TMS320F28044 Digital Signal Processor Silicon Revision 0

1 Introduction

This document describes the silicon updates to the functional specifications for the TMS320F28044 digital signal processor (DSP).

The updates are applicable to:

- 100-ball MicroStar BGA™, GGM, and ZGM suffix
- 100-pin low-profile quad flatpack, PZ suffix

2 Device and Development Tool Support Nomenclature

To designate the stages in the product development cycle, TI assigns prefixes to the part numbers of all TMS320™ DSP devices and support tools. Each TMS320 DSP commercial family member has one of three prefixes: TMX, TMP, or TMS (for example, **TMS**320F28044). 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).

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, GGM) and temperature range (for example, A).

3 Device Markings

Figure 1 provides an example of the TMS320F28044 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 the TMS320F28044 device nomenclature.

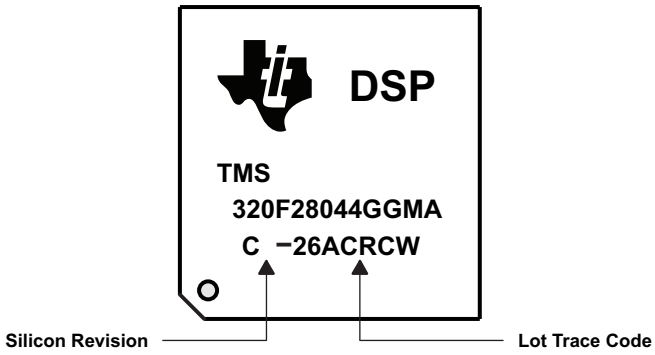


Figure 1. Example of Device Markings

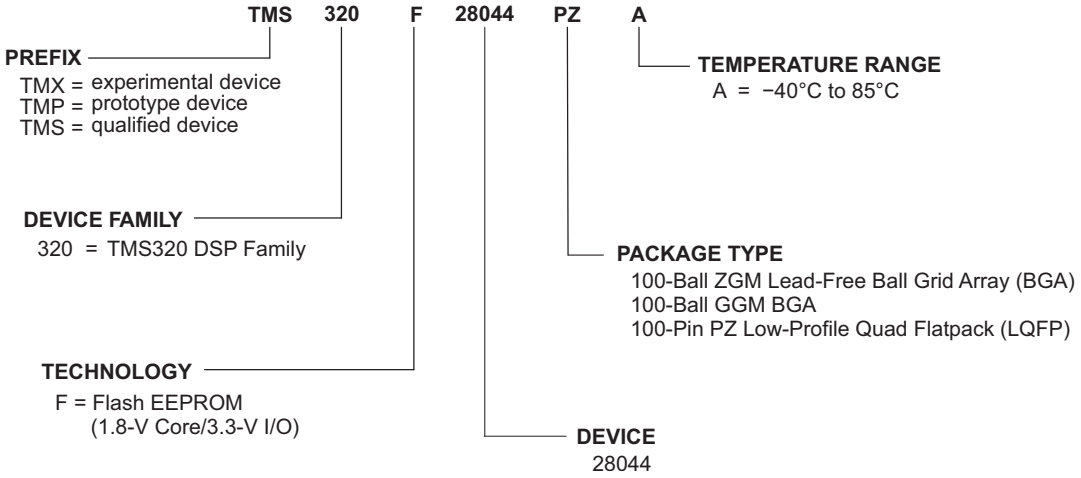


Figure 2. Device Nomenclature

Table 1. Determining Silicon Revision From Lot Trace Code (F28044)

SECOND LETTER IN PREFIX OF LOT TRACE CODE	SILICON REVISION	REVISION ID Address: 0x0883	F28044 COMMENTS
Blank (no second letter in prefix)	Indicates Revision 0	0x0000	This silicon revision is available as TMX or TMS

4 Silicon Change Overview

Table 2 lists the change(s) made to each silicon revision.

Table 2. TMS320F28044 Silicon Change Overview

REVISION	CHANGES MADE
0	First silicon release

5 Usage Notes and Known Design Exceptions to Functional Specifications

5.1 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.

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

Revision(s) Affected: 0

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
```

5.2 Known Design Exceptions to Functional Specifications

Table 3. Table of Contents for Advisories

Advisory — *Input Clock: Device Startup Using XCLKIN Input*

Revision(s) Affected 0

Details When clock to the device is supplied using the XCLKIN pin, device may intermittently fail to startup correctly.

Workaround(s) Do not use the XCLKIN pin to supply clock to the device. Instead, use either a crystal/resonator or a 1.8-V external oscillator on the X1 pin to clock the device. This will be fixed in the next revision of the silicon.

Advisory — *Memory: Prefetching Beyond Valid Memory*

Revision(s) Affected

0

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 8x16 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 ok.

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 it. 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 — *Memory: Flash and OTP Prefetch Buffer Overflow*

Revision(s) Affected

0

Details

This advisory applies to code executing from flash or OTP with the flash prefetch buffer enabled. On ROM devices this applies to the ROM that replaces flash and OTP.

The flash prefetch buffer may overflow if a SBF or BF instruction is within eight 16-bit words preceding an operation using indirect or direct program-memory addressing. The window for which this can occur is shown below:

```
Address
Offset

0x0000    BF LSW (32-bit opcode)
0x0001    BF MSW or SBF (16-bit opcode)
-------------------------------------
0x0002    SBF/BF + 1 word     //
0x0003    SBF/BF + 2 words    //
0x0004    SBF/BF + 3 words    // If an instruction within this window
0x0005    SBF/BF + 4 words    // uses program-memory addressing it
0x0006    SBF/BF + 5 words    // can cause the flash prefetch buffer to
0x0007    SBF/BF + 6 words    // overflow.
0x0008    SBF/BF + 7 words    //
0x0009    SBF/BF + 8 words    //
-------------------------------------
0x0010    SBF/BF + 9 words
```

Whether or not an overflow actually occurs depends on the instruction sequence, flash wait states and CPU pipeline stalls. If an overflow occurs it will result in execution of invalid opcodes. Instructions that use program-memory addressing include MAC/XMAC, DMAC/XMACD, QMACL, IMACL, PREAD/XPREAD and PWRITE/XPWRITE.

Workaround(s)

1. Hand-coded assembly:

Use the SB/B instructions instead of SBF/BF for code targeted to execute from flash or OTP. The SB/B instructions are more efficient in wait-stated memory so a performance improvement may also be seen.

2. Compiler-generated assembly:

Use the compiler switch -me to force the compiler to generate SB/B instructions instead of SBF/BF instructions. In heavily wait stated memory the SB/B instructions are more efficient than SBF/BF. In SARAM the SBF/BF instructions are more efficient. Therefore, this switch should be applied as follows:

- Use the compiler switch -me on source code that runs from flash or OTP.
- Do not use the compiler switch -me on source code that runs from SARAM.
- Use -me If a file contains functions that runs from flash as well as functions that run from SARAM.

The -me switch is available in C28x compiler as of V4.1.4, V5.0.0 and above.

Advisory — *ADC: Simultaneous Sampling Latency*

Revision(s) Affected

0

Details

When the ADC conversions are initiated in simultaneous mode, the first sample pair will not give correct conversion results.

Workaround(s)

1. If the ADC is used with a sampling window ≤ 160 nS, then the first sample pair must be discarded and a second sample of the same pair must be taken. For instance, if the sequencer is set to sample channel A0:B0/A1:B1/A2:B2 in that order, then load the sequencer with A0:B0/A0:B0/A1:B1/A2:B2 and only use the last three conversions.

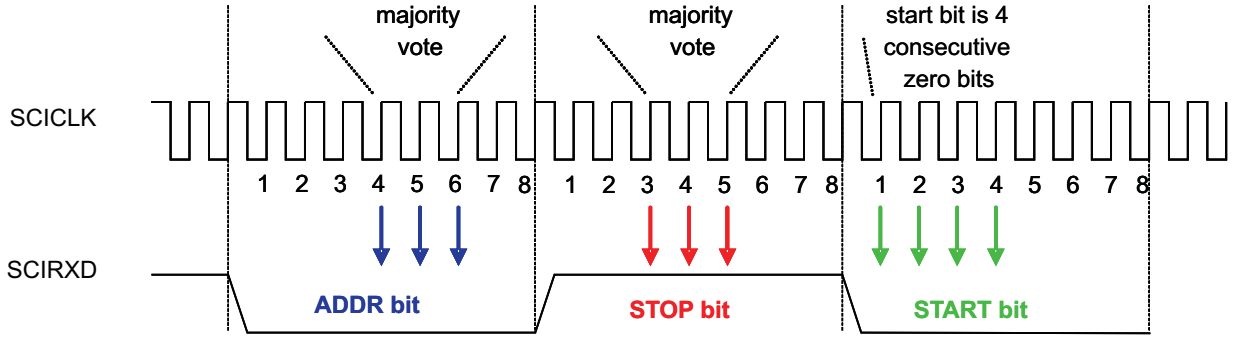
2. If the ADC is used with a sampling window greater than 160 ns, there is no issue.

Advisory — *SCI: Incorrect Operation of SCI in Address Bit Mode*

Revision(s) Affected 0

Details SCI does not look for STOP bit after the ADDR bit. Instead, SCI starts looking for the start bit beginning on sub-sample 6 of the ADDR bit. Slow rise-time from ADDR to STOP bit can cause the false START bit to occur since the 4th sub-sample for the start bit may be sensed low.

Expected Operation:



Erroneous Operation:

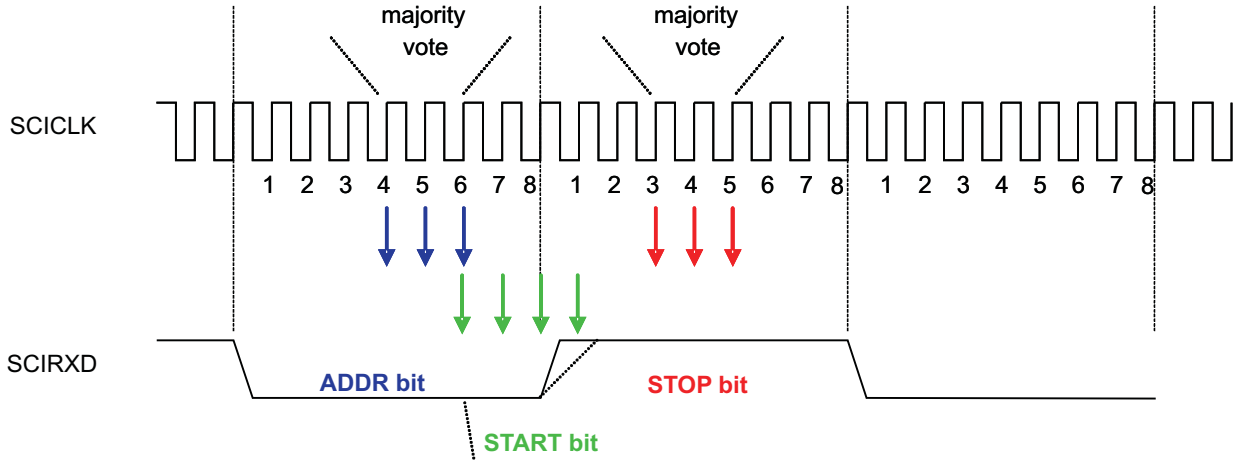


Figure 3. Difference Between Expected and Erroneous Operation of START Bit

Workaround(s) Program the baud rate of the SCI to be slightly slower than the actual. This will cause the 4th sub-sample of the false START bit to be delayed in time, and therefore occur more towards the middle of the STOP bit (away from the signal transition region). The amount of baud slowing needed depends on the rise-time of the signal in the system. Alternatively, IDLE mode of the SCI module may be used, if applicable.

Advisory — *SCI: Bootloader Does Not Clear the ABD Bit After Auto-Baud Lock*

Revision(s) Affected

0

Details

The SCI ROM bootloader code does not clear the Auto-Baud Detect (ABD) bit in the SCIFFCT register after the auto-baud process completes. If the SCI-A port is used after the bootloader is executed, transmit interrupts (SCITXINTA) will not be able to occur, nor will the auto-baud lock feature of SCI-A work correctly.

Workaround

If the SCI bootloader has been executed, the user's application code should clear the ABD bit by writing a 1 to ABD CLR (bit 14) in the SCIFFCT register before enabling the SCITXINTA interrupt, and before using the auto-baud feature.

Advisory

WD: Change to Watchdog Module: Bad Key Writes to WDKEY No Longer Cause RESET/Interrupt to be Generated

Revision(s) Affected

0

Details

The “Bad Key Detect” function of the WDKEY register has been disabled. When using the Watchdog (WD) module, a write of anything other than 0x55 or 0xAA to the WDKEY register will have no effect. See the *TMS320x280x, 2801x, 2804x DSP System Control and Interrupts Reference Guide* for more information.

Workaround(s)

To trigger an immediate reset or interrupt, perform an invalid write to the WDCHK bits in the WDCR register.

Advisory — *GPIO: GPIO Qualification*

Revision(s) Affected

0

Details

If a GPIO pin is configured for "n" SYSCLKOUT cycle qualification period (where 1 ≤ n ≤ 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). This 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 — *Pulldown Resistor for $\overline{\text{TRST}}$ Pin*

Revision(s) Affected 0

Details In the device data sheet, the recommendation of an external pulldown resistor has now been made a requirement. Earlier, the data sheet suggested leaving this pin unconnected in low-noise environments. Since the term "low-noise" is not easily quantified, an external pulldown resistor has been made a requirement for more robust operation.

Workaround(s) An external pulldown resistor is required on the $\overline{\text{TRST}}$ pin.

6 Documentation Support

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

For further information regarding the TMS320F28044 DSP, please see the *TMS320F28044 Digital Signal Processor Data Manual*.

Trademarks

MicroStar BGA, TMS320 are trademarks of Texas Instruments.
All other trademarks are the property of their respective owners.

Revision History

Changes from April 7, 2011 to June 4, 2020 (from E Revision (April 2011) to F Revision)	Page
• Figure 2 (Device Nomenclature): Updated figure.	2
• Section 5.1 (Usage Notes): Added section.	4
• Section 5.2 (Known Design Exceptions to Functional Specifications): Removed "eQEP: Missed First Index Event" advisory.	5

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

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