Active Errata List

• USB – Ping-Pong databank 1 Re-transmission Failure
• USB – Ping-Pong OUT Bad Reception
• USB – Bad Remote Wake-up Generation
• UART Interface – During Reception, Clearing REN may Generate Unexpected IT
• C51 Core – Power-down Exit Failure in X2 Mode
• Timer 0/1 – Unexpected Interrupt
• USB Interface – Data Corruption in Endpoint0 and FIFO
• Timer 2 – Baud Rate Generator – Long Start Time
• Bad Suspend Resume Initialization
• Stretch MOVX Does Not Work

Errata History

<table>
<thead>
<tr>
<th>Lot Number</th>
<th>Errata List</th>
</tr>
</thead>
<tbody>
<tr>
<td>A03707, A03707J, A03707K,</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12</td>
</tr>
<tr>
<td>A04084A, A04272, A04272F,</td>
<td></td>
</tr>
<tr>
<td>A04272G, A04272J, A04272V,</td>
<td></td>
</tr>
<tr>
<td>A04426, A04427, A04428,</td>
<td></td>
</tr>
<tr>
<td>A04472S</td>
<td></td>
</tr>
<tr>
<td>A04427A, A04427B, A04564 and</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 10, 12</td>
</tr>
<tr>
<td>above</td>
<td></td>
</tr>
</tbody>
</table>

Errata Description

1. USB – Ping-Pong Databank 1 Re-transmission Failure

When the host does not acknowledge an IN data packet from the databank 1 of a ping-pong endpoint, the endpoint retry-mechanism sends corrupted data. Then a normal USB traffic takes place.

Workaround
None.

2. USB - Ping-Pong OUT Bad Reception

When the host sends a packet with a size lower than the size defined in the DPRAM endpoint (Ping-Pong Only), there is a risk of having a corrupted packet in the DPRAM with a wrong number of bytes reported. This problem occurs only in Ping-Pong mode if the 2 banks are full and bank 1 is cleared when the host is sending a packet at that time. If the packets are of the size of the DPRAM endpoint, there is no problem even in Ping-Pong mode.

If the device application software is quick enough to read the received packets to avoid the case where the 2 banks are full, there is no problem even if the packet sizes are not of the same size of the DPRAM endpoint.

Workaround
None.
3. **USB - Bad Remote Wake-up Generation**
   The remote wake-up generates an SE0 and J state (at the end of the Upstream Resume K) that are reserved by the Host. When a device is in suspend state and wants to notify an event to the host, it can send an upstream resume in order to wake up the host. The upstream resume consists of emitting a K state between 1 ms and 15 ms. At the end of this period, the device should leave the bus in idle state (J state) and wait for a SOF coming before 3ms. But at the end of the upstream resume, the USB controller drives a SEO (D+ and D- at 0 for 2 bit time) during 100ns before driving the J state.

   **Workaround**
   None.

4. **UART Interface – During Reception, Clearing REN may Generate Unexpected IT**
   During UART reception, if the REN bit is cleared between start bit detection and the end of reception, the UART will not discard the data (RI is set).

   **Workaround**
   Test REN bit at the beginning of interrupt routine just after CLR RI, and run the Interrupt routine code only if REN is set.

5. **C51 Core – Power-down Exit Failure in X2 Mode**
   If CPU is configured in X2 mode when exiting from power down, the first address fetched may be lost.

   **Workaround** Two solutions are possible:
   a) Set CPU in X1 mode before entering in power-down mode and then restore CPU to X2 mode when the CPU is woken up.
   b) Add a NOP (0x00) opcode just after the instruction which activates the power down mode. As this NOP is randomly non executed, the behavior of the software is correct.

   **Example:**
   ```assembly
   MOV PCON, #02H; Power down mode activation
   NOP ; This NOP is randomly not executed
   ............. ; Put here the first opcode to execute after exiting from power down mode
   ```

6. **Timer 0/1 – Unexpected Interrupt**
   If one of the timers 0 and 1 is in X1 mode while the other one is in X2 mode, an unexpected interrupt may randomly occur for one of the timers.

   **Workaround**
   Use the same mode X1 or X2 for both timers. This condition is met if PLL is used to clock the CPU.

7. **USB Interface – Data Corruption in Endpoint0 and FIFO**
   Data in Control Endpoint and FIFO may be corrupted if USB macro and CPU write in simultaneously. This condition occurs if the host cancels a control IN transaction with premature OUT and sends the following SETUP while the C51 is writing into the FIFO instead of the cancellation.

   **Workaround:** There are two ways to avoid this problem.
   Use 32 bytes FIFO to avoid fragmented data transaction on Control Endpoint.

8. **Timer 2 – Baud Rate Generator – Long Start Time**
   When Timer 2 is used as a baud rate generator, TH2 is not loaded with RCAP2H at the beginning, then UART is not operational before 10,000 machine cycles.

   **Workaround**
   Add the initialization of TH2 and TL2 in the initialization of Timer 2.
9. **Wrong Latch of Hardware Conditions**

   If the Reset input is controlled externally and is not synchronous with internal clock, the Hardware Conditions could be latched in the wrong status. These hardware conditions are used in the boot process and the device may not boot in the right target (Bootloader or Application)

   **Workaround**

   1) The internal synchronous Reset provided by the POR/PFD feature guarantees a reliable Reset at power-up.
   2) The internal Hardware Watchdog Reset can also be used without failure.

10. **USB Interface – Bad Suspend Resume Initialization**

    Sometimes and randomly when the USB device is plugged to the USB interface, it may not enumerate properly. It appears that the device remains in suspend state.

    **Workaround**

    After enabling the USB macro by means of bit USBE bit in USBCON, it is necessary to clear just after the bit WUPCPU in USBINT register.

11. **USB Interface – CPU Wake-up Interrupt Not Cleared**

    The WUPCPU bit in USBINT register is set by the hardware when the USB macro exits from suspend mode. The firmware acknowledges this event by clearing the WUPCPU bit in the interrupt routine. This bit fails to be cleared by software, therefore the USB Wake-up interrupt is not acknowledged and always executed until next suspend state.

    **Workaround**

    Disable WakeUp interrupt after resume sequence.

12. **Stretch MOVX Does Not Work**

    When setting M0 bit in AUXR SFR (08Eh), the RD or WR pulse on a MOVX instruction on external memory is not 30 XTAL length but always standard 6 XTAL length. Thus slow external peripherals mapping in the XDATA space could not work properly.

    **Workaround**

    None.