TMS470R1VF288

TMS470 Microcontroller

Silicon Errata

Silicon Revision A

SPNZ143B
February 2006

REVISION HISTORY

This silicon errata revision history highlights the technical changes made to the SPNZ143A revision to make it an SPNZ143B revision.

PAGE(S) NO.	ADDITIONS/CHANGES/DELETIONS
5	Erratum ADM#11 added
6	Errata ADM #12, ADM#13, and ADM#14 added
7	Erratum DEV#42 added
11	Erratum RTI#7 added

Contents

1 Known Design Marginality/Exceptions to Functional Specifications

The following is a list of advisories on modules in this version of silicon, errata matrix version 8.92. Documentation may differ from the user guide or data sheet. The advisory reference number is shown first (i.e.; ADM#8), followed by a description and any known workarounds. The reference numbers may not always be sequential for this device.

Modules include the following:

Multi-buffered analog-to-digital converter (ADM)

Direct memory access controller (DMA)

Device-specific (DEV)

Flash wrapper (FW)

Frequency-Modulated Zero-Pin Phase-Locked-Loop(FM_ZPLL)

General Purpose Input/Output (GIO)

High-end timer (HET)

Memory Security Module (MSM)

Real-time interrupt (RTI)

Standard CAN controller (SCC)

Serial communication interface (SCI)

Serial peripheral interface (SPI)

Advisory ADM#8	*Stopping and Starting ADC When Conversions Are Ongoing*

Description: When used in FIFO mode, if the A to D module is diabled or if the channel select registers are cleared while conversions are still ongoing, the operation will be unpredictable when the module is restarted.

Workaround: Stop the ADC by clearing ADCR1(5).
Restart the ADC by setting ADCR1(5) again.
Configure all groups to be in single conversion mode.
Configure one channel to be converted in all three groups and start the conversions.
Wait for these conversions to end by polling the ”conversion end” flags in the ADSR register.
Clear the channel select registers for the three groups.
Continue with the desired configuration for the ADC.

Advisory ADM#9 — *Freeze Feature Error for Conversion Groups*

Description: When multiple conversion groups are being used and the ADC is used in the multi-buffered mode, the use of the freeze feature for conversion groups can lead to conversion results being written to the wrong FIFO. If a conversion group (say group A) is configured to be ”freezable”, and if there is a request for servicing another conversion group (say group B) while group A conversion is still ongoing, then the conversion result for the last channel converted in group A will be written to the FIFO for group B.

Workaround: Do not use the freeze ability for the conversion groups.
OR
For applications that must use the freeze ability, please use only the compatibility mode of the ADC.

Advisory ADM#10 — *Data Not Written to MibADC FIFO*

Description: In buffered mode, if the channel select register is written and if no other MibADC registers are accessed, the converted data does not get written to the FIFO but the threshold counter is updated upon each conversion. This occurs only when the ICLK/SYSCLK ratio is more than 2.

Workaround: Do a read operation from the same group input channel select register after writing it.

Advisory ADM#11 — *Disabling EN Bit Does Not Properly Stop and Restart the ADC*

Description: Disabling the ADC by clearing the ADC EN bit of the ADCR1 register does not reset some internal flags used to store the conversion statuses of the three groups, causing problems when stopping and restarting the ADC conversions in buffered mode.

Workaround: TBD

Advisory ADM#12 — *Clearing Channel Selection Register Does Not Clear FIFO Completely*

Description: Clearing the channel selection register does not clear FIFO completely under certain conditions.

Workaround: Depending upon the number of selected Channels, the over−head on ”time” is in the increasing order with each work−around. Depending upon the requirement and criticality, any of them can be chosen.

1. Irrespective of whether a group is in continuous or single conversion mode, read−out the corresponding group’s FIFO data until it’s empty, write all ’0’s to the Group Channel Select Register. After this, wait for the duration of one Channel Conversion completion including the sampling and the conversion time, before writing the next set of channels to the Channel Select Register.

2. Alternatively, if the group conversion is in continuous mode, and an application wants to change the Group Channel Select Register, then first change the mode of the group to single conversion. Read−out all the conversion data from the FIFO, until the FIFO becomes empty. Write a single channel into the Channel Select Register. Wait until the Group Conversion End flag gets set.Write the next set of Channels to the Group Channel Select Register.

3. If the group is in Single conversion mode, wait until the Group Conversion End flag is set, read out the FIFO data until it’s empty and then write the new set of Channels to the Group Channel Select Register.

Advisory ADM#13 — *Current Channel Conversion Does Not Complete When Suspend Mode is Entered*

Description: Ongoing conversion is not completed on entering suspend mode when COS = 1.

Workaround: TBD

Advisory ADM#14 — *Calibration Results Are Written to Group FIFO*

Description: If calibration is started during a group conversion, the result of the calibration is written to that group FIFO until the end of group conversion.

Workaround: Do not do calibration during a group conversion. The documentation will be updated to reflect this requirement. (SPNU193, 9/2002)

Advisory DEV#42	*5V Tolerant Pins Do Not Have Internal Clamp Diode to Positive Voltage*

Description: Pins with 100µA internal pull-ups should be specified with an input current of between −200µA to −100µA.

Workaround: The documentation will be updated to reflect this requirement. (SPNU193, 9/2002)

Advisory DMA#4	*BMSS=1 Mode Not Supported*

Description: DMA transfers in BMSS=1 mode will be corrupted due to a bug in the DMA state machine.

Workaround: BMSS=1 mode is no longer supported. Use BMSS=0. The documentation will be updated.(SPNU194, 11/2002)

Advisory DMA#17	*Reads of MPU Registers Corrupt Data*

Description: If the ARM7 CPU is reading memory protection unit register while the DMA is operating, the data read or written by the DMA can be corrupted.

Workaround: Avoid any reads of the MPU registers while the DMA is operating. The CPU reads of the MPU registers while the DMA is operating are not supported. The documentation will be updated. (SPNU194, 11/2002)

Advisory DMA#20	*No Exception for DMA Access to Unmapped Memory on Expansion Bus*

Description: No reset or abort occurs when the source or destination address of the DMA is an unmapped memory area on the expansion bus.

Workaround: None. The documentation will be updated to clarify that memory bounds checking is not supported on DMA accesses to the expansion bus. (SPNU194, 11/2002)

Advisory DMA#21	*One Transfer with Zero Transfer Count*

Description: If a control packet is set up and enabled with a transfer count of zero, one DMA transfer occurs.

Workaround: Do not enable a DMA control packet with a transfer count of zero.

Advisory DMA#23 — *DMA Stop Corrupts Command Buffer Memory*

Description: Using DMA Stop may corrupt the DMA command buffer memory.

Workaround: Use DMA Halt, not DMA Stop.

Advisory DMA#24 — *DMA Writes to Read-Only Memory Do Not Generate an Illegal Address*

Description: When a particular region of memory is set as read-only by the address decoder or the MPU, any write to that memory region should generate an illegal access. This works properly in the case of CPU writes, but DMA writes do not cause an illegal access. In both cases, writes to the RAM are blocked by blocking the chip selects.

Workaround: None

Advisory DMA#25 — *DMA Channel Switch Size Not Properly Documented*

Description: For DMA transfers on the expansion bus, the channel switch size is documented properly − that is, values of 0 to 15 give a switch size of 1 to 16. For transfers on the CPU bus, channel switch size of zero gives one transfer. Channel switch sizes 1 to15 give 1 to15 transfers.

Workaround: The documentation will be updated. (SPNU194, 11/2002)

Advisory DMA#26 — *Half-Word and Byte Writes to Unimplemented Bits Corrupt Register*

Description: Half-word or bytes to the high order bytes of DMA Global Control register or the DMA Global Disable register will corrupt these registers.

Workaround: The documentation will be updated to warn users about this condition. There is no reason to write to these unimplemented bits. (SPNU194, 11/2002)

Advisory DMA#28 — *DMA Fails During Execution of the SWP Instruction*

Description: When a DMA transaction is supposed to happen during the CPU execution of an SWP instruction that accesses memory, the DMA transaction does not happen.

Workaround: Halt the DMA whenever the SWP instruction must be used.

Advisory DMA#29	*DMA Corrupts PSA*

Description: If a DMA transaction occurs on the cycle before writing to the PSA enable bit to disable the PSA, the PSA will be corrupted.

Workaround: Halt the DMA before disabling PSA.

Advisory FW#3	*Configuration Mode Required for Sleep or Standby*

Description: The configuration mode must be set to enter sleep or standby modes.

Workaround: The documentation will be updated to reflect this requirement. (SPNU213, 12/2002)

Advisory FW#13	*Fails Initial Read of 0x0−0x7 in Pipeline Mode*

Description: Immediately after entering pipeline mode, a data read of location 0x04 immediately following a data read of location 0x0 will cause 0x04 to read as all 0's.

Workaround: Do a dummy data read of any location other than zero or four immediately after entering pipeline mode. The documentation will be updated to reflect this requirement. (SPNU213, 12/2002)

Advisory FMzPLL#1	*FM_ZPLL Must Operate in Multiply By Eight Mode*

Description: The multiplier bits defined in the System Module Global Control Register are fixed to multiply by 8.

Workaround: The prescaler bits in the System Module Global Control register offer bits to divide the clock by 1 through 8. This allows the software to choose any frequency of operation desired within the range defined in the device specification, hence there is no loss of resolution.

Advisory FMzPLL#2	*Frequency Modulation Spikes Too High*

Description: The effective modulation depth is higher than specified.

Workaround: None

Advisory FMzPLL#5	*OSCIN Limited to 10 MHz*

Description: When the PLL is enabled, the oscillator frequency can not be larger than 10MHz due to maximum VCO gain limitation.

Workaround: Maximum OSCIN frequency should not be greater than10MHz.

Advisory GIO#1	*Reading the Interrupt Offset Registers*

Description: When either of the two interrupt offset registers are read, and a higher priority interrupt occurs in the same cycle, the interrupt pending flag for the higher−priority interrupt is wrongly cleared, but the offset for the lower−priority interrupt is read. As a result, the lower−priority interrupt will be serviced twice and the higher−priority interrupt will not be serviced at all.

Workaround: Do not read the interrupt offset register to identify the pending interrupt with the highest priority. Instead, read from the interrupt pending flag register and use bit tests to decode the pending interrupt with the highest priority by software. An additional write to the flag register is necessary to clear the pending interrupt flag.

Advisory HET#19	*Reading the Interrupt Offset Registers*

Description: When either of the two interrupt offset registers are read, and a higher priority interrupt occurs in the same cycle, the interrupt pending flag for the higher−priority interrupt is wrongly cleared, but the offset for the lower−priority interrupt is read. As a result, the lower−priority interrupt will be serviced twice and the higher−priority interrupt will not be serviced at all.

Workaround: Do not read the interrupt offset register to identify the pending interrupt with the highest priority. Instead, read from the interrupt pending flag register and use bit tests to decode the pending interrupt with the highest priority by software. An additional write to the flag register is necessary to clear the pending interrupt flag.

Advisory MSM#1	*Unable to Read Locations Outside of the Zone*

Description: Unable to read the locations which are outside of the zone after reset with JTAG connected. This is dependent on the TI / Spectrum Digital driver that is used.

Workaround: Do a read of the MSM password location first. Flash470 tool versions 3.24 and later include a workaround that enables the user to choose the device name so that Flash470 can force the use of that device configuration and unlock the MSM without having to read the device ID.

Advisory RTI#6 — *Asynchronous Clear of RTI Tap Flag Not Recommended*

Description: When using the oscillator to clock the RTI counter, any asynchronous clear of the RTI Tap flag could cause an arbitration condition between the clear and the RTI module trying to set the flag. This will cause the Tap flag to not get set, and hence the Tap interrupt to not occur.

Workaround: Before attempting to clear the RTI Tap flag, the RTI counter needs to be checked to make sure that the RTI module is not about to set the flag again.

Advisory RTI#7 — *Asynchronous SYSCLK and OSCIN Clocks*

Description: The SYSCLK and the OSCIN clocks are practically asynchronous to each other. This causes a timing issue for the Tap interrupt flag signal, which is generated using the OSCIN and is registered using the SYSCLK. This can cause the Tap flag not to be set when the Tap interrupt condition has occurred. This prevents the Tap interrupt from occurring and causes the CPU OS to hang if it is using the Tap interrupt as the OS tick.

Workaround: For applications that need a periodic RTI interrupt (e.g., for maintaining the Time-Of-Day clock) at all times (awake or in low power mode), use the RTICLK (PLL output clock) to clock the RTI instead of the oscillator when awake. The RTI clock source can be switched over to the oscillator when entering standby mode to provide a periodic wakeup interrupt, thus preventing an increase in the standby current consumption.

For applications that do not need a periodic RTI interrupt, use the RTICLK to clock the RTI, and use the halt mode as the low power mode instead of the standby mode. This will lead to a current saving for the overall application.

Advisory SCC#4	*CAN Does Not Perform Resynchronization As Expected*

Description: Due to the proposed update of the ISO-WD-16485 CAN Test specification (2001-05-31), the HCC/SCC on this device has a non-conformance to the Bosch CAN Specification and the ISO-11898 Standard as described below:

If the following conditions are met, the CAN does not perform a resynchronization as it is expected.

Conditions:

1. The node must be transmitter
2. The node must transmit a dominant bit
3. The dominant bit must be sampled back as recessive
4. A recessive to dominant edge must be detected after the sample point

But since the recessive sampling of the bit transmitted as dominant is an error anyway, an error frame will be transmitted at the beginning of the following bit.

Therefore, the effect of the non-conformance is a delay of this error frame. The maximum for this delay is five (max(SJW) + 1 Tq) time quanta.

Workaround: This non-conformance is classified as non-serious and does not have any impact on proper communication and inter-operability with other nodes. See above description.

Advisory SCC#6	*CANHRX Must be High During Self-test*

Description: The CANSRX pin must be high during self-test.

Workaround: The CANSRX pin is usually driven high by the bus transceiver. As long as there is no bus activity during the self-test, this is not a problem. If there is nothing driving the CANHRX pin, it can be configured as a digital output and set high during the self-test.

Advisory SCC#7	*Abort Acknowledge Bit Not Set After Transmission Request Reset*

Description:

After aborting a message using the Transmission Request Reset (TRR) register bit, there are some rare instances where the TRR bit will clear without setting the Abort Acknowledge (AA) bit.

In order for this rare condition to occur all the following three conditions must happen:

1. The current message has a message error or lost arbitration. This message does not need to have the same mailbox number as the following TRR bit mailbox.
2. The TRS bit of the same mailbox as the TRR mailbox must be set from either this current message, prior to the current message and still pending, or just set.
3. The TRR bit must be set in the exact ICLK cycle were the wrapper state machine is in IDLE for one cycle. (One ICLK before or after and the condition will not occur). This IDLE state can occur just after the current message. It can also occur just a few ICLKs after setting the TRS bit of any mailbox after the current message (point 1 above).

If these conditions occur then the TRR and TRS bits for the mailbox will clear t_{clr} ICLKs after the TRR bit is set where:

t_{clr} = ((16−mailbox_number)*2)+3 ICLK cycles

The TA and AA bits will not be set if this condition occurs. Normally, either the TA or AA bit sets after TRR bit goes to zero.

Workaround:

When this problem occurs, the TRR and TRS bits will clear within t_{clr} ICLK cycles. To check for this condition, first disable the interrupts. Check the TRR bits' t_{clr} ICLK cycles after setting the TRR bits to make sure that they are still set. A set TRR bit indicates the problem did not occur. If TRR is cleared, then maybe it was the normal end of a message and the TA or AA bits are set. Check both the TA and AA bits. If they are both zero, then the conditions did occur. Handle the condition like the interrupt service routine would, except that the AA bit does not need clearing now. If the TA or AA bit is set, then the normal interrupt routine will happen when the interrupt is re-enabled.

Advisory SPI#1	*Slave Baud Rate Setting*

Description: When the SPI is operated in slave mode, the SPI clock must be configured to a baud rate as close to the master's baud rate as possible. If the baud rate is too slow, the enable signal will not be generated in time to keep the master from sending additional data. If the baud rate is too fast, the slave will capture the data before the last bit is shifted in.

Workaround: The documentation will be updated to reflect this requirement. (SPNU195C, 7/2003)

Advisory SPI#2	*Clearing, Setting SPI EN Bit Does Not Clear Internal Flag*

Description: Clearing and then setting the SPI EN bit does not clear an internal flag that indicates there is valid data in the SPI data register. This could lead to an inadvertent overrun error. The software should do a dummy read of SPIBUF after setting the SPIEN bit to clear the internal flag.

Workaround: The documentation will be update to reflect this requirement. (SPNU195C, 7/2003)

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

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

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI 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 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. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

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

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
Low Power Wireless	www.ti.com/lpw	Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

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