Errata
SWRZ045A – April 2013 – Revised - January 2014

CC2538 Errata Note

EP/WCS/LPRF/ZigBeeIoT

ABSTRACT

The CC2538 System on Chip (SoC) is described in the CC2538 Data sheet and the CC2538 User guide. This document describes issues that may be encountered when using the part according to the above mentioned documents. Where applicable, the relevant revisions of the SoC, the data sheet, or the user guide are specified.

Contents

1 Known Issues 2
1.1 ICEPICK system reset clears break points 2
1.1.1 Issue Description 2
1.1.2 Suggested Workarounds 2
1.2 GP timer edge mode (Prescaler) 3
1.2.1 Issue Description 3
1.2.2 Suggested Workarounds 3
1.3 Using emulator during power modes, EMUOVR behavior 4
1.3.1 Issue Description 4
1.3.2 Suggested Workarounds 4
1.5 Possible Incorrect Value of Clock Dividers after PM2 and PM3 5
1.5.1 Issue Description 5
1.5.2 Suggested Workarounds 5
1.6 USB controller can respond with NAK to other USB devices' packets, causing these devices to stop working 6
1.6.1 Issue Description 6
1.6.2 Suggested Workarounds 6
2 References 6

1 Known Issues

1.1 ICEPICK system reset clears break points

1.1.1 Issue Description

The ICEPICK system reset also (incorrectly) resets the following debugging modules in Cortex M3: FPB, DWT, ITM, TPIU.

This affects, amongst other things, the state of breakpoints and CPU debug mode. Programmed breakpoints are cleared, and the CPU will be in a free running mode after the ICEPICK system reset.

1.1.2 Suggested Workarounds

The following workaround is proposed for implementation in tools used for debugging of the device (for instance IAR Embedded Workbench, or Code Composer Studio).

For more information about which tools that integrates this workaround, and the specific implementation for each tool see SWRU345

The following are the steps to be performed by the debugger tool:

1. A system reset is triggered by the debugger (for example by user action, or as a reset strategy during connection or after a flash download)
2. Before doing the actual reset, read out all FPB, DWT, ITM, and TPIU registers.
3. Perform ICEPICK system reset, which will reset entire chip (including FPB, DWT, ITM, and TPIU units). The CM3 DAP is not reset, so connection with the debugger will not be lost.
4. Halt the CPU. It is recommended to implement a mechanism that avoids user code to be executed in the time elapsed between step 3 and 4, where the CPU is free running. This can for example be done with a conditional spin loop in the reset interrupt handler in software. If this method is used, the condition for the spin loop must be controllable from the debugger and changed at the end of the sequence to allow user code to be executed again.
5. Restore debug logic in CM3; FPB, DWT, ITM, and TPIU registers. (Note that FP_CTRL[1] is read-as-zero write-as-one, so simply writing back what you read from FP_CTRL will not re-enable the FPB unit)
6. Enable debug trap on core reset by writing DEMCR [0] = 1. This cause the CPU to halt at reset handler after the next core reset.
7. Perform a CPU reset (either from Icepick or NVIC registers).

1.2 GP timer edge mode (Prescaler)

1.2.1 Issue Description

When the GPTimer is configured in input time edge mode counting up with prescaler, the timer really becomes 24-bits wide since the prescaler acts as a timer extension. When the TAILR register is loaded with 0xFFFF and the prescaler value != 0, the captured time counts beyond the maximum allowed value. This results in the captured time possibly being larger than expected maximum value.

1.2.2 Suggested Workarounds

If the TAILR value is loaded with another value then 0xFFFF, the counter wraps correctly and the captured time is never larger than the maximum allowed value.

1.3 Using emulator during power modes, EMUOVR behavior

1.3.1 Issue Description

If an emulator is attached with the device in PM2, or PM3, the EMUOVR register state will be 0x00 after wakeup/restore instead of 0xFF as desired. If the device tries to enter PM1, PM2 or PM3 modes later when EMUOVR is 0x00, the emulator will lose connection.

1.3.2 Suggested Workarounds

If the emulator attaches to the CC2538 when in Sleep, PM0, PM1, PM2 or PM3 modes, the behavior of the EMUOVR register upon wakeup depends on the power mode state. If the emulator is attached with the device in Sleep, PM0, or PM1, the EMUOVR register state will be 0xFF upon wakeup as expected, assuming this was the state when the power mode was entered. If the emulator is attached with the device in PM2, or PM3, the EMUOVR register state will be 0x00. If the device tries to enter PM1, PM2 or PM3 modes later when EMUOVR is 0x00, the emulator will lose connection. To avoid losing the debugger connection on re-entering power modes, the EMUOVR register must be written to 0xFF by the debugger when it connects to the device. This workaround can be implemented in the debugger tools.. For more information about which tools implements this workaround, and how it is implemented for each tool, see SWRU345.

1.5 Possible Incorrect Value of Clock Dividers after PM2 and PM3

1.5.1 Issue Description

The value of SYS_CTRL_CLOCK_CTRL.SYS_DIV is sampled by the system controller while the restore sequence is running, i.e. before the value of SYS_DIV is properly restored. If the value of SYS_DIV is 000b, the system controller will not resample the value of SYS_DIV once the register is properly restored. As a consequence, the system controller could end up using a random value for the system clock divider.

The same issue applies to SYS_CTRL_CLOCK_CTRL.IO_DIV.

1.5.2 Suggested Workarounds

If SYS_DIV and IO_DIV are different from 000b when entering PM2 or PM3, the system controller will always use the correct divider values from SYS_DIV and IO_DIV immediately after the restore sequence is done.

The suggested workaround is to write a non-zero value to IO_DIV and SYS_DIV immediately before entering the affected power modes and restore the correct value after waking up. One way of implementing the workaround is illustrated in the pseudo code below:

```
If (IO_DIV == 0)  { reset_io_div_to_zero = true;  IO_DIV = 1; }

If (SYS_DIV == 0) { reset_sys_div_to_zero = true; SYS_DIV = 1; }

wfi();

if (reset_sys_div_to_zero == true) { SYS_DIV = 0; reset_sys_div_to_zero = false; }

if (reset_sys_io_to_zero == true)  { IO_DIV = 0;  reset_io_div_to_zero = false; }
```

Note that the alternative setting is 1, which will give the same system clock rate as for a system running at 16 MHz. Also note that the system will use the internal 16 MHz high speed RC oscillator as clock source until the external 32 MHz clock source is active and stable after waking up from PM2 or PM3. As a consequence, the SYS_DIV and IO_DIV setting of 1 will not affect the timing of the system as long as they are set back to 0 before the 32 MHz external clock source is stable.

1.6 USB controller can respond with NAK to other USB devices' packets, causing these devices to stop working

1.6.1 Issue Description

The CC2538 USB controller will sometimes interrupt communication between other USB devices on the bus by responding with NAK to packets not destined for itself.

The bug can be triggered if the USB host interleaves the following:

- IN transactions (from device to host) of the data stage of a CC2538 endpoint 0 control transfer.
- Any full-speed transaction containing a DATA1 packet to another USB device.

More specifically, the bug occurs if a DATA1 packet (to another USB device) is received by CC2538 before it has loaded the next IN data packet.

1.6.2 Suggested Workarounds

This problem is rare, and will only occur when the CC2538 shares a USB root hub with other USB devices. It has only been observed when using USB 3.0 host controllers, and only when the other USB device uses outgoing data transfers actively (e.g. file transfer by a Bluetooth USB dongle).

There are two workarounds that will reduce, but not eliminate, the occurrence of this issue:

- Avoid using endpoint 0 control transfers to operate the CC2538 USB device.
- Load the IN data packets for control transfers as quickly as possible.

2 References

1. *CC2538 Data sheet (SWRS096)*
2. *CC2538 IDE User’s guide (SWRU345)*

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265