*Mask Set Errata 1*

# MMC2001 32-Bit Microcontroller Unit

## INTRODUCTION

This mask set errata provides information concerning the MMC2001's interval mode in the SPI module, the SPI_EN pin, the instruction set, debug mode, and electrical specifications. This information is applicable to these specific MMC2001 MCU mask set devices:

- 1J56W
- 3J56W
- 0K22K

## MCU DEVICE MASK SET IDENTIFICATION

The mask set is identified by a 5-character code consisting of a version number, a letter, two numerical digits, and a letter, for example 3J74Y. Slight variations to the mask set identification code may result in an altered version number, for example 4J74Y.

## MCU DEVICE DATE CODES

Device markings indicate the week of manufacture and the mask set used. The data is coded as four numerical digits where the first two digits indicate the year and the last two digits indicate the work week. For instance, the date code "9915" indicates the 15th week of the year 1999.

## MCU DEVICE PART NUMBER PREFIXES

Some MCU samples and devices are marked with an SC or XC prefix. An SC prefix denotes special/custom device. An XC prefix denotes that the device is tested but is not fully characterized or qualified over the full range of normal manufacturing

*When contacting a Motorola representative for assistance, please have the MCU device mask set and date code information available.*



MSE Published Date: 4/30/01

process variations. After full characterization and qualification, devices will be marked with the MC prefix.

## SPI_EN PIN

The SPI_EN pin can be toggled only under these conditions:

- MSTR = 1, SNS = 1, DRV = 0

or

- MSTR = 1, SNS = 0, DRV = 1, pullup on SPI_EN pin

**WORKAROUND:**

To put an active LOW on the SPI_EN pin, add a light pullup externally. To put an active HIGH on the SPI_EN pin, add a pull-down externally.

## INTERVAL MODE SPI MODULE

When writing to the SPICR register, the data that is loaded into the internal counter is incorrect for each ISPI interval. However, the register which latches the data written functions properly. Reading back the register returns the correct value.

- Writing 0x20 (bit 5) actually loads 0x60 (bits 5 and 6) into the interval counter.
- Writing 0x40 (bit 6) actually loads 0x80 (bit 7 only) into the interval counter.
- Writing 0x80 (bit 7) actually loads nothing into the interval counter.
- Writing 0x400 (bit 10) actually loads 0xC00 (bits 10 and 11) into the interval counter.
- Writing 0x800 (bit 11) actually loads nothing into the interval counter.

| Bit Number | Bit Intended to Set | Actual Bit Set |
|---|---|---|
| 0 (0x01) | 0 | 0 |
| 1 (0x02) | 1 | 1 |
| 2 (0x04) | 2 | 2 |
| 3 (0x08) | 3 | 3 |
| 4 (0x10) | 4 | 4 |
| 5 (0x20) | 5 | 5 and 6 |
| 6 (0x40) | 6 | 7 |
| 7 (0x80) | 7 | None |
| 8 (0x100) | 8 | 8 |
| 9 (0x200) | 9 | 9 |
| 10 (0x400) | 10 | 10 and 11 |
| 11 (0x800) | 11 | None |
| 12 (0x1000) | 12 | 12 |

## Jump Indirect (JMPI)

The misaligned JMPI with pending interrupt (INT) generates an incorrect EPC shadow register result. The EPC shadow register should point to the JMPI and not at the destination of the jump.

**WORKAROUND:**

None

## Load (LD.[B,H])

LD.[B,H] followed by a jump (JMP) with a data dependency will not zero extend the JMP address correctly.

**WORKAROUND:**

The compiler should not generate this code. If it does, put a synchronize (SYNC) instruction between the LC.[B,H] and the JMP.

## Execute Exception

On an execute exception, with a fast interrupt (FINT) and a marked instruction breakpoint, the fast interrupt enable (FE) bit is not cleared.

**WORKAROUND:**

None

## Breakpoint (BKPT)

On a data side breakpoint followed by a multicycle instruction, the breakpoint is not decoded.

**WORKAROUND:**

None

## Fast Interrupt (FINT)

On an FINT with the exception enable (EE) bit cleared, any exception in conjunction with the FINT will cause an unrecoverable exception instead of an FINT. On the unrecoverable exception, the FE bit will get cleared and FINT will not be taken.

One exception to this is a data breakpoint (DBKPT), which will cause a BKPT to be taken instead of an unrecoverable.

**WORKAROUND:**

None

## DEBUG MODE

### I-Side Breakpoint (IBKPT)

On an IBKPT, an FINT, a normal interrupt (INT) or a trace exception occurs with a debug request assertion from the debug block. The FINT, INT, or trace exception is taken instead of the IBKPT.

**WORKAROUND:**

None

### Multiply/Divide

If, at the end of a multiply or a divide, there is still an outstanding instruction fetch and a debug request is made, the program counter (PC) may be inadvertently incremented.

**WORKAROUND:**

None

If, at the end of a multiply or a divide with wait stated memory and a change of flow (JMP/BR) is being executed and a debug is requested, we decode an inadvert instruction upon exiting debug mode, and the PC points to change of destination - 2.

**WORKAROUND:**

Perform a SYNC instruction after multiply.

### Memory Breakpoint Occurrence (MBO) Bit

The MBO bit is being set on memory breakpoint occurrences regardless of how debug mode is entered.

**WORKAROUND:**

None

### Software Debug Occurrence (SWO) Bit

The SWO bit is inadvertently being set upon entry into debug mode. The on-chip emulation (OnCE) status register bits are not being cleared correctly when exiting debug mode with the go (GO) and exit (EX) bits set in the OnCE command register (OCMR).

**WORKAROUND:**

None

## LDQ Breakpoint

If a load register quadrant (LDQ) instruction is executed with the first data fetch breakpointed and the TEA asserted on second data fetch, the machine will not go into debug mode, an access error exception will be taken instead of entering debug mode. The EPC shadow register is indeterminant.

**WORKAROUND:**

None

## Wait State Memories

On wait stated memories, when load quadrant (LDQ), store register quadrant (STQ), load multiple registers (LDM), store multiple registers (STM), load (LD), and store (ST) are executing with simultaneous DBKPT and TEA asserted on second data fetch, the PC is not incremented correctly. When coming out of debug mode, the EPC points to the instruction before the data transaction, which is incorrect.

**WORKAROUND:**

None

## RTE/RFI Trace

If debug mode is entered immediately after a return from exception (RTE) or return from fast interrupt (RFI) and all instructions are being traced, the RTE or RFI will be traced after debug mode is exited.

**WORKAROUND:**

None

# ELECTRICAL SPECIFICATIONS

The power-on reset (POR) circuit draws 3 μA during normal mode. Total standby current is 8 μA.

Electrostatic discharge (ESD) protection fails for 2.0-kV human body model on the $V_{BATT}$ pin.



Additional mask set erratas can be found on the World Wide Web at http://mcu.motsps.com/documentation.