TMS470R1B512

TMS470 Microcontroller

Silicon Errata

SPNZ141
June 2005

Contents

1 Known Design Marginality/Exceptions to Functional Specifications

The following is a list of advisories on modules in the D version of silicon. 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)

Clock control module (CCM)

Direct memory access controller (DMA)

Flash pump (FP)

Flash wrapper (FW)

General-purpose input/output (GIO)

High-end CAN controller (HCC)

High-end timer (HET)

Real-time interrupt (RTI)

Serial peripheral interface (SPI)

Zero-pin phase-locked loop (ZPLL)

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

Description: When used in FIFO mode, if the A to D module is disabled 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 CCM#1 — *ICLK Not 50% Duty Cycle*

Description: The ICLK signal output from the CCM is not a 50% duty cycle signal when the SYSCLK to ICLK divide ratio is odd. This affects the SCI and SPI modules and occurs when the divide ratio is 3 or above.

Workaround: There are two prescalers between the SYSCLK and the SCI or SPI baud clocks: the one between SYSCLK and ICLK, and the one between ICLK and the baud clock. If at least one of these prescalers is even, the SCI or SPI baud clock will be a 50% duty cycle clock.

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#15 — *CPU Reads of DMA Control Packet Memory*

Description: If the ARM7 CPU is reading the DMA control packet memory while the DMA is operating, the DMA control packet configuration word or the DMA control packet transfer count can be corrupted.

Workaround: The DMA SPD version 1.10 avoids this problem by keeping a copy of the DMA control packet configuration words in RAM. Using the latest version of this SPD will avoid the problem. Do not read the DMA control packet memory while the DMA is operating. Be careful to avoid instructions that perform a read-modify-write operation on the DMA control packet memory while the DMA is operating.

Advisory DMA#17 — *CPU Reads of MPU Registers During DMA*

Description: If the ARM7 CPU is reading a 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 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 to clarify this point. (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 FP#7	*VNV Voltage Adjusted to Set 0xA Step to −7.86V*

Description: This is to improve yield and reduce erase time. No functional impact or changes to software.

Workaround: None

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 FW#14 — *Wait States Must Be Set From Highest to Lowest*

Description: Wait states must be set by bank from highest to lowest wait states. Otherwise, if the higher number of wait states is written last, this value will apply to all banks.

Workaround: Set the wait states in each bank by writing to the bank requiring the most wait states first and proceeding to the bank requiring the least wait states last. The documentation will be updated to reflect this requirement. (SPNU213, 12/2002)

Advisory FW#15 — *No Wakeup From Powerdown in Pipeline Mode*

Description: On this device, Flash banks in pipeline mode put into standby/sleep mode can not wake up by doing a normal read access or any other wake-up interrupt. Therefore, it is not possible to use the automatic powerdown feature of banks that are not accessed for a given number of cycles.

Workaround: Do not use automatic powerdown of banks on this device during normal operation in pipeline mode.

Advisory FW#17 — *Access to Non-Existing Bank Hangs CPU*

Description: If all banks are in sleep or standby mode and an access to a non-existing bank is performed, the CPU will hang.

Workaround: Make sure the decoder MFBAH/L0 and MFBAH/L1 registers are set properly so that an access to a non-existing memory bank will generate an illegal access exception. The documentation will be updated to reflect this requirement. (SPNU213, 12/2002)

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 HCC#4 — *Delayed Frame Error*

Description: Due to the proposed update of the ISO-WD-16485 CAN Test specification (2001−05−31), the HCC 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 re-synchronization 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 HCC#6 — *CANHRX Must be High During Self-test*

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

Workaround: The CANHRX 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 HCC#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:

$$\mathbf{t_{clr} = ((16-mailbox\_number)*2)+3 \quad ICLK\ cycles}$$

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 HET#15 — *Auto Read Clear Malfunction*

Description:

The HET Auto Read Clear feature does not always work properly. Specifically, the data field of instruction X is NOT cleared if the following conditions a) and b) are true at the same time:

a) The 64-bit CPU read access happens exactly in the two HET time slots PRECEDING the time slot Y in which instruction X is executed.

b) Instruction X just changes its data field (in time slot Y). (Example: Instruction X is an ECNT instruction, which just detected an edge). The malfunction does NOT occur if the data field of instruction X does not change, since then b) is not true.

Workaround:

See above.

Advisory HET#16 — *No PWM With MCMP*

Description: MCMP causes a constant signal instead of a PWM, if both of the following conditions are met:

1. Consecutive compare match in every LRP for order=reg_ge_data (only when [data=0]).
2. The high resolution delay (in number of SYSCLK cycles) is equal to the time slot the MCM is executed.

Workaround: Replace each MCMP with a two instruction sequence: ECMP and MOV32

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 RTI#3 — *Tap Interrupt When Clearing Counter*

Description: Write accesses to the RTICNTR register will clear the CNTR (21 bit counter), which causes a Tap interrupt if the corresponding bit switches from a “1” to a “0”.

Workaround: Disable the RTI prior to changing the RTICNTR value.

Advisory RTI#4 — *Tap Interrupt When Clearing Counter in Suspend Mode*

Description: Write accesses to the RTICNTR register will clear the CNTR (21 bit counter), which causes a Tap interrupt if the corresponding bit switches from a “1” to a “0” when the suspend signal is asserted.

Workaround: This is the same problem as RTI#3, however, on the initial fix of RTI#3, the case where the suspend signal is asserted because of an emulator breakpoint was not considered. This problem occurs when the emulator has set a breakpoint on one of the instructions closely following the instruction which writes to the counter.

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 less than or equal to the master SPI clock, but not so slow that there are less than two slave device ICLK cycles for each actual SPICLK period. An SPI baud rate for the slave too much slower than the actual SPI baud rate of the master will delay the assertion of the SPInENA signal from the slave, allowing for the possibility that the master will start a new SPI transmission before the slave is ready.

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)

Advisory ZPLL#1 — *Clock Corruption When Changing Multiplier*

Description: All interrupt requests coming to the CIM module must be disabled when changing between multiply-by-4 and multiply-by-8.

Workaround: Disable the interrupt request at the peripheral source if possible.

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
		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