# 56F826

*Chip Errata*

# 56F826 Digital Signal Controller

This document reports errata information on chip revision C. Errata numbers are in the form n.m, where n is the number of the errata item and m identifies the document revision number. This document is a pre-publication draft.
**Note:** **Differences between Chip Revisions are listed on page 8** and errata information for chip revisions prior to revision C have been archived and can be requested from Motorola Sales.

## Chip Revision C Errata Information:

The following errata items apply only to Revision C 56F826 devices. These parts are either marked as DSP56F826 or as PC56F826 with date codes of 0137 or (bottom line of marking).

| Errata Number | Description | Impact and Work Around |
|---|---|---|
| 1.3 | GPIO interrupts may be missed when clearing other interrupts. | Impact:<br>Clearing the interrupt can unintentionally result in clearing another interrupt occurring between the time the status register is read and written back to clear the interrupt.<br><br>Work Around:<br>None |
| 2.3 | Writes to internal XRAM during the first cycle out of reset does not work properly. | Impact:<br>Does not write to the XRAM during the first cycle out of reset.<br><br>Work Around:<br>To insert a NOP at the program entry point or move any other type of instruction onto the first location. |
| 3.3 | N register is not available in the cycle immediately after it is changed. | Impact:<br>In the case of an index+ offset move into the N register, N is not available in the cycle immediately following the change in value.<br>Example:<br>move x:( r2+ 3), N<br>lea (R2) +N<br><br>Work Around:<br>A no-operation (NOP) or some other instruction that does not use the N register will need to be inserted between the two statements. As an aid the assembler will be modified to flag this as a problem. |

## Chip Revision C Errata Information:

The following errata items apply only to Revision C 56F826 devices. These parts are either marked as DSP56F826 or as PC56F826 with date codes of 0137 or (bottom line of marking).

| Errata Number | Description | Impact and Work Around |
|---|---|---|
| 4.3 | While in operation if TOD is disabled and the value written onto the TOD alarm registers happens to match with the value on the counters, a TOD alarm interrupt is raised. | Impact:<br>Generates an interrupt even if TOD is not enabled.<br><br>Work Around:<br>When disabling TOD, also disable the TOD alarm bit. When using TOD, configure the alarm registers, wait for the first one second interrupt after enabling the TOD and then enable the alarm interrupt. |
| 5.3 | PLL Stabilization Time | Impact:<br>Maximum PLL stabilization time is 200ms under worst case (-40°C) conditions. Typical PLL stabilization time remains at 10ms (25°C and above).<br><br>Work Around:<br>Insert a 200ms delay after power up to allow the PLL to settle or verify the Loss of Lock bits (LCK0 and LCK1) in the PLL Status Register (PLLSR) are set to 1 prior to program execution. |
| 6.4 | A one second first alarm value causes an alarm interrupt after two seconds instead of one. | Impact:<br>A one second first alarm value will cause an alarm interrupt after two seconds instead of one. The second occurrence of this alarm will trigger correctly. For example, if the initial day/hr/min/sec counters are set to zero and the alarm registers set to 1 second. Alarms will happen at 00:00:02, 00:01:01, 00:02:01, etc.<br><br>Work Around:<br>The SDK adds one second to the initial seconds counter time to cause the first alarm to occur after one second instead of two. |
| 7.8 | The interrupt controller uses COPR bit in SIM_RSTSTS register to determine whether to use COP reset vector in the vector table. | Impact:<br>The user must clear this bit at startup after a COP reset, or any subsequent resets will use the COP reset vector.<br><br>Work Around:<br>Clear the COP Reset bit in the SIM STATUS register. |
| **SPI Errata Items** | | |
| 8.3 | Slave Mode SPI TE (transmitter empty) flag set too early. | Impact:<br>The problem only occurs in Slave mode when CPHA = 0. The Transmitter Empty (TE) flag may be set too early, thus allowing the user software to write a new data value into the transmit buffer before the current contents are loaded into the transmit shift register.<br><br>Work Around:<br>Use the receiver full flag as an indication of when to write new data into the transmit buffer. |

## Chip Revision C Errata Information:

The following errata items apply only to Revision C 56F826 devices. These parts are either marked as DSP56F826 or as PC56F826 with date codes of 0137 or (bottom line of marking).

| Errata Number | Description | Impact and Work Around |
|---|---|---|
| 9.3 | Slave Mode SPI transmit shift register data corruption. | Impact:<br>The problem only occurs in Slave mode when an external master has deselected the internal SPI ($\overline{SS}$ =1) but it has provided a shift clock to the internal SPI. This scenario is expected in a SPI system with multiple slave devices. The deselected SPI slave transmitter shift register will shift in response to the applied shift clock. This action will cause the existing data in the transmitter shift register to become corrupted.<br>Work Around:<br>Modify communications protocol so the first word returned by the Slave after being reselected ($\overline{SS}$ =1 to 0) is discarded. The second and subsequent data words after being reselected are valid. |
| 10.7 | SPI halts on receiver overflow when being used to transmit only. | Impact:<br>Same as description.<br><br>Work Around:<br>Read receiver in transmit Interupt Service Routine (ISR). |
| 11.7 | SPI misses one data word by double loading Xmit register when double buffering. | Impact:<br>Same as description.<br><br>Work Around:<br>Use only single buffering inside ISR |
| 12.7 | Bit counter is not reset on each transmission. | Impact:<br>Must reset part if external master malfunctions in this way. This only happens in slave mode and if external master generates extra clocks.<br><br>Work Around:<br>External, master SPI must be working correctly and not generate extra clocks. |
| 13.7 | Value from data transmit register not moved to shift register. | Impact:<br>When using CPHA=0, the value from the data transmit register does not move to the shift register when the value has been double buffered by a previous transmission.<br><br>Work Around:<br>Use CPHA=1 |
| 14.7 | SPI receiver shift register residual after overflow resulting in duplicate transmission. | Impact:<br>Same as description.<br>Work Around:<br>The software should mask the value to the expected word length during access to the receive register. |

## Chip Revision C Errata Information:

The following errata items apply only to Revision C 56F826 devices. These parts are either marked as DSP56F826 or as PC56F826 with date codes of 0137 or (bottom line of marking).

| Errata Number | Description | Impact and Work Around |
|---|---|---|
| 15.7 | Intermittent duplicate transmission in slave mode when CPHA=0. | Impact:<br>Same as description.<br>Work Around:<br>The problem can be eliminated if the SPE control bit is toggled after the SPI receiver full flag is set in the Slave mode or the baud rates are slow enough on the Master side SPI such that the $\overline{SS}$ signal can be de-asserted before the last inactive edge of the SPICK signal. |
| 16.7 | While SPI is enabled in master mode and the transmit data register (less than 16 bits) is filled, the OVRF flag stops further transmission. | Impact:<br>This occurs since the clock is common to MOSI and MISO and “1s” are latched into the SPI data register. After transmitting two words, OVRF flag is set while the SPI receive data register is not read by the software.<br>To re-initiate transmission, SPI needs to be disabled and then re-enabled. Clearing the OVRF flag will not re-initiate transmission at this point.<br><br>Note: This does not occur with 16-bit words.<br>Work Around:<br>Always use read receive data, even if it is to be discarded. |
| **SSI Errata Items** | | |
| 17.3 | When transmit FIFO empty or under run, previously sent data is not retransmitted. Instead previous data in next FIFO is transmitted. | Impact:<br>SSI sends “unknown” data when Transmit empty or under run.<br><br>Work Around:<br>Avoid under run by re-filling the Tx FIFO before transfering the last word in the Tx FIFO. |
| 18.3 | When Tx frame and clock are generated internally, SSI generates an extra STFS pulse even after the SSI transmitter (TE bit) has been disabled. | Impact:<br>SSI sends “unknown” data even though Tx is disabled. This occurs with early frame sync configuration.<br><br>Work Around:<br>Disable the SSI EN bit for the next frame once the TE is disabled. (provided there is no valid data to be received in asynchronous mode of operation). |
| 19.3 | In network mode, Tx frame and clock are generated externally. In TXD goes to inactive (tri-stated) after one IP_CLK clock period. | Impact:<br>Leads to a problem when SSI has given up the Mastership if any other device drives the TXD pin during this clock period.<br><br>Work Around:<br>Put idle cycle for this clock period. |
| 20.3 | Tx frame can go out of sync. | Impact:<br>In normal mode, Tx frame and clock are generated externally. If the frame is not anticipated in the periodic manner and clock is running countinuously, data will go out of sync with the frame.<br><br>Work Around:<br>External frame generation should be dependent on divide clock (DC) value.<br>Example: 16 bit data, must have frame sync every 16 bits. |

## Chip Revision C Errata Information:

The following errata items apply only to Revision C 56F826 devices. These parts are either marked as DSP56F826 or as PC56F826 with date codes of 0137 or (bottom line of marking).

| Errata Number | Description | Impact and Work Around |
|---|---|---|
| 21.3 | SSI SYNRST can clear RX_FIFO even when RE is cleared. | Impact:<br>When disabling the Receiver during the last word of a frame, data may be lost because the frame sync signal, while not output to the pin, is generated and has internal effects.<br><br>Work Around:<br>When clearing RE, also clear SYNRST in the SOR. |
| 22.3 | The SSI RFS and TFS flags are not synchronized with the system clock. | Impact:<br>RFS asserts differently for internal/external frames.<br><br>Work Around:<br>none. |

## Chip Revision C Errata Information:

The following errata items apply only to Revision C 56F826 devices. These parts are either marked as DSP56F826 or as PC56F826 with date codes of 0137 or (bottom line of marking).

| Errata Number | Description | Impact and Work Around |
|---|---|---|
| 23.4 | A pipeline dependency problem occurs on the secondary data RAM bus when the read addresses for the second parallel read crosses a 1K address boundary.<br><br>Instruction sequences which can induce this problem are any combination of the following: Dual Move or Dual Move coupled with any of MAC, MACR, ADD, SUB, MPY and MPYR.<br><br>The memory boundaries that are at issue for the DSP56F826 are 0X3FF-0X400, 0X7FF-0X800, 0XBFF-0XC00. | Impact:<br>This problem is only applicable to assembly coded algorithms and will not occur in algorithms written entirely in C. It is also applicable when C code uses certain SDK libraries or primitives that are written in assembly when the data structure is not a local variable (i.e. not allocated on the stack). Typically, the second read of a dual parallel read is for coefficients in a FIR filter. Calculating FIRs is the context of all of the SDK primitives affected by this condition, with one exception (xfr16mult).<br><br>SDK libraries that are affected by this erratum are V.22, Caller ID, G165 and DSP Function library (dfr16FIR, dfr16FIRs, dfr16FIRInt, dfr16FIRDec and xfr16Mult). For more details please see Motorola FAQs @ http://www.motorola.com/semiconductors under Technical Support & Contacts.<br><br>Example:<br>`mac y0,x0,a x:(r1)+,y0 x:(r3)+,x0`<br>`mac y0,x0,a x:(r1)+,y0 x:(r3)+,x0`<br>yields a different final result in A than,<br>`mac y0,x0,a x:(r1)+,y0 x:(r3)+,x0`<br>`nop`<br>`mac y0,x0,a x:(r1)+,y0 x:(r3)+,x0`<br>when the initial value of R3 is 0X3FF (1023).<br><br>The same problem occurs when R3 is decremented to move from above a 1K boundary to below.<br><br>This problem only occurs on sequential read accesses which use the secondary data bus and which cross a 1K word boundary on internal data RAM. The problem results from incorrect logic used to mediate between multiple internal 1Kx16 memory blocks.<br><br>Work Around:<br>Work arounds for code written by the user are (in order of preference):<br>• Move constant coefficient tables from data RAM to data FLASH. (Define coefficients as *const* in appconst.c, then the linker command file will automatically put them into data FLASH.)<br>• Don't change your project but check the link map located in the debug folder to see that no coefficient table crosses a 1K boundary in data RAM. (If your project is MyProject.mcp, you will find the file MyProject.elf.xMAP in the debug daughter folder of your Codewarrior project.)<br>• Dynamically allocate coefficient tables (out of internal memory heap space) but modify the linker command file, breaking heap space crossing a 1K boundary into two separate pieces of memory.<br>• Statically allocate coefficient tables so that you don't have to worry about the location of them in heap space. Then modify the linker command file so that coefficient tables (or equivalent) do not cross a 1K boundary in data RAM. (This may require moving coefficients into separate source files.)<br>• Define coefficient tables as local variables so that they are allocated out of stack space. Verify that stack space (as defined by the linker command file) does not cross a 1K boundary. (This approach is OK if you don't do a lot of filtering and filters are relatively short.)<br>• Add a NOP or other instruction between the sequential accesses. |

## Chip Revision C Errata Information:

The following errata items apply only to Revision C 56F826 devices. These parts are either marked as DSP56F826 or as PC56F826 with date codes of 0137 or (bottom line of marking).

| Errata Number | Description | Impact and Work Around |
|---|---|---|
| 24.5 | In network mode, when using externally generated clocks and a word wide tx frame sync, there is a potential transmit contention problem. If the SSI transmitter is disabled during the first timeslot (with a write to the STSR register) the SSI incorrectly drives a zero during the last bit time of the timeslot. | Impact:<br>This will cause a problem with any other device trying to use the first timeslot of the frame. Our SSI will corrupt the last bit of the first timeslot.<br><br>Work Around:<br>Tx data is not corrupted when a bit wide frame sync is used and this is the recommended solution. If a word wide frame sync must be used (both solutions below waste data bandwidth):<br>1. then the first timeslot should not be used as data will be corrupted<br>2. the word size should be configured to be wider than actually needed (configure as 12 bit wide for 10 bit word size). |
| 25.9 | With a Quad Timer counter, when using a single compare register to generate timing intervals and clocking the timer at a rate other than at the IPbus_clock rate the timer may count incorrectly when the compare register is changed. | Impact:<br>When the compare register matches the counter register and is updated before the next timer clock the counter increments/decrements instead of reloading.<br><br>Work Around:<br>1. Use both compare registers, such that the compare register that is not active is updated for use in the next count period.<br>2. Instead of updating the compare register, architect the software so the LOAD register can be updated, with the compare register held constant.<br><br>A more in depth FAQ can be found on the Freescale website. *freescale.com* |

## Differences between Chip Revisions

| **Chip Rev. A**<br>*Date codes = ≥ 0108 ≤ 0110* | **Chip Rev. B**<br>*Date codes = ≥ 0111 ≤ 0136* | **Chip Rev. C**<br>*Date codes = ≥ 0137* |
|---|---|---|
| Device can not meet flash data retention specification of 10 years. | Improved flash data retention. Device can meet a data retention specification of 10 years at 25$^{o}$C after 10,000 program-erase cycles. | **Corrected** |
| GPIO interrupts may be missed when clearing other interrupts.<br>*See errata item 1 for clarification.* | Same as A | Same as B |
| Writes to internal XRAM during the first cycle out of reset does not work properly.*See errata item 2 for clarification.* | Same as A | Same as B |
| N register is not available in the cycle immediately after it is changed<br>*See errata item 3 for clarification.* | Same as A | Same as B |
| While in operation if TOD is disabled and the value written onto the TOD alarm registers happens to match with the value on the counters, a TOD alarm interrupt is raised.<br>*See errata item 4 for clarification.* | Same as A | Same as B |
| Flash access time can be marginal. | **Corrected** | |
| Intermittent incorrect data return from Data, Program and BootFLASH modules. | **Corrected** | |
| Intermittent internal data (X memory) RAM corruption. | No intermittent internal data (X memory) RAM corruption. **Corrected** | |
| Access to the upper byte of some of the TOD registers gives an incorrect value. | **Corrected** | |
| PLL Stabilization Time<br>*See errata item 5 for additional information.* | Same as A | Same as B |
| A one second first alarm value causes an alarm interrupt after two seconds instead of one.<br>*See errata item 6 for additional information.* | Same as A | Same as B |
| Slave Mode SPI data is corrupted on the MISO output | Same as A | **Corrected** |
| Interrupt controller uses COPR bit in SIM_RSTSTS register to determine whether to use COP reset vector in the vector table<br>*See errata item 7 for additional information.* | Same as A | Same as B |
| Slave Mode SPI TE (transmitter empty) flag set too early.<br>*See errata item 8for additional information.* | Same as A | Same as B |

## Differences between Chip Revisions

| **Chip Rev. A**<br>*Date codes = ≥ 0108 ≤ 0110* | **Chip Rev. B**<br>*Date codes = ≥ 0111 ≤ 0136* | **Chip Rev. C**<br>*Date codes = ≥ 0137* |
|---|---|---|
| Slave Mode SPI transmit shift register data corruption.<br>*See errata item 9 for additional information.* | Same as A | Same as B |
| SPI halts on receiver overflow when being used to transmit only.<br>*See errata item 10 for additional information.* | Same as A | Same as B |
| SPI misses one data word by double loading Xmit register when double buffering.<br>*See errata item 11 for additional information.* | Same as A | Same as B |
| Bit counter is not reset on each transmission.<br>*See errata item 12 for additional information.* | Same as A | Same as B |
| Value from data transmit register not moved to shift register.<br>*See errata item 13 for additional information.* | Same as A | Same as B |
| SPI receiver shift register residual after overflow resulting in duplicate transmission.<br>*See errata item 14 for additional information.* | Same as A | Same as B |
| Intermittent duplicate transmission in slave mode when CPHA=0.<br>*See errata item 15 for additional information.* | Same as A | Same as B |
| While SPI is enabled in master mode and the transmit data register (less than 16 bits) is filled, the OVRF flag stops further transmission.<br>*See errata item 16 for additional information.* | Same as A | Same as B |
| SSI sends “unknown” data when Transmit empty or under run.<br>*See errata item 17 for additional information.* | Same as A | Same as B |
| SSI sends “unknown” data even though Tx is disabled.<br>*See errata item 18 for additional information.* | Same as A | Same as B |
| In network mode, if Tx clock and frame are generated externally, the first word of each frame has TXD active if the first slot is invalid.<br>*See errata item 19 for additional information.* | Same as A | Same as B |
| Tx frame can go out of sync.<br>*See errata item 20 for additional information.* | Same as A | Same as B |
| SSI SYNRST can clear RX_FIFO even when RE is cleared.<br>*See errata item 21 for additional information.* | Same as A | Same as B |

## Differences between Chip Revisions

| **Chip Rev. A**<br>*Date codes = ≥ 0108 ≤ 0110* | **Chip Rev. B**<br>*Date codes = ≥ 0111 ≤ 0136* | **Chip Rev. C**<br>*Date codes = ≥ 0137* |
|---|---|---|
| The SSI frame sync setup and hold requirements limit input timing when in slave mode. | Same as A | **Corrected** |
| The SSI RFS and TFS flags are not synchronized with the system clock.<br>*See errata item 22 for additional information.* | Same as A | Same as B |
| A pipeline dependency problem occurs on the secondary data RAM bus when dual parallel reads to XRAM in adjacent instruction cycles, and the read addresses straddle a 1K address boundary.<br>*See errata item 23 for additional information.* | Same as A | Same as B |
| In network mode, when using externally generated clocks and a word wide tx frame sync, there is a potential transmit contention problem.<br>*See errata item 24 for additional information.* | Same as A | Same as B |
| When the compare register matches the counter register and is updated before the next timer clock the counter increments/decrements instead of reloading.<br>*See errata item 25 for additional information.* | Same as A | Same as B |

## *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 Muenchen, 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


RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics of their non-RoHS-compliant and/or non-Pb-free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to http://www.freescale.com/epp.

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 that 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.





DSP56F826E
Rev. 11.0
12/2005