Mask Set Errata for Mask 1N15J

Introduction
This report applies to mask 1N15J for these products:
  • KINETIS_L

<table>
<thead>
<tr>
<th>Errata ID</th>
<th>Errata Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>6946</td>
<td>Core: A debugger write to the I/O port might be corrupted during a processor write</td>
</tr>
<tr>
<td>6945</td>
<td>Core: Processor executing at HardFault priority might enter Lockup state if an NMI occurs during a waited debugger transaction</td>
</tr>
<tr>
<td>6395</td>
<td>MCG: Fast IRC Fine Trim bit is not used by the MCG Auto Trim Machine</td>
</tr>
</tbody>
</table>

**e6946: Core: A debugger write to the I/O port might be corrupted during a processor write**

**Errata type:** Errata  
**Description:** A debugger can perform memory accesses through the Cortex-M0+ processor bus matrix while the processor is running.  
Because of this erratum, a debugger write to the I/O port might be corrupted if it occurs while the processor is executing a write. The processor write completes successfully. However, under specific timing conditions, the matrix might incorrectly replace the debugger write data with the value zero.  
This erratum does not affect debugger writes outside the I/O port region of the memory map, or debugger reads.  
**Conditions:**  
The following timing-specific conditions must all be met:  
• The processor is running (not halted in Debug state).  
• The debugger performs a write within the I/O port region of the memory map.  
• The processor performs a write.
Implications:
The debugger might corrupt the targeted memory or configure the targeted device incorrectly.

Workaround: The debugger can work around this erratum by halting the processor in Debug state before performing writes to the I/O port region of the memory map.

e6945: Core: Processor executing at HardFault priority might enter Lockup state if an NMI occurs during a waited debugger transaction

Errata type: Errata
Description: A debugger can perform memory accesses through the Cortex-M0+ processor bus matrix while the processor is running.

Because of this erratum, the processor might erroneously enter Lockup state if a debugger-initiated access on the AHB-Lite master port is subject to wait states while the processor is running, executing at HardFault priority and taking a Non Maskable Interrupt (NMI). Under very specific timing conditions, the processor might incorrectly stack a ReturnAddress of 0xFFFFFFFF on NMI entry. On NMI return, the processor unstacks the incorrectly stacked ReturnAddress and enters Lockup state at HardFault priority.

Conditions:
The following timing-specific conditions must all be met:
• The processor is running (not halted in Debug state) and is executing at HardFault priority.
• The processor executes a single-cycle instruction at a word-aligned address.
• The debugger performs an access through the AHB-Lite master port that is subject to wait states.
• An NMI becomes pending.

Implications:
The processor stops executing the code in the HardFault handler and enters Lockup state at HardFault priority as if a fault had occurred.

Workaround: The debugger can work around this erratum by halting the processor in Debug state before performing accesses outside the Private Peripheral Bus (PPB) region of the memory map.

e6395: MCG: Fast IRC Fine Trim bit is not used by the MCG Auto Trim Machine

Errata type: Errata
Description: The Fast IRC fine trim bit, MCG_C2[FCFTRIM], is not used by the Auto Trim Machine. This means that when the Auto Trim Machine is used to set the Fast IRC frequency, it will not be trimmed with the finest resolution. If the finer resolution is required, the MCG_C2[FCFTRIM] bit must be manually changed and the resulting fast IRC frequency must then be measured to determine whether this bit should be set or cleared to be closest to the target frequency.

Workaround: If the finest resolution of the Fast IRC trim is not required, then no workaround is required. However, if the finest resolution is required, the following steps must be performed:
1) After the auto trim machine has completed running, measure the Fast IRC frequency.
2) Clear the MCG_C2[FCFTRIM] and re-measure the Fast IRC frequency.
3) Determine which value of MCG_C2[FCFTRIM] provided the closest frequency to the desired target frequency.

4) Store the MCG_C4[FCTRIM] and MCG_C2[FCFTRIM] values in a suitable flash location. Refer to the individual device reference manual for the recommended flash location.