TRANSPORTATION SYSTEMS GROUP
CUSTOMER ERRATA AND INFORMATION SHEET
Part: MPC555.K
General Business Use
Report Generated: Tue Dec 10, 2002, 16:11:10

MPC555.K

ERRATA AND INFORMATION SUMMARY

AR_252 Censorship has been disabled within the CMF
AR_1082 TPU ROM: Channels with the COMM ROM function affect other channels
AR_697 Revised operating currents
AR_381 New Features on MPC555 mask revision J76N and later
AR_412 Avoid instruction fetches from IMB/UIMB memory map
AR_588 MASKNUM field in USIU is 0x20
AR_597 AC timing changes
AR_851 Documentation: COLIE is bit 10 of COLIR Register
AR_1067 MPC555: Errata Lists for discontinued revisions no longer updated
AR_636 CMF: Program at reduced temperature and voltage ranges
AR_730 Excessive pulses required for setting censorship.
AR_522 CMF: PAWS bits always read as 0
AR_449 Updated Flash Programming Algorithm
AR_485 Disable of TPU emulation mode while MISC enabled corrupts data in RAM
AR_810 MIOS: Synchronize writes to DASM B channel in OPWM mode.
AR_445 Potential trap state in MIOS MDASM in OPWM mode.
AR_468 Configure MIOS/VF/VFLS pins as all MIOS or all VF/VFLS
AR_443 MIOS: Do not write data into the MDASMBR when in an input mode.
AR_444 MIOS: Warning in MDASM OPWM mode when MDASMAR = MDASMBR.
AR_446 MIOS: Avoid 100% pulse in MDASM OPWM mode.
AR_517 MIOS: Read MIOS1VNR and MIOS1TPCR registers are undefined
AR_624 MIOS: Use even values in MPWMSMPERR
AR_452 MIOS: "non-reset" registers are undefined after reset.
AR_470 150V MM ESD issues
AR_946 Pull not disabled by SPRDS on Bus control pins
AR_454 Use external resistors/drivers when using external reset configuration word
AR_524 TS_B input needs additional input hold time
AR_680 CLKOUT and ENGCLK drive strengths will change
AR_736 CLKOUT well tie connected to wrong supply.
AR_940 JTAG: Do Not Switch All Pads Simultaneously When JTAG Enabled
AR_982 PADS: QADC64 Port A (and MUX Out) have CMOS Digital Outputs
AR_1019 RCPU: Don’t execute overflow type before update type MUL/DIV instruction
AR_1077 RCPU: Do not run multi-master compressed application with Show Cycles and BTB
AR_1076 RCPU: Treat VF queue flush information value of 6 as 2
AR_907 RCPU: Issue ISYNC command when entering debug mode
AR_440 RCPU: Execute any IMUL/DIV instruction prior to entering low power modes.
AR_211 Do not set breakpoint on mtspr ICTRL instruction
AR_214 Only negate interrupts while the EE bit (MSR) disables interrupts
AR_478 QADC64: Don’t use channel 63 "End Of Queue".
AR_563 QSM/QSMCM/QADC64 corrupts data after an IACK cycle in CISC parts.
AR_754 QADC64:Do not use queue1 in external gated mode with queue2 in continuous mode.
AR_768 QADC64: Queue2 activity may reset Queue1 ExtGates Single Scan SSE
AR_421 QADC64: Don’t switch to software triggered continuous scan after completing Q1.
AR_422 QADC64: Do not rely on set of TOR1 in external gated continuous scan mode
AR_419 QADC64: False trigger upon configuration (depends on chip configuration)
AR_420 QADC64: Don’t change BQ2 with a set of SSE2 without a mode change.
AR_435 QADC64: TOR1 flag operates in both single and continuous external gated modes.
AR_563 QSM/QSMCM/QADC64 corrupts data after an IACK cycle in CISC parts.
AR_584 QSMCM: Do not use link baud and ECK modes
AR_1045 CAN: Bus Off recovery not ISO compliant
AR_627 TPU: (Microcode) Add neg_mrl with write_mer and end_of_phase

AR_577 TPU3 - TCR2PSCK2 bit does not give TCR2 divide ratios specified.
AR_498 UIMB: Read failures occur for IMB accesses when IMB clock is half speed
AR_896 UIMB: Avoid external code in addresses 0xZ[3,7,B,F]0_7F80 to 0xZ[3,7,B,F]0_7FFC
AR_985 USIU: Do not use ORx[EHTR] with Dual Mapping
AR_582 USIU:Under certain conditions, XFC stuck at 0V on power-on
AR_595 USIU: PLL will not lock on power-on, use limp mode and switch via software
AR_925 USIU: TEXP feature does not function when VDD supply is off
AR_610 Additional current on KAPWR when power is not applied
AR_909 USIU: Do not assert cr_b to abort pending store reservation access
AR_910 USIU: PITRTC Clock may not work when SCCR[RTDIV] is 0
AR_984 USIU: Setting of SCCR[EBDF] may slow execution of code
AR_287 USIU: System to Time Base frequency ratio must be greater than 4
AR_479 USIU: The MEMC does not support external master burst cycles
AR_679 USIU: In slave mode, do not use write slave accesses
AR_389 Little Endian modes are not supported
AR_442 Avoid loss of clock during HRESET
AR_594 USIU: Changing PLL MF to 1:1 mode can have 180 degree phase shift
AR_598 USIU: Ensure proper configuration for proper startup
AR_687 USIU: Program reserved bits in PDMCR to preserve compatibility

DETAILED ERRATA DESCRIPTIONS

CDR_AR_252 Customer Erratum MPC555.K

Censorship has been disabled within the CMF

DESCRIPTION:
The censorship mode for the CMF modules has been disabled. The value of the CMF Censor bits will not provide any protection of the contents of the CMF memory array.

WORKAROUND:
To ensure similar behavior on future revisions with censorship enabled, boot code residing within the CMF flash module should set the ACCESS bit.

CDR_AR_1082 Customer Erratum MPC555.K

TPU ROM: Channels with the COMM ROM function affect other channels

DESCRIPTION:
The TPU COMM ROM Function causes problems in other channels. When the Host Service request is set to 0b11, all channels that do not use the COMM function will be forced to outputs and a random state will be selected.

WORKAROUND:
Either: 1) Re-initialize all other channels after the COMM function has been initialized; or 2) If a fixed COMM TPU function is required, download an updated TPU ROM image into the DPTRAM and use the TPU in emulation mode.

CDR_AR_697 Customer Information **MPC555.K**

Revised operating currents

DESCRIPTION:
Characterization of silicon indicates that the operating current specifications must be updated. The total current is not anticipated to change significantly, but will be redistributed amongst VDDL, VDDI, KAPWR, VDDSRAM, VDDSYN, and VDDF.

WORKAROUND:
Refer to electrical specification 3.3 or later for revised values.

CDR_AR_381 Customer Information **MPC555.K**

New Features on MPC555 mask revision J76N and later

DESCRIPTION:
Several new features were added to the MPC555 starting with mask revision 00J76N and also included in revisions K02A. In the USIU, the DBCT and DBSLR clock control bits were added ("Disable backup clock for timers", "Disable clock switch in loss of lock and reset"). In the USIU, a mode was added to allow the WE pins to also assert on reads, allowing the usage of some SRAMS. An additional "MTS" function has been multiplexed onto the IRQ2/CR/SGPIO2 pin. The MTS pin allows for sharing of additional types of devices in a multi-master system. In addition, the CMF FLASH programming control has changed. The recommended connection of the VSSSYN pin has changed. The recommended connection of the crystal has changed (resistor is now internal).

WORKAROUND:
Consult a revised users manual (15 September 1998 or later) to determine how to use these features. Use the latest version of the FLASH programming tools (version 1.1 or later of CMF_DEMO routines).

CDR_AR_412 Customer Information **MPC555.K**

Avoid instruction fetches from IMB/UIMB memory map

DESCRIPTION:
Instruction fetches on the IMB or to UIMB control registers may result in improper operation, possibly requiring reset to continue.

WORKAROUND:
Avoid instruction fetches from the IMB/UIMB memory map. Program the IMPU to disable instruction accesses to the IMB/UIMB memory map.

CDR_AR_588 Customer Information **MPC555.K**

MASKNUM field in USIU is 0x20

DESCRIPTION:
MASKNUM field in USIU has been changed to 0x20 and will change on future revisions.

WORKAROUND:
Modify software to expect new value (0x20) for the MASKNUM field.

CDR_AR_597 Customer Information **MPC555.K**

AC timing changes

DESCRIPTION:
Some of the AC timing specifications have changed. Refer to electrical specification 3.3 or later for new values. See CDR_AR_524 for AC timing specification 3.0. In addition, the following electrical specifications have changed to the following new values: {sp7, sp7a, sp7b, sp7c, sp7d} 4ns, {sp8a, sp8c, sp8d} 14ns, sp8b 15ns, sp10 14ns, sp11 14ns, sp15 12ns, sp15b 8ns, sp22 9ns, sp28 9ns, sp41 18ns. D(0:31) has been moved from sp7 and sp8 to sp7d and sp8d.

WORKAROUND:
Ensure external devices are matched to these specifications. These updated electrical specifications

---

CDR_AR_851 Customer Information **MPC555.K**

Documentation: COLIE is bit 10 of COLIR Register

DESCRIPTION:
The COLIE (Change of Lock Interupt Enable) bit was incorrectly listed as bit 9 of the COLIR register. The correct location is bit 10.

WORKAROUND:
Software should be updated to reflect the proper bit location. Refer to Reference Manuals dated October 15,2000 or later for the correct bit location.

CDR_AR_1067 Customer Information **MPC555.K**

MPC555: Errata Lists for discontinued revisions no longer updated

DESCRIPTION:
As of October 2002, Errata lists for discontinued versions of the MPC555 will no longer be updated. All revisions 0, A, C, G, and K have been out of production for more than 2 years. New errata that affect these revisions will no longer be updated.

WORKAROUND:
Convert to later versions of the MPC555, either K2, K3 or M.

CDR_AR_636 **Customer Erratum** **CMF.192KB_CDR1UBUS_03_0A**

CMF: Program at reduced temperature and voltage ranges

DESCRIPTION:
There may be insufficient program margin to be able to correctly read all bits of the array at cold with 3.0Vdd if the part was programmed at hot.

WORKAROUND:
Workarounds in order of effectiveness: First, reduce temperature while programming. Second, reduce Vpp while programming. Third, increase VDD while programming and reading. During programming, limit the maximum ambient temperature to 85C, and Vdd to 3.3V +/- 5%. This allows sufficient margin to read flash cells over the entire specified temperature and voltage ranges. By further restricting Vdd to 3V +/- 5% during all operations (including flash read), the maximum programming temperature may be increased to 90C with sufficient program margin to operate over the entire temperature range.

CDR_AR_730 **Customer Information** **CMF.192KB_CDR1UBUS_03_0A**

Excessive pulses required for setting censorship.

DESCRIPTION:
Setting the censor bits with the released driver code takes an excessive amount of pulses (100’s when it should be 10-15).Root Cause: With the previous release driver code which follows the old published censorship set algorithm, the row addresses to the array varied during high voltage set operations. These row addresses caused some high voltage logic in the array to vary, placing a load on the charge pump whenever the address changed. As a result, the charge pump was overloaded and insufficient voltage was applied to the fuse during the set pulses.

WORKAROUND:
Use the latest driver code (CMF Parallel Driver v2.2 or later) which includes the following fix. For Setting the censor bit, insert program interlock write prior to writing EHV. But if the transition is from 00 to 01 and the ACCESS bit[s] is/are not set and software is running in censored mode, do not perform the interlock write[s], and instead after writing the EHV bit[s] ensure that U-bus addresses17:25 do not change for the 100ms pulse duration. (Perform a tight loop which lasts > 100ms, do not poll HVS bit[s] during that loop, ensure that the loop and any associated prefetching or operand accesses does not result in addresses which alter A[17:25] on the u-bus.)

CDR_AR_522 **Customer Information** **CMF.192KB_CDR1UBUS_03_0A**

CMF: PAWS bits always read as 0

DESCRIPTION:
The PAWS bits in the CMFTST register are not readable. They will always read as 0, even if they are in a different state.

WORKAROUND:
Avoid reading the PAWS bits to confirm the proper programming.

CDR_AR_449 **Customer Information** **CMF.192KB_CDR1UBUS_03_0A**

Updated Flash Programming Algorithm

DESCRIPTION:
The program/erase pulse algorithm has changed. Consult the latest flash programming algorithm for the number and method of applying pulses. The program pulse width (subject to change) should be 25.6us. The erase pulse width should be 1s (for all revisions of all parts). Longer program or erase pulse widths can result in improper program or erase. Previous revisions (Rev. A, Rev. C) of the part should continue using shorter programming pulses. The programming algorithm and pulse widths are still subject to change.

WORKAROUND:
Update erase pulse widths. Obtain the latest programming algorithm.

CDR_AR_485 **Customer Information** **DPTRAM.6K_CDR1IMB3_03_0**

Disable of TPU emulation mode while MISC enabled corrupts data in RAM

DESCRIPTION:
If the TPU emulation mode is negated while MISC is enabled, the DPTRAM data may be corrupted.

WORKAROUND:
In test mode / TPU development mode, disable the MISCEN (DPTMCR) before negation of TPEMEM in the TCR. In normal mode, disable MISCEN prior to performing a soft reset of the TPU (TPUMCR2).

CDR_AR_810 **Customer Erratum** **MIOS1.CDR1IMB3_02_0**

MIOS: Synchronize writes to DASM B channel in OPWM mode.

DESCRIPTION:
In some cases of using DASM in OPWM mode, when writing B register, the output of the OPWM channel will remain asserted when it was expected to be negated. When B register of an OPWM channel is updated in the same system clock that a match on B is expected, the match will be ignored, and the OPWM output will remain asserted until the match on the new value of B. The problem sequence is: (1) OPWM output is asserted when A match occurs. (2) This match may trigger a SW task (e.g., via interrupt on A match) that updates B register (B is double buffer in this mode, i.e., host writes a new compare value to B, the OPWM output negates when the timer matches to the original value of B, then the new value is copied to the comparator to be used in the next PWM cycle) (3) If the write to B occurs at the same system clock that B match is expected, the match will not be recognized. (4) In this case, the OPWM output will remain asserted until the next B match that follows the next A match.

WORKAROUND:
Follow one of the following procedures: (1) read the relevant counter value (such as MCSM) before doing the DASM write. If the counter value is "just below" the old DASM B value, then the B register update should be delayed. (2) write to B register, and then check if the value of the relevant counter (such as MCSM) is bigger than the OLD B value. If so, it is required to force the pin value to it’s desired state (FORCB). (3) When using DASM interrupt to update B register, verify that the PWM pulse width is larger than the interrupt latency. (4) keep the B value constant and only perform writes to the A register to alter the pulse width. In this case the value of A is updated after the interrupt from the previous channel A match. For this to work the pulse period needs to be greater than the interrupt latency, so that the new A value is written before the next A channel compare is enabled. (Note that the A register is not double buffered).

CDR_AR_445 **Customer Erratum** **MIOS1.CDR1IMB3_02_0**

Potential trap state in MIOS MDASM in OPWM mode.

DESCRIPTION:
A trap state is entered when a value of MDASMBR is written in OPWM mode, to a value which is out of the counter bus range. For example, if the modulus value of the MCSM driving the counter bus is $FF00 and if MDASMBR is written to a value less than $FF00, then a match is never made on channel B hence a B1 to B2 transfer never occurs. To get out of the trap, the MDASM mode should be reset back to idle.

WORKAROUND:
Ensure that the software never writes MDASMBR (in OPWM mode) to a value which is less than the MCSMMOD value.

CDR_AR_468 Customer Erratum MIOS1.CDR1IMB3_02_0

Configure MIOS/VF/VFLS pins as all MIOS or all VF/VFLS

DESCRIPTION:
The MIOS VF/VFLS multiplexer must not be individually programmed. These pins can be configured either as VF/VFLS or as all MIOS pins. Do not configure bit[0:1] of the MIOS1TPCR register as 2’b01 or 2’b10.

WORKAROUND:
Whenever the user wishes to configure the MIOS/VF/VFLS pins, software should write 2’b11 or 2’b00 to bit[0:1] of the MIOS1TPCR register. This will allow the pins to be either all MIOS functions or all development support functions. The pins should never be configured separately.

CDR_AR_443 Customer Erratum MIOS1.CDR1IMB3_02_0

MIOS: Do not write data into the MDASMBR when in an input mode.

DESCRIPTION:
The MDASMBR register can be loaded via a write from the IMB, from the counter bus OR from a transfer from the MDASMAR to the MDASMBR register. The transfer from MDASMAR to MDASMBR only happens in input modes, when the software should not normally write into MDASMBR. However, no hardware exists to prevent a simultaneous transfer from MDASMAR and write to MDASMBR. As a result, during a simultaneous transfer and write, the resulting data in MDASMBR will be undefined. The specification does not define what happens in this case.

WORKAROUND:
When in an input mode, do not write to MDASMBR.

CDR_AR_444 Customer Erratum MIOS1.CDR1IMB3_02_0

MIOS: Warning in MDASM OPWM mode when MDASMAR = MDASMBR.

DESCRIPTION:
In OPWM mode when a comparison occurs simultaneously on register A and B (i.e. they have the same value stored), the pin is reset or stays reset. The specification states that the transfer between B1 and B2 should occur when the pin is low. However this is not necessarily the case when a simultaneous A&B compare occurs. If the pin was previously low then the transfer would not happen until after the next compare.

WORKAROUND:
1): Avoid setting MDASMAR = MDASMBR when in OPWM mode. 2): To come out of MDASMAR = MDASMBR when in OPWM mode change the value of MDASMAR first. 3): Be aware that it may take an extra match to update MDASMBR B2 than expected, after a new value is written to MDASMBR B1.

CDR_AR_446 Customer Erratum **MIOS1.CDR1IMB3_02_0**

MIOS: Avoid 100% pulse in MDASM OPWM mode.

DESCRIPTION:
A two cycle system clock "glitch" (to logic "0") may occur when 100% output state is entered in MDASM OPWM mode. 100% pulse is entered by writing B register bit 15 high when using less than 16 bit resolution. The problem occurs only when B register bit 15 is set while the pin is high; the glitch occurs on the next match on the B register.This glitch is only seen the first time a match on B causes 100% mode to be entered. No glitches will be seen on subsequent matches.

WORKAROUND:
- Use the pads with the slow slew rate. Then at 40 Mhz no glitch will be seen on the output pin.- If B register bit 15 is only set while the pin is low then there will not be any glitch on the pad. The change to 100% will occur 2 cycles after the setting of B register bit 15.Invert the polarity of the output. Then setting A=B will cause a 100% pulse. A glitch free 0% pulse is now no longer possible.

CDR_AR_517 Customer Erratum **MIOS1.CDR1IMB3_02_0**

MIOS: Read MIOS1VNR and MIOS1TPCR registers are undefined

DESCRIPTION:
A read of the MIOS1VNR and MIOS1TPCR registers will produce undefined data. All writes to the MIOS1TPCR will be performed correctly and cause the appropriate actions, but the values read from MIOS1TPCR will undefined.

WORKAROUND:
Avoid reading the MIOS1VNR and MIOS1TPCR registers.

CDR_AR_624 Customer Erratum **MIOS1.CDR1IMB3_02_0**

MIOS: Use even values in MPWMSMPERR

DESCRIPTION:
In some operating conditions on some parts, the MPWMSM period may be off by one count. The load of the MIOS MPWMSM counter LSB is not guaranteed to function correctly over all conditions. In some cases, the MIOS MPWMSM cannot load a MPWMSM period LSB (Least significant bit of MPWMSMPERR) as a "1" into the MPWMSM counter (MPWMSMCNTR). The counter LSB will be incorrectly loaded as a "0". The period of the PWMSM counter will then be one MPWMSM prescaler clock period less than programmed. There will be no problem if MPWMSM period LSB (MPWMSMPERR_PER0) is a "0". This problem is most likely to occur at lower temperatures and higher voltages, but may occur in other operating conditions. Example of failure: If the MPWMSMPERR is set to %057F, the actual value loaded into the MPWMSMCNTR will be %057E.

WORKAROUND:
Use even values in the MPWMSMPERR.

CDR_AR_452 **Customer Information** **MIOS1.CDR1IMB3_02_0**

MIOS: "non-reset" registers are undefined after reset.

DESCRIPTION:
The specification states that many of the MIOS data registers are unaffected by reset. This should really be "undefined" after a reset. Note that after reset all the MIOS submodules are correctly in their idle state with the pads as inputs.

WORKAROUND:
After a reset of the MIOS a full initialization routine should be run, rather than assuming that the same values remain in the MIOS data registers.

CDR_AR_470 **Customer Erratum** **PKPADRING.555_CDR1_02_0B**

150V MM ESD issues

DESCRIPTION:
Not all ESD specifications are met when tested using machine model (MM) tests. All specifications are met at 100V MM. Vdda pin fails at 150V MM; low level leakage is seen on 5v output pins at 200V MM; 3v pads with keep alive power exhibit low-level leakage at 200V MM. All pads pass Human Body Model (HBM) ESD tests at 3000V and below.

WORKAROUND:
Avoid indicated ESD levels on these pins.

CDR_AR_946 **Customer Erratum** **PKPADRING.555_CDR1_02_0B**

Pull not disabled by SPRDS on Bus control pins

DESCRIPTION:
Setting PDMCR[SPRDS] in the USIU will not disable the pullup of the pads: bi_b_sts_b, burst_b, bdip_b, ta_b, ts_b, tsiz1, tsiz0, tea_b, rd_wr_b, br_b_vf1_iwp2, bg_b_vf0_iwp0, bb_b_vf2_iwp3 when the pads are functioning as inputs. The spec says that these pins should be PU3 when driver not enabled, or until SPRDS is set.

WORKAROUND:
Insure that external pull downs or drivers can overcome the 130 uA maximum pull up.

CDR_AR_454 Customer Erratum **PKPADRING.555_CDR1_02_0B**

Use external resistors/drivers when using external reset configuration word

DESCRIPTION:
When asserting RSTCONF to direct the MPC555 to sample the reset configuration word from the external data pins, the weak pulldowns on the data pads may not fully discharge the pins during reset. If the data pins were driven high by the MPC555 just prior to the assertion of reset, the weak pulldowns will not be able to discharge the pins due to contention with the P-channel transistor of the output buffer. This transistor is not fully turned off by the pre-driver stage.

WORKAROUND:
Program the internal flash to provide the reset configuration word. Or, use external resistors/drivers to drive all of the configuration word during reset (including bits set to 0) when providing the reset configuration word from external. An external 10K resistor is sufficient to pull a data pin to 0 during reset.

CDR_AR_524 Customer Erratum **PKPADRING.555_CDR1_02_0B**

TS_B input needs additional input hold time

DESCRIPTION:
The TS_B signal, when an input (spec. 30), requires an input hold time of 5ns.

WORKAROUND:
Keep asserting TS_B for the additional hold time. In a multi-555 system, the TS_B output hold time of one MPC555 is sufficient to meet the TS_B input hold time of another MPC555.

CDR_AR_680 Customer Information **PKPADRING.555_CDR1_02_0B**

CLKOUT and ENGCLK drive strengths will change

DESCRIPTION:
Beginning with Revision M, the CLKOUT pad driver will be sized to drive loads of 30pf or 90pf, selectable by software. The ENGCLK pad driver will be sized to drive loads of 25pf or 50pf, selectable by software.

WORKAROUND:
Designs with clkout loads between 30pf and 45pf should evaluate setting the clkout driver to the 90pf drive mode. Designs with ENGCLK loads above 25pf should evaluate setting the ENGCLK driver to the 50pf drive mode. Designs with ENGCLK loads above 50pf should reduce the ENGCLK frequency to 10Mhz or below.

CDR_AR_736 Customer Information **PKPADRING.555_CDR1_02_0B**

CLKOUT well tie connected to wrong supply.

DESCRIPTION:
CLKOUT pad’s pchannel driver has different supplies connected to the well tie and the source.

WORKAROUND:
Customers need to power up VDD_CLK ahead of VDDi, or at the same time, to prevent parasitic diode from sinking high current into the pin.

CDR_AR_940 Customer Information PKPADRING.555_CDR1_02_0B

JTAG: Do Not Switch All Pads Simultaneously When JTAG Enabled

DESCRIPTION:
JTAG mode puts all output pins in fast slew rate mode. The power supply pins of the device cannot supply enough current to allow all pins to be changed at the same time in fast slew rate mode. During normal operation, this is not an issue since all pins on the device do not switch at the same time.

WORKAROUND:
When using JTAG, all pins should not be switched simultaneously.

CDR_AR_982 Customer Information PKPADRING.555_CDR1_02_0B

PADS: QADC64 Port A (and MUX Out) have CMOS Digital Outputs

DESCRIPTION:
The 15 October 2000 MPC555 User Manual incorrectly lists the QADC64 Port A (A:PQA[0:7], A:MA[0:2], B:PQA[0:7], B:MA[0:2]) pins as open drain when selected as digital outputs. This is not correct. These pins are normal CMOS drivers in digital output mode.

WORKAROUND:
Expect the QADC64 Port A pins (and multiplexor out pins when enabled) to drive both high and low when selected as digital outputs.

CDR_AR_1019 Customer Erratum RCPU.CDR1LBUSIBUS_13_0

RCPU: Don’t execute overflow type before update type MUL/DIV instruction

DESCRIPTION:
When an integer overflow type non multiply or divide instruction (designated by an ’o’ in the instruction mneumonic, such as addo) starts to execute before a previously started Condition Register 0 (CR0) update type integer multiply or divide instruction (designated by a ’.’ in the instruction mneumonic, such as divw.) completes, the CR0[SO] bit may be wrongly updated from the XER[SO] bit earlier changed by the overflow type instruction. For example, instruction sequence "divw. Rx,Ry,Rz , subfo Rt,Ru,Rv" may cause this problem. It does not happen if the overflow type instruction is also a CR0 update type instruction (designated by ’o.’ in the instruction mneumonic, such as addo.), or if register dependencies exist.

WORKAROUND:
Do any one of the following: 1) Keep a gap of at least 1 instruction between a CR0 update type integer multiply instruction and an overflow type instruction or a gap of 4 integer or 6 other instructions between a CR0 update integer divide instruction and an overflow type instruction; 2) Use the CR0 update type for both instructions; 3) Run the RCPU in serialized mode; 4) Place a "sync" instruction between the integer multiply/divide instruction and the overflow type instruction; 5) Don’t use the update form of integer multiply or divide instructions; or 6) Don’t use overflow type integer instructions. (Note: most compiler vendors do not generate the error case.)

CDR_AR_1077 Customer Erratum **RCPU.CDR1LBUSIBUS_13_0**

RCPU: Do not run multi-master compressed application with Show Cycles and BTB

DESCRIPTION:
If instruction show cycles (ICTRL[ISCT_SER] not equal to 0x7) and BTB are enabled in a compressed application with interrupts and another master (READI or External Bus master) initiates internal accesses on UBUS, the RCPU may execute incorrect instructions.

WORKAROUND:
Do not enable instruction show cycles together with BTB while running compressed application with interrupts if a UBUS master (READI or External Master) other than the RCPU or the L2U operated by the RCPU, accesses MCU internal resources through the UBUS.

CDR_AR_1076 Customer Erratum **RCPU.CDR1LBUSIBUS_13_0**

RCPU: Treat VF queue flush information value of 6 as 2

DESCRIPTION:
When the RCPU fetches instructions from zero wait state slaves on UBUS (Internal flash or SIU when in enchanced burst mode), the VF queue flush information may have the reserved value of 6.

WORKAROUND:
If a VF instruction queue flush value of 6 is shown on the VF pins, tools should treat this value as 2 for program tracking purposes.

CDR_AR_907 Customer Information **RCPU.CDR1LBUSIBUS_13_0**

RCPU: Issue ISYNC command when entering debug mode

DESCRIPTION:
If the ICTRL[28] bit is set (non-serialized mode) then the RCPU issues two instruction fetch requests into the instruction pipeline after entering debug mode. The debug port and the debug tool may get confused when processing an "mtspr DPDR,Rx" instruction. The debug tool loses synchronization with debug port and receives the wrong data for the "Rx" register. The typical case is when the debug tool tries to save scratch registers or read the debug mode cause.

WORKAROUND:
Issue an ISYNC instruction to the debug port prior to any other instructions when the RCPU enters debug mode after running code.

CDR_AR_440 Customer Information **RCPU.CDR1LBUSIBUS_13_0**

RCPU: Execute any IMUL/DIV instruction prior to entering low power modes.

DESCRIPTION:
There is a possibility of higher than desired currents during low power modes. This is caused by a possible contention in the IMUL/DIV control area. This contention may only exist prior to the execution of any IMUL/DIV instruction.

WORKAROUND:
Execute a MULLW instruction prior to entering into any low power mode (anytime after reset, and prior to entering the low power mode).

CDR_AR_211 Customer Information **RCPU.CDR1LBUSIBUS_13_0**

Do not set breakpoint on mtspr ICTRL instruction

DESCRIPTION:
When a breakpoint is set on an "mtspr ICTRL,Rx" instruction and the value of bit 28 (IIFM) is 1, the result will be unpredictable. The breakpoint can be taken or not on the instruction and value of the IFM bit can be either 0 or 1.

WORKAROUND:
Do not put a break point on mtspr ICTRL, Rx instruction when bit 28 of Rx is set to a 1.

CDR_AR_214 Customer Information **RCPU.CDR1LBUSIBUS_13_0**

Only negate interrupts while the EE bit (MSR) disables interrupts

DESCRIPTION:
An IRQ to the core, which is negated before the core services it, may cause the core to stop fetching until reset.

WORKAROUND:
Interrupt request to the core should only be negated while interrupts are disabled by the EE bit in the MSR. Software should disable interrupts in the CPU core prior to masking or disabling any interrupt which might be currently pending at the CPU core. For external interrupts, it is recommended to use the edge triggered interrupt scheme. After disabling an interrupt, sufficient time should be allowed for the negated signal to propagate to the CPU core, prior to re-enabling interrupts. For an interrupt generated from an IMB module, 6 clocks is sufficient (for IMBCLK in 1:1 mode).

CDR_AR_478 Customer Erratum **QADC64.CDR1IMB3_02_0B**

QADC64: Don’t use channel 63 "End Of Queue".

DESCRIPTION:
When operating at 150 C (junction temperature), low voltage, and high frequency a channel 63 written to a CCW does not properly act as an End Of Queue. The appropriate flags will recognize the End Of Queue and be set, but the queue will continue to operate past this point.

WORKAROUND:
Characterization of a small sample of parts indicates that this problem will not be seen if any of the following conditions are met: (1) Vdd must remain above 3.12 volts, or (2) Frequency must remain below 38 MHz, or (3) Temperature of part must remain below 100C, or (4) Channel 63 "End Of Queue" not used.

CDR_AR_563 Customer Erratum QADC64.CDR1IMB3_02_0B

QSM/QSMCM/QADC64 corrupts data after an IACK cycle in CISC parts.

DESCRIPTION:
This problem does not affect parts that do not run IACK cycles (i.e. RISC CPUs). The Common BIU state machine, used by the QSM/QSMCM/QADC64, mis-tracks an IACK cycle if an interrupt is issued while an IACK cycle for the same level is in progress. In this case, the next access on the IMB3 will be corrupted by the QSM/QSMCM/QADC64. On CPU32 based parts (or CPU32X parts where the FASRAM is not used for the stack), the first access after an IACK cycles is the stacking of the vector offset The risk to the system by corrupting this stacked value is very low, since it is not used by the processor or most interrupt service routine software. On CPU32X based parts which have the stack located in the FASRAM, however, the first IMB3 access is the fetch from the vector table. Corruption of this value (handler address) causes the processor to jump to an incorrect location or to produce an address error.

WORKAROUND:
Workarounds exist for both CPU32 and CPU32X based parts. On CPU32 based parts the first access after an IACK cycles is the stacking of the vector offset The risk to the system by corrupting this stacked value is very low, it is not used by the processor. On CPU32X based parts which have the stack located in the FASRAM the first IMB3 access is the fetch from the vector table. Corruption of this value (handler address) causes the processor to jump to an incorrect location or to produce an address error. The suggested workarounds for the QSM/QSMCM/QADC64 are listed below. For CPU32 based parts: - assign the QSM/QSMCM/QADC64 it’s own interrupt levels separate from any other modules if the corruption of the vector offset in the stack frame is an issue. For CPU32X based parts: (a) assign the QSM/QSMCM/QADC64 its own interrupt levels separate from any other module in the system or (b) move the stack out of the FASRAM.

CDR_AR_754 Customer Erratum **QADC64.CDR1IMB3_02_0B**

QADC64:Do not use queue1 in external gated mode with queue2 in continuous mode.

DESCRIPTION:
When the gate for queue1 opens when queue2 is converting the last word in its queue, queue1 completion flag will immediately set and no conversions will occur. Queue1 will remain in a hung state for the duration of the gate (no conversions will occur regardless of how long the gate is open). This failure will only occur when the QADC64 is configured with queue1 in external gated mode (continuous or single scan) and queue2 is in continuous mode. The failure mode can be detected if it is known that the gate for queue 1 is shorter than the length of the queue, and the completion flag becomes set. The failure can also be detected as follows: software writes invalid results to the result register (3ff when it is known the input will never go to full scale); after the gate has closed if the invalid result is still in result space 0, then the failure has occured.

WORKAROUND:
There are 2 workarounds: (1) Do not use queue 2 if queue1 is set for external gated mode. Or, (2) SETUP: (a) queue 2 mode : 'Interval Timer Single-Scan Mode' (MQ2 = 11000) so the interval is (1/(2MHz/2048)) = 1.024ms (b) Pause bit set in CCW60 (c) Pause bit set in CCW61. FUNCTIONALITY: SSE2 bit gets set, the timer starts, and the internal trigger comes after 1.024ms. queue2 will then start converting and will continue until it sees the pause bit in CCW60. So, a reset could occur every 2ms, and the SSE2 bit should be set allowing the queue to begin again never having reached an end of queue. If 'Task jitter' does occur, and the queue does not get reset before another internal trigger is created, then it will do a one word conversion and immediately pause again due to the pause bit set in CCW61. Even if there is enough 'Task jitter' to allow this sub-queue to begin, it will be paused after only one conversion and will not reach the end of queue. Finally, it is assumed that it would not be possible to have enough uncertainty for another level of sub-queues to be needed.

CDR_AR_768 Customer Erratum **QADC64.CDR1IMB3_02_0B**

QADC64: Queue2 activity may reset Queue1 ExtGates Single Scan SSE

DESCRIPTION:
If queue1 is in External Gated Single Scan mode, the SSE bit is written, and the queue is awaiting a trigger, an EOQ condition on queue2 will cause the queue1 SSE bit to be reset. This causes queue1 to not acknowledge any trigger, unless the SSE bit is again set.

WORKAROUND:
Do not let queue2 reach a EOQ while queue1 is in External Gated single scan mode, with SSE bit set, and awaiting a trigger. This may be done by breaking queue2 into sub-queues,and not allowing it to run to the end.

CDR_AR_421 Customer Erratum **QADC64.CDR1IMB3_02_0B**

QADC64: Don’t switch to software triggered continuous scan after completing Q1.

DESCRIPTION:
In the case when application software switches Q1 to software triggered continuous scan mode after Q1 completes a single scan where BQ2 provides the end of queue, an indeterminate response results.

WORKAROUND:
Don’t select software triggered continuous scan after using Q1.

CDR_AR_422 Customer Erratum **QADC64.CDR1IMB3_02_0B**

QADC64: Do not rely on set of TOR1 in external gated continuous scan mode

DESCRIPTION:
In External Gated Continuous Scan mode: If the external gate is negated during the last conversion (after the ccw has started, but before the result is converted) the TOR1 flag will not set.

WORKAROUND:
Control software needs to reflect the following: In external gated continuous scan mode, setting of TOR1 is guaranteed only if the gate remains open thru the completion of the last conversion in queue1.

CDR_AR_419 Customer Information **QADC64.CDR1IMB3_02_0B**

QADC64: False trigger upon configuration (depends on chip configuration)

DESCRIPTION:
In some implementations, the QADC64 may have a false trigger upon entering an external trigger mode. The potential for a false trigger only exists on QADC64’s which are implemented with trigger pin(s) muxed through PortA[3 or 4]. If the triggers have dedicated pins, then no difference exists between the value on the pin and the value between the pad and the module.The false trigger can result when an edge triggered mode is enabled and the logic value at the pin and the previously latched value in the pad are not equal.

WORKAROUND:
A port data register read may be performed prior to entering an external trigger mode to ensure that the latched value between the pad and the module matches the value on the pin. This read ensures that an edge will not be caused by the latch in the pad becoming transparent when the external trigger mode is entered. This issue does not exist on the following parts: MPC555.

CDR_AR_420 Customer Information QADC64.CDR1IMB3_02_0B

QADC64: Don’t change BQ2 with a set of SSE2 without a mode change.

DESCRIPTION:
Changing BQ2 and setting SSE2 with no mode change will cause Q2 to begin but not recognize the change in BQ2. Further, changes of BQ2 after SSE2 is set, but before Q2 is triggered are also not recognized. All other sequences involving a change in BQ2 are recognized.

WORKAROUND:
Be sure to do mode change when changing BQ2 and setting SSE2. Recommend setting BQ2 first then setting SSE2.

CDR_AR_435 Customer Information QADC64.CDR1IMB3_02_0B

QADC64: TOR1 flag operates in both single and continuous external gated modes.

DESCRIPTION:
TOR1 response was added to QADC64 to provide an indication of more than 1 pass through queue1. It was described in the specification as a continuous mode only flag. The flag is however, operating in both single and continuous modes.

WORKAROUND:
None. Simply expect the flag to respond in both single scan and continuous scan modes.

CDR_AR_563 Customer Erratum QSMCM.CDR1IMB3_02_0

QSM/QSMCM/QADC64 corrupts data after an IACK cycle in CISC parts.

DESCRIPTION:
This problem does not affect parts that do not run IACK cycles (i.e. RISC CPUs). The Common BIU state machine, used by the QSM/QSMCM/QADC64, mis-tracks an IACK cycle if an interrupt is issued while an IACK cycle for the same level is in progress. In this case, the next access on the IMB3 will be corrupted by the QSM/QSMCM/QADC64. On CPU32 based parts (or CPU32X parts where the FASRAM is not used for the stack), the first access after an IACK cycles is the stacking of the vector offset The risk to the system by corrupting this stacked value is very low, since it is not used by the processor or most interrupt service routine software. On CPU32X based parts which have the stack located in the FASRAM, however, the first IMB3 access is the fetch from the vector table. Corruption of this value (handler address) causes the processor to jump to an incorrect location or to produce an address error.

WORKAROUND:
Workarounds exist for both CPU32 and CPU32X based parts. On CPU32 based parts the first access after an IACK cycles is the stacking of the vector offset The risk to the system by corrupting this stacked value is very low, it is not used by the processor. On CPU32X based parts which have the stack located in the FASRAM the first IMB3 access is the fetch from the vector table. Corruption of this value (handler address) causes the processor to jump to an incorrect location or to produce an address error. The suggested workarounds for the QSM/QSMCM/QADC64 are listed below. For CPU32 based parts: - assign the QSM/QSMCM/QADC64 it’s own interrupt levels separate from any other modules if the corruption of the vector offset in the stack frame is an issue. For CPU32X based parts: (a) assign the QSM/QSMCM/QADC64 its own interrupt levels separate from any other module in the system or (b) move the stack out of the FASRAM.

CDR_AR_584 Customer Information QSMCM.CDR1IMB3_02_0

QSMCM: Do not use link baud and ECK modes

DESCRIPTION:
Reads of the SCI control and status registers do not read correctly when using the link baud or the external clock source feature of the QSMCM. These modes are enabled by the SCCxR0 control register bits 0 and 1 (OTHR and LNKBD). These modes are not fully operational.

WORKAROUND:
Do not use the link baud or external clock modes of the QSMCM. The OTHR bit in the SCCxR0 control register 0 must be set = 0 to use normal mode operation only.

CDR_AR_1045 **Customer Information** **TOUCAN.CDR1IMB3_04_0**

CAN: Bus Off recovery not ISO compliant

DESCRIPTION:
The Bus Off recovery is not ISO compliant on the FlexCAN and TouCAN modules. The ISO specification indicates that the CAN node should remain inactive until user intervention restarts it. The FlexCAN and TouCAN modules both include an automatic recovery mechanism for the Bus Off condition.

WORKAROUND:
The Bus Off condition interrupt should be enabled and an interrupt service routine implemented to disable the CAN. The user’s software should then determine when the CAN should be re-activated.

CDR_AR_627 **Customer Information** **TPU3.CDR1IMB3_02_2**

TPU: (Microcode) Add neg_mrl with write_mer and end_of_phase

DESCRIPTION:
Incorrect generation of 50% duty cycle is caused by the command combination "write_mer, end". If the write_mer is the last instruction together with the end, this may create an additional match using the old contents of the match register (which are in the past now and therefore handled as an immediate match)

WORKAROUND:
Add neg_mrl together with the last write_mer and with end-of-phase. The negation of the flag overrides the false match which is enabled by write_mer and postpones the match effect by one micro-instruction. In the following micro-instruction the NEW MER value is already compared to the selected TCR and no false match is generated. The neg_mrl command has priority over the match event recognition, separating the write_mer and the end command. This gives enough time for the new MER to update before the channel transition re-enables match events.

CDR_AR_577 **Customer Information** **TPU3.CDR1IMB3_02_2**

TPU3 - TCR2PSCK2 bit does not give TCR2 divide ratios specified.

DESCRIPTION:
The TCR2PSCK2 bit was originally specified to cause the TCR2 timebase to be divided by 2. Actually, it causes the TCR2 timebase to be divided as follows: The /16 of external clock and /128 of internal clock are eliminated and /3, /7, /15 of the external clock and /24, /56, /120 of the internal clock are added.

WORKAROUND:
When the TCR2PSCK2 is set, instead of the specified divides of /16, /32, /64, /128, expect the internal clock source to be /8, /24, /56 and /120 for TCR2 Prescaler values of 00, 01, 10 and 11, respectively. Likewise, for the external clock source expect /1, /3, /7, /15 instead of /2, /4, /8, /16.

CDR_AR_498 Customer Erratum UIMB.CDR1UBUSIMB3_02_0

UIMB: Read failures occur for IMB accesses when IMB clock is half speed

DESCRIPTION:
When the IMB clock is at half speed, a speed path occurs which prevents the proper data in the UIMB internal data latches from being observed by the user. Data is transferred from the latches before the latches are updated with data for the current cycle. This failure occurs when the part is heated (80-100C) and the frequency is at 40Mhz.

WORKAROUND:
There are 3 possible workarounds: (1). Since the internal latches are late in being updated with IMB data, it takes 2 consecutive reads from the same IMB location to observe the proper data from that location. The data from the first access should be disregarded when in half speed mode. (2). When running half speed on the IMB, keep the part as close to room temp. (25C) as possible. or (3). Only use full speed IMB mode. This workaround only applies to RevC or later.

CDR_AR_896 Customer Erratum UIMB.CDR1UBUSIMB3_02_0

UIMB: Avoid external code in addresses 0xZ[3,7,B,F]0_7F80 to 0xZ[3,7,B,F]0_7FFC

DESCRIPTION:
When two UBUS cycles are precisely pipelined, such that the first cycle is to an address within the IMB address range, (Internal memory map base address + 0x300000:0x307F7F), and the 2nd cycle is a fetch to an external address in which A[10:29] match A[10:29] of any unimplemented register of the UIMB module, then the IMB cycle will be tagged with an error resulting in a machine check exception. During operation, the pipelining of fetches relative to an IMB access will vary if an interrupt occurs between the last change of flow and the IMB access.

WORKAROUND:
1) Do not place instructions which might be fetched after an IMB access in external memory which matches A[10:24] = 0x60FF. In other words, the instruction address must not fall in the ranges: 0xZY0_7F80 to 0xZY0_7FFC, where ZY is in the external address space, Z=0x00 to 0xFF and Y is 3,7,B, or F. Or 2) Ensure that an external fetch is not pipelined with an IMB access by (a) running from internal memory, (b) running in serialized mode.

CDR_AR_985 Customer Erratum USIU.CDR1UBUS_05_0

USIU: Do not use ORx[EHTR] with Dual Mapping

DESCRIPTION:
When an access is matched through the Dual Mapping registers (DMBR/DMOR), extended hold time (from a previous access region) or Burst length (from the new access region) may cause execution of wrong code.

WORKAROUND:
1) Do not set ORx[EHTR] while a dual mapping region is enabled. Or: 2) Do not enable dual mapping if an extended hold time is required for any memory in the system.

CDR_AR_582 Customer Erratum **USIU.CDR1UBUS_05_0**

USIU:Under certain conditions, XFC stuck at 0V on power-on

DESCRIPTION:
In some systems, the PLL does not lock on power-up. This issue occurs on some board designs, and not on others. On some boards, the not lock on startup occur if 3V VDD has a DC offset of ~150mV. This also happens when the VDDSYN ramps up much slower than the VDDL/VDDi. This can also be repeated if the XFC pin is temporarily shorted to ground. If the PLL does not lock and limp mode is disabled, the part will never negate HRESET, and CLKOUT will run at the backup clock frequency. If the PLL does not lock and limp mode is enabled, the part will exit reset clocked by the backup clock. For the sensitivities mentioned above, additional assertions of the PORESET pin will not cause the PLL to lock, only cycling the power (and avoiding the condition) will resolve the issue.Observation of the XFC pin (with a high impedance probe), can distinguish that XFC is "stuck" at 0.

WORKAROUND:
First, make sure that the PLL and reset circuitry is correct: see CDR_AR_598. If XFC is at 0V, add a circuit (VDDsyn voltage divider set at 1.1-1.3V over temp/voltage connected to small signal diode connected to XFC) to ensure XFC does not drop below ~0.8V. It has been reported that some boards may be able to avoid this issue by avoiding a DC offset on VDDL and controlling the ramp rate of VDDsyn and KAPWR. Injection into 3V only pins may result in an offset on VDDL -- refer to CDR_AR_595.

CDR_AR_595 Customer Erratum **USIU.CDR1UBUS_05_0**

USIU: PLL will not lock on power-on, use limp mode and switch via software

DESCRIPTION:
In some systems, the PLL does not indicate lock on power-up. However, examination of the XFC with a high impedance, low capacitance probe indicates the VCO is at the correct frequency (20MHz XFC typically at ~1.6-1.8V). Additional assertions of the PORESET pin result in the PLL locking. This issue occurs on some board designs, and not on others. The PLL does seem to have some sensitivities related to power supplies. On some boards, the not lock on startup occur if 3V VDD has a DC offset of ~150mV. This also happens when the VDDSYN ramps up much slower than the VDDL/VDDi, or when KAPWR ramps faster. If the PLL does not lock and limp mode is disabled, the part will never negate HRESET, and CLKOUT will run at the backup clock frequency. If the PLL does not lock and limp mode is enabled, the part will exit reset clocked by the backup clock.

WORKAROUND:
Verify proper setup as noted in CDR_AR_598. To avoid system failure, always enable limp mode, allowing the system to boot using the backup clock even though lock is not yet indicated. After booting, switch from backup clock to PLL clock under software control. If EXTCLK is the clock input, use a higher frequency (15MHz or greater) and boot in 1:1 mode with limp mode enabled, or choose MODCK = 010, and overdrive the XTAL input from the oscillator output, and set EXTAL to a voltage of 0.5-0.6V. In this case, the oscillator output should be designed to meet an XTAL Vil of 300mv, with Iil= 1ma. Vih of 0.8v. To reduce fail to indicate lock, reduce the VDDL DC offset below 100mv by use of a discharge resistor. Ensure that current injected into 5V and 3V/5V pads is absorbed by circuitry attached to VDDH. While VDDL is off, avoid injecting current into 3V only pads since the PU3 circuit will cause an increase in VDDL.

CDR_AR_925 Customer Erratum **USIU.CDR1UBUS_05_0**

USIU: TEXP feature does not function when VDD supply is off

DESCRIPTION:
The TEXP function does not work if the main power supplies are powered down. Whenever VDD (low voltage supplies other than KAPWR and VDDSRAM) is powered down, hreset_b will be asserted by the chip and low power mode exited. The TEXP pin will never be asserted.

WORKAROUND:
The TEXP pin will never be asserted if VDD is powered down. Use an external counter to indicate the length of power down. As an alternate solution, put the part into Deep Sleep mode to reduce power consumption and leave the power supplies powered.

CDR_AR_610 Customer Erratum **USIU.CDR1UBUS_05_0**

Additional current on KAPWR when power is not applied

DESCRIPTION:
There is a leakage path between KAPWR and VDDi/VDDL. This results in additional KAPWR current when VDDi/VDDL is not powered. The amount of current varies, partially based upon the MODCK setting.

WORKAROUND:
Design KAPWR supply to handle additional 15ma of current, if KAPWR is powered while VDDi and VDDL are not powered. For MODCK=011, expect up to 9ma of leakage current, for MODCK=010, expect up to 7ma of leakage current.

CDR_AR_909 Customer Erratum **USIU.CDR1UBUS_05_0**

USIU: Do not assert cr_b to abort pending store reservation access

DESCRIPTION:
If an external cancel reservation (cr_b) is asserted then a pending store reservation may show on the external bus. This may occur with or without transfer start (ts_b), and will terminate after 1 clock. If the region is in the memory controller of the chip generating the store with reservation, then no chip-select or other memory controller attributes will assert on the bus, and the memory will not be altered.

WORKAROUND:
1) Do not assert cr_b; or 2) following assertion of cr_b, external logic must prevent the erroneous store with reservation bus cycle from altering memory, and must not assert ta_b to terminate the erroneous store with reservation bus cycle.

CDR_AR_910 Customer Erratum USIU.CDR1UBUS_05_0

USIU: PITRTC Clock may not work when SCCR[RTDIV] is 0

DESCRIPTION:
The RCPU RTC/PIT may not count in all operating conditions if the ratio of System clock to the PITRTC Clock is less than or equal to 4. This may happen if the SCCR[RTDIV] is set to 0 and either 1) the part is running on the limp clock, or 2) the PLPRCR[MF] = 0 and both the System PLL and the PITRTC Clock use the same clock source (EXTCLK or the crystal oscillator).

WORKAROUND:
Keep the System Clock to PITRTC clock frequency ratio greater than 4. This can be done the easiest by setting the SCCR[RTDIV] to a value of 1 (reset value).

CDR_AR_984 Customer Erratum USIU.CDR1UBUS_05_0

USIU: Setting of SCCR[EBDF] may slow execution of code

DESCRIPTION:
If the SCCR[EBDF] is greater than 0 and the RCPU is running not serialized, the USIU may issue external read bus cycles that are not complete. The TS_B will assert with an address, but without a chip select or STS_B assertion. These cycles will cause a delay in execution of code, but will not cause the code to fail. These cycles will self terminate in 1 to 3 clocks (depending on the external pull up strength).

WORKAROUND:
There are two possible workarounds: 1) In a program with critical timing, do not run from external memory with the SCCR[EBDF] set to a value greater than 0. Or 2) If external logic is used as a memory controller, define the logic to disregard these extra bus cycles.

CDR_AR_287 Customer Erratum USIU.CDR1UBUS_05_0

USIU: System to Time Base frequency ratio must be greater than 4

DESCRIPTION:
The Time Base and Decrementer may not count properly if the ratio of the System clock to Time Base Clock is 4 or less.

WORKAROUND:
Keep the ratio of the System Clock to the Time Base clock above 4. Always set SCCR[TBS] = 1 when running on the limp clock.

CDR_AR_479 Customer Erratum USIU.CDR1UBUS_05_0

USIU: The MEMC does not support external master burst cycles

DESCRIPTION:
The MTS function of the Memory Controller (MEMC) will not work properly to control external devices when an external master initiates a burst.

WORKAROUND:
Use external logic to control devices which can have burst accesses from multiple masters.

CDR_AR_679 Customer Erratum USIU.CDR1UBUS_05_0

USIU: In slave mode, do not use write slave accesses

DESCRIPTION:
In slave mode, a write data driven by the Core might be corrupted by the data driven by the external master.

WORKAROUND:
When the device is in SLAVE mode, do not use write accesses from the external master. Alternatively, use peripheral mode if write accesses from the external master are required.

CDR_AR_389 Customer Information USIU.CDR1UBUS_05_0

Little Endian modes are not supported

DESCRIPTION:
The little Endian modes are not functional.

WORKAROUND:
Do not activate little endian modes. The reference manual will be updated to remove all little endian mode references.

CDR_AR_442 Customer Information USIU.CDR1UBUS_05_0

Avoid loss of clock during HRESET

DESCRIPTION:
The chip may fail to switch to backup clock. This mode may occur if the input reference clock fails to toggle during hreset while switching from normal clock to backup clock. This condition may occur while switching from backup clock to normal clock (during hreset) if the PLL is not locked and there is no reference clock. In order to resume operation, the part may require the input reference clock to resume (for 1-2 more clocks) or for PORESET to be asserted.

WORKAROUND:
Avoid loss of the reference clock during hreset; ensure that the PLL is locked before switching to PLL clock. Do not enable reset upon loss of lock if limp mode is enabled, instead enable an change of lock interrupt by setting the COLIE bit (COLIR).

CDR_AR_594 Customer Information USIU.CDR1UBUS_05_0

USIU: Changing PLL MF to 1:1 mode can have 180 degree phase shift

DESCRIPTION:
After software changes MF from >1 to MF = 1, a 180 degree skew between EXTCLK and CLKOUT could occur.

WORKAROUND:
If synchronization between EXTCLK and CLKOUT is required, set MODCK to boot in 1:1 mode, and do not alter the MF bits to exit 1:1 mode.

CDR_AR_598 Customer Information USIU.CDR1UBUS_05_0

USIU: Ensure proper configuration for proper startup

DESCRIPTION:
In some systems, the PLL does not lock on power-up, or the system does not properly execute software out of reset. This issue occurs on some board designs, and not on others. Locking may be improved by board design and component selection, and can be resolved by paying attention to the design and setup, and ensuring that the PLL and Oscillator components are correct and as noise free as possible.

WORKAROUND:
First, make sure that the PLL and reset circuitry is correct: ensure that the PLL components are properly selected and that the PLL power (VDDSYN) is not noisy. Refer to appendix E of the users manual, "Clock and Board Guidelines". Verify that the XFC capacitor is connected to VDDSYN. Validate that the TRST pin is asserted upon power-up. Do not connect TRST to HRESET or SRESET. Validate that all power supplies are stable and all MODCK pins are at the correct levels in time for the PLL and Oscillator to be stable prior to PORESET rising above VIL. Verify that the proper reset configuration word is used. Validate the reset and post reset pin state for each pin controlled by the reset configuration word, and ensure there is not a conflict with an external driver. Preferably use the internal reset configuration word. If using an external reset configuration word, do not rely on the internal pull-downs to operate (refer to CDR_AR_454) and ensure that RSTCONF is asserted until SRESET is negated. After the part exits reset with the system running via the backup clock, validate the clock control registers settings and the PLL status. If the PLL is slow on locking, or the register settings indicate the MODCK pins are incorrect, address the board issues listed above. To avoid risk of system failure for no start, enable limp mode, allowing the system to boot using the backup clock even though lock is not yet indicated. After booting, switch from backup clock to PLL clock under software control after the PLL has gained lock.

CDR_AR_687 Customer Information USIU.CDR1UBUS_05_0

USIU: Program reserved bits in PDMCR to preserve compatibility

DESCRIPTION:
Future revisions of the PDMCR will have additional bits to control enabling and disabling of pad pull-up / pull-down resistors. Software should be written so that it is compatible with these changes. In this revision, PDMCR[8] (TPRDS) does not change the function of the TPU T2CLK pull-up resistors -- the pull-ups remain enabled.

WORKAROUND:
To ensure identical control in future revisions, when programming the PDMCR. PDMCR[8] should remain cleared. PDMCR[9:13] should be programmed to the same value as PRDS (PDMCR[6]). PDMCR[16:17] should be programmed to the same value as SPRDS (PDMCR[7]). The future function of PDMCR[14:15] has not been determined, and should be programmed to 0. For this revision, software should ignore any the read values of PDMCR[8:15].