TMS320VC5441

Digital Signal Processor

Silicon Errata

SPRZ190C
June 2002 − Revised April 2006

Texas Instruments

Contents

1 Introduction

This document describes the silicon updates to the functional specifications for the TMS320VC5441 silicon. The advisories are applicable to:

- TMS320VC5441 (176-pin LQFP, PGF suffix)
- TMS320VC5441 (169-pin MicroStar BGA™, GGU suffix)

1.1 Quality and Reliability Conditions

TMX Definition

Texas Instruments (TI) does not warranty either (1) electrical performance to specification, or (2) product reliability for products classified as “TMX.” By definition, the product has not completed data sheet verification or reliability performance qualification according to TI Quality Systems Specifications.

The mere fact that a “TMX” device was tested over a particular temperature and voltage ranges should not, in any way, be construed as a warranty of performance.

TMP Definition

TI does not warranty product reliability for products classified as “TMP.” By definition, the product has not completed reliability performance qualification according to TI Quality Systems Specifications; however, products are tested to a published electrical and mechanical specification.

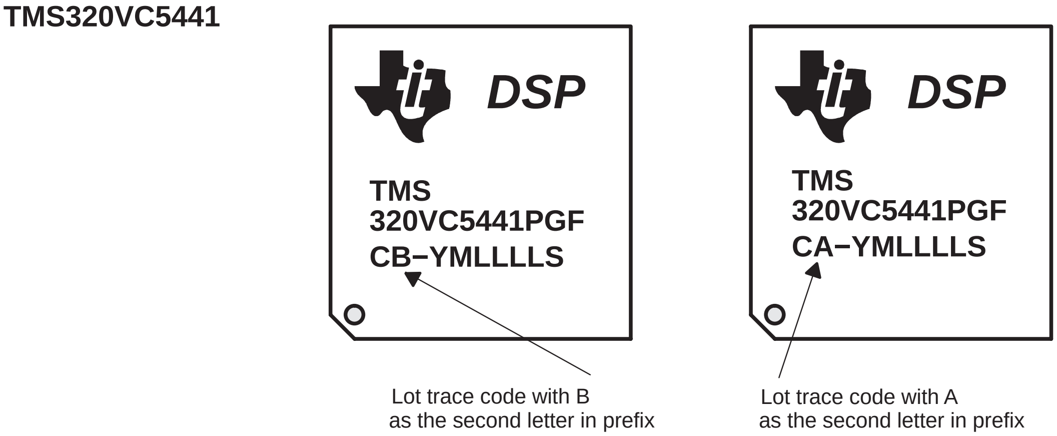
TMS Definition

Fully-qualified production device.

1.2 Revision Identification

The device revision can be determined by the lot trace code marked on the top of the package. The locations for the lot trace codes for the PGF and the GGU packages are shown in Figure 1 and Figure 2, respectively. The location of other markings may vary per device.

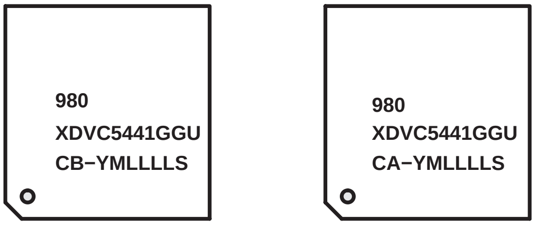
Figure 1. Example, Typical Lot Trace Code for TMS320VC5441 (PGF)


TMS320VC5441
DSP
TMS
320VC5441PGF
CB−YMLLLLS
DSP
TMS
320VC5441PGF
CA−YMLLLLS
Lot trace code with B as the second letter in prefix
Lot trace code with A as the second letter in prefix

Lot Trace Code	Silicon Revision
Blank (no second letter in prefix)	Initial Silicon
A (second letter in prefix is A)	Silicon Revision A
B (second letter in prefix is B)	Silicon Revision B
C (second letter in prefix is C)	Silicon Revision C
D (second letter in prefix is D)	Silicon Revision D

NOTE: Qualified devices in the PGF package are marked with the letters “TMS” at the beginning of the device name, while nonqualified devices in the PGF package are marked with the letters “TMX” or “TMP” at the beginning of the device name

Figure 2. Example, Typical Lot Trace Code for TMS320VC5441 (GGU)



NOTE: Qualified devices in the GGU package are marked with the letters “DV” at the beginning of the device name, and nonqualified devices in the GGU package are marked with the letters “XDV” or “PDV” at the beginning of the device name.

2 Known Design Marginality/Exceptions to Functional Specifications

Table 1. Summary of Advisories

Description	Revision Affected	Page
Single-Stepping Through Code	Initial Silicon and Revision A	5
DMPREC	All Revisions	6
Watchdog Timer Out Pulse	All Revisions	7
Memory Array Clock Circuitry Not Initialized After Power-On Reset	All Revisions	7
133-MHz Performance	All Revisions	9
HPI HINT	All revisions	9
HPI Write Increment Behavior	All revisions	9
Far Branches/Calls/Interrupts from Active Repeat Blocks (BRAF)	All revisions	10
Round (RND) Instruction Clears Pending Interrupts	All revisions	11
TOUT Jitter	All revisions	11

Advisory *Single-Stepping Through Code*

Revision(s) Affected: Initial Silicon and Revision A

Details: While single-stepping through the code in emulation mode, the code fails if it writes data to the same memory block under some *special* conditions. There are several things that can affect the pass/fail and they only happen in SINGLE STEP mode. The code passes in RUN mode.

In the following test case, the code fails on the emulator when single-stepping through the code (addressing mode is SP-relative).

```
mvkd 13h,*ar4
nop
nop
stl a,1h
stl b,4h
add #8,a (or mvkd 13h, 2h)
```

The fail signature is that both SP(1) and SP(4) are loaded with the contents of AREG instead of having the contents of AREG loaded in SP(1) and the contents of BREG in SP(4).

The code passes in RUN mode.

Workaround: The following workarounds are applicable.

1. While debugging a code block similar to the given test case, if the results of single step do not make sense, try setting the breakpoint out of this block and then running the code.
2. Add a **“nop”** between the last **“stl”** and **“add”** (or **“mvkd”**):

This problem is corrected in Revision B of the silicon.

Advisory *DMPREC*

Revision(s) Affected: All Revisions

Details: When updating the DE bits of the DMPREC register while one or more DMA channel transfers are in progress, it is possible for the write to the DMPREC to cause an additional transfer on one of the active channels.

The problem occurs when an active channel completes a transfer at the same time that the user updates the DMPREC register. When the transfer completes, the DMA logic attempts to clear the DE bit corresponding to the complete channel transfer, but the register is instead updated with the CPU write (usually an ORM instruction) which can set the bit and cause an additional transfer on the channel. Refer to the example below for further clarification:

Example:

DMPREC value = 00C1h, corresponding to the following channel activity:

Channel 0 – enabled and running	(DE0 = 1)
Channel 1 – disabled	(DE1 = 0)
Channel 2 – disabled	(DE2 = 0)
Channel 3 – disabled	(DE3 = 0)
Channel 4 – disabled	(DE4 = 0)
Channel 5 – disabled	(DE5 = 0)

If the following conditions occur simultaneously, the user code will reenable channel 0 (DMPREC value written = 00C3h), and an additional, unintended transfer will begin on channel 0.

- Channel 0 transfer completes and DMA logic clears DE0 internally.
- User code attempts to enable another channel (e.g., ORM #2, DMPREC)

Workaround: There are a few user conditions under which this problem does not occur. If all active DMA channels are configured in ABU mode or in autoinitialization mode, then the problem does not occur because the channels remain enabled until they are disabled by user code. The problem is also avoided in applications that use only one DMA channel at a time.

Systems that use multiple DMA channels simultaneously in multiframe mode without autoinitialization are most likely to have this problem. In such systems, one of the following methods can be used to avoid the problem:

- Always wait for all channels to complete existing transfers before reenabling any channels, and always enable all channels at the same time.
- Before enabling a channel, check the progress of any on-going transfers by reading the element and frame counts of each active channel. If any active channel is within two element transfers of completing a block transfer, then wait until the active channel completes the block transfer before writing to the DMPREC register. Otherwise, if all active channels have more than two element transfers left in a block transfer, it is safe to update the DMPREC register.

Advisory *Watchdog Timer Out Pulse*

Revision(s) Affected: All revisions

Details: When the watchdog timer reaches time out, the output pin has no active-low pulse. When the watchdog timer is enabled, the output is driven high. After the watchdog timer has timed out and automatically disabled, the output pin goes into GPIO mode and is a general-purpose input. As a result, the is no active-low pulse. The internal functionality of the watchdog timer is not affected.

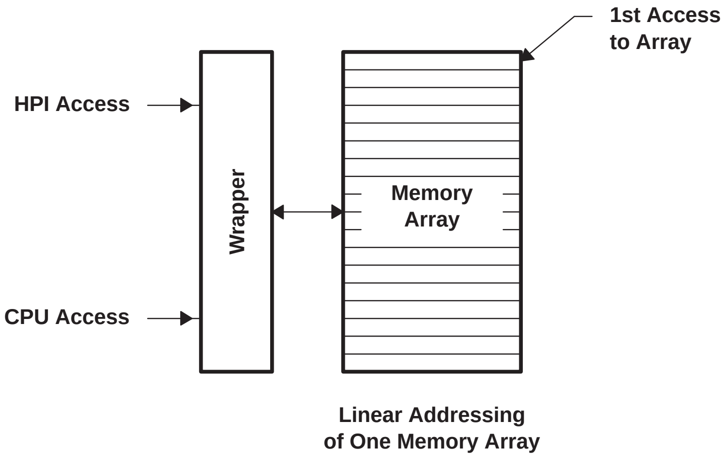
Workaround: None

Advisory *Memory Array Clock Circuitry Not Initialized After Power-On Reset*

Revision(s) Affected: All revisions

Details: During testing of the 5441 devices, an issue was found on the 5441 digital signal processor (DSP) that affected the initial access to memory blocks. This includes accesses performed by the DSP CPU or by a host using the HPI peripheral. The **first** access to any local memory block, regardless of the address within the block, may be erroneous (i.e., erroneous data is either read from or written to only the first address location accessed). Depending on the application, the first memory access could be the first logical address in the memory array, the last, or anywhere in between.

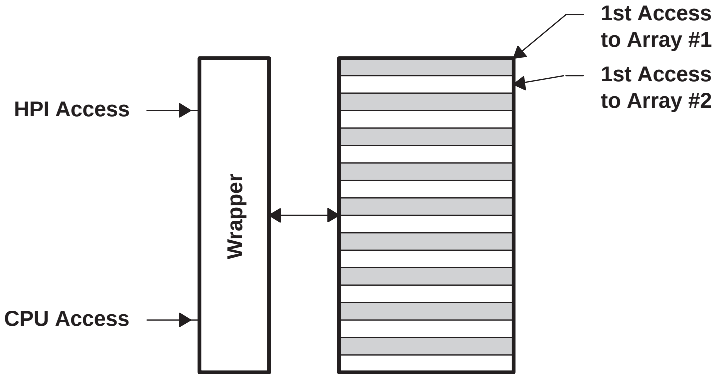
Figure 3. Linear Addressing



The problem is related to the memory array clock circuitry after a Power-On Reset (POR). Correct operation is to turn on the clock circuitry so that the memory arrays are initialized and then turn off the memory clocks for power savings. However, the clock circuitry does not always initialize fully before the clock output is turned off. As a result, the memory arrays are not initialized after the device is powered. An initial access to the memory array turns on the clock circuitry completely and the invalid memory array state is cleared.

Memory Array Clock Circuitry Not Initialized After Power-On Reset (Continued)

Figure 4. Interleaved Addressing



Interleaved Addressing Between 2 Memory Arrays

The memory access failure also exists in shared memory. However, the memory architecture differs slightly for shared memory and may appear to fail in more than one address location. The multiple failures occur because the logical memory blocks consist of two physical memory arrays that are interleaved — even address locations in one array and odd address locations in the second array. In shared memory, the failure may appear to be in either or both the first access to the even address array and the first access to the odd address array.

In either case — local or shared memory — the memory access failure may or may not appear on every memory block. Due to its nature, the failure can be affected by voltage level and temperature. However, if systems are designed within the specified power supply voltage ranges, temperature limits are not exceeded, and the workaround is implemented, then there will not be subsequent failures.

Workaround: In order to be certain that no failure occur during normal operation, all local memory arrays should be “dummy” accessed at least once after POR. For shared memory, at least two consecutive and contiguous “dummy” accesses must be performed. The memory array block configuration (from an HPI perspective) is shown below:

Local Memory (Program/Data)	**Shared Memory (Program)**
0x0060 – 0x3FFF	0x8000 – 0xFFFF *(see Note)*
0x4000 – 0x7FFF	
0x8000 – 0xBFFF (DMMR=0/1)	
0xC000 – 0xFFFF (DMMR=0/1)	

Note: Depending on the configuration of the DSP, the memory range 0x0000−0xFFFF is located in either Page 0,1,2 or 3. Please refer to Figure 2−8 through Figure 2−11 in the *TMS320VC5441 Fixed-Point Digital Signal Processor Data Manual* (literature number SPRS122).

Advisory *133-MHz Performance*

Revision(s) Affected: All revisions

Details: At the rated core supply voltage of 1.42 V − 1.5V, CPU can not function reliably at the maximum speed of 133 MHz across the full commercial case temperature range of 0°C to 85°C.

Workaround: Increase the core supply voltage to 1.6 V at the internal CPU clock speed to 133 MHz for functionality across the full commercial case temperature range of 0°C to 85°C.

The 5441 devices will be tested and characterized at the rated core supply voltage of 1.55 V to 1.65 V and at the maximum speed of 133 MHz across the full commercial case temperature range of 0°C to 85°C.

Advisory *HPI HINT*

Revision(s) Affected: All revisions

Details: The HPI will become locked up, with HRDY stuck low, if both the host processor and the TMS320VC5441 CPU write a one (1) to HINT as the same time.

Workaround: Avoid performing redundant operations to the HINT bit. Both the Host and the CPU should check to see if HINT is set before trying to write a one (1) to this bit.

For:	**If...**	**Then...**
the HOST	HINT is ***not*** set...	Do not try to clear HINT by writing a one (1) to it, because the CPU may try to set it.
the CPU	HINT is ***already*** set...	Do not try to set HINT again by writing a one (1) to it, since the HOST may try to clear it.

Advisory *HPI Write Increment Behavior*

Revision(s) Affected: All revisions

Details: For HPID read access with increment, the HPIA is incremented immediately at the end of the access when HSTRB goes high.

For HPID write access with increment, an internal write_increment signal is set at the end of the access (when HSTRB goes high) which will cause the HPIA to be incremented for subsequent HPID writes (either increment or non−increment versions). The write_increment signal will be cleared by any other type of access (HPIA write or read, HPIC write, HPID read) except for HPIC read. The HPIC read has no effect in order to allow for reading status without impacting current mode of operation.

Workaround: None

Advisory — *Far Branches/Calls/Interrupts from Active Repeat Blocks (BRAF)*

Revision(s) Affected: All revisions

Details: When a block repeat is interrupted by a far call, far branch, or interrupt to another page; and a program memory address in the called routine happens to have the same lower 16 bits as the block-repeat end address (REA), a branch to the 16-bit block-repeat start address (RSA) is executed on the current page until the block-repeat counter decrements to 0. The XPC is ignored during these occurrences.

Workaround: Use one of the following workarounds:

1. If the called routine must be on a different page and has a program memory address that has the same lower 16 bits as the REA, save ST1 and clear the BRAF in the vector table before entering the called routine with the following two instructions:

   ```
   PSHM ST1
   RSBX BRAF
   ```

   Then, restore ST1 before returning from the called routine. In the case of an interrupt service routine, these two instructions can be included in the delay slots following a delayed-branch instruction (BD) at the interrupt vector location. Then, the ST1 is restored before returning from the routine. With this method, BRAF is always inactive while in the called routine. If BRAF was not active at the time of the call, the RSBX BRAF has no effect.
2. Put the called routine on the same page as the interruptible block-repeat code. This can be achieved automatically by placing the interrupt vector table and the interrupt service routines or other called routines on the overlay pages. If this approach is used, far branches/calls are not necessary and the situation is completely avoided.
3. Avoid putting the called routine on other pages where a program memory address has the same lower 16 bits as the REA.
4. Use the BANZ instruction as a substitute for the block repeat.

Advisory — *Round (RND) Instruction Clears Pending Interrupts*

Revision(s) Affected: All revisions

Details: The RND (round) instruction opcode is decoded incorrectly and will write to the interrupt flag register (IFR) with the data from the data write bus (E bus). Therefore, it could cause the pending interrupt to be missed.

Workaround: Do not use the RND instruction. Replace the RND instruction with an ADD instruction as follows:

For this instruction ...	Use ...
`RND src[,dst]`	`ADD #1,15,src[,dst]`

Advisory — *TOUT Jitter*

Revision(s) Affected: All revisions

Details: When the CLKOUT divide factor is set to a non-zero value, the timer output exhibits jitter because pulse width differ by one or two CLKOUT periods.

Workaround: Set DIVFCT to 0.

3 Documentation Support

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

To access documentation on the web site:

1. Go to **http://www.ti.com**
2. Click on the **“TMS320™ DSP Products”** link
3. Scroll to “**C54X™ DSP Generation**” and click on “**DEVICE INFORMATION**”
4. Click on a device name and then click on the documentation type you prefer.

For further information regarding the TMS320VC5441, please refer to:

- *TMS320VC5441 Fixed-Point Digital Signal Processor* data manual, literature number SPRS122
- *TMS320C54x™ DSP Functional Overview,* literature number SPRU307

The five-volume *TMS320C54x DSP Reference Set,* literature number SPRU210, consisting of:

- *Volume 1: CPU and Peripherals,* literature number SPRU131
- *Volume 2: Mnemonic Instruction Set,* literature number SPRU172
- *Volume 3: Algebraic Instruction Set,* literature number SPRU179
- *Volume 4: Applications Guide,* literature number SPRU173
- *Volume 5: Enhanced Peripherals,* literature number SPRU302

The reference set describes in detail the TMS320C54x™ DSP products currently available and the hardware and software applications, including algorithms, for fixed-point TMS320™ DSP family of devices.

TMS320, C54x, and TMS320C54x are trademarks of Texas Instruments.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

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