

DSP56311 Device Errata for Mask 0K92M

General remark: In order to prevent the use of instructions or sequences of instructions that do not operate correctly, we encourage you to use the “lint563” program to identify such cases and use alternative sequences of instructions. This program is available as part of the Freescale DSP Tools CLAS package.

Silicon Errata

	<u>Errata Description</u>	<u>Applies to Mask</u>
ES131	<p>Description (added 11/28/00):</p> <p>When all of the following conditions are true:</p> <ol style="list-style-type: none"> 1. Executing a conditional change of flow (branch, jump) instruction and the branch or jump is <i>not</i> taken, AND 2. One of the two program memory words following that instruction includes an address or displacement (as a whole word, or as a field), AND 3. DMA transactions <i>are</i> occurring simultaneously, <p>The address or displacement might not be calculated correctly.</p> <p>Workaround: Do one of the following:</p> <ol style="list-style-type: none"> 1. If only the second word after the instruction includes the address or displacement, add one NOP after the conditional branch, or, 2. If the first word after the instruction includes the address or displacement, add two NOPs after the conditional branch. 	2J23D

Documentation Errata

<p>ED1</p>	<p>Description (revised 11/9/98):</p> <p>XY memory data move does not work properly under one of the following two situations:</p> <ol style="list-style-type: none"> 1. The X-memory move destination is internal I/O and the Y-memory move source is a register used as destination in the previous adjacent move from non Y-memory 2. The Y-memory move destination is a register used as source in the next adjacent move to non Y-memory. <p>Here are examples of the two cases (where x:(r1) is a peripheral):</p> <p>Example 1:</p> <pre>move #\$12, y0 move x0, x:(r7) y0, y:(r3) (while x:(r7) is a peripheral).</pre> <p>Example 2:</p> <pre>mac x1, y0, a x1, x:(r1)+ y:(r6)+, y0 move y0, y1</pre> <p>Any of the following alternatives can be used:</p> <ol style="list-style-type: none"> 1. Separate these two consecutive moves by any other instruction. 2. Split XY Data Move to two moves. <p>Pertains to: DSP56300 Family Manual, Section B-5 “Peripheral pipeline restrictions.</p>	<p>2J23D</p>
<p>ED3</p>	<p>Description (added before 2/18/1996):</p> <p>$\overline{\text{BL}}$ pin timings T198 and T199 in the data sheet are changed, improving the arbitration latency: T198 is 5 ns (max), T199 is 0 ns (min).</p> <p>Pertains to: Data Sheet, Synchronous Timings (SRAM) table, Table 2-17.</p>	<p>2J23D</p>
<p>ED7</p>	<p>Description (added 1/27/98):</p> <p>When activity is passed from one DMA channel to another and the DMA interface accesses external memory (which requires one or more wait states), the DACT and DCH status bits in the DMA Status Register (DSTR) may indicate improper activity status for DMA Channel 0 (DACT = 1 and DCH[2:0] = 000).</p> <p>Workaround: None.</p> <p>This is not a bug, but a specification update.</p>	<p>2J23D</p>

<p>ED9</p>	<p>Description (added 1/27/98):</p> <p>When the SCI is configured in Synchronous mode, internal clock, and all the SCI pins are enabled simultaneously, an extra pulse of 1 DSP clock length is provided on the SCLK pin.</p> <p>Workaround:</p> <ol style="list-style-type: none"> 1. Enable an SCI pin other than SCLK. 2. In the next instruction, enable the remaining SCI pins, including the SCLK pin. <p>This is not a bug, but a specification update.</p>	<p>2J23D</p>
<p>ED15</p>	<p>Description (added 7/21/98):</p> <p>The DRAM Control Register (DCR) should not be changed while refresh is enabled. If refresh is enabled only a write operation that disables refresh is allowed.</p> <p>Workaround: First disable refresh by clearing the BREN bit, than change other bits in the DCR register, and finally enable refresh by setting the BREN bit.</p>	<p>2J23D</p>
<p>ED17</p>	<p>Description (added 9/28/98):</p> <p>In all DSP563xx technical data sheets, a note is to be added under “AC Electrical Characteristics” that although the minimum value for “Frequency of Extal” is 0MHz, the device AC test conditions are 15MHz and rated speed.</p> <p>Workaround: N/A</p>	<p>2J23D</p>
<p>ED20</p>	<p>Description (added 11/24/98):</p> <p>In the Technical Datasheet Voh-TTL should be listed at 2.4 Volts, not as:</p> $TTL = V_{CC} - 0.4$ <p>Workaround: This is a documentation update.</p>	<p>2J23D</p>
<p>ED24</p>	<p>Description (added 11/24/98):</p> <p>The technical datasheet supplies a maximum value for internal supply current in Normal, Wait, and Stop modes. These values will be removed because we will specify only a “Typical” current.</p> <p>Workaround: This is a documentation update.</p>	<p>2J23D</p>
<p>ED26</p>	<p>Description (added 1/6/99):</p> <p>The specification DMA Chapter is wrong.</p> <p>“Due to the DSP56300 Core pipeline, after DE bit in DCRx is set, the corresponding DTDx bit in DSTR will be cleared only after two instruction cycles.”</p> <p>Should be replaced with:</p> <p>“Due to the DSP56300 Core pipeline, after DE bit in DCRx is set, the corresponding DTDx bit in DSTR will be cleared only after three instruction cycles.”</p>	<p>2J23D</p>

<p>ED28</p>	<p>Description (added 1/7/1997; identified as Documentation Errata 2/1/99):</p> <p>When two consecutive LAs have a conditional branch instruction at LA-1 of the internal loop, the part does not operate properly. For example, the following sequence may generate incorrect results:</p> <pre> DO #5, LABEL1 NOP DO #4, LABEL2 NOP MOVE (R0) + BSCC _DEST ; conditional branch at LA-1 of internal loop NOP ; internal LA LABEL2 NOP ; external LA LABEL1 NOP NOP _DEST NOP NOP RTS </pre> <p>Workaround: Put an additional NOP between LABEL2 and LABEL1.</p> <p>Pertains to: DSP56300 Family Manual, Appendix B, Section B-4.1.3, “At LA-1.”</p>	<p>2J23D</p>
<p>ED29</p>	<p>Description (added 9/12/1997; identified as a Documentation errata 2/1/99):</p> <p>When the ESSI transmits data with the CRA Word Length Control bits (WL[2:0]) = 100, the ESSI is designed to duplicate the last bit of the 24-bit transmission eight times to fill the 32-bit shifter. Instead, after shifting the 24-bit word correctly, eight 0s are being shifted.</p> <p>Workaround: None.</p> <p>Pertains to: UM, Section 7.4.1.7, “CRA Word Length Control.” The table number is 7-2.</p>	<p>2J23D</p>
<p>ED30</p>	<p>Description (added 9/12/1997; identified as a Documentation errata 2/1/99):</p> <p>When the ESSI transmits data in the On-Demand mode (i.e., MOD = 1 in CRB and DC[4:0] = \$00000 in CRA) with WL[2:0] = 100, the transmission does not work properly.</p> <p>Workaround: To ensure correct operation, do not use the On-Demand mode with the WL[2:0] = 100 32-bit Word-Length mode.</p> <p>Pertains to: UM, Section 7.5.4.1, “Normal/On-Demand Mode Selection.”</p>	<p>2J23D</p>

<p>ED31</p>	<p>Description (added 9/12/1997; modified 9/15/1997; identified as a Documentation errata 2/1/99):</p> <p>Programming the ESSI to use an internal frame sync (i.e., SCD2 = 1 in CRB) causes the SC2 and SC1 signals to be programmed as outputs. If however, the corresponding multiplexed pins are programmed by the Port Control Register (PCR) to be GPIOs, then the GPIO Port Direction Register (PRR) chooses their direction, but this causes the ESSI to use an external frame sync if GPIO is selected.</p> <p style="text-align: center;">Note: This errata and workaround apply to both ESSI0 and ESSI1.</p> <p>Workaround: To assure correct operation, either program the GPIO pins as outputs or configure the pins in the PCR as ESSI signals.</p> <p style="text-align: center;">Note: The default selection for these signals after reset is GPIO.</p> <p>Pertains to: UM, Section 7.4.2.4, “CRB Serial Control Direction 2 (SCD2) Bit 4”</p>	<p>2J23D</p>
<p>ED32</p>	<p>Description (added 11/9/98; identified as a Documentation errata 2/1/99):</p> <p>When returning from a long interrupt (by RTI instruction), and the first instruction after the RTI is a move to a DALU register (A, B, X, Y), the move may not be correct, if the 16-bit arithmetic mode bit (bit 17 of SR) is changed due to the restoring of SR after RTI.</p> <p>Workaround: Replace the RTI with the following sequence:</p> <pre style="margin-left: 40px;"> movec ssl, sr nop rti </pre> <p>Pertains to: DSP56300 Family Manual. Add a new section to Appendix B that is entitled “Sixteen-Bit Compatibility Mode Restrictions.”</p>	<p>2J23D</p>

<p>ED33</p>	<p>Description (added 12/16/98; identified as a Documentation errata 2/1/99):</p> <p>When Stack Extension mode is enabled, a use of the instructions BRKcc or ENDDO inside do loops might cause an improper operation.</p> <p>If the loop is non nested and has no nested loop inside it, the errata is relevant only if LA or LC values are being used outside the loop.</p> <p>Workaround:</p> <p>If Stack Extension is used, emulate the BRKcc or ENDDO as in the following examples. We split between two cases, finite loops and do forever loops.</p> <p>1) Finite DO loops (i.e. not DO FOREVER loops)</p> <p>=====</p> <p>BRKcc</p> <p>Original code:</p> <pre> do #N, label1 do #M, label2 BRKcc label2 label1 </pre> <p>Will be replaced by:</p> <pre> do #N, label1 do #M, label2 Jcc fix_brk_routine </pre>	<p>2J23D</p>
--------------------	--	--------------

<p>ED33 cont.</p>	<pre> nop_before_label2 nop ; This instruction must be NOP. label2 label1 fix_brk_routine move #1,lc jmp nop_before_label2 ENDDO ----- Original code: do #M,label1 do #N,label2 ENDDO label2 label1 Will be replaced by: do #M, label1 do #N, label2 JMP fix_enddo_routine </pre>	<p>2J23D</p>
--------------------------	---	--------------

<p>ED33 cont.</p>	<pre> nop_after_jump NOP ; This instruction must be NOP. label2 label1 fix_enddo_routine move #1,lc move #nop_after_jump,la jmp nop_after_jump 2) DO FOREVER loops ===== BRKcc ----- Original code: do #M,label1 do forever,label2 BRKcc label2 label1 </pre>	<p>2J23D</p>
--------------------------	--	--------------

<p>ED33 cont.</p>	<p>Will be replaced by:</p> <pre> do #M,label1 do forever,label2 JScC fix_brk_forever_routine ; <--- note: JScC and not Jcc nop_before_label2 nop ; This instruction must be NOP. label2 label1 fix_brk_forever_routine move ssh,x:<..> ; <..> is some reserved not used address (for temporary data) move #nop_before_label2,ssh bclr #16,ssl ; move #1,lc rti ; <---- note: "rti" and not "rts" ! ENDDO ----- Original code: do #M,label1 </pre>	<p>2J23D</p>
--------------------------	---	--------------

<p>ED33 cont.</p>	<pre> do forever,label2 ENDDO label2 label1 Will be replaced by: do #M,label1 do forever,label2 JSR fix_enddo_routine ; <--- note: JSR and not JMP nop_after_jump NOP ; This instruction should be NOP label2 label1 fix_enddo_routine nop move #1,lc bclr #16,ssl move #nop_after_jump,la rti ; <--- note: "rti" and not "rts" </pre> <p>Pertains to: DSP56300 Family Manual, Section B-4.2, “General Do Restrictions.”</p>	<p>2J23D</p>
--------------------------	---	--------------

<p>ED34</p>	<p>Description (added 1/5/99; identified as a Documentation errata 2/1/99):</p> <p>When stack extension is enabled, the read result from stack may be improper if two previous executed instructions cause sequential read and write operations with SSH. Two cases are possible:</p> <p>Case 1:</p> <p>For the first executed instruction: move from SSH or bit manipulation on SSH (i.e. jclr, brclr, jset, brset, btst, bsset, jsset, bsclr, jsclr).</p> <p>For the second executed instruction: move to SSH or bit manipulation on SSH (i.e. jsr, bsr, jscc, bscc).</p> <p>For the third executed instruction: an SSL or SSH read from the stack result may be improper - move from SSH or SSL or bit manipulation on SSH or SSL (i.e., bset, bclr, bchg, jclr, brclr, jset, brset, btst, bsset, jsset, bsclr, jsclr).</p> <p>Workaround:</p> <p>Add two NOP instructions before the third executed instruction.</p> <p>Case 2:</p> <p>For the first executed instruction: bit manipulation on SSH (i.e. bset, bclr, bchg).</p> <p>For the second executed instruction: an SSL or SSH read from the stack result may be improper - move from SSH or SSL or bit manipulation on SSH or SSL (i.e., bset, bclr, bchg, jclr, brclr, jset, brset, btst, bsset, jsset, bsclr, jsclr).</p> <p>Workaround:</p> <p>Add two NOP instructions before the second executed instruction.</p> <p>Pertains to: DSP56300 Family Manual, Appendix B, add a new section called “Stack Extension Enable Restrictions.” Cover all cases. Also, in Section 6.3.11.15, add a cross reference to this new section.</p>	<p>2J23D</p>
<p>ED38</p>	<p>Description (added 7/14/99):</p> <p>If Port A is used for external accesses, the BAT bits in the AAR3-0 registers must be initialized to the SRAM access type (i.e. BAT = 01) or to the DRAM access type (i.e. BAT = 10). To ensure proper operation of Port A, this initialization must occur even for an AAR register that is not used during any Port A access. Note that at reset, the BAT bits are initialized to 00.</p> <p>Pertains to: <i>DSP56300 Family Manual</i>, Port A Chapter (Chapter 9 in Revision 2), description of the BAT[1 –0] bits in the AAR3 - AAR0 registers. Also pertains to the core chapter in device-specific user’s manuals that include a description of the AAR3 - AAR0 registers with bit definitions (usually Chapter 4).</p>	<p>2J23D</p>

<p>ED40</p>	<p>Description (added 11/11/99):</p> <p>When an instruction with all the following conditions follows a repeat instruction, then the last move will be corrupted.:</p> <ol style="list-style-type: none"> 1. The repeated instruction is from external memory. 2. The repeated instruction is a DALU instruction that includes 2 DAL registers, one as a source, and one as destination (e.g. tfr, add). 3. The repeated instruction has a double move in parallel to the DALU instruction: one move's source is the destination of the DALU instruction (causing a DALU interlock); the other move's destination is the source of the DALU instruction. <p>Example:</p> <pre> rep #number tfr x0,a x(r0)+,x0 a,y0 ; This instruction is from external memory _ _____ ------ -----> This is condition 3 second part. _____ ------> This is condition 3, first part - DALU interlock </pre> <p>In this example, the second iteration before the last, the "x(r0)+,x0" doesn't happen. On the first iteration before the last, the X0 register is fixed with the "x(r0)+,x0", but the "tfr x0,a" gets the wrong value from the previous iteration's X0. Thus, at the last iteration the A register is fixed with "tfr x0,a", but the "a,y0" transfers the wrong value from the previous iteration's A register to Y0.</p> <p>Workaround:</p> <ol style="list-style-type: none"> 1. Use the DO instruction instead; mask any necessary interrupts before the DO. 2. Run the REP instructions from internal memory. 3. Don't make DALU interlocks in the repeated instruction. After the repeat make the move. In the example above, all the "move a,y0" are redundant so it can be done in the next instruction: <pre> rep #number tfr x0,a x(r0)+,x0 move a,y0 </pre> <p>If no interrupts before the move is a must, mask the interrupts before the REP.</p> <p>Pertains to: <i>DSP56300 Family Manual</i>, Rev. 2, Section A.3, "Instruction Sequence Restrictions."</p>	<p>2J23D</p>
--------------------	---	--------------

<p>ED42</p>	<p>Description (added on 3/22/2000)</p> <p>Revised 10/29/2004.</p> <p>When the device is operating in a mode in which DE is not cleared at the end of the block transfer (DTM = 100 or 101), the DMA end-of-block-transfer interrupt may not be latched when the external bus arbitration controller asserts the bus grant/BG pin. This causes the end-of-block-transfer interrupt to be lost.</p> <p>Pertains to:</p> <p><i>DSP56300 Family Manual, Rev. 2, Section 10.4.1.2, “End-of-Block-Transfer Interrupt.” Also, Section 10.5.3.5, “DMA Control Registers (DCR[5–0],” discussion of bits 21 – 19 (DTM bits).</i></p>	<p>2J23D</p>
<p>ED43</p>	<p>Description (added on 7/6/2000)</p> <p>Two HI08 registers, the Host Command Vector Register (HCVR) and the Host Interrupt Vector Register (HIVR), do not have a known value at reset. That is, their initial value after reset is unknown.</p> <p>Workaround :In order to use these registers, the user must write a value to these registers.</p>	<p>2J23D</p>
<p>ED44</p>	<p>Description (added 3/31/2001):</p> <p>When used with MS=1, a very small fraction of devices may seem to have bad memory bits, although the problem is actually with the busses used in the switch mode and/or the logic that enables them. This impacts all mask revisions of the DSP56311. When used with MS=0, this problem is not present.</p> <p>Diagnostic: With the mode set for maximum program memory, write and read both 1’s and 0’s to each bit of extended program memory. Repeat the process via both core and DMA accesses. If possible, the screen should be performed at the highest temperature and lowest voltage the parts will see in the application. If high temperature testing is difficult, the following procedure can be followed:</p> <ol style="list-style-type: none"> 1. run the test at room temperature, 10 MHz, and 1.9V 2. run the test at room temperature, (Fmax + 6) MHz, and 1.7V 3. confirm any failures at spec temperature, speed and voltage <p>Workaround: The production test program has been modified to address the issue. This test has been run on all material with date codes of WW10, 2001 and later, and for all material with a drypack seal date of March 8, 2001 or later.</p>	<p>2J23D</p>
<p>ED47</p>	<p>Description: (added 1/19/2002):</p> <p>Reclassified from ES135 to ED47 (spec change) on 5/9/2002:</p> <p>When DMA line-by-line block transfers are used with the EFCOP to perform IIR filtering with two or fewer IIR coefficients, the first output of the IIR filter is lost. The rest of the outputs are shifted and inaccurate.</p> <p>Workaround: Instead of DMA line-by-line block transfers, use DMA word-by-word block transfers.</p>	<p>2J23D</p>

ED50	Description (added 9/10/1996 as ES29; reclassified as a documentation erratum on 8/2/2002): When the SCI transmitter is used in Synchronous mode, the last bit of the transmitted byte might be truncated to the half of the serial cycle. Workaround: Not available.	2J23D
-------------	---	-------

NOTES

1. An over-bar (i.e., $\overline{\text{xxxx}}$) indicates an active-low signal.
2. The letters in the right column tell which DSP56321 mask numbers apply.

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations not listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

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

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.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T. Hong Kong
+800 2666 8080

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.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 which 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.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. StarCore is a trademark of StarCore LLC. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2004.