# RL78/L23 Simulator V1.03.00

## Release Note

Thank you for using the RL78/L23 simulator.

This document describes restrictions on and points for caution regarding the simulator.

Read this document before using the product.

### Contents

# Chapter 1. Target Devices and Supported Simulation Functions

The RL78/L23 simulator supports the following target devices.

| Device | Part Number | |
|---|---|---|
| RL78/L23 | 44 pins | R7F100LFE, R7F100LFG, R7F100LFJ, R7F100LFL |
| | 48 pins | R7F100LGE, R7F100LGG, R7F100LGJ, R7F100LGL |
| | 52 pins | R7F100LJE, R7F100LJG, R7F100LJJ, R7F100LJL |
| | 64 pins | R7F100LLE, R7F100LLG, R7F100LLJ, R7F100LLL |
| | 80 pins | R7F100LMG, R7F100LMJ, R7F100LML |
| | 100 pins | R7F100LPG, R7F100LPJ, R7F100LPL |

As well as CPU instructions, the RL78/L23 simulator is capable of simulating the following items in the target devices.

- Peripheral modules such as timers, the serial array unit, the serial interface, and SNOOZE mode sequencer
- Virtual target board (simulation via the [I/O panel] window)
- MCU pin signal waveforms (simulation via the [Timing chart] window)
- Current drawn

# Chapter 2. Changes

This chapter describes changes from V1.02.00 to V1.03.00 of the RL78/L23 simulator.

## 2.1 Improvements to the RL78/L23 simulator

### 2.1.1 Support for SMS debugging function

The simulator supports debugging of the SNOOZE Mode Sequencer (SMS) for RL78/L23 in this version.

- Function to break automatically when an SMS activating trigger occurs, before executing the processing specified by the SMSI0 register
- SMS step execution
- Function to forcibly start SMS debugging from the SMSI0 register

Note: SMS is executed in synchronization with the CPU.

For details, refer to the e² studio help.

# Chapter 3. Points for Caution

This section lists points for caution on using the RL78/L23 simulator. These points for caution are in the following two categories.

• Differences in behavior between the target devices and the simulator due to simulator specifications

• Usage of simulation functions (operations in and configuration of the GUI windows)

CS+ for CC supports the [Virtual Board] panel which is described in those points for caution.

## 3.1 Differences in behavior between the target devices and the simulator

### 3.1.1 Peripheral functions not supported by the simulator

The simulator is not capable of simulating the following peripheral functions of the target devices.

- Regulator
- Power-on-reset circuit
- Voltage detector
- Operation state control
- Capacitive sensing unit (CTSU2La)
- Security function
- Oscillation stop detector

### 3.1.2 Special function registers (SFRs) for controlling port functions

The following SFRs which control port functions are not simulated.

Although read/write access for each register can proceed normally, the operation does not change even though the value is changed.

- Port input mode registers (PIMxx)
- Port digital input disable registers (PDIDISxx)
- Output current control enable register (CCDE)
- Output current select registers (CCSx)
- 40-mA port output control register (PTDC)
- Port mode control T registers (PMCTxx)

In products with 128 or fewer Kbytes of code flash memory and 52 or fewer pins, the PM62 bit in the port mode register (PMxx) and the P62 bit in the port register (Pxx) are control bits that are connected to an internal terminal. After release from the reset state, these bits must be set to 0 by software. However, the simulator does not support functionality of the PM62 and P62 bits of the products stated above; the PM62 and P62 bits are respectively fixed to 1 and 0 and their values cannot be changed. Note that P62 is not mounted on the stated products, so this does not affect the simulation of programs.

### 3.1.3 Power supply for battery backup (VBAT)

The simulator does not simulate the power supply for battery backup (VBAT).

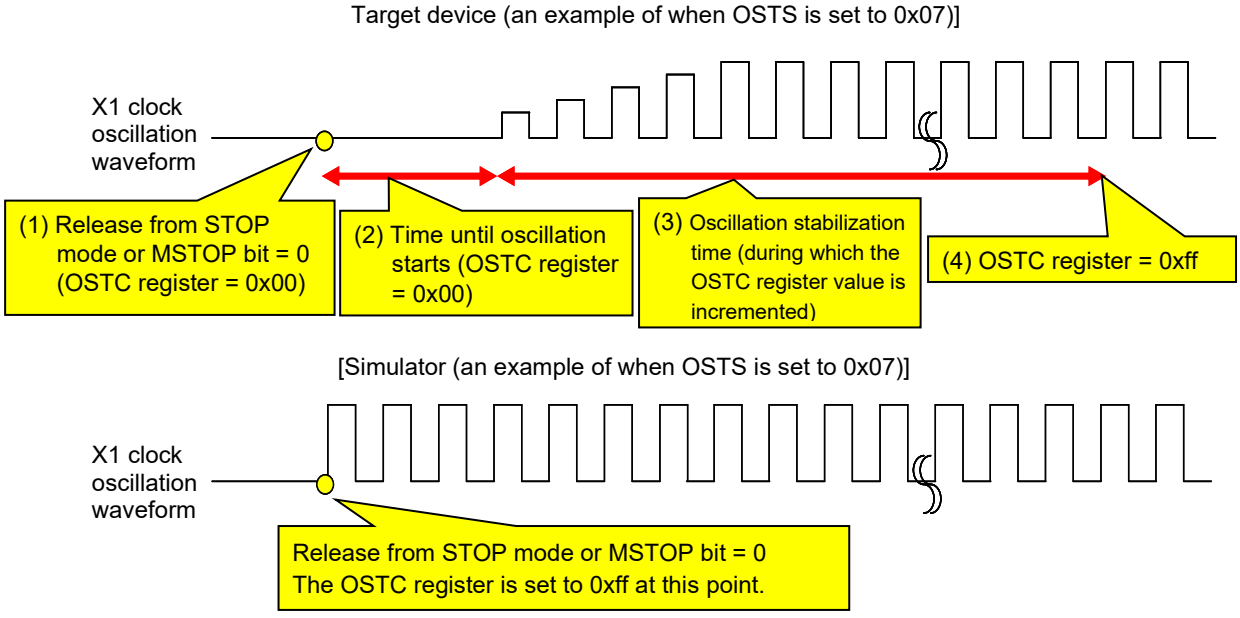
### 3.1.4 Oscillation stabilization time for the clock generator

Since the simulator does not simulate the clock oscillator oscillation stabilization time, stabilization always takes no time. When the oscillation is started, the OSTC register is set to one of the following values (i.e. not incremented).

| OSTS Setting | OSTC Value |
|---|---|
| 0x0 : $2^8$/fx | 0x80 |
| 0x1 : $2^9$/fx | 0xc0 |
| 0x2 : $2^{10}$/fx | 0xe0 |
| 0x3 : $2^{11}$/fx | 0xf0 |
| 0x4 : $2^{13}$/fx | 0xf8 |
| 0x5 : $2^{15}$/fx | 0xfc |
| 0x6 : $2^{17}$/fx | 0xfe |
| 0x7 : $2^{18}$/fx | 0xff |

The following figure illustrates this operation.

In the target device, oscillation by the X1 clock starts after operation has passed through states (1) to (4). In the simulator, states (1) through (4) are skipped and oscillation instantly starts.



Therefore, pay attention to the code that waits for oscillation stabilization.

There is no problem if a program is created under the condition that execution proceeds after the oscillation stabilization wait period when the OSTC register value becomes the maximum value, or when the OSTC register value exceeds the specified value, but if a program is created under the condition that execution proceeds after the oscillation stabilization wait period when the OSTC register value becomes a value other than the maximum value, execution will enter an endless loop.

The following shows examples of code that causes and does not cause problems.

The examples are when the OSTS is set to 0x07.

Correct code example (1)

```
while(OSTC != 0xff)
{
NOP();/* wait */
}
```

Correct code example (2)

```
while(OSTC <= 0xf0)
{
NOP();/* wait */
}
```

Example of code that may cause problems

```
while(OSTC != 0xf0)
{
NOP();/* wait */
}
```

### 3.1.5 SFRs (CMC, OSMC, HIOTRM, MIOTRM, LIOTRM, MODRV, SOMRG, and CSC) in the clock generator

The following SFRs which belong to the clock generator are not simulated. Although read/write access for each register can proceed normally, the operation does not change even if the value is changed.

- Bits 0, 1, and 2 (AMPH, AMPHS0, and AMPHS1) of the clock operating mode control register (CMC)
- Bit 0 (HIPREC)[Note] of the subsystem clock supply mode control register (OSMC)
- High-speed on-chip oscillator trimming register (HIOTRM)
- Middle-speed on-chip oscillator trimming register (MIOTRM)
- Low-speed on-chip oscillator trimming register (LIOTRM)
- Main oscillator oscillation mode select register (MODRV)
- XT1 oscillator margin checking control register (SOMRG)

Note: In the simulator, the HIPREC bit is fixed to 0. However, if the STOP instruction is executed, the RL78/L23 will enter the STOP mode.

The operation in initialization of the EXCLKS, OSCSELS, AMPHS1, and AMPHS0 bits of the clock operation mode control register (CMC) and the XTSTOP bit of the clock operation status control register (CSC) differs between the target device and the simulator.

[Target device]

EXCLKS, OSCSELS, AMPHS1, AMPHS0, and XTSTOP bits are only initialized by a power-on reset and retain their values following resetting by other reset sources.

[Simulator]

Since the simulator does not support resetting by a power-on reset, the EXCLKS, OSCSELS, AMPHS1, AMPHS0, and XTSTOP bits are not initialized by the reset. The values of these bits are retained in response to resets from the following reset sources, which are those supported by the simulator.

- External reset input via the RESET pin
- Internal reset due to detection of a program malfunction by the watchdog timer
- CPU reset by the debugger

### 3.1.6 Operating clock of the timer array unit

Do not specify an operating clock that runs at or below 233 Hz. If the operating clock for the timer array unit runs at or below 233 Hz, then the timer array unit will not work properly (it will behave as if operating with a clock that is faster than the one selected).

### 3.1.7 Noise filter of the timer array unit

Although the target device's timer array unit has a function to turn the noise filters on and off in order to reduce noise from the timer input pins, the simulator does not simulate this function since there is no noise in the simulator's signals. That is, whether filtering is on or off makes no difference to the behavior.

### 3.1.8 Digital filters in timer RJ

The simulator does not simulate the operation of the digital filters in timer RJ.

### 3.1.9 16-bit timers KB40, KB41, and KB42

The simulator does not simulate the following function of 16-bit timers KB40, KB41, and KB42.

- Cancel trigger for forced output stop 2 on the TKBOnp output

  The simulator allows writing of the value 1 to a TKBPAFTTnp (n = 0-2; p = 0, 1) bit in a forced output stop function cancel trigger register n (TKBPAHFTn) (n = 0-2) but does not support cancel trigger for forced output stop 2 on the TKBOnp output. Thus, forced output stop 2 on the TKBOnp output cannot be canceled by writing 1 to a TKBPAHFTn.TKBPAFTTnp bit.

### 3.1.10 Operation with fixed off when a short pulse is input to 16-bit timers KB40, KB41, and KB42

For the fixed-off function in the device, the start of forced output stopping is not synchronized with the clock and the output is stopped as soon as forced output stop input 2 becomes active. However, the cancellation of stopping is synchronized with the clock. Therefore, if the duration of the active level of forced output stop input 2 is shorter than one cycle of the fKBKC clock, forced output stop 2 may not be canceled and the output may continue to be in the forced output stop state in some cases.

In the simulator, when a falling edge of the forced output stop input 2 signal is received, forced output stopping is immediately canceled even if the duration of the active level of forced output stop input 2 is shorter than one cycle of the fKBKC clock.

### 3.1.11 Noise filters for 16-bit timers KB40, KB41, and KB42

Although 16-bit timers KB40, KB41, and KB42 of the target device have functions to turn the noise filter for the INTPx external interrupt on or off and to select the noise elimination digital filter, the simulator does not simulate these functions.

Regardless of the settings of the INTFCKx1, INTFCKx0, and PNFENx bits in an external interrupt control register x (INTPCTLx) (x = 0-7), the simulator behaves as if these 16-bit timers do not have an INTPx filter or noise filter.

### 3.1.12 SFRs (RTCC0, RTCC1, and SUBCUD) of the realtime clock (RTC)

The operation of realtime clock control register 0 (RTCC0), realtime clock control register 1 (RTCC1), and the time error correction register (SUBCUD) of the realtime clock (RTC) differs between the target device and the simulator.

[Target device]

The registers are cleared to 00H in response to an internal reset by the power-on reset circuit.

[Simulator]

The registers are not cleared to 00H in response to a reset because the simulator does not support the power-on reset circuit.

### 3.1.13 Delay times of EXSD

The simulator does not support the delay times of EXSD. This causes the following differences in behavior between the target devices and the simulator.

[Target device]

(1) When INTEXSD is generated, a delay of “1/(2 x fSXP) + 1/fCLK” or “1/(2 x fSXP) + 2/fCLK” always applies to the internally shaped waveform of the EXSDI1 signal.

(2) In setting or clearing of the EXSDD00, EXSDD01, and PRTY0 flags, a delay of “1/(2 x fSXP) + 1/fCLK” or “1/(2 x fSXP) + 2/fCLK” always applies to the internally shaped waveform of the EXSDI1 signal.

(3) For (1) and (2) above, the delay of fCLK occurs when the EXSDLPC bit in external signal sampler control register 1 (EXSDM1) is 0 but does not occur when the EXSDLPC bit is 1.

[Simulator]

(1) INTEXSD is generated on both rising and falling edges of the shaped EXSDI1 signal.

(2) The EXSDD00, EXSDD01, and PRTY0 flags are set or cleared at the same time as the shaped EXSDI1 signal without delay.

(3) The delay of fCLK does not occur regardless of whether the EXSDLPC bit in external signal sampler control register 1 (EXSDM1) is 0 or 1.

### 3.1.14 Clock output/buzzer output controller

When $f_{MAIN}$ is selected as an output clock, the [Timing chart] window does not show the clock waveform of the PCLBUZn signal.

When $f_{MAIN}/2$ or a slower signal is selected as an output clock, the [Timing chart] window shows the clock waveform.

### 3.1.15 Interval interrupts generated by the watchdog timer

The timing of the generation of interval interrupts by the watchdog timer differs between the target device and the simulator.

[Target device]

When 75% + 1/4$f_{IL}$ of overflow time is reached

[Simulator]

When 75% of overflow time is reached

### 3.1.16 A/D converter

When no voltage is being applied to the VDD or AVREFP pin, the default reference voltage of the A/D converter is 5.0 V.

To change the reference voltage, input the desired voltage values for VDD and AVREFP via the [Signal Data Editor] window.

The temperature sensor output voltage is always 1.05 V.

### 3.1.17 Conversion start time in the A/D converter

The simulator does not support the conversion start time but recognizes the time as 0 clock cycles in simulation.

### 3.1.18 D/A converter

When no voltage is being applied to the VDD pin, the simulator operates on the assumption that 5 V is being input to the VDD pin. To change the voltage, specify the desired voltage value for the VDD pin via the [Signal Data Editor] window or some other means.

### 3.1.19 Reference voltage of the comparators (CMP)

When no voltage value is set for the VDD pin, the simulator generates the reference voltage on the assumption that 5 V is being input to the VDD pin.

To change the reference voltage, specify the desired voltage value for the VDD pin via the [Signal Data Editor] window or some other means.

### 3.1.20 Digital filters in the comparators (CMP)

The simulator does not simulate the digital filters in the comparators (CMP).

### 3.1.21 Response time of the comparators

Since the simulator does not simulate the response time of the comparators, the response time is always 0 second. This does not change even if the speed of the comparators is changed in the comparator output control register (COMPOCR).

### 3.1.22 Clock used in the serial array unit

Do not specify a clock that is 233 Hz or lower in the following cases. If the following clock of the serial array unit is 233 Hz or lower, then the serial array unit will not operate correctly (it will behave as if operating via a clock that is faster than the one selected).

- Operating clock($f_{MCK}$) is 233Hz or lower.
- Transfer clock setting by dividing the operation clock ($f_{MCK}$ ÷ (SDRmn[15:9] + 1)) is 233Hz or lower.

### 3.1.23 Noise filter of the serial array unit

Although the target device's serial array unit has a function to turn the noise filter on and off in order to reduce noise on the input pin, the simulator does not simulate this function since there is no noise in the simulator's signals. That is, whether filtering is on or off makes no difference to the behavior.

### 3.1.24 SDRmn registers of the serial array unit

The values read from the seven higher-order bits of the serial data registers (SDRmn) during serial operation differ between the target device and the simulator.

[Target device]

0 is read.

[Simulator]

The value read is that at the time serial operation starts.

### 3.1.25 SSm registers in the serial array unit

During serial communications, when the operation start trigger of channel n (SSmn) in the serial channel start register m (SSm) is set to 1, operation of the simulator differs from that of the actual target device in the way stated below.

[Target device]

The target device stops communications and enters the suspended state.

[Simulator]

The simulator does not stop communications. Accordingly, the TSFmn and BFFmn bits in the serial status register mn (SSRmn) are not cleared to 0.

### 3.1.26 Input switch control register (ISC) of the serial array unit

When ISC2 to ISC7 bits of the input switch control register (ISC) are not set for normal operation, CSI communications are disabled in the [Serial] window. To enable CSI communications in the [Serial] window, clear the bits to 0.

### 3.1.27 IICA serial interface

IICA supports pin waveform generation and the communications through the [Serial] window. The following functions are not supported.

- Digital filter
- Arbitration
- Detection of transmission errors
- Communication reservation

### 3.1.28 SFR (IICCTLn1) for the IICA serial interface

The following SFR which controls the IICA serial interface is not simulated.

Although read/write access for the register can proceed normally, the operation does not change even if the value is changed.

- Bit 2 (DFCn) [Note] and bit 3 (SMCn) of the IICA control register n1 (IICCTLn1)

Note: In the simulator, the DFCn bit can be set to 1 only when the SMCn bit is 1.

### 3.1.29 LCD controller/driver

The simulator does not support simulation of the following items.

- LCD driver waveforms (waveforms A and B)
- LCD driver voltage generator (external resistance division, internal voltage boosting, and capacitive splitting)
- Biasing methods for LCD panels
- Output waveforms of common and segment signals
- LCD clock control

Even when no voltage is applied to the VDD pin, the simulator behaves as if 5 V is being applied to the VDD pin. If you wish to avoid this behavior, input a desired voltage value via the [Signal Data Editor] window.

For the segment LCD for internal driver and the 14-segment & 8-digit LCD for internal driver, which are the components on the [Virtual Board] panel, you can check the operation of the control program of the LCD driver by illuminating the segments of this LCD in response to control signals sent from the LCD driver.

### 3.1.30 ISCLCD register in the LCD controller/driver

The simulator does not support the functions of the ISCVL4 to ISCVL1 and ISCCAP bits (to control input through Schmitt trigger buffers) of the LCD input switch control register (ISCLCD).

### 3.1.31 Registers PFSEG0 to PFSEG7 in the LCD controller/driver

When a PFCOMy bit (y = 0 to 3) in the LCD port function register (PFSEGx) (x = 0) is 1 and a PFSEGz bit (z = 00 to 55) in the LCD port function register (PFSEGx) (x = 1 to 7) is 1, the behavior differs as follows.

[Target device]

- PFCOMy: The pin is used as a common output pin.
- PFSEGz: The pin is used as a segment output pin.

[Simulator]

- PFCOMy: The pin is used as a common output or port pin.
- PFSEGz: The pin is used as a segment output or port pin.

### 3.1.32 Segment signal output pins for use by the LCD controller/driver

When a pin is to be used as a segment signal output pin, registers that control port functions (PUm, POMm, PIMm, PMCAxx, PMCTxx, PMxx, and Pxx) must be set accordingly on the target device.

In the simulator, on the other hand, each pin works as a segment signal output pin without such settings and ports operate according to the settings of registers PUxx, POMm, PIMm, PMCAxx, PMCTxx, PMxx, and Pxx.

### 3.1.33 Times taken for data transfer by the data transfer controller (DTC)

The times taken for data transfer by the data transfer controller (DTC) differ between the target device and the simulator.

[Target device]

- A response time is required from detection of a DTC activation source until data transfer starts.
- A waiting time is required for access to extended special function registers (2nd SFRs).
- The DTC puts the data transfer on hold when the CPU executes any instruction that holds the DTC pending.
- Access to the data bus by the CPU is put on hold during DTC transfer.

[Simulator]

- Data transfer starts immediately after detection of a DTC activation source.
- No waiting time is required even for access to extended special function registers (2nd SFRs).
- The DTC does not put the data transfer on hold even when the CPU executes any instruction that should hold the DTC pending.
- Access to the data bus by the CPU is not put on hold even during DTC transfer.

### 3.1.34 Repeat mode of the data transfer controller (DTC)

If any of the conditions listed below is satisfied while the data transfer controller (DTC) is in repeat mode, the DTC ignores activation sources and will thus fail to transfer data.

- A DTC transfer count register j (DTCCTj) is set to 00H (number of transfers: 256 times).
- A DTC block size register j (DTBLSj) is set to 00H (block size: 256 or 512 bytes).
- A DTC control register j (DTCCRj) is used to set the transfer data size to 16 bits and the corresponding DTC block size register j (DTBLSj) is used to set the block size to 256 bytes or more.

### 3.1.35 Output signals from the logic and event link controller (ELCL)

When any among $f_{CLK}$, $f_{IHP}$, $f_{IMP}$, or $f_{SXP}$ is selected as an input signal for the logic and event link controller (ELCL) and this leads to the frequency of the clock in the logic cell block being higher than that of the main clock, the output signals of the ELCL may not be correctly displayed in the simulator GUI or in the [Virtual Board] panel.

### 3.1.36 SFRs (WKUPMD and PSMCR) for the standby function

The following SFRs which control standby functions are not simulated.

Although read/write access for each register can proceed normally, the operation does not change even if the value is changed.

- Bit 0 (FWKUP) of the standby mode release setting register (WKUPMD)
- Bit 0 (RAMSDS) and bit 1 (RAMSDMD) of the memory power reduction control register (PSMCR)

### 3.1.37 Reset

Among the sources for generating reset signals, the following types of internal reset do not occur in the simulator.

- Internal reset by comparison of supply voltage and detection voltage of the power-on-reset circuit (POR)
- Internal reset by comparison of supply voltage and detection voltage of the voltage detectors (LVD0 and LVD1)
- Internal reset due to execution of an illegal instruction
- Internal reset due to a RAM parity error
- Internal reset due to illegal-memory access

In addition, the behavior differs as follows if a reset signal is input from the RESET pin.

[Target device]

The MCU is reset when the RESET pin goes low. Release from the reset state proceeds when the RESET pin goes high.

[Simulator]

The MCU is not reset when the RESET pin goes low. The simulator is reset momentarily and then released when the RESET pin goes high.

### 3.1.38 Reset control flag register (RESF)

The simulator only supports the WDTRF bit of the reset control flag register (RESF).

The simulator is not capable of simulating the operations of the other bits (TRAP, RPERF, IAWRF, and LVIRF). Only the default values of these bits are indicated.

Additionally, in the simulator, the reset control flag register (RESF) is only cleared by a reset input via the RESET pin.

### 3.1.39 Executing illegal instructions

If an illegal instruction (instruction code: 0xFF) is executed, the target device will be reset, but the simulator will go into an endless loop (the illegal instruction will be executed repeatedly).

### 3.1.40 Safety functions

The simulator does not support the following safety functions.

- Flash memory CRC operation function (high-speed CRC, general-purpose CRC)
- Flash memory guard function
- RAM parity error detection
- RAM guard function
- SFR guard function
- Illicit memory access detection
- Guard function of invalid memory access detection control register (IAWCTL)

### 3.1.41 Values calculated for current drawn when the ELCL is selected as the operating clock of UARTAn

With the simulator, when the ELCL is selected as the operating clock of UARTAn, the current value for UARTAn is 0 uA

### 3.1.42 Values calculated for current drawn when an event input from the ELCL is selected as the counter clock of the 32-bit interval timer

With the simulator, when an event input from the ELCL is selected as the counter clock of the 32-bit interval timer, the current value for the 32-bit interval timer is 0 uA.

### 3.1.43 Flash memory

The simulator does not support serial programming.

### 3.1.44 FSASTL register in flash memory

In the simulator, the WRER and ERER bits in the flash memory sequencer status register L (FSASTL) are always 0 because writing or erasure of flash memory will not fail.

### 3.1.45 FLSEC register in flash memory

The simulator does not support the BAPR bit in the flash security flag monitor register (FLSEC). The BAPR bit is fixed to 1, so changing of the boot cluster size or bank swapping setting (BTBLS[3:0]) is enabled.

### 3.1.46 Setting of the flash memory control mode

Setting each of the flash memory control modes requires writing the values to the flash protect command register (PFCMD) and flash programming mode control register (FLPMC) according to the procedure for executing the specific sequence.

In the target device, if writing to any other memory area or register is attempted during execution of the specific sequence, a protection error occurs, writing to the specified register does not proceed, and the FPRERR flag of the flash status register (PFS) is set to 1.

In the simulator, if writing to any other memory area is attempted during execution of the specific sequence, a protection error does not occur, writing to the specified register proceeds, and the FPRERR flag of the flash status register (PFS) is not set to 1. If writing to any other register is attempted during execution of the specific sequence, the operation will be the same as that of the target device.

### 3.1.47 Entry to the non-programmable mode in self-programming

In the target device, after executing the procedure for entry to the non-programmable mode from the code flash memory programming mode or data flash memory programming mode and waiting, reading from the target flash memory for the programming mode before the mode transition becomes possible.
In the simulator, reading from the target flash memory for the programming mode before the mode transition becomes possible without the wait.

### 3.1.48 Security settings of flash memory

The simulator supports neither control of connection to the programmer and on-chip debugger nor programmer connection ID authentication.

### 3.1.49 Access to the data flash memory

In the target device, the DFLEN bit in the data flash control register (DFLCTL) must be set to 1 to enable access to the data flash memory. However, the simulator can read the data flash memory with a CPU instruction even if the setting of the DFLEN bit is 0.

## 3.2 Usage of simulation functions

### 3.2.1 Simulation speed

The simulation speed of RL78/L23 simulator depends on the number of operating peripheral functions.

If many peripheral functions are operating, the simulation speed becomes from several to ten times slower than the actual device. Note

With the use of only a few, or even no peripheral functions, the simulation speed may become faster than the actual device.

Note: The measurement environment for simulation speed is as follows.

CPU: 3.20 GHz (Quad-Core); memory: 8 Gbytes; OS: Windows11

### 3.2.2 Pin waveforms in the [Timing chart] window

The maximum length of a pin waveform is 4096 signal-level changing points. After reaching this maximum length, the data will be overwritten from the oldest value. If this length is not sufficient, use the following methods.

- Reduce the number of registered pins
- Stop the user program at the place where you want to confirm the waveform by using a breakpoint

### 3.2.3 Controlling windows

The following keyboard operations are not available in the simulator windows ([Signal Data Editor], [I/O panel], and [Serial]).

- Navigation via tab or arrow keys (←, ↑, →, ↓)
- Deletion via the Del or Backspace keys
- Cut & paste and other operations via the Ctrl + C, V, X, A, or Z keys.

Perform the above operations as follows.

- Navigation: Navigate by using the mouse.
- Deletion: Right-click and perform the action from the context menu.
- Cut & paste, etc.: Right-click and perform the action from the context menu.

### 3.2.4 Closing the [Simulator GUI] window

The [Simulator GUI] window can only be closed by disconnecting from the debugging tool, or by closing CS+ in proper manner. The X button cannot be used.

Additionally, although it appears that the X button can be pressed if Aero is enabled in Windows, pressing this button will not close the [Simulator GUI] window.

### 3.2.5 Disconnecting the debug tool

CS+ may be closed if the debugging tool is disconnected while any of the following dialog boxes is open from the [Simulator GUI] window. Be sure that the following dialog boxes have been closed before disconnecting the simulator.

- •Save As
- •Open
- •New
- •Color
- •Font
- •Customize
- •Loop
- •Select Pin
- •Search Data
- •Format (UART)
- •Format (CSI)
- •Format (IIC)
- •Message (e.g. Error)
- •Parts Button Properties
- •Analog Button Properties
- •Parts Key Properties
- •Parts Level Gauge Properties
- •Parts Led Properties
- •Parts Segment LED Properties
- •Parts Matrix Led Properties
- •Parts Buzzer Properties
- •Pull up / Pull down
- •Entry Bitmap
- •Object Properties

### 3.2.6 [Serial] window

When using the [Serial] window as the data receiver for the simplified $I^2C$ of the serial array unit or IICA, only ACK can be generated after receiving the data. NACK cannot be generated.

### 3.2.7 Setting the pins in the simulator GUI or the [Virtual Board] panel

The peripheral I/O redirection register (PIOR) can be manipulated by a program or debugger operations to re-assign specific multiplexed pin functions to alternative port pins in the same way as on the actual device. However, if the assignment of serial interface functions to port pins is changed to use the [Serial] window or the serial communication component of the [Virtual Board] panel, set the changed port pins when making settings of the [Format] dialog box in the [Serial] window or the serial communication component of the [Virtual Board] panel. This enables the use of the [Serial] window or the serial communication component of the [Virtual Board] panel when port pins to which multiplexed functions are assigned by the PIOR are changed.

After re-assigning a given pin function by using the PIOR, be sure to select the name of the pin you are currently using in the [Select Pin] dialog box of the simulator GUI or “Connected To” of the component in the [Virtual Board] panel.

### 3.2.8 Simulation of current drawn

The following notes apply to the function of measuring current.

- The current is calculated roughly as that drawn by the MCU alone based on the typical values (TYP.) for the actual devices. Note that the current values other than for the MCU are not included.
- The number of change points of measurable current is 200,000. The program stops when the number exceeds 200,000.

### 3.2.9 Events for debugging during the use of boot swapping or bank swapping

When boot swapping or bank swapping of flash memory is to be used, if an event for debugging is set in a boot cluster or a bank before swapping, correct swapping cannot proceed with the data for the address of the event or the event may be invalid.

Delete all events before swapping and set any events again after swapping.

### 3.2.10 Events for debugging during copying of the code for rewrite processing to the RAM in self-programming

When the code flash area or extra area is to be rewritten through self-programming, if an event for debugging is set at a RAM address before copying of the code for rewrite processing to the RAM, the event may be invalid after copying to the RAM has proceeded.

### 3.2.11 SMS debugging function

The following notes apply to the SMS debugging function.

- During SMS step execution, if the clock required to execute a sequencer processing command stops, if SMS is reset (PRR1.SMSRES = 1), or if SMS is forcibly terminated (SMSC.SMSSTOP = 1), SMS step execution stops and the program breaks.
- When the SMS is in the "Activation pending state" or "Activating trigger waiting state", clearing the SMSTRGWAIT bit in the SMSC register to 0 or executing the function to forcibly start SMS debugging causes a break before the processing specified by the SMSI0 register is executed. Note that if the SMSI0 processing takes 1 cycle of fCLK, the break may occur after execution.

# Chapter 4. Restrictions

This section describes restrictions on using the RL78/L23 simulator.

## 4.1 Values calculated for current drawn when using 16-bit timers KB40, KB41, and KB42

[Details] The current value is incorrect when using the 16-bit timer KB40 on 44- to 64-pin products with 64 to 128 Kbytes of flash memory.
It uses the current value on 100-pin products with 512 Kbytes of flash memory.

[Workaround] There is no workaround.

[Schedule for fixing the problem] This problem will be fixed in the next or a subsequent version.

## Revision History

| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| Rev.1.00 | Mar.01.26 | - | First Edition |

# 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 be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. 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.
6. 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.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. 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.
9. 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.
10. 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.
11. 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.
12. 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.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

## Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

## Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

## Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.