

# MC9328MXS Integrated Processor (i.MXS)

## Chip Errata for Mask 2L45N

This document covers silicon errata information that relates to the 2L45N mask of the MC9328MXS (i.MXS).  
 Rev. 8 adds Erratum 14 to this document.

**Table 1. Silicon Errata to MC9328MXS**

Erratum Number	Erratum Description	Impact, Workaround, Fix Status	Applies to Mask
1.	<p><b>Module:</b> SDRAMC</p> <p><b>Failure:</b> It is necessary to use a separate timer to prevent access to the SyncFlash immediately after the SyncFlash is re-enabled from deep low-power mode.</p>	<p><b>Impact:</b>            If SyncFlash channel is disabled, the SDRAMC puts the syncflash into a mode called deep low-power mode. When the SyncFlash is re-enabled, it requires a 100 <math>\mu</math>s period to initialize before allowing any access to it.            The current design does not have a built-in timer to prevent access before 100 <math>\mu</math>s has passed. Therefore, an access occurring during this time period could return invalid data from the SyncFlash.</p> <p><b>Workaround:</b>            Use the TIMER register to count 100 <math>\mu</math>s to determine the time period during which the user is not allowed to access the SyncFlash.</p> <p><b>Fix Status:</b>            No metal-fix solution.</p>	2L45N



**Table 1. Silicon Errata to MC9328MXS (Continued)**

Erratum Number	Erratum Description	Impact, Workaround, Fix Status	Applies to Mask
2.	<p><b>Module:</b> LCDC</p> <p><b>Failure:</b> The Memory Controller arbiter (MEMC) does not break and release the ARM™ core's back-to-back burst access to the other (for example, LCDC, DMA, and MMA) bus masters. In this situation, the ARM core's back-to-back access does not use an IDLE cycle as a separator.</p>	<p><b>Impact:</b> Typically, the back-to-back burst access that does not use the IDLE cycle as a separator occurs for operations such as cache fill, miss, and burst operations by the ARM processor. In cases such as this, the MEMC arbiter incorrectly treats the back-to-back sequence as a single instruction process that causes the other bus masters to wait longer to access the bus. On the AHB bus, only the LCDC and MMA modules are affected. Because the MMA is not a periodic and time critical module with regards to requesting the grant of the bus, the impact of this issue is minimal, although the LCDC frame buffer refresh can be greatly impacted.</p> <p><b>Workaround:</b> Use a dummy DMA transfer to and from a dummy memory space to force the MEMC to release the bus for other bus masters. For more detailed information and code example, please visit our website <a href="http://www.freescale.com/imx">www.freescale.com/imx</a>.</p> <p><b>Fix Status:</b> No metal-fix solution.</p>	2L45N

**Table 1. Silicon Errata to MC9328MXS (Continued)**

Erratum Number	Erratum Description	Impact, Workaround, Fix Status	Applies to Mask
3.	<p><b>Module:</b> USB Device Port</p> <p><b>Failure:</b> Limitation on using DMA for USB IN data transfer.</p> <p>There is a limitation on using DMA to transfer continuous data from memory to a 64-byte USB FIFO (FIFO1 and FIFO2) when using a short packet (&lt;37 bytes) followed by a long packet (&gt; 32 bytes).</p> <p><i>For example:</i> A 6 byte packet followed by a 64 byte packet, or a 32 byte packet followed by a 40 byte packet such that the short packet length + the long packet length is greater than 64 bytes (FIFO depth).</p>	<p><b>Impact:</b></p> <p>The DMA request does not clear after the reception of a short packet, causing a long packet to start filling in the FIFO before the data in the short packet is moved out of the FIFO. This causes the USB FIFO to overflow and generate an error that prevents a DMA interrupt DMA_ISR from being generated.</p> <p>This occurs because the ALARM bit (DMA request) being cleared is based on the setting of the 4*GR[2:0] granularity bits in USB_EPn_FCTRL register. The ALARM bit for IN transfer is set when the amount of data bytes that remain in the FIFO is below the alarm value (ALARM[5:0] in the USB_Epn_FALARM register and cleared when there is less than 4*GR[2:0] of free bytes that remain in the FIFO.</p> <p>The Granularity bits GR[2:0] are fixed at 3 bits regardless of the FIFO depth. For IN transfer, DMA requests are cleared when FREE data bytes are 4*GR[3:0]. Because GR[2:0]=3 * 111 as a maximum value, there can only be 28 free bytes. Therefore, a 6 byte short packet in the 64-byte FIFO will not desert the DMA request.</p> <p><b>Workaround:</b></p> <p>Two methods are used to prevent this limitation depending on the data packet size:</p> <ol style="list-style-type: none"> <li><i>For data packet sizes of less than 5 bytes:</i> After the short packet data transfer, software must poll the USBD BYTE_COUNT==0 before initiating the next DMA transfer. This ensures that FIFO can accept the next long data packet of 64-bytes.</li> <li><i>For data packet sizes of 5 bytes and greater:</i> Use a 32-byte FIFO instead of a 64-byte FIFO to ensure that the DMA request is cleared. The granularity bits GR[2:0] must be set to 111.</li> </ol> <p><b>Fix Status:</b></p> <p>No metal-fix solution.</p>	2L45N

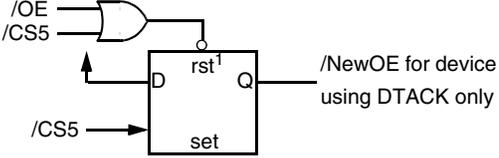
**Table 1. Silicon Errata to MC9328MXS (Continued)**

Erratum Number	Erratum Description	Impact, Workaround, Fix Status	Applies to Mask
4.	<p><b>Module:</b> RTC</p> <p><b>Failure:</b> An alarm set to an odd number of seconds results in a false interrupt one second before the actual alarm time, followed by the actual interrupt.</p>	<p><b>Impact:</b> This double interrupt is the result of a glitch produced by the seconds counter bits 1 and 0. Bit 0 of the seconds counter has greater loading than bit position 1. Upon receiving a clock signal to update the seconds counter, the seconds counter bit 0 changes state to be slower than counter bit 1. This glitch causes the alarm to see a momentary match, triggering the interrupt one second early. When the seconds counter actually does match the alarm value, another interrupt is triggered (if it has not been disabled following the previous interrupt) at the correct time.</p> <p><i>For example:</i> If you set the alarm to “xx : xx : xx : 15 sec”, when the second counter changes from 13(1101) to 14(1110), the bit position 1 changes faster than bit position 0. The result is 13(1101) to 15(1111) to 14(1110). This 15(1111), the glitch, matches the alarm setting and triggers the interrupt.</p> <p><b>Workaround:</b> Either set the seconds alarm register to an even value or, if set to an odd value, when the interrupt occurs, check the current time against the alarm time. If it does not match, then clear the interrupt and return. The actual interrupt will occur one second later.</p> <p><b>Fix Status:</b> No metal-fix solution.</p>	2L45N
5.	<p><b>Modules:</b> IOPAD</p> <p><b>Failure:</b> Power-up leakage during the recommended power-up sequence, NVDD&gt;AVDD&gt;QVDD. In the normal configuration, the maximum current leakage occurring during the power-up period can be up to 600mA.</p>	<p><b>Impact:</b> Using the workaround, the power-up current can be minimized to less than maximum specified operating current.</p> <p><b>Workaround:</b> Attach a 1 K<math>\Omega</math> series resistor to each tied-low input pin.</p> <p><b>Fix Status:</b> No metal-fix solution.</p>	2L45N

**Table 1. Silicon Errata to MC9328MXS (Continued)**

Erratum Number	Erratum Description	Impact, Workaround, Fix Status	Applies to Mask
6.	<p><b>Modules:</b> MEMC/AHBC</p> <p><b>Failure:</b> The LCD flicker issue is not completely resolved by the use of dummy DMA transfer (See Erratum number 7).</p> <p>The LCDC DMA high/low setting does not ensure that the next bus request is triggered in the event the current DMA burst is not completed. The problem occurs when the next bus request is initiated while the LCDC is waiting for the last data in the current DMA burst. This is because when the LCDC is waiting for the last data, it only expects an HREADY signal to be asserted. So, if the next bus request is triggered while the LCDC is waiting, the bus request will be missed. A data fetch can not be initiated while data in FIFO is being continuously drained so an under-run will eventually occur.</p> <p><i>In Summary:</i> If the next bus request is issued when the LCDC is waiting for the last data in the current DMA burst it will not be recognized.</p>	<p><b>Impact:</b> The low mark must never be set higher than 10 and the high mark must be set at 3.</p> <p><b>Workaround:</b> For a bus that is heavily loaded and that requires SDRAM access, a dynamic burst length is recommended:</p> <ul style="list-style-type: none"> <li>• fixed burst length = 0</li> <li>• high mark = 3</li> <li>• low mark = 8</li> </ul> <p>For a very heavily loaded system, increasing the low mark value increases the chance of a bus grant of the system bus, at the expense of more frequent bus requests.</p> <p><b>Fix Status:</b> No metal-fix solution.</p>	2L45N

**Table 1. Silicon Errata to MC9328MXS (Continued)**

Erratum Number	Erratum Description	Impact, Workaround, Fix Status	Applies to Mask
7.	<p><b>Modules:</b> EIM</p> <p><b>Failure:</b> CS5 fails read operation when using the DTACK function.</p>	<p><b>Impact:</b> When CS5 is configured to an external input DTACK signal to terminate the bus cycle (for example WSC = 0x3F), using either DTACK mode (DTACK_SEL = 0) or WAIT mode (DTACK_SEL = 1), the CPU or DMA read cycles in CS5 memory space may terminate incorrectly. CS5 can function normally by using wait state control bus termination—that is, WSC is not set to 0x3F. All write cycles will function correctly.</p> <p><b>Workaround:</b> <i>DTACK mode (DTACK_SEL = 0):</i> No workaround. <i>WAIT mode (DTACK_SEL = 1):</i> In Wait mode the OE signal for read cycle is negated up to 1 system clock before a data latch occurs at the end of bus cycle, possibly causing incorrect data to be latched by the i.MXS because the external device releases the data bus after OE negates. The workaround is to use external logic (one D-flip flop and one OR-gate) to extend OE negation time until the CS cycle is completed (after or at CS5 rising edge). D-FF connection must be D=1, RST= /OE OR /CS5, CLK= CS5, Q=NewOE for DTACK device only. To avoid disrupting other devices using /OE, the NewOE signal only replaces the /OE signal going to a DTACK device. See Figure 1.</p>  <p>1. If using LCX logic from ON Semiconductor, expect propagation delays from /OE and /CS assertion to /newOE of 3.0 to 12.5 ns. Freescale Semiconductor cannot recommend one supplier over another and in no way suggests that ON Semiconductor is the only supplier of logic devices.</p> <p><b>Figure 1. D-FF Connection Workaround</b></p> <p><b>Fix Status:</b> No metal-fix solution.</p>	2L45N
8.	<p><b>Modules:</b> EIM</p> <p><b>Failure:</b> Port A pin 17 (multiplexed with DTACK) inadvertently affects other chip-select functions when DTACK is not used.</p>	<p><b>Impact:</b> When the DTACK signal is not used, toggling the GPIO pin PA17 will adversely affect the other EIM chip-select functions.</p> <p><b>Workaround:</b> PA17 is restricted for the DTACK functionality only. Thus, in any other situation when DTACK is not used, this pin must be pulled up or tied high (may be done by simply setting the corresponding PUEN bit, configuring it as an input, and leaving the pin unconnected). If the DTACK functionality is used, the pin will only toggle during CS5 accesses and should not toggle during any non-CS5 access.</p> <p><b>Fix Status:</b> No metal-fix solution.</p>	2L45N

**Table 1. Silicon Errata to MC9328MXS (Continued)**

Erratum Number	Erratum Description	Impact, Workaround, Fix Status	Applies to Mask
9.	<p><b>Modules:</b> SDRAMC</p> <p><b>Failure:</b> Cannot support SDRAMs with less than 1kilobyte page size.</p>	<p><b>Impact:</b> Not able to support single chip 4M x 16-bit (8 Mbyte total size) SDRAM devices or devices with a page size of less than 1 kilobyte. A page size is defined as the number of bits in a row in the SDRAM device.</p> <p><b>Workaround:</b> Must use SDRAM devices with 1 kilobyte or greater page sizes.</p> <p><b>Fix Status:</b> No metal fix is planned.</p>	2L45N
10.	<p><b>Modules:</b> SDRAMC</p> <p><b>Failure:</b> Cannot support SDRAMs with column address sizes of 10.</p>	<p><b>Impact:</b> Not able to support single chip 32M x 16-bit (64 Mbyte total size) SDRAM devices or devices with a column address size of 10.</p> <p><b>Workaround:</b> For 64 Mbyte SDRAM devices the configuration of 16 M x 32 or two 16M x 16-bit must be used to form a 32-bit data bus, where either configuration contains only 9 column addresses.</p> <p><b>Fix Status:</b> No metal fix is planned.</p>	2L45N
11.	<p><b>Modules:</b> SDRAMC</p> <p><b>Failure:</b> In 16-bit mode, only SDRAM device with 9 Cols are supported.</p>	<p><b>Impact:</b> In 16-bit mode, burst access is not guaranteed for SDRAM with number of columns not equal to 9. Known impact is failure to support following devices in 16-bit mode:</p> <ul style="list-style-type: none"> <li>a. SDRAM with 12 Rows and 8 Cols (4Mx16-bit, 8MB total size)</li> <li>b. SDRAM with 13 Rows and 10 Cols (32Mx16-bit, 64 Mbyte total size)</li> </ul> <p><b>Workaround:</b> No workaround is available. Recommend system designer to use 32-bit SDRAM if 8 Mbyte or 64 Mbyte memory configuration is desired.</p> <p><b>Fix Status:</b> No metal fix is planned.</p>	2L45N

**Table 1. Silicon Errata to MC9328MXS (Continued)**

Erratum Number	Erratum Description	Impact, Workaround, Fix Status	Applies to Mask
12.	<p><b>Modules:</b> ARM920T AHB Wrapper</p> <p><b>Failure:</b> LDM instruction fails to load non-cached data from memory.</p>	<p><b>Impact:</b> When a load multiple (LDM) instruction is used to load two specific registers from an un-cached region of memory, and the load instruction begins immediately after a delayed buffered write, then the LDM will not load the second register correctly.</p> <p><b>Workaround:</b> The ARM Realview 3.0 SP1 Build 617 compiler and linker have a patch that addresses this issue. With this patch applied, the compiler will not generate this LDM instruction. The Build 617 patch is available to download from the ARM website.</p> <p>To use the patch, users must insert “--branchpatch 920t-ldm2” to the compiler and linker command lines.</p> <p>For code compiled by a compiler other than the above stated tool, a manual search and replace for the LDM instruction of two values in assembly code can be done with multiple options for equivalent code replacements.</p> <p>Example 1: LDMIA r0, {r1, r2} is functionally IDENTICAL to: LDR r1, [r0] LDR r2, [r0, #4]</p> <p>Example 2: If updating the base register to a new value, such as: LDMIA r0!, {r1, r2} then this is functionally IDENTICAL to: LDR r1, [r0], #4 LDR r2, [r0], #4</p> <p><b>Fix Status:</b> No hardware fix planned.</p>	2L45N

**Table 1. Silicon Errata to MC9328MXS (Continued)**

Erratum Number	Erratum Description	Impact, Workaround, Fix Status	Applies to Mask
13.	<p><b>Module:</b> UART</p> <p><b>Failure:</b> Parity bug in UART transmitter</p>	<p><b>Impact:</b> A parity error in UART character transmissions may occur. TxFIFO data being shifted out is not alterable after the start bit is issued. However, the parity bit can change after the start bit is issued and is not fixed until data Bit0 is issued. If the TxFIFO is overwritten by the CPU during the interval from the end of the start bit to the end of the data Bit0, there will be a mismatch between the data shifted out and the parity bit. In this case, a parity error may occur.</p> <p><b>Workaround:</b> The following software solutions are recommended to be used in Interrupt Service Routines (ISRs) to avoid the parity bug. Polling can also be used instead of interrupts.</p> <ol style="list-style-type: none"> <li>1. A maximum of 32 bytes can be written into the TxFIFO when transmission is complete (USR2.TXDC = 1).</li> <li>2. A maximum of 31 bytes can be written into the TxFIFO when the TxFIFO is empty (USR2.TXFE = 1).</li> <li>3. A maximum of 32 - n bytes, where n = UFCR.TXTL, can be written into the TxFIFO when the data level in the TxFIFO falls below the selected threshold (when USR1.TRDY = 1).</li> </ol> <p><b>Fix Status:</b> No hardware fix planned.</p>	2L45N

**Table 1. Silicon Errata to MC9328MXS (Continued)**

Erratum Number	Erratum Description	Impact, Workaround, Fix Status	Applies to Mask
14.	<p><b>Module:</b> SSI</p> <p><b>Failure:</b> In word-wide frame-sync mode, if SSI_EN and TX/RX_EN are set near new frame start, SSI Receive and Transmit does not function properly and there is a chance of words getting missed/shifted or transmit data line being driven low for 1/2 clock cycles during the first frame.</p>	<p><b>Impact:</b> <i>Receive malfunction:</i> Rx data is usually taken with the period of FS=H. However, in some special use cases, it is changed to period of FS=L. Once this occurs, this period never returns to the period of FS=H. The issue occurs in the following conditions: Normal mode Sync mode Slave mode</p> <p>Some additional conditions of this scenario are: TX and RX are used at the same time (full duplex) SYN=1,RXDIR=0,TXDIR=0,TFDIR=0 External clock is 2.048MHz Frame frequency is 8kHz Frame pulse width is 1 word (8 clock) SSI_EN and RE is set to 1 within latter 4 clocks of FS=H period (8 clocks)</p> <p><i>Transmit malfunction:</i> TxData is changed to different value at only the first time of transmitting when TE bit is set just before FS signal or within FS timing.</p> <p><b>Workaround:</b> To ensure a gap of at least 2 bit-clock cycles between SSI_EN and TX/RX bit setting (only for External word-wide frame-sync patterns), here is an example code for this workaround.</p> <pre> void ssi_transmit_receive_test() { //parameters defined here Statement1 (); //clock configuration here Statement2(); //AUDMUX configuration here Statement3(); //SSI configuration here Statement4(); //Enable SSI by programming SSI_SCR MEMWD(SSi2_SCR,scr_mask1); //delay some time for the SSI state machine to function //properly for(i=0;i&lt;=5;i++); //Enable transmit and receive of SSI by programming //SSI_SCR MEMWD(SSi2_SCR,scr_mask2); ..... } </pre> <p><b>Fix Status:</b> No metal-fix solution planned.</p>	2L45N

## NOTES

**How to Reach Us:**

**USA/Europe/Locations Not Listed:**

Freescale Semiconductor Literature Distribution Center  
P.O. Box 5405  
Denver, Colorado 80217  
1-800-521-6274 or 480-768-2130

**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 Hong Kong Ltd.  
2 Dai King Street  
Tai Po Industrial Estate  
Tai Po, N.T., Hong Kong  
852-26668334

**Home Page:**

[www.freescale.com](http://www.freescale.com)

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.

**Learn More:** For more information about Freescale products, please visit [www.freescale.com](http://www.freescale.com).

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. The ARM POWERED logo is the registered trademarks of ARM Limited. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2005, 2006, 2007. All rights reserved.