STM32W108xx Errata sheet

STM32W108xx device limitations

Silicon identification

This errata sheet applies to the STMicroelectronics STM32W108xx microcontroller family.

The STM32W108xx family features an ARM™ 32-bit Cortex®-M3 core, for which an errata notice is also available (see *Section 1: ARM™ 32-bit Cortex®-M3 limitations* for details).

The full list of part numbers is shown in *Table 2*. The products are identifiable as shown in the following table.

Table 1. Device identification

Reference	Silicon ID (stored in location 0x40004000)	Comment
STM32W108C8U6x	0x269A862B	Cut 1.3, x=Blank,1,3,4
STM32W108HBU6x	0x269A862B	Cut 1.3, x=Blank,1,3,4
STM32W108CBU6x	0x269A862B	Cut 1.3, x=Blank,1,3,4
STM32W108CCU6x	0x069B0041	Cut 1.0, x=Blank,1,3,4
STM32W108CZU6x	0x069B0041	Cut 1.0, x=Blank,1,3,4

Table 2. Device summary

Reference	Part number
STM32W108xx	STM32W108C8, STM32W108HB, STM32W108CB, STM32W108CC, STM32W108CZ

Contents

1 ARM™ 32-bit Cortex®-M3 limitations

An errata notice of the STM32W108xx core is available from the following web address: *http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.eat0420a/*.
The direct link to the errata notice pdf is:
http://infocenter.arm.com/help/topic/com.arm.doc.eat0420a/Cortex-M3-Errata-r1p1-v0.2.pdf.

All the described limitations are minor and related to the revision r1p1-01rel0 of the Cortex-M3 core. *Table 3* summarizes these limitations and their implications on the behavior of high-density STM32W108xx devices.

Table 3. Cortex-M3 core limitations and impact on microcontroller behavior

ARM ID	ARM category	ARM summary of errata	Impact on high-density STM32W108xx devices
752419	Cat 2	Interrupted loads to SP can cause erroneous behavior	Minor
740455	Cat 2	SVC and BusFault/MemManage may occur out of order	Minor
602117	Cat 2	LDRD with base in list may result in incorrect base register when interrupted or faulted	Minor
563915	Cat 2	Event register is not set by interrupts and debug	Minor
531064	impl	SWJ-DP missing POR reset sync	No
511864	Cat 3	Cortex-M3 may fetch instructions using incorrect privilege on return from an exception	No
532314	Cat 3	DWT CPI counter increments during sleep	No
538714	Cat 3	Cortex-M3 TPIU clock domain crossing	No
548721	Cat 3	Internal write buffer could be active whilst asleep	No
463763	Cat 3	BKPT in debug monitor mode can cause DFSR mismatch	Minor
463764	Cat 3	Core may freeze for SLEEPONEXIT single instruction ISR	Minor
463769	Cat 3	Unaligned MPU fault during a write may cause the wrong data to be written to a successful first access	No

1.1 Cortex-M3 limitations description for STM32W108xx high-density devices

Only the limitations described below have an impact, even though minor, on the implementation of STM32W108xx high-density devices.

All the other limitations described in the ARM errata notice (and summarized in *Table 3* above) have no impact and are not related to the implementation of STM32W108xx high-density devices (Cortex-M3 r1p1-01rel0).

1.1.1 Cortex-M3 LDRD with base in list may result in incorrect base register when interrupted or faulted

Description

The Cortex-M3 Core has a limitation when executing an LDRD instruction from the system-bus area, with the base register in a list of the form LDRD Ra, Rb, [Ra, #imm]. The execution may not complete after loading the first destination register due to an interrupt before the second loading completes or due to the second loading getting a bus fault.

Workarounds

1. This limitation does not impact the STM32W108xx code execution when executing from the embedded Flash memory, which is the standard use of the microcontroller.
2. Use the latest compiler releases. As of today, they no longer generate this particular sequence. Moreover, a scanning tool is provided to detect this sequence on previous releases (refer to your preferred compiler provider).

1.1.2 Cortex-M3 event register is not set by interrupts and debug

Description

When interrupts related to a WFE occur before the WFE is executed, the event register used for WFE wakeup events is not set and the event is missed. Therefore, when the WFE is executed, the core does not wake up from WFE if no other event or interrupt occur.

Workaround

Use STM32W108xx external events instead of interrupts to wake up the core from WFE by configuring an external or internal EXTI line in event mode.

1.1.3 Cortex-M3 BKPT in debug monitor mode can cause DFSR mismatch

Description

A BKPT may be executed in debug monitor mode. This causes the debug monitor handler to be run. However, the bit 1 in the Debug fault status register (DFSR) at address 0xE000ED30 is not set to indicate that it was originated by a BKPT instruction. This only occurs if an interrupt other than the debug monitor is already being processed just before the BKPT is executed.

Workaround

If the DFSR register does not have any bit set when the debug monitor is entered, this means that we must be in this “corner case” and so, that a BKPT instruction was executed in debug monitor mode.

1.1.4 Cortex-M3 may freeze for SLEEPONEXIT single instruction ISR

Description

If the Cortex-M3 SLEEPONEXIT functionality is used and the concerned interrupt service routine (ISR) contains only a single instruction, the core becomes frozen. This freezing may occur if only one interrupt is active and it is preempted by an interrupt whose handler only contains a single instruction.

However, any new interrupt that causes a preemption would cause the core to become unfrozen and behave correctly again.

Workaround

This scenario does not happen in real application systems since all enabled ISRs should at least contain one instruction. Therefore, if an empty ISR is used, then insert an NOP or any other instruction before the exit instruction (BX or BLX).

1.1.5 Interrupted loads to SP can cause erroneous behavior

Description

If an interrupt occurs during the data-phase of a single word load to the stack-pointer (SP/R13), erroneous behavior can occur. In all cases, returning from the interrupt will result in the load instruction being executed an additional time. For all instructions performing an update to the base register, the base register will be erroneously updated on each execution, resulting in the stack-pointer being loaded from an incorrect memory location.

The affected instructions are:

1. LDR SP,[Rn],#imm
2. LDR SP,[Rn,#imm]!
3. LDR SP,[Rn,#imm]
4. LDR SP,[Rn]
5. LDR SP,[Rn,Rm]

Workaround

As of today, there is no compiler generating these particular instructions. This limitation can only occur with hand-written assembly code.

Both issues may be worked around by replacing the direct load to the stack-pointer, with an intermediate load to a general-purpose register followed by a move to the stack-pointer.

Example: the following instruction "LDR SP, [R0]" can be replaced by

“LDR R2,[R0]

MOV SP,R2 "

1.1.6 SVC and BusFault/MemManage may occur out of order

Description

If an SVC exception is generated by executing the SVC instruction while the following instruction fetch is faulted, then the MemManage or BusFault handler may be entered even though the faulted instruction which followed the SVC should not have been executed.

Workaround

A workaround is only required if the SVC handler will not return to the return address that has been stacked for the SVC exception and the instruction access after the SVC will fault. If this is the case then padding can be inserted between the SVC and the faulting area of code, for example, by inserting NOP instructions.

2 STM32W108xx silicon limitations

Table 4. Summary of silicon limitations

Links to silicon limitations	Silicon rev.
Section 2.1: SPI master mode does not work up to 12 MHz	Cut 1.3, x=Blank,1,3,4
Section 2.2: CPU cannot be run at half selected oscillator frequency with zero flash wait states	Cut 1.3, x=Blank,1,3,4
Section 2.3: Debugger accesses to invalid addresses wrongly set by AFSR flags	Cut 1.3, x=Blank,1,3,4
Section 2.4: ADC Data register reads wrong value if read at wrong time	Cut 1.3, x=Blank,1,3,4
Section 2.5: Flash half-cycle mode	Cut 1.3, x=Blank,1,3,4
Section 2.6: ADC high voltage mode drift	Cut 1.3, x=Blank,1,3,4
Section 2.7: ADC high voltage limitation in differential mode	Cut 1.0, x=Blank,1,3,4
Section 2.8: SLEEP_COUNT_H/L after a deep sleep sometimes reports a stale value	Cut 1.3, x=Blank,1,3,4
Section 2.9: UART can miss bytes when there is a baud rate error	Cut 1.3, x=Blank,1,3,4
Section 2.10: JTAG is not accessible when the system is under reset	– Cut 1.0, x=Blank,1,3,4 – Cut 1.3, x=Blank,1,3,4
Section 2.11: Serial controllers used in SPI slave mode do not tri-state MISO	– Cut 1.0, x=Blank,1,3,4 – Cut 1.3, x=Blank,1,3,4

2.1 SPI master mode does not work up to 12 MHz

Part number affected

- STM32W108C8U6x
- STM32W108HBU6x
- STM32W108CBU6x

Description

The specified part numbers work up to 6 MHz in master mode.

Workaround

None

2.2 CPU cannot be run at half selected oscillator frequency with zero flash wait states

Part number affected

- STM32W108C8U6x
- STM32W108HBU6x
- STM32W108CBU6x

Description

There is a bug in the chip, which means that operating the CPU with zero flash wait states at 12 MHz derived from the 24 MHz crystal, or 6 MHz derived from the high frequency RC oscillator, can cause a memory protection fault with certain instruction combinations.

Workaround

None

2.3 Debugger accesses to invalid addresses wrongly set by AFSR flags

Part number affected

- STM32W108C8U6x
- STM32W108HBU6x
- STM32W108CBU6x

Description

If the debugger reads or writes an address within the gap between the IC peripheral register block, AFSR (Auxiliary Fault Status Register), the RESERVED bit flag is set. If the access was 8 or 16 bit, then the WRONGSIZE flag is also set. AFSR flags remain set until cleared by the processor or by a reset. The AFSR flags must only be set if a processor bus fault is registered. If an AFSR flag was set due to debugger activity, it may confuse crash diagnosis and in general reduces the value of the AFSR information.

Workaround

None

2.4 ADC Data register reads wrong value if read at wrong time

Part number affected

- STM32W108C8U6x
- STM32W108HBU6x
- STM32W108CBU6x

Description

The ADC returns the wrong value if the data register is read while correction calculations are in progress. If the multiplication by the gain factor is in progress, the value returned is the partial product as it has been developed up until that point. The window in which the bad results are returned is relatively brief, depends on the gain factor, and is, at most, 16 or 17 ADC clocks.

Workaround

None

2.5 Flash half-cycle mode

Part number affected

- STM32W108C8U6x
- STM32W108HBU6x
- STM32W108CBU6x

Description

Half-cycle mode cannot be used safely. The result of not using this mode is that the Flash is turned on for the full CPU access cycle in half-clock mode (12 MHz from crystal, or 6 MHz from oschf), which increases current consumption.

Workaround

None

2.6 ADC high voltage mode drift

Part number affected

- STM32W108C8U6x
- STM32W108HBU6x
- STM32W108CBU6x

Description

There is an issue in the high voltage mode of the general purpose ADC. The input buffer used in this mode experiences long term drift of its input offset voltage. The input buffer is a rail-to-rail op-amp, which means that the offset is not constant over the whole input voltage range, and hence it cannot be calibrated out. The offset voltage results in a severe degradation to ADC accuracy in this mode. The standard low voltage (0-1.2 V) input range is not affected by this issue.

Workaround

Only the 1.2 V input range mode of the ADC should be used. If measurement of signals greater than 1.2 V is required, then external attenuation should be added.

2.7 ADC high voltage limitation in differential mode

Part number affected

- STM32W108CCU6x
- STM32W108CZU6x

Description

There is an issue in the high voltage mode of the general purpose ADC in differential mode.

Workaround

Only high voltage single mode is supported with software control of the HVSELN register to control the chopper.

2.8 SLEEP_COUNT_H/L after a deep sleep sometimes reports a stale value

Part number affected

- STM32W108C8U6x
- STM32W108HBU6x
- STM32W108CBU6x

Description

There is an issue in the sleep timer read register, with the result that it does not contain the correct value immediately after waking up from deep sleep. The register is updated with the correct value when the next sleep timer clock tick occurs.

Workaround

The system timer software module provided in the HAL library handles all interactions with the sleep timer, including time-based deep sleeping via the deep sleep module in the HAL library. These modules implement a software workaround. The system timer module will return the correct value in all situations. However, in the case where the chip performs a deep sleep that maintains the system time and is woken up from an external event (for example, not a sleep timer compare event), the deep sleep module in the HAL library delays until the next sleep timer clock tick (up to 1 ms), in order to guarantee that the read register updates correctly.

2.9 UART can miss bytes when there is a baud rate error

Part number affected

- STM32W108C8U6x
- STM32W108HBU6x
- STM32W108CBU6x

Description

There is a bug in the UART receiver that means it can receive corrupt bytes if the interbyte gap is long, or there is a baud rate mismatch between receive and transmit. The UART may detect a parity and/or framing error on the corrupt byte, but there will not necessarily be any error detected.

Errors may occur when T_{gap} >= 106 / (baud * f_{error})

where:

- T_{gap} = inter-byte gap in s
- baud = baud rate in bps
- f_{error} = relative frequency error in ppm

For example, if the baud rate tolerance between receive and transmit is 200 ppm (reasonable if both sides are derived from a crystal), and the baud rate is 115200 bps, then errors will not occur until the inter-byte gap exceeds 43 ms. If this gap is exceeded, the chance of an error is essentially random, with a probability of approximately:

P = baud / 24e6

At 115200 bps, the probability of corruption is 0.5%.

Workaround

The EM35x UART should be operated in systems where the other side of the communication link also uses a crystal as its timing reference, and baud rates should be selected to minimize the baud rate mismatch to the crystal tolerance.

UART protocols should contain some form of error checking (CRC, for example) at the packet level to detect, and retry in the event of errors. Since the probability of corruption is low, then there will only be a small effect on UART throughput due to retries.

2.10 JTAG is not accessible when the system is under reset

Part number affected

- STM32W108C8U6x
- STM32W108HBU6x
- STM32W108CBU6x
- STM32W108CCU6x
- STM32W108CZU6x

Description

There is a limitation that prevents access to JTAG when the chip is under reset. This is normally not a problem and the only case where it becomes critical is when a part is programmed with a code that continuously issues software reset, and the time between releasing the reset pin and the software reset is below a certain threshold. In such a scenario, the part is no longer be accessible with JTAG.

Workaround

Loss of JTAG access is a critical effect of this bug and the only reliable workaround is to force the device to boot from the system memory, which contains the reliable code, and then force a mass erase to recover the part.

2.11 Serial controllers used in SPI slave mode do not tri-state MISO

Part number affected

- STM32W108C8U6x
- STM32W108HBU6x
- STM32W108CBU6x
- STM32W108CCU6x
- STM32W108CZU6x

Description

There is an issue in the serial controllers of the chip which prevents it from being used on a SPI bus that is shared with other SPI slave peripherals. The MISO (Master-In /Slave-Out) output from the chip does not tri-state when nSSEL (Slave Select) is deasserted.

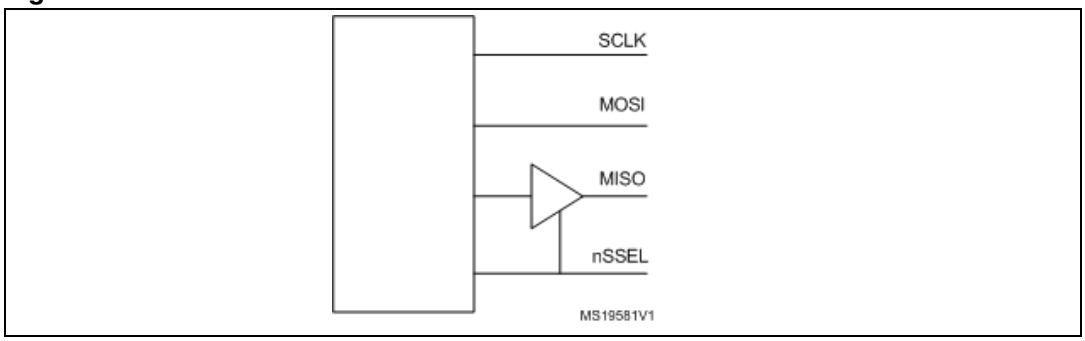
This causes contention on the shared MISO signal with other peripherals when they are selected via their own nSSEL inputs. The MISO output will continue to be driven either high or low, depending on the state of the last bit that was transmitted prior to nSSEL deasserting.

If the chip is directly connected to a SPI master with no other SPI peripherals on the bus, the bug does not cause any contention or other communications issues.

Workaround

As shown in *Figure 1*, an external buffer may be added to the MISO output with its enable connected to nSSEL. This buffer will allow the MISO signal presented to the SPI bus to be tri-stated, and will allow other SPI slave peripherals to share the same bus.

Figure 1. External MISO buffer



Revision history

Table 5. Document revision history

Date	Revision	Changes
19-Oct-2011	1	Initial release.