MPC5200B (M62C) Errata

by: Microcontroller Division

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.

Contents

freescale™ semiconductor

1 Overview

Table 1. MPC5200B (1M62C) errata summary

Module	ID	Date Added	Title
ATA	158	08/17/2007	Section 2.1, ATA interrupt is not affected by FIFO errors
e300 Core	ERR 003383	12/01/2011	Section 3.1, If critical interrupt and normal interrupt are used in a system, the e300 core may hang
ETH	353	08/17/2007	Section 4.1, FEC intermittently resets following RX packets after reset
	463	08/17/2007	Section 4.2, Interrupts do not occur
CLOCKS	498	08/17/2007	Section 5.1, System PLL – startup issue
MSCAN	499	08/17/2007	Section 6.1, Corrupt ID may be sent in Early-SOF condition
	MUCts 01373	12/01/2011	Section 6.2, Message erroneously accepted if bus error in bit 6 of EOF
PCI	—	08/17/2007	Section 7.1, Glitches on PCI Reset pin
	—	01/23/2008	Section 7.2, Violation of PCI T_{VAL} max time
Interrupt	500	08/17/2007	Section 8.1, Interrupt controller can block interrupts
PSC I2S	501	08/17/2007	Section 9.1, TX channel switch is possible if TX FIFO becomes empty during transmission

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:

```
// 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
```

```
… restore GPRs

rfci
```

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

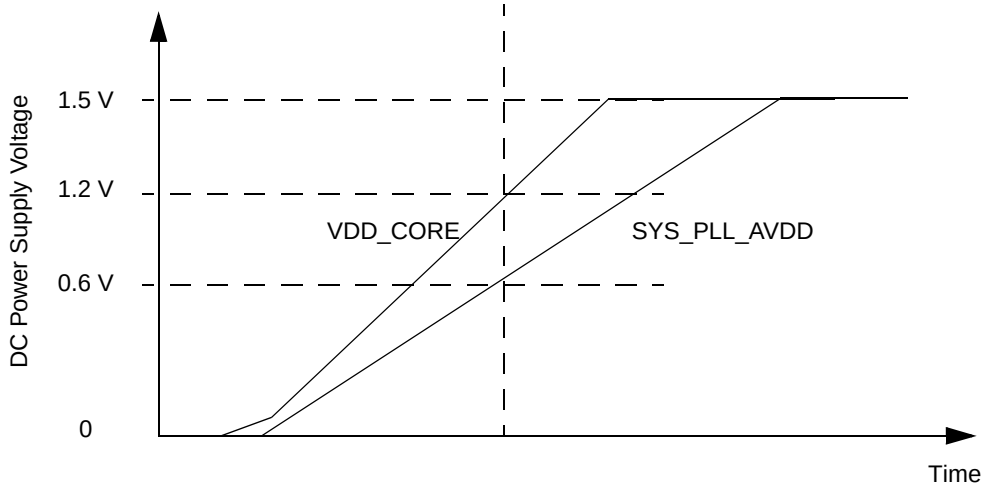
5.1 System PLL – startup issue

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 2. PCI Timing Parameters

Sym	Description	66 MHz		33 MHz		Units
		Min	Max	Min	Max	
t_{val}	CLK to Signal Valid Delay - bused signals	2	6	2	11	ns

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

Rev. Number	Substantive Changes	Date of Release
1.0	Initial release.	12/2006
2.0	Added items: • 500 Interrupt • 501 PSC I2S	08/2007
3.0	Added item Violation of PCI T_{VAL}	01/2008
4.0	Added items: • ERR003383, e300 Core • MUCts01373, MSCAN	12/2011

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or +1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Document Number: MPC5200BE
Rev. 4
12/2011

freescale™ semiconductor