GigaDevice Semiconductor Inc.

Device limitations of GD32F30x

Errata Sheet

Table of Contents

List of Figures

List of Tables

1. Introduction

This document applies to GD32F30x 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.

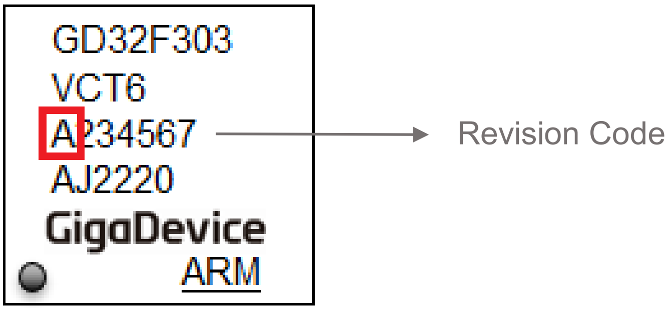
Table 1-1. Applicable products

Type	Part Numbers
MCU	GD32F303xx series
	GD32F305xx series
	GD32F307xx 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 GD32F30x***.

Figure 1-1. Device revision code of GD32F30x



1.2. Summary of device limitations

The device limitations of GD32F30x 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 C
PMU	*Standby mode can not be entered normally*	Y	Y
	Power consumption is higher in deep-sleep mode	Y	Y
	Standby mode cannot be waked up due to frequent wakeup signals before or after entering standby mode	N	N

RCU	***MCU can not be waked up after entering deep-sleep / standby mode when DSLP_HOLD / STB_HOLD bit is set***	Y	Y
ADC	***ADC sampling distorts during the calibration***	Y	Y
DAC	***DAC output pin exists electric leakage to VREF+ pin when DAC is disabled***	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	***Data lost when using USART DMA transmission***	Y	Y
	Mute mode can be waked up as long as the USART_CTL0 register is operated after mute mode is enabled	Y	Y
I2C	***Read one more data because the BTC flag was not cleared***	Y	Y
EXMC	***NE timing can not satisfy the requirement when using NAND pre-waiting function***	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 can not be entered normally

Description & impact

When system application programme exists other interrupt code (such as systick 1us period interrupt) and needs to enter standby mode through WFI instruction, the system can not enter the standby mode normally.

Workarounds

Application programme needs to mask all interrupts except RTC wakeup source before entering standby mode.

2.1.2. Power consumption is higher in deep-sleep mode

Description & impact

Power consumption is higher in deep-sleep mode.

Workarounds

Application programme can configure I/O (not used, including internal I/O) to analog mode to reduce the power consumption.

2.1.3. 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 can not be waked up after entering deep-sleep / standby mode when DSLP_HOLD / STB_HOLD bit is set

Description & impact

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

Workarounds

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

2.3. ADC

2.3.1. ADC sampling distorts during the calibration

Description & impact

When application programme executes the adc calibration function after power up, V_{DDA} voltage generates fluctuation which results in the ADC sampling value distortion.

Workarounds

1) Add 1ms delay after ADC is enabled and before calibrating.
2) Connect a 1uF (better for an extra 10nF) capacitor in parallel with V_{DDA} pin.
3) Select a right bead between V_{DDA} and V_{DD}.

2.4. DAC

2.4.1. DAC output pin exists electric leakage to V_{REF+} pin when DAC is disabled

Description & impact

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

Workarounds

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

2.5. TIMER

2.5.1. 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

Do not use data 0x00 in DMA transfer buffer.

2.6. USART

2.6.1. Data lost when using USART DMA transmission

Description & impact

Application programme configurations follow the such step:

1) Disable USART transmitter.
2) Configure the DMA channel counter.
3) Enable DMA channel.
4) Enable USART transmitter.

When using the above configuration, the transmission data is lost.

Workarounds

Adjust the configurations code sequence as follow:

1) Disable USART transmitter.
2) Configure the DMA channel counter.
3) Enable USART transmitter.
4) Enable DMA channel.

2.6.2. 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.

2.7. I2C

2.7.1. 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

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. EXMC

2.8.1. 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.

2.9. 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 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	May.19 2022
1.1	1. Update the I2C limitation workarounds description 2. Update note of chapter 1.2	Apr.6 2023
1.2	1. Add PMU limitation, referring to chapter 2.1.3 2. Add core limitation, referring to chapter 2.9.1 3. Add limitations of Rev. Code C	Nov.2 2023

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.