GigaDevice Semiconductor Inc.

Arm® Cortex®-M4 32-bit MCU

Application Note
AN029

Table of Contents

List of Figures

List of Tables

1. Introduction

GD32F3xx and GD32F403 with Cortex-M4 kernel are launched by GD in recent years. These two series products can run 103 program directly. But the 103 program doesn't have the DSP instruction set. Is there a way to let the users to use the DSP instruction set and FPU when the program is based on 103? The answer is yes. The detailed method is show as below.

2. Operation steps

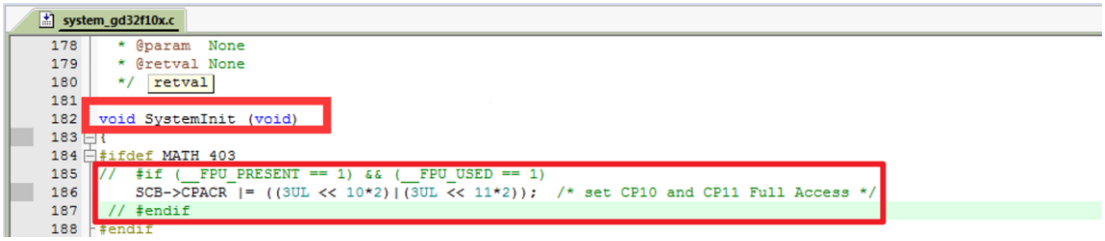
The following steps use Keil4 as an example, and the configuration method of Keil5 is exactly the same.

2.1 Turn on FPU

The 303 and 403 series have the FPU function. Some particular configuration need to be set as below.

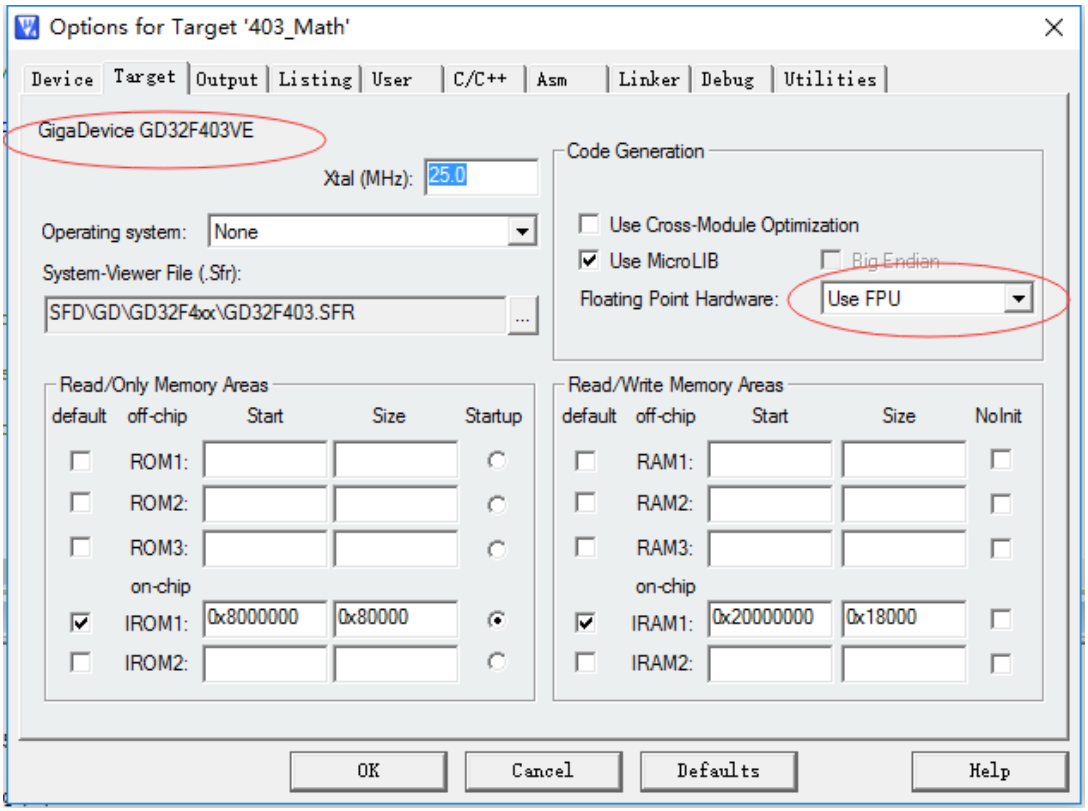
1. Enable FPU before executing code

Figure 2-1. Enable FPU



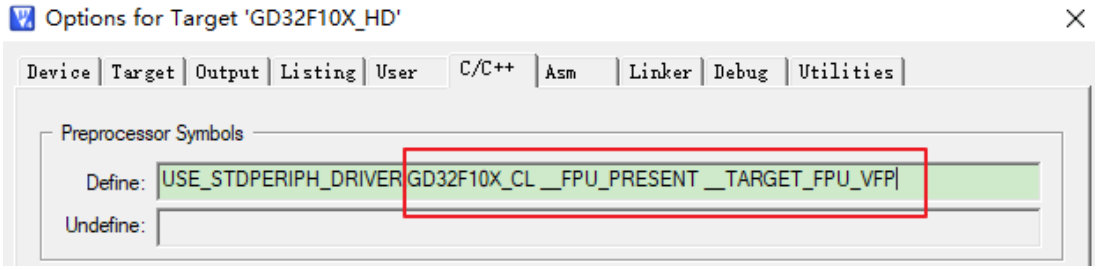
2. Select 303 or 403 model and use FPU

Figure 2-2. Select 303 or 403 and enable FPU



3. Fill in the corresponding compilation macro definition.

Figure 2-3. Add the corresponding compilation macro



Note: GD32F10X_CL must be configured for F403, and F303 and F103 must be the same.

2.2 Check whether the FPU is turned on successfully

In the JLINK debugging control interface, enter the command:

mem32 0xE000ED88 1

See the following figure for detailed operation. After entering the command, 00f00000 indicates that the FPU has been turned on. If it is 00000000, it indicates that the FPU has not been turned on.

Figure 2-4. Check whether the FPU is turned on

```
  T) cJTAG
TIF>swd
Specify target interface speed [kHz]. <Default>: 4000 kHz
Speed>
Device "GD32F303RE" selected.


Connecting to target via SWD
Found SW-DP with ID 0x2BA01477
Scanning AP map to find all available APs
AP[1]: Stopped AP scan as end of AP map has been reached
AP[0]: AHB-AP (IDR: 0x24770011)
Iterating through AP map to find AHB-AP to use
AP[0]: Core found
AP[0]: AHB-AP ROM base: 0xE00FF000
CPUID register: 0x410FC241. Implementer code: 0x41 (ARM)
Found Cortex-M4 r0p1, Little endian.
FPUnit: 6 code (BP) slots and 2 literal slots
CoreSight components:
ROMTbl[0] @ E00FF000
ROMTbl[0][0]: E000E000, CID: B105E00D, PID: 000BB00C SCS-M7
ROMTbl[0][1]: E0001000, CID: B105E00D, PID: 003BB002 DWT
ROMTbl[0][2]: E0002000, CID: B105E00D, PID: 002BB003 FPB
ROMTbl[0][3]: E0000000, CID: B105E00D, PID: 003BB001 ITM
ROMTbl[0][4]: E0040000, CID: B105900D, PID: 000BB9A1 TPIU
ROMTbl[0][5]: E0041000, CID: 00000000, PID: 00000000 ???
Cortex-M4 identified.
J-Link>mem32 0xE000ED88 1
E000ED88 = 00F00000
J-Link>
```

2.3 Performance test comparison

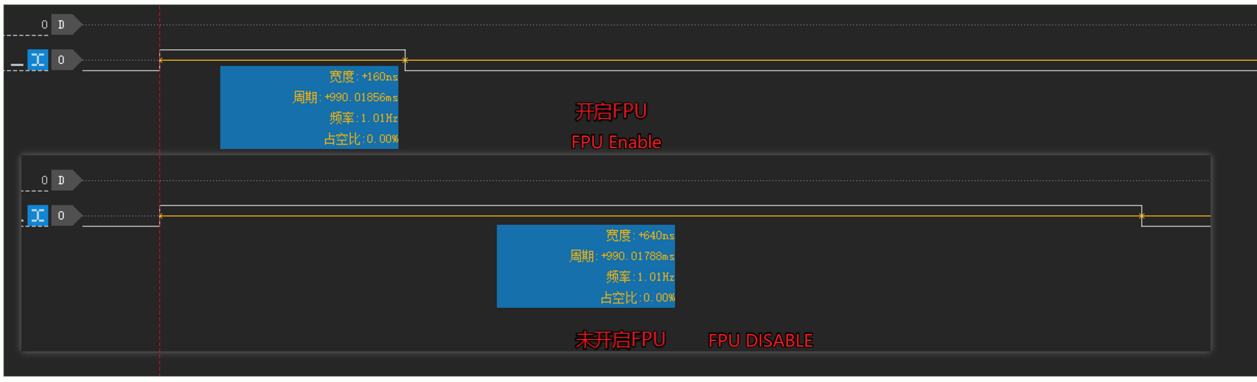
Compile the code, then the performance of M4 can be tested on 103 code.

```
int main(void)
{
    float i;
    float m = 2.5f;
    float n = 4;

    /* configure systick */
    systick_config();
    /* initilize the LEDs, USART and key */
    gd_eval_led_init(LED2);
    gd_eval_led_init(LED3);
    gd_eval_led_init(LED4);
    gd_eval_com_init(EVAL_COM0);
    gd_eval_key_init(KEY_WAKEUP, KEY_MODE_GPIO);

    while(1){
        if(RESET == gd_eval_key_state_get(KEY_WAKEUP)){
            gd_eval_led_on(LED3);
            i = m * n;
            gd_eval_led_off(LED3);
```

```
            printf("\r\ni = %f", i);
            while(RESET == gd_eval_key_state_get(KEY_WAKEUP));
        }
    }
}
```

Grab the pin waveform of LED3 with the logic analyzer and check the calculation time of floating-point operation. The following is the test comparison between opening FPU and not opening FPU:

Figure 2-5. Comparison of performance test

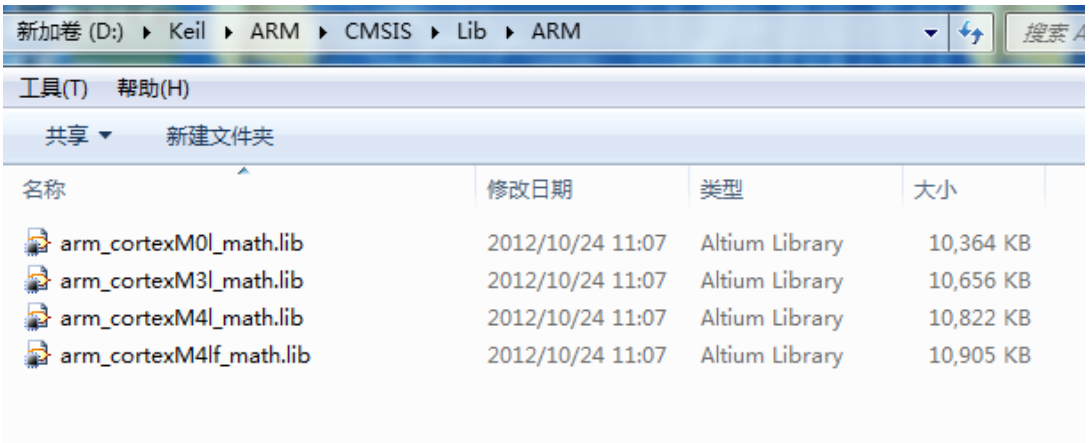


If FPU is not enabled, a floating-point multiplication takes 640ns, and only 160ns after FPU is enabled.

2.4 Turn on DSP

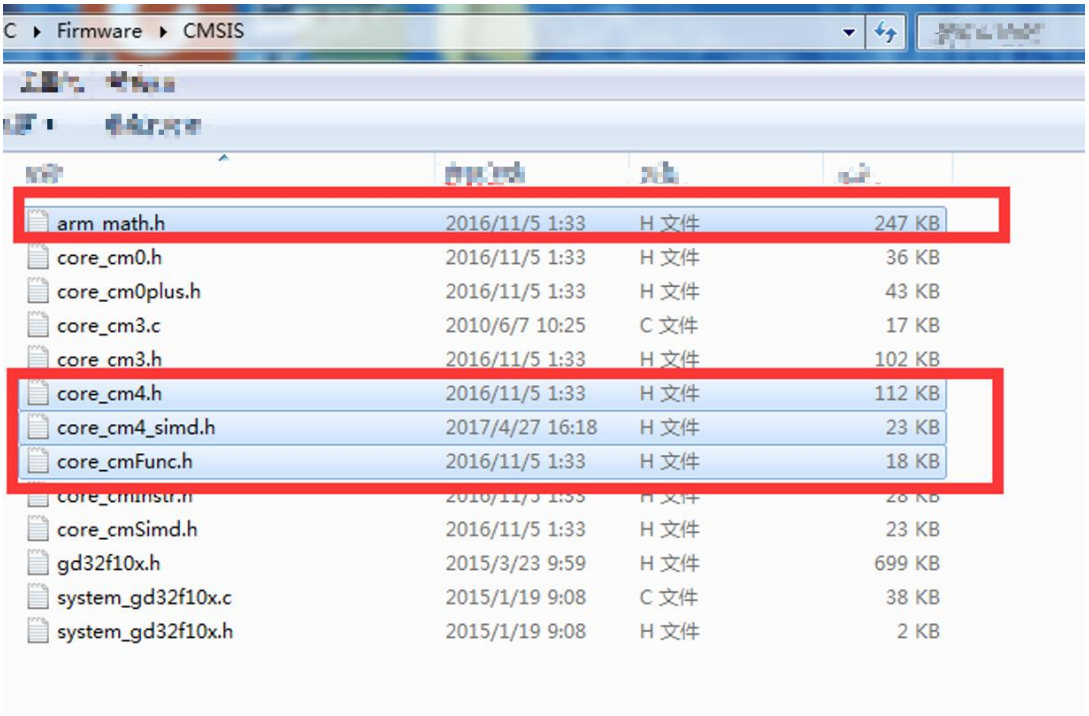
1. Copy math.lib from the MDK path and add to the project. The specific document is arm_cortexM4l_math.lib. (If the FPU is used, arm_cortexM4lf_math.lib will be selected.)

Figure 2-6. math.lib file



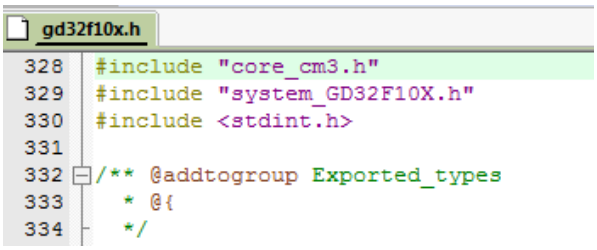
2. Copy the corresponding .H file to the project. The core_cm4.h file is needed if DSP is used.

Figure 2-7. The required .H file



3. Change the core_cm3.h to core_cm4.h in gd32f10x.h file.

Figure 2-8. Core_cm3.h changed to core_cm4.h



4. Modify the selection of the MCU model. 303 select the corresponding 303 model, 403 select the corresponding 403 model and the corresponding Math compilation macro.

Figure 2-9. Modify the chip model

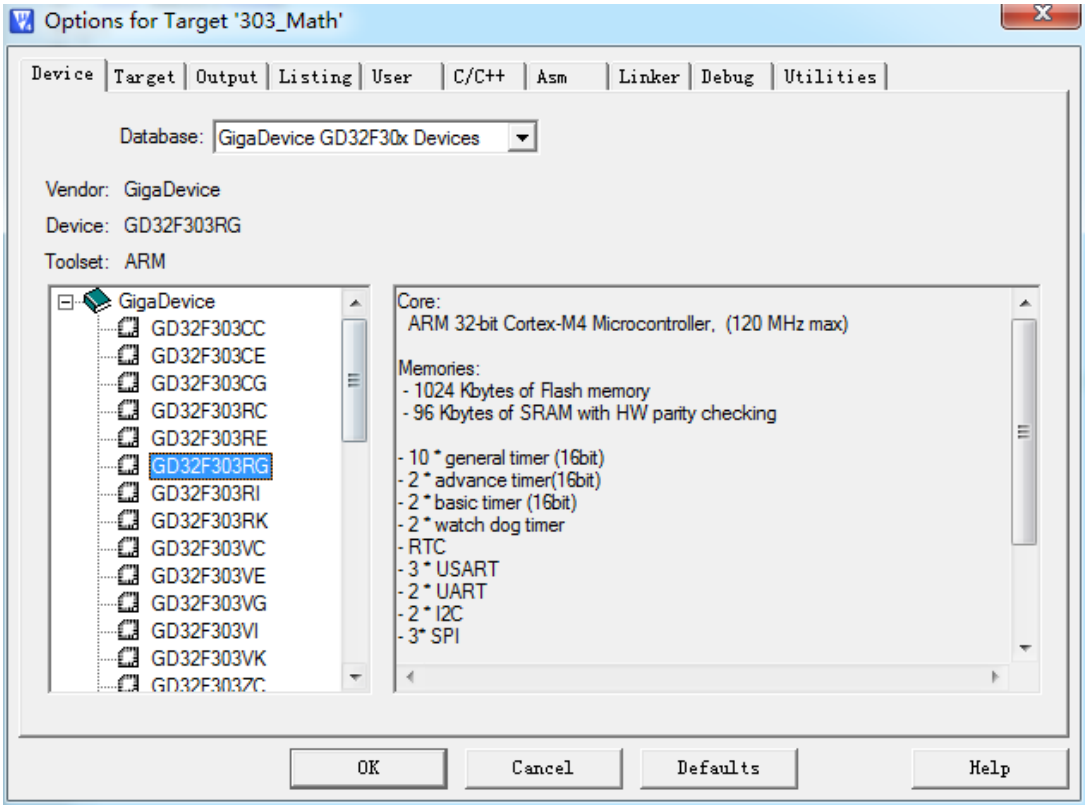
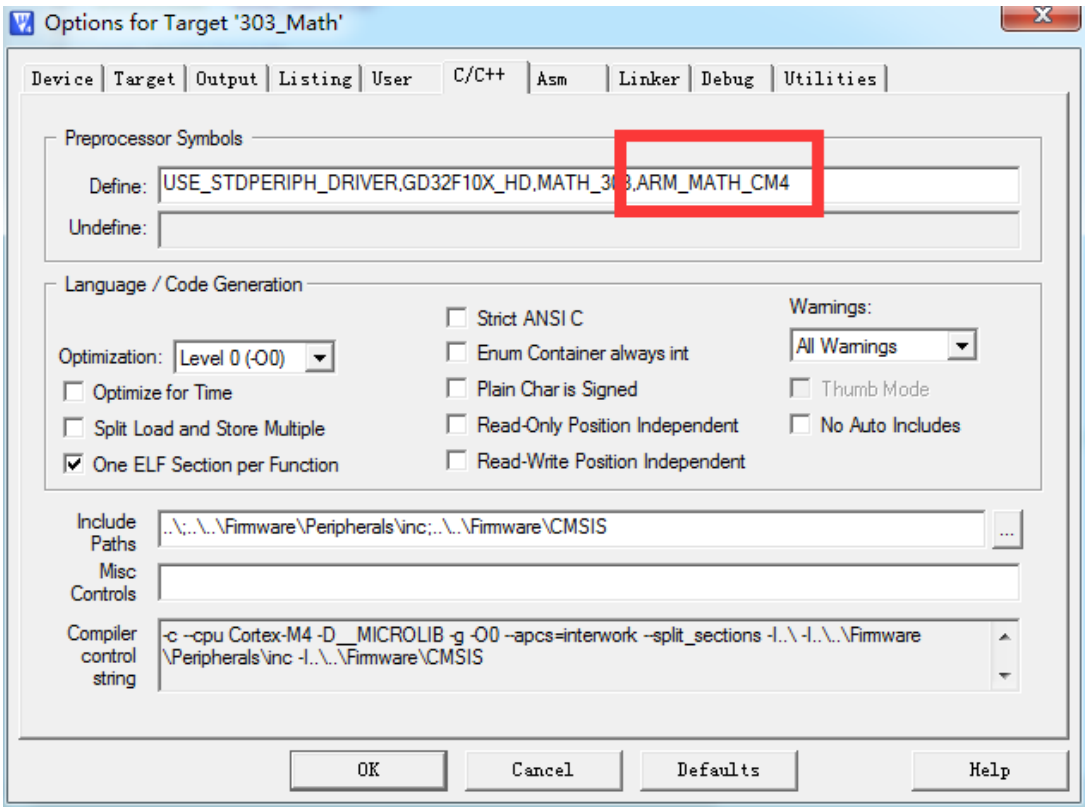


Figure 2-10. Add the corresponding Math compilation macro



3. Revision history

Table 3-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Apr.30, 2021

Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the “Product”), is owned by the Company under the intellectual property laws and treaties of the People’s Republic of China and other jurisdictions worldwide. The Company reserves all rights under such laws and treaties and does not grant any license under its patents, copyrights, trademarks, or other intellectual property rights. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

The Company makes no warranty of any kind, express or implied, with regard to this document or any Product, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Company does not assume any liability arising out of the application or use of any Product described in this document. Any information provided in this document 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. Except for customized products which has been expressly identified in the applicable agreement, the Products are designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only. The Products are not designed, intended, or authorized for use as components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, atomic energy control instruments, combustion control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, 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 Product could cause personal injury, death, property or environmental damage ("Unintended Uses"). Customers shall take any and all actions to ensure using and selling the Products in accordance with the applicable laws and regulations. The Company is not liable, in whole or in part, and customers shall and hereby do release the Company as well as it’s suppliers and/or distributors from any claim, damage, or other liability arising from or related to all Unintended Uses of the Products. Customers shall indemnify and hold the Company as well as it’s suppliers and/or distributors 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 the Products.

Information in this document is provided solely in connection with the Products. The Company reserves the right to make changes, corrections, modifications or improvements to this document and Products and services described herein at any time, without notice.