Freescale Semiconductor
Engineering Bulletin

EB338
Rev. 3, 12/2005

Functional Differences Between Masks 3F48S and 2K30A of the DSP56301

This document describes the differences between masks of the DSP56301: mask 2K30A and the preceding mask 3F48S. The newer 2K30A mask uses the communications design rules (CDRII) process. The 2K30A mask set also has new I/O and a new PLL, with the requisite change in the PLL capacitor equation (PCAP).

CONTENTS

1 2K30A Mask Set

The 2K30A mask set replaced the 3F48S mask set in production.

2 PLL Input Capacitor (C_{PCAP})

The process change results in a changed requirement for computing the size of C_{PCAP}, the capacitor used with the PCAP input. **Table 1** lists the new formulas for computing the value of this input capacitor for the DSP.

3 ESD Protection

The 2K30A mask set incorporates improved I/O structures to ensure Electro Static Discharge (ESD) protection to qualification levels of 2000 volts Human Body Model (HBM) and 200 volts Machine Model (MM).

4 Differences Overview

The primary functional differences between the mask sets are due to inherent differences between the two design technologies. **Table 1** compares the mask sets.

Table 1. Functional Comparison of 3F48S and 2K30A

Feature	**3F48S**			**2K30A**		
	Recommended	**Min**	**Max**	**Recommended**	**Min**	**Max**
Technology	0.5 micron	—	—	Sub 0.4 micron	—	—
PLL input capacitor (C_{PCAP})	Uses the following rules: For MF ≤4: C_{PCAP} = [(500 × MF) – 150] pF For MF > 4: C_{PCAP} = (690 × MF) pF	(MF × 425) – 125 MF × 520	(MF × 590) – 175 MF × 920	Use the following rules: For MF ≤4: C_{PCAP} = [(680 × MF) – 120] pF For MF > 4: C_{PCAP} = (1100 × MF) pF	(MF × 580) – 100 MF × 830	(MF × 780) – 140 MF × 1470

5 Change in Substrate Design

In the DSP56301 technical data sheet, several pins are designated as “not connected” (NC). These pins are reserved for future development. Do not connect any line, component, or trace to these pins. In the new substrate, the following groups of NC pins are shorted to each other and retain their NC status:

- pins A2 and B[1–B2] are shorted together
- pins A15, B[15–16], C[14–16], and D14 are shorted together
- pins N3, R[1–2], and T2 are shorted together
- pins N16, P13, P15, R[15–16], and T15 are shorted together

6 Asynchronous Bus Arbitration

In the Operating Mode Register (OMR) of the 2K30A is a new Asynchronous Bus Arbitration (ABE) bit to support external access at high frequencies. As **Figure 1** shows, bit 13 in the OMR enables Asynchronous Bus Arbitration. For convenience, **Table 2** defines all the OMR bits available in the 2K30A mask set of the DSP56301. The new ABE bit is highlighted in a bold red font.

SCS								EOM								COM							
23	22	21	20	19	18	17	16	15	14	**13**	12	11	10	9	8	7	6	5	4	3	2	1	0
▨	▨	▨	SEN	WRP	EOV	EUN	XYS	ATE	APD	**ABE**	BRT	TAS	BE	CDP1:0		MS	SD	▨	EBD	MD	MC	MB	MA

SEN—Stack Extension Enable

WRP—Extended Stack Wrap Flag

EOV—Extended Stack Overflow Flag

EUN—Extended Stack Underflow Flag

XYS—Stack Extension Space Select

ATE—Address Trace Enable

APD—Address Attribution Priority Disable

ABE—Async. Bus Arbitration Enable

BRT—Bus Release Timing

TAS—$\overline{\text{TA}}$ Synchronize Select

BE—Burst Mode Enable

CDP1—Core-DMA Priority 1

CDP0—Core-DMA Priority 0

MS—Memory Switch Mode

SD—Stop Delay

EBD—External Bus Disable

MD—Operating Mode D

MC—Operating Mode C

MB—Operating Mode B

MA—Operating Mode A

▨ — Reserved bit; read as zero; should be written with zero for future compatibility

Figure 1. Operating Mode Register (OMR)

Table 2. Operating Mode Register (OMR) Bit Definitions

Bit Number	Bit Name	Reset Value	Description
23–21	▨	0	Reserved. Set to 0 for future compatibility.
20	SEN	0	**Stack Extension Enable** Enables/disables the stack extension in data memory. If the SEN bit is set, the extension is enabled. Hardware reset clears this bit, so the default out of reset is a disabled stack extension.
19	WRP	0	**Stack Extension Wrap Flag** Set when copying from the on-chip hardware stack (System Stack Register file) to the stack extension memory begins. You can use this flag during the debugging phase of the software development to evaluate and increase the speed of software-implemented algorithms. The WRP flag is a *sticky bit* (that is, cleared only by hardware reset or by an explicit MOVE operation to the OMR).
18	EOV	0	**Stack Extension Overflow Flag** Set when a stack overflow occurs in Stack Extended mode. Extended stack overflow is recognized when a push operation is requested while SP = SZ (Stack Size register), and the Extended mode is enabled by the SEN bit. The EOV flag is a *sticky bit* (that is, cleared only by hardware reset or by an explicit MOVE operation to the OMR). The transition of the EOV flag from zero to one causes a Priority Level 3 (Non-maskable) stack error exception.

Table 2. Operating Mode Register (OMR) Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
17	EUN	0	**Stack Extension Underflow Flag** Set when a stack underflow occurs in Extended Stack mode. Extended stack underflow is recognized when a pull operation is requested, SP = 0, and the SEN bit enables Extended mode. The EUN flag is a *sticky bit* (that is, cleared only by hardware reset or by an explicit MOVE operation to the OMR). Transition of the EUN flag from zero to one causes a Priority Level 3 (Non-maskable) stack error exception. NOTE: While the chip is in Extended Stack mode, the UF bit in the SP acts like a normal counter bit.
16	XYS	0	**Stack Extension XY Select** Determines whether the stack extension is mapped onto X or Y memory space. If the bit is clear, then the stack extension is mapped onto the X memory space. If the XYS bit is set, the stack extension is mapped to the Y memory space.
15	ATE	0	**Address Trace Enable** When set, the Address Trace Enable (ATE) bit enables Address Trace mode. The Address Trace mode is a debugging tool that reflects internal memory accesses at the external address lines.
14	APD	0	**Address Attribution Priority Disable** Disables the priority assigned to the Address Attribute signals (AA0-AA3). When APD = 0 (default setting), the four Address Attribute signals each have a certain priority: AA3 has the highest priority, AA0 has the lowest priority. Therefore, only one AA signal can be active at one time. This allows continuous partitioning of external memory; however, certain functions, such as using the AA signals as additional address lines, require the use of additional interface hardware. When APD is set, the priority mechanism is disabled, allowing more than one AA signal to be active simultaneously. Therefore, the AA signals can be used as additional address lines without the need for additional interface hardware.
13	**ABE**	**0**	**Asynchronous Bus Arbitration Enable** **Eliminates the setup and hold time requirements (with respect to CLKOUT) for $\overline{BB}$ and $\overline{BG}$, and substitutes a required non-overlap interval between the deassertion of one $\overline{BG}$ input to a DSP56300 family device and the assertion of a second BG input to a second DSP56300 family device on the same bus. When the ABE bit is set, the $\overline{BG}$ and $\overline{BB}$ inputs are synchronized. This synchronization causes a delay between a change in $\overline{BG}$ or $\overline{BB}$ until this change is actually accepted by the receiving device.**
12	BRT	0	**Bus Release Timing** Selects between fast or slow bus release. If BRT is cleared, a Fast Bus Release mode is selected (that is, no additional cycles are added to the access and BB is not guaranteed to be the last Port A pin that is tri-stated at the end of the access). If BRT is set, a Slow Bus Release mode is selected (that is, an additional cycle is added to the access, and BB is the last Port A pin that is tri-stated at the end of the access).
11	TAS		**TA Synchronize Select** Selects the synchronization method for the input Port A pin—$\overline{TA}$ (Transfer Acknowledge). If TAS is cleared, you are responsible for asserting the $\overline{TA}$ pin in synchrony with the chip clock, as described in the technical data sheet. If TAS is set, the $\overline{TA}$ input pin is synchronized inside the chip, thus eliminating the need for an off-chip synchronizer. Note that the TAS bit has no effect when the $\overline{TA}$ pin is deasserted: you are responsible for deasserting the **$\overline{TA}$** pin in synchrony with the chip clock, regardless of the value of TAS.
10	BE	0	**Cache Burst Mode Enable** Enables/disables Burst mode in the memory expansion port during an instruction cache miss. If the bit is cleared, Burst mode is disabled and only one program word is fetched from the external memory when an instruction cache miss condition is detected. If the bit is set, Burst mode is enabled, and up to four program words are fetched from the external memory when an instruction cache miss is detected.

Table 2. Operating Mode Register (OMR) Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
9–8	CPD	1	**Core-DMA Priority** Specify the priority of core and DMA accesses to the external bus. 00 Determined by comparing status register CP[1:0] to the active DMA channel priority 01 DMA accesses have higher priority than core accesses 10 DMA accesses have the same priority as the core accesses 11 DMA accesses have lower priority than the core accesses
7	MS	0	**Memory Switch Mode** Allows some internal data memory (X, Y, or both) to become part of the chip internal Program RAM. **Note:** **1.** Program data placed in the Program RAM/Instruction Cache area changes its placement after the MS bit is set (that is, the Instruction Cache always uses the highest internal Program RAM addresses). **2.** To ensure proper operation, place six NOP instructions after the instruction that changes the MS bit. **3.** To ensure proper operation, do not set the MS bit while the Instruction Cache is enabled (CE bit is set in SR).
6	SD	0	**Stop Delay Mode** Determines the length of the delay invoked when the core exits the Stop state. The STOP instruction suspends core processing indefinitely until a defined event occurs to restart it. If SD is cleared, a 128K clock cycle delay is invoked before a STOP instruction cycle continues. However, if SD, the delay before the instruction cycle continues is 16 clock cycles. The long delay allows a clock stabilization period for the internal clock to begin oscillating and to stabilize. When a stable external clock is used, the shorter delay allows faster start-up of the DSP56300 core.
5		0	Reserved. Set to 0 for future compatibility.
4	EBD	0	**External Bus Disable** Disables the external bus controller to reduce power consumption when external memories are not used. When EBD is set, the external bus controller is disabled and external memory cannot be accessed. When EBD is cleared, the external bus controller is enabled and external access can be performed. Hardware reset clears the EBD bit.
3–0	MD–MA		**Chip Operating Mode** Indicate the operating mode of the DSP56300 core. On hardware reset, these bits are loaded from the external mode select pins, MODD, MODC, MODB, and MODA, respectively. After the DSP56300 core leaves the Reset state, MD, MC, MB, and MA can be changed under program control.

7 Boot Mode Changes

The 2K30A mask set of the DSP56301 modifies the boot modes of the microprocessor in order to add the ability to boot from a serial EPROM (see **Table 3**). Adding this new booting capability requires that MODD no longer be a “don’t care.” The MODD column of **Table 3** shows the correct MODD setting for each boot mode. These settings are communicated to customers in Product Change Notice (PCN) No. 4302. Changes in boot mode that occur with 2K30A appear in bold red font.

Table 3. New DSP56301 Boot Modes for CDR2

Boot Mode	MODD	3F48S	2K30A
0	0	expanded mode	expanded mode
1	**0**	bootstrap from byte-wide memory	**reserved**
2	0	bootstrap through SCI	bootstrap through SCI
3	**0**	reserved	reserved **(burn-in)**
4	**0**	host bootstrap PCI mode (32 bit wide)	**bootstrap from serial EEPROM**
5	**0**	host bootstrap 16-bit wide UB mode (ISA)	**reserved**
6	0	host bootstrap 8-bit wide UB mode in double-strobe pin configuration	host bootstrap 8-bit wide UB mode in double-strobe pin configuration
7	**0**	host bootstrap 8-bit wide UB mode in single-strobe pin configuration	**reserved**
8	1	expanded mode	expanded mode
9	1	bootstrap from byte-wide memory	bootstrap from byte-wide memory
A	1	bootstrap through SCI	bootstrap through SCI
B	1	host bootstrap in onyx-to-onyx mode	host bootstrap in onyx-to-onyx mode
C	1	host bootstrap PCI mode (32-bit wide)	host bootstrap PCI mode (32 bit wide)
D	1	host bootstrap 16-bit wide UB mode (ISA)	host bootstrap 16-bit wide UB mode (ISA)
E	1	host bootstrap 8-bit wide UB mode in double-strobe pin configuration	host bootstrap 8-bit wide UB mode in double-strobe pin configuration
F	1	host bootstrap 8-bit wide UB mode in single-strobe pin configuration	host bootstrap 8-bit wide UB mode in single-strobe pin configuration

8 Identification Registers

Values in the Device Identification register (IDR) and JTAG Identification (ID) register are changed to reflect the new 2K30A mask set, as shown in **Figure 2**, **Figure 3**, and **Figure 4**.

The IDR is a 24-bit, read-only factory-programmed register that identifies DSP56300 family members. It specifies the derivative number and revision number of the device. This information is used in testing or by software. **Figure 2** shows the contents of the IDR for mask set 2K30A. Revision numbers are assigned as follows: $0 is revision 0, $1 is revision A, and so on.

23 16	15 12	11 0
Reserved	**Revision Number**	**Derivative Number**
0000 0000	0011	0011 0000 0001

Figure 2. Identification Register Configuration (Revision 3)

The JTAG ID register is a 32-bit read-only factory-programmed register that distinguishes the component on a board according to the IEEE 1149.1 standard. **Figure 3** shows the JTAG ID register configuration for mask set 2K30A. Version information corresponds to the revision number ($0 for revision 0, $1 for revision A, etc.).

31 28	27 22	21 12	11 1	0
Version Information	**Customer Part Number**	**Sequence Number**	**Manufacturer Identity**	**1**
0011	000110	0000000001	0000 0000 1110	1

Figure 3. JTAG Identification Register Configuration (Revision 3)

The Class/Code/Revision ID Configuration Register (CCCR/CRID) is a 32-bit read-only factory-programmed register that is mapped into the PCI configuration space. The lower 8 bits define the revision ID and have changed as follows:

31 8	7 0
Class Code Register	**Configuration Register**
0000 0100 1000 0000 0000 0000	0000 0011

Figure 4. Class/Code/Revision ID Configuration Register (CCR/CRID)

9 Errata Removal

The 2K30A mask set removes all known functional errata on the 3F48S mask set. This section shows the functional errata that the 2K30A mask set removes. For further documentation updates, consult the errata for each device. The specific differences in errata items are listed in the following tables. A detailed description of each errata item is available at the Freescale website listed on the back page of this document. The silicon errata present on the 3F48S mask but not on the 2K30A masks are listed as follows:

Errata No.	Description	Status
ES16	**Description:** When the chip is powered up with PLL enabled (PINIT = 1), the skew between EXTAL and CLKOUT after the PLL locks cannot be guaranteed at high frequency (over 50 MHz, not 100% tested). Workaround: If skew between EXTAL and CLKOUT is needed, power up with PINIT = 0, and then enable the PLL by software.	Fixed on 2K30A
ES30	**Description:** After the $\overline{BB}$ pin output is driven high and released, the pin output voltage level may not reach V_{CC}. The issue depends on the application board layout and the parameters of the chip process. Workaround: Use a restricted board layout that includes a 1 kΩ pull-up resistor connected to the $\overline{BB}$ pin with a 100 Ω resistor connected in series with, and as close as possible to, the pin. The board route from the $\overline{BB}$ pin to any component should guarantee the following parameters: • Route inductance < 40 nH • Route capacitance < 15 pF • Input capacitance < 8 pF Such restrictions guarantee that when $\overline{BB}$ is driven high (deasserted), the output voltage level will be above 2.25 V at V_{CC} = 3.3 V.	Fixed on 2K30A
ES37	**Description**: In PCI mode, improper HI32 operation may result if the HTXR/HRXS registers are accessed by the PCI master at byte address Base_Address + (N × 2048 + 16), where N is an integer from 1–31. **Workaround**: Not available.	Fixed on 2K30A
ES41	**Description:** The HCLK pin of the HI32 presents an input capacitive load of almost 30 pF, which exceeds the permissible maximum load of 12 pF as specified in the PCI Specification Version 2.1. This may cause improper HI32 operation in PCI systems. The effect of this extra load may vary from system to system, depending on PCI clock driver strength. **Workaround:** Use a zero-propagation-delay external PLL device (e.g., CY2305) to buffer the PCI clock signal. This solution does not enable spread-spectrum PCI clocking.	Fixed on 2K30A
ES42	**Description:** When a Direct Memory Access (DMA) channel is in Line mode (that is, the DMA Transfer Mode is DTM = 010) with address modes defined by DMA Three Dimensional mode D3D = 0 and DMA = 10010x (that is, the DMA Counter (DCO) is in mode A), and the DCO value is greater than $FFF, then the DMA does not function properly. This address mode implies “no update” at the destination and “no update” or “post increment by 1” mode at the source. **Workaround:** Use Block Transfer mode (that is, DTM = 000). For the DCO and DMA Address Mode (DAM) settings described in this erratum, the Line Transfer mode of DMA is identical to its Block Transfer mode, so this combination is redundant. In fact, a block containing only one line is still a block.	Fixed on 2K30A

Errata No.	Description	Status
ES44	**Description:** Let’s say that “channel A” is the DMA channel servicing the HI32, and that “channel B” is another DMA channel that has been disabled by software. Then, depending on the DMA Request Source field (DRS[4:0]) of the two channels, channel A may be stalled by channel B being disabled. Channel A may be stalled when the DMA Channel Enable (DE) bit in the DMA Control Register is cleared by software in the following cases: • DE bit of channel B cleared by software because of —a Transfer Done from DMA channel 0 (DRSb = 00100) or —an ESSI1 Receive Data (DRSb = 01100) or then channel A may be stalled by a Host Slave Receive Data (DRSa = 11100). • DE bit of channel B cleared by software because of —a Transfer Done from DMA channel 1 (DRSb = 00101) or —an ESSI1 Transmit Data (DRSb = 01101) or then channel A may be stalled by a Host Master Receive Data (DRSa = 11101). • DE bit of channel B cleared by software because of —a Transfer Done from DMA channel 2 (DRSb = 00110) or —an SCI Receive Data (DRSb = 01110) or then channel A may be stalled by a Host Slave Transmit Data (DRSa = 11110). • DE bit of channel B cleared by software because of – a Transfer Done from DMA channel 3 (DRSb = 00111) or – an SCI Transmit Data (DRSb = 01111) or then channel A may be stalled by a Host Master Transmit Data (DRSa = 11111). **Workaround:** Use either one of the following alternatives: • Clear and set the DE bit of channel A immediately after you clear the DE bit of channel B. • Avoid a software clear of the DE bit of channel B.	Fixed on 2K30A
ES45	**Description:** When the Host Command Vector Register (HCVR) is written in Peripheral Component Interconnect (PCI) mode while the Receive Buffer Lock Enable (RBLE) bit is set in the DSP PCI Control Register (DPCR), the Host Data Transfer Complete (HDTC) status bit in DSP PCI Status Register (DPSR) may be set falsely, thus also causing an HDTC interrupt if that interrupt has been enabled by the Transfer Complete Interrupt Enable (TCIE) bit in the DPCR. **Workaround:** Use either one of the following alternatives: • Clear HDTC, if it is set, by writing it with 1 in the Host Command Interface Status Register (ISR). • Clear HDTC, if it is set, by writing it with 1; use software-dependent information to distinguish between a false and true HDTC setting. For example, you do either of the following: —Alter the destination address pointer if the DSP Receive Data Register (DRXR) data is being transferred by the DSP core. The pointer is changed if the HDTC setting is true. —Alter the destination address or counter registers of the DMA channel if the DRXR data is being transferred by the DMA. The registers are changed if the HDTC setting is true.	Fixed on 2K30A

ES47	**Description:**
	If the DMA channel and the core access the same 1/4 K internal X data, Y data, or program memory page, and the DMA interrupt is enabled, a false interrupt may occur in addition to the correct one.
	
	Workaround:
	Ensure that the channel’s DTD status bit in the DSTR is set before jumping to the Interrupt Service Routine (that is, the interrupt is correct only when DTD is set).
	
	Example:
	`ORG P:I_DMA0`
	`JSSET #M_DTD0,X:M_DSTR,ISR_      ; ISR_ is the Interrupt Service`
	`                                 ; Routine label for DMA channel 0`
		Fixed on 2K30A

Errata No.	Description	Status
ES53	**Description:** Using the JTAG instruction code 1111 ($F) or 1101 ($D) for the BYPASS instruction may cause the chip to enter Debug mode (which then correctly sets the Status bits (OS[1:0]) in the OnCE Status and Control Register (OSCR[7:6]) and asserts the $\overline{\text{DE}}$ output to acknowledge the Debug mode status). **Workaround:** Use one of the following alternatives: • If possible, do not use instruction code 1111 ($F) or 1101 ($D) for the BYPASS instruction. Use one of the other defined BYPASS instruction codes (that is, any code from 1000–1100 ($8–$C) or 1110 ($E)). • If you must use instruction code 1111 ($F) or 1101 ($D), use the following procedure: —While the $F or $D instruction code is in the Instruction Register, ensure that the JTAG Test Access Port (TAP) state machine does not pass through the JTAG Test-Logic-Reset state while accessing any JTAG registers (that is, Instruction Register, Boundary Scan Register, or ID Register). —Before using any other JTAG instruction, load one of the other BYPASS instruction codes (that is, any code from 1000–1100 ($8–$C) or 1110 ($E)) into the instruction register. Then, any other JTAG instruction may be used.	Fixed on 2K30A
ES54	**Description** (added 1/27/98): When a DMA channel is configured using its DMA Control Register (DCR) in the following way: • •Line Transfer mode is selected (DTM[2:0] = 010) •Non-Three-Dimensional Address mode is selected (D3D = 0) •Destination Address Offset Register DOR1 or DOR3 is selected (DAM[5:3] = 001 or 011) •No Source Address Offset is selected (DAM[2:0] = 100 or 101) The DMA transfer does not function as intended. **Workaround:** Select Destination Address Offset Register DOR0 or DOR2 by setting DAM[5:3] = 000 or 010.	Fixed on 2K30A
ES72	**Description** (added 10/30/1997): During external memory accesses, noise may be generated by the Bus Strobe ($\overline{\text{BS}}$) signal and placed on the Test Clock (TCK) signal, causing the JTAG Test Access Port (TAP) controller to change states unpredictably. This problem may be more severe at higher speeds or in applications using multiple DSP56300 families. **Note**: Applies to PBGA package only. **Workaround:** Bypass the $\overline{\text{BS}}$ line to signal ground with a 10 pF capacitor.	Fixed on 2K30A
ES81	**Description:** The HI32 may generate a wrong PAR signal. **Workaround:** If possible, the system should ignore parity errors generated in such a case.	Fixed on 2K30A

ES82	**Description:**
	The $\overline{\text{BL}}$ pin may operate improperly when two consecutive manipulation instructions (bset/bclr/bchg) use external memory as the destination.
	Example of the sequence:
	`bset #5,x:(r0) ;; r0 is a pointer on an external memory address`
	`bclr #7,x:(r3) ;; r3 is a pointer on an external memory address`
	Workaround :
	Separate the consecutive bit manipulation instructions by any other instruction, as in the following example:
	`bclr #7,x:(r3) ;; r3 is a pointer on an external memory address`
	`nop`
	`bset #5,x:(r0) ;; r0 is a pointer on an external memory address`
		Fixed on 2K30A

Errata No.	Description	Status

ES84	**Description:**
	When software disables a DMA channel (by clearing the DE bit of the DCR) , the DTD status bit of the channel may not be set if any of the following events occur:
	• Software disables the DMA channel just before a conditional transfer stall (Described by App B-3.5.1,UM).
	• Software disables the DMA channel at the end of the block transfer (that is, after the counter is loaded with its initial value and transfer of the last word of the block is completed).
	As a result, the Transfer Done interrupt might not be generated.
	
	Workaround:
	Avoid using the instruction sequence causing the conditional transfer stall (See DSP56300 UM, App B-3.5.1 for description) in fast interrupt service routines. Every time the DMA channel needs to be disabled by software, the following sequence must be used :
	`                bclr    #DIE,x:M_DCR    ; not needed if DIE is cleared`
	`                bclr    #DE,x:M_DCR`
	`; instead of two instructions above, one 'movep' instruction may be used`
	`; to clear DIE and DE bits`
	`                movep   #DCR_Dummy_Value,x:M_DCR`
	`                bclr    #DE,x:M_DCR`
	`                nop`
	`                nop`
	Here, the `DCR_Dummy_value` is any value of the DCR register that complies with the following requirements:
	• DE is set;
	• DIE is set if Transfer Done interrupt request should be generated and cleared otherwise;
	• DRS[4:0] bits must encode a reserved DMA request source (see the following list of reserved DRS values);
	
	List of reserved DRS[4:0] values (per device):
	• DSP56302, DSP56309, DSP56303, DSP56304, DSP56362 — `10101-11111`
	• DSP56305 — `11011`
	• DSP56301 — `10011-11011`
	• DSP56307 — `10111-11111`
		Fixed on 2K30A

ES86	**Description:** If the HI32 performs a write transaction as a PCI master and the transaction is disconnected by the target, the value of the MTRQ status bit in the DPSR register may be wrong. **Workaround:** Do not use an MTRQ status bit-related interrupt or polling. (The related DMA functionality is not affected by this issue.)	Fixed on 2K30A
ES87	**Description:** When the HI32 is an active PCI target, it does not set the DPE bit in the CSTR register if an address parity error occurs. **Workaround :** The Host can get information about the Address Parity status either by reading the SSE bit (in the CSTR) or by indirectly reading the (e.g. via Host Command) the APER bit in the DPSR register.	Fixed on 2K30A
ES89	**Description:** If the SCI Receiver is programmed to work with a different serial clock than the SCI Transmitter so that either the Receiver or Transmitter is using the external serial clock and the other is using the internally-generated serial clock—RCM and TCM in the SCCR are programmed differently)—then the internal serial clock generator will not operate and the SCI portion (Receiver or Transmitter) clocked by the internal clock will be stuck. **Workaround:** Do not use SCI with the two SCI portions (Receiver and Transmitter) clocked by different serial clocks; use either both externally or both internally clocked. Or: When using both portions of the SCI (Receiver & Transmitter), do not program different values on RCM and TCM in the SCCR.	Fixed on 2K30A

Errata No.	Description	Status
ES90	**Description**: A deadlock occurs during DMA transfers if all the following conditions exist: • 1. DMA transfers data between internal memory and external memory through port A. • 2. DMA and the core access the same internal 0.25K memory module. • 3. The bus arbitration system is implemented in such way that the bus is granted to the DSP ($\overline{\text{BG}}$ is asserted) only after the DSP requests it ($\overline{\text{BR}}$ is asserted). The symptom is a deadlock on DMA activity, that is, a DMA transfer stops for no apparent reason. $\overline{\text{BR}}$ is not asserted when it should be asserted because of DMA. **Workaround:** One of the following: • 1. Use intermediate internal memory on which there is no contention with the core. • 2. Do not use bus arbitration (tie $\overline{\text{BG}}$ to the ground), or have an external arbiter that asserts $\overline{\text{BG}}$ even if $\overline{\text{BR}}$ is not asserted by the DSP. • 3. Set the DSP core priority higher than the DMA (for Port A accesses), and do a periodic external access by the core. The core access solves the deadlock.	Fixed on 2K30A
ES95	**Description:** If more than a single DMA channel is enabled while the DSP stays in the WAIT processing state, and triggering one of the DMA channels causes an exit from the WAIT state (See A-6.115, UM), triggering another DMA channel might cause improper DMA operation. **Workaround:** Assure that only a single DMA channel can be triggered during DSP WAIT state. If the application cannot guarantee this, other DMA channels should be disabled before the WAIT processing state is entered and then reenabled after WAIT state is exited.	Fixed on 2K30A

ES101	**Description**:
	If the reset mode is expanded mode (for example, mode 0 or mode 8 on the DSP5630x), A MOVE (not a PROGRAM FETCH) from internal P memory to any destination may not work properly.
	
	Workaround:
	After each reset ($\overline{\text{RESET}}$) negation and before the first move from internal program memory, execute the following sequence:
	`BSET   #M_CE,sr`
	`NOP`
	`NOP`
	`NOP`
	`BCLR   #M_CE,sr`
		Fixed on 2K30A

ES104	**Description:** An improper operation may occur when all the following conditions apply: • The DMA channel is in a mode that does not automatically clear the DE bit at the end of the block (DTM[2:0] = 1xx in DCR). • This channel is disabled by software (by clearing DE in DCR) while it is triggered for a new transfer. • The previous operation is not yet completed. **Workaround:** The DMA channel should be disabled only when it is not triggered for a new transfer, that is, when the DACT bit in the DSTR register is cleared. To perform this operation most efficiently, all other DMA channels should be disabled.	Fixed on 2K30A
ES107	**Description:** The HDTC status bit (relevant only if the RBLE control bit is set) may not be set properly when both of the following conditions apply: • a) DSP software clears the HDTC bit while the PCI bus is parked on the HI32. • b) The PCI master read transaction is initiated by the HI32 while the bus is still parked on the HI32. **Workaround:** Use one of the following alternatives: • Avoid bus parking on the HI32. • Enter the Personal Software Reset (HM[2:0]=0) in HDTC ISR. • Poll the MRRQ and SRRQ status bits before the start of each master read transaction (e.g. in MARQ ISR). Start this transaction only when both MRRQ and SRRQ are cleared. The HDTC status bit should be cleared by the DSP software as defined in the specification.	Fixed on 2K30A

Errata No.	Description	Status
ES114	**Description** (added 4/19/99): A DMA channel may operate improperly when the address mode of this channel is defined as three-dimensional (D3D=1) and DAM[5:0] = 1xx 1 10 or DAM[5:0] = 01x1 10 (that is, triple counter mode is E). **Workaround:** Use the triple counter modes C(DAM[1:0]=00) or D(DAM[1:0]=01) instead of the E(DAM[1:0]=10) mode.	Fixed on 2K30A

ES115	**Description** (added 4/19/99):
	When a DMA channel (called channel A) is disabled by software clearing the channel's DCR[DE] bit, the DTD bit may not get set, and the DMA end of the block interrupt may not happen if one of the following occurs:
	• 1. There is another channel (channel B) executing EXTERNAL accesses, and the DE bit of channel A is being cleared by software at the end of the channel B word transfer - if channel B is in Word transfer mode, or at the end of the channel B line transfer - if channel B is in Line Transfer mode, or at the end of the channel B block transfer - if channel B is in Block transfer mode.
	• 2. This channel (A) is executing EXTERNAL accesses, and the DE bit of this channel (A) is being cleared by software at the end of the channel B word transfer - if channel B is in Word transfer mode, or at the end of the channel B line transfer - if channel B is in Line transfer mode.
	
	Workaround:
	Avoid executing a DMA external access when any DMA channel should be disabled. This can be done as follows. Every time the DMA channel needs to be disabled by software, the following sequence must be used:
	`;; initialize an unused DMA channel "C"`
	`movep #DSR_swflag, x:M_DSRC ;; here DSR_swflag is an`
	`;; unused X, Y or P memory`
	`;; location, should`
	`;; be initialized to`
	`;; $800000`
	`;; M_DSRC - address of the`
	`;; channel C DSR register.`
	`movep #DDR_swflag, x:M_DDRC ;; DDR_swflag is an unused`
	`;; X, Y or P memory`
	`;; location, should be`
	`;; initialized to $000000`
	`;; M_DDRC -`
	`;; address of the channel C`
	`;; DDR register .`
	`movep #TR_LENGTH, x:M_DCOC ;; see below the definition`
	`;; of the TR_LENGTH value,`
	`;; M_DCOC - address`
	`;; of the channel C DCO register`
	`movep #1f0240, x:M_DCRC ;; M_DCRB - address of the`
	`;; channel C DCR register.`
	`;; Set transfer mode -`
	`;; block transfer,`
	`;; triggered by`
	`;; software highest`
	`;; priority, continuous`
	`;; mode on no-update`
	`;; source and destination`
	`;; address mode X memory`
	`;; location for source`
	`;; and destination (can be`
	`;; chosen by`
	`;; user accordingly to`
	`;; DSR_swflag/DDR_swflag)`
		Fixed on 2K30A

Errata No.	Description	Status

ES115 cont.	`;; disable DMA channel "A"`
	`ori #3, mr ;; mask all interrupts`
	`bset #23, x:M_DCRC ;; enable DMA channel C`
	`bclr #23,x:DDR_swflag,* ;; wait until DMA channel C`
	`;; begin transfer`
	`bclr #23, x:M_DCRA ;; disable DMA channel A`
	`nop`
	`nop`
	`jclr #M_DTDA, x:M_DSTR,* ;; polling DTD bit of the DMA`
	`;; channel A,`
	
	The TR_LENGTH value can be defined as the maximum length of the external DMA transfer——from the length of the read DMA cycle and from the length of the write DMA cycle. The length of the external read/write DMA cycle can be defined as the length of the PORTA external access. The length of the internal read/write DMA cycle can be defined in the errata case as 2 DSP clock cycles. The TR_LENGTH can be found as sum of the lengths of the DMA read and DMA write cycles.
		Fixed on 2K30A

ES122	**Description** (added 10/21/99): The JTAG boundary scan functionality is guaranteed to operate only up to 1MHz at room temperature (25C) and at relaxed input levels of 0.5V maximum input voltage (Vil) for TCK and 3.0V minimum input voltage (Vih) for TRST. For reference, see the JTAG/OnCE Interface table in chapter 1 of the DSP56301 technical data sheet; see the DC Electrical Characteristics table and JTAG Timing tables and figures in chapter 2 of the DSP56301 technical data sheet. Workaround: None	Fixed on 2K30A
ES124	**Description** (added 9/11/99) (reclassified from documentation to silicon errata 11/11/99): When an external PCI master executes a configuration space read from the HI32 with an odd number of byte lanes enabled (for example, $\overline{BE3}$ – $\overline{BE0}$ = 1000), the DSP drives the parity signal (HPAR) with the wrong value. This is because the $\overline{BE3}$ – $\overline{BE0}$ signals are ignored (erroneously) when generating the parity value during configuration space reads. Workaround: None.. Pertains to the HI32 (PCI) chapter of the user’s manual, in the section on PCI Mode (DCTR[HM]=$1). In Revision 2 of the *DSP56301 User’s Manual*, this section is 6.5.2 on page 6-14. The information should accompany the bullet on Memory-Space and configuration transactions as a target. NOTE: Was documentation errata, ED39.	Fixed on 2K30A

10 Documentation Errata

Following are errata items that have been designated as documentation errata that are specification changes for the entire DSP56300 family, including the DSP56301. They have been documented as specification changes in the *DSP56301 User’s Manual* and *Technical Data* sheet.

Errata No.	Description	Status

ED1	**Description** (revised 11/9/98):
	XY memory data move does not work properly if 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 OR the Y-memory move destination is a register used as source in the next adjacent move to non Y-memory.
	
	Here are examples of the two cases (where x:(r1) is a peripheral):
	Example 1:
	`move #$12,y0`
	`move x0,x:(r7) y0,y:(r3) (while x:(r7) is a peripheral).`
	Example 2:
	`mac     x1,y0,a x1,x:(r1)+      y:(r6)+,y0`
	`move    y0,y1`
	
	This is not a bug, but a documentation update. Any of the following alternatives can be used:
	• Separate these two consecutive moves by any other instruction.
	• Split XY Data Move to two moves.
		DSP563xx
		Documentation Update

ED2	**Description** (added 10/09/1997): $\overline{\text{BL}}$ 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). This is not a bug, but a documentation update.	DSP563xx Documentation Update
ED3	**Description** (added 10/09/1997): A one-word conditional branch instruction at LA-1 is not allowed. This is not a bug, but a documentation update.	DSP563xx Documentation Update

ED4	**Description** (added 10/09/1997):
	The following instructions should not start at address LA:
	MOVE to/from Program space {MOVEM, MOVEP (only the P space options)}
	This is not a bug but a documentation update (Appendix B, DSP56300 Family Manual).
		DSP563xx
		Documentation Update

ED6	**Description** (added 4/13/98): When the $\overline{\text{HIRQ}}$ pin is used in pulse mode (HIRH=0 in DCTR), the LT[7:0] value (in CLAT) should not be zero. This is not a bug but a documentation update.	DSP563xx Documentation Update
ED7	**Description** (added 1/27/98): When activity passes 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). **Workaround**: None. Pertains to: *DSP56300 Family Manual*, **Sections 8.1.6.3** and **8.1.6.4.**	DSP563xx Documentation Update
ED8	**Description** (added 10/09/1997): The timing for HSAK is no longer qualified by the data strobe. The new timing numbers are: • **T318**—HSAK assertion from HA0–HA10 and HAEN valid is 30.0 ns maximum. • **T319**—HSAK assertion hold from HA0-HA10 and NAEN not valid is 2.0 ns minimum. This is not a bug, but a documentation update of a specification change.	DSP563xx Documentation Update
ED10	**Description** (added 5/13/98): The HI32 may operate improperly in PCI mode when the TWSD bit is set in the HCTR register. **Workaround**: Do not set the TWSD bit in the HCTR register; this bit is reserved. This is a documentation change.	DSP563xx Documentation Update
ED12	**Description** (added 5/13/98): When the HI32 is in PCI mode, the HTF control bits affect the address insertion (the IAE bit is set in the DPCR register) in the same way they affect the transferred data. Address as appears on the PCI bus: $12345678 **HTF[1:0]** **Inserted Address** 00 $005678, $001234 01 $345678 10 $345678 11 $123456 **Workaround**: This is a documentation update.	DSP563xx Documentation Update

Errata No.	Description	Status
ED13	**Description** (added 5/15/98): When the HI32 is in PCI mode, the Insert Address Enable control bit (IAE=1) can be set only with the Receive Buffer Lock Enable control bit set (RBLE=1 in the DPCR register.)	DSP563xx Documentation Update
ED15	**Description** (added 7/21/98): 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. **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.	DSP563xx Documentation Update
ED17	**Description** (added 9/28/98): In all DSP563xx technical datasheets, 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. **Workaround**: N/A	DSP563xx Documentation Update
ED18	**Description** (added 11/2/98): The PCI host must not change the values of the HBE[3:0] bits during PCI read transactions from the HI32 as a PCI target.	DSP563xx Documentation Update
ED19	**Description** (added 11/9/98): To guarantee the proper HI32 operation, the DMA should service the HI32 under the following restrictions: • Two DMA channels should not service the DRXR FIFO if master and slave data is mixed there. • The DMA data transfers should not be concurrent with the 56300 Core data transfers to/from the same HI32 data FIFO.	DSP563xx Documentation Update
ED20	**Description** (added 11/24/98): In the Technical Datasheet Voh-TTL should be listed at 2.4 Volts, not as: `TTL = Vcc-0.4` **Workaround**: This is a documentation update.	DSP563xx Documentation Update
ED21	**Description** (added 11/24/98): In the Technical Datasheet Iol should be listed as 1.6 mA, not as 3.0 mA. **Workaround**: This is a documentation update.	DSP563xx Documentation Update
ED24	**Description** (added 11/24/98): 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. **Workaround**: This is a documentation update.	DSP563xx Documentation Update

Errata No.	Description	Status
ED25	**Description** (added 12/16/98): *Current definition*: HDTC is set if SRRQ and MRRQ are cleared (that is, the host-to-DSP data path is emptied by DSP56300 core reads) under one of the following conditions: • a non-exclusive PCI write transaction to the HTXR terminates or completes • $\overline{\text{HLOCK}}$ is negated after the completion of an exclusive write access to the HTXR • the HI32 initiates a read transaction. The HI32 disconnects (retry or disconnect-C) forthcoming write accesses to the HTXR as long as HDTC is set. *New definition*: HDTC is set if SRRQ and MRRQ are cleared (that is, the host-to-DSP data path is emptied by DSP56300 Core reads) under one of the following conditions: • a non-exclusive PCI write transaction to the HTXR terminates or completes • $\overline{\text{HLOCK}}$ is negated after the completion of an exclusive write access to the HTXR. The HI32 disconnects (retry or disconnect-C) forthcoming write accesses to the HTXR as long as HDTC is set. **Note**: The HDTC bit is not set after a read transaction initiated by the HI32 as a PCI master. **Workaround**: NTR	DSP563xx Documentation Update
ED26	**Description** (added 1/6/99): The specification DMA Chapter is wrong. The sentence, “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.” should be replaced with: “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.”	DSP563xx Documentation Update
ED27	**Description** (added 1/12/99): The PBGA mechanical package drawing in the 56301 and 56305 data sheets is incorrect. The figure numbers of the incorrect drawings are Figure 3-6 for the 56301 and Figure 3-3 for the 56305. The only incorrect part is the bottom view above the label "VIEW M-M." This view erroneously shows the number of pins on the package to be 256, but the actual number of pins is 252. In the drawing, the four balls in the corners should not appear.	DSP563xx Documentation Update

ED28	**Description** (added 1/7/1997; identified as Documentation Errata 2/1/99):
	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:
	
	`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`
	
	Workaround:
	Put an additional NOP between LABEL2 and LABEL1. Pertains to *DSP56300 Family Manual*, **Appendix B**, **Section B-4.1.3**.
		DSP563xx
		Documentation Update

Errata No.	Description	Status
ED29	**Description** (added 9/12/1997; identified as a Documentation errata 2/1/99): 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. **Workaround**: None at this time. Pertains to the *User's Manual*, **Section 7.4.1.7**, **Table 7-2**.	DSP563xx Documentation Update
ED30	**Description** (added 9/12/1997; identified as a Documentation errata 2/1/99): When the ESSI transmits data in the On-Demand mode (that is, MOD = 1 in CRB and DC[4:0] = $00000 in CRA) with WL[2:0] = 100, the transmission does not work properly. **Workaround**: To ensure correct operation, do not use the On-Demand mode with the WL[2:0] = 100 32-bit Word-Length mode. Pertains to the *User's Manual*, **Section 7.5.4.1**.	DSP563xx Documentation Update
ED31/ ES40	**Description**: Programming the ESSI to use an internal frame sync (that is, 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 GPI is selected. This errata and workaround apply to both ESSI0 and ESSI1. **Workaround**: To assure correct operation, either program the GPIO pins as outputs or configure the pins in the PCR as ESSI signals. **Note**: The default selection for these signals after reset is GPI.	DSP563xx Documentation Update

ED32/ ES103	**Description** (added 11/9/98):
	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.
	Workaround:
	Replace the RTI with the following sequence:
	`movec   ssl,sr`
	`nop`
	`rti`
		DSP563xx
		Documentation Update

<table>
<tr><th>Errata No.</th><th>Description</th><th>Status</th></tr>
<tr><td>ED33/
ES109</td><td>

Description (added 12/16/98):
When Stack Extension mode is enabled, a use of the instructions BRKcc or ENDDO inside do loops might cause an improper operation.
If the loop is non nested and has no nested loop inside it, the erratais relevant only if LA or LC values are being used outside the loop.

Workaround:
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.

1) Finite DO loops (that is,not DO FOREVER loops)

```
==============================================
BRKcc
```

Original code:

```
        do #N,label1
        .....
        .....
                do #M,label2
                .....
                .....
                BRKcc
                .....
                .....
label2
        .....
        .....
label1
```

Will be replaced by:

```
        do #N, label1
        .....
        .....
                do #M, label2
                .....
                .....
                Jcc     fix_brk_routine
                .....
                .....
nop_before_label2
                nop     ; This instruction must be NOP.
label2
        .....
        .....
```

</td><td>DSP563xx
Documentation Update</td></tr>
</table>

<table>
<tr><th>Errata No.</th><th>Description</th><th>Status</th></tr>
<tr><td>

ED33/ ES109 cont.

</td><td>

```
        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
        nop_after_jmp
                        NOP  ; This instruction must be NOP.
                        .....
                        .....
        label2
                .....
                .....
        label1
        ....
        ....
        fix_enddo_routine
                move #1,lc
                move #nop_after_jmp,la
                jmp  nop_after_jmp
```

2) DO FOREVER loops

```
        ===================
        BRKcc
        -----
```

</td><td>DSP563xx Documentation Update</td></tr>
</table>

Errata No.	Description	Status

ED33/ ES109 cont.	Original code:
	`        do #M,label1`
	`        .....`
	`        .....`
	`                do forever,label2`
	`                .....`
	`                .....`
	`                BRKcc`
	`                .....`
	`                .....`
	`label2`
	`        .....`
	`        .....`
	`label1`
	Will be replaced by:
	`        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`
	`        .....`
	`        .....`
		DSP563xx Documentaiton Update

<table>
<tr><th>Errata No.</th><th>Description</th><th>Status</th></tr>
<tr><td><b>ED33/ ES109 cont.</b></td><td>

```
                        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_jmp
                NOP  ; This instruction should be NOP
                .....
                .....
        label2
                .....
                .....
        label1
....
        ....
        fix_enddo_routine
                        nop
                        move #1,lc
                        bclr #16,ssl
                        move #nop_after_jmp,la
                        rti            ; <--- note: "rti" and not "rts"
```

</td><td>DSP563xx
Documentation Update</td></tr>
<tr><td><b>ED34/ ES110</b></td><td>

Description (added 1/5/99):
When stack extansion 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:

Case 1:

- For the first executed instruction: move from SSH or bit manipulation on SSH (that is, jclr, brclr, jset, brset, btst, bsset, jsset, bsclr, jsclr).
- For the second executed instruction: move to SSH or bit manipulation on SSH (that is, jsr, bsr, jscc, bscc).
- 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 (that is, bset, bclr, bchg, jclr, brclr, jset, brset, btst, bsset, jsset, bsclr, jsclr).

Workaround:
Add two NOP instructions before the third executed instruction.

Case 2:
For the first executed instruction: bit manipulation on SSH (that is, bset, bclr, bchg).
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 (that is, bset, bclr, bchg, jclr, brclr, jset, brset, btst, bsset, jsset, bsclr, jsclr).

Workaround:
Add two NOP instructions before the second executed instruction.

</td><td>DSP563xx
Documentation Update</td></tr>
</table>

Errata No.	Description	Status
ED37	**Description** (added 4/19/99): In paragraph 6.1.1.11 on page 6-12 of the 301 User’s Manual, there is an error, as follows: "HIRQ_ is asserted by the HI32 when a host interrupt request (recieve and/or transmit) is generated in the HI32" **Workaround/correction**: Should be: "HIRQ_ is asserted by the HI32 when a host interrupt request (receive and/or transmit) is generated in the HI32 (as described in paragraphs 6.2.1.1, 6.2.1.1 and 6.2.1.4)."	DSP563xx Documentation Update

11 Errata Deletion

The following documentation change should be omitted from the DSP56301 because it pertains to HI08, which does not exist on the DSP56301. This item was incorrectly documented as a DSP56301 errata.

Errata No.	Description	Status
ED14	The data sheets of the various DSP56300 host interfaces (HI32 excluded) must be modified to make the HI08/HDI08 compatible with PortA timing 114, which is included here as a reference. Timing 321 "Write data strobe deassertion width" should be split (similar to timing 319 "Read data strobe deassertion width"), as described here: Write data strobe deassertion width: after HCTR, HCVR and "Last Data Register" writes 2.5*Tc+10.0 @66MHz 2.5*Tc+8.3 @80MHz 2.5*Tc+6.6 @100MHz after TXH:TXM writes (with HBE=0), TXM:TXL writes (with HBE=1) 25 @66MHz 20.6 @80MHz 16.5 @100MHz That is, a minimum of 4 WS for PortA is required for 100 MHz operation. Reference: Timing 114 @ 100MHz WR_ deassertion time 0.5 x TC - 3.5 1.5ns [WS = 1] TC - 3.5 6.5ns [2 <= WS <= 3] 2.5 x TC - 3.5 21.5ns [4 <= WS <= 7] 3.5 x TC - 3.5 31.5ns [WS >= 8]	DELETION: Does Not Apply to Any Mask of the DSP56301

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
support.asia@freescale.com

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.