GigaDevice Semiconductor Inc.

Device limitations of GD32F10x

Errata Sheet

Revision 1.1

(Jun. 2025)

Table of Contents

Table of Contents **2**
List of Figures **5**
List of Tables **6**
1. Introduction **7**
1.1. Revision identification **7**
1.2. Summary of device limitations **7**
2. Descriptions of device limitations **10**
2.1. PMU **10**
2.1.1. MCU can not be waked up when an interrupt occurs before entering deep-sleep 10
2.2. RCU **11**
2.2.1. MCU can not be waked up after entering deep-sleep mode when DSLP_HOLD bit is set 11
2.2.2. When the HXTAL uses bypass mode, MCU power consumption will increase in sleep and standby mode 11
2.2.3. When the HXTAL clock monitor is enabled, the HXTAL clock stuck interrupt flag will be set after entering sleep mode 12
2.3. GPIO **12**
2.3.1. When the OSCIN and OSCOUT pins are remapped to PD0/PD1 pins, PD0/PD1 can be used as a normal GPIO function but cannot be used as an EXTI function 12
2.3.2. When the GPIO port is configured as a pull-up input and there is leakage in this GPIO, the voltage on the GPIO port will be pulled down 12
2.3.3. When PA11 and PA12 are not used as USB function and there is a large voltage difference between PA11 and PA12, there will be leakage from VDD to VSS 13
2.3.4. When both the USART1 and Ethernet clocks are enabled simultaneously, if the USART1 pins are not remapped, the Ethernet function will not work properly 13
2.4. DMA **13**
2.4.1. In the DMA interrupt handler function, exiting the interrupt immediately after performing the flag clear operation will cause the CPU to enter DMA interrupt again 13
2.5. ADC **14**
2.5.1. ADC data acquisition error occurs when the ADC clock is much slower than the PCLK clock 14
2.5.2. When the voltage difference between V_{REFP} and V_{DDA} exceeds 300mV, it will lead to poor ADC accuracy 14
2.6. DAC **14**
2.6.1. In the DAC noise mode, when the DH value is configured too high, the DAC output voltage will have an abnormal break point 14

2.7. TIMER 15
2.7.1. Data lost when using timer capture / compare event to trigger DMA transfer and enabling the output compare shadow function 15
2.7.2. TIMER in different channels, channel comparison/interrupt flag clearing and channel comparison/capture interrupt flag setting conflict, resulting in flag bit not set 15
2.7.3. The master TIMER triggers slave timers under different bus, resulting in phase differences in the output of different slave timers under the same comparison value 16
2.8. USART 16
2.8.1. The idle frame length will be 1 bit longer than the actual configured length 16
2.8.2. When USART is woken up from mute mode by an idle frame, it will not be woken up when it enters mute mode again 16
2.8.3. When USART works in multi-processor communication mode and the wake-up mode is address match detection, the parity check function does not work 17
2.8.4. USART sends one more baud rate cycle stop bit when sending data 17
2.9. I2C 17
2.9.1. Read one more data because the BTC flag was not cleared 17
2.9.2. I2C works in SMBUS mode, if SCL is pulled down, the first timeout will be 25ms, after that, if SCL continues to pull down, the timeout will be 130ms 18
2.9.3. Switching I2CEN from the disabled state to the enabled state will probabilistically cause the I2C to send a start signal 18
2.10. SPI 18
2.10.1. When SPI acts as a slave, the existence of other SPI slaves on the bus is not supported 18
2.10.2. SPI acts as a slave and when the number of SPI input clocks is not a multiple of 8 or 16, the chip selection is turned off at this time and the clock reception count will not be cleared 19
2.11. EXMC 19
2.11.1. When using the ECC function of NAND controller, ECC detection fails if 1bit error occurs and the 13th bit occurs in the data transmission process 19
2.12. CAN 19
2.12.1. The automatic offline recovery time of CAN is long 19
2.13. ENET 20
2.13.1. Reception data frame is dropped when enable hardware checksum and the header checksum is 0x0000 20
2.13.2. When the operate on second frame feature is enabled, the ENET module will probabilistically encounter transmission issues after running for a period of time 20
2.13.3. When the ENET RxFIFO operates in cut-through mode, receive descriptor errors will probabilistically occur 20
2.13.4. In scenarios of extremely frequent frame reception, the receive interrupt will probabilistically fail to trigger 21
2.14. USBD 21
2.14.1. The USBD ESOF interrupt is triggered during normal operation of a USB device 21
2.14.2. Large amounts of IN and OUT direction data transmission between the USBD device and the

host will probabilistically result in errors in the transmitted and received data content 22
2.15. USBFS 22
2.15.1. When USBFS-OTG is communicating with low-speed devices through the HUB, the data transmission is abnormal 22
2.15.2. The USBFS-OTG suspension state entry time is 6 ms instead of 3 ms 22
3. Revision history 23

List of Figures

Figure 1-1. Device revision code of GD32F10x 7

List of Tables

Table 1-1. Applicable products **7**
Table 1-2. Device limitations **7**
Table 3-1. Revision history **23**

1. Introduction

This document applies to GD32F10x product series, as shown in ***Table 1-1. Applicable products***. It offers technical guidance for using GD32MCU and provides workaround to current device limitations.

Table 1-1. Applicable products

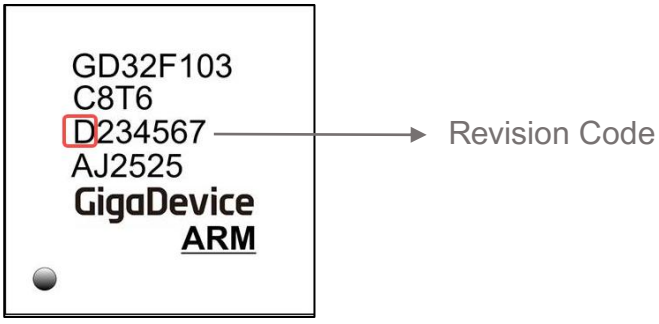
Type	Part Numbers
MCU	GD32F101xx series
	GD32F103xx series
	GD32F105xx series
	GD32F107xx series

Note: For GD32F101xx and GD32F103xx series, this document is only for the chip that flash is more than 128KB.

1.1. Revision identification

The device revision can be identified according to the mark on the top of the package. The 1st code on Line 3 of the mark is the product revision code, as shown in ***Figure 1-1. Device revision code of GD32F10x***.

Figure 1-1. Device revision code of GD32F10x



1.2. Summary of device limitations

The device limitations of GD32F10x 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 D
PMU	***MCU can not be waked up when an interrupt occurs before entering deep-sleep***	Y
RCU	***MCU can not be waked up after entering deep-sleep mode when***	Y

Module	Limitations	Workaround
		Rev. Code D
	DSLP_HOLD bit is set	
	When the HXTAL uses bypass mode, MCU power consumption will increase in sleep and standby mode	Y
	When the HXTAL clock monitor is enabled, the HXTAL clock stuck interrupt flag will be set after entering sleep mode	Y
GPIO	*When the OSCIN and OSCOUT pins are remapped to PD0/PD1 pins, PD0/PD1 can be used as a normal GPIO function but cannot be used as an EXTI function*	Y
	When the GPIO port is configured as a pull-up input and there is leakage in this GPIO, the voltage on the GPIO port will be pulled down	Y
	When PA11 and PA12 are not used as USB function and there is a large voltage difference between PA11 and PA12, there will be leakage from VDD to VSS	Y
DMA	*In the DMA interrupt handler function, exiting the interrupt immediately after performing the flag clear operation will cause the CPU to enter DMA interrupt*	Y
ADC	*ADC data acquisition error occurs when the ADC clock is much slower than the PCLK clock*	Y
	When the voltage difference between VREFP and VDDA exceeds 300mV, it will lead to poor ADC accuracy	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
TIMER	*Data lost when using timer capture / compare event to trigger DMA transfer and enabling the output compare shadow function*	Y
	TIMER in different channels, channel comparison/interrupt flag clearing and channel comparison/capture interrupt flag setting conflict, resulting in flag bit not set	Y
	The master TIMER triggers slave timers under different bus, resulting in phase differences in the output of different slave timers under the same comparison value	N
USART	*The idle frame length will be 1 bit longer than the actual configured length*	N
	When USART is woken up from mute mode by an idle frame, it will not be woken up when it enters mute mode again	Y
	When USART works in multi-processor communication mode and the wake-up mode is address match detection, the parity check function does not work	Y
	USART sends one more baud rate cycle stop bit when sending data	N
I2C	*Read one more data because the BTC flag was not cleared*	Y

Module	Limitations	Workaround
		Rev. Code D
	I2C works in SMBUS mode, if SCL is pulled down, the first timeout will be 25ms, after that, if SCL continues to pull down, the timeout will be 130ms	N
	Switching I2CEN from the disabled state to the enabled state will probabilistically cause the I2C to send a start signal	N
SPI	*When SPI acts as a slave, the existence of other SPI slaves on the bus is not supported*	N
	SPI acts as a slave and when the number of SPI input clocks is not a multiple of 8 or 16, the chip selection is turned off at this time and the clock reception count will not be cleared	Y
EXMC	*When using the ECC function of NAND controller, ECC detection fails if 1bit error occurs and the 13th bit occurs in the data transmission process*	N
CAN	*The automatic offline recovery time of CAN is long*	Y
ENET	*Reception data frame is dropped when enable hardware checksum and the header checksum is 0x0000*	Y
	When the operate on second frame feature is enabled, the ENET module will probabilistically encounter transmission issues after running for a period of time	Y
	When the ENET RxFIFO operates in cut-through mode, receive descriptor errors will probabilistically occur	Y
	In scenarios of extremely frequent frame reception, the receive interrupt will probabilistically fail to trigger	Y
USBD	*The USBD ESOF interrupt is triggered during normal operation of a USB device*	Y
	Large amounts of IN and OUT direction data transmission between the USBD device and the host will probabilistically result in errors in the transmitted and received data content	Y
USBFS	*When USBFS-OTG is communicating with low-speed devices through the HUB, the data transmission is abnormal*	N
	The USBFS-OTG suspension state entry time is 6 ms	N

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. MCU can not be waked up when an interrupt occurs before entering deep-sleep

Description & impact

When system application programme exists other interrupt code (such as systick / timer period interrupt) and needs to enter deep-sleep mode through WFE or WFI instruction, the system will exist the risk of not being wakedup after running for some time.

Workarounds

Application programme needs to mask all interrupts except EXTI for wakeup source before entering deep-sleep mode, then unmask the needed interrupts after wakeup.The latest firmware library has circumvented this problem. The reference code is as follow:

```
void pmu_to_deepsleepmode(uint32_t ldo,uint8_t deepsleepmodecmd)
{
    static uint32_t reg_snap[ 4 ];
    /* clear stbmod and ldolp bits */
    PMU_CTL &= ~((uint32_t)(PMU_CTL_STBMOD | PMU_CTL_LDOLP));

    /* set ldolp bit according to pmu_ldo */
    PMU_CTL |= ldo;

    /* set sleepdeep bit of Cortex-M3 system control register */
    SCB->SCR |= SCB_SCR_SLEEPDEEP_Msk;

    reg_snap[0] = REG32(0xE000E010U);
    reg_snap[1] = REG32(0xE000E100U);
    reg_snap[2] = REG32(0xE000E104U);
    reg_snap[3] = REG32(0xE000E108U);
    /* clear all interrupt except EXTI line wakeup source interrupt */
    REG32(0xE000E010U) &= 0x00010004U;
    REG32(0xE000E180U)   = 0XFF7FF83DU;
    REG32(0xE000E184U)   = 0XBFFFF8FFU;
    REG32(0xE000E188U)   = 0xFFFFFFFFU;

    /* select WFI or WFE command to enter deepsleep mode */
    if(WFI_CMD == deepsleepmodecmd){
```

```
        __WFI();
    }else{
        __SEV();
        __WFE();
        __WFE();
    }
    /* restore all interrupt */
    REG32(0xE000E010U) = reg_snap[0] ;
    REG32(0xE000E100U) = reg_snap[1] ;
    REG32(0xE000E104U) = reg_snap[2] ;
    REG32(0xE000E108U) = reg_snap[3] ;

    /* reset sleepdeep bit of Cortex-M3 system control register */
    SCB->SCR &= ~((uint32_t)SCB_SCR_SLEEPDEEP_Msk);
}
```

2.2. RCU

2.2.1. MCU can not be waked up after entering deep-sleep mode when DSLP_HOLD bit is set

Description & impact

When DSLP_HOLD is set, the MCU cannot be woked up after entering deep-sleep mode.

Workarounds

The application programme need switch the system clock to IRC8M before entering the deep-sleep mode.

2.2.2. When the HXTAL uses bypass mode, MCU power consumption will increase in sleep and standby mode

Description & impact

When the HXTAL uses bypass mode (HXTALBPS bit is set in RCU_CTL register), MCU power consumption will increase in sleep and standby mode.

Workarounds

Disable the HXTAL bypass mode (HXTALBPS bit is reset in RCU_CTL register) before entering sleep and standby mode.

2.2.3. When the HXTAL clock monitor is enabled, the HXTAL clock stuck interrupt flag will be set after entering sleep mode

Description & impact

When the HXTAL clock monitor function is enabled (CKMEN bit is set in the RCU_CTL register), the HXTAL clock stuck interrupt flag (CKMIF) will be set after entering sleep mode.

Workarounds

Disable the HXTAL clock monitor function (CKMEN bit is reset in the RCU_CTL register) before entering sleep mode.

2.3. GPIO

2.3.1. When the OSCIN and OSCOUT pins are remapped to PD0/PD1 pins, PD0/PD1 can be used as a normal GPIO function but cannot be used as an EXTI function

Description & impact

When the external crystal function is not used and the OSCIN and OSCOUT pins are remapped to PD0/PD1 pins, the remapped PD0/PD1 can be used as a normal GPIO function but cannot be used as an EXTI function.

Note: The above problem only exists in small package products with OSCIN/OSCOUT and PD0/PD1 sharing the same pin.

Workarounds

When the OSCIN and OSCOUT pins are remapped to PD0/PD1 pins, the use of PD0/PD1 for EXTI functionality is avoided on the application.

2.3.2. When the GPIO port is configured as a pull-up input and there is leakage in this GPIO, the voltage on the GPIO port will be pulled down

Description & impact

When a GPIO port is configured for pull-up input and there is a leakage in the GPIO port, the voltage on the GPIO port will be pulled down. For example, when the MCU is operating at 3.3V and the above problem is present, the GPIO port voltage will be pulled down to 2.6V instead of the normal 3.3V.

Workarounds

Not affect the use of GPIO function in the application.

2.3.3. When PA11 and PA12 are not used as USB function and there is a large voltage difference between PA11 and PA12, there will be leakage from VDD to VSS

Description & impact

When PA11 and PA12 are not used as USB functions and there is a large voltage difference between PA11 and PA12, there will be dozens of uA leakage from VDD to VSS, which will affect the power consumption in standby mode.

Workarounds

When PA11 and PA12 are not used as USB function, avoid voltage inconsistency between PA11 and PA12.

2.3.4. When both the USART1 and Ethernet clocks are enabled simultaneously, if the USART1 pins are not remapped, the Ethernet function will not work properly

Description & impact

Under the default configuration, since the Ethernet and USART1 pins are shared, when the USART1 and Ethernet clocks are enabled simultaneously, if the USART1 pins are not remapped, the Ethernet function will not work properly..

Workarounds

When using both USART1 and Ethernet functions simultaneously, remap the USART1 pins before using the Ethernet function.

2.4. DMA

2.4.1. In the DMA interrupt handler function, exiting the interrupt immediately after performing the flag clear operation will cause the CPU to enter DMA interrupt again

Description & impact

In the DMA interrupt handler function, exiting the interrupt immediately after performing the flag clear operation will cause the CPU to enter DMA interrupt again.

Workarounds

Place the flag clear operation at the beginning of the interrupt handler function and ensure that there is a delay of more than one clock cycle before the flag clear is performed and the interrupt is exited.

2.5. ADC

2.5.1. 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

After the EOC flag is set, delay two ADC clocks before reading the ADC_RDATA register.

2.5.2. When the voltage difference between V_{REFP} and V_{DDA} exceeds 300mV, it will lead to poor ADC accuracy

Description & impact

When the voltage difference between V_{REFP} and V_{DDA} exceeds 300mV, it will lead to poor ADC accuracy.

Workarounds

In the hardware circuit design, avoid the voltage difference between V_{REFP} and V_{DDA} exceeding 300mV.

2.6. DAC

2.6.1. 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.

2.7. TIMER

2.7.1. Data lost when using timer capture / compare event to trigger DMA transfer and enabling the output compare shadow function

Description & impact

When using timer capture / compare event to trigger DMA transfer and enabling the output compare shadow function, DMA transfers data 0x00 to TIMERx_CHyCV register which will result in the second data lost after data 0x00.

Workarounds

Use one of the following methods:

1) Do not use data 0x00 in DMA transfer buffer.
2) Transfer the second data after the 0x00 twice.
3) Use the timer update event to trigger DMA transfer.

2.7.2. TIMER in different channels, channel comparison/interrupt flag clearing and channel comparison/capture interrupt flag setting conflict, resulting in flag bit not set

Description & impact

TIMER in the same cycle that triggered the CHx channel comparison/capture interrupt, the software clear another channel comparison/capture interrupt flag CHyIF, will cause CHxIF not set.

Workarounds

Use one of the following methods:

1) Clear the CHyIF flag after CHxIF is set
2) Read the CHxCV value to determine whether the CHyIF flag can be cleared

2.7.3. The master TIMER triggers slave timers under different bus, resulting in phase differences in the output of different slave timers under the same comparison value

Description & impact

When configuring CK_APB2 =2*CK_APB1. TIMER3 under the APB1 bus acts as the master timer to simultaneously trigger TIMER0 on the APB2 bus and TIMER1 on the APB1 bus. The same overload value and are relatively straight, and there is a phase difference between the output of TIMER0 and TIMER1.

Workarounds

Not available.

2.8. USART

2.8.1. The idle frame length will be 1 bit longer than the actual configured length

Description & impact

The idle frame length is not consistent with that described by user manual. For example, when a data frame is configured with 1 start bit, 8 data bits, and 1 stop bit, the actual idle frame length will be 11 bits instead of 10 bits.

Workarounds

Not available.

2.8.2. When USART is woken up from mute mode by an idle frame, it will not be woken up when it enters mute mode again

Description & impact

When USART works in multiprocessor communication mode and the USART is woken from mute mode by an idle frame, it will cause the USART will not be woken up when the bus is in idle mode and the USART enters mute mode.

Workarounds

When an idle frame is used to wake the USART mute mode, it is not allowed to enter mute mode while the bus is idle.

2.8.3. When USART works in multi-processor communication mode and the wake-up mode is address match detection, the parity check function does not work

Description & impact

When USART works in multiprocessor communication mode and the wake-up mode is address match detection, when the parity function is enabled, the PERR cannot be set when a parity error occurs.

Workarounds

Disable parity function or use idle frame wakeup instead of address match wakeup.

2.8.4. USART sends one more baud rate cycle stop bit when sending data

Description & impact

USART sends one more baud rate cycle stop bit when sending data. When there is a requirement on the stop bit duration and no interval is allowed between two frames, there may be an impact on the communication.

Workarounds

If there is no requirement on the stop bit duration, there is no impact on the communication, otherwise GD32F30x series MCU is recommended.

2.9. I2C

2.9.1. 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 can not clear the BTC flag.

Workarounds

Use one of the following methods:

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)

2.9.2. I2C works in SMBUS mode, if SCL is pulled down, the first timeout will be 25ms, after that, if SCL continues to pull down, the timeout will be 130ms

Description & impact

In SMBUS mode, if SCL keeps low, the first timeout interrupt entry time is 25ms. If SCL keeps low, the subsequent interrupt entry time will be 130ms.

Workarounds

Not available.

2.9.3. Switching I2CEN from the disabled state to the enabled state will probabilistically cause the I2C to send a start signal

Description & impact

Switching I2CEN from the disabled state to the enabled state will probabilistically cause the I2C to send a start signal.

Workarounds

Not available.

2.10. SPI

2.10.1. When SPI acts as a slave, the existence of other SPI slaves on the bus is not supported

Description & impact

When SPI acts as a slave, other slaves are not allowed on the bus, and only one-to-one communication with the master is supported.

Workarounds

Not available. If the application requires support for SPI master-slaves mode, it is recommended to use the GD32F30x series as an alternative.

2.10.2. SPI acts as a slave and when the number of SPI input clocks is not a multiple of 8 or 16, the chip selection is turned off at this time and the clock reception count will not be cleared

Description & impact

When SPI acts as a slave and the number of SPI input clocks is not a multiple of 8 or 16 (determined by the FF16 bit in the SPI_CTL0 register), turning off chip selection will not clear the clock receive count, resulting in an additional wait for the corresponding clock number before sending or receiving new data.

Workarounds

When the above issue occurs, it can be avoided by first disabling SPI and then enabling SPI. Taking SPI0 as an example:

```
/* disable and then enable SPI0 */
spi_disable (SPI0);
spi_enable(SPI0);
```

2.11. EXMC

2.11.1. When using the ECC function of NAND controller, ECC detection fails if 1bit error occurs and the 13th bit occurs in the data transmission process

Description & impact

When using the ECC function of NAND controller, ECC detection fails if 1bit error occurs and the 13th bit occurs in the data transmission process.

Workarounds

Not available.

2.12. CAN

2.12.1. The automatic offline recovery time of CAN is long

Description & impact

When the CAN is in bus off state due to bus short or other bus exception, which cause bus to remain in a recessive state. After the CAN automatic offline recovery (ABOR) function is

enabled, this will result in long offline recovery time due to the strict recovery time requirements of this CAN IP.

Workarounds

Enable an offline interrupt and reinitialize the CAN module in the offline interrupt.

2.13. ENET

2.13.1. 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.

2.13.2. When the operate on second frame feature is enabled, the ENET module will probabilistically encounter transmission issues after running for a period of time

Description & impact

When the operate on second frame feature is enabled (OSF = 1), the ENET module will probabilistically encounter issues after running for a period of time, such as being unable to send frames or each frame actually sent being the frame that was intended to be sent during the previous transmission.

Workarounds

Do not use the second frame operation and ensure that the OSF bit in the ENET_DMA_CTL register remains set to 0 at all times. Currently, in GD's official firmware library and ENET examples, this bit is kept cleared by default during the initialization process. If developer is not using GD's firmware library, the corresponding configuration in the ENET initialization code needs to be modified to ensure that the OSF bit is set to 0.

2.13.3. When the ENET RxFIFO operates in cut-through mode, receive descriptor errors will probabilistically occur

Description & impact

When the ENET RxFIFO operates in cut-through mode and the receive threshold is set too low, the ENET DMA will probabilistically write back an incorrect status, causing descriptor errors.

Workarounds

Set the RxFIFO to store-and-forward mode or configure a higher receive threshold in cut-through mode.

2.13.4. In scenarios of extremely frequent frame reception, the receive interrupt will probabilistically fail to trigger

Description & impact

If the received data is frequent and the software cannot process it in time, it will probabilistically cause a receive overflow, eventually leading to the failure of the receive interrupt to be triggered.

Workarounds

Keep the DAFRF bit in the ENET_DMA_CTL register at its reset value to allow RxDMA to flush received data frames.

2.14. USBD

2.14.1. The USBD ESOF interrupt is triggered during normal operation of a USB device

Description & impact

The ESOF interrupt for USBD devices is triggered during normal operation of a USB device. If the USB_EPxCS register is configured or the USBD force reset operation is performed in the ESOF interrupt handler function, USB data communication will be interrupted, and even user Buffer overflow will be caused.

Workarounds

During ESOF interrupts, the USB_EPxCS register is not configured and no USBD force reset is performed. It is recommended to follow the ESOF interrupt handling mechanism in the GD official firmware library.

2.14.2. Large amounts of IN and OUT direction data transmission between the USBD device and the host will probabilistically result in errors in the transmitted and received data content

Description & impact

When the USBD device communicates with the host, if a large amount of IN and OUT direction data transmission occurs simultaneously, it will probabilistically lead to errors in the content of data packets sent and received in the IN and OUT directions.

Workarounds

The USBD only uses half of the SRAM sector space, limiting the USBD SRAM usage to either 0x40006000-0x40006200 (lower half of the sector space) or 0x40006200-0x40006400 (upper half of the sector space.

2.15. USBFS

2.15.1. When USBFS-OTG is communicating with low-speed devices through the HUB, the data transmission is abnormal

Description & impact

When USBFS-OTG is communicating with low-speed devices through the HUB, the data transmission is abnormal.

Workarounds

Not available.

2.15.2. The USBFS-OTG suspension state entry time is 6 ms instead of 3 ms

Description & impact

The USBFS-OTG suspension state entry time is 6 ms instead of 3 ms.

Workarounds

Not available. But it does not affect the normal use of USBFS-OTG.

3. Revision history

Table 3-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	May.21 2025
1.1	1. Delete I2C limitations, ***After winning arbitration as the I2C master, I2C master will probabilistically pull down the SCL clock line, causing the bus to be blocked*** and ***The I2C master cannot initiate a new access when arbitration is lost*** 2. Update the description of I2C limitation, refer to ***I2C works in SMBUS mode, if SCL is pulled down, the first timeout will be 25ms, after that, if SCL continues to pull down, the timeout will be 130ms*** 3. Update the description of CAN limitation, refer to ***The automatic offline recovery time of CAN is long***	Jun.10 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