Introduction
This application note is related to the clock generators used in the RL78/F13 and RL78/F14 microcontrollers and describes the structure of clocks, their setting procedures and the points to note.

Target Device
RL78/F14 and RL78/F13 groups (CAN and LIN incorporated), and RL78/F13 group (LIN incorporated)
Contents

1. Hardware description ........................................................................................................ 4
  1.1 CPU clock status transition ...................................................................................... 5

2. Procedures for setting each clock generator .................................................................. 7
  2.1 X1 oscillator .................................................................................................................. 7
    2.1.1 Procedure for starting X1 oscillator operation ....................................................... 7
    2.1.2 Procedure for stopping X1 oscillator operation ..................................................... 8
    2.1.3 Procedure for switching CPU clock (pre-swiitchover clock: \( f_{MX} \)) ............... 9
    2.1.4 Cautions on the X1 oscillator ............................................................................... 10
  2.2 High-speed on-chip oscillator ...................................................................................... 10
    2.2.1 Procedure for starting high-speed on-chip oscillator operation ......................... 10
    2.2.2 Procedure for stopping high-speed on-chip oscillator ........................................ 11
    2.2.3 Procedure for switching CPU clock (pre-swiitchover clock: \( f_{IH} \)) ................. 12
    2.2.4 Cautions on high-speed on-chip oscillator ......................................................... 13
  2.3 Low-speed on-chip oscillator ...................................................................................... 13
    2.3.1 Procedure for starting low-speed on-chip oscillator ............................................. 13
    2.3.2 Procedure for stopping low-speed on-chip oscillator .......................................... 14
    2.3.3 Procedure for switching CPU clock (pre-swiitchover clock: \( f_{IL} \)) ................. 15
    2.3.4 Cautions on low-speed on-chip oscillator ........................................................... 16
    2.4 XT1 oscillator ............................................................................................................ 16
      2.4.1 Procedure for starting XT1 oscillator ................................................................. 16
      2.4.2 Procedure for stopping XT1 oscillator ............................................................... 17
      2.4.3 Procedure for switching CPU clock (pre-swiitchover clock: \( f_{SUB} \)) ......... 18
      2.4.4 Cautions on XT1 oscillator .............................................................................. 19
  2.5 PLL circuit .................................................................................................................. 19
    2.5.1 Procedure for starting oscillation of the PLL circuit ........................................... 20
    2.5.2 Procedure for stopping the oscillation of the PLL circuit .................................. 20
    2.5.3 Procedure for switching CPU clock (pre-swiitchover clock: \( f_{PLL} \)) ............. 21
    2.5.4 Cautions on PLL clock .................................................................................... 23
2.6 Watchdog-dedicated low-speed on-chip oscillator .......................................................... 24

3. Cautions on setting clocks ................................................................................................... 25
1. Hardware description

The RL78/F13 and RL78/F14 groups have the following clock generators. Table 1.1 and Table 1.2 show outlined specifications of each clock generator.

- X1 oscillator
- High-speed on-chip oscillator (high-speed OCO)
- PLL circuit
- XT1 oscillator
- Low-speed on-chip oscillator (low-speed OCO)
- Watchdog-dedicated low-speed on-chip oscillator (WDT-dedicated low-speed OCO)

Notes: 1. Products with 20, 30 and 32 pins have no XT1 oscillator.

2. Do not use the XT1 oscillator (subsystem clock) in grade-Y products.

Table 1.1 Outline of clocks used in RL78/F13 and RL78/F14 groups (1)

<table>
<thead>
<tr>
<th>After reset release</th>
<th>High-speed OCO clock</th>
<th>X1 clock</th>
<th>XT1 clock</th>
<th>PLL clock</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Starts oscillating</td>
<td>Stops</td>
<td>Stops</td>
<td></td>
</tr>
</tbody>
</table>

Clock frequency

- grade L
  Select one from the following:
  1/4/8/12/16/24/32/48/64 [MHz]
- grades K and Y
  Select one from the following:
  1/4/8/12/16/24/32/48 [MHz]

  1 to 20[MHz] grades L and K
  32.768[kHz] grade Y
  Usage disabled

  grade L
  Select one from the following:
  24/32/48/64 [MHz]
- grades K and Y
  Select one from the following:
  24/32/48 [MHz]

Application

- *CPU clock
  + peripheral hardware clock
- *CPU clock
  + peripheral hardware clock
- *CPU clock
  + peripheral hardware clock

  grade L
  32[MHz]
- grades K and Y
  24[MHz]

  grade L
  32[MHz]
- grades K and Y
  24[MHz]

Conditions for starting oscillation

- *after reset release
  *CSC.HIOSTOP=0
  *when STOP mode is released
- *CSC.MSTOP=0
  *when STOP mode is released
- *CSC.XTSTOP=0
- *PLLCTL.PLLON=1

Conditions for stopping oscillation

- *Execution of the STOP instruction
  *CSC.HIOSTOP=1
- *Execution of the STOP instruction
  *CSC.MTOP=1
- *CSC.XTSTOP=1
- *PLLCTL.PLLON=0

Method for deciding whether the clock oscillation has stabilizes

- —
- *Wait for the clock oscillation to stabilize using the OSTC register
- *Wait for the clock oscillation to stabilize using software
- *PLLSTS.LOCK=1
Table 1.2 Outline of clocks used in RL78/F13 and RL78/F14 (2)

<table>
<thead>
<tr>
<th></th>
<th>Low-speed OCO</th>
<th>WDT-dedicated low-speed OCO clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>After reset release</td>
<td>Stops</td>
<td>Select the operation (stops/oscillates) with the WDTON bit in the user option byte (000C0H/020C0H).</td>
</tr>
<tr>
<td>Clock frequency</td>
<td>15[kHz]</td>
<td>15[kHz]</td>
</tr>
<tr>
<td>CPU/peripheral hardware clock frequency (max.)</td>
<td>15[kHz]</td>
<td>Cannot be used as the CPU/peripheral hardware clock.</td>
</tr>
<tr>
<td>Application</td>
<td>*CPU clock</td>
<td>*WDT clock</td>
</tr>
<tr>
<td></td>
<td>*peripheral hardware clock</td>
<td></td>
</tr>
<tr>
<td></td>
<td>*clock monitor operation clock</td>
<td></td>
</tr>
<tr>
<td>Conditions for starting oscillation</td>
<td>*OSMC.WUTMMCK0=1, or CKSEL.SELLOSC=1</td>
<td>*Set the WDTON bit in the user option byte (000C0H/020C0H) to 1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conditions for stopping oscillation</td>
<td>*OSMC.WUTMMCK0=0 and CKSEL.SELLOSC=0</td>
<td>*Set the WDTON bit in the user option byte (000C0H/020C0H) to 0.</td>
</tr>
<tr>
<td>Method for deciding whether the clock oscillation stabilizes</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

1.1 CPU clock status transition

Figure 1.1 is a state transition chart of the CPU/peripheral hardware clock frequency (fCLK) and Figure 1.2 is a block diagram of the CPU clock. Table 1.3 shows the state transitions of the CPU clock.
Clock Generator

Source selector for main system clock

MCS, MCM0, PLLCTL, PLISTS, MDIV

fMX: High-speed system clock frequency (fH, fMX)
fX: X1 clock oscillation frequency
fx: High-speed on-chip oscillator clock frequency
fsub: Subsystem clock frequency (fx, fsub)
fXT: XT1 clock frequency
f xls: Low-speed oscillator clock frequency
fmain: Main system clock frequency
fPLL: Main PLL select clock frequency
fsub: CPU/Peripheral hardware clock frequency

MCM0, MCS, CSS, CLS: bits in the CKC register
SELLOSC: a bit in the CKSEL register
MDIV, PLLCTL, PLISTS: registers

Figure 1.2 Block diagram of CPU clock

Table 1.3 Transitions of CPU/Peripheral hardware clock (fCLK)

<table>
<thead>
<tr>
<th>Clock before switchover (fCLK)</th>
<th>fIH</th>
<th>fMX</th>
<th>fPLL(fIH)</th>
<th>fPLL(fMX)</th>
<th>fSUB (Note 1)</th>
<th>fIL</th>
</tr>
</thead>
<tbody>
<tr>
<td>fIH</td>
<td>fH</td>
<td>fMX</td>
<td>Switchover enabled (See Figure 2.4.)</td>
<td>Switchover enabled (See Figure 2.10.)</td>
<td>Prohibited to change</td>
<td>Switchover enabled (See Figure 2.4.)</td>
</tr>
<tr>
<td>fMX</td>
<td>fMX</td>
<td>fMX</td>
<td>Switchover enabled (See Figure 2.2.)</td>
<td>Prohibited to change</td>
<td>Switchover enabled (See Figure 2.12)</td>
<td>Switchover enabled (See Figure 2.2.)</td>
</tr>
<tr>
<td>fPLL(fIH)</td>
<td>Switchover enabled (See Figure 2.11.)</td>
<td>Prohibited to change</td>
<td>Switchover enabled (See Figure 2.12)</td>
<td>Switchover enabled (See Figure 2.12)</td>
<td>Prohibited to change</td>
<td></td>
</tr>
<tr>
<td>fPLL(fMX)</td>
<td>Prohibited to change</td>
<td>Switchover enabled (See Figure 2.12)</td>
<td>Prohibited to change</td>
<td>Switchover enabled (See Figure 2.12)</td>
<td>Prohibited to change</td>
<td></td>
</tr>
<tr>
<td>fSUB (Note 2)</td>
<td>Switchover enabled (See Figure 2.8.)</td>
<td>Prohibited to change</td>
<td>Switchover enabled (See Figure 2.12)</td>
<td>Prohibited to change</td>
<td>Prohibited to change</td>
<td></td>
</tr>
<tr>
<td>fIL</td>
<td>fIL</td>
<td>fIL</td>
<td>Prohibited to change</td>
<td>Switchover enabled (See Figure 2.8.)</td>
<td>Prohibited to change</td>
<td>Prohibited to change</td>
</tr>
</tbody>
</table>

fCLK: CPU/Peripheral hardware clock
fIH: High-speed on-chip oscillator clock
fMX: X1 clock
fPLL(fIH): PLL clock (clock source: High-speed on-chip oscillator)
fPLL(fMX): PLL clock (clock source: X1 clock)
fSUB: XT1 clock
fIL: Low-speed on-chip oscillator clock

Note: Products with 20, 30, and 32 pins are not provided with the XT1 clock.
2. Procedures for setting each clock generator

This chapter describes the overview of each clock generator and the procedures for setting the clock generators.

2.1 X1 oscillator

The X1 oscillator is a clock generator which is connected to the X1/X2 pin, or an external clock circuit which input to the EXCLK pin, and respectively generates the internal clock (the X1 clock oscillation frequency $f_X$, the external main system clock frequency $f_{EX}$). The clock generated by the X1 oscillator can be used as a clock source of the CPU clock. When the clock generated by the X1 oscillator is selected as a main system clock $f_{MAIN}$, the oscillation status of the X1 clock can be monitored by the clock monitor function. Figure 2.1 is a block diagram of the X1 oscillator.

![X1 oscillator block diagram](image)

**Figure 2.1 Block diagram of X1 oscillator**

2.1.1 Procedure for starting X1 oscillator operation

After reset release, the pins P121 and P122, to which the pins X1 and X2 are respectively assigned, function as an input port and the X1 oscillator stops its operation. To restart the X1 oscillator, set the following clock-related registers according to the procedure below:

1. Set the EXCLK, OSCSEL and AMPH bits in the CMC register. \(^{Note 1}\)

   CMC.[EXCLK:OSCSEL] = 01b (X1 oscillation mode) or 11b (external clock input mode)

2. Set the oscillation stabilization time with the OSTS register. \(^{Note 2}\)

3. Write 0 to the MSTOP bit in the CSC register to enable the X1 oscillator operation.

4. Monitor the OSTC register to check the elapsed oscillation stabilization time.

**Notes:**
- The CMC register can be written only once after reset release.
- The oscillation stabilization time varies depending on the resonator used. Please consult the resonator manufacture to select an appropriate oscillation stabilization time and adequately evaluate the oscillation on your system. When external clock input mode is selected, oscillation stabilization processing is not needed. However, be sure to use the input clock within the range specified in the electrical characteristics.

For the procedure for selecting the high-speed system clock frequency $f_{MX}$ as the CPU clock, refer to **Section 2.2** to **Section 2.5**.
2.1.2 Procedure for stopping X1 oscillator operation
To stop the oscillation of the X1 oscillator, set the following clock-related registers according to the procedure below:

(1) Set the CPU clock to a clock whose count source is not the high-speed system clock frequency \( f_{MX} \).

(2) Write 1 to the MSTOP bit in the CSC register to stop the operation of the X1 oscillator.
2.1.3  Procedure for switching CPU clock (pre-switchover clock: \( f_{MX} \))

Figure 2.2 shows the procedure for switching the X1 oscillation clock to any available clock as the CPU clock.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. CMC register</td>
<td>To oscillate high-speed OCO</td>
</tr>
<tr>
<td>2. OSMC register</td>
<td>To oscillate X1 clock</td>
</tr>
<tr>
<td>3. CKSEL register</td>
<td>To oscillate set XT1 clock</td>
</tr>
<tr>
<td>4. OSTS register</td>
<td>To oscillate low-speed OCO</td>
</tr>
<tr>
<td>5. HOCODIV register</td>
<td>*HOCODIV[2:0]=xxxb</td>
</tr>
<tr>
<td>6. CSC register</td>
<td>*HIOSTOP=0 (High-speed OCO starts operating.)</td>
</tr>
<tr>
<td>7. Wait for oscillation to stabilize</td>
<td>*Secure the stabilization wait time by a program.</td>
</tr>
<tr>
<td>8. MDIV register</td>
<td>*MDIV[2:0]=xxxb</td>
</tr>
<tr>
<td>9. CKC register (Switch the clock)</td>
<td>*MCM0=0 (Select ( f_{CM0}=f_{IN} ))</td>
</tr>
<tr>
<td>10. CKC register (Check switchover)</td>
<td>*Confirm MCS=0.</td>
</tr>
<tr>
<td>j: Settings not required</td>
<td>Settings not required</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
& f_{CLK}=f_{IH} \quad \text{(High-speed OCO clock has been selected)} \\
& f_{CLK}=f_{IL} \quad \text{(Low-speed OCO clock has been selected)} \\
& f_{CLK}=f_{SUB} \quad \text{(XT1 clock has been selected)} \\
& f_{CLK}=f_{MX} \quad \text{(X1 clock has been selected)}
\end{align*}
\]

Figure 2.2 Switchover from X1 clock
2.1.4 Cautions on the X1 oscillator

- When the X1 oscillator is not used, set both the EXCLK bit and the OSCSEL bit in the CMC register to 0 (input port mode).
- If the X1 clock oscillation frequency exceeds 10MHz, set the AMPH bit in the CMC register to 1. When the X1 clock oscillation frequency is in the range from 1 to 10MHz, setting the AMPH bit to 1 improves the oscillation margin.
- Before the X1 oscillator starts operating, the OSTS2 to OSTS0 bits in the OSTS register need to be set to a value equal to or greater than the count value which is to be checked by the OSTC register.
- When setting the MDIV register, make the frequency after division of fMP be within 1MHz to 32MHz (or 1MHz to 24MHz for grade-K and grade-Y products).
- If your X1 oscillator has any resonator, oscillation constants and oscillation stabilization time required for the X1 oscillator vary depending on the resonator used. Please consult the resonator manufacture to select an appropriate resonator and to determine the proper oscillator constants. Also adequately evaluate the oscillation on your system.

2.2 High-speed on-chip oscillator

After reset release, the high-speed on-chip oscillator is a clock source of the CPU clock. With option byte, one high-speed on-chip oscillator clock frequency can be selected from the incorporated clock sources 1/4/8/12/16/24/32/48/64 [MHz]. With a program, the high-speed on-chip oscillator clock frequency can be switched to 1/2/3/4/6/8/12/16/24/32/48/64 [MHz]. Figure 2.3 shows a block diagram of the high-speed on-chip oscillator.

![Block diagram of high-speed on-chip oscillator](image)

**Figure 2.3 Block diagram of high-speed on-chip oscillator**

2.2.1 Procedure for starting high-speed on-chip oscillator operation

After reset release, the high-speed on-chip oscillator starts operating. To start the oscillator after the oscillator operation is stopped, set the following clock-related registers according to the procedure below.

1. Write 0 to the HIOSTOP bit in the CSC register to start the high-speed on-chip oscillator.
2. Check the elapsed oscillation stabilization time. **Note:** Set the oscillation stabilization time with your software using a timer or software wait. The oscillation stabilization time varies depending on the settings to the FRQSEL4 bit in the option byte (000C2H/020C2H).

   FRQSEL4=0: (max) 65μs, FRQSEL4 =1: (max) 105μs

For the procedure for selecting the high-speed on-chip oscillation clock frequency fIH as the CPU clock, refer to Sections 2.1, 2.3 to 2.5.
2.2.2 Procedure for stopping high-speed on-chip oscillator
To stop the high-speed on-chip oscillator, set the following clock-related registers according to the procedure below.

(1) Set the CPU clock to a clock whose count source is not the high-speed on-chip oscillator clock frequency \( f_{IH} \).

(2) Write 1 to the HIINSTOP bit in the CSC register to stop the high-speed on-chip oscillator.
### Procedure for switching CPU clock (pre-switchover clock: \(f_{\text{IH}}\))

Figure 2.4 shows the procedure for switching the high-speed on-chip oscillator clock to any available clock as the CPU clock.

#### Figure 2.4 Switchover from high-speed on-chip oscillator clock

<table>
<thead>
<tr>
<th>Step</th>
<th>Register/Setting</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>CMC register</td>
<td>(f_{\text{CLC}} = f_{\text{IH}}) (High-speed OCP clock has been selected.)</td>
</tr>
<tr>
<td>2.</td>
<td>OSMC register</td>
<td>(f_{\text{CLC}} = f_{\text{MX}}) (X1 clock has been selected.)</td>
</tr>
<tr>
<td>3.</td>
<td>CKSEL register</td>
<td>(f_{\text{CLC}} = f_{\text{SUB}}) (XT1 clock has been selected.)</td>
</tr>
<tr>
<td>4.</td>
<td>OSTS register</td>
<td>(f_{\text{CLC}} = f_{\text{IL}}) (Low-speed OCO clock has been selected.)</td>
</tr>
<tr>
<td>5.</td>
<td>HOCODIV register</td>
<td>-</td>
</tr>
<tr>
<td>6.</td>
<td>CSC register</td>
<td>-</td>
</tr>
<tr>
<td>7.</td>
<td>Wait for oscillation to stabilize</td>
<td>-</td>
</tr>
<tr>
<td>8.</td>
<td>MDIV register</td>
<td>-</td>
</tr>
<tr>
<td>9.</td>
<td>CKC register</td>
<td>-</td>
</tr>
<tr>
<td>10.</td>
<td>CKC register</td>
<td>-</td>
</tr>
</tbody>
</table>

Note: To change the high-speed on-chip oscillator clock frequency \(f_{\text{IH}}\) to 64 or 48MHz from 32MHz or lower, set the MDIV[2:0] bits in the MDIV register to 001b (divided by 2) before setting the HOCODIV[2:0] bits in the HOCODIV register.

\[ f_{\text{CLK}} = f_{\text{SI}} \]

\[ f_{\text{CLK}} = f_{\text{MH}} \]

\[ f_{\text{CLK}} = f_{\text{SL}} \]
2.2.4 Cautions on high-speed on-chip oscillator

- When setting the MDIV register, make the frequency after division of the main/PLL select clock frequency $f_{MP}$ be within between 1MHz and 32MHz (or between 1MHz and 24MHz for grade-K and grade-Y products).

- To select the PLL clock as the CPU/peripheral hardware clock when the high-speed on-chip oscillator clock frequency is set to 48MHz or 24MHz with the FRQSEL[4:0] bits in the user option byte (000C2H/020C2H), do not set the CPU/peripheral hardware clock frequency ($f_{CLK}$) to 32MHz.

- When the high-speed on-chip oscillator clock frequency is set to 64MHz or 48MHz, set the MDIV register to 01h ($f_{MP}/2$).

- To restart the high-speed on-chip oscillator, the oscillation stabilization time is required. The oscillation stabilization time varies depending on the settings to the FRQSEL4 bit in the user option byte (000C2H/020C2H): [when FRQSEL4=0, the time is 65μs (max.), when FRQSEL4=1, the time is 105μs (max.)]. Implement the oscillation stabilization time with your software using timers or software wait.

2.3 Low-speed on-chip oscillator

The low-speed on-chip oscillator operates at 15kHz and can be selected as a clock source of the CPU clock. After reset release, the low-speed on-chip oscillator stops operating. In STOP mode, the low-speed on-chip oscillator does not stop its oscillation. Figure 2.5 shows a block diagram of the low-speed on-chip oscillator.

![Low-speed on-chip oscillator](image)

**Figure 2.5 Block diagram of low-speed on-chip oscillator**

2.3.1 Procedure for starting low-speed on-chip oscillator

After reset release, the low-speed on-chip oscillator stops operating. To restart the low-speed on-chip oscillator, set the clock-related registers according to the procedure below:

1. Write 1 to the SELLOSC bit in the CKSEL register to start the low-speed on-chip oscillator operation. **Note 1**

**Note:** The low-speed on-chip oscillator also starts its operation when the WUTMMCK0 bit in the OSMC register is set to 1.

For the procedure for selecting the low-speed on-chip oscillator clock frequency $f_{IL}$ as the CPU clock, refer to Sections 2.1 and 2.2.
2.3.2 Procedure for stopping low-speed on-chip oscillator

To stop the low-speed on-chip oscillator, set the following clock-related registers according to the procedure below.

(1) Set the CPU clock to a clock whose count source is not the high-speed on-chip oscillator clock frequency $f_{IH}$.

(2) Set the SELLOSC bit in the CKSEL register and the WUTTMMCK0 bit in the OSMC register to 0 to stop the low-speed on-chip oscillator operation.
2.3.3 Procedure for switching CPU clock (pre-switchover clock: fIL)

Figure 2.6 shows the procedure for switching the low-speed on-chip oscillator clock to any available clock as the CPU clock.

<table>
<thead>
<tr>
<th>Step</th>
<th>Register/Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>CMC register</td>
<td>*EXCLK, OSCSEL = 01b (X1 oscillation mode) *AMPH = x</td>
</tr>
<tr>
<td>2.</td>
<td>OSTS register</td>
<td>*OSTS[2:0]=xxxb</td>
</tr>
<tr>
<td>3.</td>
<td>HOCODIV register</td>
<td>*HOCODIV[2:0]=xxxb</td>
</tr>
<tr>
<td>4.</td>
<td>CSC register</td>
<td>*HSTOP= 0 (High-speed OCO starts operating.)</td>
</tr>
<tr>
<td>5.</td>
<td>Wait for oscillation to stabilize</td>
<td>Secure the oscillation stabilization wait time by a program.</td>
</tr>
<tr>
<td>6.</td>
<td>Set fMP</td>
<td>*SELPLL = 0 (Select clock through mode.)</td>
</tr>
<tr>
<td>7.</td>
<td>Check the clock switched from fMP</td>
<td>Conf</td>
</tr>
<tr>
<td>8.</td>
<td>MDIV register</td>
<td>*MDIV[2:0] = xxxb</td>
</tr>
<tr>
<td>9.</td>
<td>CKC register</td>
<td>*MCM0 = 0</td>
</tr>
<tr>
<td>10.</td>
<td>CKC register</td>
<td>*MCM0 = 1</td>
</tr>
<tr>
<td>11.</td>
<td>CKC register</td>
<td>*CSS = 0 (Select fCLK = fMAIN selected)</td>
</tr>
<tr>
<td>12.</td>
<td>CKC register</td>
<td>*CSS = 0 (Select fCLK = fMAIN selected)</td>
</tr>
</tbody>
</table>

Figure 2.6 Switchover from low-speed on-chip oscillator clock
2.3.4 Cautions on low-speed on-chip oscillator

- To select the low-speed on-chip oscillator for the count source of the timer RJ, write 1 to the WUTMCK0 bit in the OSMC register. Writing 1 to the WUTMCK0 bit starts the operation of the low-speed on-chip oscillator.

- When the low-speed on-chip oscillator clock is used as the peripheral hardware clock, operations of A/D converters and I2C cannot be guaranteed.

- When setting the TRD_CKSEL bit in the CKSEL register to 1 (select the subsystem/low-speed on-chip oscillator select clock frequency $f_{SL}$ as the timer RD clock), select $f_{SL}$ as the CPU clock before setting the TRD0EN bit in the PER1 register to 1.

2.4 XT1 oscillator

The XT1 oscillator is a clock generator which is connected to the XT1/XT2 pin, or an external circuit which inputs to the EXCLKS pin, and respectively generates the internal clock (the XT1 clock oscillation frequency $f_{XT}$, the external subsystem clock frequency $f_{EXS}$). The clocks generated by the XT1 oscillator can be used for the source of the CPU clock. Even after the transition to STOP mode, the XT1 oscillator does not stop oscillating. Figure 2.7 is a block diagram of the XT1 oscillator.

![Figure 2.7 Block diagram of XT1 oscillator](image)

2.4.1 Procedure for starting XT1 oscillator

After reset release, the pins P123 and P124 to which the pins XT1 and XT2 are respectively assigned, function as an input port pin and the XT1 oscillator stops operating. To restart the XT1 oscillator, set the following clock-related registers according to the procedure below:

1. Set the EXCLKS, OSCSELS, AMPHS1 and AMPHS0 bits in the CMC register. \( \text{Note 1} \)

   \[
   \text{CMC.[EXCLKS:OSCSELS]} \triangleq \{01 \text{b} \ (\text{XT1 oscillation mode}) \text{ or } 11 \text{b} \ (\text{external clock input mode}) \}
   \]

2. Set the SELLOSC bit in the CKSEL register to 0.

3. Write 0 to the XTSTOP bit in the CSC register to enable the XT1 oscillator operation.

4. Confirm the elapsed oscillation stabilization time. \( \text{Note 2} \)

Notes: 1. The CMC register can be written only once after reset release.

2. The stabilization oscillation time varies depending on the resonator used. Please consult the resonator manufacture to select an appropriate oscillation stabilization time and adequately evaluate the oscillation on your system. When external clock input mode is selected, oscillation stabilization time is not required. However, be sure to use the input clock within the specified range of electrical specifications.

For the procedure for selecting the subsystem clock frequency $f_{SUB}$ as the CPU clock, refer to Sections 2.1 and 2.2.
2.4.2 Procedure for stopping XT1 oscillator
To stop the XT1 oscillator, set the following clock-related registers according to the procedure below:

(1) Set the CPU clock to a clock whose count source is not the subsystem clock $f_{SUB}$.

(2) Write 1 to the XTSTOP bit in the CSC register to stop the XT1 oscillator operation.
### 2.4.3 Procedure for switching CPU clock (pre-switchover clock: $f_{\text{SUB}}$)

Figure 2.8 shows the procedure for switching the XT1 oscillator clock to any available clock as the CPU clock.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. CMC register</td>
<td>↓</td>
</tr>
<tr>
<td>2. OSTS register</td>
<td>↓</td>
</tr>
<tr>
<td>3. HOCODIV register</td>
<td>*HOCODIV[2:0] = xxxb</td>
</tr>
<tr>
<td>4. CSC register</td>
<td>*HIOSTOP = 0 (High-speed OCO starts operating.)</td>
</tr>
<tr>
<td>5. Wait for oscillation to stabilize</td>
<td>*Secure the oscillation stabilization time with a program.</td>
</tr>
<tr>
<td>6. Set $f_{\text{MP}}$ (Clock switchover)</td>
<td>*SELPLL = 0 (Select clock through mode.)</td>
</tr>
<tr>
<td>7. Check the clock switched from $f_{\text{MP}}$.</td>
<td>*Confirm SELPLLS = 0. ($f_{\text{MP}} = f_{\text{MAX}}$ selected)</td>
</tr>
<tr>
<td>9. CKC register (Select main system clock $f_{\text{MAX}}$)</td>
<td>*MCM0 = 0 ($f_{\text{MP}} = f_{\text{MAX}} = f_{\text{S}}$ selected)</td>
</tr>
<tr>
<td>10. CKC register (Check switchover)</td>
<td>*Confirm MCS = 0.</td>
</tr>
<tr>
<td>11. CKC register (Switch the clock.)</td>
<td>*CSS = 0 ($f_{\text{CLK}} = f_{\text{MAX}}(f_{\text{S}})$ selected)</td>
</tr>
<tr>
<td>12. CKC register (Check switchover)</td>
<td>*Confirm CLS = 0.</td>
</tr>
</tbody>
</table>

↓: settings not required
-: no settings

- $f_{\text{OCO}} = f_{\text{S}}$ (High-speed OCO clock has been selected.)
- $f_{\text{OCO}} = f_{\text{S}}$ (X1 clock has been selected.)
- $f_{\text{OCO}} = f_{\text{SUB}}$ (XT1 clock has been selected.)

Figure 2.8 Switchover from XT1 clock
2.4.4 Cautions on XT1 oscillator

- The subsystem clock (f_SUB) cannot be used in products with 20, 30 and 32 pins as well as grade-Y products.
- When the XT1 oscillator is not used, set both the EXCLKS and OSCSELS bits in the CMC register to 0 (input port mode).
- When a resonator is used in your XT1 oscillator, oscillation constants and the oscillation stabilization time vary depending on the resonator used. Thus, please consult the resonator manufacturer to select an appropriate resonator and the oscillation stabilization time and sufficiently evaluate oscillation with your system.
- When the subsystem clock is used as the peripheral hardware clock, the operations of the A/D converter and IICA are not guaranteed.
- When setting the TRD_CKSEL bit in the CKSEL register to 1 (select the subsystem/low-speed on-chip oscillator select clock frequency f_SL as the timer RD clock), select f_SL as the CPU clock before setting the TRD0EN bit in the PER1 register to 1.

2.5 PLL circuit

The clocks that can be obtained by multiplying the high-speed system clock frequency (f_MX) or the high-speed on-chip oscillator clock frequency (f_HI) are used as a clock source of the CPU clock. Figure 2.9 shows a block diagram of the PLL circuit.

![PLL circuit block diagram](image-url)

**Figure 2.9 Block diagram of PLL circuit**
2.5.1 Procedure for starting oscillation of the PLL circuit

After reset release, the PLL circuit stops operating. To restart the PLL circuit operation, set the following clock-related registers according to the procedure below:

(1) Set the PLLMUL, PLLDIV0, PLLDIV1 and LCKSEL[1:0] bits in the PLLCTL register.

(2) Wait for the selection of the PLL multiplication value to become effective. (After setting the PLLMUL bit, wait for at least 1μs.)

(3) Write 1 to the PLLON bit in the PLLCTL register to enable the operation of the PLL circuit.

(4) Monitor the LOCK bit in the PLLSTS register to check the elapsed oscillation stabilization time. (The desired time is 40μs or more.)

Note: 1. Refer to Table 2.1 for the combination of input clocks and output clocks that can be used in the PLL circuit. Do not apply any combination of clocks which is not specified in the table.

2.5.2 Procedure for stopping the oscillation of the PLL circuit

To stop the oscillation of the PLL circuit, set the following clock-related registers according to the procedure below:

(1) Write 0 to the SELPLL bit in the PLLCTL register to enter clock through mode.

(2) With the SELPLLS bit in the PLLSTS register, check that the transition to clock through mode has been completed.

(3) Write 0 to the PLLON bit in the PLLCTL register to stop the PLL circuit.
2.5.3 Procedure for switching CPU clock (pre-switchover clock: f_PLL)

Figure 2.10 shows the procedure for switching the PLL clock to any available clock as the CPU clock. Figure 2.11 shows the procedure for switching the clock source f_MH or f_MX to the PLL clock.

![Diagram](image-url)

Figure 2.10 Switchover from PLL clock
To oscillate X1 clock

1. PLLCTL register (Set PLL multiplication.)
   *PLLDIV1 = x
   *PLLDIV0 = x
   *LCKSEL[1:0] = x'b (Set the lock-up wait counter.)
   *PLLMUL = x
   *Wait for the selection of the PLL multiplication value to become effective (1μs)

2. PLLCTL register (PLL starts operating.)
   *PLLON = 1

3. PLLSTS register (Check PLL locked state.)
   *Confirm LOCK = 1.

4. MDIV register
   *MDIV[2:0] = xxxb

5. PLLCTL register (Clock switchover)
   *SELPLL = 1
   \( f_{CLK} = f_{PLL}(f_{MX}) \)
   *SELPLL = 1
   \( f_{CLK} = f_{PLL}(f_{MX}) \)

6. PLLSTS register (Check switchover)
   *Confirm SELPLL = 1.

\( f_{CLK} = f_{PLL} \) selected
(PLL clock whose clock source is high-speed OCO clock)

\( f_{CLK} = f_{PLL} \) selected
(PLL clock whose clock source is X1 clock)

Figure 2.11 Switchover to PLL clock
2.5.4 Cautions on PLL clock

- When setting the MDIV register, make the frequency after division of the main/PLL select clock frequency \(f_{MP}\) be within between 1MHz and 32MHz (or 1MHz to 24MHz for grade-K and grade-Y products). In addition, when the PLLDIV1 bit in the PLLCTL register is set to 1 \((f_{PLL}>32MHz)\), set the MDIV register to 01h \((f_{MP}/2)\).

- For a combination of input clocks and output clocks that can be used as the PLL clock, refer to Table 1. Do not apply any combination which is not specified in Table 2.1.

**Table 2.1 List of multiplied clocks**

<table>
<thead>
<tr>
<th>Input clock</th>
<th>PLLCTL register</th>
<th>Output clock</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PLLMUL</td>
<td>PLLDIV0</td>
</tr>
<tr>
<td>4MHz±2%</td>
<td>0 (multiplied by 12)</td>
<td>0 (divided by 2)</td>
</tr>
<tr>
<td></td>
<td>1 (multiplied by 16)</td>
<td>0 (divided by 2)</td>
</tr>
<tr>
<td>8MHz±2%</td>
<td>0 (multiplied by 12)</td>
<td>1 (divided by 4)</td>
</tr>
<tr>
<td></td>
<td>0 (multiplied by 12)</td>
<td>0 (divided by 2)</td>
</tr>
<tr>
<td></td>
<td>1 (multiplied by 16)</td>
<td>1 (divided by 4)</td>
</tr>
<tr>
<td></td>
<td>1 (multiplied by 16)</td>
<td>1 (divided by 4)</td>
</tr>
</tbody>
</table>

**Note:**
1. When the FRQSEL4 bit in the user option byte (000C2H/020C2H) is 1, set the PLLDIV1 bit to 0.

- When the clock monitor detects that the main system/PLL select clock has been stopped, the SELPLLS bit in the PLLSTS register is set to 0 (clock through mode). However, the value of the SELPLL bit in the PLLCTL register remains 1 (PLL-clock-selected mode).

- When selecting the PLL clock as the CPU/peripheral hardware clock while the high-speed on-chip oscillator clock frequency is set to 48MHz or 24MHz with the FRQSEL[4:0] bits in the user option byte (000C2H/020C2H), do not set the CPU/peripheral hardware clock frequency \((f_{CLK})\) to 32MHz.
2.6  Watchdog-dedicated low-speed on-chip oscillator

The watchdog-dedicated low-speed on-chip oscillator (WDT-dedicated low-speed on-chip oscillator) is used as the watchdog timer clock. When the WDTON bit in the user option byte (000C0H/020C0H) is set to 1, the WDT-dedicated low-speed on-chip oscillator starts oscillating after reset release, and the watchdog timer starts counting.

When the WDTON and WDSTBYON bits in the user option byte (000C0H/020C0H) have been respectively set to 1 and 0, and also the operating mode is shifted to HALT/STOP/SNOOZE mode, the WDT-dedicated low-speed on-chip oscillator stops oscillating. When HALT/STOP/SNOOZE mode is released, the WDT-dedicated low-speed on-chip oscillator restarts oscillating.

Figure 2.12 is a block diagram of the WDT-dedicated on-chip oscillator.

![WDT-dedicated on-chip oscillator](image)

**Figure 2.12 Block diagram of WDT-dedicated on-chip oscillator**
3. **Cautions on setting clocks**

- Peripheral functions using the CPU/peripheral hardware clock need to be stopped before the CPU/peripheral hardware clock is changed.
- The CMC register can be written only once by an 8-bit memory manipulation instruction after reset release. If the setting to the CMC register is already done, the procedure for clock switchover and the following steps can be omitted.
- Execute the write access to the CMC, CSC, CKC, OSTS, and MDIV registers when the GCSC bit in the IAWCTL register is set to 0.
Website and Support <website and support, ws>

Renesas Electronics Website
http://www.renesas.com/

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

All trademarks and registered trademarks are the property of their respective owners.
## Revision History

<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>Sep. 1, 2016</td>
<td>1st Edition</td>
</tr>
</tbody>
</table>
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.

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 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.