This document identifies implementation differences between the MPC5200B, M62C (Rev. 1), and the processor’s description contained in the MPC5200B User’s Guide and the MPC5200B Microcontroller Data Sheet. Refer to http://www.freescale.com for the latest updates.
# Overview

Table 1. MPC5200B (1M62C) errata summary

<table>
<thead>
<tr>
<th>Module</th>
<th>ID</th>
<th>Date Added</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATA</td>
<td>158</td>
<td>08/17/2007</td>
<td>Section 2.1, ATA interrupt is not affected by FIFO errors</td>
</tr>
<tr>
<td>e300 Core</td>
<td>ERR</td>
<td>12/01/2011</td>
<td>Section 3.1, If critical interrupt and normal interrupt are used in a system, the e300 core may hang</td>
</tr>
<tr>
<td>ETH</td>
<td>353</td>
<td>08/17/2007</td>
<td>Section 4.1, FEC intermittently resets following RX packets after reset</td>
</tr>
<tr>
<td></td>
<td>463</td>
<td>08/17/2007</td>
<td>Section 4.2, Interrupts do not occur</td>
</tr>
<tr>
<td>CLOCKs</td>
<td>498</td>
<td>08/17/2007</td>
<td>Section 5.1, System PLL – startup issue</td>
</tr>
<tr>
<td>MSCAN</td>
<td>499</td>
<td>08/17/2007</td>
<td>Section 6.1, Corrupt ID may be sent in Early-SOF condition</td>
</tr>
<tr>
<td>MUCts</td>
<td>01373</td>
<td>12/01/2011</td>
<td>Section 6.2, Message erroneously accepted if bus error in bit 6 of EOF</td>
</tr>
<tr>
<td>PCI</td>
<td>—</td>
<td>08/17/2007</td>
<td>Section 7.1, Glitches on PCI Reset pin</td>
</tr>
<tr>
<td></td>
<td>—</td>
<td>01/23/2008</td>
<td>Section 7.2, Violation of PCI TVAL max time</td>
</tr>
<tr>
<td>Interrupt</td>
<td>500</td>
<td>08/17/2007</td>
<td>Section 8.1, Interrupt controller can block interrupts</td>
</tr>
<tr>
<td>PSC I2S</td>
<td>501</td>
<td>08/17/2007</td>
<td>Section 9.1, TX channel switch is possible if TX FIFO becomes empty during transmission</td>
</tr>
</tbody>
</table>
2  ATA Controller

2.1  ATA interrupt is not affected by FIFO errors

2.1.1  Description
FIFO error flags do not generate an ATA interrupt to the CPU.

2.1.2  Workaround
To identify a FIFO error, the FIFO status register must be polled.

3  e300 Core

3.1  If critical interrupt and normal interrupt are used in a system, the e300 core may hang

3.1.1  Description
If critical interrupt and normal interrupt are being used in a system, the e300 core may hang.

3.1.2  Workaround
Instead of using an RFI at the end of every exception handler, replace the RFI with the following:
Disable critical interrupts by setting MSR[CE] to 0 with mtspr.
Copy SRR0 and SRR1 to CSRR0 and CSRR1, respectively.
Execute an RFCI. This enables MSR[CE] and any other bits that original RFI would have set, including MSR[EE]
Sample code:

```assembly
// Disable MSR[CE]
mfmsr r2
lis r3, 0xffff
ori r3, r3, 0xff7f
and r2, r2, r3
sync
mtmsr r2
isync

// Copy SRR0, SRR1 to CSRR0 and CSRR1
mfspr r2, srr0
mfspr r3, srr1
mtspr csrr0, r2
mtspr csrr1, r3
```
4 ETH

4.1 FEC intermittently resets following RX packets after reset

4.1.1 Description
After the FEC is reset by software, the FEC hardware can reset itself, after which it recovers without difficulty. No CPU interrupts are generated to indicate errors. This problem impacts only RX packets.

This only causes some occasional lost data after software reset. After the block comes out of software and self-induced hardware reset, no packets are lost. Higher level of protocols built on the Ethernet could request the lost packets and streaming applications are not affected.

4.1.2 Workaround
Higher level of protocols built on the Ethernet could request the lost packets.

4.2 Interrupts do not occur

4.2.1 Description
The late collision (LC), retry limit (RL), and underrun (UN) interrupts do not trigger on consecutive transmit frames. For example, if back-to-back frames cause a transmit underrun, only the first frame generates an underrun interrupt. No other underrun interrupts are generated until a frame is transmitted that does not underrun or the FEC is reset.

4.2.2 Workaround
Because late collision, retry limit, and underrun errors are not directly correlated to a specific transmit frame, a workaround for this problem is not needed for most cases. If a workaround is required, there are two independent workarounds:

- Ensure that a correct frame is transmitted after detecting late collision, retry limit, or underrun errors.
- Perform a soft reset of the FEC by setting ECR[RESET] when late collision, retry limit, or underrun errors are detected.
5  Clocks

5.1  System PLL – startup issue

5.1.1  Description
The system PLL can fail to start properly if the power supply pins on the MPC5200B are not properly sequenced. In particular, if the SYS_PLL_AVDD pin rises too early with respect to VDD_CORE, the system PLL can drive the VCO to oscillate at a frequency beyond the capability of the PLL to respond. The overall result is that the internal bus clocks run too fast and the device fails to fetch instructions.

5.1.2  Workaround
A filter/delay circuit must be implemented so that SYS_PLL_AVDD is delayed with respect to VDD_CORE. Specifically, VDD_CORE must reach a value of 1.2 volts before SYS_PLL_AVDD reaches 0.6 volts.

6  MSCAN

6.1  Corrupt ID may be sent in Early-SOF condition

6.1.1  Description
This erratum is only relevant for applications using more than one transmit buffer and with oscillators operating at opposite ends of the tolerance range. A corrupt ID is sent out if a Tx message with highest priority is set up for transmission during bit 3 of INTERMISSION and a dominant bit is sampled leading to an early-SOF condition. The message sent is taken from the newly setup Tx buffer with the exception of the first eight ID bits taken from the originally selected Tx buffer. The CRC is correctly calculated on the resulting bit stream so that receiving nodes validate the message. In a typical CAN network, with
oscillator tolerances inside of the specified limits, this is not an issue because an early SOF condition should not occur. This may be a problem if the oscillators operate at opposite ends of the tolerance range (max. 1.58%), which could lead to a cumulated phase error after 10 bit times larger than phase segment 2.

NOTE
The CAN protocol condition referred to as early SOF in this erratum is detailed in a note in the “Bosch CAN Specification Version 2.0 Part B”, section 3.2.5 INTERFRAME SPACING – INTERMISSION.

6.1.2 Workaround
Use only one transmit buffer and do priority sorting in software. If more than one transmit buffer must be used, one of the following workarounds can be chosen:

- Do not release higher priority messages than those already scheduled (either determined by local priority or if equal, by hard-coded buffer priority) before all buffers are empty.
- Abort messages before scheduling higher priority transmissions.
- Prevent bad IDs passing acceptance filters by not assigning IDs (x) consisting of a combination of other assigned IDs (y,z), i.e. IDx[11:0] != {IDy[11:3], IDz[2:0]} for standard and IDx[28:21] != {IDy[28:21],IDz[20:0]} for extended identifiers.
- Allocate dedicated data length codes (DLC) to every ID and check for correspondence after reception.

6.2 Message erroneously accepted if bus error in bit 6 of EOF

6.2.1 Description
If a particular error condition occurs within the end of frame segment (EOF) of a CAN message, the MSCAN module recognizes and accepts a non-valid message as being valid, contrary to the CAN specification. The MSCAN module incorrectly validates messages after five recessive bits of the end of frame instead of after six bits. If a bus error occurs during the sixth bit of end of frame, the MSCAN module will already have accepted the message as valid, even although an error frame is transmitted and the receive error counter is incremented.

The CAN protocol states that message validation differs between bus transmitter and receiver devices (refer to part B, section 5 of CAN protocol for details). In the case where the seventh bit of the EOF segment is dominant, the message is valid for the receiver but not for the transmitter. This erratum extends this case to the sixth bit of the EOF segment.

6.2.2 Workaround
This erratum will not be an issue if the application software is protected against the known double receive problem of the CAN protocol. This problem occurs when a message is not recognized as valid by the transmitter, but is recognized as valid by a receiver, as described above. When this happens, the message is re-transmitted and hence the receiver will receive the same message twice.
7 PCI

7.1 Glitches on PCI Reset pin

7.1.1 Description
The PCI reset line may have glitches 1 – 2 ns in duration.

7.1.2 Workaround
Put a series resistor of approximately 300 ohms between the PCI reset pin and any external devices served by the PCI Reset pin.

7.2 Violation of PCI $T_{VAL}$ max time

7.2.1 Description
The PCI timing specification defines the timing parameter $T_{VAL}$ (CLK to signal valid delay):

<table>
<thead>
<tr>
<th>Sym</th>
<th>Description</th>
<th>66 MHz</th>
<th>33 MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>$t_{val}$</td>
<td>CLK to Signal Valid Delay - bused signals</td>
<td>2</td>
<td>6</td>
</tr>
</tbody>
</table>

MPC5200B provides $T_{VAL}$ max of 6.8 ns (instead of 6 ns), which is a violation of the timing specification for 66 MHz.

7.2.2 Workaround
Make sure that this reduced setup time budget is considered by the design of the external PCI bus system.

8 Interrupt

8.1 Interrupt controller can block interrupts

8.1.1 Description
The interrupt controller can block any further interrupts if the three conditions are valid:

- MEE bit in ICTL external enable and external type register is set and individual enable (IRQ1) is not
• IRQ1 has elevated priority and/or redirect to smi_int by setting any bit(s) in the Main1_pri field in ICTL main interrupt priority and INT/SMI select 1 register
• IRQ1 occurs

8.1.2  Workaround
Avoid the scenario for the configuration of the external interrupts.

9   PSC I2S

9.1  TX channel switch is possible if TX FIFO becomes empty during transmission

9.1.1  Description
During an I2S transfer, the TX FIFO becomes empty and the SW writes new data to TX FIFO, the assignment of the TX channels could be switched. The first data word (normally left channel) is transmitted on the right channel slot. The second data word is transmitted on the left channel.

9.1.2  Workaround
The software must detect the current state of the I2S transmission by reading the frame sync line and make sure the enable of the TX/RX or the write of the first TX data to the FIFO (if the TX is enabled) occurs more than two serial clock cycles before the next active edge.
### Table 3. Revision History

<table>
<thead>
<tr>
<th>Rev. Number</th>
<th>Substantive Changes</th>
<th>Date of Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Initial release.</td>
<td>12/2006</td>
</tr>
<tr>
<td>2.0</td>
<td>Added items:</td>
<td>08/2007</td>
</tr>
<tr>
<td></td>
<td>• 500 Interrupt</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 501 PSC I2S</td>
<td></td>
</tr>
<tr>
<td>3.0</td>
<td>Added item Violation of PCI TVAL</td>
<td>01/2008</td>
</tr>
<tr>
<td>4.0</td>
<td>Added items:</td>
<td>12/2011</td>
</tr>
<tr>
<td></td>
<td>• ERR003383, e300 Core</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• MUCts01373, MSCAN</td>
<td></td>
</tr>
</tbody>
</table>

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-compliant and/or non-Pb-free counterparts. For further information, see [http://www.freescale.com](http://www.freescale.com) or contact your Freescale sales representative.

For information on Freescale’s Environmental Products program, go to [http://www.freescale.com/epp](http://www.freescale.com/epp).

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org © Freescale Semiconductor, Inc. 2007, 2008, 2011. All rights reserved.