TMS320F28PLC84, TMS320F28PLC83 Power Line Communications (PLC) Processors

Silicon Errata

Literature Number: SPRZ379A
September 2014–Revised June 2016

Contents

List of Figures

List of Tables

TMS320F28PLC8x PLC Processors Silicon Errata

1 Introduction

This document describes the silicon updates to the functional specifications for the TMS320F28PLC8x PLC processors.

The updates are applicable to:

- 80-pin Low-Profile Quad Flatpack, PN Suffix

2 Device and Development Support Tool Nomenclature

To designate the stages in the product development cycle, TI assigns prefixes to the part numbers of all TMS320™ microcontroller (MCU) devices and support tools. Each TMS320 MCU commercial family member has one of three prefixes: TMX, TMP, or TMS (for example, **TMS**320F28PLC83). 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 (with TMX for devices and TMDX for tools) through fully qualified production devices and tools (with TMS for devices and TMDS for tools).

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.

TI device nomenclature also includes a suffix with the device family name. This suffix indicates the package type (for example, PN) and temperature range (for example, T).

3 Device Markings

Figure 1 provides an example of the F28PLC8x device markings and defines each of the markings. The device revision can be determined by the symbols marked on the top of the package as shown in Figure 1. Some prototype devices may have markings different from those illustrated. Figure 2 shows an example of the device nomenclature.

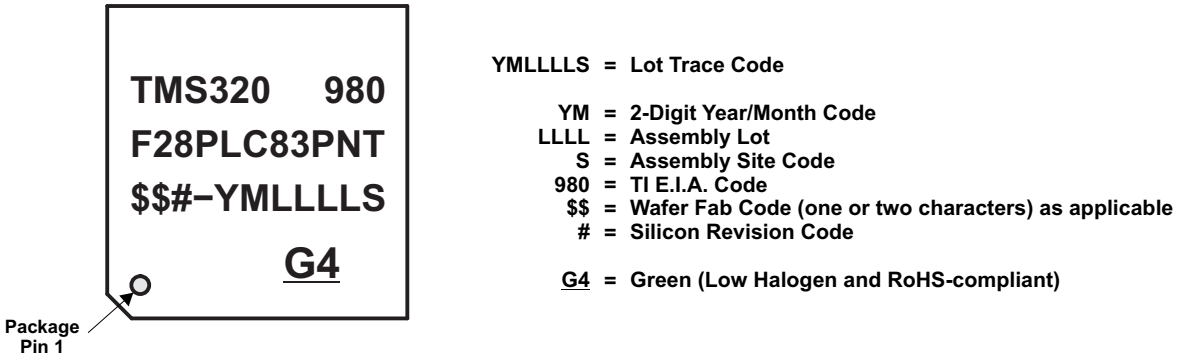

Figure 1. Example of Device Markings

Table 1. Determining Silicon Revision From Lot Trace Code

SILICON REVISION CODE	SILICON REVISION	REVISION ID Address: 0x0883	COMMENTS
A	Indicates Revision A	0x0001	This silicon revision is available as TMS.

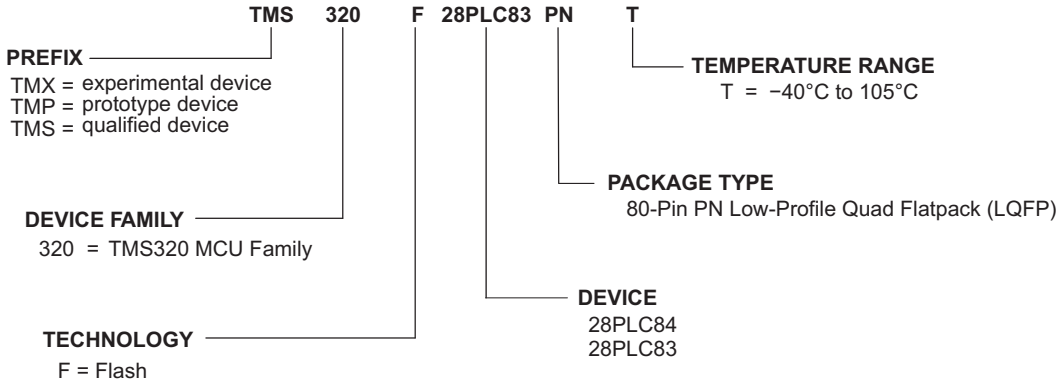

Figure 2. Example of Device Nomenclature

4 Usage Notes and Known Design Exceptions to Functional Specifications

4.1 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 usage 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.

4.1.1 PIE: Spurious Nested Interrupt After Back-to-Back PIEACK Write and Manual CPU Interrupt Mask Clear Usage Note

Revision(s) Affected: A

Certain code sequences used for nested interrupts allow the CPU and PIE to enter an inconsistent state that can trigger an unwanted interrupt. The conditions required to enter this state are:

1. A PIEACK clear is followed immediately by a global interrupt enable (EINT or asm(" CLRC INTM")).
2. A nested interrupt clears one or more PIEIER bits for its group.

Whether the unwanted interrupt is triggered depends on the configuration and timing of the other interrupts in the system. This is expected to be a rare or nonexistent event in most applications. If it happens, the unwanted interrupt will be the first one in the nested interrupt's PIE group, and will be triggered after the nested interrupt re-enables CPU interrupts (EINT or asm(" CLRC INTM")).

Workaround: Add a NOP between the PIEACK write and the CPU interrupt enable. Example code is shown below.

```
//Bad interrupt nesting code
PieCtrlRegs.PIEACK.all = 0xFFFF;      //Enable nesting in the PIE
EINT;                                 //Enable nesting in the CPU

//Good interrupt nesting code
PieCtrlRegs.PIEACK.all = 0xFFFF;      //Enable nesting in the PIE
asm(" NOP");                          //Wait for PIEACK to exit the pipeline
EINT;                                 //Enable nesting in the CPU
```

4.1.2 FPU32 and VCU Back-to-Back Memory Accesses

Revision(s) Affected: A

This usage note applies when a VCU memory access and an FPU memory access occur back-to-back. There are three cases:

Case 1. Back-to-back memory reads: one read performed by a VCU instruction (VMOV32) and one read performed by an FPU32 instruction (MOV32).

If an R1 pipeline phase stall occurs during the first read, then the second read will latch the wrong data. If the first instruction is not stalled during the R1 pipeline phase, then the second read will occur properly.

The order of the instructions—FPU followed by VCU or VCU followed by FPU—does not matter. The address of the memory location accessed by either read does not matter.

Case 1 Workaround: Insert one instruction between the two back-to-back read instructions. Any instruction, except a VCU or FPU memory read, can be used.

Case 1, Example 1:

```
VMOV32  VR1,mem32       ; VCU memory read
NOP                     ; Not a FPU/ VCU memory read
MOV32   R0H,mem32       ; FPU memory read
```

Case 1, Example 2:

```
VMOV32  VR1,mem32       ; VCU memory read
VMOV32  mem32, VR2      ; VCU memory write
MOV32   R0H,mem32       ; FPU memory read
```

Case 2. Back-to-back memory writes: one write performed by a VCU instruction (VMOV32) and one write performed by an FPU instruction (MOV32).

If a pipeline stall occurs during the first write, then the second write can corrupt the data. If the first instruction is not stalled in the write phase, then no corruption will occur.

The order of the instructions—FPU followed by VCU or VCU followed by FPU—does not matter. The address of the memory location accessed by either write does not matter.

Case 2 Workaround: Insert two instructions between the back-to-back VCU and FPU writes. Any instructions, except VCU or FPU memory writes, can be used.

Case 2, Example 1:

```
VMOV32  mem32,VR0        ; VCU memory write
NOP                      ; Not a FPU/VCU memory write
NOP                      ; Not a FPU/VCU memory write
MOV32   mem32,R3H        ; FPU memory write
```

Case 2, Example 2:

```
VMOV32  mem32,VR0        ; VCU memory write
VMOV32  VR1,  mem32      ; VCU memory read
NOP
MOV32   mem32,R3H        ; FPU memory write
```

Case 3. Back-to-back memory writes followed by a read or a memory read followed by a write. In this case, there is no interaction between the two instructions. No action is required.

Workaround: See Case 1 Workaround and Case 2 Workaround.

4.2 Known Design Exceptions to Functional Specifications

Table 2. Table of Contents for Advisories

Advisory — *VCU: First CRC Calculation May Not be Correct*

Revision(s) Affected

A

Details

Due to the internal power-up state of the VCU module, it is possible that the first CRC result will be incorrect. This condition applies to the first result from each of the eight CRC instructions. This ***rare*** condition can only occur after a power-on reset, but will not necessarily occur on every power on. A warm reset will not cause this condition to reappear.

Workaround(s)

The application can reset the internal VCU CRC logic by performing a CRC calculation of a single byte in the initialization routine. This routine only needs to perform one CRC calculation and can use any of the CRC instructions. At the end of this routine, clear the VCU CRC result register to discard the result.

An example is shown below.

```
_VCUcrc_reset
      MOVZ        XAR7, #0
      VCRC8L_1    *XAR7
      VCRCCLR
      LRETR
```

Advisory — *ADC: Initial Conversion*

Revision(s) Affected

A

Details

When the ADC conversions are initiated by any source of trigger in either sequential or simultaneous sampling mode, the first sample may not be the correct conversion result.

Workaround(s)

For sequential mode, discard the first sample at the beginning of every series of conversions. For instance, if the application calls for a given series of conversions, SOC0→SOC1→SOC2, to initiate periodically, then set up the series instead as SOC0→SOC1→SOC2→SOC3 and only use the last three conversions, ADCRESULT1, ADCRESULT2, ADCRESULT3, thereby discarding ADCRESULT0.

For simultaneous sample mode, discard the first sample of both the A and B channels at the beginning of every series of conversions.

User application should validate if this workaround is acceptable in their application.

This issue is fixed completely by writing a 1 to the ADCNONOVERLAP bit in the ADCTRL2 register, which only allows the sampling of ADC channels when the ADC is finished with any pending conversion.

Advisory — *ADC: Temperature Sensor Minimum Sample Window Requirement*

Revision(s) Affected

A

Details

If the minimum sample window is used (6 ADC clocks at 45 MHz, 155.56 ns), the result of a temperature sensor conversion can have a large error, making the result unreliable for the system.

Workaround(s)

1. If double-sampling of the temperature sensor is used to avoid the corrupted first sample issue, the temperature sensor result is valid. Double-sampling of the temperature sensor is equivalent to giving the sample-and-hold (S/H) circuit adequate time to charge.
2. In all other conditions, the sample-and-hold window used to sample the temperature sensor should not be less than 550 ns.

Advisory — *ADC: ADC Result Conversion When Sampling Ends on 14th Cycle of Previous Conversion, ACQPS = 6 or 7*

Revision(s) Affected A

Details The on-chip ADC takes 13 ADC clock cycles to complete a conversion after the sampling phase has ended. The result is then presented to the CPU on the 14th cycle post-sampling and latched on the 15th cycle into the ADC result registers. If the next conversion's sampling phase terminates on this 14th cycle, the results latched by the CPU into the result register are not assured to be valid across all operating conditions.

Workaround(s) Some workarounds are as follows:

- Due to the nature of the sampling and conversion phases of the ADC, there are only two values of ACQPS (which controls the sampling window) that would result in the above condition occurring—ACQPS = 6 or 7. One solution is to avoid using these values in ACQPS.
- When the ADCNONOVERLAP feature (bit 1 in ADCTRL2 register) is used, the above condition will never be met; so the user is free to use any value of ACQPS desired.
- Depending on the frequency of ADC sampling used in the system, the user can determine if their system will hit the above condition if the system requires the use of ACQPS = 6 or 7. For instance, if the converter is continuously converting with ACQPS = 6, the above condition will never be met because the end of the sampling phase will always fall on the 13th cycle of the current conversion in progress.

Advisory — *ADC: Offset Self-Recalibration Requirement*

Revision(s) Affected A

Details The factory offset calibration from Device_cal() may not ensure that the ADC offset remains within specifications under all operating conditions in the customer's system.

Workaround(s)

- To ensure that the offset remains within the data sheet's “single recalibration” specifications, perform the AdcOffsetSelfCal() function after Device_cal() has completed and the ADC has been configured.
- To ensure that the offset remains within the data sheet's “periodic recalibration” specifications, perform the AdcOffsetSelfCal() function periodically with respect to temperature drift.

Advisory — *ADC: ADC Revision Register (ADCREV) Limitation*

Revision(s) Affected A

Details The ADC Revision Register, which is implemented to allow differentiation between ADC revisions and ADC types, will always read "0" for both fields.

Workaround(s) On devices with CLASSID (at address 0x882) of 0x009F, 0x008F, 0x007F, or 0x006F, the “TYPE” field in the ADCREV register should be assumed to be 3 and the “REV” field” should be inferred from the table below.

REVID (0x883)	ADCREV.REV Field
0	2
1	2

Advisory — *Memory: Prefetching Beyond Valid Memory*

Revision(s) Affected A

Details The C28x CPU prefetches instructions beyond those currently active in its pipeline. If the prefetch occurs past the end of valid memory, then the CPU may receive an invalid opcode.

Workaround The prefetch queue is 8 x16 words in depth. Therefore, code should not come within 8 words of the end of valid memory. This restriction applies to all memory regions and all memory types (flash, OTP, SARAM) on the device. Prefetching across the boundary between two valid memory blocks is all right.

Example 1: M1 ends at address 0x7FF and is not followed by another memory block. Code in M1 should be stored no farther than address 0x7F7. Addresses 0x7F8-0x7FF should not be used for code.

Example 2: M0 ends at address 0x3FF and valid memory (M1) follows M0. Code in M0 can be stored up to and including address 0x3FF. Code can also cross into M1 up to and including address 0x7F7.

Advisory — *GPIO: GPIO Qualification*

Revision(s) Affected A

Details If a GPIO pin is configured for "n" SYSCLKOUT cycle qualification period (where $1 \le n \le 510$) with "m" qualification samples (m = 3 or 6), it is possible that an input pulse of [n * m – (n – 1)] width may get qualified (instead of n * m). The occurrence of this incorrect behavior depends upon the alignment of the asynchronous GPIO input signal with respect to the phase of the internal prescaled clock, and hence, is not deterministic. The probability of this kind of wrong qualification occurring is "1/n".

Worst-case example:

If n = 510, m = 6, a GPIO input width of (n * m) = 3060 SYSCLKOUT cycles is required to pass qualification. However, because of the issue described in this advisory, the minimum GPIO input width which may get qualified is [n * m – (n – 1)] = 3060 – 509 = 2551 SYSCLKOUT cycles.

Workaround(s) None. Ensure a sufficient margin is in the design for input qualification.

Advisory — *Watchdog: Incorrect Operation of CPU Watchdog When WDCLK Source is OSCCLKSRC2*

Revision(s) Affected A

Details When OSCCLKSRC2 is used as the clock source for CPU watchdog, the watchdog may fail to generate a device reset intermittently.

Workaround(s) WDCLK should be sourced only from OSCCLKSRC1 (INTOSC1). The CPU may be sourced from OSCCLKSRC2 or OSCCLKSRC1 (INTOSC1).

Advisory — *Oscillator: CPU Clock Switching to INTOSC2 May Result in Missing Clock Condition After Reset*

Revision(s) Affected

A

Details

After at least two system resets (not including power-on reset), when the application code attempts to switch the CPU clock source to internal oscillator 2, a missing clock condition will occur, and the clock switching will fail under the following conditions:

- X1 and X2 are unused (X1 is always tied low when unused).
- GPIO38 (muxed with TCK and XCLKIN) is used as JTAG TCK pin only.
- JTAG emulator is disconnected.

The missing clock condition will recover only after a power-on reset when the failure condition occurs.

Workaround(s)

Before switching the CPU clock source to INTOSC2 via the OSCCLKSRCSEL and OSCCLKSRC2SEL bits in the CLKCTL register, the user must toggle the XCLKINOFF and XTALOSCOFF bits in the CLKCTL register as illustrated in the below sequence:

```
CLKCTL |= 0x6000;     // XCLKINOFF = 1, XTALOSCOFF = 1
CLKCTL &=~0x6000;     // XCLKINOFF = 0, XTALOSCOFF = 0
CLKCTL |= 0x6000;     // XCLKINOFF = 1, XTALOSCOFF = 1
CLKCTL &=~0x6000;     // XCLKINOFF = 0, XTALOSCOFF = 0
CLKCTL |= 0x6000;     // XCLKINOFF = 1, XTALOSCOFF = 1
```

Once the above procedure is executed, then the OSC2 selection switches can be configured.

If the JTAG emulator is connected, and GPIO38 (TCK) is toggling, then the above procedure is unnecessary, but will do no harm.

If no clock is applied to GPIO38, TI recommends that a strong pullup resistor on GPIO38 be added to V_{DDIO}.

Advisory — *DMA: ePWM Interrupt Trigger Source Selection via PERINTSEL is Incorrect*

Revision(s) Affected A

Details The MODE.CHx[PERINTSEL] field bit values of 18–29 should select ePWM1SOCA–ePWM6SOCB as DMA trigger sources. Instead, PERINTSEL values of 18–29 select ePWM2SOCA–ePWM7SOCB as DMA trigger sources as shown below in Table 3. ePWM1SOCA and ePWM1SOCB are not implemented as PERINTSEL trigger sources.

Workaround(s) None

Table 3. PERINTSEL Field of Mode Register (MODE)

Bit	Field	Value	Description		
4-0	PERINTSEL		Peripheral Interrupt Source Select Bits: These bits select which interrupt triggers a DMA burst for the given channel. Only one interrupt source can be selected. A DMA burst can also be forced via the PERINTFRC bit.		
		Value	**Interrupt**	**Sync**	**Peripheral**
		0	None	None	No peripheral connection
		1	ADCINT1	None	ADC
		2	ADCINT2	None	
		3	XINT1	None	External Interrupts
		4	XINT2	None	
		5	XINT3	None	
		6	Reserved	None	No peripheral connection
		7	Reserved	None	No peripheral connection
		8	Reserved	None	
		9	Reserved	None	
		10	Reserved	None	
		11	TINT0	None	CPU Timers
		12	TINT1	None	
		13	TINT2	None	
		14	MXEVTA	MXSYNCA	McBSP-A
		15	MREVTA	MRSYNCA	
		16	Reserved	None	No peripheral connection
		17	Reserved	None	
		18	ePWM2SOCA	None	ePWM2
		19	ePWM2SOCB	None	
		20	ePWM3SOCA	None	ePWM3
		21	ePWM3SOCB	None	
		22	ePWM4SOCA	None	ePWM4
		23	ePWM4SOCB	None	
		24	ePWM5SOCA	None	ePWM5
		25	ePWM5SOCB	None	
		26	ePWM6SOCA	None	ePWM6
		27	ePWM6SOCB	None	
		28	ePWM7SOCA	None	ePWM7
		29	ePWM7SOCB	None	
		30	Reserved	None	No peripheral connection
		31	Reserved	None	

Advisory — *ePWM: SWFSYNC Does Not Properly Propagate to Subsequent ePWM Modules or Output on EPWMSYNCO Pin*

Revision(s) Affected

A

Details

When generating a software synchronization pulse using the TBCTL[SWFSYNC] bit, the sync signal does not propagate down to subsequent ePWM modules' EPWMSYNCI inputs. In the case of ePWM1, the software sync also does not generate an output on the EPWMSYNCO pin. Propagation of the synchronization signal between ePWM modules and to EPWMSYNCO operates as expected when generating a synchronization pulse via an external signal on the EPWMSYNCI pin.

Workaround(s)

If the application needs to generate a software sync:

1. The application code should configure a single GPIO mux setting for EPWMSYNCI operation.
2. The EPWMSYNCI pin should be tied to ground or consistently driven low.
3. After the above, the application can use the TBCTL[SWFSYNC] bit as intended to generate a software synchronization pulse that passes to subsequent ePWM modules and onto the EPWMSYNCO pin.

5 Documentation Support

For device-specific data sheets and related documentation, visit the TI web site at: http://www.ti.com.

For more information about the TMS320F28PLC8x device, see the *TMS320F28PLC8x Power Line Communications (PLC) Processors Data Manual*.

Revision History

Changes from September 15, 2014 to June 10, 2016 (from SPRZ379 Revision (September 2014) to SPRZ379A Revision) **Page**

- Figure 1 (Example of Device Markings): Updated figure. 5
- Section 4.1.2 (FPU32 and VCU Back-to-Back Memory Accesses): Added usage note. 7

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