Customer Errata and Information Sheet Transportation Systems Group

Part: MPC555.M Mask Set: 00K62N. Report generated 30 March 2000 .

TRANSPORTATION SYSTEMS GROUP
CUSTOMER ERRATA AND INFORMATION SHEET
Part: MPC555.M Mask Set: 00K62N
General Business Use
Report Generated: Thu Mar 30, 2000, 13:50:59

MPC555.M 00K62N Modules
Current Module Revision
BBC.CDR1UBUS_04_0
CMF.192KB_CDR1UBUS_05_0
CMF.256KA_CDR1UBUS_05_0
DPTRAM.6K_CDR1IMB3_03_0
JTAG.CDR1_01_0
L2U.CDR1LBUSUBUS_02_0
LRAM.10KA_CDR1LBUS_02_0B
LRAM.16KB_CDR1LBUS_02_0B
MIOS1.CDR1IMB3_04_2
PKPADRING.555_CDR1_03_0
RCPU.CDR1LBUSIBUS_13_0
QADC64.CDR1IMB3_03_0
QSMCM.CDR1IMB3_02_0
TOUCAN.CDR1IMB3_04_0
TPU3.CDR3IMB3_03_0
UIMB.CDR1UBUSIMB3_03_0
USIU.CDR1UBUS_06_0

ERRATA AND INFORMATION SUMMARY

DETAILED ERRATA DESCRIPTIONS

CDR_AR_678 **Customer Information** **MPC555.M**

Additional current on KAPWR

DESCRIPTION:
KAPWR current exceeds the initial design targets. During operation, KAPWR may be 8ma. Currents during power-down modes have not been fully characterized, and should be assumed to be the same value.

WORKAROUND:
Design KAPWR supply to handle the additional current. Characterize the current consumption in the final application board.

CDR_AR_697 **Customer Information** **MPC555.M**

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.M**

New Features on MPC555 mask revision J76N an 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.M**

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_597 **Customer Information** **MPC555.M**

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 30. 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.

CDR_AR_593 **Customer Information** **MPC555.M**

MASKNUM field in USIU is 0x40

DESCRIPTION:
MASKNUM field in USIU has been changed to 0x40, and will change on future revisions.

WORKAROUND:
Modify software to expect new value (0x40) for the MASKNUM field.

CDR_AR_750 **Customer Information** **MPC555.M**

Pad hardware and software changes on revision K62N

DESCRIPTION:
The user should be aware that fields in the USIU PDMCR have changed. Bit 8 was previously a reserved bit. It now holds the control for the pull devices on the t2clk pins (previously, they were always enabled). Other bits in the PDMCR are still reserved for additional control on future devices.The default value for the ENGCLK was previously defined as system clock divided by 2 (20Mhz if system frequency is 40mhz). The ENGCLK default is now set to the system clock divided by 64. The default is 625KHz if the system frequency is at 40Mhz.The ENGCLK pad driver is now sized to drive loads of 25pf or 50pf, selectable by software. The CLKOUT pad driver is now sized to drive loads of 30pf or 90pf, selectable by software.

WORKAROUND:
PDMCR bit 8 of should be written to a logic "0" if the user wishes for the t2clk pull devices to behave as on previous versions of the chip. To have best software compatibility with future devices, 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 the read values of PDMCR[9:15].If greater frequencies are required on ENGCLK, the user must write to SCCR in the USIU (see ENGDIV bits).The drive strength of ENGCLK and CLKOUT should be selected based upon the external load, along with EMC considerations.

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_440 **Customer Information** **RCPU.CDR1LBUSIBUS_13_0**

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 IMULDIV control area. This contention may only exist prior to the execution of any IMULDIV instruction.

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

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_563 **Customer Erratum** **QADC64.CDR1IMB3_03_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_754 **Customer Erratum** **QADC64.CDR1IMB3_03_0**

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) If queue2 is used and queue1 is in external gated mode, set queue2 to single scan mode.

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 Erratum** **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_627 **Customer Information** **TPU3.CDR3IMB3_03_0**

TPU: (Microcode) Add neg_mrl with write_mer and end_of_phase

DESCRIPTION:
Wrong generation of 50% d.c. caused when we have 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 content of the match register (which is 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 only u-instruction. In the following u-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.CDR3IMB3_03_0**

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_479 **Customer Erratum** **USIU.CDR1UBUS_06_0**

USIU: The MEMC does not support external master burst cycles

DESCRIPTION:
The MTS function 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_389 **Customer Information** **USIU.CDR1UBUS_06_0**

Little Endian modes are not supported

DESCRIPTION:
The little Endian modes are not functional.

WORKAROUND:
Do not activate little endian modes.

CDR_AR_442 **Customer Information** **USIU.CDR1UBUS_06_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_06_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_06_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_L pin is asserted upon power-up. Do not connect TRST_L to HRESET_L or SRESET_L. 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_L 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.