MCF5271 Chip Errata
Silicon Revision: All

This document identifies implementation differences between the MCF5270 and MCF5271 processors and the description contained in the MCF5271 ColdFire® Reference Manual. Refer to http://www.freescale.com/coldfire for the latest updates.

All current MCF5270 and MCF5271 devices are marked as L23W mask set. The date code on the marking can be used to determine which errata have been corrected on a particular device, as shown in Table 1. The datecode format is XXXXYYWW, where YY represents the year and WW represents the work week. The four leading digits can be ignored.

Table 1. Summary of MCF5270 and MCF5271 Errata

<table>
<thead>
<tr>
<th>Errata</th>
<th>Module Affected</th>
<th>Date Errata Added</th>
<th>Date Code Affected?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>&lt; XXXX0501</td>
</tr>
<tr>
<td>SECF005</td>
<td>Cache</td>
<td>8/11/04</td>
<td>Yes</td>
</tr>
<tr>
<td>SECF001</td>
<td>Cache</td>
<td>8/11/04</td>
<td>Yes</td>
</tr>
<tr>
<td>SECF009</td>
<td>FEC</td>
<td>8/11/04</td>
<td>Yes</td>
</tr>
<tr>
<td>SECF007</td>
<td>FEC</td>
<td>8/11/04</td>
<td>Yes</td>
</tr>
<tr>
<td>SECF010</td>
<td>FEC</td>
<td>8/11/04</td>
<td>Yes</td>
</tr>
<tr>
<td>SECF030</td>
<td>PLL</td>
<td>12/30/05</td>
<td>Yes</td>
</tr>
<tr>
<td>SECF131</td>
<td>PLL</td>
<td>9/15/08</td>
<td>Yes</td>
</tr>
<tr>
<td>SECF163</td>
<td>PLL</td>
<td>3/4/10</td>
<td>Yes</td>
</tr>
</tbody>
</table>
The table below provides a revision history for this document.

### Table 2. Document Revision History

<table>
<thead>
<tr>
<th>Rev. No.</th>
<th>Date</th>
<th>Substantive Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Initial revision</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td></td>
<td>Removed “Internal Pull-ups on FEC Signals” as this errata does not pertain to the MCF5270 or MCF5271 devices.</td>
</tr>
<tr>
<td>1</td>
<td>9/2008</td>
<td>Updated errata with new revision of silicon; Added SECF030.</td>
</tr>
<tr>
<td>2</td>
<td>9/2008</td>
<td>Added SECF131</td>
</tr>
<tr>
<td>3</td>
<td>4/2010</td>
<td>Added SECF163</td>
</tr>
<tr>
<td>4</td>
<td>12/2010</td>
<td>Added XXXX1011 datecode which fixes SECF163</td>
</tr>
</tbody>
</table>
| 5        | 6/2011  | • Changed datecode which fixes SECF163 to XXXX1031  
|          |         | • Corrected revision numbers in revision history table                                |

**SECF001: Incorrect Operation of Cache Freeze (CACR[CFRZ])**

**Errata type:** Silicon  
**Affects:** Version 2 ColdFire Cache  
**Description:** The cache on the V2 ColdFire core is controlled by the cache control register (CACR). When the CACR[CFRZ] bit is set, the cache freeze function is enabled and no valid cache array entry is displaced. However, this feature does not always work as specified, sometimes allowing valid lines to be displaced when CACR[CFRZ] is enabled.

This does not cause any corrupted accesses. However, there could be cache misses for data that was originally loaded into the cache but was subsequently deallocated, even though the CACR[CFRZ] bit was set.

Also, incoherent cache states are possible when a frozen cache is cleared via the CACR[CINV] bit.

**Workaround:** Unfreeze the cache by clearing CACR[CFRZ] when invalidating the cache using the CACR[CINV] bit

**Workaround:** Use the internal SRAM to store critical code/data if the system cannot handle a potential cache miss

**Fix plan:** Currently, there are no plans to fix this.

**SECF005: Possible Cache Corruption After Clearing Cache (Setting CACR[CINV])**

**Errata type:** Silicon  
**Affects:** Version 2 ColdFire Cache  
**Description:** The cache on the V2 ColdFire core may function as either:
a unified data and instruction cache
an instruction cache
a data cache

The cache function and organization is controlled by the cache control register (CACR). The CACR[CINV] bit causes a cache clear. If the cache is configured as a unified cache and the CINV bit is set, the scope of the cache clear is controlled by two other bits in the CACR:

- CACR[INVI] invalidates instruction cache only
- CACR[INVD] invalidates data cache only

If a write to the CACR is performed to clear the cache (CACR[CINV] = 1) and only a partial clear is done (CA

Workaround: All loads of the CACR that perform a cache clear operation (CACR[CINV] set) should be followed immediately by a NOP instruction. This avoids the cache corruption problem.

Fix plan: Currently, there are no plans to fix this.

SECF009: FEC Receive Buffer Overrun in 10BaseT Mode

Errata type: Silicon
Affects: FEC
Description: When the FEC is connected to a 10BaseT network, if length of the data stored in a descriptor is not evenly divisible by 16 (not line-aligned), the FEC writes extra lines at the end of the buffer. The entire line that contains the last valid data is written and at least one extra line, but up to four lines after the end of the valid data can also be written. In most cases, this is not a problem because the extra lines of data continue falling within the limits of the buffer. However, if the valid data ends near the end of the buffer, the extra lines written by the FEC might be outside of the data buffer. This leads to corruption of the next buffer, descriptor, data, or code stored in the adjacent memory.

For example, as shown in the figure below, if the max buffer size is programmed to 0x600 and a frame that is 0x5F8 bytes long is received, a line is written starting at buffer start + 0x5F0. The first half of the line at buffer start + 0x5F0 is valid frame data that should be processed by the FEC driver; the second half of the line is additional data that is written because the FEC only writes complete lines. This data should be ignored by the FEC driver. So far, this is correct FEC behavior as originally specified. However, the FEC repeats the last line of valid data a number of times. The line at buffer start + 0x600 is written, and as many as three additional lines beyond the end of the data buffer could be written.
Workaround: Only use 100BaseT.

Workaround: Allocate extra lines for the receive data buffers. The actual allocated memory for each buffer should be equal to the receive buffer size programmed in the FEC’s EMRBR register plus four lines (16 byte-sized lines).

Workaround: Program the data buffer size one line larger than the max packet size (data buffer size = EMRBR + 0x40).

Fix plan: Currently, there are no plans to fix this.

**SECF007: Concatenation of Received Frames in 10BaseT Mode**

**Errata type:** Silicon  
**Affects:** FEC  
**Description:** When the FEC is connected to a 10BaseT network, sometimes the FEC combines the data from multiple frames to generate a single frame. The data from the frames is received correctly, but the frame boundary is not reported correctly. This causes the descriptor to report the length as the data length for all of the concatenated frames added together. The incorrect data length might exceed the max frame length programmed in the RCR[Max_FL] field.

When TCP is used as a transport mechanism, this errata manifests itself as lost packets and reduced throughput. Data continues to be received correctly because TCP requests retransmission of bad packets. However, UDP does not include any mechanism for packet retransmission, as it is a send and forget protocol. Consequently, while UDP should be able to identify an incorrectly received packet (because its checksum will fail), higher level software in the protocol stack must be capable of requesting retransmission to work around this errata.

Workaround: Higher level Ethernet layer code should compare the length reported by the descriptor to the length included in its header. If the lengths do not match, the packet should be truncated or discarded as needed. The protocol stack must be responsible for requesting retransmission of any frames that are discarded due to the data length mismatch.

Fix plan: Currently, there are no plans to fix this.
**SECF010: FEC Interrupts will not Trigger on Consecutive Transmit Frames**

**Errata type:** Silicon  
**Affects:** FEC  
**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.

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

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

**Fix plan:** Currently, there are no plans to fix this.

**SECF030: Frequency Modulation Mode on PLL Not Functional**

**Errata type:** Silicon  
**Affects:** FMPLL  
**Description:** A short exists between internal test points within the frequency modulation (FM) portion of the PLL. The spacing between the shorted test points was not sufficient, preventing FM from operating as expected. This results in the PLL not locking after FM is enabled.

**Workaround:** No workaround.

**Fix plan:** Fixed in datecodes XX0511 and later.

**SECF131: PLL Does Not Lock in Normal PLL Mode with External Clock Reference**

**Errata type:** Silicon  
**Affects:** PLL  
**Description:** During a power on reset, if the CLKMOD[1:0] equals 10 setting is used (normal PLL mode with external clock reference), the PLL does not lock and the device never comes out of reset.

**Workaround:**

**NOTE**
If a workaround for errata SECF163 is implemented, a workaround for this errata is not necessary.

When configuring the PLL for normal PLL mode with external clock reference, tie CLKMOD1 to RESET and not straight to 3.3V. This allows the PLL to correctly detect the desired operating mode and lock.

**Fix plan:** Currently, there are no plans to fix this.
SECF163: Some Devices May Not Exit Reset After Power On

Errata type: Silicon
Affects: PLL
Description: During power-on reset (POR), the frequency of the PLL’s internal oscillator (ICO) may overrun the ability of the ICO’s internal feedback to track to that frequency. This results in an open-loop condition where the frequency of the ICO remains at its upper limit. The internal feedback loop recovers, but might not be able to reach the target frequency and exit reset. This affects all PLL-enabled modes, but does not affect bypass mode.

Workaround: After RESET deasserts and before RSTOUT deasserts, change the target clock mode to a PLL-enabled mode in which the input clock is disabled for more than 160us. Then, change back to the target clock mode.

<table>
<thead>
<tr>
<th>Start in</th>
<th>CLK MOD</th>
<th>After RESET deasserts and before RSTOUT deasserts, switch to</th>
<th>CLK MOD</th>
<th>After 160us, switch back to the target clock mode</th>
<th>CLK MOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal mode with crystal reference</td>
<td>11</td>
<td>Normal mode with external reference</td>
<td>10</td>
<td>Normal mode with crystal reference</td>
<td>11</td>
</tr>
<tr>
<td>Normal mode with external reference</td>
<td>10</td>
<td>Normal mode with crystal reference</td>
<td>11</td>
<td>Normal mode with external reference</td>
<td>10</td>
</tr>
<tr>
<td>1:1 mode</td>
<td>01</td>
<td>Normal mode with crystal reference</td>
<td>11</td>
<td>1:1 mode</td>
<td>01</td>
</tr>
</tbody>
</table>

Workaround: At POR, start in a PLL-enabled mode in which the input clock is disabled. Keep RESET asserted for at least 160us. After RESET deasserts and before RSTOUT deasserts change to the desired clock mode.

<table>
<thead>
<tr>
<th>Start in</th>
<th>CLK MOD</th>
<th>After RESET deasserts and before RSTOUT deasserts, switch to target clock mode</th>
<th>CLK MOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal mode with external reference</td>
<td>10</td>
<td>Normal mode with crystal reference</td>
<td>11</td>
</tr>
<tr>
<td>Normal mode with crystal reference</td>
<td>11</td>
<td>Normal mode with external reference</td>
<td>10</td>
</tr>
<tr>
<td>Normal mode with crystal reference</td>
<td>11</td>
<td>1:1 mode</td>
<td>01</td>
</tr>
</tbody>
</table>

Fix plan: Fixed on datecodes XXXX1031 and later.