GigaDevice Semiconductor Inc.

Device limitations of GD32E51x

Errata Sheet

Revision 1.1

(Jun. 2025)

Table of Contents

Table of Contents **2**
List of Figures **3**
List of Tables **4**
1. Introduction **5**
1.1. Revision identification **5**
1.2. Summary of device limitations **5**
2. Descriptions of device limitations **7**
2.1. RCU **7**
2.1.1. MCU cannot be waked up from Deepsleep mode when DSLP_HOLD bit is set 7
2.1.2. The LXTALSTB bit cannot be cleared by disabling LXTAL when LXTAL stops unexpectedly 7
2.2. ADC **8**
2.2.1. ADC data acquisition error occurs when the ADC clock is much slower than the PCLK clock 8
2.3. DAC **8**
2.3.1. In the DAC noise mode, when the value (DH+DWBW) is configured more than 4095, the DAC output voltage will have an abnormal break point 8
2.4. I2C **8**
2.4.1. Read one more data because the BTC flag was not cleared 8
2.4.1. When I2C2 works in 7/10 address slave mode, receiving an abnormal timing will cause the SDA line to stuck 9
2.5. SDIO **9**
2.5.1. 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 9
2.6. EXMC **10**
2.6.1. NE timing can not satisfy the requirement when using NAND pre-waiting function 10
2.7. USBHS **11**
2.7.1. When USBHS device uses an internal PHY and is suspended by the host, its suspension interrupt flag and wakeup interrupt flag are frequently set 11
3. Revision history **12**

List of Figures

Figure 1-1. Device revision code of GD32E51x 5

List of Tables

Table 1-1. Applicable products 5
Table 1-2. Device limitations 5
Table 3-1. Revision history 12

1. Introduction

This document applies to GD32E51x 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.

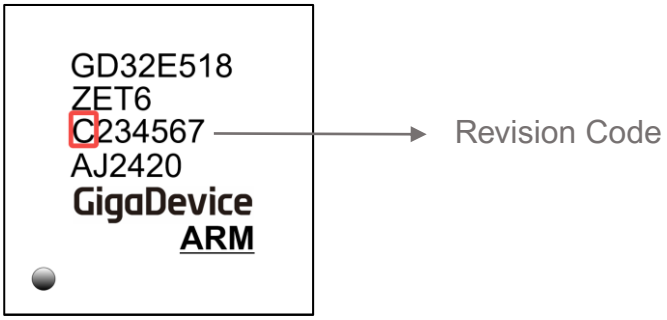
Table 1-1. Applicable products

Type	Part Numbers
MCU	GD32E513xx series
	GD32E517xx series
	GD32E518xx series
	GD32EPRTxxA 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 GD32E51x***.

Figure 1-1. Device revision code of GD32E51x



1.2. Summary of device limitations

The device limitations of GD32E51x 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 C
RCU	*MCU cannot be waked up from Deepsleep mode when DSLP_HOLD bit is set*	Y
	The LXTALSTB bit cannot be cleared by disabling LXTAL when LXTAL stops unexpectedly	Y
ADC	*ADC data acquisition error occurs when the ADC clock is much*	Y

Module	Limitations	Workaround
		Rev. Code C
	slower than the PCLK clock	
DAC	***In the DAC noise mode, when the value (DH+DWBW) is configured more than 4095, the DAC output voltage will have an abnormal break point***	Y
I2C	***Read one more data because the BTC flag was not cleared***	Y
	When I2C2 works in 7/10 address slave mode, receiving an abnormal timing will cause the SDA line to stuck	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
EXMC	***NE timing can not satisfy the requirement when using NAND pre-waiting function***	Y
USBHS	***When USBHS device uses an internal PHY and is suspended by the host, its suspension interrupt flag and wakeup interrupt flag are frequently set***	N

Note:

Y = Limitation present, workaround available

N = Limitation present, no workaround available

'--' = Limitation fixed

2. Descriptions of device limitations

2.1. RCU

2.1.1. MCU cannot be waked up from Deepsleep mode when DSLP_HOLD bit is set

Description & impact

When the DSLP_HOLD bit in the DBG_CTL register is set, the MCU cannot be waked up after the MCU enters deepsleep mode.

Workarounds

When the DSLP_HOLD bit is set to enable low power debugging, the application program needs to switch the system clock to IRC8M before entering the deepsleep mode.

2.1.2. 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);
    }
}
```

2.2. ADC

2.2.1. 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, the ADC_RDATA register is read immediately after the EOC is set and a data acquisition error occurs.

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.

2.3. DAC

2.3.1. In the DAC noise mode, when the value (DH+DWBW) is configured more than 4095, 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.

2.4. I2C

2.4.1. 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.

Note: This limitation applies to I2C0 / I2C1.

Workarounds

Use one of the following solutions:

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).

2.4.1. When I2C2 works in 7/10 address slave mode, receiving an abnormal timing will cause the SDA line to stuck

Description & impact

When the I2C is operating as a slave device in 7-bit address mode and the I2C master simulates I2C communication via IO. If the master sends the following sequence, the I2C slave will enter an error state, causing it to malfunction and the SDA line to remain low:

Start + 10-bit Match Head Address + Start + 7-bit Address Read + Wait ACK + Start

When the I2C is operating as a slave device in 10-bit address mode and the I2C master simulates I2C communication via IO. If the master sends the following sequence, the I2C slave will enter an error state, causing it to malfunction and the SDA line to remain low:

Start + 10-bit Mismatch Head Address + Start
or
Start + 10-bit Match Head Address + Wait ACK + 10-bit Mismatch 8-bit Address + Start

Note: This limitation applies to I2C2.

Workarounds

Software periodically checks the status of the SDA line. If SDA is detected to be stuck low, reinitialize the I2C module.

2.5. SDIO

2.5.1. 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.

2.6. EXMC

2.6.1. NE timing can not satisfy the requirement when using NAND pre-waiting function

Description & impact

For some EXMC_NCE-sensitive NAND Flash, NE timing can not satisfy the requirement when using NAND pre-waiting function. NE signal keeps the low level when EXMC_INTx is active.

Workarounds

Using general I/O port to simulate the NE timing to finish the NAND reading and writing, NE

signal keeps the low level after starting reading or writing.

2.7. USBHS

2.7.1. When USBHS device uses an internal PHY and is suspended by the host, its suspension interrupt flag and wakeup interrupt flag are frequently set

Description & impact

When a USBHS device uses an internal PHY and is suspended by the host, its suspension interrupt flag (SP) and wakeup interrupt flag (WKUPIF) are frequently set.

Workarounds

Not available. But it does not affect the normal use of USBHS.

3. Revision history

Table 3-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Aug.22, 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. Update I2C limitation, refer to ***When I2C2 works in 7/10 address slave mode, receiving an abnormal timing will cause the SDA line to stuck***	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