GigaDevice Semiconductor Inc.

Device limitations of GD32F47x/F42x

Errata Sheet

Revision 1.5

(Feb. 2025)

Table of Contents

List of Figures

List of Tables

1. Introduction

This document applies to GD32F47x/F42x product series, as shown in ***Table 1-1. Applicable products***. It provides the technical details that need to be paid attention to in the process of using GD32 MCU, as well as solutions to related problems.

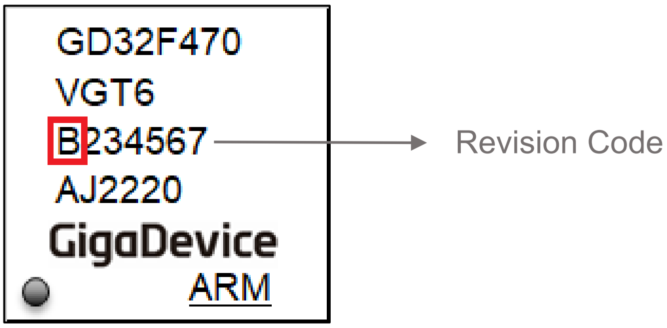
Table 1-1. Applicable products

Type	Part Numbers
MCU	GD32F425xx series
	GD32F427xx series
	GD32F470xx series

1.1. Revision identification

The device revision can be determined by the mark on the top of the package. The 1st code on the line 3 of the mark represents product revision code. As the picture shown in ***Figure 1-1. Device revision code of GD32F47x/F42x***.

Figure 1-1. Device revision code of GD32F47x/F42x



1.2. Summary of device limitations

The device limitations of GD32F47x/F42x are shown in ***Table 1-2. Device limitations***, please refer to section 2 for more details.

Table 1-2. Device limitations

Module	Limitations	Workaround		
		Rev. Code A	Rev. Code B	Rev. Code C
PMU	***Standby mode cannot be waked up due to frequent wakeup signals before or after entering standby mode***	N	N	--
RCU	***The LXTALSTB bit cannot be cleared by disabling LXTAL when LXTAL stops unexpectedly***	Y	Y	Y

Module	Limitations	Workaround		
		Rev. Code A	Rev. Code B	Rev. Code C
GPIO	*IO compensation invalidation*	N	--	--
ADC	*ADC samples abnormally when using both 6-bit sampling resolution and MSB alignment*	Y	Y	Y
	ADC alignment mode is not consistent with the user manual description	Y	Y	Y
RTC	*Calibrate abnormally when using both smooth digital calibration and FREQI calibration*	Y	Y	Y
	RTC smoothing digital calibration function is only effective within the current window	Y	Y	--
TIMER	*The shadow preloaded value takes effect only on the rising edge of the counter after modification*	N	N	N
	Count error when timer works at single pulse mode	Y	Y	Y
USART	*Mute mode can be waked up as long as the USART_CTL0 register is operated after mute mode is enabled*	Y	Y	Y
I2C	*I2C_FCTL register is only configurable on GD32F470xx*	N	N	N
SDIO	*Do not support low power mode*	N	N	--
	In SDIO multi-line mode (4-bit or 8-bit data width), after completing data reception, if the FIFO is not cleared in time and subsequent data arrives on the data line, it will cause the STBITE error to remain set state	Y	Y	Y
EXMC	*Auto refresh function of SDRAM controller is influenced by other EXMC controller*	Y	Y	--
ENET	*Data reception faults in MII mode*	Y	Y	Y
	Reception data frame is dropped when enable hardware checksum and the header checksum is 0x0000	Y	Y	--
Core	*VDIV or VSQRT instructions might not complete correctly when very short ISRs are used*	Y	Y	Y

Note:

Y = Limitation present, workaround available

N = Limitation present, no workaround available

'--' = Limitation fixed

2. Descriptions of device limitations

2.1. PMU

2.1.1. Standby mode cannot be waked up due to frequent wakeup signals before or after entering standby mode

Description & impact

When reset the internal signal STBY_CTL to enter to standby mode, if the T_{glitch} is smaller than 100ns, which will cause the mcu cannot be waked up. The narrow glitch will result in incorrect Vcore voltage.

Note: The T_{glitch} is the time between STBY_CTL low level and the wakeup signal (PA0 high level)

Workarounds

Not available.

2.2. RCU

2.2.1. The LXTALSTB bit cannot be cleared by disabling LXTAL when LXTAL stops unexpectedly

Description & impact

When LXTAL stops unexpectedly, the LXTALSTB bit cannot be cleared by disabling the LXTAL due to six LXTAL clock cannot be generated, which prevents the LXTAL from restarting.

Workarounds

Use one of the following solutions:

1) By repeatedly setting and resetting the LXTALBPS more than ten times to clear the LXTALSTB bit, and then reconfiguring the LXTAL. The reference code for clearing LXTALSTB bits is as follows:

```
void lxtal_stb_clear(void)
{
    volatile   uint32_t i = 0U;
    /* close LXTAL clock */
    rcu_osci_off(RCU_LXTAL);
    for(i = 0; i < 10; i++) {
            /* enable the LXTAL bypass mode */
            rcu_osci_bypass_mode_enable(RCU_LXTAL);
            /* disable the LXTAL bypass mode */
            rcu_osci_bypass_mode_disable(RCU_LXTAL);
    }
}
```

2) Write 0x000000C3, 0x0000003C, 0x00000000 sequentially to the address 0x400028FC. The reference code for clearing LXTALSTB bits is as follows (This workaround is only for Rev. Code C):

```
/* clear LXTALSTB bit */
REG32(0x400028FC) = 0x000000C3;
REG32(0x400028FC) = 0x0000003C;
REG32(0x400028FC) = 0x00000000;
```

2.3. GPIO

2.3.1. IO compensation invalidation

Description & impact

IO compensation function is invalid.

Workarounds

Not available.

2.4. ADC

2.4.1. ADC samples abnormally when using both 6-bit sampling resolution and MSB alignment

Description & impact

ADC samples abnormally when using both 6-bit sampling resolution and MSB alignment mode.

Workarounds

Use LSB alignment mode or use 8-bit sampling resolution.

2.4.2. ADC alignment mode is not consistent with the user manual description

Description & impact

ADC alignment mode is not consistent with the user manual description.

Workarounds

The correct alignment mode description is as follow:

Table 2-1. Alignment mode of routine conversion

<table>
<tr><th>Alignment</th><th>Resolution</th><th>bit15</th><th>bit14</th><th>bit13</th><th>bit12</th><th>bit11</th><th>bit10</th><th>bit9</th><th>bit8</th><th>bit7</th><th>bit6</th><th>bit5</th><th>bit4</th><th>bit3</th><th>bit2</th><th>bit1</th><th>bit0</th></tr>
<tr><td rowspan="4">LSB</td><td>12bit</td><td colspan="4">0x0</td><td colspan="12">data</td></tr>
<tr><td>10bit</td><td colspan="4">0x0</td><td colspan="10">data</td><td colspan="2">0x0</td></tr>
<tr><td>8bit</td><td colspan="4">0x0</td><td colspan="8">data</td><td colspan="4">0x0</td></tr>
<tr><td>6bit</td><td colspan="4">0x0</td><td colspan="6">data</td><td colspan="6">0x0</td></tr>
<tr><td rowspan="4">MSB</td><td>12bit</td><td colspan="12">data</td><td colspan="4">0x0</td></tr>
<tr><td>10bit</td><td colspan="10">data</td><td colspan="6">0x0</td></tr>
<tr><td>8bit</td><td colspan="8">data</td><td colspan="8">0x0</td></tr>
<tr><td>6bit</td><td colspan="16">invalid</td></tr>
</table>

2.5. RTC

2.5.1. Calibrate abnormally when using both smooth digital calibration and FREQI calibration

Description & impact

Using both smooth digital calibration and FREQI calibration will cause calibration result abnormal.

Workarounds

1) Use RTC shift function to replace smooth digital calibration, such as setting A1S bit and configuring appropriate SFS bits to satisfy the accuracy requirement.
2) Use two 16 seconds calibration window to replace one 32 seconds calibration window.

2.5.2. RTC smoothing digital calibration function is only effective within the current window

Description & impact

RTC smoothing digital calibration function is only effective within the current window (a configurable period time).

Workarounds

Periodically reconfigure the smoothing calibration function after the smoothing calibration window time arrives.

2.6. TIMER

2.6.1. The shadow preloaded value takes effect only on the rising edge of the counter after modification

Description & impact

The preloaded value takes effect only on the rising edge of the counter after modification which may cause problem to pwm applications with high timing requirements.

Workarounds

Not available.

2.6.2. Count error when timer works at single pulse mode

Description & impact

Timer works at single pulse mode and CK_APBx is CK_AHB / 4 and CK_TIMER is CK_AHB / 2, which causes count error.

Workarounds

1) Do not use above clock configuration.
2) Use timer update interrupt to clear error count.

2.7. USART

2.7.1. Mute mode can be waked up as long as the USART_CTL0 register is operated after mute mode is enabled

Description & impact

After mute mode is enabled, the operation on USART_CTL0 register will wake up USART from mute mode.

Workarounds

When mute mode is enabled and USART uses hardware method to detect idle frame wakeup, operation on USART_CTL0 register is not allowed. When mute mode is enabled and USART uses software method to detect idle frame wakeup, operation on USART_CTL0 register only be allowed when need to exit mute mode.

2.8. I2C

2.8.1. I2C_FCTL register is only configurable on GD32F470xx

Description & impact

The filter control register (I2C_FCTL) is only configurable on GD32F470xx but not on GD32F425xx or GD32F427xx.

Workarounds

Not available.

2.9. SDIO

2.9.1. Do not support low power mode

Description & impact

SDIO_CLK can not be closed automatically in bus idle state when CLKPWRSAV is set in SDIO_CLKCTL register.

Workarounds

Not available.

2.9.2. In SDIO multi-line mode (4-bit or 8-bit data width), after completing data reception, if the FIFO is not cleared in time and subsequent data arrives on the data line, it will cause the STBITE error to remain set state

Description & impact

In SDIO multi-line mode (4-bit or 8-bit data width), after completing data reception, if the FIFO is not cleared in time and subsequent data arrives on the data line, it will cause the STBITE error to remain set state.

Workarounds

Use one of the following solutions:

1) When using polling mode, software should check both the STBITE flag and the DTEND flag at the same time. The STBITE flag is considered valid only when STBITE is set and DTEND is not set. The reference code is as follows:

```
if ((RESET != sdio_flag_get(SDIO_FLAG_STBITE)) && \\
   (RESET == sdio_flag_get(SDIO_FLAG_DTEND))){
    /* user code */
    ....
}
```

2) When using interrupt mode, the interrupt function should process the DTEND flag before the STBITE flag, and the interrupt enable bit STBITEIE should be cleared within the code of handling the DTEND flag. The reference code is as follows:

```
void SDIO_IRQHandler(void){
    if((RESET != sdio_flag_get(SDIO_FLAG_DTEND)) && \\
      (0U != (SDIO_INTEN & SDIO_INTEN_DTENDIE))){
        sdio_interrupt_disable(SDIO_INT_STBITE);
         /* user code */
         ....
    }
    if((RESET != sdio_flag_get(SDIO_FLAG_STBITE)) && \\
      (0U != (SDIO_INTEN & SDIO_INT_STBITE))){
         /* user code */
         ....
    }
}
```

3) Instead of using CMD12 to stop the open transmission of data, the multiblock read uses CMD23 to notify the slave device in advance of the number of data blocks to be transmitted. Then when DTEND is set, there will be no more extra data sent to the data line from the slave.

2.10. EXMC

2.10.1. Auto refresh function of SDRAM controller is influenced by other EXMC controller

Description & impact

Auto refresh function of SDRAM controller is influenced by other EXMC controller. When SDRAM controller execute auto refresh command, if the SDRAM bank is active, the precharge command shall be generated, which need EXMC_A10 port be 1. At that time, EXMC_A10 port is used in other EXMC controller, then the SDRAM auto refresh command execute abnormally which lead SDRAM data error.

Workarounds

Step1: enable EXMC SDRAM controller works simultaneously with other controllers after EXMC initialization.

```
/* code example */
REG32(EXMC + 0x184U) = 0x9EF02310U;
EXMC_SDRSCTL |= BIT(9);
```

Step2:

Method 1: When SDRAM controller selects the BANK address of the operation, the pin output does not use the AF function and accesses the corresponding BANK directly through GPIO to drive the BANK address.

Method 2: Before EXMC operates on NAND FLASH, the global precharge instruction of SDRAM is added, so that the self-refresh operation of SDRAM does not need to rely on the original precharge instruction, so even if the self-refresh and nand occur at the same time, there is no error.

```
/* code example */
REG32(0xA0000150U) = (uint32_t)0x00000012U;
while(0x00000000 != (REG32(0xA0000158U) & 0x00000020)) {
}
```

2.11. ENET

2.11.1. Data reception faults in MII mode

Description & impact

ENET_MII_COL / ENET_MII_CRS / ENET_MII_RX_ER pins are floating on MCU and external PHY has no these pins, which will cause data reception faults.

Workarounds

Configure ENET_MII_COL pin and ENET_MII_RX_ER pin as AF function and keep them low level. The ENET_MII_CRS pin mode and status can be ignored.

2.11.2. Reception data frame is dropped when enable hardware checksum and the header checksum is 0x0000

Description & impact

When enable hardware checksum and header checksum is 0x0000, this frame will be mistaken for error frame and dropped by hardware.

Workarounds

Use software checksum.

2.12. Core

2.12.1. VDIV or VSQRT instructions might not complete correctly when very short ISRs are used

This limitation refers to Arm ID number 776924 in “Cortex-M4 & Cortex-M4 with FPU Software Developers Errata Notice”.

Description & impact

The VDIV and VSQRT instructions take 14 cycles to execute. When an interrupt is taken a VDIV or VSQRT instruction is not terminated, and completes its execution while the interrupt stacking occurs. If lazy context save of floating point state is enabled then the automatic stacking of the floating point context does not occur until a floating point instruction is executed inside the interrupt service routine.

Lazy context save is enabled by default. When it is enabled, the minimum time for the first instruction in the interrupt service routine to start executing is 12 cycles. In certain timing conditions, and if there is only one or two instructions inside the interrupt service routine, then the VDIV or VSQRT instruction might not write its result to the register bank or to the FPSCR.

The failure occurring conditions are as follows:

1) The floating point unit is enabled.
2) Lazy context saving is not disabled.
3) A VDIV or VSQRT is executed.
4) The destination register for the VDIV or VSQRT is one of s0 - s15.
5) An interrupt occurs and is taken.
6) The interrupt service routine being executed does not contain a floating point instruction.
7) Within 14 cycles after the VDIV or VSQRT is executed, an interrupt return is executed.

A minimum of 12 of these 14 cycles are utilized for the context state stacking, which leaves 2 cycles for instructions inside the interrupt service routine, or 2 wait states applied to the entire stacking sequence (which means that it is not a constant wait state for every access).

In general, this means that if the memory system inserts wait states for stack transactions then this erratum cannot be observed.

The implications of this limitation is that the VDIV or VSQRT instruction does not complete correctly and the register bank and FPSCR are not updated, which means that these registers hold incorrect, out of date, data.

Workarounds

A workaround is only required if the floating point unit is enabled. A workaround is not required if the stack is in external memory.

There are two possible workarounds:

1) Disable lazy context save of floating point state by clearing LSPEN to 0 (bit 30 of the FPCCR at address 0xE000EF34).

2) Ensure that every interrupt service routine contains more than 2 instructions in addition to the exception return instruction.

3. Revision history

Table 3-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Mar.16 2023
1.1	Update description of chapter 2.8.1	Apr.3 2023
1.2	Update note of chapter 1.2	Apr.6 2023
1.3	1. Add PMU limitation, referring to chapter 2.1.1 2. Add core limitation, referring to chapter 2.11.1	Nov.2 2023
1.4	Update workarounds of EXMC, refer to ***Auto refresh function of SDRAM controller is influenced by other EXMC controller***	Jul.18 2024
1.5	1. Add limitations of Rev. Code C. 2. Add RCU limitation, referring to ***The LXTALSTB bit cannot be cleared by disabling LXTAL when LXTAL stops unexpectedly*** 3. Add RTC limitation, referring to ***RTC smoothing digital calibration function is only effective within the current window*** 4. Add SDIO limitation, refer to ***In SDIO multi-line mode (4-bit or 8-bit data width), after completing data reception, if the FIFO is not cleared in time and subsequent data arrives on the data line, it will cause the STBITE error to remain set state***	Feb.5 2025

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 according to the laws of the People’s Republic of China and other applicable laws. The Company reserves all rights under such laws and no Intellectual Property Rights are transferred (either wholly or partially) or licensed by the Company (either expressly or impliedly) herein. 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.

To the maximum extent permitted by applicable law, the Company makes no representations or warranties of any kind, express or implied, with regard to the merchantability and the fitness for a particular purpose of the Product, nor does the Company assume any liability arising out of the application or use of any Product. Any information provided in this document is provided only for reference purposes. It is the sole responsibility of the user of this document to determine whether the Product is suitable and fit for its applications and products planned, and properly design, program, and test the functionality and safety of its applications and products planned using the Product. The Product is designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only, and the Product is not designed or intended for use in (i) safety critical applications such as weapons systems, nuclear facilities, atomic energy controller, combustion controller, aeronautic or aerospace applications, traffic signal instruments, pollution control or hazardous substance management; (ii) life-support systems, other medical equipment or systems (including life support equipment and surgical implants); (iii) automotive applications or environments, including but not limited to applications for active and passive safety of automobiles (regardless of front market or aftermarket), for example, EPS, braking, ADAS (camera/fusion), EMS, TCU, BMS, BSG, TPMS, Airbag, Suspension, DMS, ICMS, Domain, ESC, DCDC, e-clutch, advanced-lighting, etc.. Automobile herein means a vehicle propelled by a self-contained motor, engine or the like, such as, without limitation, cars, trucks, motorcycles, electric cars, and other transportation devices; and/or (iv) other uses where the failure of the device or the Product can reasonably be expected to result in personal injury, death, or severe property or environmental damage (collectively "Unintended Uses"). Customers shall take any and all actions to ensure the Product meets the applicable laws and regulations. The Company is not liable for, in whole or in part, and customers shall hereby release the Company as well as its suppliers and/or distributors from, any claim, damage, or other liability arising from or related to all Unintended Uses of the Product. Customers shall indemnify and hold the Company, and its officers, employees, subsidiaries, affiliates as well as its 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 Product.

Information in this document is provided solely in connection with the Product. The Company reserves the right to make changes, corrections, modifications or improvements to this document and the Product described herein at any time without notice. The Company shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. Information in this document supersedes and replaces information previously supplied in any prior versions of this document.