Functional Differences Between Masks 4J22A and 0K36A of the DSP56303

This document describes the differences between masks of the DSP56303: the new mask, 0K36A, and the mask immediately preceding it, 4J22A. The new 0K36A mask of the DSP56303 uses the communications design rules (CDR2) process. The 0K36A mask set also has new I/O and a new PLL, with the requisite change in the PLL capacitor's equation (PCAP).

CONTENTS

1 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.

2 ESD Protection

The 0K36A mask set incorporates improved I/O structures to ensure Electro Static Discharge (ESD) protection to Motorola's MC qualification levels of 2000 volts Human Body Model (HBM) and 200 volts Machine Model (MM).

freescale™ semiconductor

3 Differences Overview

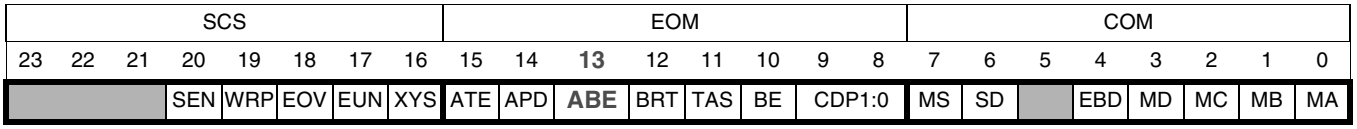
The primary functional differences between the two mask sets are due to inherent differences between the two design technologies. **Table 1** compares the mask sets.

Table 1. Functional Comparison of 4J22A and 0K36A

Feature	4J22A			0K36A		
	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

4 Asynchronous Bus Arbitration

In the Operating Mode Register (OMR) of the 0K36A 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 0K36A mask set of the DSP56303. The new ABE bit is highlighted in a bold red font.



ATE—Address Trace Enable
APD—Address Attribution Priority Disable
ABE—Async. Bus Arbitration Enable
BRT—Bus Release Timing
TAS—$\overline{TA}$ 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

SEN—Stack Extension Enable
WRP—Extended Stack Wrap Flag
EOV—Extended Stack Overflow Flag
EUN—Extended Stack Underflow Flag
XYS—Stack Extension Space Select

— 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.
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	Sta**ck 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 (AA[0–3]). 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.

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

Bit Number	Bit Name	Reset Value	Description
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.
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. **Notes:** **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.

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

Bit Number	Bit Name	Reset Value	Description
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.

4.1 Identification Registers

Values in the Device Identification register (IDR) and JTAG Identification (ID) register are changed to reflect the new 0K36A mask set, as shown in **Figure 2** and **Figure 3**. 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 0K36A. 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	0101	0011 0000 0011

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 0K36A. Version information corresponds to the revision number ($0 for revision 0, $1 for revision A, etc.).I)

31 28	27 22	21 12	11 1	0
Version Information	**Customer Part Number**	**Sequence Number**	**Manufacturer Identity**	**1**
0101	000110	0000000011	000 0000 1110	1

Figure 3. JTAG Identification Register Configuration (Revision 3)

4.2 Errata Removal

The 0K36A mask set removes all known functional errata on the 4J22A mask set. This section shows the functional errata that the 0K36A 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 in the chip errata documents on the web site listed on the back cover of this document.

The silicon errata present on the 4J22A mask but not on the 0K36A masks are listed in the following table.

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 0K36A**
ES36	Description: 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.	**Fixed on 0K36A**
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 (i.e., 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 0K36A**

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: 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.	**Fixed on 0K36A**
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 0K36A**

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: 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` • DSP56303 — `10011-11011` • DSP56307 — `10111-11111`	**Fixed on 0K36A**

Errata No.	Description	Status
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 0K36A**
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, i.e. 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 0K36A**

Errata No.	Description	Status
ES91	Description (added 7/22/98): If the Core reads data from the HRX while instructions are fetched from the memory Expansion Port (Port A) using 2 or more wait states, data may be lost. Workaround: There are three possible workarounds: 1) The host should guarantee that there is no more than one word in the TXH:TXM:TXL-HRX data path at any time. This can be achieved if the host writes a word to the HI08 only when the TRDY flag is set (i.e. the data path is empty). 2) Use a service routine running from fast (i.e. one wait state) external memory or internal memory to read the HRX read code; ensure that code that is fetched from slow (i.e. more than 1 wait state) external memory is located at least 4 instructions after the HRX register is read. For example: `READ_HRX_DATA` `NOP` `NOP` `NOP` `NOP` **Note:** a) Interrupt requests that fetch instructions from slow external memory should be masked during this service routine. Nonmaskable interrupt (NMI) request routines must not be in external memory. b) If running from fast external memory and if a DMA channel accessing external memory is used, then the DMA may cause extra wait states to the core. Thus, the DMA should have a lower priority than the core so that the core can access the external memory with no more than 1 wait state. 3) Read the HRX using one of the channels of the on-chip DMA controller.	**Fixed on 0K36A**
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 0K36A**

Errata No.	Description	Status
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, i.e. when the DACT bit in the DSTR register is cleared. **Note:** To perform this operation most efficiently, all other DMA channels should be disabled.	**Fixed on 0K36A**

Errata No.	Description	Status
ES105	Description (added 11/25/98): If the core clears HCIE bit on HCR register, while the interrupt was issued, and the vector was read by the interrupt controller, then when the interrupt is serviced, HCP will not be cleared, since the clear equation is conditioned by HCIE=1. Workaround: There are two possible workarounds: 1) If only host commands are used as possible interrupt source to the CORE (i.e. HTIE and HRIE are both 0), then instead of bit-set and bit-clear to HCIE, do bit-set and bit-clear instructions on IPRP register for the HIE bit. 2) If option "1" can not be used, then the user should first turn off host interrupt requests (all the possible sources) by clearing HIE bit on IPRP register, then issue 6 NOP instructions, then clear HCIE bit on HCR, issue another 6 NOP instructions and finally re-enable the HIE bit on IPRP as shown below: `;; Clear the relevant bits on IPRP register according to` `;; the current host interrupt priority settings` `BCLR #M_HPL0,x:M_IPRP` `BCLR #M_HPL1,x:M_IPRP` `;; Issue 6 NOP instructions` `NOP` `NOP` `NOP` `NOP` `NOP` `NOP` `;; Clear the HCIE bit on HCR register to turn off host` `;; commands` `BCLR #M_HCIE,x:M_HCR` `;; Issue 6 NOP instructions` `NOP` `NOP` `NOP` `NOP` `NOP` `NOP` `;; Restore the required host interrupt level` `BSET #M_HPL0,x:M_IPRP` `BSET #M_HPL1,x:M_IPRP`	**Fixed on 0K36A**

Errata No.	Description	Status
ES108	Description (added 12/12/98): The Timer's TIOx pin does not function properly in WatchDog mode while the $\overline{\text{RESET}}$ pin is asserted. The TIOx pin is tri-stated immediately after $\overline{\text{RESET}}$ assertion; it should be driven for 2.5 internal clock cycles according to the specification (section 9.4.4.1 and 9.4.4.2 of the *User's Manual*). Workaround: Provide external logic in order to extend the TIOx pin assertion.	**Fixed on 0K36A**
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.	**Fixed on 0K36A**

Errata No.	Description	Status
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`	Fixed on 0K36A

Errata No.	Description	Status
ES115 cont.	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 0K36A
	;; 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 0K36A

5 Documentation Errata

The following table lists all documentation errata present on the 0H82G mask of the DSP56303.

Errata No.	Description	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-4 "Peripheral pipeline restrictions.	**0K36A**
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	**0K36A**
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	**0K36A**

Errata No.	Description	Mask
ED9	Description (added 1/27/98): 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. Workaround: a. Enable an SCI pin other than SCLK. b. In the next instruction, enable the remaining SCI pins, including the SCLK pin. **Pertains to:** UM, SCI Chapter (Use the 302 UM as your reference, Section 8.4.2, “SCI Initialization”)	**0K36A**

Errata No.	Description	Mask
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 114 WR_ deassertion time 0.5 x TC - 4.0 [WS = 1] 1.5ns TC - 2.0 [2 ≤WS ≤3] 6ns 2.5 x TC - 4.0 [4 ≤WS ≤7] 21ns 3.5 x TC - 4.0 [WS Š 8] 31ns **Pertains to**: Data Sheets; timing 114 is in the table on SRAM Read and Write Accesses in the Port A section entitled “External Memory Expansion Port (Port A). The table number is 2-8. Timings 321 and 319 are in the section on Host Interface Timing, Table 2-20.	**0K36A**
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.	**0K36A**

Errata No.	Description	Mask
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 **Pertains to:** Data Sheet, Chapter 2, “DC Electrical Characteristics” table at very beginning of chapter.	**0K36A**
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: **Pertains to:** Data sheet, under “DC Electrical Characteristics” at beginning of Chapter 2. Typically, the table number is 2-3.	**0K36A**
ED21	Description (added 11/24/98): In the Technical Datasheet Iol should be listed as 1.6 mA, not as 3.0 mA. Workaround: **Pertains to:** Data sheet, under “DC Electrical Characteristics” near beginning of Chapter 2. Typically, the table number is 2-3. See Output Low Voltage in the “Characteristics” column.	**0K36A**
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: **Pertains to:** Data Sheet, under “DC Electrical Characteristics” near beginning of Chapter 2. Typically, the table number is 2-3. (NOTE: Look at the way this is handled in the 303 data sheet, Revision 2. Notice that the Min and Max columns are empty.)	**0K36A**

Errata No.	Description	Mask
ED26	Description (added 1/6/99): The specification DMA Chapter is wrong. “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.” **Pertains to:** DSP56300 Family Manual, Note1 in Section 8.1.6.1, “DSTR DMA Channel Transfer Done Status (DTD) - bits 5-0.”	**0K36A**
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.”	**0K36A**

Errata No.	Description	Mask
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:** UM, Section 7.4.1.7, “CRA Word Length Control.” The table number is 7-2.	**0K36A**
ED30	Description (added 9/12/1997; identified as a Documentation errata 2/1/99): 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. 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:** UM, Section 7.5.4.1, “Normal/On-Demand Mode Selection.”	**0K36A**
ED31	Description (added 9/12/1997; modified 9/15/1997; identified as a Documentation errata 2/1/99): 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. **Note:** 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 GPIO. **Pertains to:** UM, Section 7.4.2.4, “CRB Serial Control Direction 2 (SCD2) Bit 4”	**0K36A**

Errata No.	Description	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.”	**0K36A**

Errata No.	Description	Mask
ED33/ ES109	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 (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` `.....` `.....`	**0K36A**

Errata No.	Description	Mask
ED33/ ES109 cont.	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	**0K36A**

Errata No.	Description	Mask
ED33/ ES109 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	**0K36A**

Errata No.	Description	Mask
ED33/ ES109 cont.	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` `.....` `.....`	**0K36A**

Errata No.	Description	Mask
ED33/ ES109 cont.	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"	**0K36A**

Errata No.	Description	Mask
ED34/ ES110	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 (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.	**0K36A**
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).	**0K36A**

Errata No.	Description	Mask
ED40	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 DALU 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. **Pertains to:** *DSP56300 Family Manual,* Rev. 2, Section A.3, “Instruction Sequence Restrictions.”	**0K36A**

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

EB340
Rev. 3
2/2006

freescale™
semiconductor