**Advanced CUSTOMER ERRATA AND INFORMATION SHEET Page 1**
**MCU Part: 68332.S Mask Set: Input truncated to 4 characters**
**Division Report Generated: Aug 27, 96 17:36**

| 68332.S Input truncated to 4 characters |
|---|
| Current Module Revision |
| CPU32.12 |
| QSM.9A |
| SIM.12_3 |
| STDPORT.11 |
| TPU.6_3 |
| TPUSRAM2K.6A_2 |
| VCO.7 |

**MODULAR_AR_12 Customer Information 68332.S**

**DESCRIPTION:**

After power-up (Vdd > Vdd min.) of the MCU, input/output and output-only port pins on modules other than the integration module may be in an indeterminate state for up to 15 ms (depends on ramp up conditions). Input/output pins on these modules may be in output mode (instead of high impedance) for a short time, which may create a conflict with external drive logic. (Previously 68332:005)

**WORKAROUND:**

If a known state is required on these pins, before the 15 ms port initialization period, external reset control logic must condition these lines.

**MODULAR_AR_220 Customer Erratum CPU32.12**

**DESCRIPTION:**

When there is a bus error on a released write cycle and the bus cycle is re-run by the RTE instruction immediately before a TAS instruction, then the re-run cycle is not a write, but is a read instead. Most applications do not experience this problem since software is not usually designed to interpret and recover from bus errors. The common reaction to a bus error is to abort the program flow, report the error occurrence, and then attempt to restart the application. Applications designed that way need not be concerned with this problem. (Previously CPU32:061).

**WORKAROUND:**

None

**Advanced CUSTOMER ERRATA AND INFORMATION SHEET Page 2**
**MCU Part: 68332.S Mask Set: Input truncated to 4 characters**

**Division Report Generated:** Aug 27, 96 17:36

**MODULAR_AR_213 Customer Erratum CPU32.12**

**DESCRIPTION:**

When there is a bus error on the second word of a released longword write cycle immediately before a TAS instruction, then the RR and RM bits in the special status word (SSW) are incorrect. The RR and RM bits represent the TAS cycle instead of the released write cycle. The bus cycle which had the error will not be re-run on the return from exception. Most applications do not experience this problem since software is not usually designed to interpret and recover from bus errors. The common reaction to a bus error is to abort the program flow, report the error occurrence, and then attempt to restart the application. Applications designed that way need not be concerned with this problem. (Previously CPU32:053).

**WORKAROUND:**

Sometimes it is not possible to determine this fault condition from the stack and program contents. When the released write happens to be a byte write to supervisor space, the SSW looks exactly like a bus error on the TAS write cycle, and if the released write address is the same as the TAS address, the fault address is not different between the two cases. The bus error exception handler may resolve the situation by: (a) looking at the size, R/W, and function code information, or when that fails; (b) examining the faulted address and determining whether it is the same as the address that would be generated by the TAS instruction at the return program counter. In the case of a released write bus error, then setting RR and clearing RM should be the correct action (with RTE), after resolving the original cause of the bus error.

**MODULAR_AR_222 Customer Erratum CPU32.12**

**DESCRIPTION:**

The CPU does not properly execute the DBcc instruction under the following condition. If the DBcc is about to be fetched the second time, and loop mode is possible, and the bus is slow (or to an 8-bit memory) so that the DBcc offset is NOT fetched the second time, and the looped instruction gets a released write bus error, then the DBcc instruction is not executed. The return PC points to the looped instruction, which will be executed once too often. (was CPU32:063).

**WORKAROUND:**

If loop mode is possible, force the software out of the loop mode by placing a NOP prior to the DBcc instruction.

**Advanced CUSTOMER ERRATA AND INFORMATION SHEET Page 3**
**MCU Part: 68332.S Mask Set: Input truncated to 4 characters**
**Division Report Generated: Aug 27, 96 17:36**

**MODULAR_AR_224 Customer Erratum CPU32.12**

**DESCRIPTION:**

When a BERR terminates an operand write cycle, the CPU32 enters BERR exception right after the BERR. However, in the stack, the next instruction addr. shows the wrong address. To cause this condition, the bus has to be slow (at least 6 clock cycle). This bug results in the CPU running an operand write cycle one more time than it should (was CPU32:066).

**WORKAROUND:**

Ensure that loop mode is not entered when there is a possibility of a released write bus error.

**MODULAR_AR_812 Customer Erratum CPU32.12**

**DESCRIPTION:**

To acknowledge a breakpoint, the CPU32 performs a read from CPU space $0 at address $1E. If the cycle terminates normally, instruction execution continues with the next instruction, as if no breakpoint request occurred. If the bus cycle is terminated by BERR, the CPU begin exception processing. If the breakpoint occurred while the CPU was in loop mode, and the breakpoint acknowledge cycle terminates normally, the CPU does not continue with the next instruction execution. (Previously CPU32:071).

**WORKAROUND:**

Always terminate breakpoint acknowledge cycle with BERR (bus Error)

**MODULAR_AR_206 Customer Erratum CPU32.12**

**DESCRIPTION:**

If the internal bus is arbitrated away just before the last cycle of a MOVEM to memory, and the MOVEM is followed by other instructions which write to external memory (known problem instructions are: MOVE Rn,-(An) and LINK), then the data written by the last cycle of the MOVEM will be the same data as for the subsequent instruction (ex: MOVE). In other words, the next instruction (MOVE) started execution too early. Internal bus arbitration can occur in the

following situations:1. An external bus request occurs with the SHEN[1:0] = %11 or2. If an on-chip alternate master (ex: DMA) module takes mastership of the IMB or3. An external bus request occurs when Factory Test (Slave) mode is enabled.(Previously CPU32:051).

**WORKAROUND:**

1. If external bus requests may occur, ensure SHEN[1:0] are anything except %11.2. If an alternate master (DMA) is present on-chip, and operating, arrange instruction sequences to prevent the occurrence of MOVEM followed by another instruction which writes to external memory (MOVE Rn,-(An) or LINK). It is recommended to place a NOP after the MOVEM.3. Factory Test (Slave) mode should not be used.



**MODULAR_AR_216 Customer Erratum CPU32.12**

**DESCRIPTION:**

When a jump or branch to an odd address occurs and there is a pending bus error, the address error exception is incorrectly taken and the bus error exception is not taken. (Previously CPU32:056)

**WORKAROUND:**

Do not jump or branch to odd addresses when bus errors can occur.

**MODULAR_AR_210 Customer Erratum CPU32.12**

**DESCRIPTION:**

If an IACK cycle (with AVEC asserted) is terminated by a normal BERR, a Spurious Interrupt exception will be taken as expected. If the BERR of the IACK cycle is a LATE BUS ERROR (BERR asserted after AVEC), then a Type 4 bus error exception (Faults during exception processing) will be taken, in this case the Status Register in the stack frame will be corrupted and there can be no recovery from the exception. (Previously CPU32:045).

**WORKAROUND:**

Do not allow LATE BERR during IACK cycles that are terminated with AVEC.

**MODULAR_AR_221 Customer Erratum CPU32.12**

**DESCRIPTION:**

When exiting BDM, FREEZE is negated after the DSI/IFETCH signal turns around from a input to an output. This means that an external development system could be driving DSI while the MCU is still driving IFETCH. (Previously CPU32:062).

**WORKAROUND:**

Provide sufficient isolation externally if the contention may cause system problems.

**MODULAR_AR_975 Customer Information CPU32.12**

**DESCRIPTION:**

When a spurious interrupt awakens the MCU from LPSTOP, the spurious interrupt handler is not called, and the CPU sits idle until another IRQ is detected. The effect is the same as if the STOP instruction was actually executed, from the point that the spurious interrupt occurs, instead of the LPSTOP instruction.

**WORKAROUND:**

If power consumption is important and LPSTOP is used, do not allow any spurious interrupts to occur during the LPSTOP state.



**MODULAR_AR_218 Customer Information CPU32.12**

**DESCRIPTION:**

If IRQ7 is asserted and is released just before the IACK cycle and is then reasserted, only one IRQ7 interrupt is taken. (Previously CPU32:058).

**WORKAROUND:**

NONE

**MODULAR_AR_1018 Customer Information CPU32.12**

**DESCRIPTION:**

The stack frame is incorrect when bus error occurs (internal or external BERR asserted) on a write cycle immediately followed by the ’TRAP #n’ instruction.

**WORKAROUND:**

Avoid bus error occurrence when “TRAP #n” is run. If bus error cannot be avoided, insert “NOP” before running “TRAP #n”, or modify the return address in the bus error exception frame to direct the CPU back to the correct flow.

**MODULAR_AR_1020 Customer Information CPU32.12**

**DESCRIPTION:**

Incorrect operation occurs as a result of the following sequence of conditions: A bus error or address error condition occurs during an operand cycle of a MOVEM instruction. After the exception processing, the CPU re-fetches the MOVEM instruction. If the re-fetch cycle of the MOVEM instruction is terminated by a bus error condition, an error condition occurs.

**WORKAROUND:**

Do not allow a bus error condition to occur for a fetch cycle of the MOVEM instruction, if that fetch occurs after exception processing of a bus error or address error on a MOVEM operand cycle.

**MODULAR_AR_226 Customer Information CPU32.12**

**DESCRIPTION:**

If the AVEC and DSACK signals are asserted simultaneously to terminate an IACK cycle, then the DSACK signal has higher priority. (Previously CPU32:069).

**WORKAROUND:**

None



**MODULAR_AR_818 Customer Erratum SIM.12_3**

**DESCRIPTION:**

Under the following conditions, showcycles will be truncated: showcycles and external arbitration are enabled, and BR is asserted immediately before the clock edge from which DS asserts for a show cycle. The data bus drive time for the show cycle will overlap the front end of the alternate master bus tenure by one clock (BG is asserted during the time showcycle data is driven).

**WORKAROUND:**

1. Disable show cycles when alternate master bus activity is possible2. Delay BG assertion to the system by one clock, or delay the alternate master from driving the date bus for one clock after BG asserts.

**MODULAR_AR_664 Customer Erratum SIM.12_3**

**DESCRIPTION:**

Unusual system operation may occur when bus arbitration is used in combination with additional system configuration settings and timing. As an example of the behavior, chip selects may assert while the external bus is granted away (if they are programmed to respond to the interrupt stack addresses) and interrupt stack may get corrupted. Reset is the only way to recover once this occurs.

**Conditions to initiate:**
1. SHEN bits of MCR set to %00 or %10.
2. BR is asserted coincident with the AS of an IACK cycle.(BR assertion is within the range of before and after 1 clockrelative to the falling edge of CLKOUT when AS asserts.)
3. IACK cycle is terminated with external AVEC.(Previously IM:092).

**WORKAROUND:**

1. Use SHEN=%11 to prevent the IMB from running cycles while the external bus is grantedaway.or2. Do not assert BR coincident with AS of an IACK.or3. Do not use external AVEC. Use a chip select to assert internal AVEC for external interrupts.

**MODULAR_AR_667 Customer Erratum SIM.12_3**

**DESCRIPTION:**

Under certain conditions a masked interrupt may occur with an incorrect level. The conditions are: an external unmasked interrupt must occur coincident with an internal masked periodic interrupt (PIT). Also, previous to the above interrupts, an external interrupt line at the same level as the masked PIT interrupt must have been asserted and remain asserted. Example sequence to cause problem:1. Set CPU interrupt mask to 5;2. Set PIT to level 2;3. Hold IRQ2 line low.4. Assert valid IRQ interrupts (asserting IRQ 6 or IRQ7) the PIT exception is taken if pending.(Previously IM:095)

**WORKAROUND:**

Do not allow matching levels on PIT and external pin.

**MODULAR_AR_331 Customer Erratum SIM.12_3**

**DESCRIPTION:**

The RESET assertion time specification (#77) is 4 clocks (tcyc) minimum. However, the current version of this module requires RESET to be asserted until the current bus cycle in progress completes. (Previously IM:139)

**WORKAROUND:**

Assert the RESET pin for 2 clock cycles longer than the present timeout period of the bus monitor (BMT field in SYPCR register). This will result in an internal reset, independent of other system conditions (Bus Monitor does not need to be enabled).

**MODULAR_AR_661 Customer Erratum SIM.12_3**

**DESCRIPTION:**

At power-up, integration module I/O pins should initialize to the high impedance state. The following pins may, however, drive as outputs until the first CLKOUT edge occurs to initialize the internal logic into the high impedance state. Port E, D[15:0] and HALT (open Drain). (Previously IM:086).

**WORKAROUND:**

If external conflicts result in system problems on these pins, isolate these pins from external devices using a series resistor or buffer on the offending pin.

**MODULAR_AR_635 Customer Erratum SIM.12_3**

**DESCRIPTION:**

If PIT is used in either PLL mode or external clock mode with the PIT prescalar enabled (PTP bit in PITR register set), the PIT clock is 32 KHz (crystal or EXTAL/512 if in external clock mode). If the PITM field of the PITR register is written to zero, followed immediately by a nonzero value, and then LPSTOP is entered before 1 full period of the 32 KHz PIT clock source, the PIT timer will never decrement which may prevent exiting LPSTOP using the PIT timeout. (Previously IM:082).

**WORKAROUND:**

If the user wishes to stop the PIT and restart with a new time value by writing zero followed by the new value to the PITM field, the user should delay entering LPSTOP by at least one 32kHz PIT clock period after writing the new value. Alternatively, the user can update the PITM field with the new value without first stopping the counter (writing zero to the PITM field). In this case, the timer will continue counting down to the original time value before updating to the new value.



**MODULAR_AR_639 Customer Erratum SIM.12_3**

**DESCRIPTION:**

At power-up, chip select pins may drive low (asserted) until the first CLKOUT edge occurs. False writes may result on power up if the R/W pin does not have a pull-up resistor. The R/W line is in a high impedance state at power up and while reset is asserted. (Previously IM:085).

**WORKAROUND:**

Insure external writes cannot occur on power up by pulling up the R/W pin.

**MODULAR_AR_876 Customer Information SIM.12_3**

**DESCRIPTION:**

When the internal PLL clock system is not used (MODCK=0) at RESET negation) then the following behavior may occur. During power down, if the external clock degrades such that it no longer meets the AC Timing Specification for the External Clock Input High/Low Time (tXCHL), then the Input/Output and Output-only pins of the integration module and other modules may become active. Assertion of the external RESET pin under these conditions does not guarantee the level on the RESET pin will be internally recognized and the internal RESET signal may be negated under these anomalous conditions. The internal RESET signal is used to hold the Input/Output and Output-only pins in their respective high impedance mode. If the on-chip PLL is used for the clock source there is no problem as the PLL will meet the specifications to the minimum Vdd.

**WORKAROUND:**

If an External Clock is used, then insure that the External Clock signal does not degrade and violate the specifications as power goes down. Alternately, protect external devices that may be damaged (ex: non-volatile memories).

**MODULAR_AR_930 Customer Information SIM.12_3**

**DESCRIPTION:**

Several conditions combined may introduce apparent Periodic Interrupt Timer (PIT) clock errors. The clock error occurs if LPSTOP mode is entered and exited periodically using the PIT, and the system clock is set to minimum (PLL control bits in SYNCR register set to: Y=0, and W =0) prior to entry of LPSTOP mode and system clock set to maximum at LPSTOP exit (STSIM=0, VCO off in LPSTOP, therefore the PLL must re-lock). Also, on exiting from LPSTOP, the CPU will be held off of the bus until the PLL is re-locked. Variations in the PIT clock period may appear as the PIT counter missing clocks (the PIT is clocked by the EXTAL reference clock in LPSTOP, if STSIM = 0). During normal operation (not in LPSTOP) the PIT counter clock source (EXTAL) is synchronized by logic to the system clock (CLKOUT). The combination of PLL re-lock time, low frequency clock source (too near the internal PLL reference), and the synchronization, results in this behavior. (Previously IM:097 and Issue #653).

**WORKAROUND:**

Restrict the minimum PLL frequency to at least 2 times the minimum possible reference frequency (PLL control bits in SYNCR register set to: Y=0, and W =%01) or higher. This restriction applies if the system clock is toggled from a low frequency prior to LPSTOP entry (if STSIM = 0) and back to maximum on exit of LPSTOP periodically using the PIT as the controlling source. The problem is only seen when switching the PLL clock to a frequency that is too close to the reference (EXTAL) clock frequency.

**MODULAR_AR_662 Customer Information SIM.12_3**

**DESCRIPTION:**

If E-Clock synchronized chip selects (MODE=1) are used in combination with peripherals which can retry bus cycles an addressing problem may occur. Chip selects can be set up for synchronous E-clock support, configured for an 8 bit port and used with a peripheral connected to the upper data bus (D15:8 and BYTE = upper). A problem occurs during a word access of the 8 bit peripheral, in combination with a retry (terminate cycle with BERR and HALT) being requested by the device on the second cycle of the access. Under this condition, the second cycle will be retried, but [A0] will be incorrect. (Previously IM:088)

**WORKAROUND:**

Do not attempt retry of chip selects supporting synchronous E-clock cycles on 8 bit ports.

**Advanced CUSTOMER ERRATA AND INFORMATION SHEET Page 10**

**MCU Part: 68332.S Mask Set: Input truncated to 4 characters**
**Division Report Generated: Aug 27, 96 17:36**

**MODULAR_AR_908 Customer Information SIM.12_3**

**DESCRIPTION:**

The documentation for the state of the RMC/PE3, SIZ[1:0]/PE[7:6] and DS/PE4 pins is inconsistent between the users manuals (MC68...USM) and the SIM/SCIM reference manuals (Module..RM). The users manuals indicate the pins are in a high impedance state while RESET is asserted, which is correct. The SCIM/SIM manuals indicate the pin state is determined by the data bus configuration while RESET is asserted, which is not correct.

**WORKAROUND:**

Refer to documentation in the users manuals for the RESET state of these pins.

**MODULAR_AR_658 Customer Information SIM.12_3**

**DESCRIPTION:**

The loss of clock reference feature is not supported and may not function. Disregard bit position 4 in the SYNCR register (previously the SLIMP bit), this bit is now reserved. Insure that the bit position 2 (previously the RSTEN bit) in the SYNCR register is always written to it’s RESET state of %0. (This was previously IM:077).

**WORKAROUND:**

Do not rely on the loss of clock LIMP mode feature.

**Advanced CUSTOMER ERRATA AND INFORMATION SHEET Page 11**
**MCU Part: 68332.S Mask Set: Input truncated to 4 characters**
**Division Report Generated: Aug 27, 96 17:36**

**MODULAR_AR_800 Customer Information VCO.7**

**DESCRIPTION:**

In some Phase Lock Loop (PLL) documentation a three component filter from XFC to VDDSYN is recommended (18K resistor in series with 0.1 uF capacitor between VDDSYN and XFC, the series combination in parallel with a 3300 pF capacitor for a loop multiplier of N = 512). It has been determined with this three component filter, in the presence of external leakage (in excess of that provided by ~50 M Ohm) on the XFC pin, may result in the MCU not exiting RESET at power up. During this condition, the output frequency on CLKOUT is at the target value, but the

PLL lock detect logic does not detect lock and continues to cause RESET assertion. Versions of the integration module that are configured for either a slow or fast (Typ. 32.768 kHz or 4.194 MHz) crystal source option use the same filter component values since the internal reference frequency is always slow (ex: 32.768 kHz). Also, leakage from the XFC pin must not be in excess of that provided by a 15 M Ohm resistor to meet PLL jitter specifications (with 0.1 uF XFC filter, refer to Electrical Characteristics section of device users's manual). If the PLL is not enabled (MODCK=0 at RESET) then the new XFC filter is not required. (Previously IM:179 and VCO:051).

**WORKAROUND:**

Do not use the three component filter on XFC. Use the originally documented filter (single 0.1 uF capacitor from the XFC pin to the VDDSYN supply pin).

**MODULAR_AR_986 Customer Information VCO.7**

**DESCRIPTION:**

The "PLL Lock Time" (tlpll) specification is documented as 20 ms. This value applies to the time for the PLL to lock after changing the W or Y bits in the synthesizer control register (SYNCR) while the PLL is running, and to the period required for the PLL to lock after LPSTOP is exited. This specification does not apply at warm start-up (with VDDSYN applied and crystal stable, followed by VDD power application). The warm start-up period is a maximum of 50 ms. (This was previously IM:098, VCO:059, AR_806 and in some cases AR_987).

**WORKAROUND:**

Use 50 ms for "PLL Lock Time" (tlpll) specification.

Revision 0.0, 20 NOV 96

last update: Mar 26 1998