

# **ERRATA SHEET**

**Date:** Jun 17, 2003  
**Document Release:** Version 1.1  
**Device Affected:** P87C51Mx2/02

This errata sheet describes both the functional deviations and any deviations from the electrical specifications known at the release date of this document.

2003 Jun 17



**Part identification:**

P87C51Mx2/02 microcontroller is present as 44 pin PLCC OTP or 68 pin PLCC bondout chip.

44 pin MC2/02 OTP chips can be identified with this marking on the top/bottom of the part:

P87C51MC2-G  
K2X066-W16  
0302

or

P87C51MC2-G  
K2X066-W16  
0303

68 pin MC2/02 OTP chips can be identified with this marking on the top/bottom of the part:

P87C51MC2-G  
K2X066-W16  
0302

or

P87C51MC2-G  
K2X066-W16  
0303

## Functional Deviations of P87C51Mx2/02

**UART.1:** In MODE 0 the first outputted data bit on RxD (Data Out) line is set after the first falling edge is generated on TxD (Shift Clock) line. Following 7 data bits are transmitted without problems. Reception of data is not affected.

**Introduction:** MODE 0 is one of four available operating modes for UART0 and UART1 in P87C51Mx2/02. In this mode serial data enters and exits through RxD line, while TxD outputs the shift clock. Only 8 bits are transmitted or received, sending LSB data bit first.

**Problem:** When UART is in MODE0, after valid content of data bit is set on RxD line, falling edge is generated on TxD line. This way transmitter indicates to the receiver when to read/sample data bit. This mechanism fails for the very first data bit (LSB) sent from UART: valid data is available only after the first falling edge of serial clock on TxD line occurs. The rest 7 data bits are transmitted according to the specification: first the data is set, then sampling edge is generated.

**Workaround:** None. UART affected with this problem can be utilized when only 7 bits are sufficient for the transmission (in this case LSB in the data written into UART's buffer register can not be used for data-transfer). When UART is configured to read data in MODE0, it can receive all 8 bits from the data byte. Problems with transmission do not affect receiving mechanism.

**CORE.1:** Setting EXTRAM=1 in AUXR and then using universal pointers to access data in EDATA memory space (7F: 0100-7F:FFFF), will result in incorrect readings.

**Introduction:** EXTRAM bit in AUXR register is used to control XDATA memory access: when EXTRAM=0 microcontroller will access off-chip XDATA only when data's address is out of on-chip available XRAM. Having EXTRAM=1 will result in chip's access to off-chip XDATA by default. Use of EMOV instruction when accessing EDATA is not dependent on EXTRAM's content.

**Problem:** With EXTRAM=1 EMOV instruction accessing location in EDATA memory space will act as if it is accessing location in XDATA memory.

**Workaround:** EMOV instruction must not be used for accessing anything else but real EXTRAM in systems that have EXTRAM=1. in this case DATA/IDATA can be accessed anytime using MOV instruction with no problems. If systems with EXTRAM=0, EMOV instruction can be used with no restrictions.