if an input pulse on the RxDq pin is too short to be detected as a start bit. In such cases, data may not be received correctly, and this may lead to a framing error or parity error in the next UART transfer.**

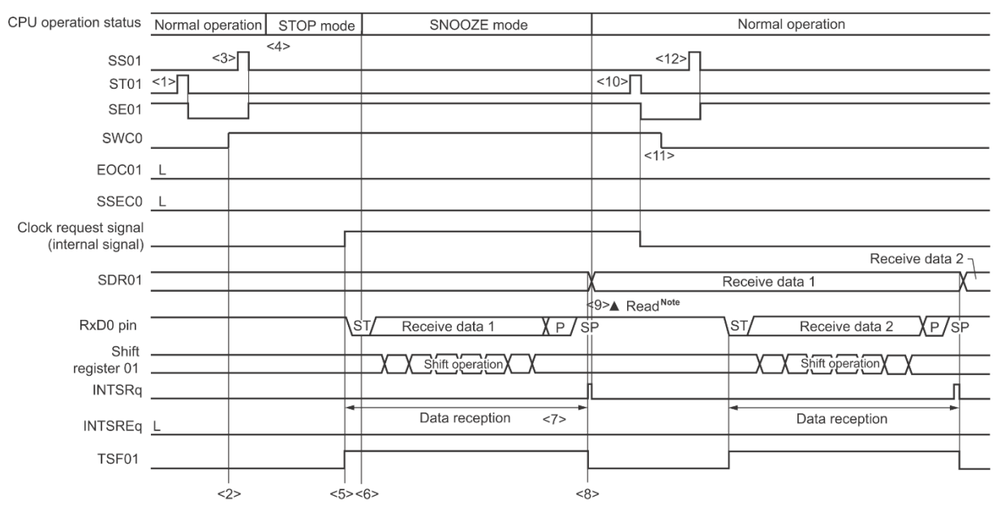
18.6.3 SNOOZE mode function

Incorrect:

(1) SNOOZE mode operation (EOCm1 = 0, SSECm = 0/1)

Because of the setting of EOCm1 = 0, even though a communication error occurs, an error interrupt (INTSREq) is not generated, regardless of the setting of the SSECm bit. A transfer end interrupt (INTSRq) will be generated.

Figure 18-90. Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)



Note Read the received data when SWCm is 1

Caution Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

Remarks 1. <1> to <12> in the figure correspond to <1> to <12> in **Figure 18-92 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)**.

2. m = 0; q = 0

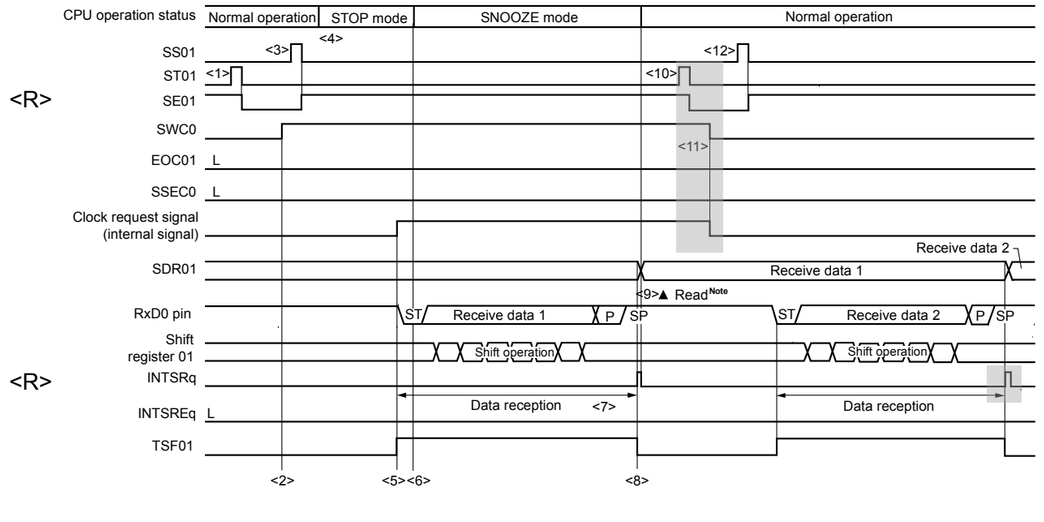
18.6.3 SNOOZE mode function

Correct:

(1) SNOOZE mode operation (EOCm1 = 0, SSECm = 0/1)

Because of the setting of EOCm1 = 0, even though a communication error occurs, an error interrupt (INTSREq) is not generated, regardless of the setting of the SSECm bit. A transfer end interrupt (INTSRq) will be generated.

Figure 18-90. Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)



Note Read the received data when SWCm is 1

Caution Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

Remarks 1. <1> to <12> in the figure correspond to <1> to <12> in **Figure 18-92 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)**.

2. m = 0; q = 0

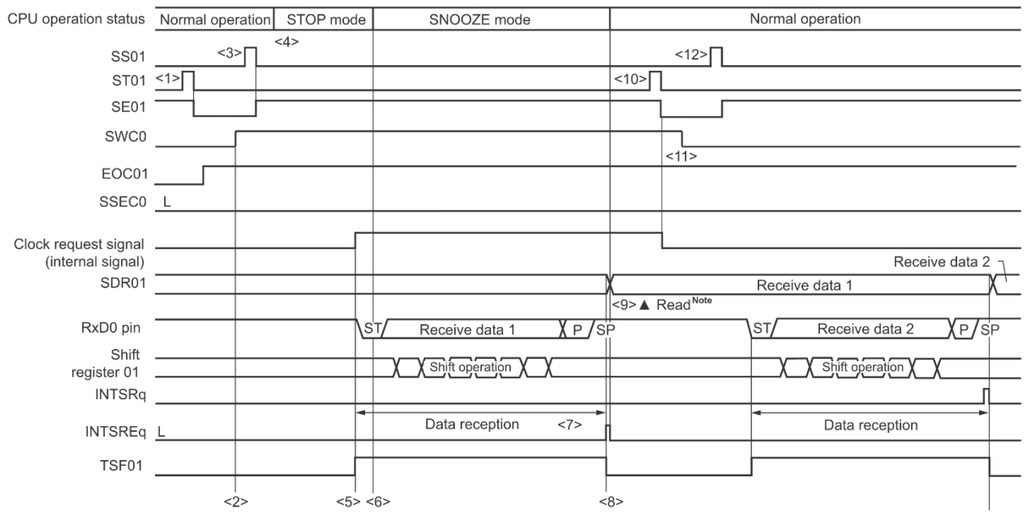
18.6.3 SNOOZE mode function

Incorrect:

(2) SNOOZE mode operation (EOCm1 = 1, SSECm = 0: Error interrupt (INTSREq) generation is enabled)

Because EOCm1 = 1 and SSECm = 0, an error interrupt (INTSREq) is generated when a communication error occurs.

Figure 18-91. Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)



Note Read the received data when SWCm is 1

Caution Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

Remarks 1. <1> to <12> in the figure correspond to <1> to <12> in **Figure 18-92 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)**.

2. m = 0; q = 0

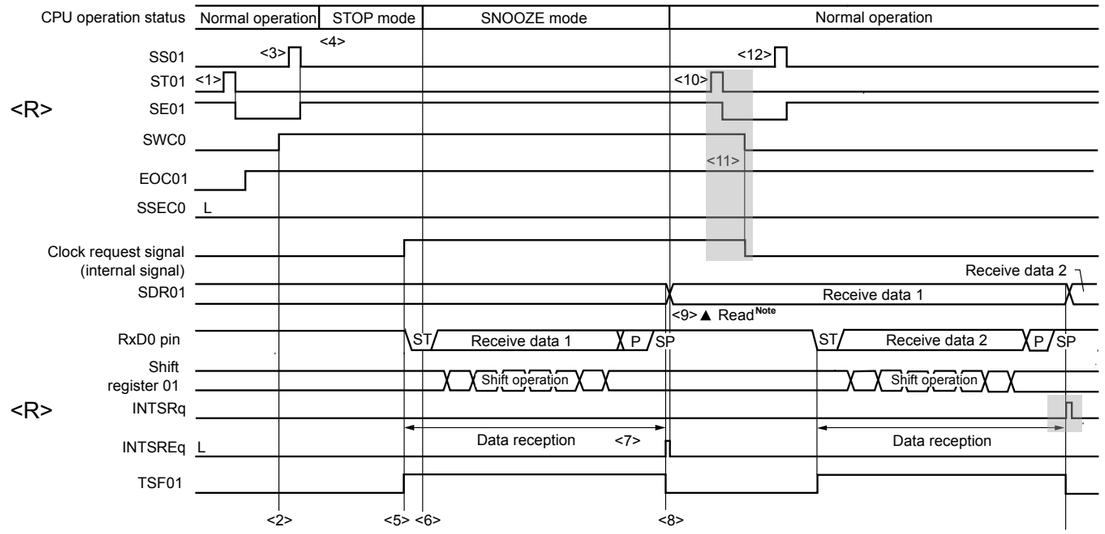
18.6.3 SNOOZE mode function

Correct:

(2) SNOOZE mode operation (EOCm1 = 1, SSECm = 0: Error interrupt (INTSREq) generation is enabled)

Because EOCm1 = 1 and SSECm = 0, an error interrupt (INTSREq) is generated when a communication error occurs.

Figure 18-91. Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)



Note Read the received data when SWCm is 1

Caution Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

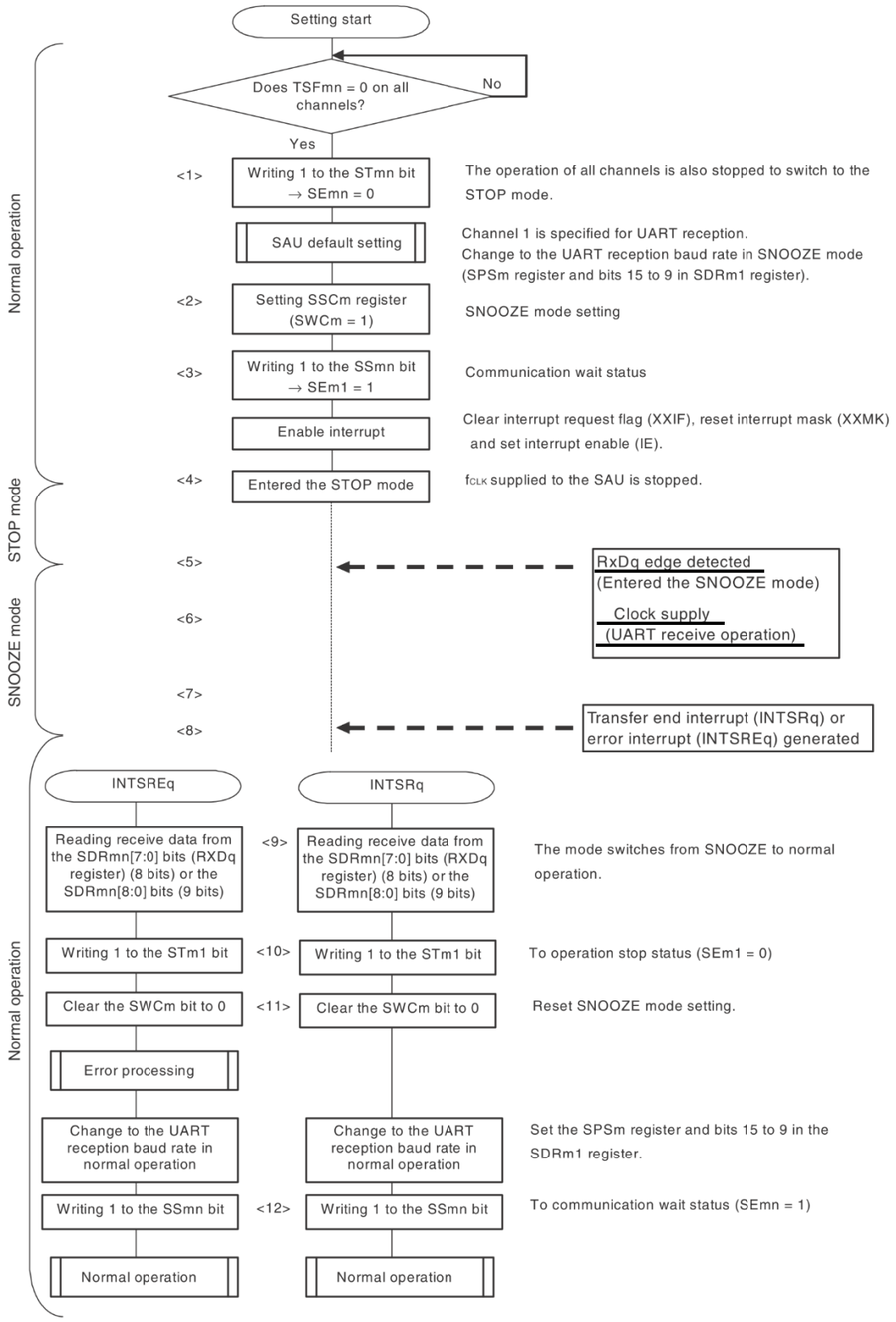
Remarks 1. <1> to <12> in the figure correspond to <1> to <12> in **Figure 18-92 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)**.

2. m = 0; q = 0

18.6.3 SNOOZE mode function

Incorrect:

Figure 18-92. Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)



Remarks 1. <1> to <12> in the figure correspond to <1> to <12> in **Figure 18-90 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)** and **Figure 18-91 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)**.

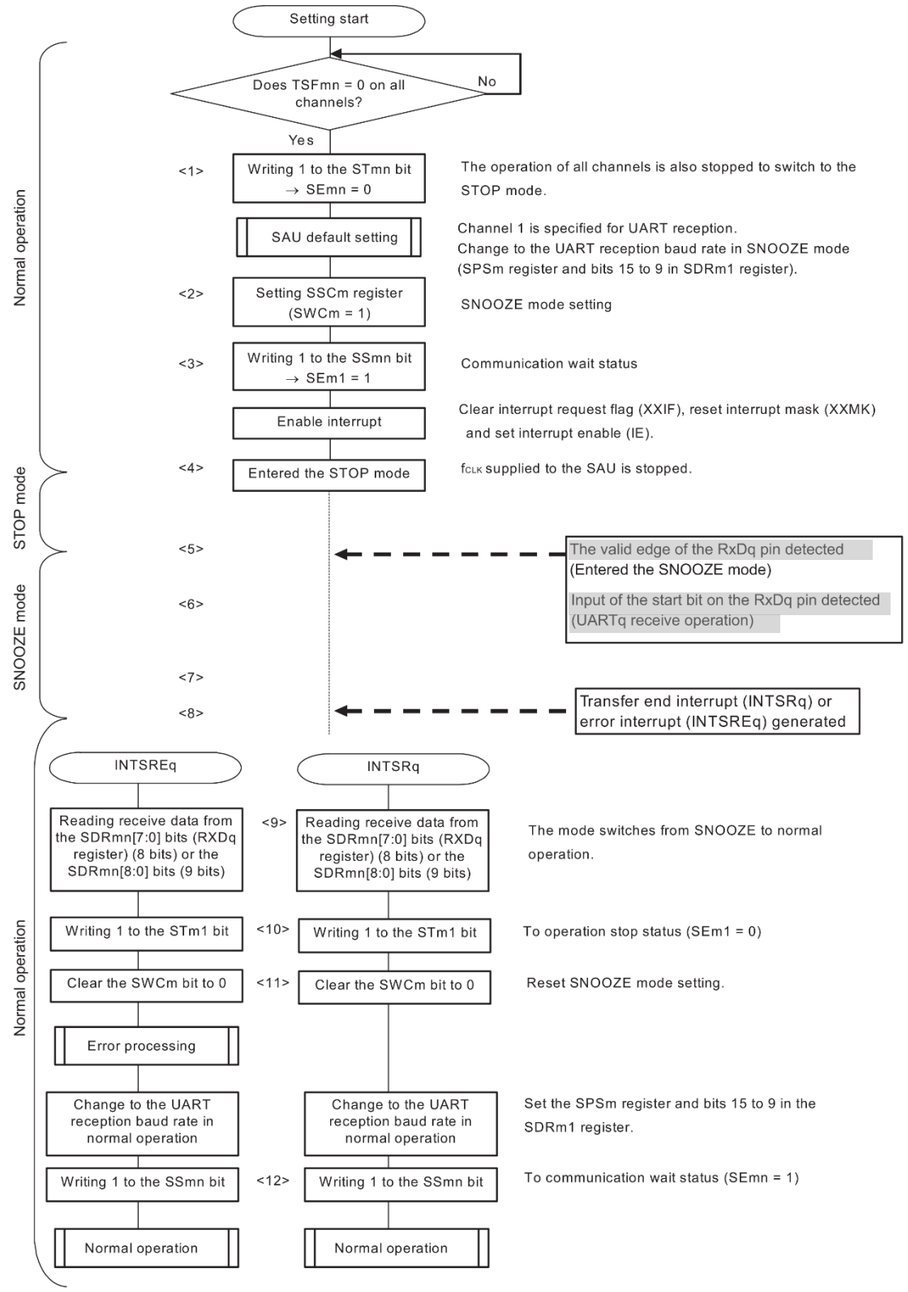
2. m = 0; q = 0

18.6.3 SNOOZE mode function

Correct:

Figure 18-92. Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)

<R>



Remarks 1. <1> to <12> in the figure correspond to <1> to <12> in **Figure 18-90 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)** and **Figure 18-91 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)**.

2. m = 0; q = 0

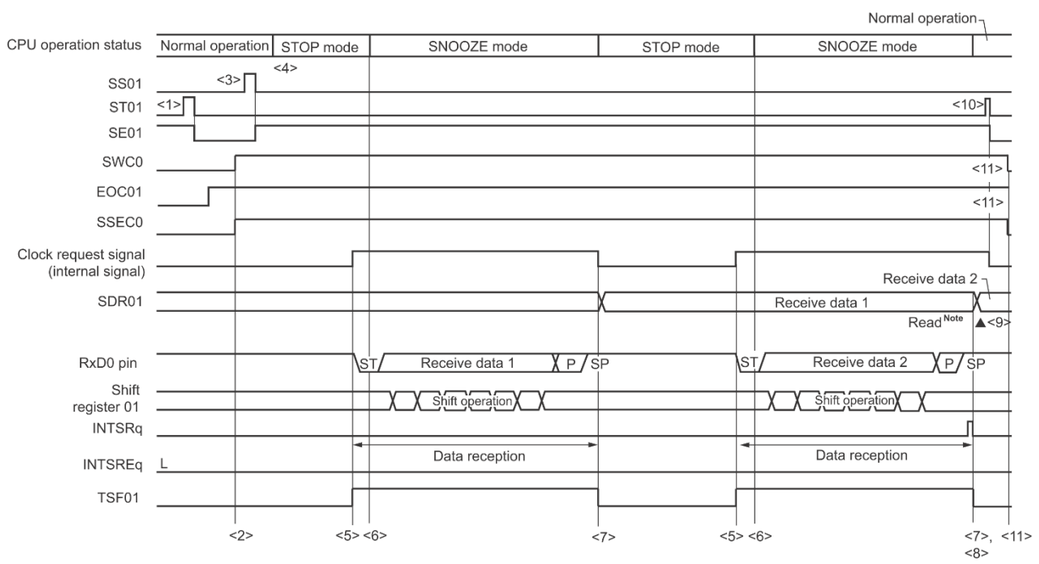
18.6.3 SNOOZE mode function

Incorrect:

(3) SNOOZE mode operation (EOCm1 = 1, SSECm = 1: Error interrupt (INTSREq) generation is stopped)

Because EOCm1 = 1 and SSECm = 1, an error interrupt (INTSREq) is not generated when a communication error occurs.

Figure 18-93. Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)



Note Read the received data when SWCm = 1.

Cautions 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit and stop the operation).
After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).
2. If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFm1, FEFm1, or OVFm1 flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFm1, FEFm1, or OVFm1 flag before setting the SWCm bit to 1 and read the value in SDRm1[7:0] (RxDq register) (8 bits) or SDRm1[8:0] (9 bits).

Remarks 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 18-94 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)**.
2. m = 0; q = 0

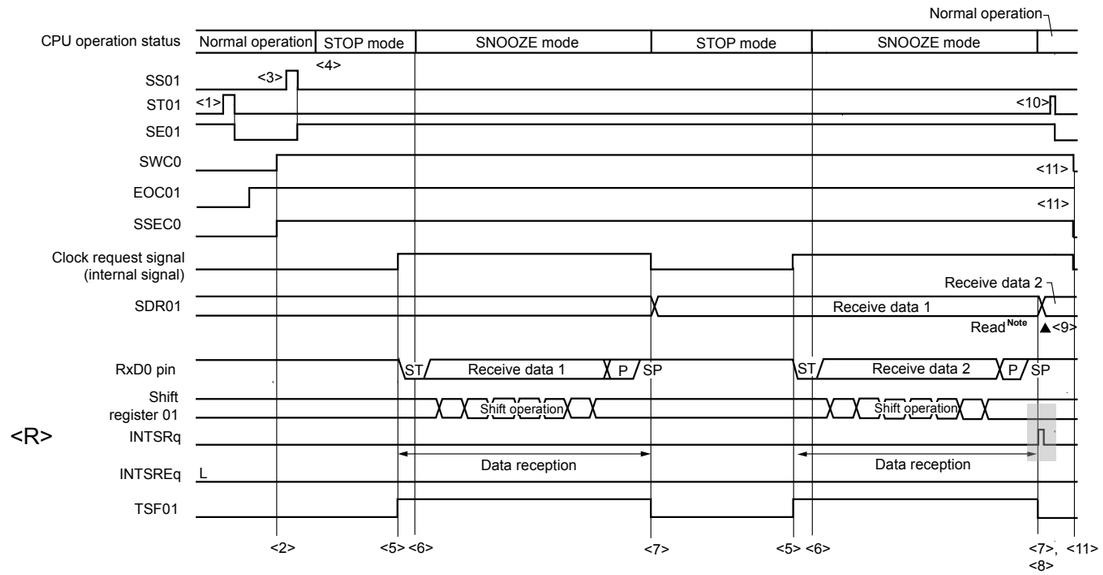
18.6.3 SNOOZE mode function

Correct:

(3) SNOOZE mode operation (EOCm1 = 1, SSECm = 1: Error interrupt (INTSREq) generation is stopped)

Because EOCm1 = 1 and SSECm = 1, an error interrupt (INTSREq) is not generated when a communication error occurs.

Figure 18-93. Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)



Note Read the received data when SWCm = 1.

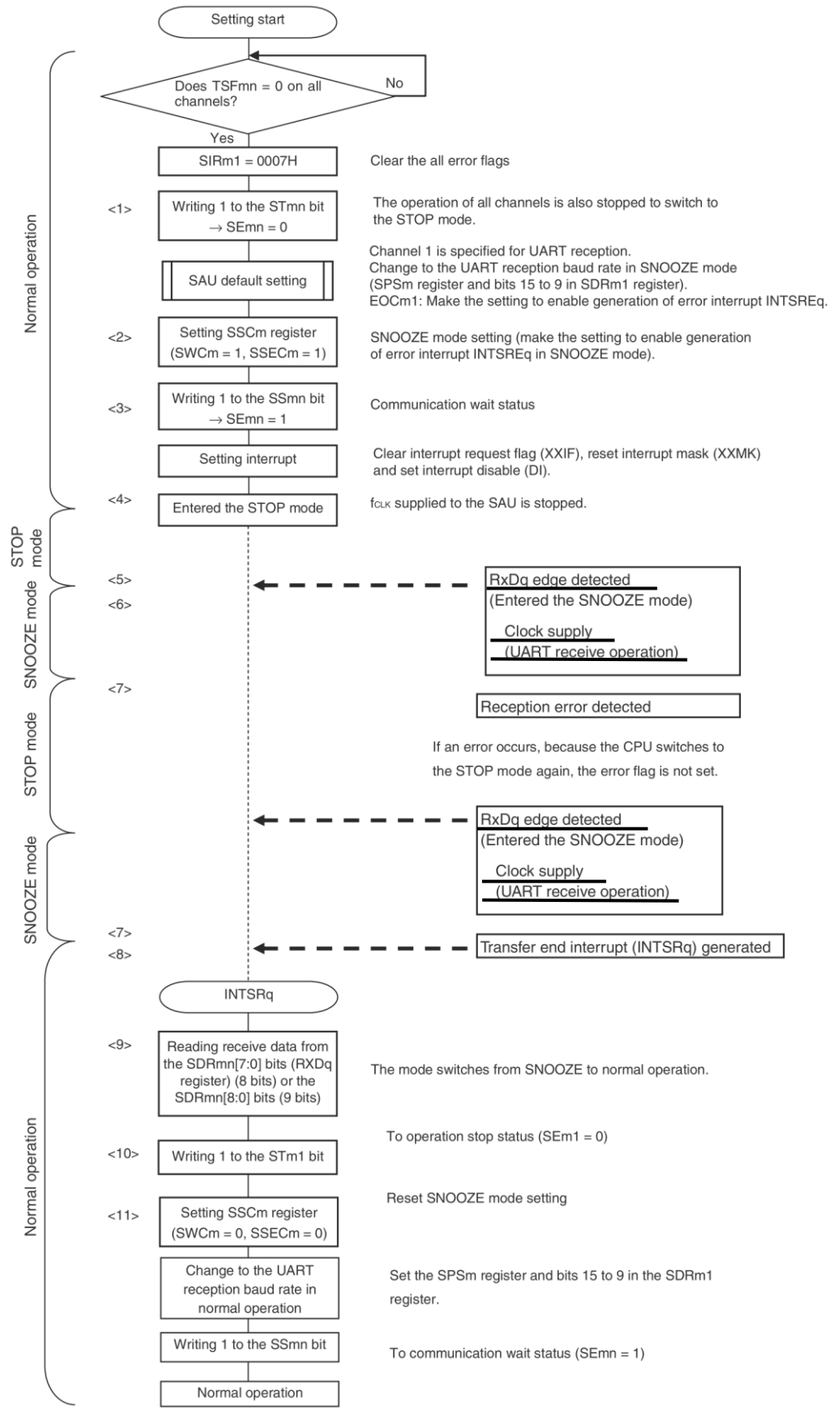
Cautions 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit and stop the operation).
After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).
2. If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFm1, FEFm1, or OVFm1 flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFm1, FEFm1, or OVFm1 flag before setting the SWCm bit to 1 and read the value in SDRm1[7:0] (RxDq register) (8 bits) or SDRm1[8:0] (9 bits).

Remarks 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 18-94 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)**.
2. m = 0; q = 0

18.6.3 SNOOZE mode function

Incorrect:

Figure 18-94. Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)

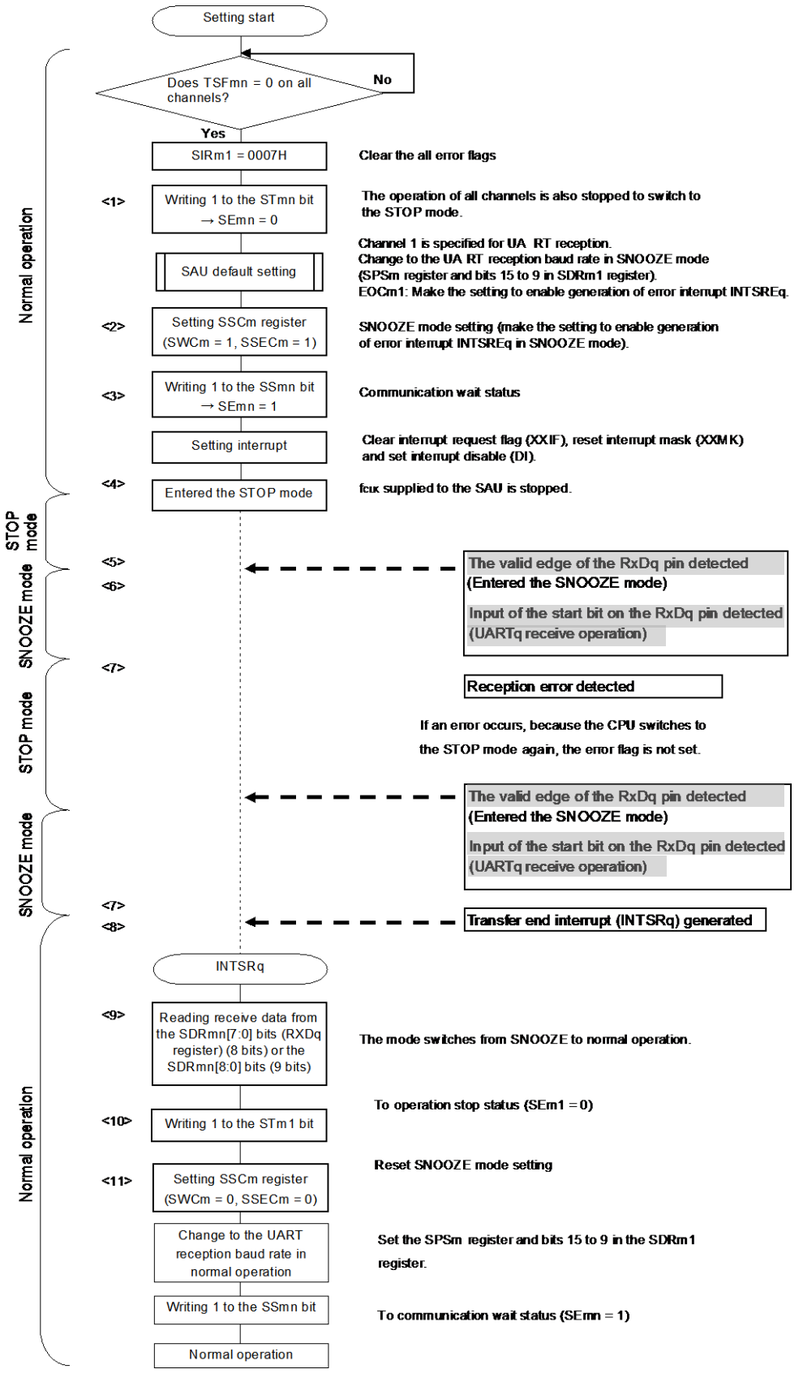


(**Caution** and **Remarks** are listed on the next page.)

18.6.3 SNOOZE mode function

Correct:

Figure 18-94. Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)

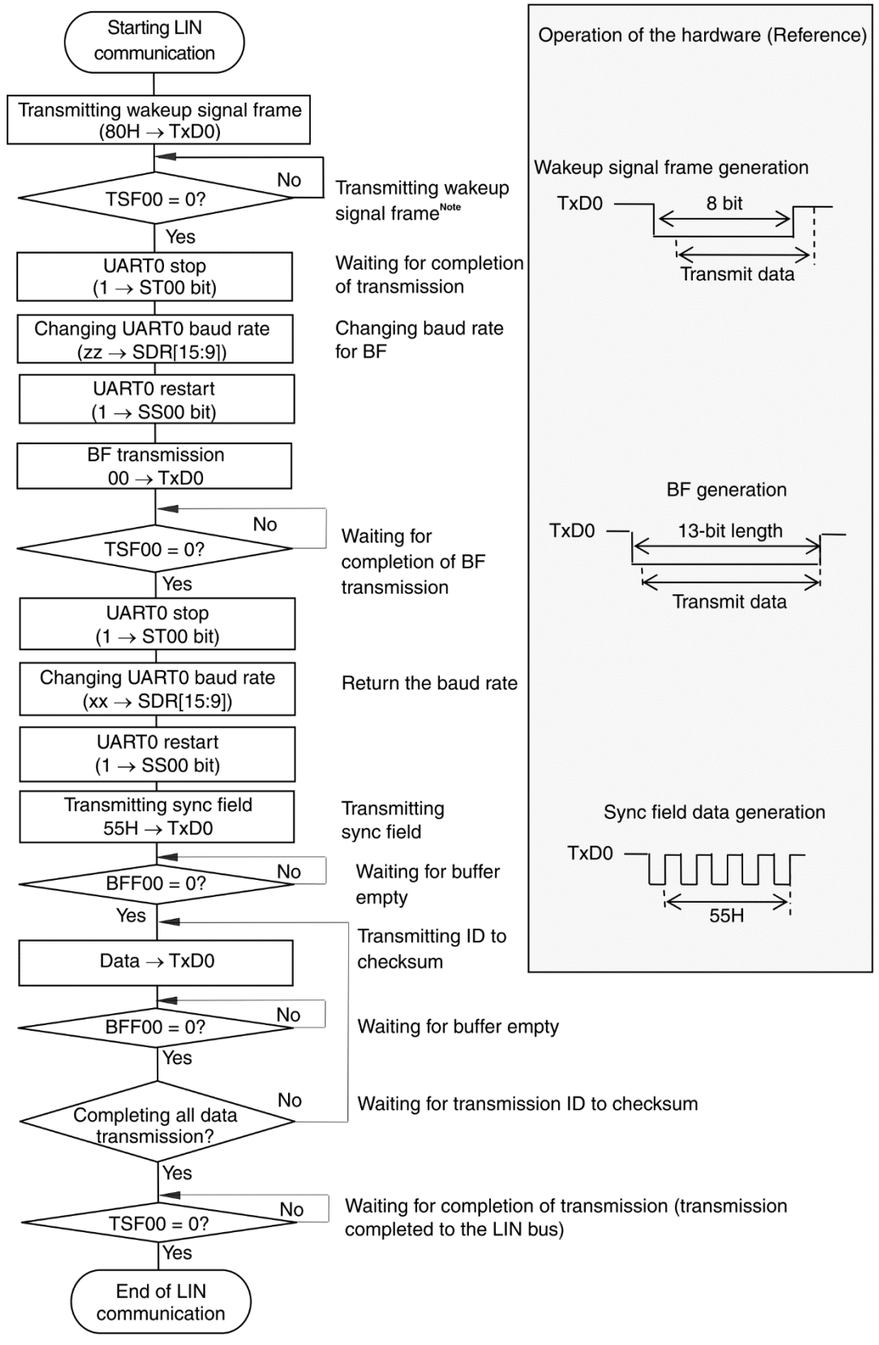


(**Caution** and **Remarks** are listed on the next page.)

18. 7. 1 LIN transmission

Incorrect:

Figure 18-99. Flowchart for LIN Transmission



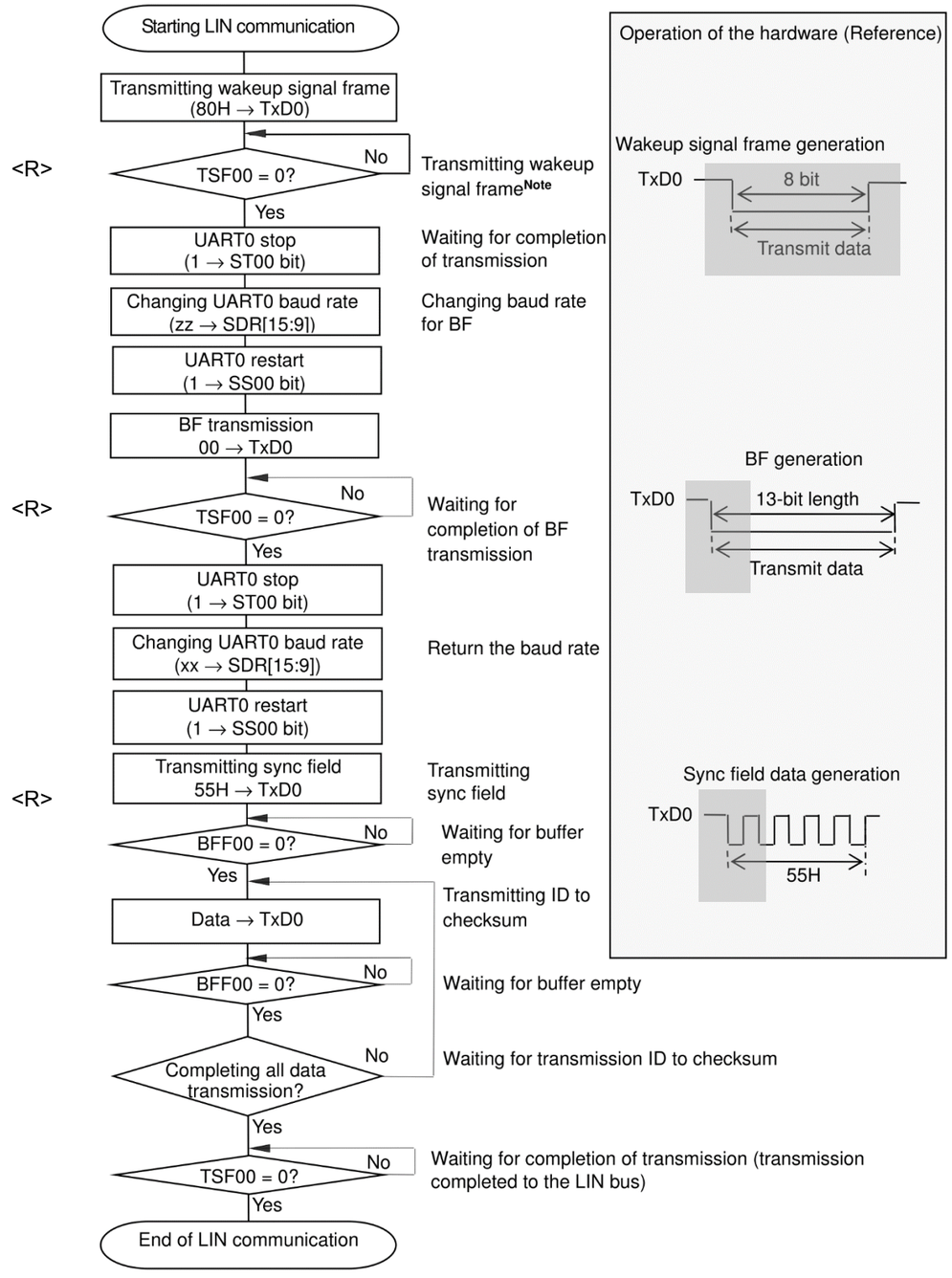
Note When LIN-bus start from sleep status only

Remark Default setting of the UART is complete, and the flow from the transmission enable status.

18. 7. 1 LIN transmission

Correct:

Figure 18-99. Flowchart for LIN Transmission



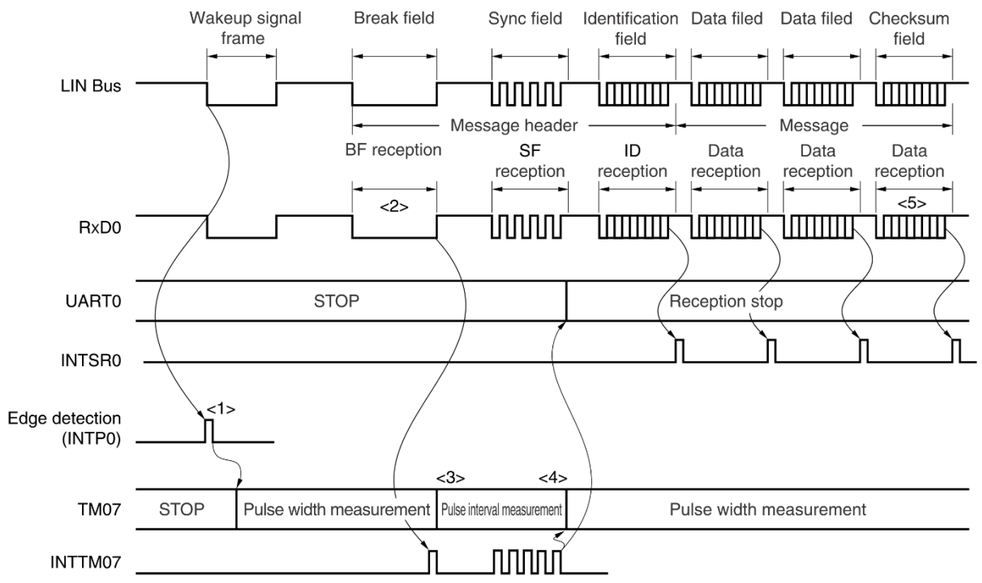
Note When LIN-bus start from sleep status only

Remark Default setting of the UART is complete, and the flow from the transmission enable status.

18. 7. 2 LIN reception

Incorrect:

Figure 18-100. Reception Operation of LIN

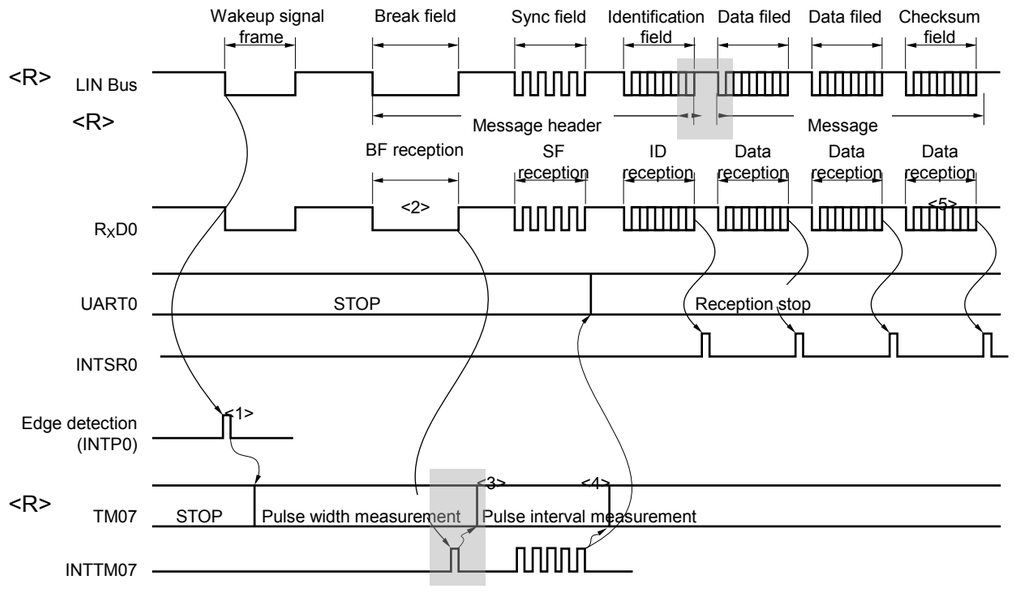


Here is the flow of signal processing.

18. 7. 2 LIN reception

Correct:

Figure 18-100. Reception Operation of LIN



Here is the flow of signal processing.

19.3.6 IICA low-level width setting register n (IICWLn)

Incorrect:

This register is used to set the low-level width (t_{LOW}) of the SCLAn pin signal that is output by serial interface IICA and to control the SDAAn pin signal.

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

Set the IICWLn register while operation of I^2C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) is 0).

Reset signal generation sets this register to FFH.

For details about setting the IICWLn register, see **19.4.2 Setting transfer clock by using IICWLn and IICWHn registers**.

Figure 19-10. Format of IICA Low-Level Width Setting Register n (IICWLn)

Address: F0232H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWLn								

19.3.6 IICA low-level width setting register n (IICWLn)

Correct:

This register is used to set the low-level width (t_{LOW}) of the SCLAn pin signal that is output by serial interface IICA and to control the SDAAn pin signal.

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

Set the IICWLn register while operation of I^2C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) is 0).

Reset signal generation sets this register to FFH.

For details about setting the IICWLn register, see **19.4.2 Setting transfer clock by using IICWLn and IICWHn registers**.

<R> The data hold time is one-quarter of the time set by the IICWLn register.

Figure 19-10. Format of IICA Low-Level Width Setting Register n (IICWLn)

Address: F0232H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWLn								

19.5.14 Communication reservation

Incorrect:

(1) When communication reservation function is enabled (bit n (IICRSVn) of IICA flag register n (IICFn) = 0)

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled (ACK is not returned and the bus was released by setting bit 6 (LRELn) of IICA control register n0 (IICCTLn0) to 1 and saving communication).

If bit 1 (STTn) of the IICCTLn0 register is set to 1 while the bus is not used (after a stop condition is detected), a start condition is automatically generated and wait state is set.
If an address is written to the IICA shift register n (IICAn) after bit 4 (SPIEn) of the IICCTLn0 register was set to 1, and it was detected by generation of an interrupt request signal (INTIICAn) that the bus was released (detection of the stop condition), then the device automatically starts communication as the master. Data written to the IICAn register before the stop condition is detected is invalid.
When the STTn bit has been set to 1, the operation mode (as start condition or as communication reservation) is determined according to the bus status.

- If the bus has been released ..... a start condition is generated
- If the bus has not been released (standby mode)..... communication reservation

Check whether the communication reservation operates or not by using the MSTSn bit (bit 7 of the IICA status register n (IICSn)) after the STTn bit is set to 1 and the wait time elapses.
Use software to secure the wait time calculated by the following expression.

Wait time from setting STTn = 1 to checking the MSTSn flag (number of f_{MCK} clocks):
(IICWLn setting value + IICWHn setting value + 4) + $t_F \times 2 \times f_{MCK}$ [clocks]

Remarks 1. IICWLn: IICA low-level width setting register n
IICWHn: IICA high-level width setting register n
t_F: SDAAn and SCLAn signal falling times
f_{MCK}: IICA operation clock frequency
2. n = 0

Correct:

(1) When communication reservation function is enabled (bit n (IICRSVn) of IICA flag register n (IICFn) = 0)

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled (ACK is not returned and the bus was released by setting bit 6 (LRELn) of IICA control register n0 (IICCTLn0) to 1 and saving communication).

If bit 1 (STTn) of the IICCTLn0 register is set to 1 while the bus is not used (after a stop condition is detected), a start condition is automatically generated and wait state is set.
If an address is written to the IICA shift register n (IICAn) after bit 4 (SPIEn) of the IICCTLn0 register was set to 1, and it was detected by generation of an interrupt request signal (INTIICAn) that the bus was released (detection of the stop condition), then the device automatically starts communication as the master. Data written to the IICAn register before the stop condition is detected is invalid.
When the STTn bit has been set to 1, the operation mode (as start condition or as communication reservation) is determined according to the bus status.

- If the bus has been released ..... a start condition is generated
- If the bus has not been released (standby mode) ..... communication reservation

Check whether the communication reservation operates or not by using the MSTSn bit (bit 7 of the IICA status register n (IICSn)) after the STTn bit is set to 1 and the wait time elapses.
Use software to secure the wait time calculated by the following expression.

<R>

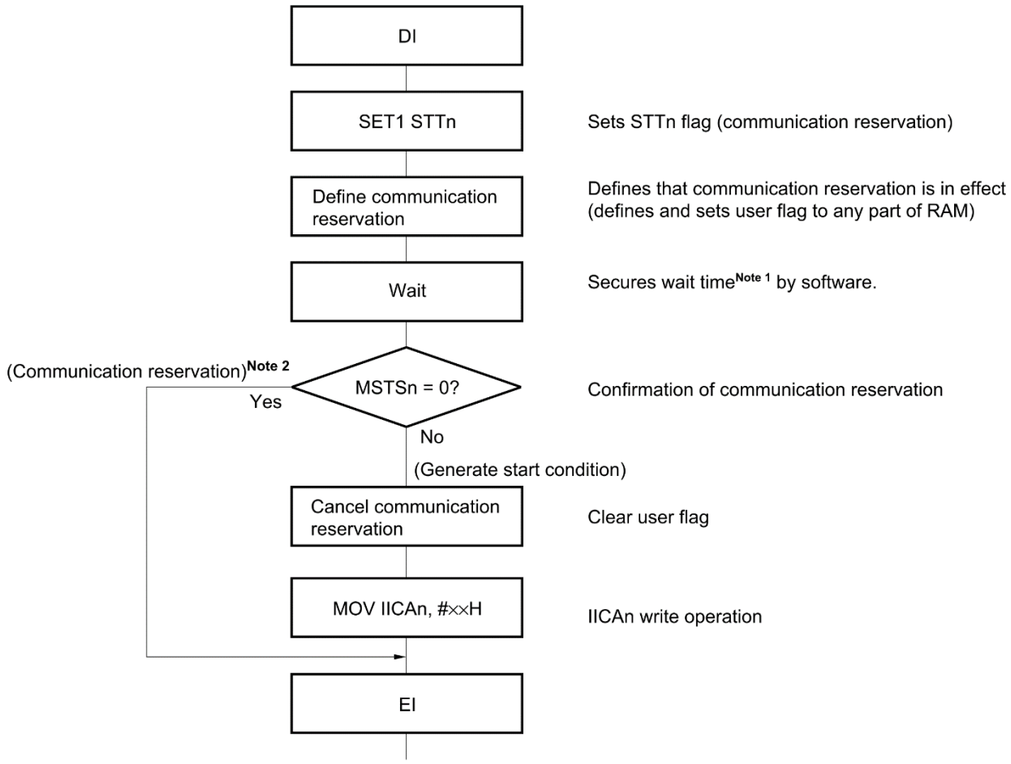
Wait time from setting STTn = 1 to checking the MSTSn flag:
(IICWLn setting value + IICWHn setting value + 4)/fMCK + tF × 2

Remarks 1. IICWLn: IICA low-level width setting register n
IICWHn: IICA high-level width setting register n
t_F: SDAAn and SCLAn signal falling times
f_{MCK}: IICA operation clock frequency
2. n = 0

19.5.14 Communication reservation

Incorrect:

Figure 19-27. Communication Reservation Protocol



Notes 1. The wait time (number of f_{MCK} clocks) is calculated as follows.

$(IICWLn \text{ setting value} + IICWHn \text{ setting value} + 4) + t_F \times 2 \times f_{MCK}$ [clocks]

2. The communication reservation operation executes a write to the IICA shift register n (IICAn) when a stop condition interrupt request occurs.

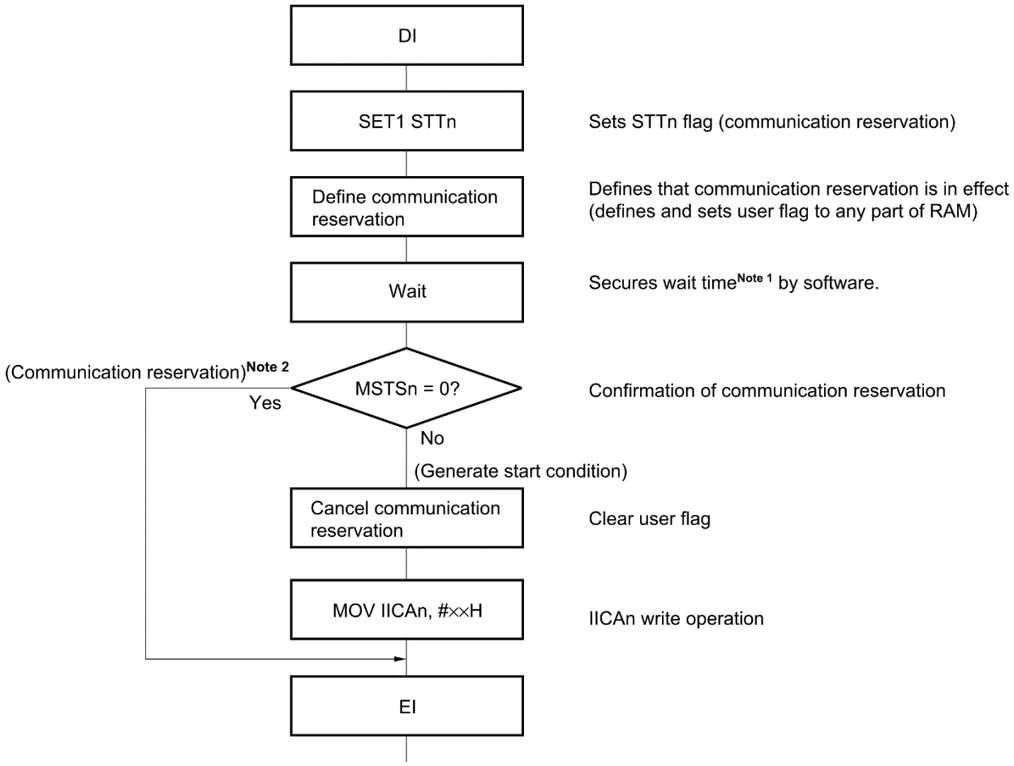
Remarks 1. STTn: Bit 1 of IICA control register n0 (IICCTLn0)
MSTSn: Bit 7 of IICA status register n (IICSn)
IICAn: IICA shift register n
IICWLn: IICA low-level width setting register n
IICWHn: IICA high-level width setting register n
t_F: SDAAn and SCLAn signal falling times
f_{MCK}: IICA operation clock frequency

2. n = 0

19.5.14 Communication reservation

Correct:

Figure 19-27. Communication Reservation Protocol



<R> **Notes 1.** The wait time is calculated as follows.

$(IICWLn \text{ setting value} + IICWHn \text{ setting value} + 4)/fMCK + tF \times 2$

2. The communication reservation operation executes a write to the IICA shift register n (IICAn) when a stop condition interrupt request occurs.

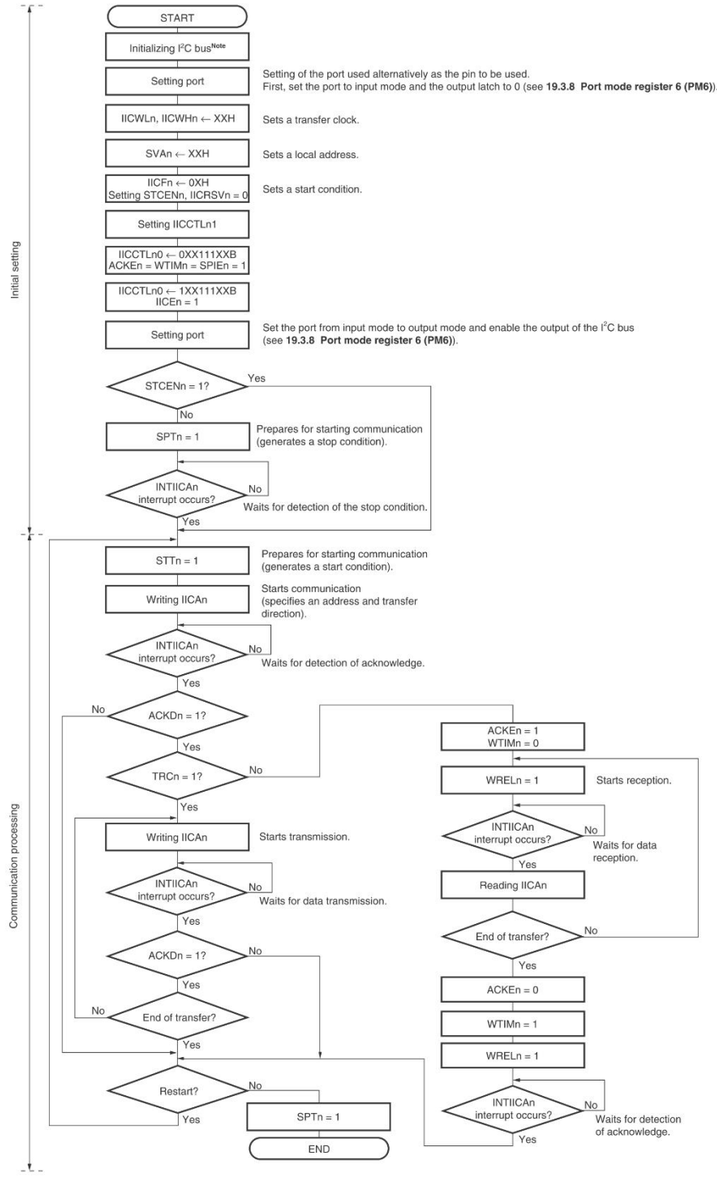
Remarks 1. STTn: Bit 1 of IICA control register n0 (IICCTLn0)
MSTSn: Bit 7 of IICA status register n (IICSn)
IICAn: IICA shift register n
IICWLn: IICA low-level width setting register n
IICWHn: IICA high-level width setting register n
t_F: SDAAn and SCLAn signal falling times
f_{MCK}: IICA operation clock frequency

2. n = 0

19.5.16 Communication operations

Incorrect:

Figure 19-28. Master Operation in Single-Master System



Note Release (SCLAn and SDAAn pins = high level) the I²C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDAAn pin, for example, set the SCLAn pin in the output port mode, and output a clock pulse from the output port until the SDAAn pin is constantly at high level.

Remarks 1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

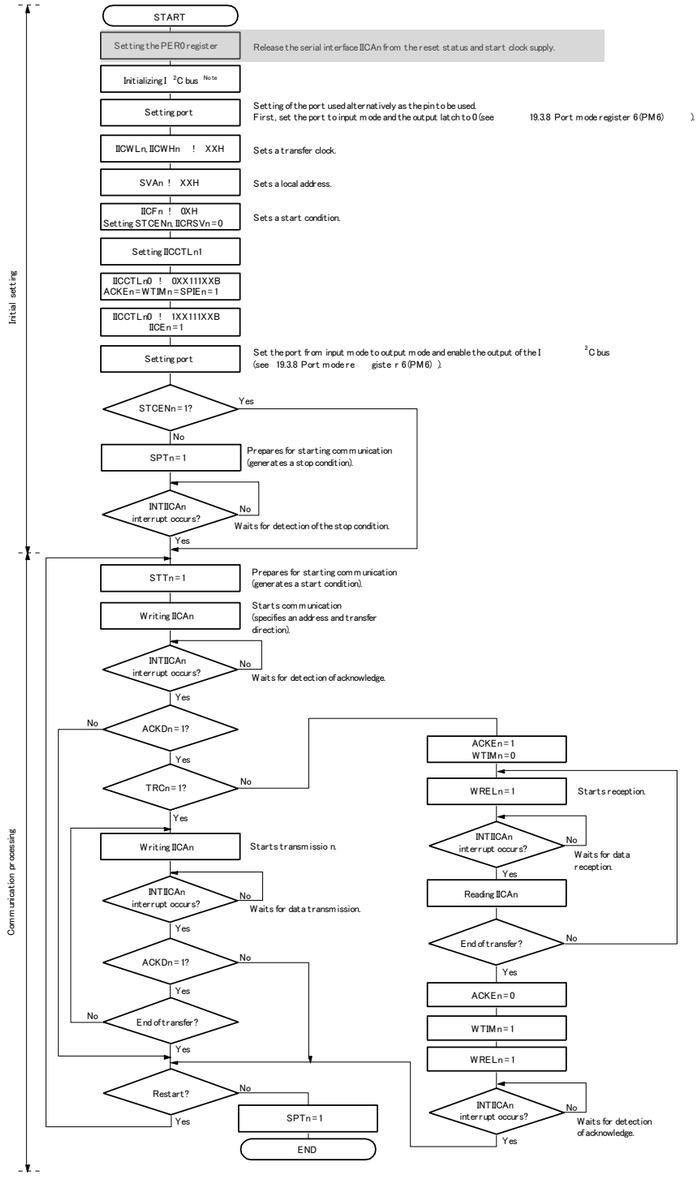
2. n = 0

19.5.16 Communication operations

Correct:

<R>

Figure 19-28. Master Operation in Single-Master System



Note Release (SCLAn and SDAAn pins = high level) the I²C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDAAn pin, for example, set the SCLAn pin in the output port mode, and output a clock pulse from the output port until the SDAAn pin is constantly at high level.

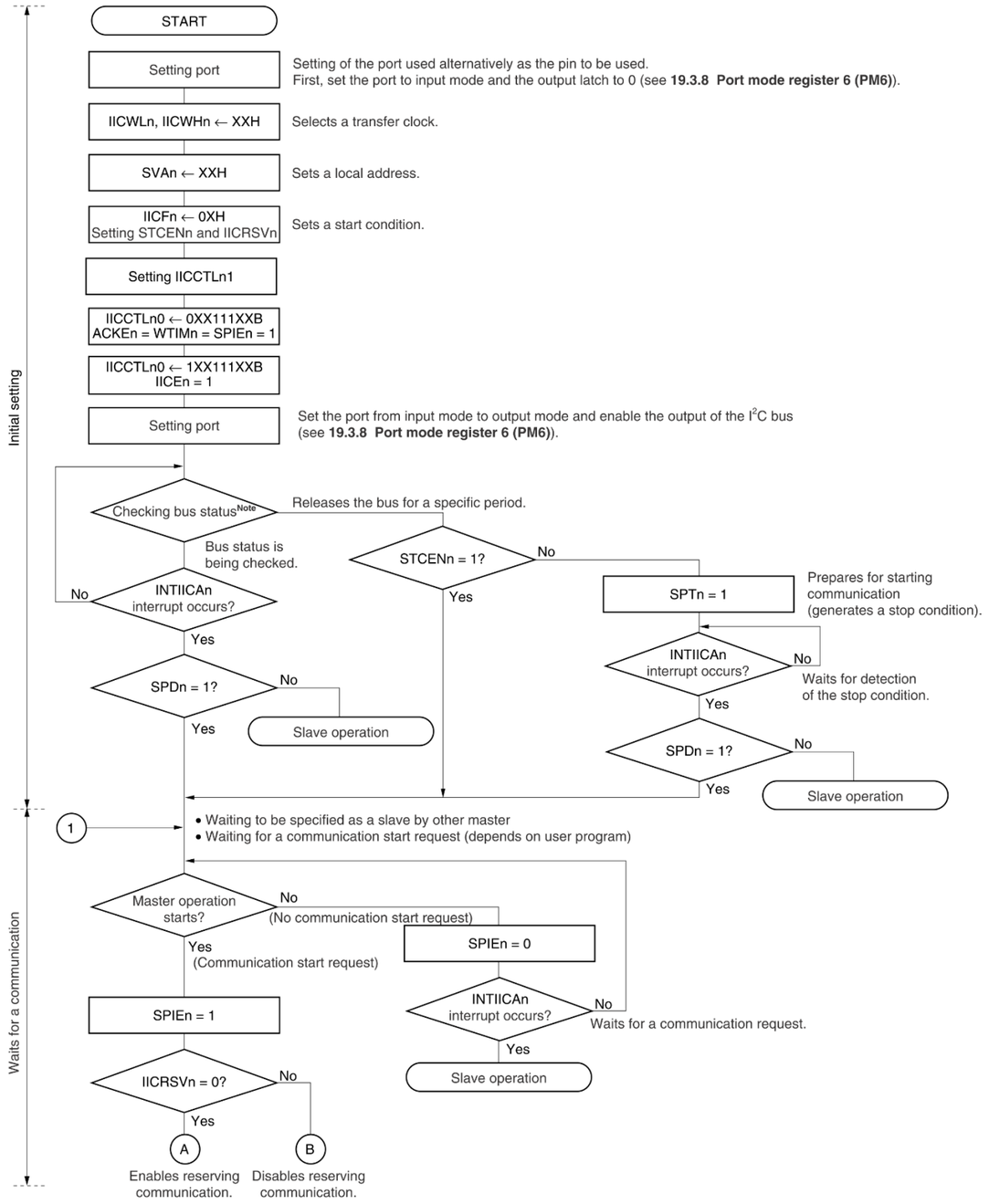
Remarks 1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

2. n = 0

19.5.16 Communication operations

Incorrect:

Figure 19-29. Master Operation in Multi-Master System (1/3)



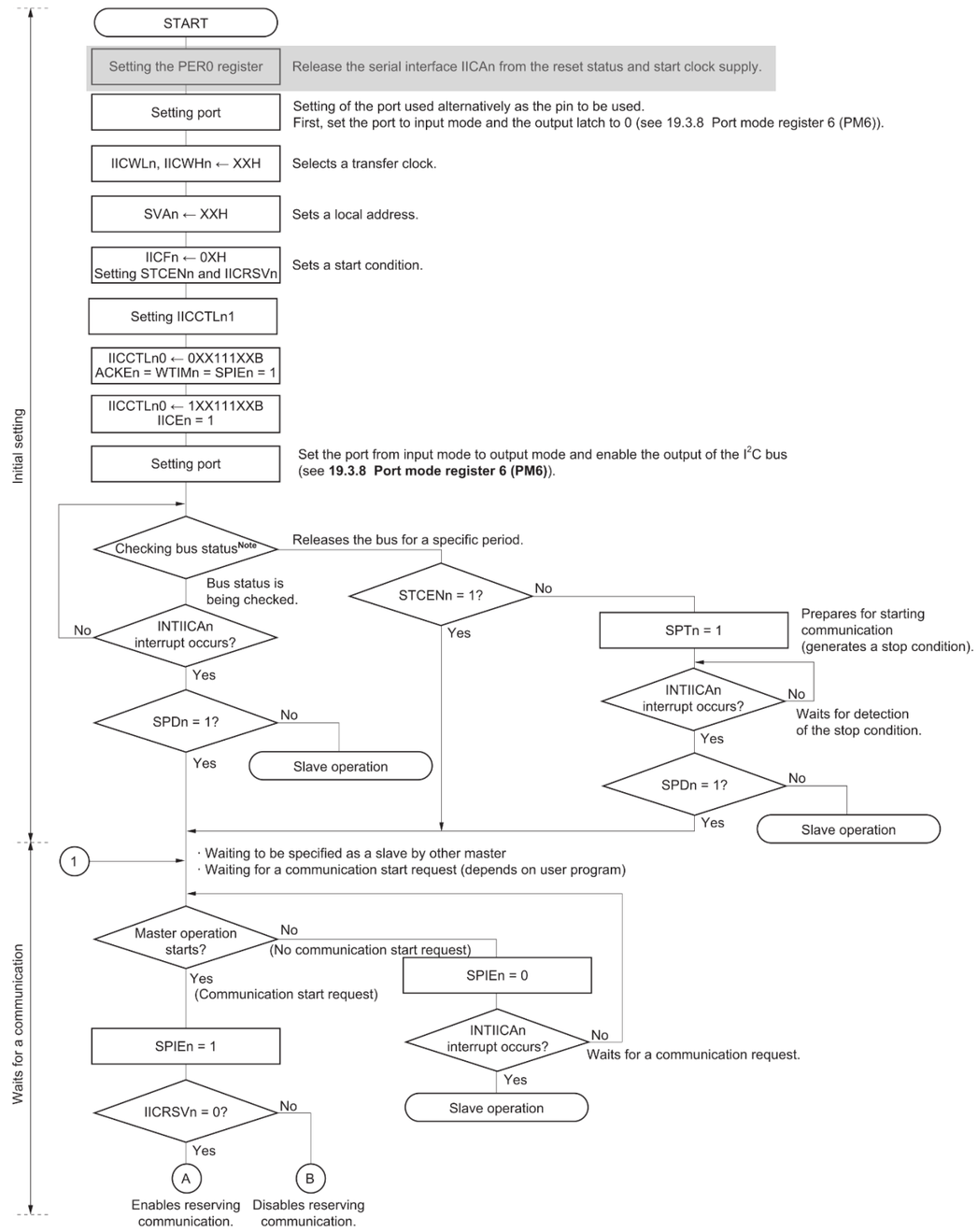
Note Confirm that the bus is released (CLDn bit = 1, DADn bit = 1) for a specific period (for example, for a period of one frame). If the SDAAn pin is constantly at low level, decide whether to release the I^2C bus (SCLAn and SDAAn pins = high level) in conformance with the specifications of the product that is communicating.

Remark n = 0

19.5.16 Communication operations

Correct:

Figure 19-29. Master Operation in Multi-Master System (1/3)



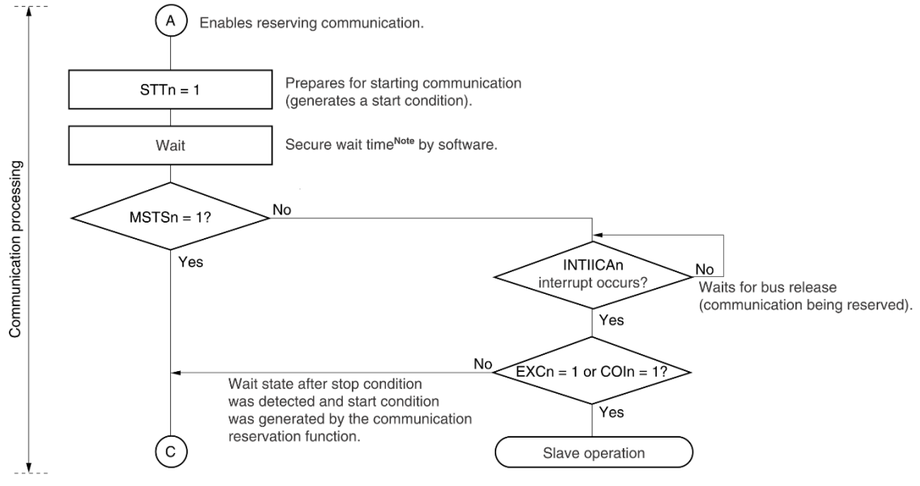
Note Confirm that the bus is released (CLDn bit = 1, DADn bit = 1) for a specific period (for example, for a period of one frame). If the SDAAn pin is constantly at low level, decide whether to release the I^2C bus (SCLAn and SDAAn pins = high level) in conformance with the specifications of the product that is communicating.

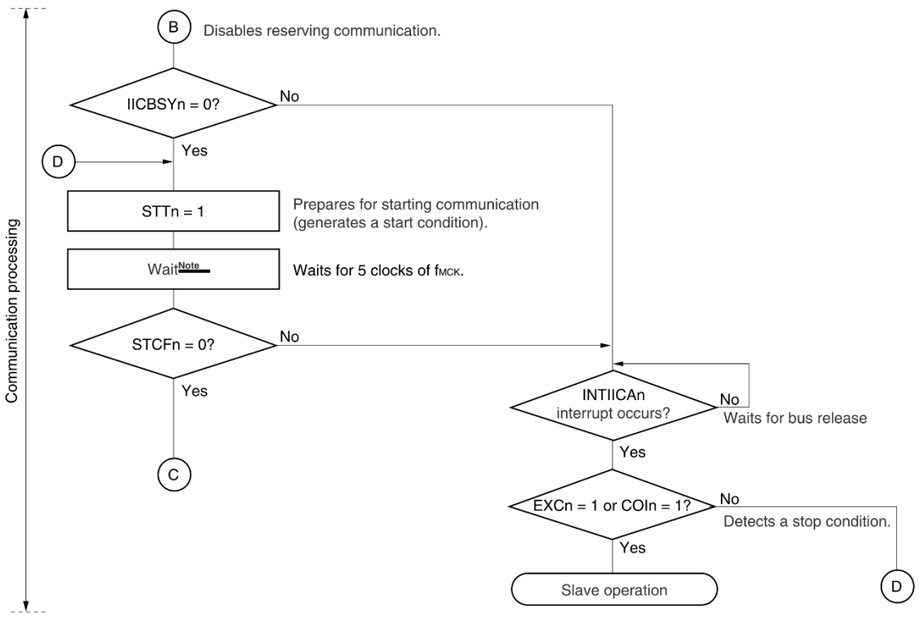
Remark n = 0

19.5.16 Communication operations

Incorrect:

Figure 19-29. Master Operation in Multi-Master System (2/3)





Note The wait time (number of f_{MCK} clocks) is calculated as follows.

(IICWLn setting value + IICWHn setting value + 4) + $t_F \times 2 \times f_{MCK}$ [clocks]

Remarks 1. IICWLn: IICA low-level width setting register n

IICWHn: IICA high-level width setting register n

t_F: SDAAn and SCLAn signal falling times

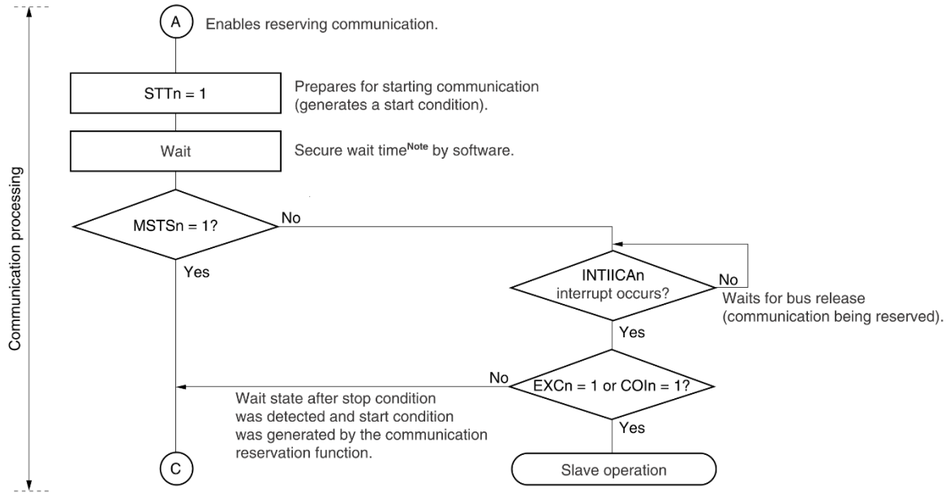
f_{MCK}: IICA operation clock frequency

2. n = 0

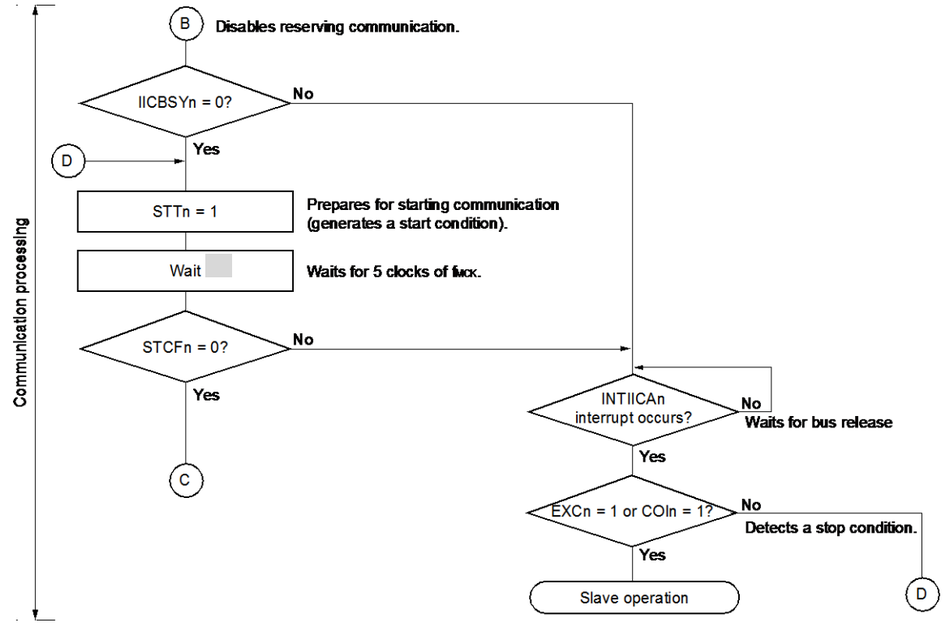
19.5.16 Communication operations

Correct:

Figure 19-29. Master Operation in Multi-Master System (2/3)



<R>



<R> **Note** The wait time is calculated as follows.

(IICWLn setting value + IICWHn setting value + 4)/fMCK + tF × 2

Remarks 1. IICWLn: IICA low-level width setting register n

IICWHn: IICA high-level width setting register n

t_F: SDAAn and SCLAn signal falling times

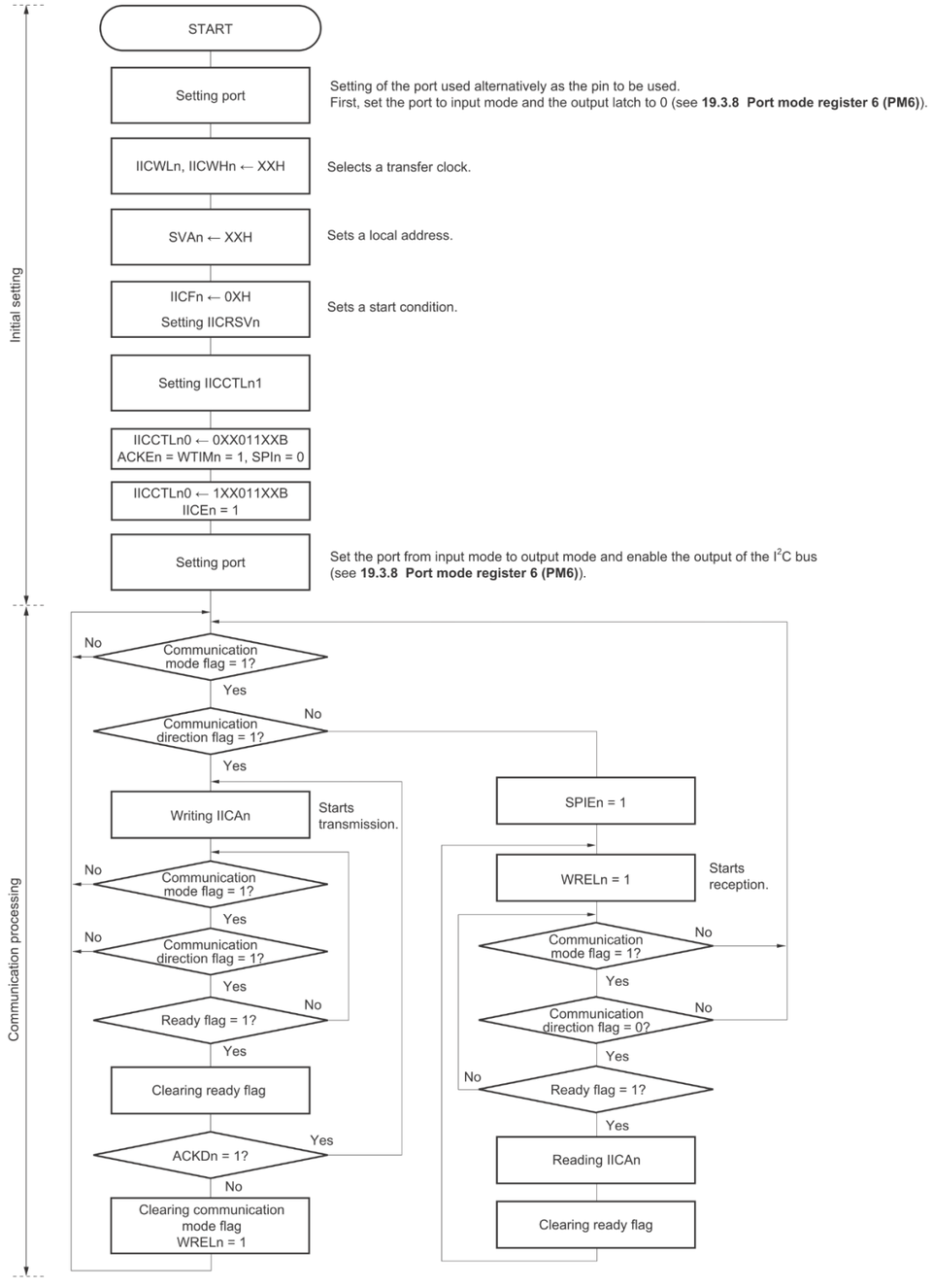
f_{MCK}: IICA operation clock frequency

2. n = 0

19.5.16 Communication operations

Incorrect:

Figure 19-30. Slave Operation Flowchart (1)



Remarks 1. Conform to the specifications of the product that is in communication, regarding the transmission and reception formats.

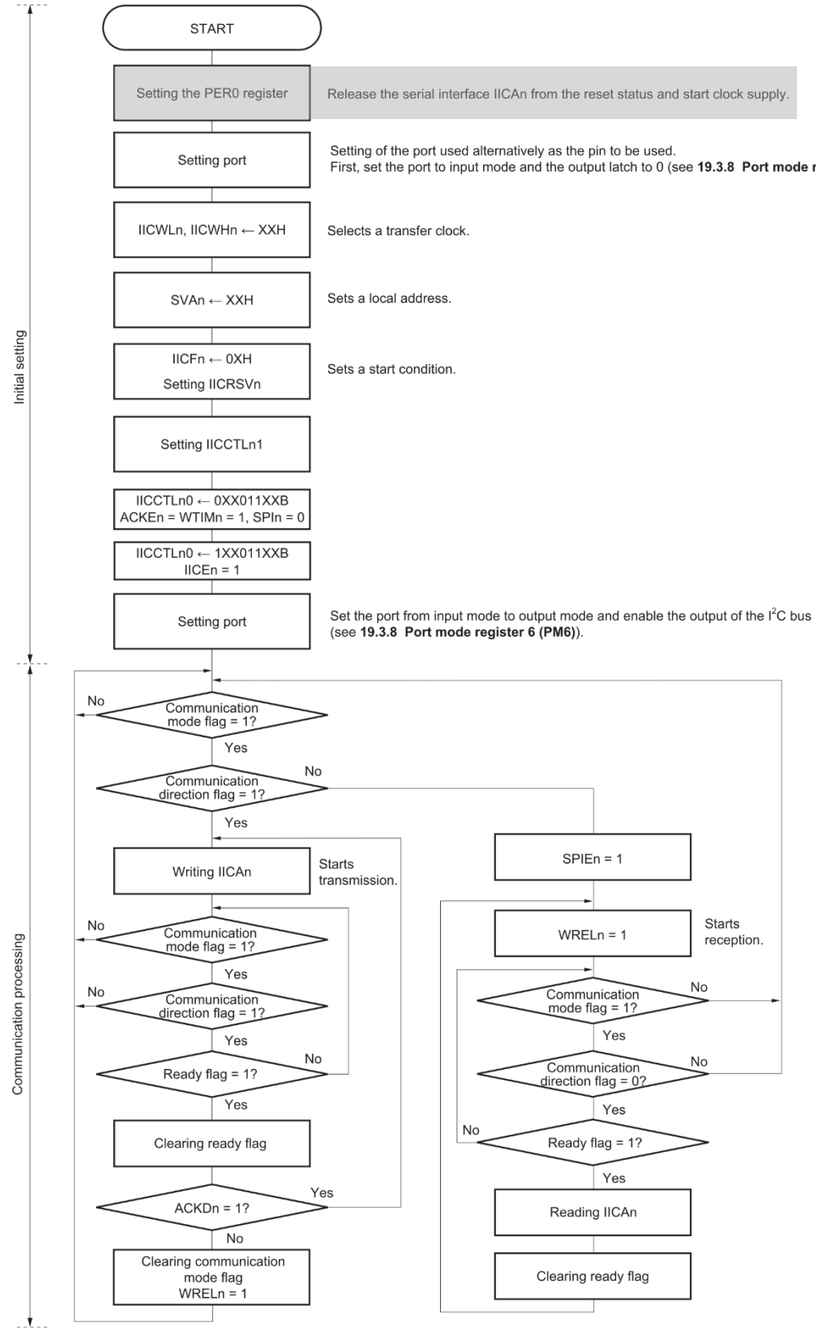
2.. n = 0

19.5.16 Communication operations

Correct:

<R>

Figure 19-30. Slave Operation Flowchart (1)



Remarks 1. Conform to the specifications of the product that is in communication, regarding the transmission and reception formats.

2.. n = 0

21.3.7 LCD port function registers 0 to 5 (PFSEG0 to PFSEG5)

Incorrect:

Figure 21-10. Format of LCD Port Function Registers 0 to 5

Address: F0300H After reset: F0H R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG0	PFSEG07	PFSEG06	PFSEG05	PFSEG04	0	0	0	0

Address: F0301H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG1	PFSEG15	PFSEG14	PFSEG13	PFSEG12	PFSEG11	PFSEG10	PFSEG09	PFSEG08

Address: F0302H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG2	PFSEG23	PFSEG22	PFSEG21	PFSEG20	PFSEG19	PFSEG18	PFSEG17	PFSEG16

Address: F0303H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG3	PFSEG31Note	PFSEG30Note	PFSEG29Note	PFSEG28Note	PFSEG27	PFSEG26	PFSEG25	PFSEG24

Address: F0304H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG4	PFSEG39Note	PFSEG38Note	PFSEG37	PFSEG36	PFSEG35	PFSEG34	PFSEG33	PFSEG32

Address: F0305H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG5	0	0	0	0	0	0	PFSEG41Note	PFSEG40Note

PFSEGxx (xx = 04 to 41)	Port (other than segment output)/segment outputs specification of Pmn pins (mn = 02 to 07, 10 to 17, 30 to 37, 50 to 57, 70 to 77, 80 to 85)
0	Used as port (other than segment output)
1	Used as segment output

Note 100-pin products only.

Correct:

Figure 21-10. Format of LCD Port Function Registers 0 to 5

Address: F0300H After reset: F0H R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG0	PFSEG07	PFSEG06	PFSEG05	PFSEG04	0	0	0	0

Address: F0301H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG1	PFSEG15	PFSEG14	PFSEG13	PFSEG12	PFSEG11	PFSEG10	PFSEG09	PFSEG08

Address: F0302H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG2	PFSEG23	PFSEG22	PFSEG21	PFSEG20	PFSEG19	PFSEG18	PFSEG17	PFSEG16

Address: F0303H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG3	PFSEG31Note	PFSEG30Note	PFSEG29Note	PFSEG28Note	PFSEG27	PFSEG26	PFSEG25	PFSEG24

Address: F0304H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG4	PFSEG39Note	PFSEG38Note	PFSEG37	PFSEG36	PFSEG35	PFSEG34	PFSEG33	PFSEG32

Address: F0305H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG5	0	0	0	0	0	0	PFSEG41Note	PFSEG40Note

PFSEGxx (xx = 04 to 41)	Port (other than segment output)/segment outputs specification of Pmn pins (mn = 02 to 07, 10 to 17, 30 to 37, 50 to 57, 70 to 77, 80 to 85)
0	Used as port (other than segment output)
1	Used as segment output

<R>

Note Be sure to set "1" for 80-pin products.

22.1 Functions of DTC

Incorrect:

The data transfer controller (DTC) is a function that transfers data between memories without using the CPU. The DTC is activated by a peripheral function interrupt to perform data transfers. The DTC and CPU use the same bus, and the DTC takes priority over the CPU in using the bus.

Table 22-1 lists the DTC specifications.

Table 22-1. DTC Specifications

Item		Specification
Activation sources		40 sources
Allocatable control data		24 sets
Address space which can be transferred	Address space	64 Kbytes (F0000H to FFFFFH), excluding general-purpose registers
	Sources	Special function register (SFR), RAM area (excluding general-purpose registers), mirror area[Note], extended special function register (2nd SFR)
	Destinations	Special function register (SFR), RAM area (excluding general-purpose registers), extended special function register (2nd SFR)
Maximum number of transfers	Normal mode	256 times
	Repeat mode	255 times
Maximum size of block to be transferred	Normal mode (8-bit transfer)	256 bytes
	Normal mode (16-bit transfer)	512 bytes
	Repeat mode	255 bytes
Unit of transfers		8 bits/16 bits
Transfer mode	Normal mode	Transfers end on completion of the transfer causing the DTCCTj register value to change from 1 to 0.
	Repeat mode	On completion of the transfer causing the DTCCTj register value to change from 1 to 0, the repeat area address is initialized and the DTRLDj register value is reloaded to the DTCCTj register to continue transfers.
Address control	Normal mode	Fixed or incremented
	Repeat mode	Addresses of the area not selected as the repeat area are fixed or incremented.
Priority of activation sources		See **Table 22-4 DTC Activation Sources and Vector Addresses**.
Interrupt request	Normal mode	When the data transfer causing the DTCCTj register value to change from 1 to 0 is performed, the activation source interrupt request is generated for the CPU, and interrupt handling is performed on completion of the data transfer.
	Repeat mode	When the data transfer causing the DTCCTj register value to change from 1 to 0 is performed while the RPTINT bit in the DTCCRj register is 1 (interrupt generation enabled), the activation source interrupt request is generated for the CPU, and interrupt handling is performed on completion of the transfer.
Transfer start		When bits DTCENi0 to DTCENi7 in the DTCENi registers are 1 (activation enabled), data transfer is started each time the corresponding DTC activation sources are generated.
Transfer stop	Normal mode	• When bits DTCENi0 to DTCENi7 are set to 0 (activation disabled). • When the data transfer causing the DTCCTj register value to change from 1 to 0 is completed.
	Repeat mode	• When bits DTCENi0 to DTCENi7 are set to 0 (activation disabled). • When the data transfer causing the DTCCTj register value to change from 1 to 0 is completed while the RPTINT bit is 1 (interrupt generation enabled).

Note In the HALT and SNOOZE modes, these areas cannot be set as the sources for DTC transfer since the flash memory is stopped.

Remark i = 0 to 3, j = 0 to 23

22.1 Functions of DTC

Correct:

The data transfer controller (DTC) is a function that transfers data between memories without using the CPU. The DTC is activated by a peripheral function interrupt to perform data transfers. The DTC and CPU use the same bus, and the DTC takes priority over the CPU in using the bus.

Table 22-1 lists the DTC specifications.

Table 22-1. DTC Specifications

<R>

Item		Specification
Activation sources		30 sources
Allocatable control data		24 sets
Address space which can be transferred	Address space	64 Kbytes (F0000H to FFFFFH), excluding general-purpose registers
	Sources	Special function register (SFR), RAM area (excluding general-purpose registers), mirror area[Note], extended special function register (2nd SFR)
	Destinations	Special function register (SFR), RAM area (excluding general-purpose registers), extended special function register (2nd SFR)
Maximum number of transfers	Normal mode	256 times
	Repeat mode	255 times
Maximum size of block to be transferred	Normal mode (8-bit transfer)	256 bytes
	Normal mode (16-bit transfer)	512 bytes
	Repeat mode	255 bytes
Unit of transfers		8 bits/16 bits
Transfer mode	Normal mode	Transfers end on completion of the transfer causing the DTCCTj register value to change from 1 to 0.
	Repeat mode	On completion of the transfer causing the DTCCTj register value to change from 1 to 0, the repeat area address is initialized and the DTRLDj register value is reloaded to the DTCCTj register to continue transfers.
Address control	Normal mode	Fixed or incremented
	Repeat mode	Addresses of the area not selected as the repeat area are fixed or incremented.
Priority of activation sources		See **Table 22-5 DTC Activation Sources and Vector Addresses**.
Interrupt request	Normal mode	When the data transfer causing the DTCCTj register value to change from 1 to 0 is performed, the activation source interrupt request is generated for the CPU, and interrupt handling is performed on completion of the data transfer.
	Repeat mode	When the data transfer causing the DTCCTj register value to change from 1 to 0 is performed while the RPTINT bit in the DTCCRj register is 1 (interrupt generation enabled), the activation source interrupt request is generated for the CPU, and interrupt handling is performed on completion of the transfer.
Transfer start		When bits DTCENi0 to DTCENi7 in the DTCENi registers are 1 (activation enabled), data transfer is started each time the corresponding DTC activation sources are generated.
Transfer stop	Normal mode	• When bits DTCENi0 to DTCENi7 are set to 0 (activation disabled). • When the data transfer causing the DTCCTj register value to change from 1 to 0 is completed.
	Repeat mode	• When bits DTCENi0 to DTCENi7 are set to 0 (activation disabled). • When the data transfer causing the DTCCTj register value to change from 1 to 0 is completed while the RPTINT bit is 1 (interrupt generation enabled).

Note In the HALT and SNOOZE modes, these areas cannot be set as the sources for DTC transfer since the flash memory is stopped.

Remark i = 0 to 3, j = 0 to 23

22.4.2 Normal mode

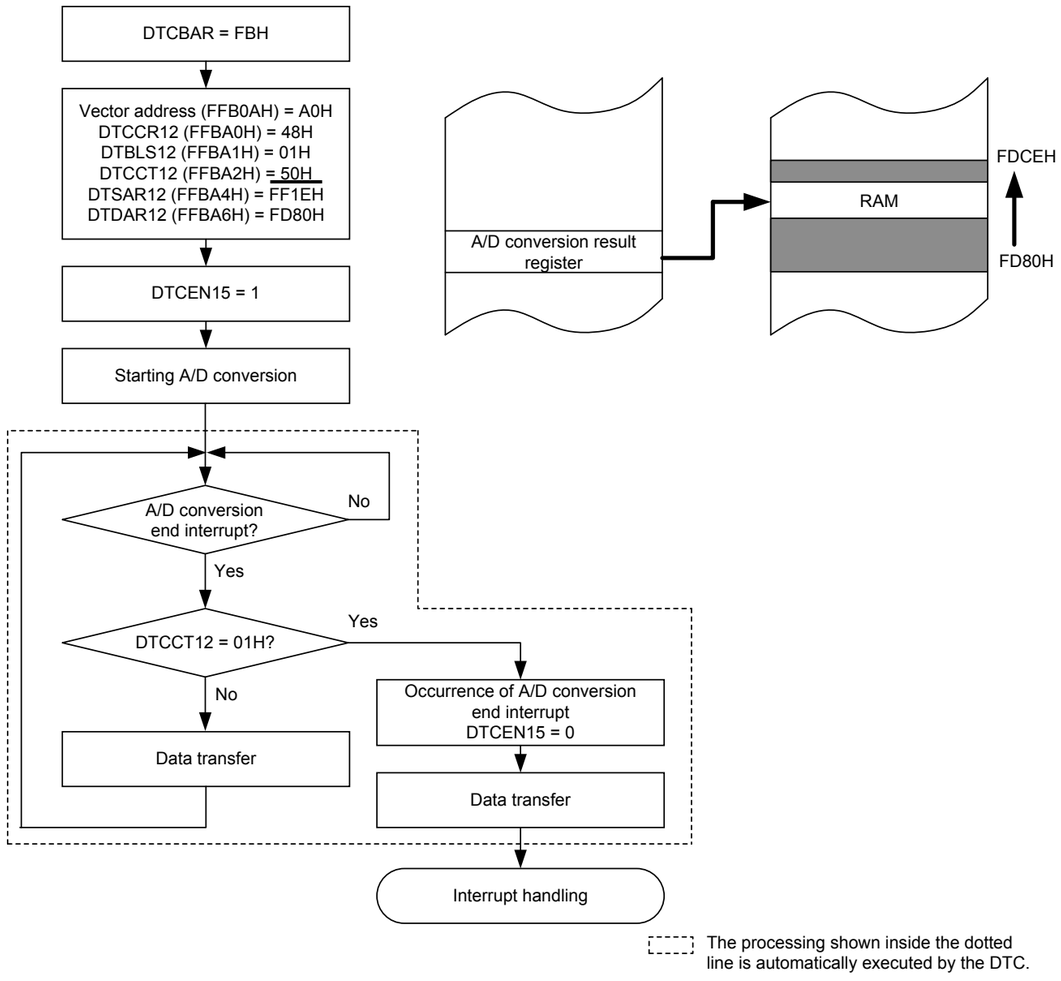
Incorrect:

(1) Example 1 of using normal mode: Consecutively capturing A/D conversion results

The DTC is activated by an A/D conversion end interrupt and the value of the A/D conversion result register is transferred to RAM.

- The vector address is FFB0AH and control data is allocated at FFBA0H to FFBA7H
- Transfers 2-byte data of the A/D conversion result register (FFF1EH, FFF1FH) to 80 bytes of FFD80H to FFDCFH of RAM

Figure 22-15. Example 1 of Using Normal Mode: Consecutively Capturing A/D Conversion Results



The value of the DTRLD12 register is not used because of normal mode, but initialize the register to 00H when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.

22.4.2 Normal mode

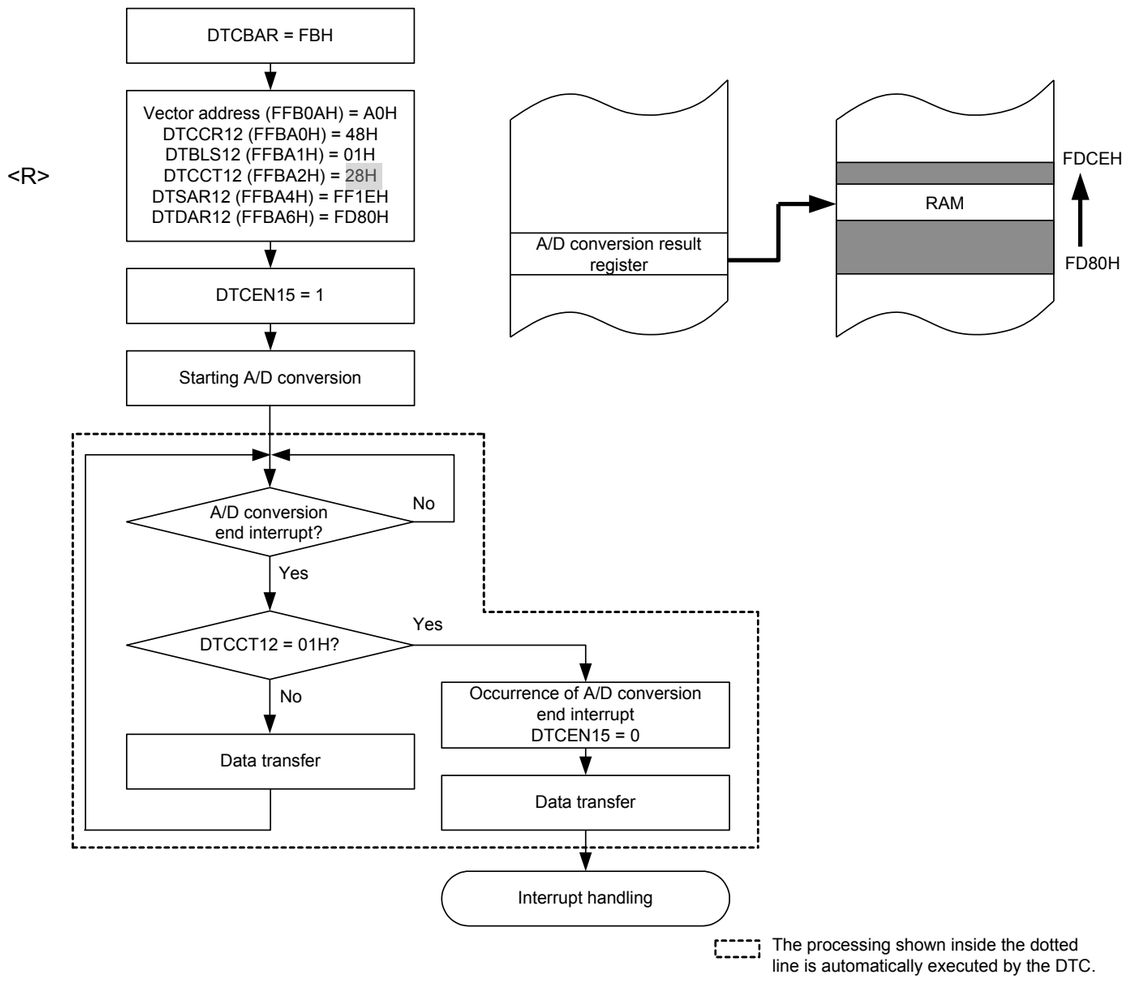
Correct:

(1) Example 1 of using normal mode: Consecutively capturing A/D conversion results

The DTC is activated by an A/D conversion end interrupt and the value of the A/D conversion result register is transferred to RAM.

- The vector address is FFB0AH and control data is allocated at FFBA0H to FFBA7H
- <R> Transfers 2-byte data of the A/D conversion result register (FFF1EH, FFF1FH) to 80 bytes of FFD80H to FFDCFH of RAM 40 times

Figure 22-15. Example 1 of Using Normal Mode: Consecutively Capturing A/D Conversion Results



The value of the DTRLD12 register is not used because of normal mode, but initialize the register to 00H when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.

22.5.3 DTC pending instruction

Incorrect:

Even if a DTC transfer request is generated, data transfer is held pending immediately after the following instructions. Also, the DTC is not activated between PREFIX instruction code and the instruction immediately after that code.

- Call/return instruction
- Unconditional branch instruction
- Conditional branch instruction
- Read access instruction for code flash memory
- Bit manipulation instructions for IFxx, MKxx, PRxx, and PSW, and an 8-bit manipulation instruction that has the ES register as operand

Cautions 1. When a DTC transfer request is acknowledged, all interrupt requests are held pending until DTC transfer is completed.

2. While the DTC is held pending by the DTC pending instruction, all interrupt requests are held pending.

22.5.3 DTC pending instruction

Correct:

Even if a DTC transfer request is generated, data transfer is held pending immediately after the following instructions. Also, the DTC is not activated between PREFIX instruction code and the instruction immediately after that code.

- Call/return instruction
- Unconditional branch instruction
- Conditional branch instruction
- Read access instruction for code flash memory
- Bit manipulation instructions for IFxx, MKxx, PRxx, and PSW, and an 8-bit manipulation instruction that has the ES register as operand

<R> • Instruction of Multiply, Divide, Multiply & Accumulate (excluding MULU)

Cautions 1. When a DTC transfer request is acknowledged, all interrupt requests are held pending until DTC transfer is completed.

2. While the DTC is held pending by the DTC pending instruction, all interrupt requests are held pending.

23.4.4 Interrupt request hold

Incorrect:

<R> 23.4.4 Interrupt servicing during division instruction

Correct:

<R> 23.4.4 Interrupt servicing during division instruction

The RL78/I1B handles interrupts during the DIVHU/DIVWU instruction in order to enhance the interrupt response when a division instruction is executed.

- When an interrupt is generated while the DIVHU/DIVWU instruction is executed, the instruction is suspended
- After the instruction is suspended, the PC indicates the next instruction after DIVHU/DIVWU
- An interrupt is generated by the next instruction
- PC-3 is stacked to execute the DIVHU/DIVWU instruction again

Normal interrupt	Interrupts while Executing DIVHU/DIVWU Instruction
(SP-1) ← PSW	(SP-1) ← PSW
(SP-2) ← (PC)S	(SP-2) ← (PC-3)S
(SP-3) ← (PC)H	(SP-3) ← (PC-3)H
(SP-4) ← (PC)L	(SP-4) ← (PC-3)L
PCS ← 0000	PCS ← 0000
PCH ← (Vector)	PCH ← (Vector)
PCL ← (Vector)	PCL ← (Vector)
SP ← SP-4	SP ← SP-4
IE ← 0	IE ← 0

The AX, BC, DE, and HL registers are used for DIVHU/DIVWU. Use these registers by stacking them for interrupt servicing.

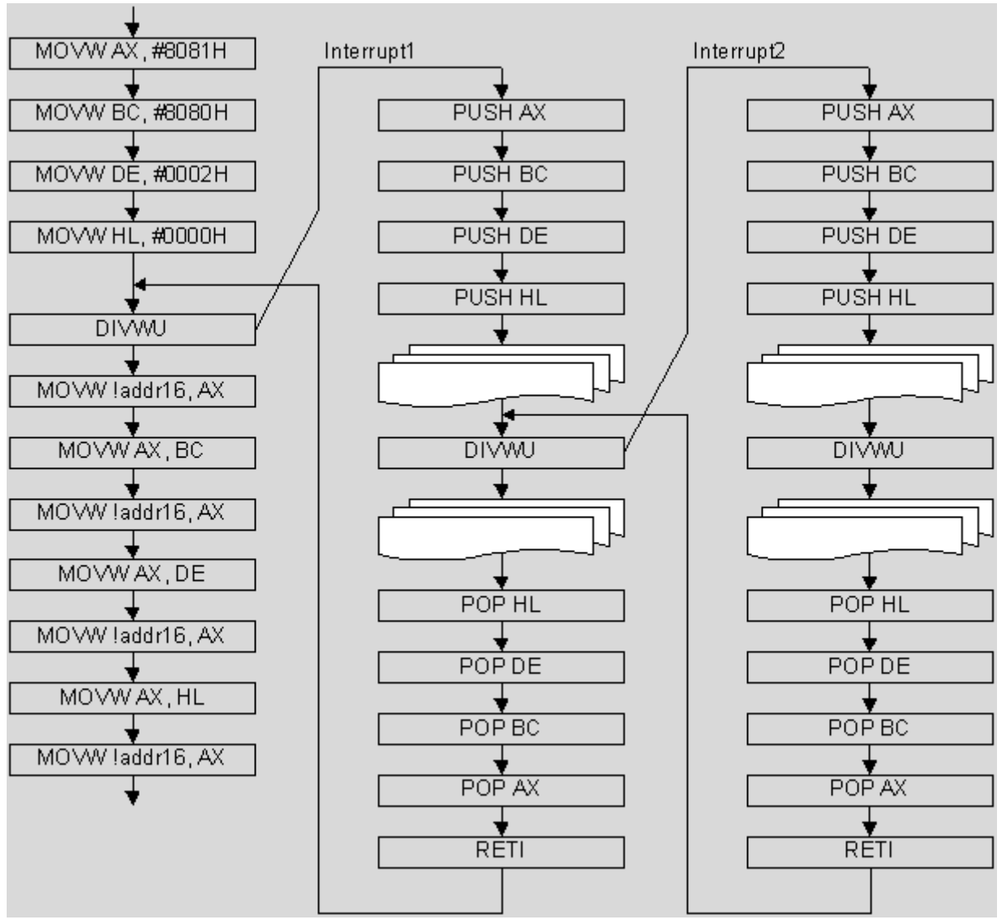
23.4.4 Interrupt request hold

Incorrect:

<R> 23.4.4 Interrupt servicing during division instruction

Correct:

<R>



Caution Disable interrupts when executing the DIVHU or DIVWU instruction in an interrupt servicing routine.

Alternatively, unless they are executed in the RAM area, note that execution of a DIVHU or DIVWU instruction is possible even with interrupts enabled as long as a NOP instruction is added immediately after the DIVHU or DIVWU instruction in the assembly language source code. The following compilers automatically add a NOP instruction immediately after any DIVHU or DIVWU instruction output during the build process.

- V. 1.71 and later versions of the CA78K0R (Renesas Electronics compiler), for both C and assembly language source code

- Service pack 1.40.6 and later versions of the EWRL78 (IAR compiler), for C language source code

- GNURL78 (KPIT compiler), for C language source code

23.4.4 Interrupt request hold

Incorrect:

There are instructions where, even if an interrupt request is issued while the instructions are being executed, interrupt request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- BTCLR PSW. bit, $addr20
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- SKH
- SKNH
- Write instructions for the IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, and PR13L registers

23.4.5 Interrupt request hold

Correct:

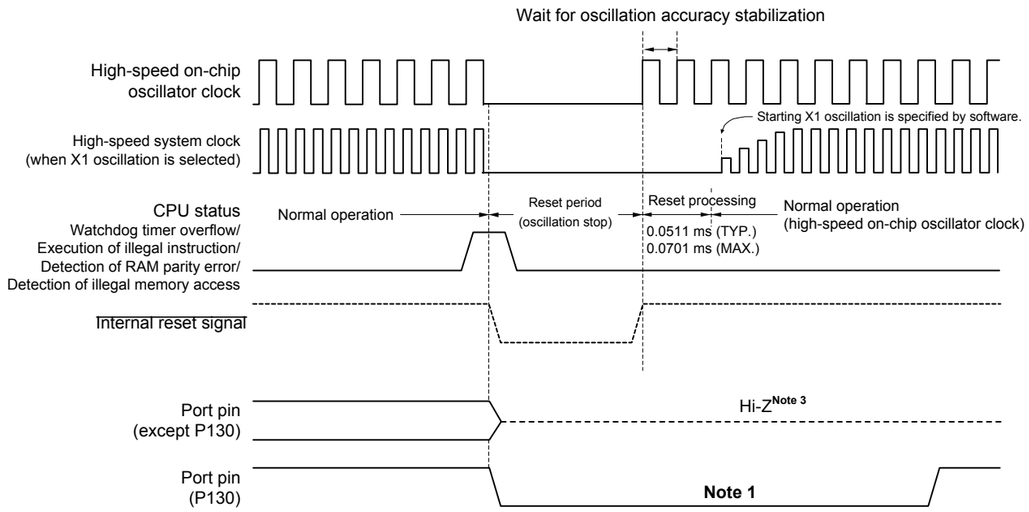
There are instructions where, even if an interrupt request is issued while the instructions are being executed, interrupt request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- BTCLR PSW. bit, $addr20
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- SKH
- SKNH
- <R> MULHU
- <R> MULH
- <R> MACHU
- <R> MACH
- Write instructions for the IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, and PR13L registers

25.1 Timing of Reset Operation

Incorrect:

Figure 25-3. Timing of Reset Due to Watchdog Timer Overflow, Execution of Illegal Instruction, Detection of RAM Parity Error, or Detection of Illegal Memory



Notes 1. When P130 is set to high-level output before reset is effected, the output signal of P130 can be dummy-output as a reset signal to an external device, because P130 outputs a low level when reset is effected. To release a reset signal to an external device, set P130 to high-level output by software.

2. Reset times (times for release from the external reset state)

After the first release of the POR: 0.672 ms (typ.), 0.832 ms (max.) when the LVD is in use.
0.399 ms (typ.), 0.519 ms (max.) when the LVD is off.

After the second release of the POR: 0.531 ms (typ.), 0.675 ms (max.) when the LVD is in use.
0.259 ms (typ.), 0.362 ms (max.) when the LVD is off.

After power is supplied, a voltage stabilization waiting time of about 0.99 ms (typ.) and up to 2.30 ms (max.) is required before reset processing starts after release of the external reset.

3. The state of P40 is as follows.

- High-impedance during the external reset period or reset period by the POR.
- High level during other types of reset or after receiving a reset signal (connected to the internal pull-up resistor).

Caution A watchdog timer internal reset resets the watchdog timer.

Reset by POR and LVD circuit supply voltage detection is automatically released when internal $V_{DD} \geq V_{POR}$ or internal $V_{DD} \geq V_{LVD}$ after the reset. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts.

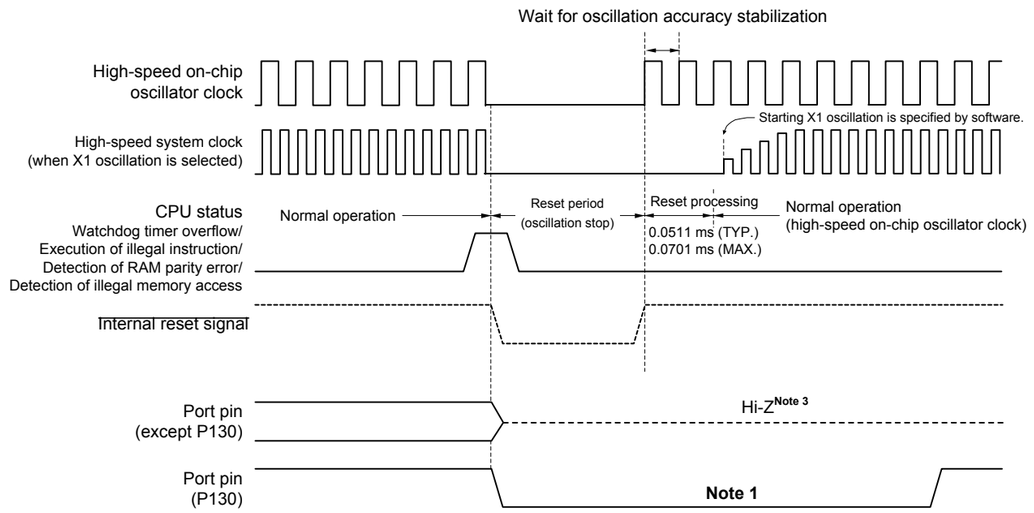
For details, see **CHAPTER 26 POWER-ON-RESET CIRCUIT** or **CHAPTER 27 VOLTAGE DETECTOR**.

Remark V_{POR}: POR power supply rise detection voltage
V_{LVD}: LVD detection voltage

25.1 Timing of Reset Operation

Correct:

Figure 25-3. Timing of Reset Due to Watchdog Timer Overflow, Execution of Illegal Instruction, Detection of RAM Parity Error, or Detection of Illegal Memory



Notes 1. When P130 is set to high-level output before reset is effected, the output signal of P130 can be dummy-output as a reset signal to an external device, because P130 outputs a low level when reset is effected. To release a reset signal to an external device, set P130 to high-level output by software.

2. Reset times (times for release from the external reset state)

After the first release of the POR: 0.672 ms (typ.), 0.832 ms (max.) when the LVD is in use.
0.399 ms (typ.), 0.519 ms (max.) when the LVD is off.

After the second release of the POR: 0.531 ms (typ.), 0.675 ms (max.) when the LVD is in use.
0.259 ms (typ.), 0.362 ms (max.) when the LVD is off.

After power is supplied, a voltage stabilization waiting time of about 0.99 ms (typ.) and up to 2.30 ms (max.) is required before reset processing starts after release of the external reset.

3. The state of P40 is as follows.

- High-impedance during the external reset period or reset period by the POR.
- High level during other types of reset or after receiving a reset signal (connected to the internal pull-up resistor).

<R>

Reset by POR and LVD circuit supply voltage detection is automatically released when internal $V_{DD} \geq V_{POR}$ or internal $V_{DD} \geq V_{LVD}$ after the reset. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts.

For details, see **CHAPTER 26 POWER-ON-RESET CIRCUIT** or **CHAPTER 27 VOLTAGE DETECTOR**.

Remark V_{POR}: POR power supply rise detection voltage
V_{LVD}: LVD detection voltage

26.3 Operation of Power-on-reset Circuit

Incorrect:

Notes 3. After the interrupt request signal (INTLVI) is generated, the LVIL and LVIMD bits of the voltage detection level register (LVIS) are automatically set to 1. After INTLVI is generated, appropriate settings should be made according to **Figure 27-8 Setting Procedure for Operating Voltage Check/Reset** and **Figure 27-9 Initial Setting of Interrupt and Reset Mode**, taking into consideration that the supply voltage might return to the high voltage detection level (V_{LVDH}) or higher without falling below the low voltage detection level (V_{LVDL}).

26.3 Operation of Power-on-reset Circuit

Correct:

<R> **Notes 3.** After the interrupt request signal (INTLVI) is generated, the LVILV and LVIMD bits of the voltage detection level register (LVIS) are automatically set to 1. After INTLVI is generated, appropriate settings should be made according to **Figure 27-8 Setting Procedure for Operating Voltage Check/Reset** and **Figure 27-9 Initial Setting of Interrupt and Reset Mode**, taking into consideration that the supply voltage might return to the high voltage detection level (VLVDH) or higher without falling below the low voltage detection level (V_{LVDL}).

27.1 Functions of Voltage Detector

Incorrect:

The operation mode and detection voltages (V_{LVDH}, V_{LVDL}, V_{LVD}) for the voltage detector is set by using the option byte (000C1H).

The voltage detector (LVD) has the following functions.

- The LVD circuit compares the internal power supply voltage (internal V_{DD}) that supplied from the V_{DD} or VBAT pin with the detection voltage (V_{LVDH}, V_{LVDL}, V_{LVD}), and generates an internal reset or internal interrupt signal.
- The detection level for the internal power supply detection voltage (V_{LVDH}, V_{LVDL}, V_{LVD}) can be selected by using the option byte as one of 11 levels (for details, see **CHAPTER 32 OPTION BYTE**).
- Operable in STOP mode.
- After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **37.4 AC Characteristics**. This is done by utilizing the voltage detection circuit or controlling the externally input reset signal. After the power supply is turned off, this LSI should be placed in the STOP mode, or placed in the reset state by utilizing the voltage detection circuit or controlling the externally input reset signal before the voltage falls below the operating range. The range of operating voltage varies with the setting of the user option byte (000C2H or 010C2H).

27.1 Functions of Voltage Detector

Correct:

The operation mode and detection voltages (V_{LVDH}, V_{LVDL}, V_{LVD}) for the voltage detector is set by using the option byte (000C1H).

The voltage detector (LVD) has the following functions.

- Te LVD circuit compares the internal power supply voltage (internal V_{DD}) that supplied from the V_{DD} or VBAT pin with the detection voltage (V_{LVDH}, V_{LVDL}, V_{LVD}), and generates an internal reset or internal interrupt signal.
- The detection level for the internal power supply detection voltage (V_{LVDH}, V_{LVDL}, V_{LVD}) can be selected by using the option byte as one of 11 levels (for details, see **CHAPTER 32 OPTION BYTE**).
- Operable in STOP mode.
- <R> After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **37.4 AC Characteristics**. This is done by utilizing the voltage detector or controlling the externally input reset signal. After the power supply is turned off, this LSI should be placed in the STOP mode, or placed in the reset state by utilizing the voltage detection circuit or controlling the externally input reset signal before the voltage falls below the operating range. The range of operating voltage varies with the setting of the user option byte (000C2H or 010C2H).

27.1 Functions of Voltage Detector

Incorrect:

The reset and internal interrupt signals are generated in each mode as follows.

Interrupt & Reset Mode (LVIMDS1, LVIMDS0 = 1, 0)	Reset Mode (LVIMDS1, LVIMDS0 = 1, 1)	Interrupt Mode (LVIMDS1, LVIMDS0 = 0, 1)
Generates an interrupt request signal by detecting internal power supply voltage (internal V_{DD}) < V_{LVDH} when the operating voltage falls, and an internal reset by detecting internal power supply voltage (internal V_{DD}) < V_{LVDL}. Releases an internal reset by detecting internal power supply voltage (internal V_{DD}) ≥ V_{LVDH}.	Releases an internal reset by detecting internal power supply voltage (internal V_{DD}) ≥ V_{LVD}. Generates an interrupt request signal by detecting internal power supply voltage (internal V_{DD}) < V_{LVD}.	The state of an internal reset by LVD is retained until internal V_{DD} ≥ V_{LVD} immediately after reset generation. The internal reset is released when internal V_{DD} ≥ V_{LVD} is detected. After that, an interrupt request signal (INTLVI) is generated when internal V_{DD} < V_{LVD} or internal V_{DD} ≥ V_{LVD} is detected.

While the voltage detector is operating, whether the internal supply voltage or the input voltage from an external input pin is more than or less than the detection level can be checked by reading the voltage detection flag (LVIF: bit 0 of the voltage detection register (LVIM)).

Bit 0 (LVIRF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of the RESF register, see **CHAPTER 25 RESET FUNCTION**.

27.1 Functions of Voltage Detector

Correct:

The reset and internal interrupt signals are generated in each mode as follows.

<R>

Interrupt & Reset Mode (LVIMDS1, LVIMDS0 = 1, 0)	Reset Mode (LVIMDS1, LVIMDS0 = 1, 1)	Interrupt Mode (LVIMDS1, LVIMDS0 = 0, 1)
Generates an interrupt request signal by detecting internal power supply voltage (internal V_{DD}) < V_{LVDH} when the operating voltage falls, and an internal reset by detecting internal power supply voltage (internal V_{DD}) < V_{LVDL}. Releases an internal reset by detecting internal power supply voltage (internal V_{DD}) ≥ V_{LVDH}.	Releases an internal reset by detecting internal power supply voltage (internal V_{DD}) ≥ V_{LVD}. Generates an internal reset by detecting internal power supply voltage (internal V_{DD}) < V_{LVD}.	The state of an internal reset by LVD is retained until internal V_{DD} ≥ V_{LVD} immediately after reset generation. The internal reset is released when internal V_{DD} ≥ V_{LVD} is detected. After that, an interrupt request signal (INTLVI) is generated when internal V_{DD} < V_{LVD} or internal V_{DD} ≥ V_{LVD} is detected.

While the voltage detector is operating, whether the internal supply voltage or the input voltage from an external input pin is more than or less than the detection level can be checked by reading the voltage detection flag (LVIF: bit 0 of the voltage detection register (LVIM)).

Bit 0 (LVIRF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of the RESF register, see **CHAPTER 25 RESET FUNCTION**.

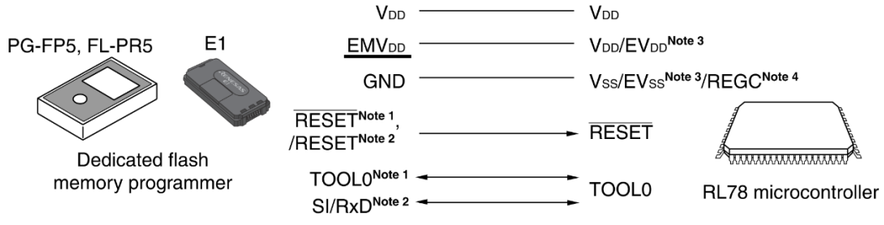
33.1.2 Communication mode

Incorrect:

Communication between the dedicated flash memory programmer and the RL78 microcontroller is established by serial communication using the TOOL0 pin via a dedicated single-line UART of the RL78 microcontroller.

Transfer rate: 1 M, 500 k, 250 k, 115.2 kbps

Figure 33-2. Communication with Dedicated Flash Memory Programmer



Notes 1. When using E1 on-chip debugging emulator.

2. When using PG-FP5 or FL-PR5.

3. 100-pin products only.

4. Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

33.1.2 Communication mode

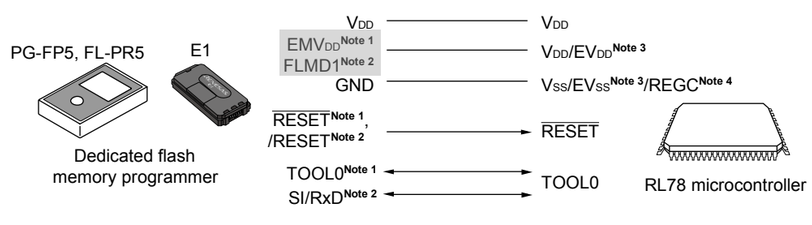
Correct:

Communication between the dedicated flash memory programmer and the RL78 microcontroller is established by serial communication using the TOOL0 pin via a dedicated single-line UART of the RL78 microcontroller.

Transfer rate: 1 M, 500 k, 250 k, 115.2 kbps

Figure 33-2. Communication with Dedicated Flash Memory Programmer

<R>



Notes 1. When using E1 on-chip debugging emulator.

2. When using PG-FP5 or FL-PR5.

3. 100-pin products only.

4. Connect REGC pin to ground via a capacitor (0.47 to 1 µF).

33.5 Self-Programming

Incorrect:

The RL78 microcontroller supports a self-programming function that can be used to rewrite the code flash memory via a user program. Because this function allows a user application to rewrite the code flash memory by using the flash self-programming library, it can be used to upgrade the program in the field.

Cautions 1. The self-programming function cannot be used when the CPU operates with the subsystem clock.
2. To prohibit an interrupt during self-programming, in the same way as in the normal operation mode, execute the self-programming library in the state where the IE flag is cleared (0) by the DI instruction. To enable an interrupt, clear (0) the interrupt mask flag to accept in the state where the IE flag is set (1) by the EI instruction, and then execute the self-programming library.
3. The high-speed on-chip oscillator needs to oscillate during self-programming. When stopping the high-speed on-chip oscillator, oscillate the high-speed on-chip oscillator clock (HIOSTOP = 0) and execute the self-programming library after 30 μs elapses.
4. The self-programming function cannot be used when the internal power is supplied from the VBAT pin.

Remarks 1. For details of the self-programming function, refer to **RL78 Microcontroller Flash Self Programming Library Type01 User's Manual (R01AN0350).**
2. For details of the time required to execute self programming, see the notes on use that accompany the flash self programming library tool.

The self-programming function has two flash memory programming modes; wide voltage mode and full speed mode.

Specify the mode that corresponds to the flash operation mode specified in bits CMODE1 and CMODE0 in option byte 000C2H.
Set to full speed mode when the HS (high speed main) mode is specified. Set to wide voltage mode when the LS (low speed main) mode is specified.
If the argument fsl_flash_voltage_u08 is 00H when the FSL_Init function of the flash self-programming library provided by Renesas Electronics is executed, full speed mode is specified. If the argument is other than 00H, the wide voltage mode is specified.

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

33.5 Self-Programming

Correct:

The RL78 microcontroller supports a self-programming function that can be used to rewrite the code flash memory via a user program. Because this function allows a user application to rewrite the code flash memory by using the flash self-programming library, it can be used to upgrade the program in the field.

Cautions 1. The self-programming function cannot be used when the CPU operates with the subsystem clock.
2. To prohibit an interrupt during self-programming, in the same way as in the normal operation mode, execute the self-programming library in the state where the IE flag is cleared (0) by the DI instruction. To enable an interrupt, clear (0) the interrupt mask flag to accept in the state where the IE flag is set (1) by the EI instruction, and then execute the self-programming library.
3. The high-speed on-chip oscillator needs to oscillate during self-programming. When stopping the high-speed on-chip oscillator, oscillate the high-speed on-chip oscillator clock (HIOSTOP = 0) and execute the self-programming library after 30 μs elapses.
4. The self-programming function cannot be used when the internal power is supplied from the VBAT pin.

<R> **Remarks 1.** For details of the self-programming function, refer to **RL78 Microcontroller Flash Self Programming Library Type01 User's Manual (R01US0050).**
2. For details of the time required to execute self programming, see the notes on use that accompany the flash self programming library tool.

The self-programming function has two flash memory programming modes; wide voltage mode and full speed mode.

Specify the mode that corresponds to the flash operation mode specified in bits CMODE1 and CMODE0 in option byte 000C2H.
Set to full speed mode when the HS (high speed main) mode is specified. Set to wide voltage mode when the LS (low speed main) mode is specified.
If the argument fsl_flash_voltage_u08 is 00H when the FSL_Init function of the flash self-programming library provided by Renesas Electronics is executed, full speed mode is specified. If the argument is other than 00H, the wide voltage mode is specified.

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

36. 2 Operation List

Incorrect:

Table 36-5. Operation List (12/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Multiply, Divide, Multiply & accumu-late	MULU	X	1	1	–	AX ← A × X			
	MULHU		3	2	–	BCAX ← AX × BC (unsigned)			
	MULH		3	2	–	BCAX ← AX × BC (signed)			
	DIVHU		3	9	–	AX (quotient), DE (remainder) ← AX ÷ DE (unsigned)			
	DIVWU		3	17	–	BCAX (quotient), HLDE (remainder) ← BCAX ÷ HLDE (unsigned)			
	MACHU		3	3	–	MACR ← MACR + AX × BC (unsigned)		×	×
	MACH		3	3	–	MACR ← MACR + AX × BC(signed)		×	×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash area is accessed.

Remarks 1. Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

2. MACR indicates the multiplication and accumulation register (MACRH, MACRL).

Correct:

Table 36-5. Operation List (12/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Multiply, Divide, Multiply & accumu-late	MULU	X	1	1	–	AX ← A × X			
	MULHU		3	2	–	BCAX ← AX × BC (unsigned)			
	MULH		3	2	–	BCAX ← AX × BC (signed)			
	DIVHU		3	9	–	AX (quotient), DE (remainder) ← AX ÷ DE (unsigned)			
	DIVWU		3	17	–	BCAX (quotient), HLDE (remainder) ← BCAX ÷ HLDE (unsigned)			
	MACHU		3	3	–	MACR ← MACR + AX × BC (unsigned)		×	×
	MACH		3	3	–	MACR ← MACR + AX × BC(signed)		×	×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash area is accessed.

<R> **Caution Disable interrupts when executing the DIVHU or DIVWU instruction in an interrupt servicing routine. Alternatively, unless they are executed in the RAM area, note that execution of a DIVHU or DIVWU instruction is possible even with interrupts enabled as long as a NOP instruction is added immediately after the DIVHU or DIVWU instruction in the assembly language source code. The following compilers automatically add a NOP instruction immediately after any DIVHU or DIVWU instruction output during the build process.**

- V. 1.71 and later versions of the CA78K0R (Renesas Electronics compiler), for both C and assembly language source code

- Service pack 1.40.6 and later versions of the EWRL78 (IAR compiler), for C language source code

- GNURL78 (KPIT compiler), for C language source code

Remarks 1. Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

2. MACR indicates the multiplication and accumulation register (MACRH, MACRL).

37.3.1 Pin characteristics

Incorrect:

(TA = −40 to +85°C, 1.9 V≤VDD = EVDD ≤ 5.5 V, VSS = EVSS = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P00 to P07, P10 to P17, P30 to P37, P40 to P44, P60 to P62, P70 to P77, P80 to P85, P125 to P127	$V_I = EV_{DD}$				1	μA
	I_{LIH2}	P20 to P25, P137, $\overline{\text{RESET}}$	$V_I = V_{DD}$Note				1	μA
	I_{LIH3}	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	$V_I = V_{DD}$Note	In input port or external clock input			1	μA
				In resonator connection			10	μA
Input leakage current, low	I_{LIL1}	P00 to P07, P10 to P17, P30 to P37, P40 to P44, P60 to P62, P70 to P77, P80 to P85, P125 to P127	$V_I = EV_{SS}$				−1	μA
	I_{LIL2}	P20 to P25, P137, $\overline{\text{RESET}}$	$V_I = V_{SS}$				−1	μA
	I_{LIL3}	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	$V_I = V_{SS}$	In input port or external clock input			−1	μA
				In resonator connection			−10	μA
On-chip pull-up resistance	R_{U1}	P10 to P17, P30 to P37, P50 to P57, P70 to P77, P80 to P85, P125 to P127	$V_I = V_{SS}$	2.4 V ≤ V_{DD}Note ≤ 5.5 V	10	20	100	kΩ
				1.9 V ≤ V_{DD}Note ≤ 5.5 V	10	30	100	kΩ
	R_{U2}	P00 to P07, P40 to P44	$V_I = V_{SS}$		10	20	100	kΩ

Note The power supply voltage (VBAT pin or V_{DD} pin) selected by the battery backup feature.

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

Correct:

(TA = −40 to +85°C, 1.9 V≤VDD = EVDD ≤ 5.5 V, VSS = EVSS = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P00 to P07, P10 to P17, P30 to P37, P40 to P44, P60 to P62, P70 to P77, P80 to P85, P125 to P127	$V_I = EV_{DD}$				1	μA
	I_{LIH2}	P20 to P25, P137, $\overline{\text{RESET}}$	$V_I = V_{DD}$Note				1	μA
	I_{LIH3}	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	$V_I = V_{DD}$Note	In input port or external clock input			1	μA
				In resonator connection			10	μA
Input leakage current, low	I_{LIL1}	P00 to P07, P10 to P17, P30 to P37, P40 to P44, P60 to P62, P70 to P77, P80 to P85, P125 to P127	$V_I = EV_{SS}$				−1	μA
	I_{LIL2}	P20 to P25, P137, $\overline{\text{RESET}}$	$V_I = V_{SS}$				−1	μA
	I_{LIL3}	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	$V_I = V_{SS}$	In input port or external clock input			−1	μA
				In resonator connection			−10	μA
<R> On-chip pull-up resistance	R_{U1}	P10 to P17, P30 to P37, P50 to P57, P70 to P77, P80 to P85, P125 to P127	$V_I = V_{SS}$	2.4 V ≤ EV_{DD} ≤ 5.5 V	10	20	100	kΩ
<R>				1.9 V ≤ EV_{DD} ≤ 5.5 V	10	30	100	kΩ
	R_{U2}	P00 to P07, P40 to P44	$V_I = V_{SS}$		10	20	100	kΩ

Note The power supply voltage (VBAT pin or V_{DD} pin) selected by the battery backup feature.

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

37.3.2 Supply current characteristics

Incorrect:

(T_A = –40 to +85°C, 1.9 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V) **(1/4)**

Parameter	Symbol	Conditions						MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD1}	Operating mode	HS (high-speed main) mode[Note 5]	f_{IH} = 24 MHz[Note 3]	Basic operation	V_{DD} = 5.0 V			2.3		mA
						V_{DD} = 3.0 V			2.3		mA
					Normal operation	V_{DD} = 5.0 V			4.1	6.6	mA
						V_{DD} = 3.0 V			4.1	6.6	mA
				f_{IH} = 12 MHz[Note 3]	Normal operation	V_{DD} = 5.0 V			2.5	3.8	mA
						V_{DD} = 3.0 V			2.5	3.8	mA
				f_{IH} = 6 MHz[Note 3]	Normal operation	V_{DD} = 5.0 V			1.6	2.5	mA
						V_{DD} = 3.0 V			1.6	2.5	mA
				f_{IH} = 3 MHz[Note 3]	Normal operation	V_{DD} = 5.0 V			1.2	1.9	mA
						V_{DD} = 3.0 V			1.2	1.9	mA
			LS (low-speed main) mode[Note 5]	f_{IH} = 6 MHz[Note 3]	Normal operation	V_{DD} = 3.0 V			1.3	2.1	mA
						V_{DD} = 2.0 V			1.3	2.1	mA
				f_{IH} = 3 MHz[Note 3]	Normal operation	V_{DD} = 3.0 V			0.9	1.5	mA
						V_{DD} = 2.0 V			0.9	1.5	mA
			HS (high-speed main) mode[Note 5]	f_{MX} = 20 MHz[Note 2], V_{DD} = 5.0 V	Normal operation	Square wave input			3.4	5.5	mA
						Resonator connection			3.6	5.7	mA
				f_{MX} = 20 MHz[Note 2], V_{DD} = 3.0 V	Normal operation	Square wave input			3.4	5.5	mA
						Resonator connection			3.6	5.7	mA
				f_{MX} = 16 MHz[Note 2], V_{DD} = 5.0 V	Normal operation	Square wave input			2.8	4.4	mA
						Resonator connection			2.9	4.6	mA
				f_{MX} = 16 MHz[Note 2], V_{DD} = 3.0 V	Normal operation	Square wave input			2.8	4.4	mA
						Resonator connection			2.9	4.6	mA
				f_{MX} = 12 MHz[Note 2], V_{DD} = 5.0 V	Normal operation	Square wave input			2.3	3.6	mA
						Resonator connection			2.4	3.7	mA
				f_{MX} = 12 MHz[Note 2], V_{DD} = 3.0 V	Normal operation	Square wave input			2.3	3.6	mA
						Resonator connection			2.4	3.7	mA
				f_{MX} = 10 MHz[Note 2], V_{DD} = 5.0 V	Normal operation	Square wave input			2.1	3.2	mA
						Resonator connection			2.1	3.3	mA
				f_{MX} = 10 MHz[Note 2], V_{DD} = 3.0 V	Normal operation	Square wave input			2.1	3.2	mA
						Resonator connection			2.1	3.3	mA
			LS (low-speed main) mode[Note 5]	f_{MX} = 8 MHz[Note 2], V_{DD} = 3.0 V	Normal operation	Square wave input			1.2	2.0	mA
						Resonator connection			1.2	2.1	mA
				f_{MX} = 8 MHz[Note 2], V_{DD} = 2.0 V	Normal operation	Square wave input			1.2	2.0	mA
						Resonator connection			1.2	2.1	mA
			Subclock operation	f_{SUB} = 32.768kHz[Note 4], T_A = –40°C	Normal operation	Square wave input			4.8	5.9	μA
						Resonator connection			4.9	6.0	μA
				f_{SUB} = 32.768kHz[Note 4], T_A = +25°C	Normal operation	Square wave input			4.9	5.9	μA
						Resonator connection			5.0	6.0	μA
				f_{SUB} = 32.768kHz[Note 4], T_A = +50°C	Normal operation	Square wave input			4.9	7.6	μA
						Resonator connection			5.0	7.7	μA
				f_{SUB} = 32.768kHz[Note 4], T_A = +70°C	Normal operation	Square wave input			5.2	9.3	μA
						Resonator connection			5.3	9.4	μA
				f_{SUB} = 32.768kHz[Note 4], T_A = +85°C	Normal operation	Square wave input			6.1	13.3	μA
						Resonator connection			6.2	13.4	μA

(**Notes** and **Remarks** are listed on the next page.)

Correct:

(T_A = –40 to +85°C, 1.9 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V) **(1/4)**

Parameter	Symbol	Conditions						MIN.	TYP.	MAX.	Unit
<R> Supply current[Note 1]	I_{DD1}	Operating mode	HS (high-speed main) mode[Note 5]	f_{IH} = 24 MHz[Note 3]	Basic operation	V_{DD} = 5.0 V			1.5		mA
<R>						V_{DD} = 3.0 V			1.5		mA
					Normal operation	V_{DD} = 5.0 V			4.1	6.6	mA
						V_{DD} = 3.0 V			4.1	6.6	mA
				f_{IH} = 12 MHz[Note 3]	Normal operation	V_{DD} = 5.0 V			2.5	3.8	mA
						V_{DD} = 3.0 V			2.5	3.8	mA
				f_{IH} = 6 MHz[Note 3]	Normal operation	V_{DD} = 5.0 V			1.6	2.5	mA
						V_{DD} = 3.0 V			1.6	2.5	mA
				f_{IH} = 3 MHz[Note 3]	Normal operation	V_{DD} = 5.0 V			1.2	1.9	mA
						V_{DD} = 3.0 V			1.2	1.9	mA
			LS (low-speed main) mode[Note 5]	f_{IH} = 6 MHz[Note 3]	Normal operation	V_{DD} = 3.0 V			1.3	2.1	mA
						V_{DD} = 2.0 V			1.3	2.1	mA
				f_{IH} = 3 MHz[Note 3]	Normal operation	V_{DD} = 3.0 V			0.9	1.5	mA
						V_{DD} = 2.0 V			0.9	1.5	mA
			HS (high-speed main) mode[Note 5]	f_{MX} = 20 MHz[Note 2], V_{DD} = 5.0 V	Normal operation	Square wave input			3.4	5.5	mA
						Resonator connection			3.6	5.7	mA
				f_{MX} = 20 MHz[Note 2], V_{DD} = 3.0 V	Normal operation	Square wave input			3.4	5.5	mA
						Resonator connection			3.6	5.7	mA
				f_{MX} = 16 MHz[Note 2], V_{DD} = 5.0 V	Normal operation	Square wave input			2.8	4.4	mA
						Resonator connection			2.9	4.6	mA
				f_{MX} = 16 MHz[Note 2], V_{DD} = 3.0 V	Normal operation	Square wave input			2.8	4.4	mA
						Resonator connection			2.9	4.6	mA
				f_{MX} = 12 MHz[Note 2], V_{DD} = 5.0 V	Normal operation	Square wave input			2.3	3.6	mA
						Resonator connection			2.4	3.7	mA
				f_{MX} = 12 MHz[Note 2], V_{DD} = 3.0 V	Normal operation	Square wave input			2.3	3.6	mA
						Resonator connection			2.4	3.7	mA
				f_{MX} = 10 MHz[Note 2], V_{DD} = 5.0 V	Normal operation	Square wave input			2.1	3.2	mA
						Resonator connection			2.1	3.3	mA
				f_{MX} = 10 MHz[Note 2], V_{DD} = 3.0 V	Normal operation	Square wave input			2.1	3.2	mA
						Resonator connection			2.1	3.3	mA
			LS (low-speed main) mode[Note 5]	f_{MX} = 8 MHz[Note 2], V_{DD} = 3.0 V	Normal operation	Square wave input			1.2	2.0	mA
						Resonator connection			1.2	2.1	mA
				f_{MX} = 8 MHz[Note 2], V_{DD} = 2.0 V	Normal operation	Square wave input			1.2	2.0	mA
						Resonator connection			1.2	2.1	mA
			Subclock operation	f_{SUB} = 32.768 kHz[Note 4], T_A = –40°C	Normal operation	Square wave input			4.8	5.9	μA
						Resonator connection			4.9	6.0	μA
				f_{SUB} = 32.768 kHz[Note 4], T_A = +25°C	Normal operation	Square wave input			4.9	5.9	μA
						Resonator connection			5.0	6.0	μA
				f_{SUB} = 32.768 kHz[Note 4], T_A = +50°C	Normal operation	Square wave input			4.9	7.6	μA
						Resonator connection			5.0	7.7	μA
				f_{SUB} = 32.768 kHz[Note 4], T_A = +70°C	Normal operation	Square wave input			5.2	9.3	μA
						Resonator connection			5.3	9.4	μA
				f_{SUB} = 32.768 kHz[Note 4], T_A = +85°C	Normal operation	Square wave input			6.1	13.3	μA
						Resonator connection			6.2	13.4	μA

(**Notes** and **Remarks** are listed on the next page.)

37.6.2 24-bit ΔΣ A/D converter characteristics

Incorrect:

(4) 2 kHz sampling mode

(T_A = –40 to +85°C, $AV_{DD} \leq V_{DD}$ + 0.3 V, 2.4 V ≤ AV_{DD} ≤ 5.5 V, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Operation clock	f_{DSAD}	f_X oscillation clock, input external clock or high-speed on-chip oscillator clock is used		12		MHz
Sampling frequency	f_S			1935.125		Hz
Oversampling frequency	f_{OS}			0.75		MHz
Output data rate	T_{DATA}			512		μs
Data width	RES			24		bit
SNDR	SNDR	x1 gain High-speed system clock is selected as operating clock of 24-bit ΔΣ A/D converter (bit 0 of PCKC register (DSADCK) = 1)		80		dB
		x16 gain High-speed system clock is selected as operating clock of 24-bit ΔΣ A/D converter (bit 0 of PCKC register (DSADCK) = 1)	69	74		
		x32 gain High-speed system clock is selected as operating clock of 24-bit ΔΣ A/D converter (bit 0 of PCKC register (DSADCK) = 1)	65	69		
Passband (low pass band)	f_{CHPF}	At –3 dB (phase in high pass filter not adjusted)		0.303		Hz
In-band ripple 1	rp1	45 Hz to 55 Hz @50 Hz 54 Hz to 66 Hz @60 Hz	–0.01		0.01	dB
In-band ripple 2	rp2	45 Hz to 275 Hz @50 Hz 54 Hz to 330 Hz @60 Hz	–0.1		0.1	
In-band ripple 3	rp3	45 Hz to 660 Hz @50 Hz 54 Hz to 550 Hz @60 Hz	–0.1		0.1	
Passband (high pass band)	f_{CLPF}	–3 dB		836		Hz
Stopband (high pass band)	fatt	–80 dB		1273		Hz
Out-band attenuation	ATT1	f_S	–80			dB
	ATT2	2 f_S	–80			dB

<R>

Correct:

(4) 2 kHz sampling mode

(T_A = –40 to +85°C, $AV_{DD} \leq V_{DD}$ + 0.3 V, 2.4 V ≤ AV_{DD} ≤ 5.5 V, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Operation clock	f_{DSAD}	f_X oscillation clock, input external clock or high-speed on-chip oscillator clock is used		12		MHz
Sampling frequency	f_S			1953.125		Hz
Oversampling frequency	f_{OS}			0.75		MHz
Output data rate	T_{DATA}			512		μs
Data width	RES			24		bit
SNDR	SNDR	x1 gain High-speed system clock is selected as operating clock of 24-bit ΔΣ A/D converter (bit 0 of PCKC register (DSADCK) = 1)		80		dB
		x16 gain High-speed system clock is selected as operating clock of 24-bit ΔΣ A/D converter (bit 0 of PCKC register (DSADCK) = 1)	69	74		
		x32 gain High-speed system clock is selected as operating clock of 24-bit ΔΣ A/D converter (bit 0 of PCKC register (DSADCK) = 1)	65	69		
Passband (low pass band)	f_{CHPF}	At –3 dB (phase in high pass filter not adjusted)		0.303		Hz
In-band ripple 1	rp1	45 Hz to 55 Hz @50 Hz 54 Hz to 66 Hz @60 Hz	–0.01		0.01	dB
In-band ripple 2	rp2	45 Hz to 275 Hz @50 Hz 54 Hz to 330 Hz @60 Hz	–0.1		0.1	
In-band ripple 3	rp3	45 Hz to 660 Hz @50 Hz 54 Hz to 550 Hz @60 Hz	–0.1		0.1	
Passband (high pass band)	f_{CLPF}	–3 dB		836		Hz
Stopband (high pass band)	fatt	–80 dB		1273		Hz
Out-band attenuation	ATT1	f_S	–80			dB
	ATT2	2 f_S	–80			dB

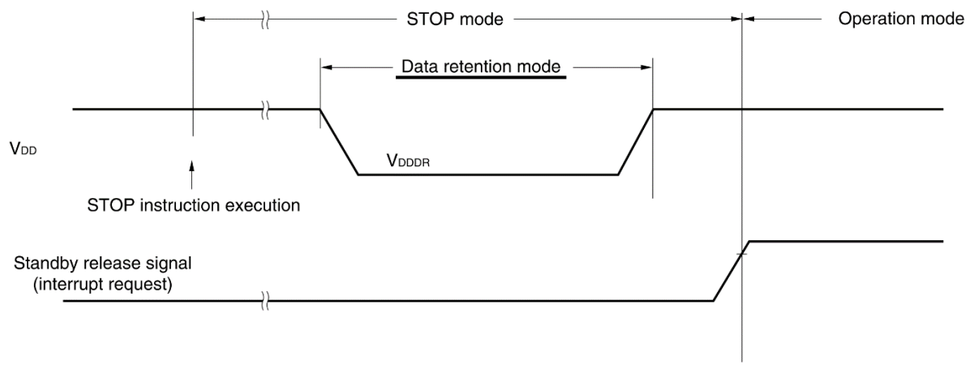
Incorrect:

37.9 Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics

(T_A = –40 to +85°C, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		1.46[Note]		5.5	V

Note The value depends on the POR detection voltage. When the voltage drops, the data is retained before a POR reset is effected, but data is not retained when a POR reset is effected.



Correct:

<R>

37.9 RAM Data Retention Characteristics

(T_A = –40 to +85°C, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		1.46[Note]		5.5	V

<R> **Note** The value depends on the POR detection voltage. When the voltage drops, the RAM data is retained before a POR reset is effected, but RAM data is not retained when a POR reset is effected.

<R>