RL78/F13, F14

Standby Function

R01AN4291EJ0100
Rev.1.00
Aug 31, 2018

Introduction

This application note describes the standby function (HALT mode, STOP mode and SNOOZE mode) of the RL78/F13 and the RL78/F14 microcontrollers by providing examples for setting each of the modes. For the clocks and each peripheral function described in this document, refer to the applicable User's Manual: Hardware.

Table of Contents

1. HALT mode

In HALT mode, power consumption is reduced by stopping the supply of the operation clock to the CPU. The CPU transitions to the HALT mode when the HALT instruction is executed. Even after the HALT instruction is executed, the state of each clock remains unchanged from the previous state. **Table 1-1** shows the clock states in HALT mode.

The HALT mode is released when a source for the enabled interrupt (the value of the interrupt mask flag is 0) is generated. **Figure 1-1** illustrates transition/release timing of HALT mode.

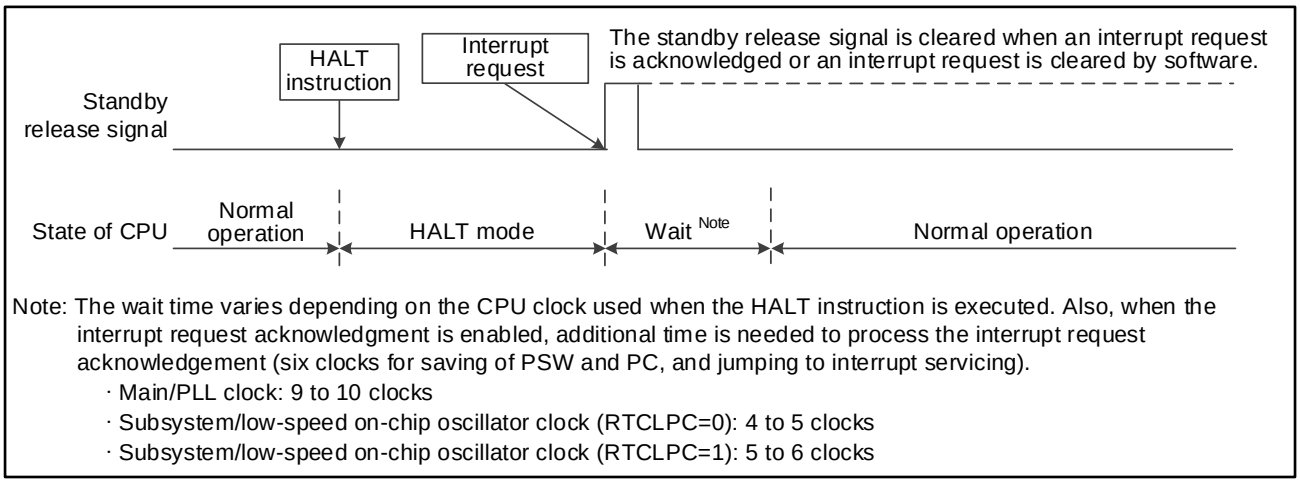
The processing after release of HALT mode varies depending on whether the interrupt request acknowledgement is enabled (EI) or disabled (DI). When the MCU transitions to HALT mode while the interrupt request acknowledgement is enabled (EI), the vectored-interrupt service will start after release of HALT mode. Meanwhile, when the MCU transitions to HALT mode while the interrupt request acknowledgement is disabled (DI), the instruction at the next address after the HALT instruction is executed after release of HALT mode. **Table 1-2** lists the peripheral function interrupts that can be used to release HALT mode.

Also, when the Flash memory CRC operation function (high-speed CRC) is completed, HALT mode will be released. Regarding the Flash memory CRC operation function, refer to the application note “Safety Function (R01AN2164)”.

Table 1-1: Clock states in HALT mode

Clock	Before transition to HALT mode	In HALT mode	After HALT mode is released
Clock supply to CPU	Not stopped	Stops	Not stopped
High-speed system clock	Oscillating or stops	Remains unchanged from the previous state	Remains unchanged from the previous state
High-speed on-chip oscillator clock	Oscillating or stops	Remains unchanged from the previous state	Remains unchanged from the previous state
PLL clock	Oscillating or stops	Remains unchanged from the previous state	Remains unchanged from the previous state
Low-speed on-chip oscillator clock	Oscillating or stops	Remains unchanged from the previous state	Remains unchanged from the previous state
Subsystem clock	Oscillating or stops	Remains unchanged from the previous state	Remains unchanged from the previous state
Watchdog timer-dedicated low-speed on-chip oscillator	Oscillating or stops	Remains unchanged from the previous state or stops Note	State before transition to HALT mode

Note: This varies depending on the settings of the WDTON bit and the WDSTBYON bit in the user option bytes (000C0H/020C0H).
When WDTON=1 and WDSTBYON=1: Continues oscillating.
When WDTON=1 and WDSTBYON=0: Stops oscillating. The counter is cleared after HALT mode is released and the clock oscillation starts.



Note: The wait time varies depending on the CPU clock used when the HALT instruction is executed. Also, when the interrupt request acknowledgment is enabled, additional time is needed to process the interrupt request acknowledgement (six clocks for saving of PSW and PC, and jumping to interrupt servicing).
· Main/PLL clock: 9 to 10 clocks
· Subsystem/low-speed on-chip oscillator clock (RTCLPC=0): 4 to 5 clocks
· Subsystem/low-speed on-chip oscillator clock (RTCLPC=1): 5 to 6 clocks

Figure 1-1: Transition/release timing of HALT mode

Table 1-2: Peripheral function interrupts used to release HALT mode

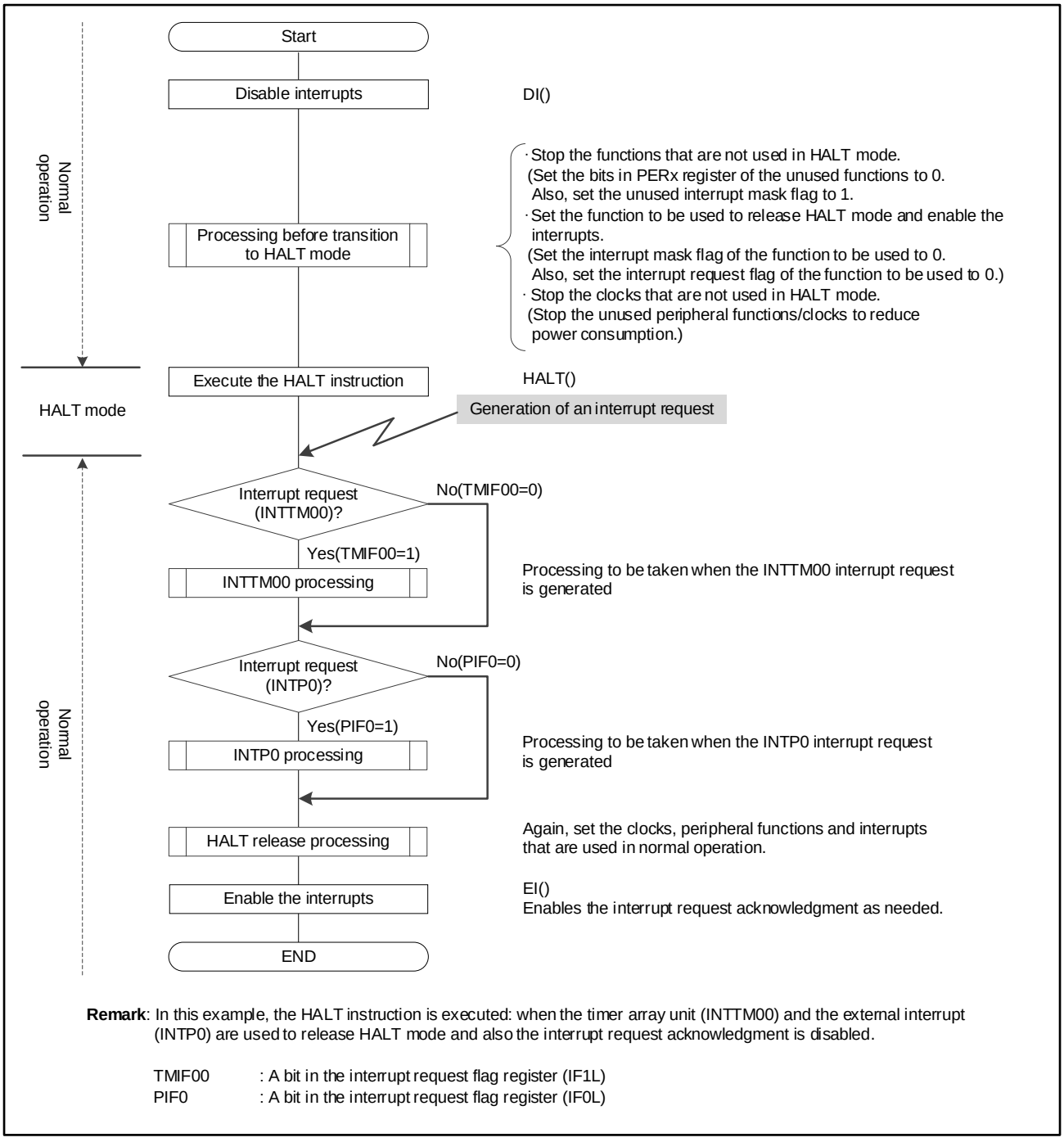
CPU's operation clock when the HALT instruction is executed	Peripheral function interrupts used to release HALT mode Note 1	
Main system clock (high-speed system clock or high-speed on-chip oscillator clock)	· Timer array unit · Realtime clock · Clock monitor · Timer RJ · Timer RD · A/D converter · Comparator · Serial array unit	· Serial interface (IICA) · LIN/UART module (RLIN3) · CAN interface · DTC · Voltage detection function · External interrupt · Key interrupt · Watchdog timer Note 2
Subsystem/low-speed on-chip oscillator clock	· Timer array unit · Realtime clock · Timer RJ · Timer RD · Serial array unit	· DTC · Voltage detection function · External interrupt · Key interrupt · Watchdog timer Note 2

Notes: 1. The peripheral function interrupts vary depending on the product used. For details, refer to the User's Manual: Hardware of the product used.

2. This varies depending on the settings of the WDTON bit and the WDSTBYON bit in the user option byte (000C0H/020C0H).
   When WDTON=1 and WDSTBYON=1: Continues oscillating.
   When WDTON=1 and WDSTBYON=0: Stops oscillating. The counter is cleared after HALT mode is released and the clock oscillation starts.

1.1 Procedure for setting HALT mode

Figure 1-2 shows an example for setting HALT mode.



Remark: In this example, the HALT instruction is executed: when the timer array unit (INTTM00) and the external interrupt (INTP0) are used to release HALT mode and also the interrupt request acknowledgment is disabled.

TMIF00 : A bit in the interrupt request flag register (IF1L)
PIF0 : A bit in the interrupt request flag register (IF0L)

Figure 1-2: Example for setting HALT mode

1.2 Cautions when HALT mode is used

- When the value of the interrupt mask flag is 0 (interrupt servicing is enabled) and also when the value of the interrupt request flag is 1 (an interrupt request signal is generated), HALT mode will be released immediately even when the HALT instruction is executed.
- The processing after the release of HALT mode varies depending on whether the interrupt request acknowledgement is enabled (EI) or disabled (DI). When the MCU transitions to HALT mode while the interrupt request acknowledgement is enabled (EI), the vectored-interrupt servicing will start when an interrupt request (interrupt servicing is enabled) is generated while the value of the interrupt mask flag is 0. When the MCU transitions to HALT mode when the interrupt request acknowledgement is disabled (DI), the instruction at the next address after the HALT instruction will be executed when an interrupt request is generated.
- HALT mode is released when a reset signal is generated.
- Whether to make the watchdog timer-dedicated low-speed on-chip oscillator continue oscillating or stop oscillating in HALT mode is selected by setting the user option byte (000C0H/020C0H).

2. STOP mode

In STOP mode, power consumption is reduced by stopping the main system clock (high-speed system clock, high-speed on-chip oscillator clock). The CPU transitions to STOP mode when the STOP instruction is executed. **Table 2-1** shows the clock states in STOP mode.

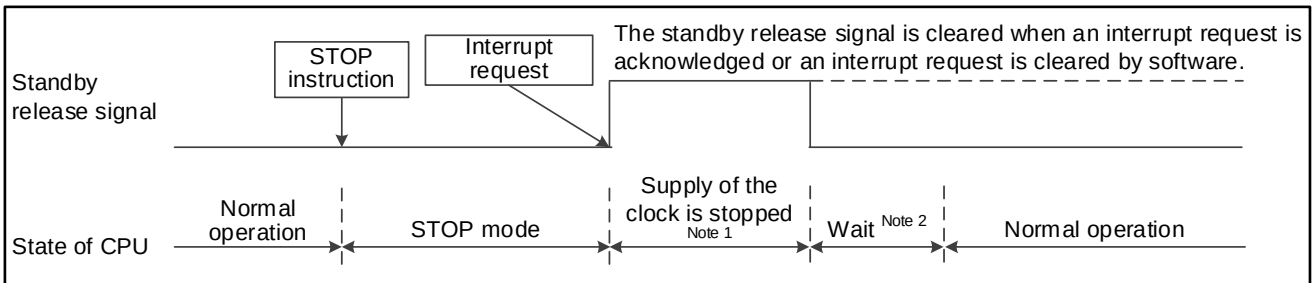
STOP mode is released when a source for the enabled interrupt (the mask flag of the interrupt enabled is set to 0) is generated. **Figure 2-1** illustrates transition/release timing of STOP mode.

The processing after the release of STOP mode varies depending on whether the interrupt request acknowledgement is enabled (EI) or disabled (DI). When the MCU transitions to STOP mode while the interrupt request acknowledgement is enabled (EI), the vectored-interrupt service will start after release of STOP mode. When the MCU transitions to STOP mode while the interrupt request acknowledgement is disabled (DI), the instruction at the next address after the STOP instruction is executed after release of STOP mode. **Table 2-2** shows the peripheral function interrupts that can be used to release STOP mode.

Table 2-1: Clock states in STOP mode

Clock	Before transition to STOP mode Note 1	In STOP mode	After release of STOP mode
Clock supply to CPU	Not stopped	Stops	Not stopped
High-speed system clock	Oscillating or stops	Stops	State before transition to STOP mode
High-speed on-chip oscillator clock	Oscillating or stops	Stops	State before transition to STOP mode
PLL clock Note 2	Stops	Stops	Stops
Low-speed on-chip oscillator clock	Oscillating or stops	Remains unchanged from the previous state	Remains unchanged from the previous state
Subsystem clock	Oscillating or stops	Remains unchanged from the previous state	Remains unchanged from the previous state
Watchdog timer-dedicated low-speed on-chip oscillator clock	Oscillating or stops	Remains unchanged from the previous state or stops Note 3	State before transition to STOP mode

Notes: 1. Be sure to execute the transition to STOP mode when the CPU clock is the main system clock (high-speed system clock or high-speed on-chip oscillator clock).
2. To make the CPU transition to STOP mode, set the PLLON bit to 0 (PLL stopped) beforehand.
3. This varies depending on the settings of the WDTON bit and the WDSTBYON bit in the user option byte (000C0H/020C0H).
When WDTON=1 and WDSTBYON=1: Continues oscillating.
When WDTON=1 and WDSTBYON=0: Stops oscillating. The counter is cleared after STOP mode is released and the clock oscillation starts.



Notes: 1. The clock supply stop time varies depending on the value set in the FRQSEL4 bit in the user option byte (000C2H/020C2H). When the CPU transitions to STOP mode with the high-speed system clock as the CPU clock, the time set in FRQSEL4 and the oscillation stabilization time set by the OSTS register are compared. As a result, the longer time will be the wait time.
· FRQSEL4 = 0: 18µs to 65µs
· FRQSEL4 = 1: 18µs to 105µs
2. The wait time varies depending on the CPU clock used when the STOP instruction is executed. When the interrupt request acknowledgment is enabled, additional time (six clocks for saving of PSW and PC, and jumping to interrupt processing) is needed to process the interrupt request acknowledgement.
· High-speed system clock: 4 to 5 clocks
· High-speed on-chip oscillator clock: 1 clock

Figure 2-1: Transition/release timing of STOP mode

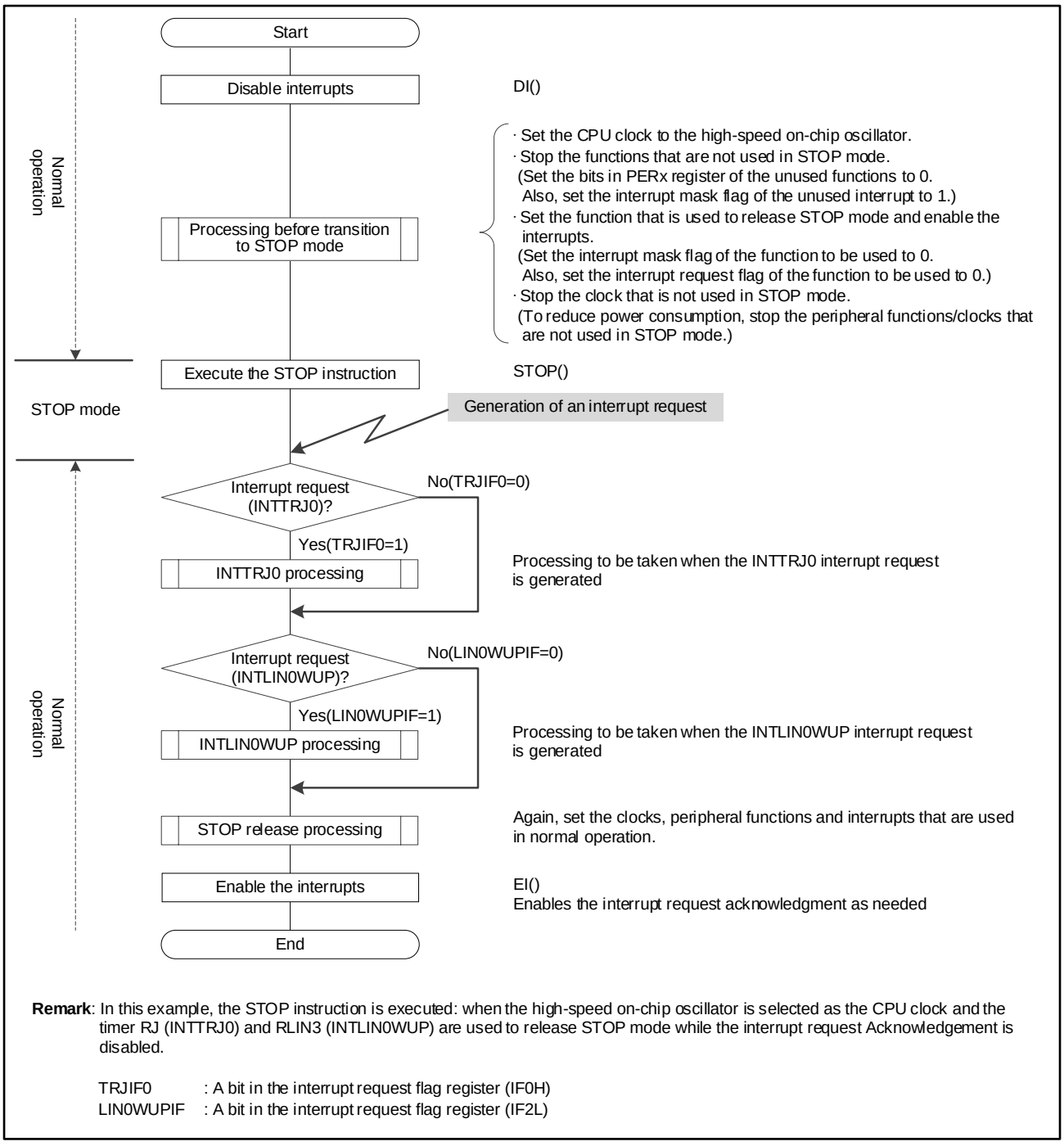
Table 2-2: Peripheral function interrupts used to release STOP mode

CPU's operation clock when STOP instruction is executed	Peripheral function interrupts used to release STOP mode Note 1
Main system clock (High-speed system clock or high-speed on-chip oscillator clock)	・ Realtime clock Note 2 ・ Timer RJ Note 2 ・ Comparator ・ Serial interface (IICA) Note 3 ・ Voltage detection function ・ External interrupt ・ Key interrupt ・ Watchdog timer Note 4

Notes: 1. The peripheral function interrupts vary depending on the product used. For details, refer to the User's Manual: Hardware of the product used.
2. This can be used when the subsystem clock that continues oscillating in STOP mode is selected as the operation clock.
3. This can be used when an extension code from the master device or a local address has been received in STOP mode.
4. This varies depending on the settings of the WDTON bit and the WDSTBYON bit in the option byte (000C0H/020C0H).
When WDTON=1 and WDSTBYON=1: Continues oscillating.
When WDTON=1 and WDSTBYON=0: Stops oscillating. The counter is cleared after STOP mode is released and the clock oscillation starts.

2.1 Procedure for setting STOP mode

Figure 2-2 shows an example for setting STOP mode.



Remark: In this example, the STOP instruction is executed: when the high-speed on-chip oscillator is selected as the CPU clock and the timer RJ (INTTRJ0) and RLIN3 (INTLIN0WUP) are used to release STOP mode while the interrupt request Acknowledgement is disabled.

TRJIF0 : A bit in the interrupt request flag register (IF0H)
LIN0WUPIF : A bit in the interrupt request flag register (IF2L)

Figure 2-2: Example for setting STOP mode

2.2 Cautions when STOP mode is used

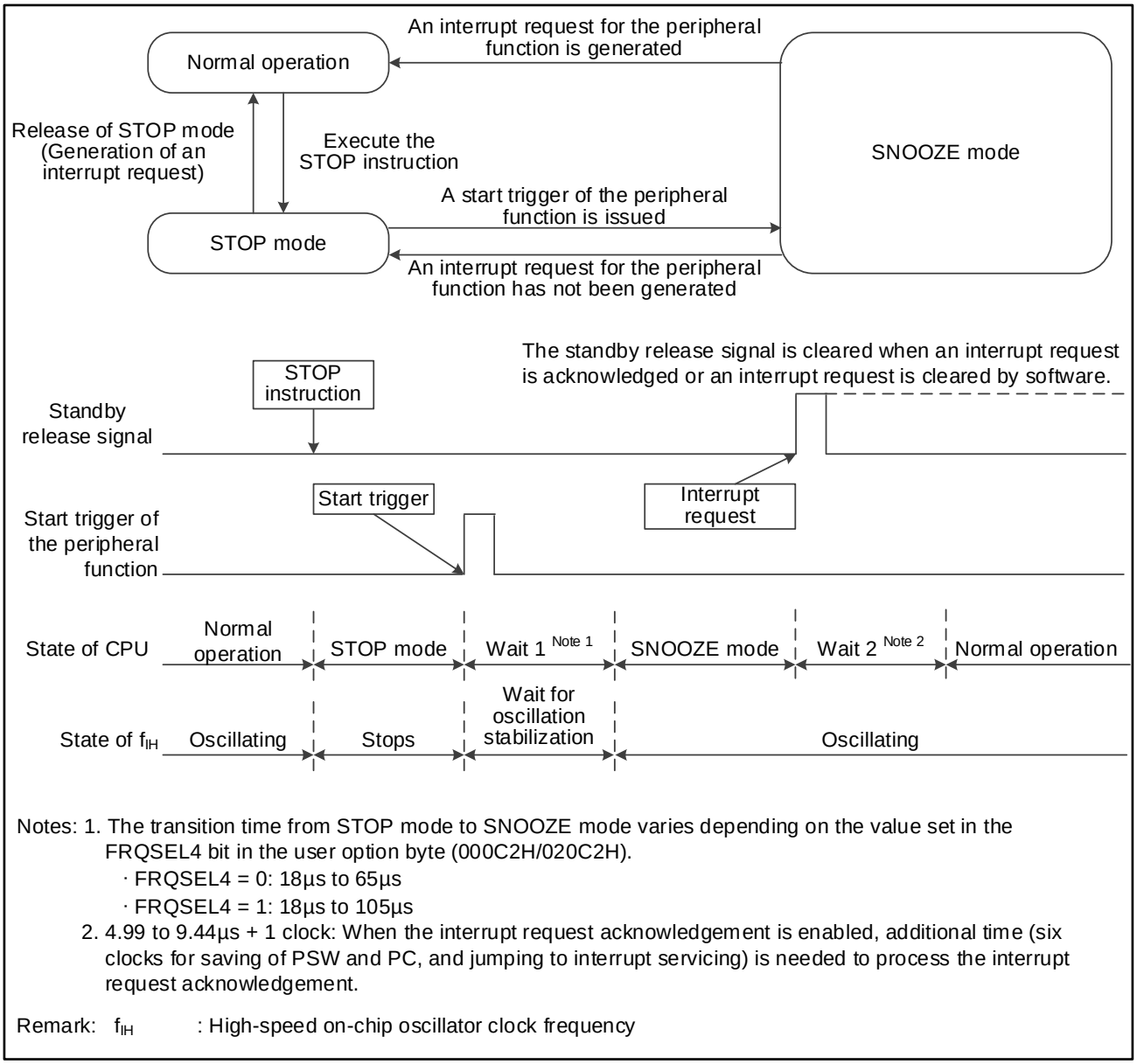
- The CPU can transition to STOP mode only when the CPU clock is the main system clock. Therefore, do not make the CPU transition to STOP mode when the CPU operates on the PLL clock, or the subsystem/low-speed on-chip oscillator clock.
- To make the CPU transition to STOP mode when the CPU clock is the high-speed system clock (X1 oscillator), the settings of the OSTS register need to be completed before executing the STOP instruction.
- Before executing the STOP instruction, be sure to stop the operation of the peripheral hardware (excluding the function(s) operating in SNOOZE mode) running on the main system clock (high-speed system clock or high-speed on-chip oscillator clock).
- When the value of the interrupt mask flag is 0 (interrupt servicing is enabled) and also when the value of the interrupt request flag is 1 (an interrupt request signal is generated), STOP mode will be released immediately even when the STOP instruction is executed.
- The processing after release of STOP mode varies depending on whether the interrupt request acknowledgement is enabled (EI) or disabled (DI). When the MCU transitions to STOP mode while the interrupt request acknowledgement is enabled (EI), the vectored-interrupt servicing will start when an interrupt request is generated while the value of the interrupt mask flag is 0 (interrupt servicing is enabled). When the MCU transitions to STOP mode while the interrupt request acknowledgement is disabled (DI), the instruction at the next address after the STOP instruction is executed when an interrupt request is generated.
- STOP mode is released when a reset signal is generated.
- Whether to make the watchdog timer-dedicated low-speed on-chip oscillator continue oscillating or stop oscillating in STOP mode is selected by setting the user option byte (000C0H/020C0H).

3. SNOOZE mode

In SNOOZE mode, A/D conversion, data reception by the LIN/UART module (RLIN3) and memory transfer by the DTC function are performed while the CPU operation is stopped. When a start trigger for the peripheral function is generated in STOP mode, the high-speed on-chip oscillator starts oscillating and the CPU transitions to SNOOZE mode.

SNOOZE mode is released when an interrupt request for a peripheral function operating in SNOOZE mode is generated. Otherwise, if the interrupt request of the peripheral function is not generated, the MCU returns to STOP mode. **Figure 3-1** illustrates transition/release timing of SNOOZE mode. **Table 3-1** shows the clock states in SNOOZE mode.

Also, the RL78/F13 and RL78/F14 MCUs are provided with a function to output the SNOOZE status at transition to SNOOZE mode and when the SNOOZE mode is released. The settings of each function (A/D conversion, data reception, DTC function and SNOOZE status output) in SNOOZE mode are described through examples in **Section 3.1** Procedure for setting SNOOZE mode.



Notes: 1. The transition time from STOP mode to SNOOZE mode varies depending on the value set in the FRQSEL4 bit in the user option byte (000C2H/020C2H).
- FRQSEL4 = 0: 18µs to 65µs
- FRQSEL4 = 1: 18µs to 105µs

2. 4.99 to 9.44µs + 1 clock: When the interrupt request acknowledgement is enabled, additional time (six clocks for saving of PSW and PC, and jumping to interrupt servicing) is needed to process the interrupt request acknowledgement.

Remark: f_{IH} : High-speed on-chip oscillator clock frequency

Figure 3-1: SNOOZE mode transition/release timing

Table 3-1: Clock states in SNOOZE mode

Clock	Normal operation mode (before transition to STOP mode)	In STOP mode Note 1	In SNOOZE mode	Normal operation mode (after release of SNOOZE mode)
Clock supply to CPU	Not stopped	Stops	Stops	Not stopped
High-speed system clock	Oscillating or stops	Stops	Stops	Stops
High-speed on-chip oscillator clock	Oscillating	Stops	Oscillating	Oscillating
PLL clock Note 2	Stops	Stops	Stops	Stops
Low-speed on-chip oscillator clock	Oscillating or stops	Remains unchanged from the previous state	Remains unchanged from the previous state	Remains unchanged from the previous state
Subsystem clock	Oscillating or stops	Remains unchanged from the previous state	Remains unchanged from the previous state	Remains unchanged from the previous state
Watchdog timer-dedicated low-speed on-chip oscillator clock	Oscillating or stops	Remains unchanged from the previous state or stops Note 3	Remains unchanged from the previous state or stops Note 3	State before transition to STOP mode

Notes: 1. Be sure to execute the transition to STOP mode when the CPU clock is the main system clock (high-speed on-chip oscillator clock).
2. To make the CPU transition to STOP mode, set the PLLON bit to 0 (PLL stopped) beforehand.
3. This varies depending on the settings of the WDTON bit and the WDSTBYON bit in the user option byte (000C0H/020C0H).
When WDTON=1 and WDSTBYON=1: Continues oscillating.
When WDTON=1 and WDSTBYON=0: Stops oscillating. The counter is cleared after transition to normal operation mode, and the clock oscillation starts.

The following peripheral functions can be used in SNOOZE mode.

- Realtime clock Note 1
- Timer RJ Note 1
- Timer RD Note 1
- Comparator
- Serial interface (IICA) Note 2
- LIN/UART module (UART mode of RLIN3) Note 3
- A/D converter Note 3
- Voltage detection function
- External interrupt
- Key interrupt
- Watchdog timer Note 4
- DTC

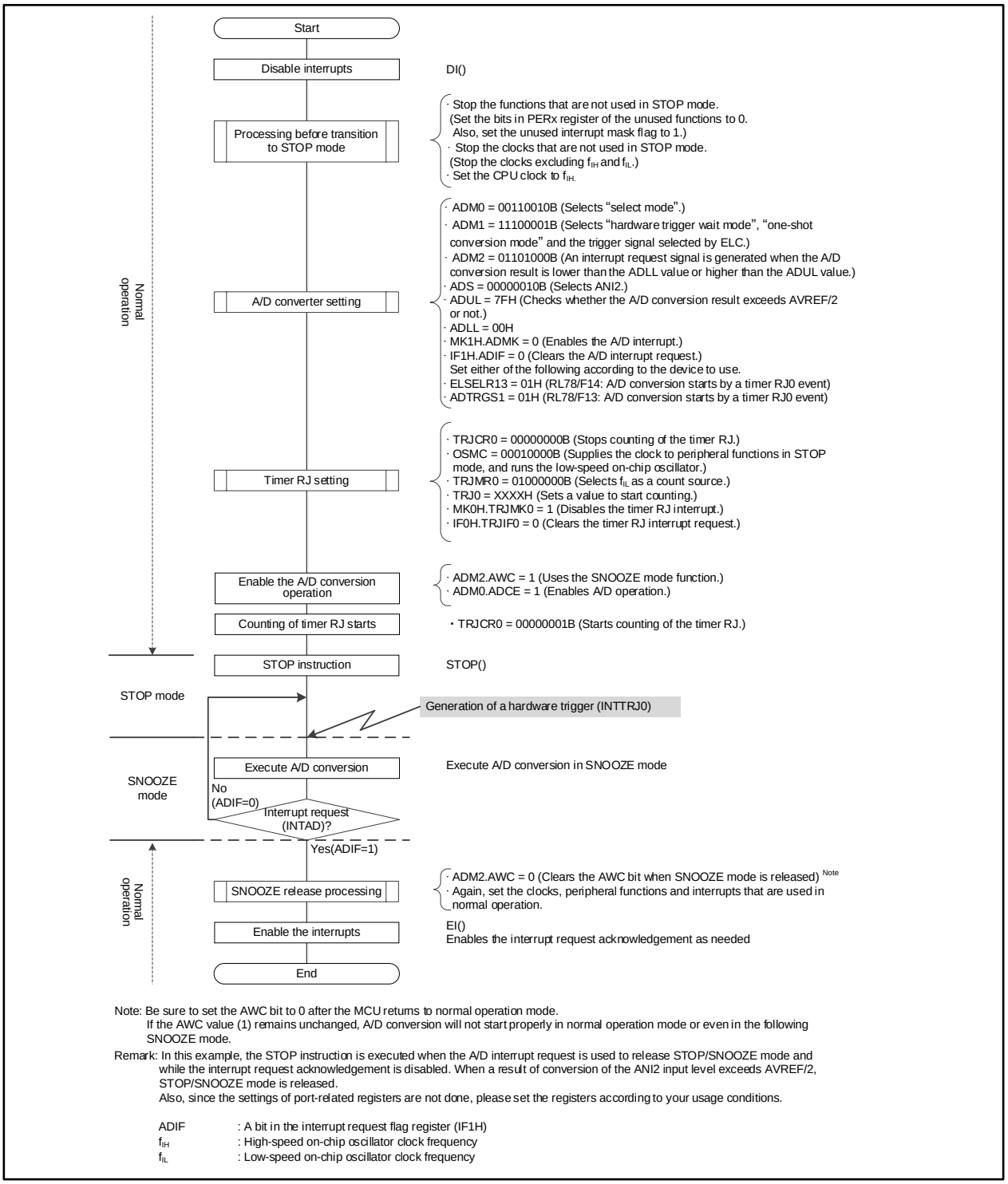
Notes: 1. These functions can be used when the subsystem clock that continues oscillating in STOP mode is selected as the operation clock. Timer RD can operate as the SNOOZE status output function or A/D trigger. However, it cannot be used as a factor for recovering from STOP/SNOOZE mode.
2. This can be used when an extension code from the master device or a local address has been received in STOP mode.
3. Operates on the high-speed on-chip oscillator as the clock source in SNOOZE mode. This can be used as a factor for recovering from SNOOZE mode when the interrupt conditions are satisfied.
4. This varies depending on the settings of the WDTON bit and the WDSTBYON in the user option byte (000C0H/020C0H).
When WDTON=1 and WDSTBYON=1: Continues oscillating.
When WDTON=1 and WDSTBYON=0: Stops oscillating. The counter is cleared after the CPU returns to the normal operation mode.

3.1 Procedure for setting SNOOZE mode

3.1.1 Example of setting A/D converter

Figure 3-2 is an example of setting the A/D converter to be used in SNOOZE mode. **Figure 3-3** is a timing chart.

In this example below, A/D conversion is performed using the timer RJ0 interrupt as a start trigger for the A/D converter. The A/D conversion result is respectively compared with its upper limit value (a value set by ADUL) and its lower limit value (a value set by ADLL). If the A/D conversion result is lower than the ADLL value or higher than the ADUL value, an interrupt is generated and the MCU transitions to normal operation mode from SNOOZE mode. When the above conditions are not satisfied, no interrupt is generated and the MCU returns to STOP mode from SNOOZE mode.

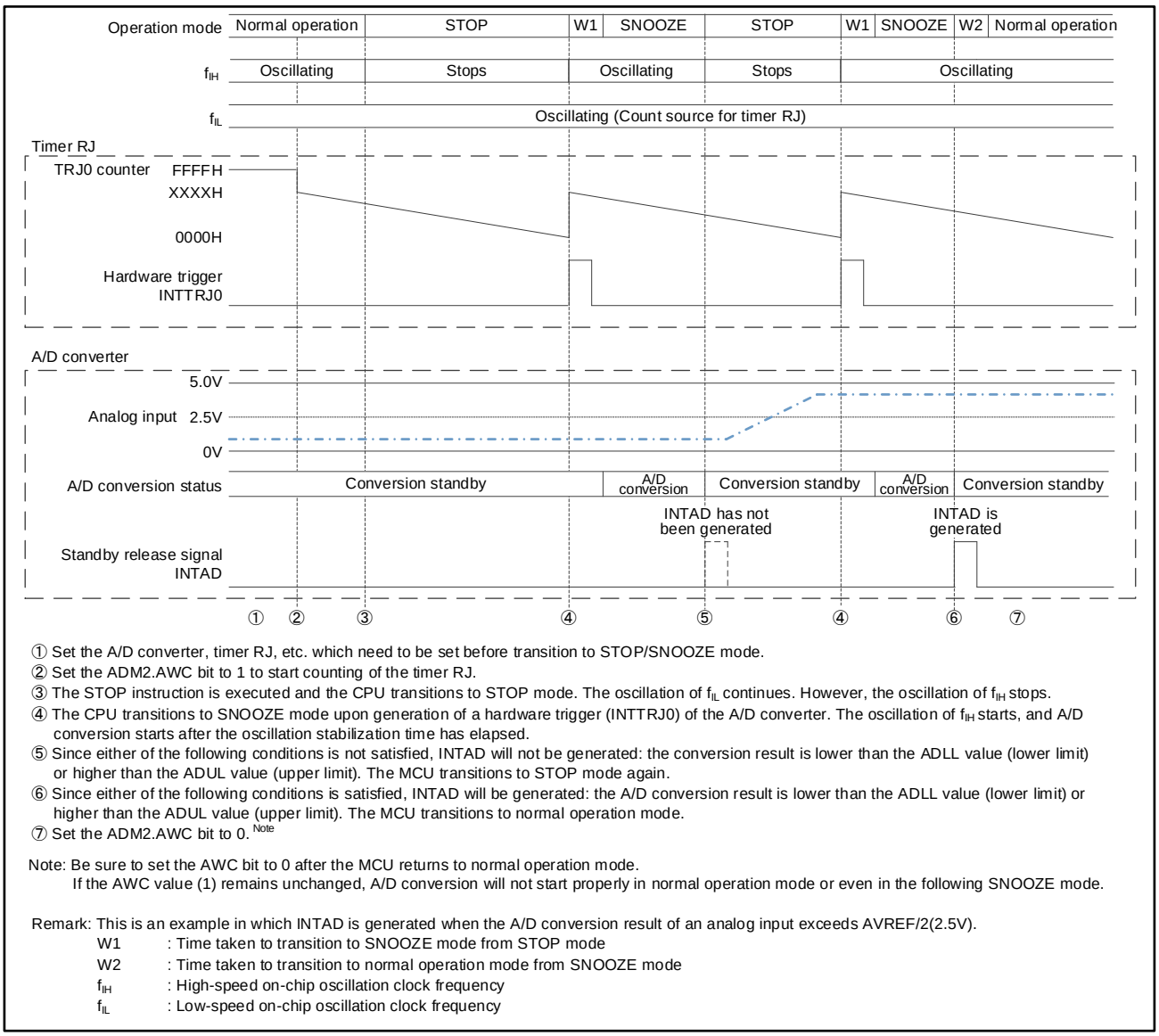


Note: Be sure to set the AWC bit to 0 after the MCU returns to normal operation mode. If the AWC value (1) remains unchanged, A/D conversion will not start properly in normal operation mode or even in the following SNOOZE mode.

Remark: In this example, the STOP instruction is executed when the A/D interrupt request is used to release STOP/SNOOZE mode and while the interrupt request acknowledgement is disabled. When a result of conversion of the ANI2 input level exceeds AVREF/2, STOP/SNOOZE mode is released.
Also, since the settings of port-related registers are not done, please set the registers according to your usage conditions.

ADIF : A bit in the interrupt request flag register (IF1H)
f_{IH} : High-speed on-chip oscillator clock frequency
f_{IL} : Low-speed on-chip oscillator clock frequency

Figure 3-2: Example for setting SNOOZE mode (A/D converter)



① Set the A/D converter, timer RJ, etc. which need to be set before transition to STOP/SNOOZE mode.
② Set the ADM2.AWC bit to 1 to start counting of the timer RJ.
③ The STOP instruction is executed and the CPU transitions to STOP mode. The oscillation of f_{IL} continues. However, the oscillation of f_{IH} stops.
④ The CPU transitions to SNOOZE mode upon generation of a hardware trigger (INTTRJ0) of the A/D converter. The oscillation of f_{IH} starts, and A/D conversion starts after the oscillation stabilization time has elapsed.
⑤ Since either of the following conditions is not satisfied, INTAD will not be generated: the conversion result is lower than the ADLL value (lower limit) or higher than the ADUL value (upper limit). The MCU transitions to STOP mode again.
⑥ Since either of the following conditions is satisfied, INTAD will be generated: the A/D conversion result is lower than the ADLL value (lower limit) or higher than the ADUL value (upper limit). The MCU transitions to normal operation mode.
⑦ Set the ADM2.AWC bit to 0. Note

Note: Be sure to set the AWC bit to 0 after the MCU returns to normal operation mode.
If the AWC value (1) remains unchanged, A/D conversion will not start properly in normal operation mode or even in the following SNOOZE mode.

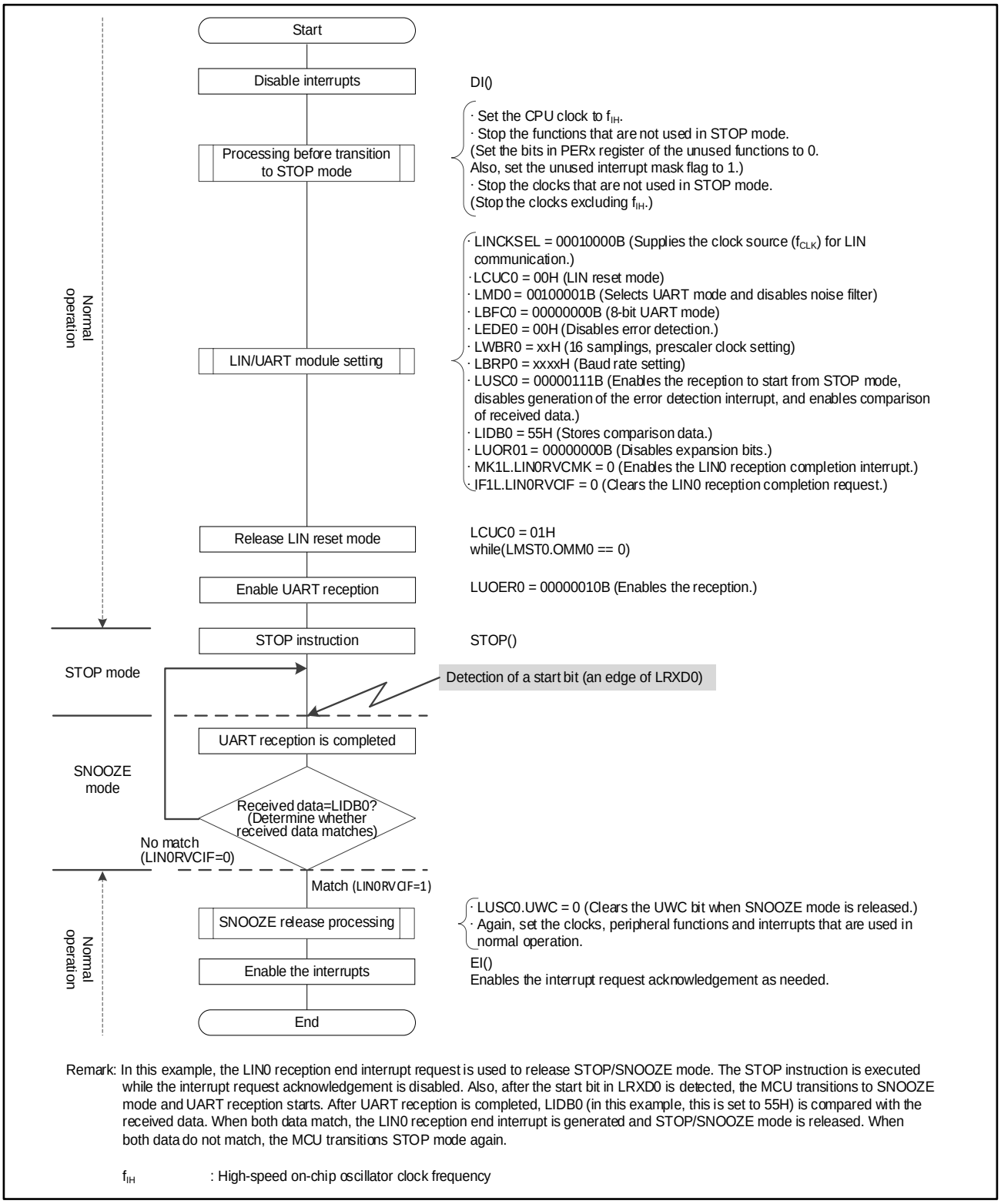
Remark: This is an example in which INTAD is generated when the A/D conversion result of an analog input exceeds AVREF/2(2.5V).
W1 : Time taken to transition to SNOOZE mode from STOP mode
W2 : Time taken to transition to normal operation mode from SNOOZE mode
f_{IH} : High-speed on-chip oscillation clock frequency
f_{IL} : Low-speed on-chip oscillation clock frequency

Figure 3-3: Timing chart of SNOOZE mode (AD converter)

3.1.2 Example of setting LIN/UART module (RLIN3)

Figure 3-4 is an example of setting the LIN/UART module (UART mode) to be used in SNOOZE mode. **Figure 3-5** is a timing chart.

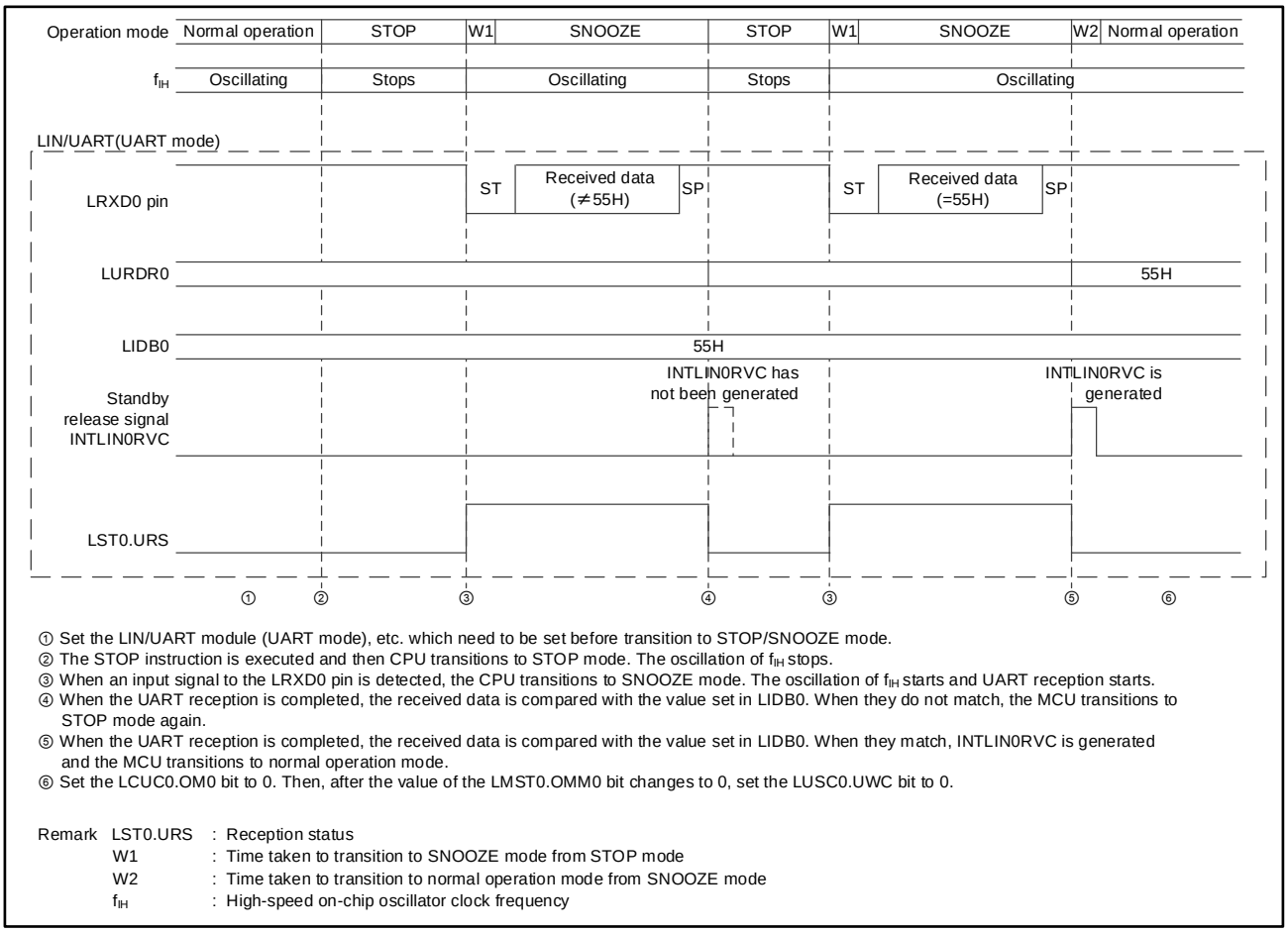
In this example below, the LIN/UART module (RLIN3) starts UART reception when the UART function of the LIN/UART module detects an edge (start bit) of LRXD0. When the data reception is completed, the data received by UART is compared with the data that has been set (the data is set in the LIDB0 register before the MCU transitions to STOP/SNOOZE mode). When the data matches, an interrupt is generated and SNOOZE mode is released.



Remark: In this example, the LIN0 reception end interrupt request is used to release STOP/SNOOZE mode. The STOP instruction is executed while the interrupt request acknowledgement is disabled. Also, after the start bit in LRXD0 is detected, the MCU transitions to SNOOZE mode and UART reception starts. After UART reception is completed, LIDB0 (in this example, this is set to 55H) is compared with the received data. When both data match, the LIN0 reception end interrupt is generated and STOP/SNOOZE mode is released. When both data do not match, the MCU transitions STOP mode again.

f_{IH} : High-speed on-chip oscillator clock frequency

Figure 3-4: Example for setting SNOOZE mode (LIN/UART)



① Set the LIN/UART module (UART mode), etc. which need to be set before transition to STOP/SNOOZE mode.
② The STOP instruction is executed and then CPU transitions to STOP mode. The oscillation of f_{IH} stops.
③ When an input signal to the LRXD0 pin is detected, the CPU transitions to SNOOZE mode. The oscillation of f_{IH} starts and UART reception starts.
④ When the UART reception is completed, the received data is compared with the value set in LIDB0. When they do not match, the MCU transitions to STOP mode again.
⑤ When the UART reception is completed, the received data is compared with the value set in LIDB0. When they match, INTLIN0RVC is generated and the MCU transitions to normal operation mode.
⑥ Set the LCUC0.OM0 bit to 0. Then, after the value of the LMST0.OMM0 bit changes to 0, set the LUSC0.UWC bit to 0.

Remark LST0.URS : Reception status
W1 : Time taken to transition to SNOOZE mode from STOP mode
W2 : Time taken to transition to normal operation mode from SNOOZE mode
f_{IH} : High-speed on-chip oscillator clock frequency

Figure 3-5: Timing chart of SNOOZE mode (LIN/UART)

Table 3-2 is an example of settings for communication speeds that are available in SNOOZE mode.

Table 3-2: Setting examples of communication speed

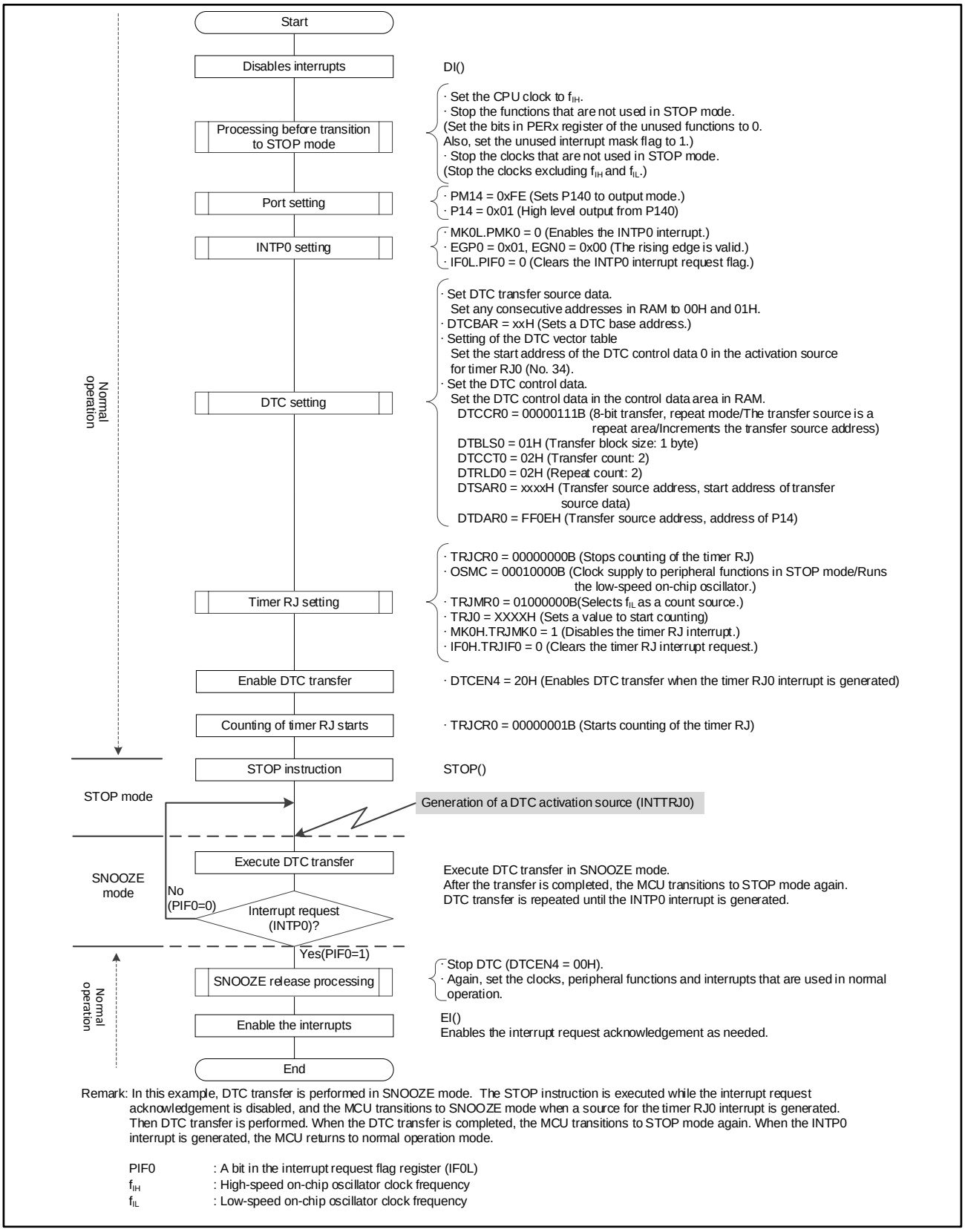
Communication format	Communication speed	LIN Communication clock source	LWBRn		LBRPn	FRQSEL4 Note
			LPRS[2:0]	NSPB[3:0]		
1ST-8DATA-1PRY-1SP	4800 bps	32MHz±2%	001B(1/2)	0000B (16 samplings)	203	0
	2400 bps	32MHz±2%	001B(1/2)	0000B (16 samplings)	412	0
	2400 bps	32MHz±2%	001B(1/2)	0000B (16 samplings)	410	1
	1200 bps	32MHz±2%	001B(1/2)	0000B (16 samplings)	826	1
1ST-8DATA-1PRY-1SP	4800 bps	24MHz±3%	001B(1/2)	0000B (16 samplings)	152	0
	2400 bps	24MHz±3%	001B(1/2)	0000B (16 samplings)	308	0
	2400 bps	24MHz±3%	001B(1/2)	0000B (16 samplings)	307	1
	1200 bps	24MHz±3%	001B(1/2)	0000B (16 samplings)	619	1
1ST-7DATA-1SP	2400bps	24MHz±5%	001B(1/2)	0000B (16 samplings)	307	0
	1200bps	24MHz±5%	001B(1/2)	0000B (16 samplings)	619	0
	1200bps	24MHz±5%	001B(1/2)	0000B (16 samplings)	617	1

Note: A bit in the user option byte (000C2H/020C2H). To set the frequency of the high-speed on-chip oscillator to 64MHz or 48MHz, set the FRQSEL4 bit to 1. To set the frequency to 32MHz or lower, set the FRQSEL4 to 0.

3.1.3 Example for setting DTC

Figure 3-6 is an example of setting the DTC transfer to be used in SNOOZE mode. **Figure 3-7** is a timing chart.

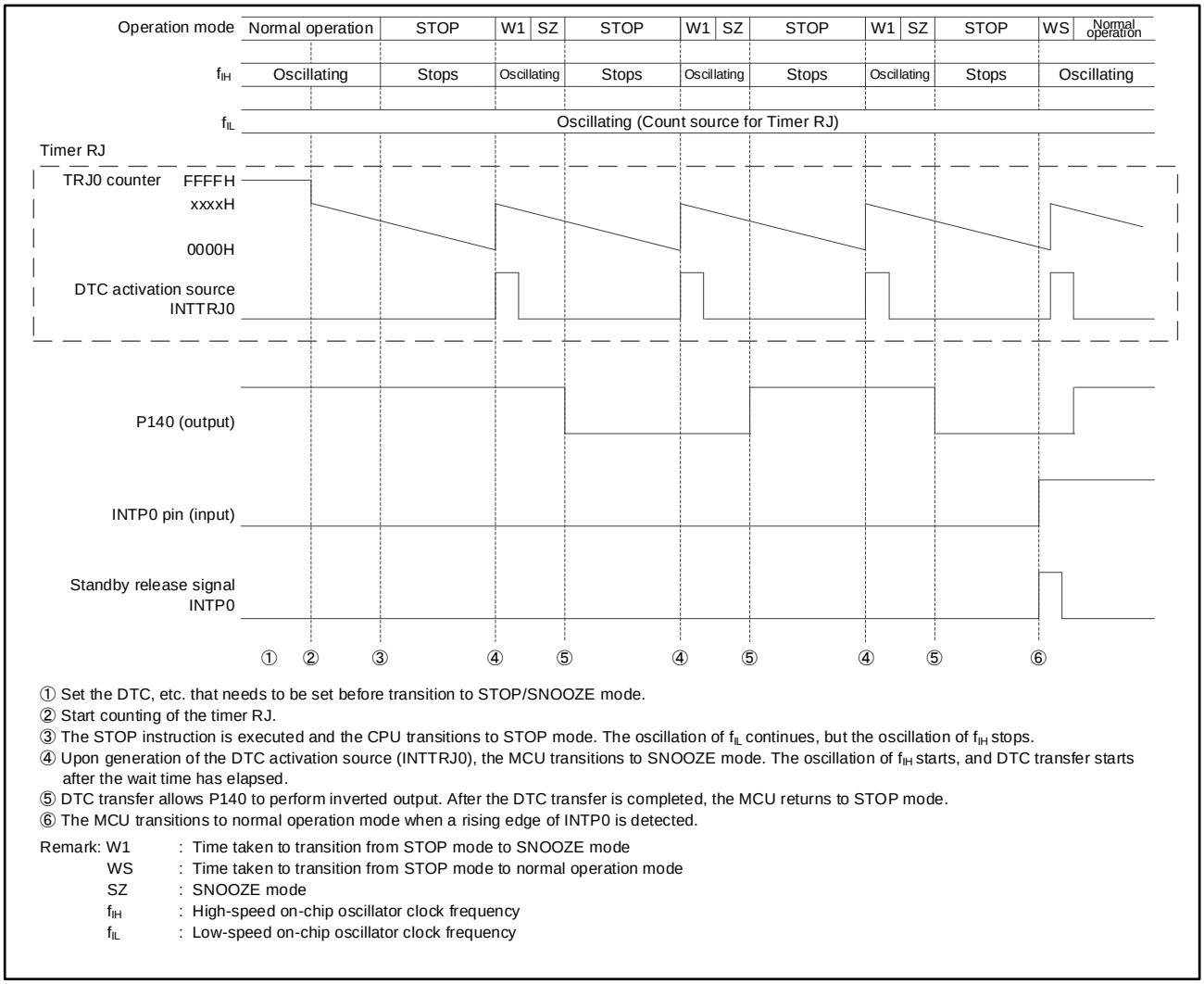
In this example below, DTC transfer is performed by using the timer RJ0 as a DTC activation source, which allows P140 to perform inverted output. The MCU transitions to normal operation mode from SNOOZE mode when the INTP0 interrupt is generated.



Remark: In this example, DTC transfer is performed in SNOOZE mode. The STOP instruction is executed while the interrupt request acknowledgement is disabled, and the MCU transitions to SNOOZE mode when a source for the timer RJ0 interrupt is generated. Then DTC transfer is performed. When the DTC transfer is completed, the MCU transitions to STOP mode again. When the INTP0 interrupt is generated, the MCU returns to normal operation mode.

PIF0 : A bit in the interrupt request flag register (IF0L)
f_{IH} : High-speed on-chip oscillator clock frequency
f_{IL} : Low-speed on-chip oscillator clock frequency

Figure 3-6: Example for setting SNOOZE mode (DTC)



① Set the DTC, etc. that needs to be set before transition to STOP/SNOOZE mode.
② Start counting of the timer RJ.
③ The STOP instruction is executed and the CPU transitions to STOP mode. The oscillation of f_{IL} continues, but the oscillation of f_{IH} stops.
④ Upon generation of the DTC activation source (INTTRJ0), the MCU transitions to SNOOZE mode. The oscillation of f_{IH} starts, and DTC transfer starts after the wait time has elapsed.
⑤ DTC transfer allows P140 to perform inverted output. After the DTC transfer is completed, the MCU returns to STOP mode.
⑥ The MCU transitions to normal operation mode when a rising edge of INTP0 is detected.

Remark: W1 : Time taken to transition from STOP mode to SNOOZE mode
WS : Time taken to transition from STOP mode to normal operation mode
SZ : SNOOZE mode
f_{IH} : High-speed on-chip oscillator clock frequency
f_{IL} : Low-speed on-chip oscillator clock frequency

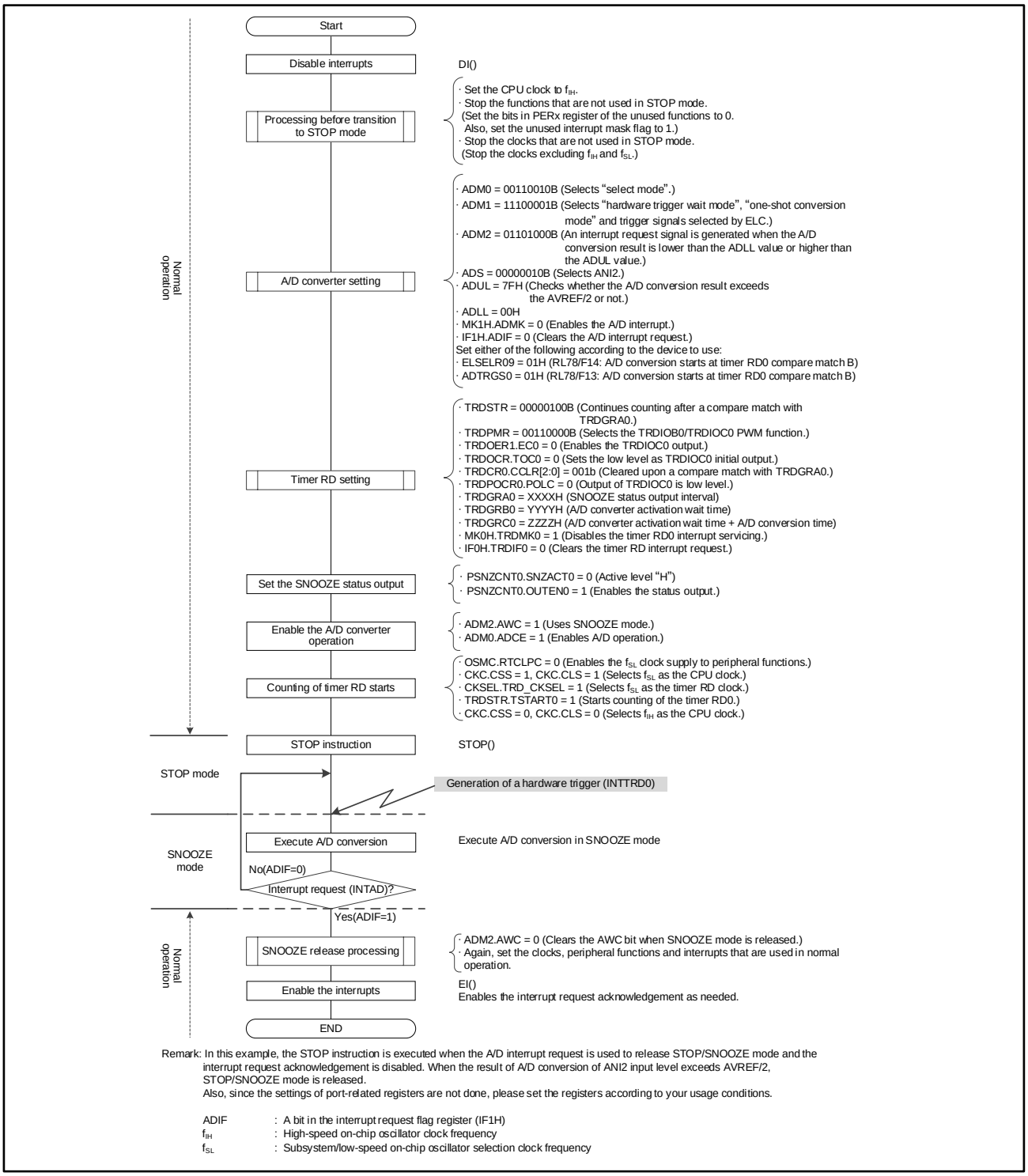
Figure 3-7: Timing chart of SNOOZE mode (DTC)

3.1.4 Example of setting SNOOZE status output

The SNOOZE status output function outputs the state of SNOOZE mode to SNZOUTi (i=0 to 7) pin. **Figure 3-8** is an example of setting the SNOOZE status output function in SNOOZE mode. **Figure 3-9** is a timing chart.

In the example below, the active level of SNOOZE status output is set to High using SNZOUT0.

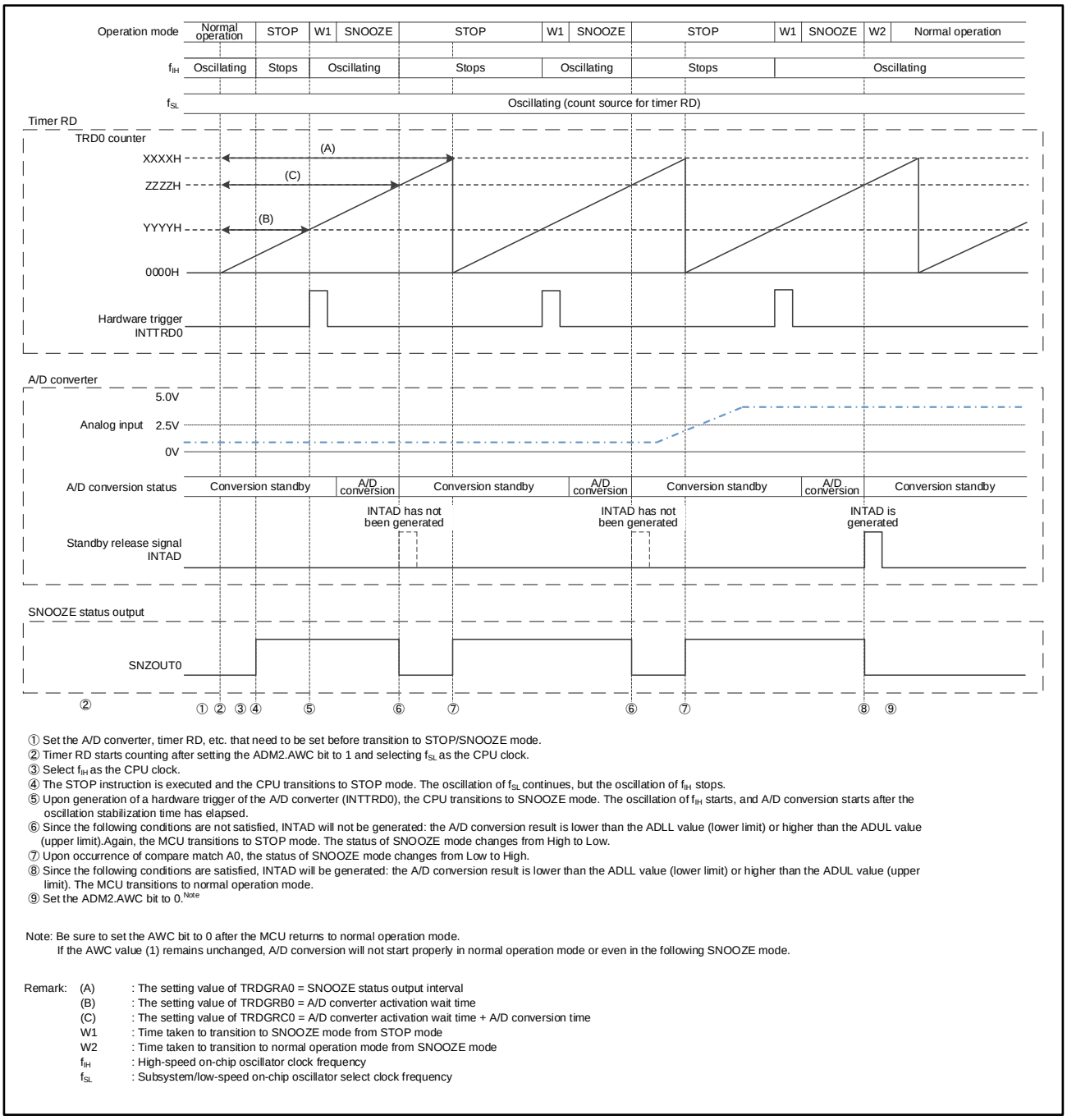
When A/D conversion is executed by using the timer RD0 interrupt as a start trigger of the A/D converter, a SNOOZE status is output. The A/D conversion result is respectively compared with its upper limit value (this is set by ADUL) and lower limit value (this is set by ADLL). An interrupt is generated and SNOOZE mode is released if the following conditions are satisfied: the A/D conversion result is lower than the ADLL value or is higher than the ADUL value. If the conditions are not satisfied, the MCU transitions to STOP mode again.



Remark: In this example, the STOP instruction is executed when the A/D interrupt request is used to release STOP/SNOOZE mode and the interrupt request acknowledgement is disabled. When the result of A/D conversion of ANI2 input level exceeds AVREF/2, STOP/SNOOZE mode is released.
Also, since the settings of port-related registers are not done, please set the registers according to your usage conditions.

ADIF : A bit in the interrupt request flag register (IF1H)
f_{IH} : High-speed on-chip oscillator clock frequency
f_{SL} : Subsystem/low-speed on-chip oscillator selection clock frequency

Figure 3-8: Example for setting SNOOZE mode (SNOOZE status output)



① Set the A/D converter, timer RD, etc. that need to be set before transition to STOP/SNOOZE mode.
② Timer RD starts counting after setting the ADM2.AWC bit to 1 and selecting f_{SL} as the CPU clock.
③ Select f_{IH} as the CPU clock.
④ The STOP instruction is executed and the CPU transitions to STOP mode. The oscillation of f_{SL} continues, but the oscillation of f_{IH} stops.
⑤ Upon generation of a hardware trigger of the A/D converter (INTTRD0), the CPU transitions to SNOOZE mode. The oscillation of f_{IH} starts, and A/D conversion starts after the oscillation stabilization time has elapsed.
⑥ Since the following conditions are not satisfied, INTAD will not be generated: the A/D conversion result is lower than the ADLL value (lower limit) or higher than the ADUL value (upper limit).Again, the MCU transitions to STOP mode. The status of SNOOZE mode changes from High to Low.
⑦ Upon occurrence of compare match A0, the status of SNOOZE mode changes from Low to High.
⑧ Since the following conditions are satisfied, INTAD will be generated: the A/D conversion result is lower than the ADLL value (lower limit) or higher than the ADUL value (upper limit). The MCU transitions to normal operation mode.
⑨ Set the ADM2.AWC bit to 0.[Note]

Note: Be sure to set the AWC bit to 0 after the MCU returns to normal operation mode.
If the AWC value (1) remains unchanged, A/D conversion will not start properly in normal operation mode or even in the following SNOOZE mode.

Remark: (A) : The setting value of TRDGRA0 = SNOOZE status output interval
(B) : The setting value of TRDGRB0 = A/D converter activation wait time
(C) : The setting value of TRDGRC0 = A/D converter activation wait time + A/D conversion time
W1 : Time taken to transition to SNOOZE mode from STOP mode
W2 : Time taken to transition to normal operation mode from SNOOZE mode
f_{IH} : High-speed on-chip oscillator clock frequency
f_{SL} : Subsystem/low-speed on-chip oscillator select clock frequency

Figure 3-9: Timing chart of SNOOZE mode (SNOOZE status output)

3.2 Cautions when SNOOZE mode is used

- To use the A/D converter in SNOOZE mode, be sure to set the AWC bit in the ADM2 register to 1 before executing the STOP instruction. To make the MCU transition to normal operation mode, set the value of the AWC bit to 0.
- To use UART communication in SNOOZE mode, set the UWC bit in LUSCn register to 1 before executing the STOP instruction. To make the MCU transition to normal operation mode, set the value of the UWC bit to 0. In the following conditions, when the UWC bit is set to 1, data reception may not be performed properly(a framing error or parity error could occur):
  - After setting UWC to 1, data reception has started before the MCU transitions to STOP mode.
  - Data reception has started while another SNOOZE mode function is being executed.
  - After the MCU returns to normal operation mode from STOP mode, data reception has started before setting the UWC bit to 0.
- SNOOZE mode is released when a reset signal is generated.
- Whether to make the watchdog timer-dedicated low-speed on-chip oscillator continue oscillating or stop oscillating in SNOOZE mode is selected by setting the user option byte (000C0H/020C0H).
- The MCU transitions to SNOOZE mode from STOP mode. For the cautions regarding STOP mode, refer to **Section 2.2** Cautions when STOP mode is used.

Website and Support

Renesas Electronics Website

http://www.renesas.com/

Inquiries

http://www.renesas.com/contact/

Revision History

Rev.	Issue Date	Change	
		Page	Description
1.00	June 30, 2018		Initial issue

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins

   Handle unused pins in accordance with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

RENESAS

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

Colophon 7.1