# DEVICE ERRATA

August 2, 1996

# 68LC302 INTEGRATED MULTIPROTOCOL PROCESSOR DEVICES

**This document covers the XC68LC302 Mask F81S. Errata listed in italics and which have change bars are new since the last published errata.**

.

## 1. Interrupt Vectors in Slave Mode

If the 68LC302 is used in slave mode in the PGA package ( the function code pins, FC2-0, are available) **AND** and an external asynchronous master is used **AND** the external master has not performed an internal read cycle since the previous IACK cycle, then during the current IACK cycle, the 68LC302 will drive the previous interrupt vector and the IPR and ISR registers will not be updated.

**Workaround**: The user should perform an internal read cycle (i.e. read the IPR or ISR or any other internal register) during the Interrupt routine.

## 2. Very Short Frames in Synchronous Protocols

If a 68LC302 SCC is used in a synchronous protocol mode (HDLC, BISYNC, or Transparent), AND the transmit frame is stored in just one data buffer, AND the transmit frame is only 1 or 2 bytes in total length (i.e. 1 or 2 bytes stored in the data buffer) AND more than one 68LC302 SCC is operating simultaneously,it is possible after some amount of time (minutes, hours, or days) for a frame to be reported as being transmitted successfully (i.e. the Tx BD shows that it has been transmitted without errors), but in reality, the frame was not transmitted over the TXD pin.

**Workarounds:**

1. In the case of transmitting a two byte frame, simply split the frame into 2 one-byte buffers. This will eliminate the problem, and will not affect serial performance.
2. Do nothing. Since the problem occurs very rarely, wait for the missing frame to be detected by the higher layers of the software, and a retransmission requested.

### *3. Low power modes current drain*

When the LC302 is put in the STOP or DOZE low power modes, there are some floating nodes in the silicon drain more current than specified. The current drain can be as high as 1mA to 4mA with these floating nodes, rather than the specified 100µA.

**Workaround:** Insert the following low power patch software code in order to drive the floating nodes and bring the current consumption down to ~ 100µA in STOP mode:

Assuming we enter low power ONLY when the SSCs are idle, and no transmission is in progress.

Before entering low power modes:

1. Disable all Serial ports:
2. PADDR = 0x0800; /* PTCLK3 output */
3. SPMODE = 0x01; /* en SCP */
4. SIMODE = 0x0001; /* PCM mode. Can be IDL or GCI as well */

After exiting from low power:

1. SIMODE
2. Enable SCCs

Notes:

1. SIMODE can be either PCM or IDL or GCI.
2. SCC2 can be left in NMSI mode by setting bit MSC2 bit in SIMODE.
3. The following table describes the effect on SCC1 pins when it gets out of NMSI mode: (Assuming SCCs are IDLE)

| | **NMSI** | **PCM** | **IDL** | **GCI** |
|---|---|---|---|---|
| RXD1/L1RXD | I | I | I | I |
| TXD1/L1TXD | O=1 | TS or O=1* | TS or O=1* | OD or O=0** |
| RCLK1/L1CLK | I or O | I | I | I |
| TCLK1/L1SY0/SDS1 | I or O | I | O=0,1*** | O=0,1*** |
| CD1/L1SY1 | I | I | I | I |
| CTS1/L1GR | I | I | I | I |
| RTS1/L1RQ/GCIDCL | O=1 | O=0 | O=0 | O=L1CLK/2 |

* output if one of L1SY1,0 are high
** output if setz=1 in SIMODE
*** 0 if not in IDL/GCI frame 1 otherwise

In order to draw little current, all inputs to the chip have to be pulled high or low. And no contentions are allowed. Therefore the specific application must be analyzed with this table. PADDR and SPMODE programming save about 1mA. SIMODE programming saves about 3mA.