MOTOROLA

Chip Errata
DSP56364 Digital Signal Processor
Mask: 4J25D

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 Motorola DSP Tools CLAS package.

Silicon Errata

Errata Number	Errata Description	Applies to Mask
ES42	Description (added 5/28/98): When a Direct Memory Access (DMA) channel is in Line mode (i.e., the DMA Transfer Mode is DTM = 010) with address modes defined by DMA Three Dimensional mode D3D = 0 and DMA = 10010x (i.e., 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 (i.e., 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.	4J25D

Errata Number	Errata Description	Applies to Mask
ES53	Description (added 10/13/1997): 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: a. 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 (i.e., any code from 1000–1100 ($8–$C) or 1110 ($E)). b. If you must use instruction code 1111 ($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 (i.e., Instruction Register, Boundary Scan Register, or ID Register). — Before using any other JTAG instruction, load one of the other BYPASS instruction codes (i.e., any code from 1000–1100 ($8–$C) or 1110 ($E)) into the instruction register. Then, any other JTAG instruction may be used.	4J25D

Errata Number	Errata Description	Applies to Mask
ES54	Description (added 1/27/98): When a DMA channel is configured using its DMA Control Register (DCR) in the following manner: • 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.	4J25D

Errata Number	Errata Description	Applies to Mask
ES84	Description (added 5/13/98):	4J25D

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:

a. Software disables the DMA channel just before a conditional transfer stall (Described by App B-3.5.1,UM).

b. 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

Errata Number	Errata Description	Applies to Mask
ES92	Description (added 7/21/98): Due to a circuit race in the DMA block, enabling any DMA channel by software (by setting DE) for transferring a block of data (TM=011) might not work properly. Workaround: This mode of a channel trigger by software can be exactly emulated by enabling the DMA channel for block transfer triggered by a peripheral request (TM=000).	4J25D
ES95	Description (added 8/15/98): 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.	4J25D

Errata Number	Errata Description	Applies to Mask

ES104	Description (added 11/20/98):
	An improper operation may occur when a DMA channel uses the following transfer modes:
	• DTM(2:0) = 100
	• DTM(2:0) = 101
	where the DE bit is not automatically cleared at the end of block and the DMA channel is disabled by software (DE bit is cleared) while it is triggered for a new transfer.
	Workaround:
	The DMA channel should be disabled only when it cannot be triggered by a new transfer. Use one of the following alternatives:
	1. The system configuration must guarantee that no DMA trigger can occur while the DE bit is cleared.
	2.The following sequence disables the DMA channel:
	`a/    Wait until the DTD bit is cleared`
	`b/    Clear the DE bit`
	`c/    Wait until the DTD bit is set`
		4J25D

ES114	Description (added 4/19/99, revised 4/30/99): 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] = 01xx 10 (i.e., 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.	4J25D

Errata Number	Errata Description	Applies to Mask
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.	4J25D

Errata Number	Errata Description	Applies to Mask
ES115 cont.	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)	4J25D

Errata Number	Errata Description	Applies to Mask
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.	4J25D
ES126	Description (added 11/17/99): When the core and the DMA are simultaneously accessing the same ROM memory module, the contention is not detected and the DMA transfers erroneous data. Workaround: The programmer must ensure that the core and DMA never access the same ROM memory module simultaneously. The 56364 has three ROM memory modules in the program space: 4K module - from $FF1000 to $FF1FFF 2K module - from $FF2000 to $FF27FF 2K module - from $FF2800 to $FF2FFF	4J25D

Errata Number	Errata Description	Applies to Mask
ES128	Description (added 4/12/2000) When the ESAI is operating in the asynchronous mode (SYN=0) and both SCKR and SCKT bit clock signals are defined as inputs from an external clock source (TCKD=0 in the TCCR register and RCKD=0 in the RCCR register), the internal clock dividers will be disabled. The result is that the HCKT and HCKR signals will not operate as ouputs even if so defined in the control register. If the ESAI is operating in the synchronous mode (SYN=1), then the HCKT signal will be disabled when SCKT is defined as an input. Workaround: To enable the operation of HCKT and HCKR signals as outputs, at least one of the SCKR or SCKT signals must be defined as output.	4J25D

Documentation Errata

	Errata Description	Applies to Mask

ED1	Description (revised 11/9/98):
	XY memory data move does not work properly under one of the following two situations:
	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.
	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`
	Any of the following alternatives can be used:
	a. Separate these two consecutive moves by any other instruction.
	b. Split XY Data Move to two moves.
	Pertains to: DSP56300 Family Manual, Section B-5 "Peripheral pipeline restrictions.
		4J25D

ED3	Description (added 5/7/1996): A one-word conditional branch instruction at LA-1 is not allowed. **Pertains to:** DSP56300 Family Manual, Appendix B, Section B.4.1.3	4J25D

	Errata Description	Applies to Mask

ED4	Description (added 10/13/1997):
	The following instructions should not start at address LA:
	MOVE to/from Program space {MOVEM, MOVEP (only the P space options)}
	This is a documentation update to the Appendix B, DSP56300 Family Manual.
		4J25D

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.	4J25D
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.	4J25D

	Errata Description	Applies to Mask

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, “At LA-1.”
		4J25D

	Errata Description	Applies to Mask

ED32	Description (added 11/9/98; identified as a Documentation errata 2/1/99):
	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`
	Pertains to: DSP56300 Family Manual. Add a new section to Appendix B that is entitled “Sixteen-Bit Compatibility Mode Restrictions.”
		4J25D

	Errata Description	Applies to Mask
ED33	Description (added 12/16/98; identified as a Documentation errata 2/1/99):	4J25D

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 (i.e. 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
             .....
             .....
```

	Errata Description	Applies to Mask
ED33 cont.		4J25D

```
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
```

	Errata Description	Applies to Mask

ED33 cont.	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
	----- 	
	Original code:
	
	       do #M,label1
	       .....
	       .....
	             do forever,label2
	             .....
	             .....
	             BRKcc
	             .....
	             .....
	label2
	       .....
	       .....
	label1
		4J25D

	Errata Description	Applies to Mask
ED33 cont.	Will be replaced by:	4J25D

```
       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
       .....
       .....
```

<table>
<tr><th></th><th>Errata Description</th><th>Applies to Mask</th></tr>
<tr><td>ED33 cont.</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"
```

Pertains to: DSP56300 Family Manual, Section B-4.2, “General Do Restrictions.”

</td><td>4J25D</td></tr>
</table>

	Errata Description	Applies to Mask
ED34	Description (added 1/5/99; identified as a Documentation errata 2/1/99): 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 (i.e. jclr, brclr, jset, brset, btst, bsset, jsset, bsclr, jsclr). For the second executed instruction: move to SSH or bit manipulation on SSH (i.e. 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 (i.e., 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 (i.e. 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 (i.e., bset, bclr, bchg, jclr, brclr, jset, brset, btst, bsset, jsset, bsclr, jsclr). Workaround: Add two NOP instructions before the second executed instruction. **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.	4J25D

	Errata Description	Applies to Mask
ED38	Description (added 7/14/99): 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. **Pertains to:** *DSP56300 Family Manual*, 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).	4J25D

<table>
<tr><th></th><th>Errata Description</th><th>Applies to Mask</th></tr>
<tr><td>ED40</td><td>Description (added 11/11/99):
When an instruction with all the following conditions follows a repeat instruction, then the last move will be corrupted.:
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.
Example:
    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
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.
Workaround:
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:
rep #number
tfr x0,a   x(r0)+,x0
move a,y0
If no interrupts before the move is a must, mask the interrupts before the REP. <b>Pertains to:</b> <i>DSP56300 Family Manual</i>, Rev. 2, Section A.3, "Instruction Sequence Restrictions."</td><td>4J25D</td></tr>
</table>

	Errata Description	Applies to Mask
ED41	Description (reclassified as a documentation errata item on 3/22/2000): If the stack extension is enabled, the instructions listed below should not be placed as the next-to-last or as the last instruction of a DO loop (i.e., should not appear at LA-1 or LA). The instructions are: XY Memory Data Move (A-6.76) X Memory Move (A-6.71) Y Memory Move (A-6.73) Long Memory Data Move (A-6.75) Immediate Short Data Move (A-6.68) Register to Register Data Move (A-6.69) Address Register Update (A-6.70) X Memory and Register Data Move (A-6.72) Y Memory and Register Data Move (A-6.74) *Arithmetic Instructions that allow Parallel Moves listed above* IFcc and IFcc.U (A-6.41) Workaround: Insert a NOP or other instruction not listed above as the next-to-last and last instructions in the DO loop. **Pertains to:** *DSP56300 Family Manual*, Rev. 2, Section 5.4.3, “System Stack Configuration and Operation Registers.” To be noted immediately after the paragraph on nested hardware DO loops.	4J25D
ED42	Description (added on 3/22/2000) The DMA End-of-Block-Transfer interrupt cannot be used if DMA is operating in the mode in which DE is not cleared at the end of the block transfer (DTM = 100 or 101). **Pertains to:** *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).	4J25D

Motorola and Ⓜ are registered trademarks of Motorola, Inc.
OnCE is a trademark of Motorola, Inc.

NOTES

1. An over-bar (i.e., $\overline{\text{xxxx}}$) indicates an active-low signal.
2. The letters in the right column tell which DSP56364 mask numbers apply.
3. The Motorola DSP website has additional documentation updates that can be accessed at the following URL:

   http://www.motorola-dsp.com/

-end-