MSP430F149 Device Erratasheet

1 Revision History

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

NOTE: Silicon Revisions AA, AB, and AD use BSL version 1.61. For specific information on this version of the BSL and its proper usage, see the MSP430 Memory Programming User's Guide (SLAU265).

Errata Number	Rev AE	Rev AD	Rev AB	Rev AA	Rev S	Rev Q	Rev O	Rev N	Rev M	Rev L
ADC1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ADC5	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ADC7	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ADC8	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ADC9	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ADC10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ADC11										✓
ADC18	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ADC25	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BCL5	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BSL3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BSL4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BSL5	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CPU4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EEM20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MPY2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PORT3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
RES3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
RES4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TA12	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TA16	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TA21	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TAB22	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TB1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TB2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TB3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TB4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TB14	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TB16	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TB24	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
US13	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
US14	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
US15	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Errata Number	Rev AE	Rev AD	Rev AB	Rev AA	Rev S	Rev Q	Rev O	Rev N	Rev M	Rev L
WDG2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

2 Package Markings

PAG64

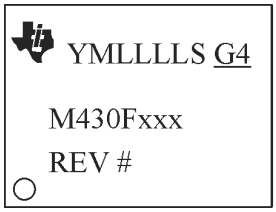
TQFP (PAG), 64 Pin

YM = Year and Month Date Code
LLLL = LOT Trace Code
S = Assembly Site Code
# = DIE Revision
o = PIN 1

YM = Year and Month Date Code
LLLL = LOT Trace Code
S = Assembly Site Code
# = DIE Revision
o = PIN 1

PM64

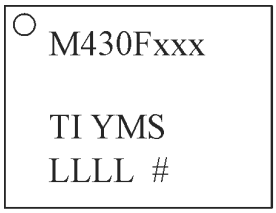
LQFP (PM), 64 Pin



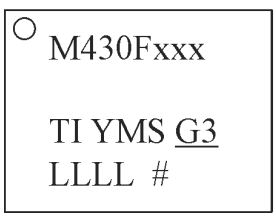
YM = Year and Month Date Code
LLLL = LOT Trace Code
S = Assembly Site Code
# = DIE Revision
○ = Pin 1

RTD64

QFN (RTD), 64 Pin



TI = TI
YM = Year and Month Date Code
LLLL = LOT Trace Code
S = Assembly Site Code
# = DIE Revision
○ = Pin 1



TI = TI
YM = Year and Month Date Code
LLLL = LOT Trace Code
S = Assembly Site Code
# = DIE Revision
○ = Pin 1

3 Detailed Bug Description

ADC1 *ADC12 Module*

Function Start of conversion

Description In single conversion/sequence mode (CONSEQ=0/1), the next conversion can be started with ADC12SC. It is not necessary to clear ENC before setting ADC12SC. This is contrary to the specification.

Workaround None

ADC5 *ADC12 Module*

Function Interrupt flag register

Description ADC12 interrupt flag may not be set when the CPU simultaneously accesses the ADC12IFG register.

Workaround There is no need to access the interrupt flag register to process interrupt situations. Please use the ADC12IV register to identify the interrupt event. The corresponding flag bits will be reset automatically. Additional details are discussed in the device family user's guide.

ADC7 *ADC12 Module*

Function Conversion time overflow

Description The timing overflow flag is set when the device is in sequence mode (CONSEQ = 1 or 3) and MSC = 0, even if no overflow has occurred.

Workaround Verify correct timing and do not enable Conversion-Time Overflow interrupt.

ADC8 *ADC12 Module*

Function Interrupt flag register

Description Clearing flags in the interrupt flag register with a CPU instruction will not clear the latest interrupt flag.

Workaround Clear interrupt flags by accessing the conversion-memory registers.

ADC9 *ADC12 Module*

Function Interrupt vector register

Description If the ADC12 uses a different clock than the CPU (MCLK) and more than one ADC interrupt is enabled, the ADC12IV register content may be unpredictable for one clock cycle. This happens if, during the execution of an ADC interrupt, another ADC interrupt with higher priority occurs.

Workaround - Read out ADC12IV twice and use only when values are equal.

or

- Use ADC12IFG to determine which interrupt has occurred.

ADC10 *ADC12 Module*

Function Unintended start of conversion

Description Accessing ADC12OVIE or ADC12TOVIE at the end of an ADC12 conversion with BIS/BIC commands can cause the ADC12SC bit to be set again immediately after it was cleared. This might start another conversion, if ADC12SC is configured to trigger the ADC (SHS = 0).

Workaround If ADC12SC is configured to trigger the ADC, the control bits ADC12OVIE and ADC12TOVIE should be modified only when the ADC is not busy (ADC12BUSY = 0).

ADC11 *ADC12 Module*

Function Temporary leakage current after conversion

Description The ADC12 causes temporary leakage current after a completed conversion. Duration and magnitude of the leakage current depends on parasitic effects.

Workaround None

ADC18 *ADC12 Module*

Function Incorrect conversion result in extended sample mode

Description The ADC12 conversion result can be incorrect if the extended sample mode is selected (SHP = 0), the conversion clock is not the internal ADC12 oscillator (ADC12SSEL > 0), and one of the following two conditions is true:

- The extended sample input signal SHI is asynchronous to the clock source used for ADC12CLK and the undivided ADC12 input clock frequency exceeds 3.15 MHz.

or

- The extended sample input signal SHI is synchronous to the clock source used for ADC12CLK and the undivided ADC12 input clock frequency exceeds 6.3 MHz.

Workaround

- Use the pulse sample mode (SHP = 1).

or

- Use the ADC12 internal oscillator as the ADC12 clock source.

or

- Limit the undivided ADC12 input clock frequency to 3.15 MHz.

or

- Use the same clock source (such as ACLK or SMCLK) to derive both SHI and ADC12CLK, to achieve synchronous operation, and also limit the undivided ADC12 input clock frequency to 6.3 MHz.

ADC25 *ADC12 Module*

Function Write to ADC12CTL0 triggers ADC12 when CONSEQ = 00

Description If ADC conversions are triggered by the Timer_B module and the ADC12 is in single-channel single-conversion mode (CONSEQ = 00), ADC sampling is enabled by write access to any bit(s) in the ADC12CTL0 register. This is contrary to the expected behavior that only the ADC12 enable conversion bit (ADC12ENC) triggers a new ADC12 sample.

Workaround When operating the ADC12 in CONSEQ=00 and a Timer_B output is selected as the sample and hold source, temporarily clear the ADC12ENC bit before writing to other bits in the ADC12CTL0 register. The following capture trigger can then be re-enabled by setting ADC12ENC = 1.

BCL5 *BCS Module*

Function RSELx bit modifications can generate high frequency spikes on MCLK

Description When DIVMx = 00 or 01 the RSELx bits of the Basic Clock Module are incremented or decremented in steps of 2 or greater, the DCO output may momentarily generate high frequency spikes on MCLK, which may corrupt CPU operation. This is not an issue when DIVMx = 10 or 11.

Workaround Set DIVMx = 10 or 11 to divide the MCLK input prior to modifying RSELx. After the RSELx bits are configured as desired, the DIVMx setting can be changed back to the original selection.

BSL3 *BSL Module*

Function Receiving frames

Description Receiving frames with a checksum value equal to a legal address can change the content of this address or the bootstrap loader may stop operation.

Workaround Software workaround is available.

BSL4 *BSL Module*

Function Flash memory can not be programmed

Description The bootstrap loader software cannot program the flash memory.

Workaround Software workaround is available.

BSL5 *BSL Module*

Function BSL might not start if RST/NMI pin is configured as NMI input

Description If the RST/NMI pin is configured to NMI, the bootstrap loader may not be started. Unpredictable operations will result.

Workaround None

CPU4 *CPU Module*

Function PUSH #4, PUSH #8

Description The single operand instruction PUSH cannot use the internal constants (CG) 4 and 8. The other internal constants (0, 1, 2, -1) can be used. The number of clock cycles is different:

PUSH #CG uses address mode 00, requiring 3 cycles, 1 word instruction

PUSH #4/#8 uses address mode 11, requiring 5 cycles, 2 word instruction

Workaround Workaround implemented in assembler.

EEM20 *EEM Module*

Function Debugger might clear interrupt flags

Description During debugging read-sensitive interrupt flags might be cleared as soon as the debugger stops. This is valid in both single-stepping and free run modes.

Workaround None.

MPY2 *MPY Module*

Function Multiplier Result register corruption

Description Depending on the address of the write instruction, writing to the multiplier result registers (RESHI, RESLO, or SUMEXT) may corrupt the result registers. The address dependency varies between a 2-word and a 3-word instructions.

Workaround Ensure that a write instruction to an MPY result register (for example, mov.w #200, &RESHI) is not located at an address with the four least significant bits shown in Table 1:

Table 1. Sensitive Addresses for Write Access to MPY Result Registers MAB[3:0]

RESLOW 013Ah		RESHI 013Ch		SUMEXT 013Eh	
3 Word	2 Word	3 Word	2 Word	3 Word	2 Word
2	4	2	4	2	4
6	8	4	6	6	8
A	C	A	C	A	C
E	0	C	E	–	–

PORT3 *PORT Module*

Function Port interrupts can get lost

Description Port interrupts can get lost if they occur during CPU

access of the P1IFG and P2IFG registers.

Workaround None

RES3 *RESET Module*

Function Reset

Description When RST/NMI is held low during power up of VCC, some internal drivers are not reset correctly. This may result in a high Icc current until the internal power-on signal has generated one clock cycle to reset the internal drivers. This limits the time when the excess current can occur to the time the power-up circuit is active.

Workaround None

RES4 *RESET Module*

Function No reset if external resistor exceeds certain value

Description No reset of the device is performed if the external pull down resistor on RST/NMI pin is above a certain limit. The limits are:

Vcc = 1.8V: maximum pull down resistor = 12 kohm

Vcc = 3.0V: maximum pull down resistor = 5 kohm

Vcc = 3.6V: maximum pull down resistor = 2.5 kohm

In addition, a higher current consumption occurs during high/low RST/NMI signal transition when using improper resistors.

Workaround Use external pulldown resistors below the listed values or directly drive RST/NMI low to generate a reset.

TA12 *TIMER_A Module*

Function Interrupt is lost (slow ACLK)

Description Timer_A counter is running with slow clock (external TACLK or ACLK)compared to MCLK. The compare mode is selected for the capture/compare channel and the CCRx register is incremented by one with the occurring compare interrupt (if TAR = CCRx). Due to the fast MCLK the CCRx register increment (CCRx = CCRx+1) happens before the Timer_A counter has incremented again. Therefore the next compare interrupt should happen at once with the next Timer_A counter increment (if TAR = CCRx + 1). This interrupt gets lost.

Workaround Switch capture/compare mode to capture mode before the CCRx register increment. Switch back to compare mode afterwards.

TA16 *TIMER_A Module*

Function First increment of TAR erroneous when IDx > 00

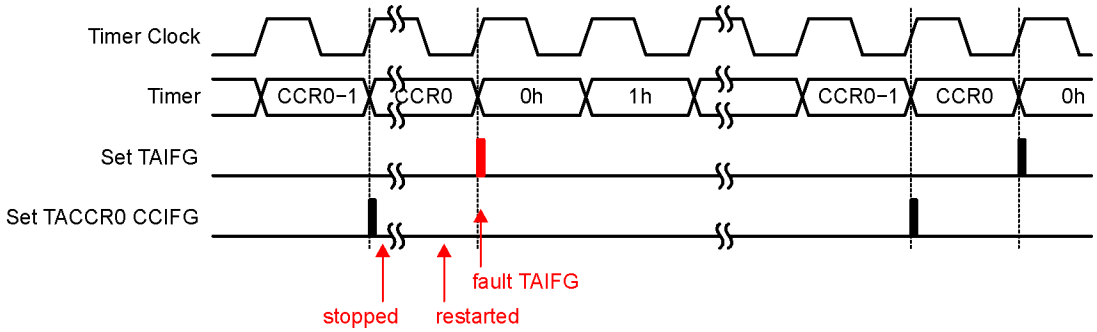
Description The first increment of TAR after any timer clear event (POR/TACLR) happens immediately following the first positive edge of the selected clock source (INCLK, SMCLK, ACLK or TACLK). This is independent of the clock input divider settings (ID0, ID1). All following TAR increments are performed correctly with the selected IDx settings.

Workaround None

TA21 *TIMER_A Module*

Function TAIFG Flag is erroneously set after Timer A restarts in Up Mode

Description In Up Mode, the TAIFG flag should only be set when the timer resets from TACCR0 to zero. However, if the Timer A is stopped at TAR = TACCR0, then cleared (TAR=0) by setting the TACLR bit, and finally restarted in Up Mode, the next rising edge of the TACLK will erroneously set the TAIFG flag.



Workaround None.

TAB22 *TIMER_A/TIMER_B Module*

Function Timer_A/Timer_B register modification after Watchdog Timer PUC

Description Unwanted modification of the Timer_A/Timer_B registers TACTL/TBCTL and TAIV/TBIV can occur when a PUC is generated by the Watchdog Timer(WDT) in Watchdog mode and any Timer_A/Timer_B counter register TACCRx/TBCCRx is incremented/decremented (Timer_A/Timer_B does not need to be running).

Workaround Initialize TACTL/TBCTL register after the reset occurs using a MOV instruction (BIS/BIC may not fully initialize the register). TAIV/TBIV is automatically cleared following this initialization.

Example code:

MOV.W #VAL, &TACTL

or

MOV.W #VAL, &TBCTL

Where, VAL=0, if Timer is not used in application otherwise, user defined per desired function.

TB1 *TIMER_B Module*

Function "Equal mode" when grouping compare latches

Description The "equal mode" for loading the compare latches (CLLD = 3) cannot be used when compare latches are grouped (TBCLGRP > 0).

Workaround None

TB2 *TIMER_B Module*

Function Interrupt is lost (slow ACLK)

Description Timer_B counter is running with slow clock (external TBCLK or ACLK) compared to MCLK. The compare mode is selected for the capture/compare channel and the CCRx register is incremented by 1 with the occurring compare interrupt (if TBR = CCRx).

Due to the fast MCLK, the CCRx register increment (CCRx = CCRx + 1) happens before the Timer_B counter has incremented again. Therefore, the next compare interrupt should happen at once with the next Timer_B counter increment (if TBR = CCRx + 1). This interrupt is lost.

Workaround Switch capture/compare mode to capture mode before the CCRx register increment. Switch back to compare mode afterward.

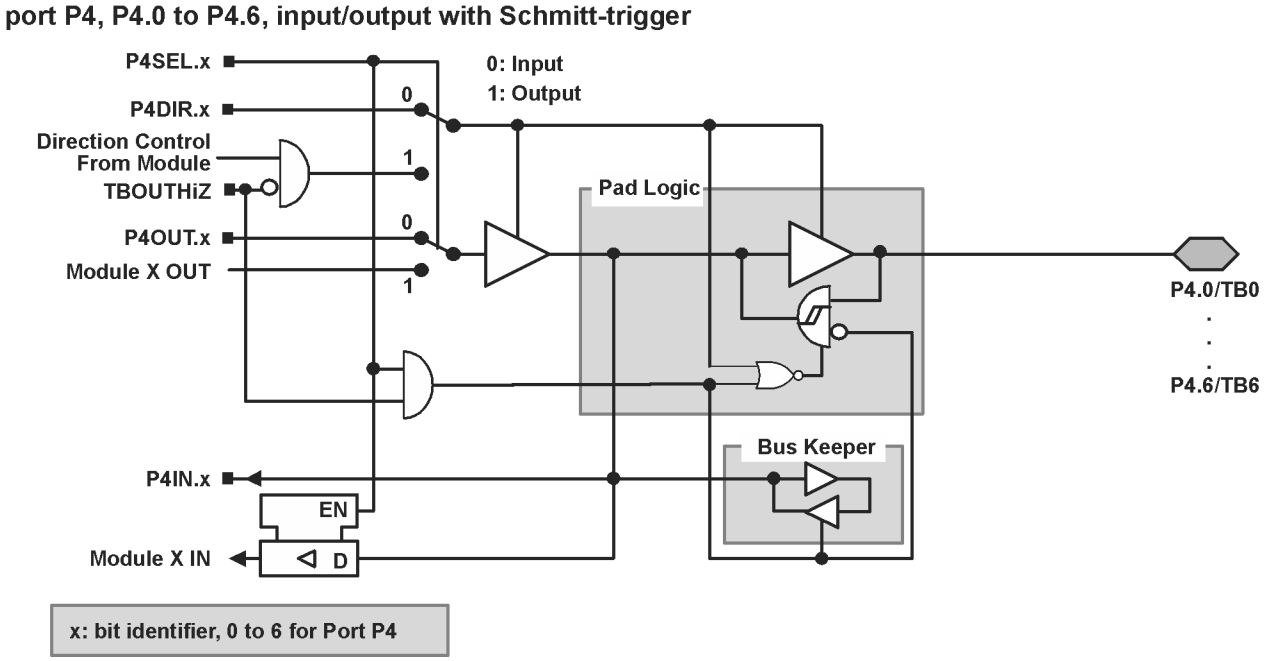
TB3 *TIMER_B Module*

Function Port is switched to 3-state independent of selected function

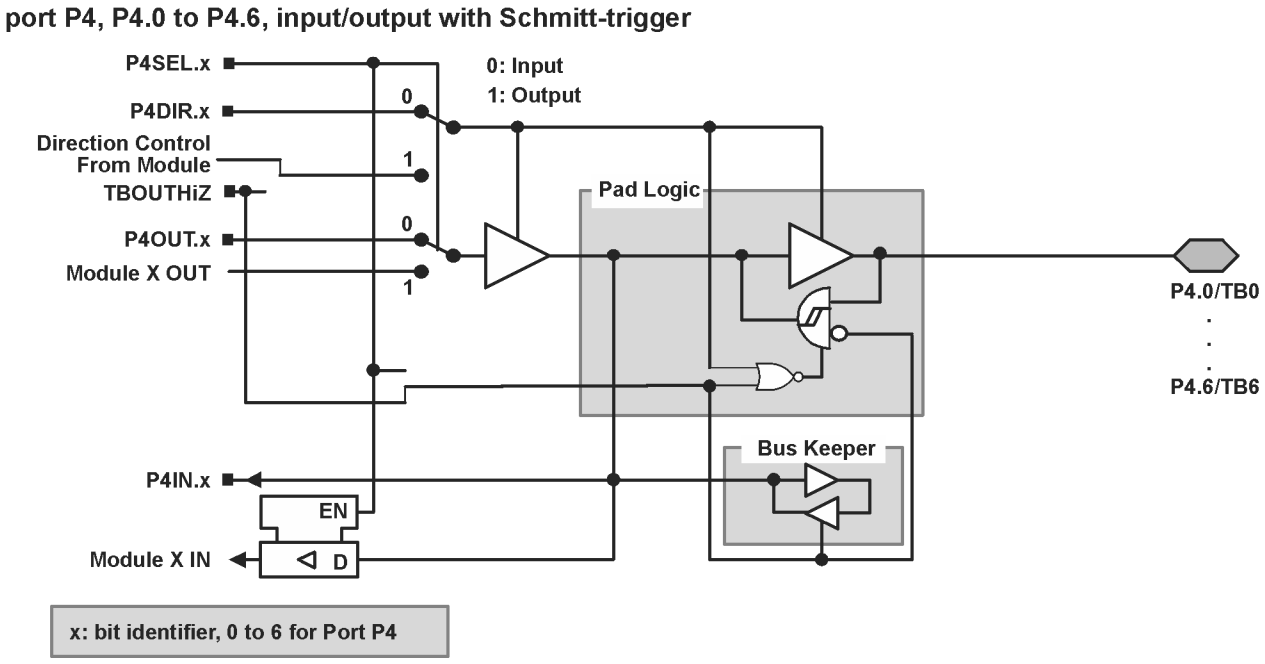
Description Incorrect 3-state function of Ports P4.0/TB0 through P4.6/TB6 (TBoutHiZ control). If TBoutHiZ is set to high, all ports P4.0/TB0 through P4.6/TB6 are set to 3-state, independent of the P4SEL.x control signals. This means a port P4.x is switched to 3-state with TBoutHiZ, even if it is not selected for Timer_B function. In addition, the ports P4.0/TB0 through P4.6/TB6 are switched to 3-state with TBoutHiZ, even if the port direction (direction control from module) is set to input. This is in accordance with the specification description but, nevertheless, is an unexpected behavior.

Workaround No workaround.

Port function as specified



Port Realization With TB3 Bug



TB4 *TIMER_B Module*

Function Group function

Description If the shadow registers are organized in groups (SHR = 1, 2, or 3), one shadow register is not loaded correctly. This happens when the last CCRx register within a group is loaded at exactly the same time that the timer counter reaches the event for loading the shadow registers (TBR = 0 or TBR = CCR0).

Workaround Ensure that all CCRx registers within a group are loaded before the shadow register load event occurs.

TB14 *TIMER_B Module*

Function PWM output

Description The PWM output unit may behave erroneously if the condition for changing the PWM output (EQUx or EQU0) and the condition for loading the shadow register TBCLx happen at the same time. Depending on the load condition for the shadow registers (CLLD bits in TBCCTLx), there are four possible error conditions:

1. Change CCRx register from any value to CCRx = 0 (for example, sequence for CCRx = 4 3 2 0 0 0)

2. Change CCRx register from CCRx = 0 to any value (for example, sequence for CCRx = 0 0 0 2 3 4)

3. Change CCRx register from any value to current SHD0 (CCR0) value (for example, sequence for CCRx = 4 2 5 SHD0 3 8)

4. Change CCRx register from current SHD0 (CCR0) value to any value (for example, sequence for CCRx = 4 2 SHD0 5 3 8)

Workaround No general workaround available.

TB16 *TIMER_B Module*

Function First increment of TBR erroneous when IDx > 00

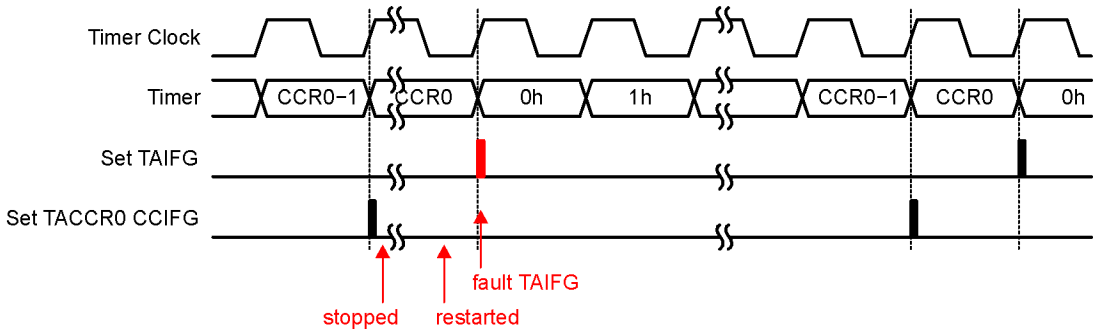
Description The first increment of TBR after any timer clear event (POR/TBCLR) happens immediately following the first positive edge of the selected clock source (INCLK, SMCLK, ACLK, or TBCLK). This is independent of the clock input divider settings (ID0, ID1). All following TBR increments are performed correctly with the selected IDx settings.

Workaround None

TB24 *TIMER_B Module*

Function TBIFG Flag is erroneously set after Timer B restarts in Up Mode

Description In Up Mode, the TBIFG flag should only be set when the timer resets from TBCCR0 to zero. However, if the Timer A is stopped at TBR = TBCCR0, then cleared (TBR=0) by setting the TBCLR bit, and finally restarted in Up Mode, the next rising edge of the TBCLK will erroneously set the TBIFG flag.



Workaround None.

US13 *USART Module*

Function Unpredictable program execution

Description USART interrupts requested by URXS can result in unpredictable program execution if this request is not served within two bit times of the received data.

Workaround Ensure that the interrupt service routine is entered within two bit times of the received data.

US14 *USART Module*

Function Start edge of received characters may be ignored

Description When using the USART in UART mode with UxBR0 = 0x03 and UxBR1 = 0x00, the start edge of received characters may be ignored due to internal timing conflicts within the UART state machine. This condition does not apply when UxBR0 is > 0x03.

Workaround None

US15 *USART Module*

Function UART receive with two stop bits

Description USART hardware does not detect a missing second stop bit when SPB = 1.

The Framing Error Flag (FE) will not be set under this condition and erroneous data reception may occur.

Workaround None (Configure USART for a single stop bit, SPB = 0)

WDG2 *WDT Module*

Function Incorrectly accessing a flash control register

Description If a key violation is caused by incorrectly accessing a flash control register, the watchdog interrupt flag is set in addition to the expected PUC.

Workaround None

4 Document Revision History

Changes from family erratasheet to device specific erratasheet.

1. Errata MPY2 was added

Changes from device specific erratasheet to document Revision A.

1. Errata EEM20 was added to the errata documentation.

Changes from document Revision A to Revision B.

1. Errata TA21 was added to the errata documentation.

Changes from document Revision B to Revision C.

1. Errata TB24 was added to the errata documentation.

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