1. Abstract

This document describes procedures for the following mode transitions, and setting and canceling slow read mode in the M16C/63 Group:

- Transition from 40 MHz on-chip oscillator mode (fOCO-F divided by 1) to 125 kHz on-chip oscillator mode (fOCO-S divided by 1)
- Transition from 125 kHz on-chip oscillator mode (fOCO-S divided by 1) to 40 MHz on-chip oscillator mode (fOCO-F divided by 1)
- Transition from 125 kHz on-chip oscillator mode to 125 kHz on-chip oscillator low power mode
- Transition from 125 kHz on-chip oscillator low power mode to 125 kHz on-chip oscillator mode
- Setting and canceling slow read mode

2. Introduction

The application example described in this document applies to the following microcomputer (MCU):

- MCU: M16C/63 Group

This application note can be used with other M16C Family MCUs which have the same special function registers (SFRs) as the above group. Check the manual for any modifications to functions. Careful evaluation is recommended before using the sample code described in this application note.
3. Clock Mode Transition Procedure

Figure 3.1 shows the transition procedure between 40 MHz on-chip oscillator and 125 kHz on-chip oscillator low power mode.

Figure 3.2 to Figure 3.5 show transition procedures for each mode.

Figure 3.6 shows the procedure for setting and canceling slow read mode.

Wait time until the main clock oscillation stabilizes varies depending on the oscillation circuit used. Use the wait time recommended by the crystal unit manufacturer.

\( \text{tsu(fOCO40M)} \): Wait time until 40 MHz on-chip oscillator stabilizes 
\( \text{tsu(fOCO-S)} \): Wait time until 125 kHz on-chip oscillator stabilizes

Refer to the “Electrical Characteristics” in the User’s Manual: Hardware (Hardware Manual) for details on \( \text{tsu(fOCO40M)} \) and \( \text{tsu(fOCO-S)} \).

---

**Figure 3.1** Transition between 40 MHz On-Chip Oscillator Mode and 125 kHz On-Chip Oscillator Low Power Mode
Figure 3.2 Transition from 40 MHz On-Chip Oscillator Mode (fOCO-F Divided by 1) to 125 kHz On-Chip Oscillator Mode

- CM14 = 0
- Wait for tsu(OCO-S) until the 125 kHz on-chip oscillator stabilizes
- CM06 = 1
- FRA01 = 0
- CM06 = 0
- PRC0 = 0
- END

(1) Disable write protection
(2) Start oscillating the 125 kHz on-chip oscillator
(3) Wait time until the 125 kHz on-chip oscillator stabilizes
(4) Switch fOCO-F divided by 1 to fOCO-F divided by 8
(5) Switch the 40 MHz on-chip oscillator to the 125 kHz on-chip oscillator
(6) Switch divide-by-8 mode to no division mode
(7) Enable write protection

Figure 3.3 Transition from 125 kHz On-Chip Oscillator Mode to 125 kHz On-Chip Oscillator Low Power Mode

- PRC0 = 1
- CM20 = 0
- CM05 = 1
- FRA00 = 0
- PRC0 = 0
- END

(1) Disable write protection
(2) Disable the oscillator stop/restart detect function
(3) Stop the main clock
(4) Stop oscillating the 40 MHz on-chip oscillator
(5) Enable write protection
Transition from 125 kHz on-chip oscillator low power mode to 125 kHz on-chip oscillator mode

PRC0 = 1  (1) Disable write protection
CM15 = 1  (2) Set XIN-XOUT drive capacity to high
CM05 = 0  (3) Start oscillating the main clock
Wait until the main clock stabilizes (4) Wait time until the main clock stabilizes
FRA00 = 1  (5) Start oscillating the 40 MHz on-chip oscillator
Wait for tsu(fOCO40M) until the 40 MHz on-chip oscillator stabilizes (6) Wait until the 40 MHz on-chip oscillator stabilizes
PRC0 = 0  (7) Enable write protection
END

Figure 3.4 Transition from 125 kHz On-Chip Oscillator Low Power Mode to 125 kHz On-Chip Oscillator Mode

Transition from 125 kHz on-chip oscillator mode to 40 MHz on-chip oscillator mode

PRC0 = 1  (1) Disable write protection
FRA00 = 1  (2) Start oscillating the 40 MHz on-chip oscillator
Wait for tsu(fOCO40M) until the 40 MHz on-chip oscillator stabilizes (3) Wait time until the 40 MHz on-chip oscillator stabilizes
CM06 = 1  (4) Switch no division mode to divide-by-8 mode
FRA01 = 1  (5) Switch the 125 kHz on-chip oscillator to the 40 MHz on-chip oscillator
CM06 = 0  (6) Switch fOCO-F divided by 8 to fOCO-F divided by 1
PRC0 = 0  (7) Enable write protection
END

Figure 3.5 Transition from 125 kHz On-Chip Oscillator Mode to 40 MHz On-Chip Oscillator Mode (fOCO-F Divided by 1)
Slow read mode can be used when \( f(\text{BCLK}) \) is less than or equal to \( f(\text{SLOW}_R) \), and the PM17 bit in the PM1 register is 1 (one wait).

\( f(\text{SLOW}_R) \): Operation frequency in slow read mode

Refer to the “Electric Characteristics” in the User's Manual: Hardware (Hardware Manual) for the maximum standard value of the operation frequency in slow read mode.

No wait time is required when the CPU clock source is the 125 kHz on-chip oscillator clock or the sub clock.

---

**Figure 3.6 Setting and Canceling Slow Read Mode**
4. Sample Code

A sample code can be downloaded from the Renesas Electronics website. To download, click “Application Notes” in the left-hand side menu of the M16C Family page.

4.1 Sample Code Operation

The sample code executes functions (1) to (6) below in order. Refer to 4.2 Function Tables for details of each function.

(1) CPU initialization
(2) Transition from 125 kHz on-chip oscillator mode to 40 MHz on-chip oscillator mode
(3) Transition from 40 MHz on-chip oscillator mode to 125 kHz on-chip oscillator mode
(4) Transition from 125 kHz on-chip oscillator mode to 125 kHz on-chip oscillator low power mode
(5) Setting and canceling slow read mode
(6) Transition from 125 kHz on-chip oscillator low power mode to 125 kHz on-chip oscillator mode

The settings in the sample code are as follows:

• Operation frequency is approximately 20 MHz in 40 MHz on-chip oscillator mode.
• Wait time until the 40 MHz on-chip oscillator stabilizes is approximately 5 ms when the CPU clock is 125 kHz.
• Wait time until the 125 kHz on-chip oscillator stabilizes is approximately 50 μs when the CPU clock is 20 MHz.
• Wait time until the main clock stabilizes is approximately 100 ms when the CPU clock is 125 kHz.

In the sample code, the 125 kHz on-chip oscillator is used as the CPU clock source when entering slow read mode and the PM17 bit remains 0 (no wait state).
### 4.2 Function Tables

Function Tables for This Document

<table>
<thead>
<tr>
<th>Declaration</th>
<th>void foco125k_from_foco40m(void)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outline</td>
<td>Transition from 40 MHz on-chip oscillator mode to 125 kHz on-chip oscillator mode</td>
</tr>
<tr>
<td>Argument</td>
<td>None</td>
</tr>
<tr>
<td>Variable</td>
<td>None</td>
</tr>
<tr>
<td>Returned value</td>
<td>None</td>
</tr>
<tr>
<td>Function</td>
<td>Switch the CPU clock from 40 MHz on-chip oscillator mode (fOCO-F divided by 1) to 125 kHz on-chip oscillator mode (fOCO-S divided by 1).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Declaration</th>
<th>void foco40m_from_foco125k(void)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outline</td>
<td>Transition from 125 kHz on-chip oscillator mode to 40 MHz on-chip oscillator mode</td>
</tr>
<tr>
<td>Argument</td>
<td>None</td>
</tr>
<tr>
<td>Variable</td>
<td>None</td>
</tr>
<tr>
<td>Returned value</td>
<td>None</td>
</tr>
<tr>
<td>Function</td>
<td>Switch the CPU clock from 125 kHz on-chip oscillator mode (fOCO-S divided by 1) to 40 MHz on-chip oscillator mode (fOCO-F divided by 1).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Declaration</th>
<th>void lowpower125k_from_foco125k(void)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outline</td>
<td>Transition from 125 kHz on-chip oscillator mode to 125 kHz on-chip oscillator low power mode</td>
</tr>
<tr>
<td>Argument</td>
<td>None</td>
</tr>
<tr>
<td>Variable</td>
<td>None</td>
</tr>
<tr>
<td>Returned value</td>
<td>None</td>
</tr>
<tr>
<td>Function</td>
<td>Stop the main clock and 40 MHz on-chip oscillator, and switch the CPU clock from 125 kHz on-chip oscillator mode to 125 kHz on-chip oscillator low power mode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Declaration</th>
<th>void foco125k_from_lowpower125k(void)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outline</td>
<td>Transition from 125 kHz on-chip oscillator low power mode to 125 kHz on-chip oscillator mode</td>
</tr>
<tr>
<td>Argument</td>
<td>None</td>
</tr>
<tr>
<td>Variable</td>
<td>None</td>
</tr>
<tr>
<td>Returned value</td>
<td>None</td>
</tr>
<tr>
<td>Function</td>
<td>Start oscillating the main clock and 40 MHz on-chip oscillator, and switch the CPU clock from 125 kHz on-chip oscillator low power mode to 125 kHz on-chip oscillator mode.</td>
</tr>
</tbody>
</table>
### Function Table for `slow_read_setup(void)`

<table>
<thead>
<tr>
<th>Declaration</th>
<th>void slow_read_setup(void)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outline</td>
<td>Setting and canceling slow read mode</td>
</tr>
<tr>
<td>Argument</td>
<td>None</td>
</tr>
<tr>
<td>Variable</td>
<td>None</td>
</tr>
<tr>
<td>Returned value</td>
<td>None</td>
</tr>
<tr>
<td>Function</td>
<td>Configure settings for slow read mode, execute slow_read(), and cancel slow read mode. This function does not include processes to set the CPU clock to f(SLOW_R) or less, or restore the CPU clock. Execute this function after setting the CPU clock to f(SLOW_R) or less. Then restore the CPU clock. When this function is executed, the PM17 bit remains 0 (no wait state). Set the PM17 bit to 1 (wait state (1 wait)) as required.</td>
</tr>
</tbody>
</table>

### Function Table for `slow_read(void)`

<table>
<thead>
<tr>
<th>Declaration</th>
<th>void slow_read(void)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outline</td>
<td>Processing in slow read mode</td>
</tr>
<tr>
<td>Argument</td>
<td>None</td>
</tr>
<tr>
<td>Variable</td>
<td>None</td>
</tr>
<tr>
<td>Returned value</td>
<td>None</td>
</tr>
<tr>
<td>Function</td>
<td>Called from slow_read_setup(). Add a program to be processed in slow read mode.</td>
</tr>
</tbody>
</table>

### Function Table for `mcu_init(void)`

<table>
<thead>
<tr>
<th>Declaration</th>
<th>void mcu_init(void)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outline</td>
<td>CPU initialization</td>
</tr>
<tr>
<td>Argument</td>
<td>None</td>
</tr>
<tr>
<td>Variable</td>
<td>None</td>
</tr>
<tr>
<td>Returned value</td>
<td>None</td>
</tr>
<tr>
<td>Function</td>
<td>Set to single-chip mode. Switch the CPU clock from 125 kHz on-chip oscillator mode divided-by-8 to 125 kHz on-chip oscillator mode divided-by-1.</td>
</tr>
</tbody>
</table>
5. Reference Documents

M16C/63 Group User’s Manual: Hardware (Hardware Manual) Rev.1.00
The latest version can be downloaded from the Renesas Electronics website.

Technical Update/Technical News
The latest information can be downloaded from the Renesas Electronics website.

M16C Series/R8C Family C Compiler Package V.5.45 C Compiler User Manual Rev.2.00
The latest version can be downloaded from the Renesas Electronics website.

Website and Support

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

Inquiries
http://www.renesas.com/inquiry
## Revision History

<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>2010.07.01</td>
<td>First edition issued</td>
</tr>
</tbody>
</table>

All trademarks and registered trademarks are the property of their respective owners.
General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins
   Handle unused pins in accord 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 one with a different part number, confirm that the change will not lead to problems.
   - The characteristics of MPU/MCU in the same group but having different part numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different part numbers, implement a system-evaluation test for each of the products.