MAX96793 DEV_REV=6 (C-0D) ERRATA SHEET

Corresponds to data sheet 19-101163; Rev 3; 4/25

DEV_REV=6 (per reading of register 0x0E)

The errata listed below describe situations where components of this revision perform differently than expected or differently than described in the data sheet. Analog Devices may, at its own discretion, take future steps to correct these errata when the opportunity to redesign the product presents itself. Prior to that, Analog Devices has determined the following potential workarounds that customers may want to consider when addressing one of the situations described below.

This errata sheet only applies to components of this revision. These components are branded on the topside of the package with a four-digit code in the form yyww, where yy and ww are two-digit numbers representing the year and work week of manufacture, respectively. The revision of these components can be found by reading DEV_REV = 6 from register 0xE.

1) Functional problems with GPIO not transmitting static signals after enabling

Description:
When GPIO transmission and reception are enabled, if no GPIO value transition occurs, the static GPIO value may not be transmitted to the other side of the link. That is, if the value change occurs before GPIO forwarding is enabled, it may not be picked up and sent to the other side.

Workaround:
If the GPIO is expected to be 1 before enabling, write GPIO_TX_EN to 1 first, and then write GPIO_RX_EN to 1. If the GPIO is expected to be 0 before enabling, write GPIO_RX_EN to 1 first, and then write GPIO_TX_EN to 1. This will ensure the static GPIO value is propagated to the other side.

Resolution:
No silicon fix is planned.

2) UART pass through may output random data when link loses lock

Description:
This issue occurs when using the UART pass-through (tunnel) mode and the link is lost, which may cause the UART to output random data for a short period of time after the connection is lost.

Workaround:
Disable the UART pass-through before resetting the GMSL link. Monitor LOCK and if lock is lost, UART data is invalid.

Resolution:
No silicon fix is planned.

3) ADC GPIO input mux BIST issue

Description:
The built-in-self-test (BIST) of the ADC's input multiplexer (MUX) is not functional.

Workaround:
The ADC's input MUX functionality can be manually verified through user software. Contact the factory for instructions and a script.

Resolution:
No silicon fix is planned.

4) Eye Opening Monitor and Eye Mapper inaccuracy at elevated junction temperature

Description:
At junction temperatures above approximately 80°C, some serializer devices may show anomalies in the eye mapper, and the eye-opening monitor (EOM) may report incorrect eye width. The eye opening may be reported as smaller (including 0 UI) or larger (> 1UI) than it is. This does not affect device operation and is a reporting issue only.

Due to this erratum, EOM reporting to ERRB should be disabled by ensuring bit EOM_ERR_OEN_A = 0 in register 0x1C. EOM_ERR_OEN_A = 0 by default.

Register 0x1406 must be left at the default value of 0x80. EOM_RST_THR[6:0] in register 0x1406 must not be modified because this could result in accidental resetting of the link.

The reverse channel receiver does not rely on the EOM circuitry for proper operation. Thus, the primary data path is unaffected by this issue.

Workaround:
None.

Resolution:
No silicon fix is planned.

5) GMSL3 PAM4 short-channel operation requires register write for robust operation

Description:
For robust GMSL3 PAM4 operation on channels with less than 1dB of insertion loss, a register write is required.

Workaround:
FFE must be disabled by writing bit enffe=0 in register 0x14CE.

Resolution:
No silicon fix is planned.

6) Enable GMSL negative output (SION pin) in coax mode for optimal performance

Description:
For optimal GMSL link signal fidelity in coax mode, a register write is required in GMSL3 12Gbps mode and recommended in GMSL2 6Gbps mode to enable the negative GMSL output pin (SION). This improves the eye quality of the SIOP transmitted signal. The SION pin must be terminated as shown in the device datasheet.

Workaround:
Enable the negative output by writing bits enminus_man = 1 and enminus_reg = 1 in register 0x14CE.

Resolution:
No silicon fix is planned.

7) MIPI Receiver Requires Reset on Startup

Description:
In D-PHY mode, if the MIPI input clock and data pins are not stable when the device powers up, the MIPI receiver may get stuck and not properly interpret the D-PHY protocol start of transmission sequence. This can occur if the MIPI source feeding the serializer powers up after the serializer, as there may be glitches or skews on the various MIPI pins as they are powered up.

Workaround:
After the serializer is powered up, and before video is enabled, write the mipi_rx_reset bit to 1 and then back to 0. This will reset the MIPI receiver and allow it to interpret the D-PHY start of transmission when video starts.

Resolution:
No silicon fix is planned.

8) After executing Sleep/Wake sequence, RESET_ALL puts part into Sleep state.

Description:
After a Sleep/Wake sequence has been executed, writing RESET_ALL=1 will put the part into Sleep state and this will cause the part not being fully reset to its POR values. The part can be woken up using the standard local or remote Wake commands. However, after the Wake sequence, the part will be in the state stored in the retention memory during the previous Sleep command instead of being reset to its POR settings. Registers not stored in retention memory will not be affected by this and will be reset to the POR values.

Workaround:
PWDNB pin can be used to fully reset the part and restore all registers to their POR settings.

Resolution:
No silicon fix is planned.

9) ADC Voltage Divider Always Active for MFP5 and MFP6

Description:
MFP5 and MFP6 can each be activated as an ADC input, general purpose input, general purpose open-drain output, or as a GMSL line fault monitor. These MFPs contain an internal voltage divider to reduce the input voltage to a valid range ($V_{IN}<V_{REF}$). MFP5 and MFP6 dividers are always active and equal to the programmed voltage divider setting (ADC_CTRL_2.adc_div, ADC_LIMIT*y*_3.div_sel*y, y* from 0 to 7) even when the MFP is not programmed as an ADC input. The maximum divide ratio of four loads the MFP with 40k ohm to GND. Note that the ADC divider setting affects all ADC inputs, so if MFP3 is used as an ADC input, MFP5 and MFP6 will use that divider setting.

MFP3 is not affected by the divider setting when not programmed as an ADC input.

Workaround:
When MFP5 or MFP6 are used for any function other than an ADC input, it is recommended to leave the ADC divider setting at '/1' and perform any necessary voltage division externally for all MFPs used as ADC inputs.

An alternative workaround for the programmable voltage divider setting of '/2' is to set the ADC scale bit (ADC_CTRL_1.adc_scale) to logic 1, which globally scales the ADC input down by two without resistive MFP loading. The ADC digital code is scaled down by two in this mode.

Resolution:
No silicon fix is planned.

10) ADC Round Robin State Machine Sets Voltage Dividers to '/3'

Description:
The ADC Round Robin State Machine changes the ADC voltage dividers to '/3' at start-up. This causes MFP5 and MFP6 to be loaded with 45KΩ to GND for 1ms after start-up of the round robin state machine. MFP3 will also be loaded if activated as an ADC input. When the state machine is deactivated, the ADC dividers are restored to the setting in ADC_CTRL_2.adc_div.

Workaround:
None.

Resolution:
No silicon fix is planned.

11) Reading the adc_done_if register bit before ADC conversion is finished can keep it from being asserted (Errata ID=830002)

Description:
Reading the adc_done_if register bit before ADC conversion is finished can keep it from being asserted for the current conversion operation.

Workaround:
Option 1:
To ensure ADC conversion is complete, allow minimum 430us delay after setting cpu_adc_start (ADC_CTRL_0.b0 = 1'b1) and before reading adc_done_if (ADC_INTR0.b0). Do not read ADC conversion done interrupt flag (adc_done_if) before the conversion is done. See Appendix A for an example implementation.

Option 2:
Before the ADC conversion starts, set the adc_done_ie (ADC_INTRIE0.b0 = 1'b1) to assert interrupt flag when the conversion has completed. This will route the adc_done_if (ADC_INTR0.b0) to ADC_INT_FLAG (INTR7.b2). After the start of the conversion (ADC_CTRL_0.b0 = 1'b1), poll ADC_INT_FLAG until asserted indicating the ADC conversion has completed. Reading this ADC_INT_FLAG will not prevent the adc_done_if from asserting. After the ADC_INT_FLAG has asserted, the ADC_INTR0-3 will need to be read to clear all enabled ADC interrupts. See an example below.

```
# Enable ADC done interrupt enable: adc_done_ie (ADC_INTRIE0.b0 = 1'b1)
WIB 80 50C 01
# Start an ADC conversion: cpu_adc_start (ADC_CTRL_0.b0 = 1'b1)
WIB 80 500 1F
# Read the ADC interrupt flag: ADC_INT_FLAG (INTR7.b2)
RIB 80 1F 1
# Read the ADC done interrupt flag: adc_done_if (ADC_INTR0.b0)
RIB 80 510 1
# Read the ADC_INTR1
```

RIB 80 511 1
Read the ADC_INTR2
RIB 80 512 1
Read the ADC_INTR3
RIB 80 513 1

Resolution:
No silicon fix is planned. Datasheet has been updated.

Appendix A – ADC GPIO Input Verification Test

The following script can be used via software to verify the ADC's input multiplexer.

Step 1: ADC Power Up (Common to all ADC GPIO input verification tests)

```
# Reset the device (all reset)
WIB 80 10 80
# enable the ADC clock
WIB 80 501 08
# Enable ADC conversion done interrupt, ADC ready interrupt enables
WIB 80 50C 03
# Power on the ADC enabling the input buffer, the charge pump, internal reference, and ADC
WIB 80 500 1E
# Turn on Input EN
WIB 80 502 01
# Select ADC0 (keep ADC clock enable on)
WIB 80 501 08
# Initiate temperature conversion
WIB 80 1D28 01
# Wait delay minimum 430us for temperature conversion
# Enable the MUX verification bit
WIB 80 1D28 10
```

Step 2: For each active ADC input, run appropriate script below.

Step 2a: ADC0 (MFP3 Pin) Input Verification Test:

```
# Enable ADC0 input
WIB 80 53E 01
# Drive the test level for the input MUX to a low all other inputs to a high
WIB 80 1D37 FE
# Start an ADC conversion
WIB 80 500 1F
# Wait delay minimum 430us for ADC conversion. Do not read ADC conversion done interrupt flag
(adc_done_if) before the conversion is done.
# Read the ADC conversion back (read addresses 0x508 and 0x509)
# The reconstructed code should be within 15LSB of 0x000 when constructed from {0x509.1-0,0x508}
RIB 80 508 2
# Switch input levels from input MUX to a high and all other inputs to a low
WIB 80 1D37 01
# Start an ADC conversion
WIB 80 500 1F
# Wait delay minimum 430us for ADC conversion. Do not read ADC conversion done interrupt flag
(adc_done_if) before the conversion is done.
# Read the ADC conversion back (read addresses 0x508 and 0x509)
```

```
# The reconstructed code should be within 35LSB of 0x3E6 when constructed from {0x509.1-0,0x508}
RIB 80 508 2
```

Step 2b: ADC1 (MFP5 Pin) Input Verification Test:

```
# Enable ADC1 input
WIB 80 53E 02
# Drive the test level for the input MUX to a low all other inputs to a high
WIB 80 1D37 FD
# Start an ADC conversion
WIB 80 500 1F
# Wait delay minimum 430us for ADC conversion. Do not read ADC conversion done interrupt flag
(adc_done_if) before the conversion is done.
# Read the ADC conversion back (read addresses 0x508 and 0x509)
# The reconstructed code should be within 15LSB of 0x000 when constructed from {0x509.1-0,0x508}
RIB 80 508 2
# Switch input levels from input MUX to a high and all other inputs to a low
WIB 80 1D37 02
# Start an ADC conversion
WIB 80 500 1F
# Wait delay minimum 430us for ADC conversion. Do not read ADC conversion done interrupt flag
(adc_done_if) before the conversion is done.
# Read the ADC conversion back (read addresses 0x508 and 0x509)
# The reconstructed code should be within 35LSB of 0x3E6 when constructed from {0x509.1-0,0x508}
RIB 80 508 2
```

Step 2c: ADC2 (MFP6 Pin) Input Verification Test:

```
# Enable ADC2 input
WIB 80 53E 04
# Drive the test level for the input MUX to a low all other inputs to a high
WIB 80 1D37 FB
# Start an ADC conversion
WIB 80 500 1F
# Wait delay minimum 430us for ADC conversion. Do not read ADC conversion done interrupt flag
(adc_done_if) before the conversion is done.
# Read the ADC conversion back (read addresses 0x508 and 0x509)
# The reconstructed code should be within 15LSB of 0x000 when constructed from {0x509.1-0,0x508}
RIB 80 508 2
# Switch input levels from input MUX to a high and all other inputs to a low
WIB 80 1D37 04
# Start an ADC conversion
WIB 80 500 1F
# Wait delay minimum 430us for ADC conversion. Do not read ADC conversion done interrupt flag
(adc_done_if) before the conversion is done.
# Read the ADC conversion back (read addresses 0x508 and 0x509)
# The reconstructed code should be within 35LSB of 0x3E6 when constructed from {0x509.1-0,0x508}
RIB 80 508 2
```

Step 3: Reset for normal operation:

```
#
# Disable input MUX test
WIB 80 1D28 00
```

MAX96793
ERRATA SHEET

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	February 25, 2020	Initial release	—
1	March 9, 2020	Removed unneeded errata	
2	February 23, 2021	Added errata to enable SION	
3	April 26, 2021	Expanded EOM errata description Added MIPI reset requirement errata	
4	1/22	Added “After executing Sleep/Wake sequence, RESET_ALL puts part into Sleep state”. ADC voltage divide always active for MFP5 and MFP6 ADC Round Robin start-up in ‘/3/ state Add Script for ADC0/1/2 BIST Testing to Appendix	3-6
5	March 3, 2022	Removed errata requiring 12Gb/s operation to be booted first in 6Gb/s mode; no longer required.	2
6	May 2, 2025	Updated datasheet revision. Added statement about leaving register 0x1406 at default in “Eye Opening Monitor and Eye Mapper inaccuracy at elevated junction temperature” section Added new errata “Reading the adc_done_if register bit before ADC conversion is finished can keep it from being asserted (Errata ID=830002).” Changed “Enable ADC done” to “Enable ADC conversion done interrupt” in Step1 of Appendix A. Added “#Wait delay minimum 430us for temperature conversion” and “#Wait delay minimum 430us for ADC conversion. Do not read ADC conversion done interrupt flag (adc_done_if) before the conversion is done” in Appendix A – ADC GPIO Input Verification Test section.	1, 4-6