GigaDevice Semiconductor Inc.

Device limitations of GD32E10x&C10x

Errata Sheet

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 **9**

2.1. FMC **9**

2.1.1. When performing double-word programming, if one of the words at the programming address is 0xFFFFFFFF, then executing the double-word programming operation at that address will not cause the PGERR bit to be set 9

2.2. PMU **9**

2.2.1. MCU cannot be waked up when an interrupt occurs before entering deep-sleep 9

2.2.2. Rapid drop and then rebound on V_{DD} will cause the MCU to crash or reset 10

2.3. RCU **10**

2.3.1. MCU cannot be waked up after entering deep-sleep mode when DSLP_HOLD bit is set .. 10

2.3.2. The LXTALSTB bit cannot be cleared by disabling LXTAL when LXTAL stops unexpectedly 10

2.3.1. When the system clock is CK_PLL and DSLP_HOLD is set, the MCU system clock is lost after entering deep sleep mode 11

2.4. ADC **11**

2.4.1. ADC data acquisition error occurs when the ADC clock is much slower than the PCLK clock 11

2.5. DAC **12**

2.5.1. DAC output pin exists electric leakage to VREFP pin when DAC is disabled 12

2.5.2. 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 12

2.6. TIMER **12**

2.6.1. Data lost when using timer capture / compare event to trigger DMA transfer and enabling the output compare shadow function 12

2.7. USART **13**

2.7.1. When USART is configured in DMA transmit mode and DMA channel is enabled before TEN is set, resulting in data loss 13

2.8. I2C **13**

2.8.1. Read one more data because the BTC flag was not cleared 13

2.8.2. 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 13
2.8.3. The 90ms timeout interrupt cannot be entered in SAM_V mode 14
2.8.4. Switching I2CEN from the disabled state to the enabled state may cause the I2C to send a start signal... 14
2.8.5. The I2C master cannot initiate a new access when arbitration is lost................................. 14
2.8.6. When SCL is pulled low, the timeout period for the first transaction is 25ms, and for all subsequent transactions, the timeout period is 130ms 14

2.1. CAN 15
2.1.1. There is a risk of data misalignment when the CAN receives a large amount of data......... 15

2.2. USBFS 15
2.2.1. The USBFS-OTG suspension state entry time is not 3ms 15

2.3. Core 16
2.3.1. VDIV or VSQRT instructions might not complete correctly when very short ISRs are used 16

3. Revision history 18

List of Figures

Figure 1-1. Device revision code of GD32E10x&C10x 6

List of Tables

Table 1-1. Applicable products 6
Table 1-2. Device limitations 6
Table 3-1. Revision history 18

1. Introduction

This document applies to GD32E10x&C10x 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.

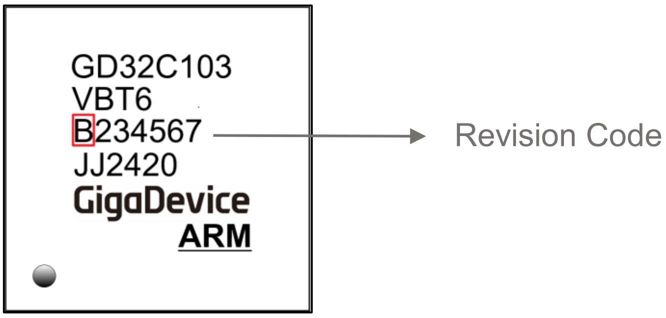
Table 1-1. Applicable products

Type	Part Numbers
MCU	GD32E103xx series
	GD32C103xx 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 GD32E10x&C10x***.

Figure 1-1. Device revision code of GD32E10x&C10x



1.2. Summary of device limitations

The device limitations of GD32E10x&C10x 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
FMC	***When performing double-word programming, if one of the words at the programming address is 0xFFFFFFFF, then executing the double-word programming operation at that address will not cause the PGERR bit to be set***	Y	Y
PMU	***MCU cannot be waked up when an interrupt occurs before entering deep-sleep***	Y	Y

	Rapid drop and then rebound on VDD will cause the MCU to crash or reset	Y	Y
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
	When the system clock is CK_PLL and DSLP_HOLD is set, the MCU system clock is lost after entering deep sleep mode	Y	Y
ADC	*ADC data acquisition error occurs when the ADC clock is much slower than the PCLK clock*	Y	Y
DAC	*DAC output pin exists electric leakage to VREFP pin when DAC is disabled*	N	N
	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	Y
TIMER	*Data lost when using timer capture / compare event to trigger DMA transfer and enabling the output compare shadow function*	Y	Y
USART	*When USART is configured in DMA transmit mode and DMA channel is enabled before TEN is set, resulting in data loss*	Y	Y
I2C	*Read one more data because the BTC flag was not cleared*	Y	Y
	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	N	N
	The 90ms timeout interrupt cannot be entered in SAM_V mode	Y	Y
	Switching I2CEN from the disabled state to the enabled state may cause the I2C to send a start signal	N	N
	The I2C master cannot initiate a new access when arbitration is lost	N	N
	When SCL is pulled low, the timeout period for the first transaction is 25ms, and for all subsequent transactions, the timeout period is 130ms	Y	Y
CAN	*There is a risk of data misalignment when the CAN receives a large amount of data*	Y	Y
USBFS	*The USBFS-OTG suspension state entry time is not 3ms*	N	N
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. FMC

2.1.1. When performing double-word programming, if one of the words at the programming address is 0xFFFFFFFF, then executing the double-word programming operation at that address will not cause the PGERR bit to be set

Description & impact

When performing double-word programming, if one of the words at the programming address is 0xFFFFFFFF, then executing the double-word programming operation at that address will not cause the PGERR bit to be set, but the programmed data will be incorrect.For example, if the value at address 0x08001000 is 0xFFFFFFFF A55AA55A, then performing a double-word programming operation at 0x08001000 next time will not cause the PGERR bit to be set.

Workarounds

Ensure that the data at the address is empty before programming. Before programming the same address, programme need to perform an erase operation on that address area.

2.2. PMU

2.2.1. MCU cannot 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

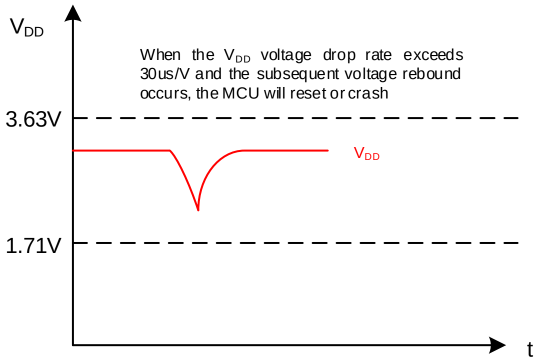
Application programme needs to mask all interrupts except EXTI for wakeup source before entering deep-sleep mode, then unmask the needed interrupts after wakeup.The latest firmware library has circumvented this problem.

2.2.2. Rapid drop and then rebound on V_{DD} will cause the MCU to crash or reset

Description & impact

When the V_{DD} drop rate exceeds 30us/V (for example, V_{DD} drops from 3.63V to 1.71V at a rate of 25μs/V) and then V_{DD} voltage rebound, the MCU will crash or reset.

For example, when the MCU operates at V_{DD}, the V_{DD} voltage drops due to the CMOS power transistor being turned on, and then the subsequent capacitor charging restores the V_{DD} voltage.


V_{DD}
When the V_{DD} voltage drop rate exceeds 30us/V and the subsequent voltage rebound occurs, the MCU will reset or crash
3.63V
V_{DD}
1.71V
t

Workarounds

Ensure that the V_{DD} drop rate does not exceed 30μs/V in the PCB design.

2.3. RCU

2.3.1. 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.

2.3.2. 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);
    }
}
```

2.3.1. When the system clock is CK_PLL and DSLP_HOLD is set, the MCU system clock is lost after entering deep sleep mode

Description & impact

When the system clock is CK_PLL and DSLP_HOLD is set, the MCU system clock is lost after entering deep sleep mode due to IRC8M clock is not switched successfully.

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.

2.4. ADC

2.4.1. 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.

2.5. DAC

2.5.1. DAC output pin exists electric leakage to VREFP pin when DAC is disabled

Description & impact

When DAC is disabled and V_{REFP} is smaller than V_{DD} exceeding 0.7V, DAC output pin exists electric leakage to VREFP pin.

Workarounds

Avoid V_{REFP} is smaller than V_{DD} exceeding 0.7V.

2.5.2. 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.

2.6. TIMER

2.6.1. 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. USART

2.7.1. 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 iin the USART_CTL0 register before the CHEN bit of the corresponding DMA_CHxCTL register is enabled.

2.8. I2C

2.8.1. 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

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).

2.8.2. 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

Description & impact

The timing of the START signal of I2C0 and I2C1 is different, which leads to the misplaced SCL clock signal sent out, and more than 8 clock signals are sent during the address transmission phase. As a result, the master that wins the arbitration cannot receive ACK, so the SCL signal line is pulled down and the I2C bus is stuck.

Workarounds

Not available.

2.8.3. The 90ms timeout interrupt cannot be entered in SAM_V mode

Description & impact

When the SAM_V interface timeout detection function is enabled, the 90ms timeout interrupt cannot be triggered.

Workarounds

Enable the SMBus host mode by setting SMBSEL and SMBEN, and disable STOEN.

2.8.4. Switching I2CEN from the disabled state to the enabled state may cause the I2C to send a start signal

Description & impact

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

Workarounds

Not available.

2.8.5. The I2C master cannot initiate a new access when arbitration is lost

Description & impact

The I2C master cannot initiate a new access when arbitration is lost.

Workarounds

Not available.

2.8.6. When SCL is pulled low, the timeout period for the first transaction is 25ms, and for all subsequent transactions, the timeout period is 130ms

Description & impact

By setting TOEN to enable the timeout counting function, when SCL is pulled low, the timeout period is 25ms for the first timeout and 130ms for subsequent timeouts.

Workarounds

In the I2C timeout interrupt handler, first disable (TOEN = 0)and then enable (TOEN = 1) the timeout counting function by.

2.1. CAN

2.1.1. There is a risk of data misalignment when the CAN receives a large amount of data

Description & impact

When CAN receives a large amount of data and a data overrun error occurs, it will cause related status flag bit errors, which will lead to subsequent data receiving misalignment and data confusion.

Workarounds

Software processing suggestions are as follows:

1) Suggestions for reducing the occurrence of data overrun:
   - Add filter
   - Improve the speed of software interrupt processing to read data promptly
2) Suggestions for handling data overrun:
   - Immediately read 3 frames of data through the overrun interrupt, then set SWRST to reset the CAN
   - Set the CAN status bits to Initial or Sleep, read 3 frames of data, and then set it back to Normal status

2.2. USBFS

2.2.1. The USBFS-OTG suspension state entry time is not 3ms

Description & impact

The USBFS-OTG suspension state entry time is not 3ms

Workarounds

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

2.3. Core

2.3.1. 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 VQSRT 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	Sep.9 2024

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.