AN205338

How to do CR Trimming with F²MC-8FX Commodity Family MB95200H/210H Series

This application note describes the main CR clock configuration and the CR trimming function.

1 Introduction

This application note describes the main CR clock configuration and the CR trimming function.

2 Main CR Oscillator Description

This chapter describes the configuration of CR Oscillator.

2.1 General

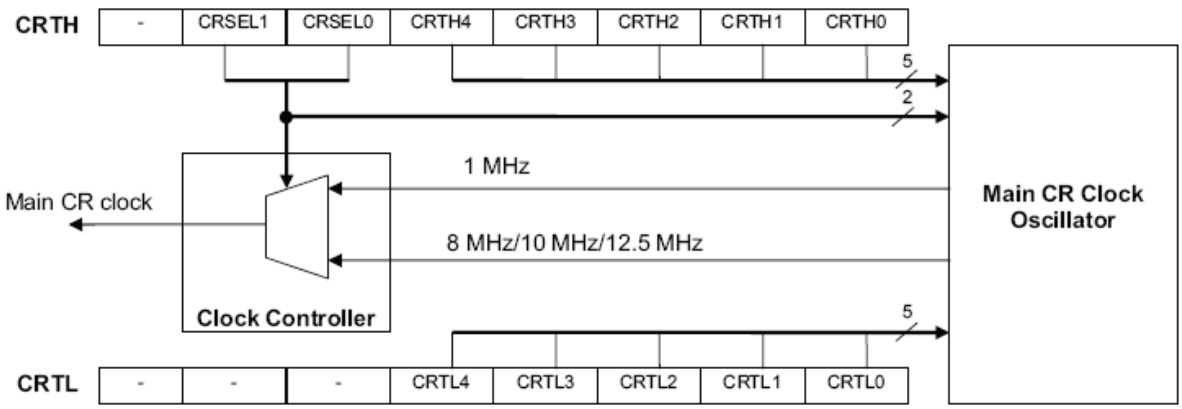
MB95200H/210H series MCU contains an internal main CR oscillator, and the CR oscillator can provide main CR clock to the MCU. The frequency of the CR oscillator is controlled by CR control registers. The basic frequency of the CR oscillator is set to 1 MHz when the MCU leaves factory.

The CR oscillator can also provide multiple frequencies: 1 MHz /8 MHz/10 MHz/12.5 MHz to the main CR clock. (See Figure 1)

2.2 CR Oscillator Control

The following block diagram shows the CR oscillator interface.

Figure 1. Main CR Clock Configuration



CRTH [4:0] and CRTL [4:0] are used to tune the frequency of CR oscillator. CRTH [6:5] is used to control the output of CR oscillator to the main CR clock.

CRTH [4:0] can tune the CR frequency coarsely (step length: 20 KHz ~ 8 KHz); CRTL [4:0] can tune the CR frequency finely (step length: 8 KHz ~ 1.6 KHz). Generally speaking, the basic CR frequency can be set from 1.9 MHz to 400 KHz.

When CRTH [4:0] and CRTL [4:0] are set to "00000_B", the CR oscillator is tuned to the fastest frequency.

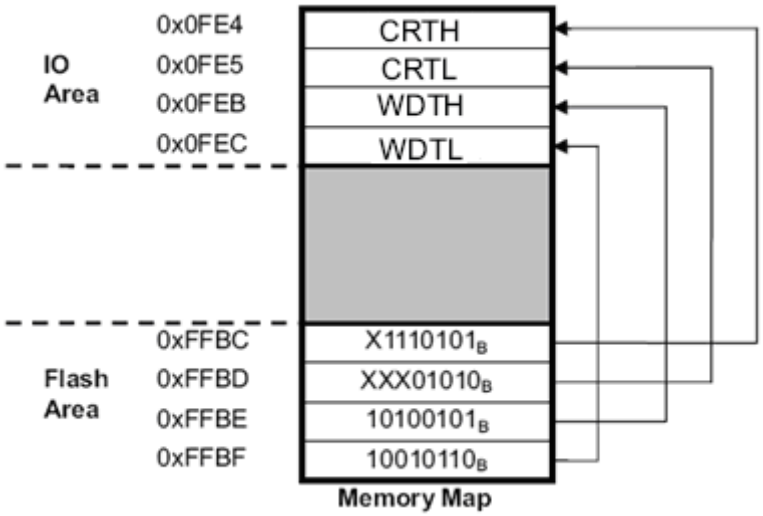
When CRTH [4:0] and CRTL [4:0] are set to "11111_B", the CR oscillator is tuned to the slowest frequency.

CRTH [6:5] is set to "11_B" when the MCU leaves factory, so the initial frequency of the main CR clock becomes 8 MHz correspondingly.

2.3 CR Control Register

CR control register is a 16-bit register and marked as CRTH and CRTL. The CR oscillator shall keep with 1 MHz frequency for the debug usage. (See Chapter 3), and the CR control register is stored in a NON-Volatile register. The following block diagram shows the CR control register interface.

Figure 2. NVR Registers Configuration



After reset, the data in NVR will be fetched from the flash area and stored into the I/O area. Once users modify the value CRTH and CRTL in the I/O area, the value could be recovered after a reset.

CRTH [4:0] and CRTL [4:0] in the flash area are reserved. Writing codes or constant data to CRTH [4:0] and CRTL [4:0] have no effect.

Even if CONST section is set in the user code as follows, CRTH [4:0] and CRTL [4:0] can keep their original values and cannot be changed to “0808H”.

```
.section NVR,CONST,LOCATE=H'FFBC    //  set CRTH & CRTL
 .DATA.W 0x0808
```

CRTH [6:5] can be modified by users.

3 CR Trimming Process

This chapter describes the CR trimming principle.

3.1 Necessity of the CR Trimming Process

The CR trimming function is used by the BGM adapter to control the CR frequency by adjusting CR control registers.

In debug mode, the CR oscillator is used to generate 62,500 bps baud rate signal to support the 1-line UART communication. And the baud rate setting is base on 1 MHz CR frequency.

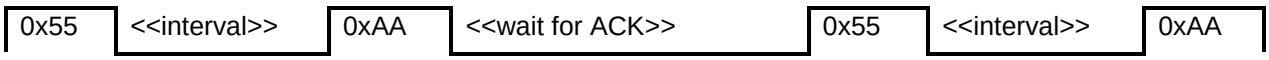
If the frequency of the CR clock is out of range: 1 MHz ± 3.75%, the target MCU will lose communication with the BGM adapter. And the debug will be stopped.

To start a debug, the BGM adapter needs to identify the CR clock status; and needs to adjust the CR frequency in case that the CR frequency is changed. Then the BGM adapter starts a CR trimming process after the target MCU is powered on.

3.2 CR Trimming Controlled by the BGM Adapter

3.2.1 The Format of the CR Trimming Command

The BGM adapter sends two bytes (0x55AA) to the target MCU as the CR trimming command. There is an interval (3.5 ms) between 0x55 and 0xAA. Then the BGM adapter waits a period of time (2.7ms) for the ACK (1 byte) from the target MCU. Repeat the command for a CR status check and a CR trimming.



3.2.2 CR Status Check

The BGM adapter checks the CR clock status after it detects the target MCU power on each time. Firstly, the BGM adapter attempts to send the trimming command 10 times after it has detected the target MCU power on.

If the BGM adapter receives a right ACK (data “51H”) from the target MCU in this 10 times commands, the BGM adapter will indicate “CR_OK” to the Softune.

If the BGM adapter receives a wrong ACK (data other than “51H”) from the target MCU in this 10 times commands, The BGM adapter will indicate “TRIM_FAIL” to the Softune.

If there is no ACK received by the BGM adapter in this 10 times commands, the BGM adapter will start a CR trimming process to adjust the CR frequency. (See Section 3.2.3)

3.2.3 CR Trimming Process

To do a CR trimming process, the BGM adapter continues to send the CR trimming command to the target MCU until it receives an ACK or retry times are out of range (i.e. the CR trimming fails).

If the NVR is broken in the target MCU, the target MCU gets a wrong CR frequency after the power on. Then the target MCU cannot receive the right trimming command.

The target MCU starts the CR trimming process after it receives trimming commands with frame errors for several times.

The Target MCU sets CRTH [4:0] and CRTL [4:0] to “00000_B” and waits for the trimming command. After the target MCU receives a wrong command each time, it increases the value of CRTH [4:0] and CRTL [4:0] as a regulation. If the target MCU receives the right trimming command, it will do a little adjustment, then return “A1H” to the BGM adapter as ACK.

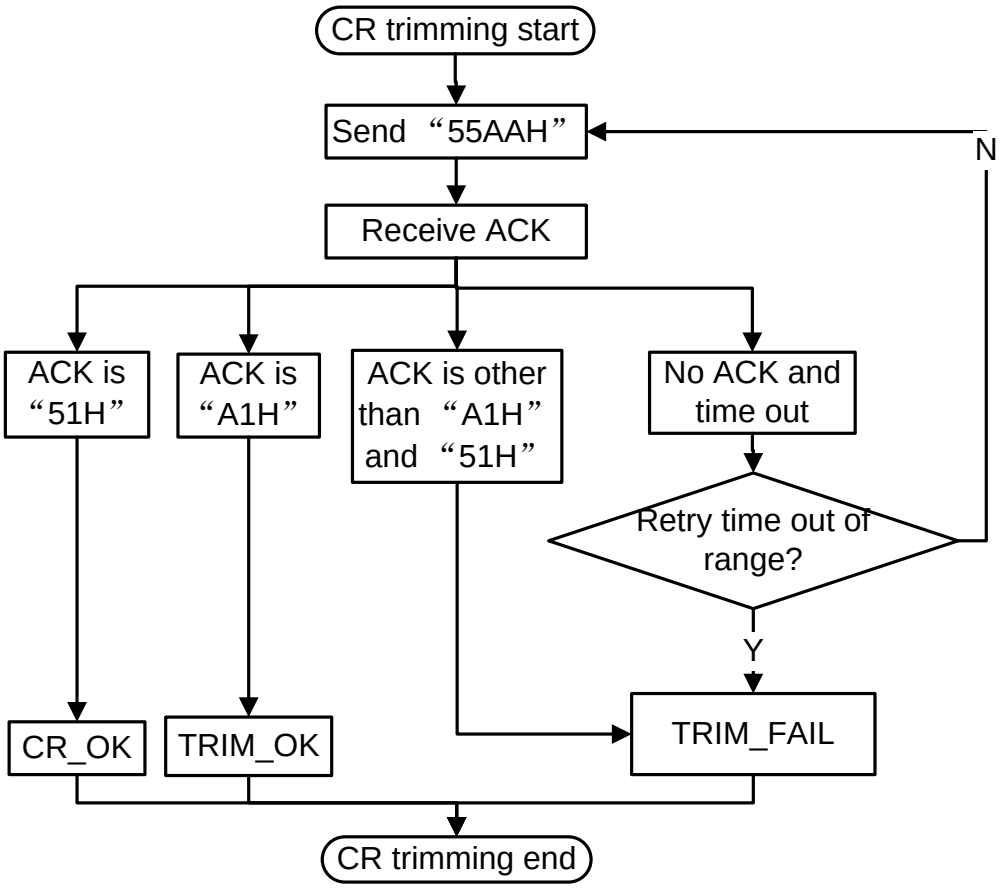
If the BGM adapter receives ACK (data “A1H”) from the target MCU during the CR trimming process, the BGM adapter will indicate “TRIM_OK” to the Softune. In normal case, the CR frequency can be adjusted to 1 MHz±3% in this situation. And the precision can ensure the 1-line UART communication.

If the CR trimming process fails, the BGM adapter will indicate “TRIM_FAIL” to the Softune.

3.2.4 Simple Flowchart of the CR Trimming Process

The following diagram is the simple CR trimming flowchart based on the BGM adapter side.

Figure 3. CR Trimming Flowchart



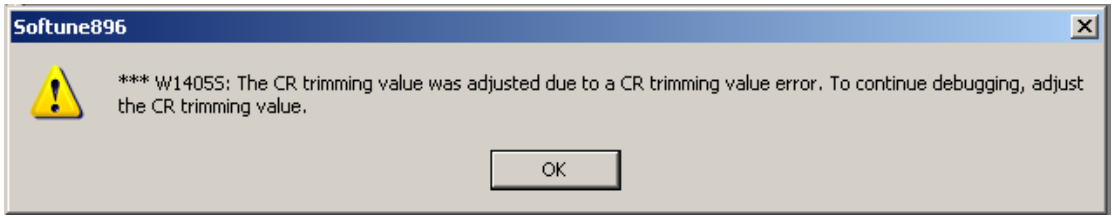
3.3 CR Trimming Result Dialogs.

The Softune will check the CR clock status and show the result to users when the debug is started:

If the CR frequency is OK, the Softune will continue debug normally.

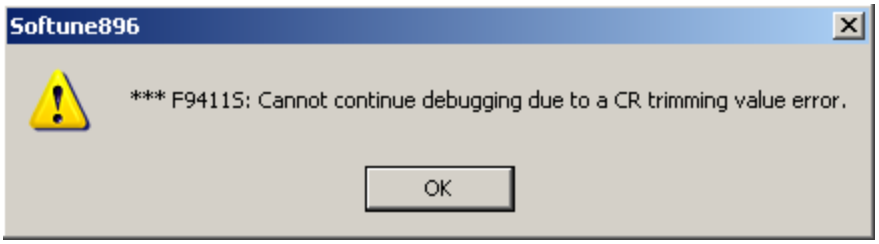
If the CR trimming process finishes, the Softune will show the following dialog:

Figure 4. End of the CR trimming



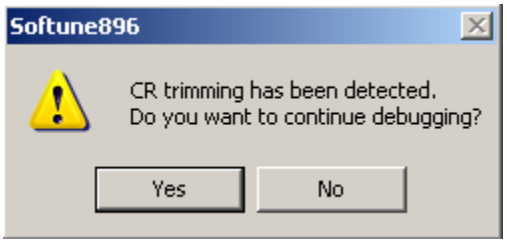
If the CR trimming process fails, the Softune will show the following dialog:

Figure 5. Break of the NVR and Failure of the CR trimming



If the BGM adapter is busy doing the CR trimming process, the Softune will show the following dialog until the CR trimming process finishes:

Figure 6. CR Trimming in Process



3.4 Precautions on CR Trimming Result

If the target MCU runs in the free-run mode, the BGM adapter may receive no response from the target MCU, then the BGM adapter will indicate “TRIM_FAIL” to the Softune. If the target MCU runs in the debug mode and the BGM adapter resets at once, then the BGM adapter may indicate “TRIM_FAIL” to the Softune too. So if the CR trimming process fails, please re-connect debug tools and try again. If the CR trimming process fails for several times, that means the target MCU is broken.

If the CR trimming process finishes, the CR clock may not support very accurate 1 MHz frequency. But it will not affect the 1-line UART communication. If the accurate CR frequency is needed in application, please contact to the shop which bought it or adjust CR trimming value according to next chapter.

4 Advanced Operation after the CR Trimming

This chapter describes the advanced operations of the CR trimming.

4.1 Measurement of the CR Frequency

Use composite timer output CR clock to an I/O port (e.g. 8 MHz/8) in a user code, then the CR frequency is measurable.

Change of the CR trimming value in the user code cannot modify the CR trimming value in the flash area. But this operation may cause that the target MCU cannot communicate with the BGM Adapter temporarily.

4.2 CR Clock Repair.

After the CR trimming process, the precision of CR frequency may be changed to 1 MHz±3%. If the precision is not enough, CR clock can be repaired manually. The process is as follows:

When start a debug with a dedicated target MCU the first time, record the value of CR control registers (flash area) of this MCU. (The value can be read in the memory window.)

If the CR frequency is not very approximate to 1 MHz after the CR trimming process, please modify the CR control registers (I/O area) to the same value as recorded before. After the Softune downloads a program to flash memory, the value of CR control registers in the flash area will be repaired.

Document History

Document Title: AN205338 - How to do CR Trimming with F²MC-8FX Commodity Family MB95200H/210H Series

Document Number: 002-05338

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	—	HUAL	03/18/2008	Initial release.
*A	5260442	HUAL	08/26/2016	Migrated Spansion Application note from MCU-AN-500014-E-10 to Cypress format.
*B	5844334	AESATMP9	08/04/2017	Updated logo and copyright.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2008-2017. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.