TMS320DM369
Digital Media System-on-Chip (DMSoC)
Silicon Revision 1.2

Silicon Errata

Literature Number: SPRZ441
August 2016

Contents

List of Figures

List of Tables

Silicon Errata
SPRZ441–August 2016

TMS320DM369 Digital Media System-on-Chip (DMSoC) Silicon Revision 1.2

1 Introduction

This document describes the known exceptions to the functional specifications for the TMS320DM369 Digital Media System-on-Chip (DMSoC). The updates are applicable to the ZCE package.

For additional information, see *TMS320DM369 Digital Media System-on-Chip (DMSoC)*.

This document also contains Usage Notes. Usage Notes highlight and describe particular situations where the device's behavior may not match presumed or documented behavior. This may include behaviors that affect device performance or functional correctness. These notes will be incorporated into future documentation updates for the device (such as the device-specific data sheet), and the behaviors they describe will not be altered in future silicon revisions.

1.1 Device and Development Support Tool Nomenclature

To designate the stages in the product development cycle, TI assigns prefixes to the part numbers of all DSP devices and support tools. Each DSP commercial family member has one of three prefixes: TMX, TMP, or TMS (for example, TMS320DM369). Texas Instruments recommends two of three possible prefix designators for its support tools: TMDX and TMDS. These prefixes represent evolutionary stages of product development from engineering prototypes (TMX/TMDX) through fully qualified production devices/tools (TMS/TMDS).

Device development evolutionary flow:

TMX Experimental device that is not necessarily representative of the final device's electrical specifications

TMP Final silicon die that conforms to the device's electrical specifications but has not completed quality and reliability verification

TMS Fully-qualified production device

Support tool development evolutionary flow:

TMDX Development support product that has not yet completed Texas Instruments internal qualification testing

TMDS Fully-qualified development support product

TMX and TMP devices and TMDX development support tools are shipped against the following disclaimer:

"Developmental product is intended for internal evaluation purposes."

TMS devices and TMDS development support tools have been characterized fully, and the quality and reliability of the device have been demonstrated fully. TI's standard warranty applies.

Predictions show that prototype devices (TMX or TMP) have a greater failure rate than the standard production devices. Texas Instruments recommends that these devices not be used in any production system because their expected end use failure rate still is undefined. Only qualified production devices are to be used.

1.2 Revision Identification

Figure 1 provides an example of device markings. The device revision can be determined by the symbols marked on the top of the package. Some prototype devices may have markings different from those illustrated.

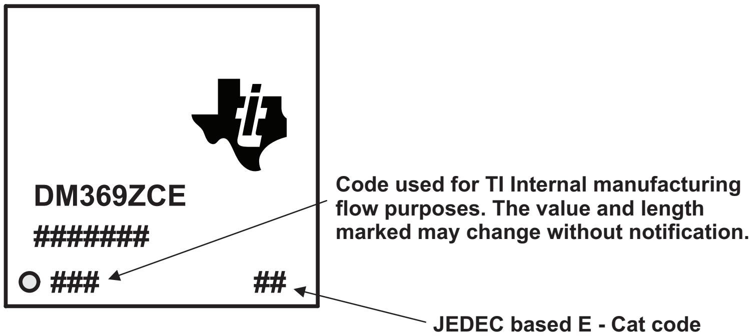

Figure 1. Example, Device Markings for TMS320DM369 (ZCE) [432 MHz]

NOTES:

(A) "#" denotes an alphanumeric character. "x" denotes an alpha character only.

Silicon revision is identified by a code on the chip as shown in Figure 1. Table 1 lists the silicon revisions associated with each device revision code for the device.

Table 1. Device Silicon Revisions

Device Revision Code (on above package symbol 'x')		SILICON REVISION	COMMENTS (part number association)
TMS	N/A	1.2	TMS320DM369ZCE, TMS320DM369ZCED, TMS320DM369ZCEF, TMS320DM369ZCEDF

2 Silicon Revision 1.2 Usage Notes and Known Design Exceptions to Functional Specifications

2.1 Usage Notes for Silicon Revision 1.2

Usage Notes highlight and describe particular situations where the device's behavior may not match presumed or documented behavior. This may include behaviors that affect device performance or functional correctness. These notes will be incorporated into future documentation updates for the device (such as the device-specific data sheet), and the behaviors they describe will not be altered in future silicon revisions.

2.1.1 ARM ROM Boot - Default MAC Address Enhancement for EMAC Boot Mode

Silicon Revision 1.2 supports an enhancement to the ARM ROM Boot-EMAC mode. In the case no magic number is found in the EEPROM, the EMAC boot mode will use a default MAC address. This feature was not present on Silicon Revision 1.1. For more details, see the *TMS320DM36x DMSoC ARM Subsystem Reference Guide*.

2.1.2 Peripherals: Electrostatic Discharge (ESD) Sensitivity Classification

JESD22-C101D, Field-induced Charged-Device Model Test method for Electrostatic-Discharge-Withstand Thresholds of Microelectronic Components: Testing results demonstrated that the device’s charged-device model (CDM) sensitivity classification is class II (200V to 500V). No work around required.

2.2 Silicon Revision 1.2 Known Design Exceptions to Functional Specifications

Table 2. Silicon Revision 1.2 Advisory List

Advisory 1.2.1 *The Buffer Logic of VPSS is Not Reset by System Reset Pin*

Revisions Affected 1.2 and earlier

Details At Power-on Reset, the Buffer Logic module, which is a DMA interface logic between the system DMA bus and the VPSS modules, is not getting reset by the system reset pin ($\overline{\text{RESET}}$) logic. As a result, the VPSS DMA transfers through the Buffer Logic will not occur.

Workaround A workaround for this issue is to apply Sync Reset via the Power and Sleep Controller module to the VPSS module followed by a system reset. This software workaround code must be implemented before PLL, DDR, and System Initialization.

The code workaround below flows as follows:

Initially, a Power-on reset or Warm reset is detected using the RSTYPE register of PLLC1, and then a sync reset to the VPSS is applied. After doing this, a system reset is generated using the System Watchdog timer. The next time the RSTYPE is read by the code (after the system reset and code re-runs), its value would then be 4, which indicates that a maximum system reset has occurred. The code will then proceed with the PLL, DDR, and system initialization functions.

In the event that a Watchdog Reset is ever executed in the code by either the Linux ‘reboot’ command or a regular system Watchdog reset occurs the **Bold** code on below workaround example has been added to address this situation.

Sample code for the software workaround is shown below:

```
#define RSTYPE                        *((volatile unsigned int *)0x01C408E4)
#define VPSS_CLK_CTRL                 *((volatile unsigned int *)0x01C40044)
#define SYSTEM_RESET_EN               *((volatile unsigned int *)0x01C21C08)
#define SYSTEM_RESET_TRIGGER          *((volatile unsigned int *)0x01C21C0C)
#define MDSTAT_VPSS                   *((volatile unsigned int *)0x01C418BC)
#define MDCTL_VPSS                    *((volatile unsigned int *)0x01C41ABC)
#define PTCMD                         *((volatile unsigned int *)0x01C41120)
#define PTSTAT                        *((volatile unsigned int *)0x01C41128)
#define TMPBUF                        *(unsigned int *)(0x17ff8)
#define TMPSTATUS                     *(unsigned int *)(0x17ff0)
#define FLAG_PORRST 0x00000001
#define FLAG_WDTRST 0x00000002
#define FLAG_FLGON 0x00000004
#define FLAG_FLGOFF 0x00000010

void VPSSSyncReset(void);
void WDT_RESET()(void);
void WDT_FLAG_ON(void);
void main()
{
  volatile unsigned int s;
  if (RSTYPE&3)                         // Execute on power on reset or warm reset
   {
   VPSS_CLK_CTRL = 0x80;                // VPSS_CLKCMD = 1:1
   VPSSSyncReset();                     // VPSS sync reset
   *TMPBUF = 0;
   *TMPSTATUS |= FLAG_PORRST;
   SYSTEM_RESET_EN = 0x00020000;        // Watchdog timer enable for system reset
   SYSTEM_RESET_TRIGGER = 0x00020002; // Watchdog timer trigger for system reset
   while(1);
   }

  if((*TMPBUF == 0x591b3ed7))
   {
                                        // Execute on Watchdog Reset situation
   *TMPBUF = 0;
   *TMPSTATUS |= FLAG_PORRST;
```

```
   *TMPSTATUS |= FLAG_FLGOFF;
   for (s=0;s<0x100;s++) {}
   VPSS_CLK_CTRL = 0x80;              // VPSS_CLKCMD = 1:1
   VPSSSyncReset();                    // VPSS sync reset
   SYSTEM_RESET_EN = 0x00020000;      // Watchdog timer enable for system reset
   SYSTEM_RESET_TRIGGER = 0x00020002; // Watchdog timer trigger for system reset
   while(1);
   }
// PLL,DDR Initialization and remaining code
WDT_FLAG_ON();
}

void VPSSSyncReset()
{
unsigned int PdNum=0;
MDCTL_VPSS = (((MDCTL_VPSS) & 0xffffffe0 ) | 0x00000001);
PTCMD = (1<<PdNum);
while(! (((PTSTAT>>PdNum) &0x00000001)==0));
while(!((MDSTAT_VPSS & 0x0000001F)==0x1));
}

void WDT_FLAG_ON()
{
SYSTEM->VPSS_CLKCTL &= 0xffffff7f; // VPSS_CLKMD 1:2
*TMPBUF = 0x591b3ed7;
*TMPSTATUS |= FLAG_FLGON;
}
```

Advisory 1.2.2 — *V_{IL} on 3.3-V LVCMOS Input Buffers*

Revisions Affected 1.2 and earlier

Details The input buffers on the device have shown a timing sensitivity to the logic-low input voltage that can cause changes to the AC input timings. Due to this issue, input voltages must be driven below 0.2V on all 3.3V LVCMOS input signals or 3.3V LVCMOS IO signals used as inputs. Some device signals are dedicated 3.3V and some signals can be configured as either 1.8V or 3.3V.

This issue applies to any input signal operated at 3.3V.

LVCMOS inputs operated at 1.8V are not affected.

The DDR2/MDDR memory interface is not affected.

Workaround Although there is no specific workaround, the following recommendations can be used to help prevent this issue:

- Minimize loads as much as possible, especially DC loads that could cause the V_{IL} to rise. **Point-to-point (single-load) connections are unlikely to be affected.**
- Falling edges should transition as rapidly as possible (so the signal passes through the 0.2V point as early as possible). Heavily loaded nodes resulting in degraded fall times may require drivers to provide rapid input edges to the DM369.

Advisory 1.2.3 *HDVICP—H.264 Encode/Decode: Intra 16 x 16 Plane MB Values ≥ 512 are Incorrect*

Revisions Affected 1.2 and earlier

Details

During the calculation of the intra prediction block, pixel values can reach 512 or higher. However, the computation engine has precision only enough to hold values equal to 511 or less.

This issue can occur only when ***all*** the following conditions are met:

- H.264 Encode or Decode
- Intra_16x16_Plane mode MB
- Luma blocks *only*; it does not affect Chroma blocks
- If the prediction block generated using the Intra_16x16_Plane mode has a pixel value of 512 or higher

The Artifacts by this advisory are:

- In Encode, an incorrect (but legal) bitstream will be generated, and a mismatch will happen between the Encoder and Decoder resulting in signficant quality degradation.
- In Decode, the reconstructed macroblock will be incorrect.
- If the following MBs are intra referring to this MB, the error is propagated spatially within the picture.
- If the following pictures refer to this MB, the error is propagated temporally to the following pictures.

This advisory is content dependent. Typically, generating Intra Prediction Values ≥ 512 is rare but can occur; therefore, the suggested workarounds below should be used. A potential exception could be for a closed-system decoder where the user can ensure that the encoder does not use Intra_16x16_Plane mode.

Workarounds

Encode

Turn off the Intra_16x16_plane mode evaluation by HDVICP IPE. Typically, the quality degradation for removing this mode is relatively small.

Decode

During a MB pipeline slot, a CALC is done, a workaround by CPU, and CALC again with corrected intra prediction macroblock (see Figure 2).

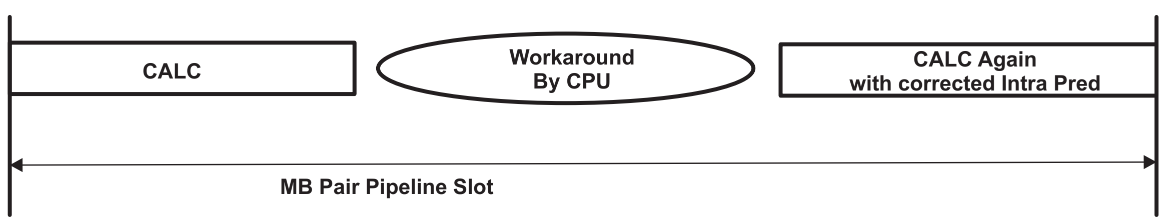

Figure 2. MB Pair Pipeline Slot Example

If the MB is of intra 16x16 plane mode, perform the following steps:

1. Let CALC run.
2. Wait for CALC to complete
3. After CALC completes, switch BFSW for iprdbuf : CALC → DMA
4. Check and correct Intra Prediction (for more details, see the *Check and Correct Intra Prediction* section below)
5. Switch BFSW for iprdbuf : DMA → CALC
6. Switch BFSW for calcmbuf : CALC → DMA
7. Set up CALC commands with intra mode for IQ/IT and inter mode for prediction (for more details, see the *Set up CALC Commands with Intra mode for IQ/IT and Inter Mode for Prediction* section below
8. If MBAFF, retrigger DMA for upper row data
9. Switch BFSW for calcmbuf : DMA → CALC
10. Switch BFSW for reconbuf : LPF → DMA
11. Read left reconstructed column
12. Switch BFSW for reconbuf : DMA → LPF
13. Switch BFSW for calcsbuf : CALC → DMA
14. Set back left reconstructed column
15. Switch BFSW for calcsbuf : DMA → CALC
16. Re-run CALC
17. Wait for CALC to complete

Check and Correct Intra Prediction

Read Pred[0,0], Pred[15,0], Pred[0,15], Pred[15,15], Pred[7,7] and Pred[7,8] (or Pred[8,7]) from iprdbuf

/* This workaround uses the fact that 1D slope is constant on a Plane. */

```
/* This workaround uses the fact that 1D slope is constant on a Plane. */

// right top pixel is wrong
if (Pred[15,0] == 0x00 && Pred[0,15] < Pred[7,8] (or Pred[8,7])) {
    Pred[15,0] = 0xFF;
}
// left bottom pixel is wrong
else if (Pred[0,15] == 0x00 && Pred[15,0] < Pred[7,8] (or Pred[8,7])) {
    Pred[0,15] = 0xFF;
}
// right bottom pixels are wrong
else if (Pred[15,15] == 0x00 && Pred[0,0] < Pred[7,7]) {
    Pred[15,15] = 0xFF
    if (Pred[11,15] == 0x00) Pred[11,15] = 0xFF;
    if (Pred[12,14] == 0x00) Pred[12,14] = 0xFF;
    if (Pred[12,15] == 0x00) Pred[12,15] = 0xFF;
    if (Pred[13,13] == 0x00) Pred[13,13] = 0xFF;
    if (Pred[13,14] == 0x00) Pred[13,14] = 0xFF;
    if (Pred[13,15] == 0x00) Pred[13,15] = 0xFF;
    if (Pred[14,12] == 0x00) Pred[14,12] = 0xFF;
    if (Pred[14,13] == 0x00) Pred[14,13] = 0xFF;
    if (Pred[14,14] == 0x00) Pred[14,14] = 0xFF;
    if (Pred[14,15] == 0x00) Pred[14,15] = 0xFF;
    if (Pred[15,11] == 0x00) Pred[15,11] = 0xFF;
    if (Pred[15,12] == 0x00) Pred[15,12] = 0xFF;
    if (Pred[15,13] == 0x00) Pred[15,13] = 0xFF;
    if (Pred[15,14] == 0x00) Pred[15,14] = 0xFF;
}
```

Set up CALC Commands With Intra Mode for IQ/IT and Inter mode for Prediction

To do the 2nd run, set CALC commands per Table 3 and Table 4. MB mode and Q/IQ information will need to be modified.

Table 3. Format of MB-mode for Spatial Intra Prediction

MB-MODE	
BIT NO.	**DESCRIPTION**
31	TQ Bypass-mode **[0]** :Disable, [1] :Enable.
30:28	Reserved.
27	Chroma DC-Transform (2x2) [0] :Disable, **[1]** : Enable.
26	Luma DC-Transform (4x4) [0] :Disable, **[1]** :Enable.
25:24	Scaling-timing [00] : Do not sue Scaling -function. [01] : Scaling per Luma/Chroma. **[10]** : Scaling per Color-Block. [01] : Scaling per Block.
23:20	Reserved.
19:18	Transform size of Block [5] [00] :8x8 , [01] :8x4 , [10] :4x8 , **[11]** :4x4
17:16	Transform size of Block [4] [00] :8x8 , [01] :8x4 , [10] :4x8 , **[11]** :4x4
15:14	Transform size of Block [3] [00] :8x8 , [01] :8x4 , [10] :4x8 , **[11]** :4x4
13:12	Transform size of Block [2] [00] :8x8 , [01] :8x4 , [10] :4x8 , **[11]** :4x4
11:10	Transform size of Block [11] [00] :8x8 , [01] :8x4 , [10] :4x8 , **[11]** :4x4
9:8	Transform size of Block [0] [00] :8x8 , [01] :8x4 , [10] :4x8 , **[11]** :4x4
7	Reserved.
6:4	Chroma Block-mode [000] :Reserved , **[001]** :Intra 8x8 , [010] :Reserved , [011] :Intra w/o Pred. , [100] :Inter , [101] : Reserved , [110] :Reserved , [111] :Skip.
3	Reserved.
2:0	Luma Block-mode [000] :Intra 16x16 , [001] :Intra 8x8 , [010] :Intra 4x4 , [011] :Intra w/o Pred. , **[100]** :Inter , [101] : Reserved , [110] :Reserved , [111] :Skip.

For Table 4, set Inter shift-scale value = Intra shift-scale value.

Table 4. Q/IQ Information

PARAMETER FOR Q/IQ		
ADDRESS OFFSET	BIT NO.	DESCRIPTION
0x08	63:24	Reserved.
	23:16	**Shift-scale for IQ Intra-DC-Y**
	15:8	**Shift-scale for IQ Inter-DC-Y**
	7:0	Shift-scale for IQ AC-Y
0x10	63:56	Reserved.
	55:48	**Shift-scale for IQ Intra-DC-Cr**
	47:40	**Shift-scale for IQ Inter-DC-Cr**
	39:32	Shift-scale for IQ AC-Cr
	31:24	Reserved.
	23:16	**Shift-scale for IQ Intra-DC-Cb**
	15:8	**Shift-scale for IQ Inter-DC-Cb**
	7:0	Shift-scale for IQ AC-Cb
0x18	63:24	Reserved.
	23:16	**Shift-scale for Q Intra-DC-Y**
	15:8	**Shift-scale for Q Inter-DC-Y**
	7:0	Shift-scale for Q AC-Y
0x20	63:56	Reserved.
	55:48	**Shift-scale for Q Intra-DC-Cr**
	47:40	**Shift-scale for Q Inter-DC-Cr**
	39:32	Shift-scale for Q AC-Cr
	31:24	Reserved.
	23:16	**Shift-scale for Q Intra-DC-Cb**
	15:8	**Shift-scale for Q Inter-DC-Cb**
	7:0	Shift-scale for Q AC-Cb
0x28	63:32	Round coefficient [1]
	31:0	Round coefficient [0]

Advisory 1.2.4 — *Bootloader: RBL Code 4bit ECC Mode Limitation*

Revisions Affected 1.2 and earlier

Details There is an issue with ROM NAND 4bit ECC mode.

The NAND code in the ROM should read 512-byte data chunks from NAND, and the AEMIF ECC hardware has the capability to correct up to 4 bit errors per 512 bytes of data. However, when 5 or more ECC errors are detected for a certain chunk of 512 bytes of data, existing TI software components (UBL, U-Boot and kernel) and the RBL will not read the NANDERRADD1/NANDERRADD2 registers. As a result, the 4BITECC_ADD_CALC_START bit in NANDFCR register is not cleared.

If there are any ECC errors in the ensuing chunks of 512 bytes of data, no ECC corrections will take place. The software workaround for this is that when the NAND driver detects 5 bit errors, the driver will perform a dummy read of the NANDERRVAL1 register or NANDERRADD1 register. This clears the 4BITECC_ADD_CALC_START (bit 13 of NANDFCR register). Doing this ensures ECC error correction takes place even after encountering 5-bit errors for the ensuing chunks of 512 bytes of data. There is no workaround for this issue in the RBL. This is expected to be rectified when TI comes up with new revisions of the device.

TI uses a 2 stage boot loader; the RBL loads a UBL into IRAM which then loads the U-Boot to DDR.

This bug exists in the RBL. This means that if the NAND driver in the RBL encounters 5-bit errors, ECC correction will stop. The UBL, which is the first software component in the bootup sequence, will do a dummy read of the NANDERRADD1 register after setting up the AEMIF as shown below (#1 of FIXES IN TI SOFTWARE COMPONENTS) . If this is not done, UBL will not be able to correct any errors due to the RBL bug. This will occur if the RBL encounters 5-bit errors.

A dummy read of the NANDERRADD1 register will take place even after the NAND driver encounters no bit errors although experiments performed did show that such a dummy read was not required.

The issue is more likely to be seen in NAND devices that require 4-bit ECC.

Even though there might be ECC errors in the UBL image, it is possible to load a UBL with errors in the UBL image. Random boot failures can occur as a result, or if boot appears to succeed, it is also possible to see system stability issues due to possible corrupted system configuration values (e.g., PLL multiplier, PSC domains).

This is a limitation in the RBL. This is not a limitation in the ECC hardware that is a feature of the Asynchronous EMIF (AEMIF) peripheral. Therefore any software outside of the RBL, such as the UBL and NAND driver in the U-Boot and kernel, can use the ECC hardware to implement NAND error correction and detection.

This limitation has no impact on any of the other boot modes: MMC/SD, AEMIF (OneNAND/NOR), USB, SPI, EMAC, UHPI or UART boot modes.

FIXES IN TI SOFTWARE COMPONENTS

1. **Mitigation of RBL BUG in UBL**
   The UBL code is modified to include the sequence of code mentioned below. The following is done in the “device.c” file that is part of the UBL projects for DM369.
   ```
   // AEMIF Setup
   if (status == E_PASS) status |= DEVICE_EMIFInit();

             temp = AEMIF->NANDERRADD1
   ```
   This ensures that 4BITECC_ADD_CALC_START is “low” thereby ensuring that the UBL can perform ECC correction if it encounters bit errors.
2. **Software Fix in NAND writer and UBL**
   The NAND writer and UBL both use the same NAND driver and are built from the

same source. The NAND driver for both the NAND writer and UBL has been modified according to the above explanation. Code excerpt from the NAND driver is shown below. The required sequence has been marked in bold.

```
if ((corrState == 1) || (corrState > 3))
{
  temp = AEMIF->NANDERRADD1;
  return E_FAIL;
}
else if (corrState == 0)
{
  temp = AEMIF->NANDERRADD1;
  return E_PASS;
}
```

3. **Software Fix in U-Boot and kernel**
   TI makes sure that the NAND driver code in both the U-Boot and kernel are exactly the same. The following is an excerpt from the kernel NAND driver that is part of LSP 2.10 and later versions.

```
If (iserror == ECC_STATE_NO_ERR) {
          val = __raw_readl(info->emifregs + NANDERRVAL1);
          return 0;
}
else if (iserror == ECC_STATE_TOO_MANY_ERRS) {
          val = __raw_readl(info->emifregs + NANDERRVAL1);
          printk(KERN_ERR "%s Too many errors to be corrected!\n", __func__);
          return -1;
}
```

In the kernel and U-boot the NANDERRVAL1 register is read, but in the UBL and NAND writer the NANDERRADD1 register is read. Doing a dummy read of any of these registers has exactly the same effect of clearing the 4BITECC_ADD_CALC_START in the NANDFCR register.

NOTE: All TI Software releases for DM369 SOCs (including the PSP 3.01 releases) have been updated with the above software fixes. TI has patches for DM369 SOCs in addition to the official LSP 2.10 and later releases.

Workarounds

Other workarounds to mitigate RBL issue:

- A workaround for this issue would be to use a different ROM boot mode, such as MMC/SD, AEMIF (OneNAND/NOR), USB, SPI, EMAC, UHPI or UART to load a secondary boot loader that can correctly access the NAND and load the remaining system software from the NAND device.
- Another recommendation would be to replace the current 4-bit ECC NAND device with a NAND device that uses less than 4-bit ECC. By employing the 4-bit ECC in the ROM we automatically cover any ECC requirements less than or equal to 4-bits (including 1-bit).

Advisory 1.2.5 *USB Communication Stops During Temperature Swing*

Revisions Affected 1.2 and earlier

Details The USB PHY internal PLL can drift significantly over large transitions in temperature, especially when the Tc temperature is outside the 0°C to 65°C range. The observed failure is that the PHY is no longer able to communicate with the MUSB controller causing all USB data transfer to stop.

Workaround When the issue happens, you can recover from it by resetting the USB PHY. This will force the USB PHY to reinitialize and re-calibrate its PLL.

The USB PHY can be reset as shown below:

```
#define USB_PHY_CTRL    *((volatile unsigned int *)0x01C40034)
#define USBPHY_PHYPDWN  0x1

Void phy_reset(void)
{
   /* Power down the USB PHY */
   USB_PHY_CTRL |= USBPHY_PHYPDWN;

   /* Wait a little */
   mdelay(1);

   /* Power up the USB PHY */
   USB_PHY_CTRL &= ~USBPHY_PHYPDWN;
}
```

For additional information, see the USB PHY Control (USB_PHY_CTRL) Register in the *TMS320DM36x DMSoC ARM Subsystem Reference Guide*.

In order to invoke the recovery mechanism (i.e. resetting the USB PHY) we need to figure out when the issue is present. One way to implement this is to look for the absence of MUSB interrupts over a 2 to 5 seconds interval. This can be done from user space by monitoring this value: [cat /proc/interrupts |grep musb_hdrc **|** awk '{print $2}']. When this counter does not changes for a 2 to 5 seconds period you can assume the condition is present and issue the PHY reset mechanism (described above).

Revision History

DATE	REVISION	NOTES
August 2016	*	Initial External Release

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as "components") are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or "enhanced plastic" are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265