GigaDevice Semiconductor Inc.

Device limitations of GD32F303xxA

Errata Sheet

Revision 1.1

(Jun. 2025)

Table of Contents

Table of Contents 2
List of Figures 4
List of Tables 5
1. Introduction 6
1.1. Revision identification 6
1.2. Summary of device limitations 6
2. Descriptions of device limitations 8
2.1. PMU 8
2.1.1. Standby mode cannot be waked up due to frequent wakeup signals before or after entering standby mode 8
2.2. RCU 8
2.2.1. MCU cannot be waked up after entering deep-sleep mode when DSLP_HOLD bit is set 8
2.2.2. The LXTALSTB bit cannot be cleared by disabling LXTAL when LXTAL stops unexpectedly 8
2.3. ADC 9
2.3.1. ADC data acquisition error occurs when the ADC clock is much slower than the PCLK clock 9
2.4. DAC 9
2.4.1. In the DAC noise mode, when the DH value is configured too high, the DAC output voltage will have an abnormal break point 9
2.5. USART 10
2.5.1. Mute mode can be waked up as long as the USART_CTL0 register is operated after mute mode is enabled 10
2.5.2. When USART is configured in DMA transmit mode and DMA channel is enabled before TEN is set, resulting in data loss 10
2.5.3. The high baud rate of USART will cause data loss when using hardware flow control mode 10
2.5.4. In idle frame wakeup mode, USART is woken up early by idle frames 11
2.6. I2C 11
2.6.1. Read one more data because the BTC flag was not cleared 11
2.7. SDIO 11
2.7.1. 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 11
2.8. EXMC 12
2.8.1. When the hold time of the NAND controller is set to 1, data transmission errors will occur 12

2.9. Core 13
2.9.1. VDIV or VSQRT instructions might not complete correctly when very short ISRs are used 13
3. Revision history 15

List of Figures

Figure 1-1. Device revision code of GD32F303xxA 6

List of Tables

Table 1-1. Applicable products **6**
Table 1-2. Device limitations **6**
Table 3-1. Revision history **15**

1. Introduction

This document applies to GD32F303xxA 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

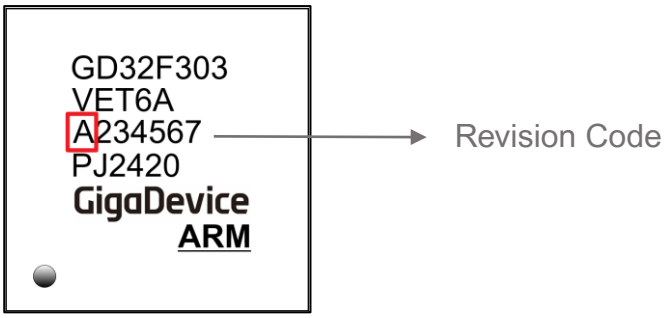
Type	Part Numbers
MCU	GD32F303xB series
	GD32F303xxA series

Note: For specific applicable products, please refer to "GD32F303xB Datasheet" and "GD32F303xxA Datasheet".

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 GD32F303xxA***.

Figure 1-1. Device revision code of GD32F303xxA



1.2. Summary of device limitations

The device limitations of GD32F303xxA 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
PMU	***Standby mode cannot be waked up due to frequent wakeup signals before or after entering standby mode***	N	N
RCU	***MCU cannot be waked up after entering deep-sleep mode when DSLP_HOLD bit is set***	Y	Y

	The LXTALSTB bit cannot be cleared by disabling LXTAL when LXTAL stops unexpectedly	Y	Y
ADC	*ADC data acquisition error occurs when the ADC clock is much slower than the PCLK clock*	Y	Y
DAC	*In the DAC noise mode, when the DH value is configured too high, the DAC output voltage will have an abnormal break point*	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
	When USART is configured in DMA transmit mode and DMA channel is enabled before TEN is set, resulting in data loss	Y	Y
	The high baud rate of USART will cause data loss when using hardware flow control mode	N	N
	In idle frame wakeup mode, USART is woken up early by idle frames	N	N
I2C	*Read one more data because the BTC flag was not cleared*	Y	Y
SDIO	*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
EXMC	*When the hold time of the NAND controller is set to 1, data transmission errors will occur*	Y	Y
Core	*VDIV or VSQRT instructions might not complete correctly when very short ISRs are used*	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 Tglitch 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. MCU cannot be waked up after entering deep-sleep mode when DSLP_HOLD bit is set

Description & impact

When DSLP_HOLD bit is set and debug the mcu in deep-sleep mode, the mcu will not be waked up.

Workarounds

When the DSLP_HOLD bit is set to enable low power debugging, the application programme need switch the system clock to IRC8M before entering the deep-sleep mode.

2.2.2. 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, which prevents the LXTAL from restarting.

Workarounds

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);
    }
}
```

2.3. ADC

2.3.1. ADC data acquisition error occurs when the ADC clock is much slower than the PCLK clock

Description & impact

When the ADC clock is much slower than the PCLK clock, reading the ADC_RDATA register immediately after the EOC is set will result in a data acquisition error.

Workarounds

When the delay between reading EOC flag and reading ADC_RDATA is no more than two ADC clocks, after the EOC flag is set, software need to delay two ADC clocks before reading the ADC_RDATA register.

2.4. DAC

2.4.1. In the DAC noise mode, when the DH value is configured too high, the DAC output voltage will have an abnormal break point

Description & impact

When the DAC is configured in noise mode and the DAC output value is set to a large value, the superimposed value (DH+DWBW) will exceed the maximum value of 4095, resulting in

an abnormal break point of zero voltage in the DAC output signal.

Workarounds

When the DAC output value and the noise wave peak value are configured, avoid the superimposed value (DH+DWBW) overflow.

2.5. USART

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

2.5.2. When USART is configured in DMA transmit mode and DMA channel is enabled before TEN is set, resulting in data loss

Description & impact

When USART is configured in DMA transmit mode (DEN bit is set), enabling the DMA channel when the transmitter is disabled (TEN bit is reset) will cause all data in the buffer to be sent to the USART data register, resulting in data loss.

Workarounds

Set TEN bit in the USART_CTL0 register before the CHEN bit of the corresponding DMA_CHxCTL register is enabled.

2.5.3. The high baud rate of USART will cause data loss when using hardware flow control mode

Description & impact

In the case of high baud rate of USART, when using the hardware flow control mode, the transmitter does not detect the RTS level in time due to the timing lag of the receiver RTS

pulling up, resulting in one more data sent by the USART transmitter.

Workarounds

Not available.

2.5.4. In idle frame wakeup mode, USART is woken up early by idle frames

Description & impact

In multiprocessor communication mode, when USART is configured to wake up for an idle frame, it is woken up from mute mode early before an idle frame of the specified length has been detected or IDELF is set to 1.

Workarounds

Not available.

2.6. I2C

2.6.1. Read one more data because the BTC flag was not cleared

Description & impact

If an interrupt occurs before reading I2C_DATA register when RBNE flag is set and BTC flag is reset, I2C will read an additional data if BTC flag is set during the interrupt processing because the read data operation cannot clear the BTC flag.

Workarounds

1) Using interrupt method to read the I2C_DATA register (need higher interrupt priority).
2) Using DMA method to read the I2C_DATA register (recommend).

2.7. SDIO

2.7.1. 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.

2.8. EXMC

2.8.1. When the hold time of the NAND controller is set to 1, data transmission errors will occur

Description & impact

When the hold time (ATTHLD/COMHLD bit fields) is set to 1, the NAND controller will send an extra frame of data during the write operation, which results in transmission error.

Workarounds

Configure the NAND controller's hold time (ATTHLD/COMHLD bit fields) to be greater than 1. The reference code is as follow:

```
exmc_nand_pccard_timing_parameter_struct nand_timing_init_struct;
/* configure the address hold time(or the data hold time for write operation) */
nand_timing_init_struct.holdtime = 3;
```

2.9. Core

2.9.1. 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	Aug.12 2024
1.1	1. 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*** 2. Delete I2C limitaion, ***Due to the timing difference of START signal between I2C0 and I2C1, the master that wins the arbitration cannot receive ACK, and the bus is blocked***	Jun.25 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.

© 2026 GigaDevice Semiconductor Inc. - All rights reserved