

MSP430C091 Device Erratasheet

1 Functional Errata Revision History

Errata impacting device's operation, function or parametrics.

✓ The check mark indicates that the issue is present in the specified revision.

| Errata Number | Rev E | Rev D |
|------------------------|-------|-------|
| APOOL8 | ✓ | ✓ |
| CPU46 | ✓ | ✓ |
| EEM18 | ✓ | ✓ |
| PORT19 | ✓ | ✓ |

2 Preprogrammed Software Errata Revision History

Errata impacting pre-programmed software into the silicon by Texas Instruments.

✓ The check mark indicates that the issue is present in the specified revision.

The device doesn't have Software in ROM errata.

3 Debug only Errata Revision History

Errata only impacting debug operation.

✓ The check mark indicates that the issue is present in the specified revision.

| Errata Number | Rev E | Rev D |
|-----------------------|-------|-------|
| EEM23 | ✓ | ✓ |

4 Fixed by Compiler Errata Revision History

Errata completely resolved by compiler workaround. Refer to specific erratum for IDE and compiler versions with workaround.

✓ The check mark indicates that the issue is present in the specified revision.

| Errata Number | Rev E | Rev D |
|-----------------------|-------|-------|
| CPU21 | ✓ | ✓ |
| CPU22 | ✓ | ✓ |

Refer to the following MSP430 compiler documentation for more details about the CPU bugs workarounds.

TI MSP430 Compiler Tools (Code Composer Studio IDE)

- [MSP430 Optimizing C/C++ Compiler](#): Check the `--silicon_errata` option

- [MSP430 Assembly Language Tools](#)

MSP430 GNU Compiler (MSP430-GCC)

- [MSP430 GCC Options](#): Check -msilicon-errata= and -msilicon-errata-warn= options
- [MSP430 GCC User's Guide](#)

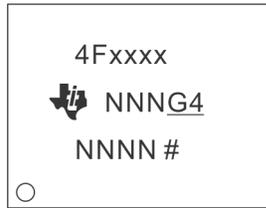
IAR Embedded Workbench

- [IAR workarounds for msp430 hardware issues](#)

5 Package Markings

PW14

TSSOP (PW), 14 Pin



= Die revision
○ = Pin 1 location
N = Lot trace code

6 Detailed Bug Description

APOOL8 *APOOL Module*

Category Functional

Function APOOL Comparator output edge may not be detected

Description If the APOOL uses the Digital Filtering feature, after each reconfiguration of the the APCTL and APCNF registers, the Comparator output edge will not be correctly detected.

Workaround

1. After any new configuration of APCTL or APCNF register, always execute a 'dummy write' instruction to APCTL (or low byte APCNF) which triggers a 'Reset of Deglitch filter' event. (e.g. 'bis' to read-only LCMP bit: bis.w #0010h,&APCNF)
2. Disable the Digital Filter by resetting DFSETx bits.

For detailed workaround guidance, refer to [MSP430x09x Analog Pool: Feature Set and Advanced Use](#).

CPU21 *CPUXv2 Module*

Category Compiler-Fixed

Function Using POPM instruction on Status register may result in device hang up

Description When an active interrupt service request is pending and the POPM instruction is used to set the Status Register (SR) and initiate entry into a low power mode , the device may hang up.

Workaround None. It is recommended not to use POPM instruction on the Status Register. Refer to the table below for compiler-specific fix implementation information.

| IDE/Compiler | Version Number | Notes |
|---|-----------------------------------|--|
| IAR Embedded Workbench | Not affected | |
| TI MSP430 Compiler Tools (Code Composer Studio) | v4.0.x or later | User is required to add the compiler or assembler flag option below. --silicon_errata=CPU21 |
| MSP430 GNU Compiler (MSP430-GCC) | MSP430-GCC 4.9 build 167 or later | |

CPU22 *CPUXv2 Module*

Category Compiler-Fixed

Function Indirect addressing mode with the Program Counter as the source register may produce unexpected results

Description When using the indirect addressing mode in an instruction with the Program Counter (PC) as the source operand, the instruction that follows immediately does not get executed.

For example in the code below, the ADD instruction does not get executed.

```
mov @PC, R7
add #1h, R4
```

Workaround Refer to the table below for compiler-specific fix implementation information.

| IDE/Compiler | Version Number | Notes |
|---|-----------------------------------|--|
| IAR Embedded Workbench | Not affected | |
| TI MSP430 Compiler Tools (Code Composer Studio) | v4.0.x or later | User is required to add the compiler or assembler flag option below. --silicon_errata=CPU22 |
| MSP430 GNU Compiler (MSP430-GCC) | MSP430-GCC 4.9 build 167 or later | |

CPU46

CPUXv2 Module

Category

Functional

Function

POPM performs unexpected memory access and can cause VMAIFG to be set

Description

When the POPM assembly instruction is executed, the last Stack Pointer increment is followed by an unintended read access to the memory. If this read access is performed on vacant memory, the VMAIFG will be set and can trigger the corresponding interrupt (SFRIE1.VMAIE) if it is enabled. This issue occurs if the POPM assembly instruction is performed up to the top of the STACK.

Workaround

If the user is utilizing C, they will not be impacted by this issue. All TI/IAR/GCC pre-built libraries are not impacted by this bug. To ensure that POPM is never executed up to the memory border of the STACK when using assembly it is recommended to either

1. Initialize the SP to
 - a. TOP of STACK - 4 bytes if POPM.A is used
 - b. TOP of STACK - 2 bytes if POPM.W is used

OR

2. Use the POPM instruction for all but the last restore operation. For the the last restore operation use the POP assembly instruction instead.

For instance, instead of using:

```
POPM.W #5,R13
```

Use:

```
POPM.W #4,R12
```

```
POP.W R13
```

Refer to the table below for compiler-specific fix implementation information.

| IDE/Compiler | Version Number | Notes |
|---|----------------|--|
| IAR Embedded Workbench | Not affected | C code is not impacted by this bug. User using POPM instruction in assembler is required to implement the above workaround manually. |
| TI MSP430 Compiler Tools (Code Composer Studio) | Not affected | C code is not impacted by this bug. User using POPM instruction in assembler is required to implement the above workaround manually. |
| MSP430 GNU Compiler (MSP430-GCC) | Not affected | C code is not impacted by this bug. User using POPM instruction in assembler is required to implement the above workaround manually. |

| | |
|--------------------|--|
| EEM18 | <i>EEM Module</i> |
| Category | Functional |
| Function | RST/NMI pin becomes nonfunctional after download to external memory |
| Description | After the procedure to download to external memory, the RST/NMI pin cannot be used to restart the device and download the code from the external memory into RAM. |
| Workaround | Power cycle the device to re-enable the RST/NMI functionality after the download to external memory. |
| EEM23 | <i>EEM Module</i> |
| Category | Debug |
| Function | EEM triggers incorrectly when modules using wait states are enabled |
| Description | When modules using wait states (USB, MPY, CRC and FRAM controller in manual mode) are enabled, the EEM may trigger incorrectly. This can lead to an incorrect profile counter value or cause issues with the EEMs data watch point, state storage, and breakpoint functionality. |
| Workaround | None. |
| | <hr/> NOTE: This erratum affects debug mode only. <hr/> |
| PORT19 | <i>PORT Module</i> |
| Category | Functional |
| Function | Port interrupt may be missed on entry to LPMx.5 |
| Description | If a port interrupt occurs within a small timing window (~1MCLK cycle) of the device entry into LPM3.5 or LPM4.5, it is possible that the interrupt is lost. Hence this interrupt will not trigger a wakeup from LPMx.5. |
| Workaround | None |

7 Document Revision History

Changes from family erratasheet to device specific erratasheet.

Initial release

Changes from device specific erratasheet to document Revision A.

1. Errata PORT19 was added to the errata documentation.

Changes from document Revision A to Revision B.

1. Errata EEM23 was added to the errata documentation.
2. Errata CPU43 was added to the errata documentation.

Changes from document Revision B to Revision C.

1. CPU43 Description was updated.

Changes from document Revision C to Revision D.

1. CPU43 Description was updated.
2. EEM23 Workaround was updated.
3. EEM23 Description was updated.

Changes from document Revision D to Revision E.

1. Package Markings section was updated.
2. EEM23 Workaround was updated.
3. EEM23 Description was updated.
4. EEM23 Function was updated.

Changes from document Revision E to Revision F.

1. Errata CPU43 was removed from the errata documentation.

Changes from document Revision F to Revision G.

1. EEM23 Description was updated.

Changes from document Revision G to Revision H.

1. Errata JTAG27 was added to the errata documentation.

Changes from document Revision H to Revision I.

1. Errata CPU46 was added to the errata documentation.

Changes from document Revision I to Revision J.

1. CPU21 was added to the errata documentation.
2. CPU22 was added to the errata documentation.
3. Workaround for CPU46 was updated.

Changes from document Revision J to Revision K.

1. Workaround for CPU46 was updated.

Changes from document Revision K to Revision L.

1. Erratasheet format update.
2. Added errata category field to "Detailed bug description" section

Changes from document Revision L to Revision M.

1. CPU47 was added to the errata documentation.

Changes from document Revision M to Revision N.

1. JTAG27 was removed from the errata documentation.
2. CPU47 was removed from the errata documentation.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2022, Texas Instruments Incorporated