TMS320VC5502 and

TMS320VC5501

Digital Signal Processors

Silicon Errata

SPRZ020L
December 2002
Revised June 2007

REVISION HISTORY

This revision history highlights the technical changes made to SPRZ020K to generate SPRZ020L.

Scope: Applicable updates relating to the VC5502/VC5501 device have been incorporated.

PAGE NO(S).	ADDITIONS/CHANGES/DELETIONS
9	Added Usage Note: McBSP: Receiver and/or Transmitter Must Be Out of Reset to Enable Frame-Sync Detection

Contents

1 Introduction

This document describes the silicon updates to the functional specifications for the TMX320VC5502 and the TMX320VC5501. The updates are applicable to:

- TMX320VC5502 (176-pin LQFP, PGF suffix)
- TMX320VC5502 (176-ball MicroStar BGA™, GGW suffix)
- TMX320VC5502 (201-ball MicroStar BGA™, GZZ suffix)
- TMX320VC5502 (201-ball MicroStar BGA™, ZZZ suffix)
- TMX320VC5501 (176-pin LQFP, PGF suffix)
- TMX320VC5501 (176-ball MicroStar BGA™, GGW suffix)
- TMX320VC5501 (201-ball MicroStar BGA™, GZZ suffix)
- TMX320VC5501 (201-ball MicroStar BGA™, ZZZ suffix)

NOTE: The GGW package option is only available on some TMX devices and will not be supported in the future. Instead, all MicroStar BGA options for the 5502 and 5501 will be moved to the GZZ and ZZZ packages. Refer to Section 4 for more information on this package change.

A quick reference table (Table 1) is included so that advisory descriptions may be quickly located from the short description of each advisory. The advisory number in the first column of the table is referenced in the title of each advisory description that follows in Section 3.

The advisory numbers in this document are not sequential. Some advisories have been removed and documented in a user’s guide and/or data sheet. When advisories are moved or deleted, the remaining advisory numbers remain the same and are not resequenced.

Issues related to CPU operation are documented in the *TMS320C55x DSP CPU Programmer’s Reference Supplement* (literature number SPRU652).

Table 1. Quick Reference Table of Advisories[†]

Advisory Number	Device:		5501		5502	
	Silicon Revision:		Original	Original	Original	Original
	Bootloader Revision:		Original	A	Original	A
Multichannel Buffered Serial Port						
McBSP_1	First Channel Will Not Transmit Correctly When Using 2-Partition Mode		x	x	x	x
McBSP_2	Sign Extension Does Not Work When McBSP is Configured for 20-Bit Data With Right-Justification		x	x	x	x
McBSP_3	First Channel in First Block Will Not Generate Transmit DMA Event or Transmit CPU Interrupt in 8-Partition Mode When Transmit Frame Length is Less Than 128 Words		x	x	x	x
Direct Memory Access						
DMA_1	DMA will not Reset Correctly if Disabled During a Source or Destination Port Access Request		x	x	x	x
DMA_2	DMA Channel Will Hang if Extra Sync Event is Received at the End of Block Transfer Before Channel has Been Disabled		x	x	x	x
Universal Asynchronous Receiver and Transmitter						
UART_2	Data Corrupted if CPU Writes to the Transmitter Holding Register (THR) Within One Baud Clock After a THR-Empty Interrupt		x	x	x	x
UART_3	Spurious Receive Event is Generated When Byte at Top of RX FIFO has Error		x	x	x	x
UART_4	State of Parity Bit is Not Checked When Detecting a BREAK Condition		x	x	x	x
Boundary Scan						
BSDL_1	Outputs Update Late		x	x	x	x
Clock Generator						
CLKGEN_1	CLKOUT Pin is Set to DSP Input Clock for a Certain Number of Clock Cycles After Reset		x	x	x	x
Miscellaneous						
MISC_1	PSENSE Pin Must be Connected to Ground	PGF	x		x	
		GGW	x		x	
		GZZ				
		ZZZ				

[†] “x” indicates advisory applies to this device revision.

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 TMS320™ DSP devices and support tools. Each TMS320™ DSP commercial family member has one of three prefixes: TMX, TMP, or TMS. 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.

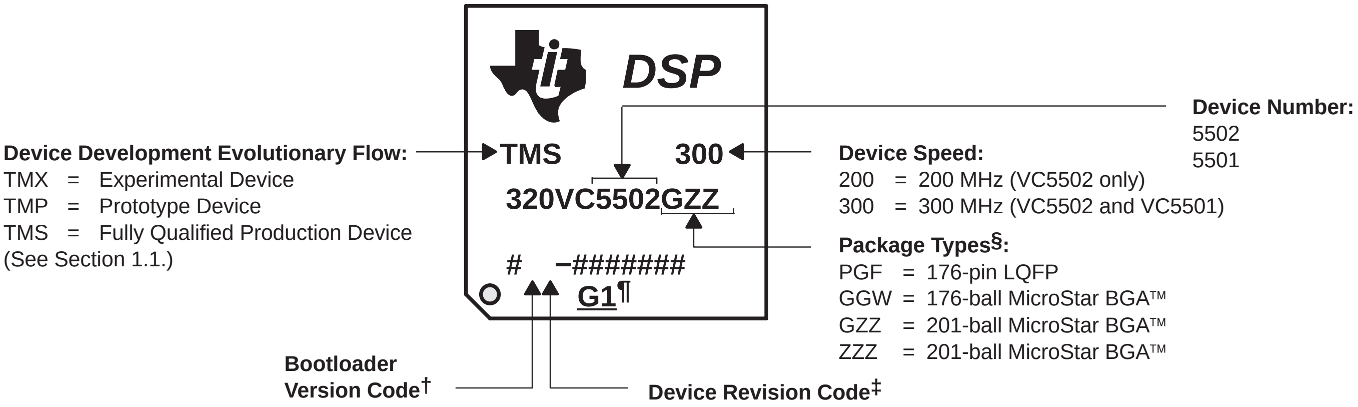
TMS320 is a trademark of Texas Instruments.

1.2 Device Revision and Bootloader Version Identification

The package markings on the TMS320VC5502 and the TMS320VC5501 can be used to determine the device speed, the device revision, and the bootloader version for the device being used. A device revision code and a bootloader version code are included in the device package markings as shown in Figure 1. Table 2 and Table 3 show how to determine the device revision and bootloader version from the codes in the package markings.

Qualified devices are marked with the letters “TMS”, while nonqualified devices are marked with the letters “TMX” or “TMP”. All GGW package devices are nonqualified devices and are marked with the letters “TMX”.

The GGW package option will not be supported in the future. Instead, all MicroStar BGA options for the 5502 and 5501 will be moved to the GZZ and ZZZ packages. Refer to Section 4, Information Regarding GGW Package and Change to GZZ Package, for more information on this package change.


DSP
TMS
300
320VC5502GZZ
# −#######
G1¶
Device Number:
5502
5501
Device Development Evolutionary Flow:
TMX = Experimental Device
TMP = Prototype Device
TMS = Fully Qualified Production Device
(See Section 1.1.)
Device Speed:
200 = 200 MHz (VC5502 only)
300 = 300 MHz (VC5502 and VC5501)
Package Types§:
PGF = 176-pin LQFP
GGW = 176-ball MicroStar BGA™
GZZ = 201-ball MicroStar BGA™
ZZZ = 201-ball MicroStar BGA™
Bootloader Version Code†
Device Revision Code‡

NOTE: “#” denotes alphanumeric character.

† The first space after the first alphanumeric character is reserved for the Bootloader Version Code (see Table 3).

‡ The second space after the first alphanumeric character is reserved for the Device Revision Code (see Table 2).

§ LQFP = Low-Profile Quad Flatpack
BGA = Ball Grid Array

¶ This is only present on the ZZZ package.

Figure 1. Example Markings for TMS320VC5502 and TMS320VC5501

Table 2. Determining Silicon Revision From Package Markings

DEVICE REVISION CODE	DEVICE REVISION	COMMENTS
Blank	Indicates Original Silicon	Some initial 5502 PGF and GGW samples will not have the TI logo; instead, they will have the number “980”. Furthermore, some of the 5502 samples in the GGW package will use “XDVC5502GGW” instead of “320VC5502GGW”; the “XDV” denotes a nonqualified device.

Table 3. Determining Bootloader Version From Package Markings

BOOTLOADER VERSION CODE	BOOTLOADER VERSION	COMMENTS
Blank	Indicates Initial Bootloader Version	Refer to the application report titled *Using the TMS320VC5501/C5502 Bootloader* (literature number SPRA911) for more information on the differences between bootloader versions.
A	Version A	

2 Usage Notes

Usage Notes highlight and describe particular situations where the device's behavior may not match the 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.

I^{2}C: Addressed-As-Slave (AAS) Bit is Not Cleared Correctly (Affects 5502 and 5501)

The addressed-as-slave (AAS) bit indicates that the I^{2}C peripheral has recognized its own slave address on the I^{2}C bus. In normal (proper) operation for both 7- and 10-bit addressing modes, the AAS bit has the capability to know that its own I^{2}C peripheral has been addressed as a slave and is now capable of transferring/receiving. Also, in normal operation for both addressing modes, the AAS bit is subsequently cleared by receiving a STOP condition *or* by a slave address different from the I^{2}C peripheral's own slave address.

Currently, in the 7-bit addressing mode, the AAS bit *is* cleared when receiving a NACK, a STOP condition, or a repeated START condition. The AAS bit currently is *not* cleared by receiving a slave address different from the I^{2}C peripheral's own slave address.

Currently, in the 10-bit addressing mode, the AAS bit *is* cleared when receiving a NACK, a STOP condition, or by a slave address different from the I^{2}C peripheral's own slave address. The AAS bit, in 10-bit addressing mode, is *not* cleared by a repeated START condition.

For either address mode, the AAS bit is properly set when addressed as a slave.

The only divergence from normal operation is how the AAS bit is cleared in either address modes.

Note the AAS bit is set correctly when the I^{2}C peripheral is addressed as a slave for both addressing modes. Also, take into account that when the AAS bit is cleared, the I^{2}C peripheral is no longer addressed as a slave, regardless of addressing mode.

(Internal reference number: 1483)

McBSP: Receiver and/or Transmitter Must Be Out of Reset to Enable Frame-Sync Detection

The McBSP transmitter and receiver on the C5502/01 device can generate an interrupt upon the detection of frame synchronization. The *TMS320VC5501/5502/5503/5507/5509/5510 DSP Multichannel Buffered Serial Port (McBSP) Reference Guide* (literature number SPRU592) states that this feature will operate even while the associated portion of the serial port is in reset. However, on the C5502/01 device, the receiver and/or transmitter **must be** out of reset to enable this feature.

Note: Frame synchronization can be detected while the receiver or transmitter is in reset by using the frame-sync pin (FSR or FSX) GPIO mode. In this configuration, the CPU can monitor the frame-sync pin status and switch to non-GPIO mode when a transition on the frame-sync pin is detected.

For more information on the GPIO mode and frame-sync detection feature of the McBSP, see the *TMS320VC5501/5502/5503/5507/5509/5510 DSP Multichannel Buffered Serial Port (McBSP) Reference Guide* (literature number SPRU592).

3 Known Design Marginality/Exceptions to Functional Specifications

3.1 Multichannel Buffered Serial Port (McBSP) Advisory Descriptions

Advisory McBSP_1 — *First Channel Will Not Transmit Correctly When Using 2-Partition Mode*

Revision(s) Affected: See Table 1

Details: As described in the *TMS320VC5501/5502/5503/5507/5509/5510 DSP Multichannel Buffered Serial Port (McBSP) Reference Guide* (literature number SPRU592), the McBSP transmitter and receiver can be independently configured for 2-partition mode and 8-partition mode. When configured for 2-partition mode, up to 32 channels can be used to transmit and receive data, where the frame-sync pulse signifies the start of the first channel.

However, when the transmitter is configured for 2-partition mode, the DX pin will not shift out the first three bits of the first channel if the last channel in the previous frame is disabled; the DX pin will actually stay in high-impedance state for the first two bits in the word being transmitted. For example, assume the following configuration for the transmitter:

- 2-partition mode (XMCME = 0)
- 8-bit word length (XWDLEN1 = 000b)
- Channels 0−15 are assigned to Partition A
- Channels 0−15 are enabled (XCERA = 0xFFFF)
- Channels 16−31 are assigned to Partition B
- Channels 16−31 are disabled (XCERB = 0x0000)
- Transmit frame length set to 32 (XFRLEN1 = 31)
- Frame period set to 256 (FPER = 255)

In this case, the first channel (channel 0) in the first frame will transmit correctly. However, at the start of the second frame and all subsequent frames, the DX pin will not shift out the first three bits of channel 0 since channel 31 is disabled.

If the first channel in the frame is never used, then this advisory can be ignored.

Assembler Notification: N/A

Workaround(s):

1. When using the McBSP transmitter in 2-partition mode, enable the last channel in the frame. For the configuration listed above, this would involve setting XCERB = 0x8000.

   If the application is reassigning channel blocks to partition A and partition B during transmission, care must be taken to ensure that the last channel in the frame is always enabled.

First Channel Will Not Transmit Correctly When Using 2-Partition Mode (Continued)

2. Use 8-partition mode with a transmitter multichannel configuration that will generate signals as a 2-partition mode would. For example, the transmitter configuration listed above can be modified as follows:

- 8-partition mode (XMCME = 1)
- 8-bit word length (XWDLEN1 = 000b)
- Channel assignments are fixed in 8-partition mode
- XCERA configured the same as the case above (XCERA = 0xFFFF)
- XCERB configured the same as the case above (XCERB = 0x0000)
- XCERC, XCERE, and XCERG configurations the same as XCERA
- XCERD, XCERF, and XCERH configurations the same as XCERB
- Transmit frame length set to 128 (XFRLEN1 = 127)
- Frame period set to 256 so that a new frame appears every 32 words (FPER = 255)
- Set Transmit Frame-Sync Ignore bit (XFIG = 1) so that the McBSP will ignore the three extra frames being generated by the sample-rate generator

Using this configuration, the signals at the transmitter pins of the McBSP would look like the signals for the 2-partition case.

Advisory McBSP_2

Sign Extension Does Not Work When McBSP is Configured for 20-Bit Data With Right-Justification

Revision(s) Affected: See Table 1

Details: The RJUST bits in SPCR1 are used to select whether data in RBR1 and RBR2 is right- or left-justified (with respect to the MSB) in DRR1 and DRR2, and how unused bits in DRR1 and DRR2 are filled—with zeros or with sign bits. If the data is negative (MSB is ‘1’) and right-justification is chosen, unused bits are filled with sign bits; otherwise, unused bits are filled with zeroes.

When the McBSP is configured for a data length of 20, it will use bit 15 of the data as the sign bit instead of bit 19; therefore, right-justification with sign-extension will not work properly. All other data lengths behave as described above and can be used with no problem.

Assembler Notification: N/A

Workaround: None

Advisory McBSP_3

First Channel in First Block Will Not Generate Transmit DMA Event or Transmit CPU Interrupt in 8-Partition Mode When Transmit Frame Length is Less Than 128 Words

Revision(s) Affected: See Table 1

Details: The McBSP transmitter can be configured for multichannel mode with 8-bit partition. When the McBSP is configured for 8-partition mode, up to eight blocks of 32 channels each (a total of 128 channels) can be used to transmit data.

When using multichannel mode, the XFRLEN1 bits of XCR1 specify the number of time slots or channels per frame-synchronization period. The FPER bits of SRGR2 determine the period of the frame-sync signal generated by the sample-rate generator.

A common approach used when programming the McBSP to use less than the full 128 available channels is:

- set the XFRLEN1 equal to the desired number of channels, and
- set FPER to a value equal to or greater than the desired number of channels (specified in shift clock cycles)

However, when using this approach for the transmitter, the McBSP will only generate the transmit event to the DMA and the transmit interrupt to the CPU for the first channel (channel 0) in block A for the first frame, but not for subsequent frames. With no new data in the DXR, the transmitter will re-send the data from the previous transfer.

As an example of this erratum, assume the following configuration is used for the transmitter for a 64-slot time-division multiplexed (TDM) data stream with 8-bit words, where channel 0 is enabled:

- 8-partition mode (XMCME = ‘1’)
- 8-bit word length (XWDLEN1 = 000b)
- Channel assignments are fixed in 8-partition mode
- XCERA = 0x1111 (other channels can be enabled, the main point here is that channel 0 in block A is enabled)
- XCERB = XCERC = XCERD = 0x2222
- XCERE, XCERF, XCERG, and XCERH are unused
- Transmit frame length set to 64 (XFRLEN1 = 63)
- Frame period set to 512 (FPER = 511) so that a new frame appears every time 64 bytes are shifted out

In this case, when the first frame is generated, the McBSP will generate the transmit interrupt and the DMA event for channel 0 and all subsequent enabled channels. However, at the start of the second frame and all subsequent frames, the McBSP will not generate the transmit interrupt or DMA event for channel 0.

Note that if the first channel in the frame is not used, then this advisory can be ignored. Also note that this erratum does not apply to the receiver; however, for simplicity in programming the McBSP, the same configuration described in the workarounds below can be used for the receiver if so desired.

First Channel in First Block Will Not Generate Transmit DMA Event or Transmit CPU Interrupt in 8-Partition Mode When Transmit Frame Length is Less Than 128 Words (Continued)

More details on the operation of the McBSP in multichannel mode are given in the *TMS320VC5501/5502/5503/5507/5509/5510 DSP Multichannel Buffered Serial Port (McBSP) Reference Guide* (literature number SPRU592).

Assembler Notification: N/A

Workaround(s):

1. Use all eight blocks in 8-partition mode with a transmitter multichannel configuration that will generate signals as the desired configuration. For example, the transmitter configuration listed above can be modified as follows:
   - 8-partition mode (XMCME = ‘1’)
   - 8-bit word length (XWDLEN1 = 000b)
   - Channel assignments are fixed in 8-partition mode
   - XCERA, XCERB, XCERC, and XCERD configured the same as the example above
   - XCERE, XCERF, XCERG, and XCERH configured the same as XCERA, XCERB, XCERC, and XCERD, respectively
   - Transmit frame length set to 128 (XFRLEN1 = 127)
   - Frame period set to 512 (FPER = 511) so that a new frame appears every time 64 bytes are shifted out
   - Set Transmit Frame-Sync Ignore bit (XFIG = ‘1’) so that the McBSP will ignore the extra frame being generated by the sample-rate generator

   Using this configuration, the signals at the transmitter pins of the McBSP would look like the signals a 64-slot time-division data stream.
2. Do not use channel 0 when using the McBSP transmitter in 8-partition mode with an XFRLEN1 < 128 words.

3.2 Direct Memory Access (DMA) Advisory Descriptions

Advisory DMA_1

DMA will not Reset Correctly if Disabled During a Source or Destination Port Access Request

Revision(s) Affected: See Table 1

Details: A DMA transfer can be thought of as two different operations: a port read access and a port write access. During the port read access, a DMA channel reads data from the source port and places it in its FIFO buffer (8 x 32-bit on the 5502/5501). The channel takes data from the FIFO buffer and writes it to the destination port during the port write access. By having the data in the channel FIFO, the DMA channel can transfer the data to the destination port much faster since it does not have to wait for the source read to complete.

The DMA will go into some unknown state if one of its channels is disabled while a port read access or a port write access request is pending. This problem is best observed by following this sequence of events:

1. Configure the DMA for repeat mode (autoinitialization on and repeat on).
2. Enable the DMA channel to start block transfer.
3. CPU disables DMA channel in the middle of a block transfer.
4. DMA channel goes into an unknown state and cannot be reused until a hardware reset is asserted.

Assembler Notification: N/A

Workaround: The DMA channel must be disabled (EN = 0) only when there are no pending port read or write requests. Since the DMA channel transfers data continuously when no synchronization events are used, the DMA channel must be disabled only after it has completed its current block transfer. When using repeat mode, this can be accomplished by following this sequence of steps:

1. Set repeat mode off (REPEAT = 0) and leave autoinitialization on (AUTO_INIT = 1).
2. Wait for the next block transfer to complete; the DMA will not start another transfer unless the CPU sets ENDPROG.
3. DMA channel can now be disabled.

A DMA block interrupt can be used to determine when the DMA channel has finished transferring the current block. NOTE: when the repeat mode is on, the DMA will generate the interrupt and immediately commence the next block transfer; therefore, disabling the DMA channel in the interrupt service routine is not a valid workaround.

DMA will not Reset Correctly if Disabled During a Source or Destination Port Access Request (Continued)

When synchronization events are used, the DMA channel can be disabled by:

- Following the procedure outlined above for the no-synchronization event case.
- Disabling the source generating the synchronization event. For example, when using the McBSP transmit event, first disable the McBSP so that it does not generate any more DMA events, and then disable the DMA channel.
- Ensuring that any port read or write request(s) started by the previous synchronization event has been completed.

NOTE: the SYNC bit of the DMACSR cannot be used to determine if there are pending port read or write requests. The operation of the SYNC bit is described in the *TMS320VC5501/5502 DSP Direct Memory Access (DMA) Controller Reference Guide* (literature number SPRU613D or later).

Advisory DMA_2

DMA Channel Will Hang if Extra Sync Event is Received at the End of Block Transfer Before Channel has Been Disabled

Revision(s) Affected: See Table 1

Details: A channel can specify that the elements in a block be transferred only when certain events are generated, e.g., a McBSP receive event or an I^{2}C transmit event. Therefore, a block transfer of 20 elements, for example, would only require 20 sync events (element synchronization is used). Furthermore, in normal operation, the DMA controller will disable a DMA channel once all the elements in the block transfer specified by that channel have been transferred.

However, an extra synchronization event received at the end of a programmed block transfer will cause the DMA channel to hang—i.e., it will not be able to request any more data transfers—if it is received before the channel has been disabled. For example, assume one frame of four elements is being transferred using a sync event. If a fifth sync event is received before the channel has been disabled by the DMA controller, then the channel will hang and will no longer be usable. Furthermore, the SYNC bit of the DMACSR register will be set to ‘1’ by the extra sync event and will stay set even after EN has been cleared to ’0’. Also, the DROP bit of the DMACSR register will not be set to ‘1’ to indicate that a synchronization event was received before the transfer started by the previous sync event had been completed.

Assembler Notification: N/A

Workaround: The CPU can detect that a channel has been affected by this errata by checking that the SYNC bit is set to ‘1’ when EN = ‘0’. To “un-hang” the DMA channel, the CPU must write ‘00000b’ to the SYNC field in DMACCR before programming the channel for more data transfers.

3.3 Universal Asynchronous Receiver and Transmitter (UART) Advisory Descriptions

Advisory UART_2

Data Corrupted if CPU Writes to the Transmitter Holding Register (THR) Within One Baud Clock After a THR-Empty Interrupt

Revision(s) Affected: See Table 1

Details: The THR-empty interrupt is generated every time the UART is ready to transmit data. An interrupt service routine can be set up such that the CPU copies data into the THR every time a THR-empty interrupt is generated by the UART. In normal operation, the THR should receive the data byte to be transmitted from the CPU, copy it into the TSR, and then generate another THR-empty interrupt to signal that it is ready to transmit more data.

However, if the data byte is written to the THR within one baud clock cycle after a THR-empty interrupt was generated, the THR will receive the data and immediately generate another THR-empty interrupt. If the CPU writes a second data byte to the THR within one baud clock of the initial interrupt, then the first data byte will be overwritten. With the loss of the first data byte, the last data byte in the transmission sequence will be transmitted twice.

The behavior just described only occurs when the UART transmitter is operated in non-FIFO mode. In FIFO mode, the THR-empty interrupt will automatically be delayed by one baud clock; therefore, the CPU can write to the THR as soon as it receives the THR-empty interrupt. Note that this behavior does not affect systems that use the DMA to write data to the THR since the UART transmitter has to be operated in FIFO mode to use the DMA.

Assembler Notification: N/A

Workaround(s):

1. When using the CPU to service the THR-Empty interrupt, place a delay large enough such that a data byte is not written to the THR until one baud clock has passed after the interrupt was generated.
2. Enable the FIFO mode of the UART by setting the FIEN bit to 1 in the FIFO Control Register (URFCR). The condition for generating a THR-empty interrupt is similar in the non-FIFO and FIFO modes: in non-FIFO mode, the THR-empty interrupt is generated when the THR register is empty; while in the FIFO mode, the THR-empty interrupt is generated when the transmitter FIFO is empty.

   Notice that by enabling the FIFO mode of the transmitter, the FIFO mode of the receiver is also enabled. However, the RFITR bits of the URFCR can be set to 00b (trigger level of one byte) if the user needs the UART to generate the receiver data-ready interrupt after receiving a single byte.

   (Internal reference number: 1494)

Advisory UART_3 *Spurious Receive Event is Generated When Byte at Top of RX FIFO has Error*

Revision(s) Affected: See Table 1

Details: The UART can detect a number of errors when receiving data, e.g., frame error, parity error, etc. The UART generates a receive event and a receive interrupt which can be used by the DMA or CPU, respectively, to read data from the UART receive FIFO. In normal operation, when the UART is operated in FIFO mode and the DMA is used to read incoming data, the UART receive event will not be generated when the FIFO reaches the trigger level if one of the data bytes currently in the FIFO was received with an error. An error interrupt will be generated at the time the error is detected; this error interrupt must be serviced by the CPU.

If the erroneous byte is located at the top of the FIFO at the time the trigger level has been reached, the UART will operate as described above, i.e., the receive event will not be generated and an error interrupt will be generated by the UART. However, at the time the erroneous byte is read off the FIFO, a spurious receive event will be generated, causing the DMA to read data from the FIFO. This issue is further complicated if the DMA is programmed to read N bytes from the FIFO and the trigger level is set to N. In this case when the erroneous byte is read off the FIFO, there will be N − 1 data bytes left, but the DMA will try to read N data bytes from the FIFO when it receives the spurious DMA event, causing the DMA to read invalid data.

Assembler Notification: N/A

Workaround: To avoid this problem, the CPU must be programmed to service interrupt errors from the UART. When an error is detected by the UART, the interrupt service routine must follow this procedure:

1. Disable the DMA.
2. Clear the erroneous byte from the FIFO (the spurious receive event will be generated immediately).
3. Enable the DMA (by this time, the spurious receive event will have already passed).
4. Return to normal operation.

Steps 3 and 4 are optional depending on how the user wants to deal with receiving erroneous data, i.e., discard data and continue, stop all serial transfers from the UART, etc. Note that the DMA will stop copying data from the UART FIFO until the error condition is cleared.

(Internal reference number: 1495)

Advisory UART_4 — *State of Parity Bit is Not Checked When Detecting a BREAK Condition*

Revision(s) Affected: See Table 1

Details: The break indicator bit (BI) is set to '1' if the receive data input (RX) is held low for longer than a full-word transmission time (BREAK condition). A full-word transmission time is defined as the total time to transmit the start, data, parity, and stop bits.

The UART can also detect a number of errors, e.g., frame error, parity error, etc. A frame error occurs (FE bit is set to '1') when the receive character does not have the valid stop bit (see Figure 2). A parity error is generated (PE bit is set to '1') when the parity of the received character does not match the parity selected with the even parity select bit (EPS) of the URLCR.

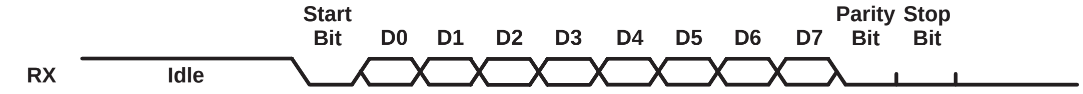


Figure 2. Frame Error

A scenario in which the received data is all zeros, the parity bit is '1', and the stop bit is '0' (see Figure 3) is correctly identified by the UART as a frame error because the stop bit is not received. In addition, although the parity bit was received, the UART will not check the state of the parity bit and it incorrectly interprets this scenario as a BREAK condition (BI bit is set to '1').

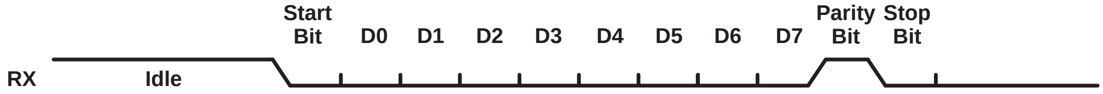


Figure 3. Advisory Scenario

Assembler Notification: N/A

Workaround: When both a BREAK condition and a frame error are identified by the UART, several steps must be taken to determine if the BREAK condition is true. When the received data is all 0s, the state of EPS and PE can be used to determine the state of the parity bit at the time the data was received. In other words,

- When EPS is 0 and PE is 0, the parity bit was '1'
- When EPS is 0 and PE is 1, the parity bit was '0'
- When EPS is 1 and PE is 0, the parity bit was '0'
- When EPS is 1 and PE is 1, the parity bit was '1'

If the BI bit is set to '1' and the parity bit was '1', then the BI indicator has detected a false BREAK condition. Table 4 can be used to determine if the BI bit was set by a true BREAK condition.

State of Parity Bit is Not Checked When Detecting a BREAK Condition (Continued)

Table 4. Distinguishing Between BREAK Condition and Frame Error[†]

BI	FE	PE	PES	RESULT
1	0	X	X	N/A
1	1	0	0	Framing Error (data is zero)
1	1	0	1	BREAK Condition[‡]
1	1	1	0	BREAK Condition

[†] Table assumes parity is being used. If parity is not used, then it is not possible to distinguish between a BREAK condition and a frame error in the case where the data is zero and the stop bit is not received.

[‡] In this case, both the frame error condition and the BREAK condition are met since there is no stop bit and there is no parity bit. It is up to the user to determine a proper course of action in this case.

(Internal reference number: 1502)

3.4 Boundary Scan Advisory Descriptions

Advisory BSDL_1 — *Outputs Update Late*

Revision(s) Affected: See Table 1

Details: This advisory applies to all boundary scan test instructions that pass through the UPDATE_DR or UPDATE_IR states.

Case 1: Outputs change one half-cycle late in Update-IR state

According to the IEEE 1149.1[†] standard, when the EXTEST instruction is selected through an instruction register scan, the EXTEST state should become valid on the falling edge of TCK in the *Update-IR* controller state. Also at this point, the function of the device output pins should change from normal functional mode to boundary-scan mode and the contents of the boundary register are driven on the pins.

On the TMS320VC5502 and the TMS320VC5501 devices, this output state change happens one half-cycle late. During the BYPASS and SAMPLE/PRELOAD instructions, the device outputs are in their normal functional states. During EXTEST, the device outputs are in a state controlled by the device boundary register. It is common to load an output state in the boundary register using the SAMPLE/PRELOAD instruction, and then cause the outputs to change to that state by executing the EXTEST instruction. Since the outputs on the DSP will change a half-cycle after other IEEE 1149.1-compliant devices on the board, the DSP will still be driving its normal functional mode on the pins while the other IEEE 1149.1-compliant devices on the board are driving the boundary registers contents on their output pins. This produces a risk of drive contention during this half-cycle.

Case 2: Outputs change one half-cycle late in Update-DR state

According to the IEEE 1149.1 standard, when the EXTEST, INTEST, or SAMPLE/PRELOAD instructions are selected, latched parallel outputs of the boundary-scan shift register cells should change state on the falling edge of TCK in the *Update-DR* controller state. The DSP output pins should also change state at the same time.

On the TMS320VC5502 and the TMS320VC5501 devices, this output state change happens one half-cycle late. Since the outputs on the DSP will change a half-cycle after other IEEE 1149.1-complaint devices on the board, there is the risk of drive contention during this half-cycle.

Assembler Notification: N/A

[†] IEEE Standard 1149.1-1990 Standard Test Access Port and Boundary-Scan Architecture.

Workaround: The following operations may be used to minimize or eliminate the drive contention:

1. Minimize the TCK period to minimize the length the half TCK cycle where drive contention may occur.
2. When a bus with multiple output drivers changes ownership (drive direction) from one driver to another, use one of the following methods:

   a. Use the HIGH-Z instruction prior to each EXTEST that causes a change in the direction of the bus. The HIGH-Z instruction can ensure all output drivers connected to the bus are in a high-impedance state prior to the execution of the EXTEST instruction. This will prevent drive contention.

   b. Use the EXTEST instruction to load a high-impedance condition to all device outputs on the affected bus prior to each EXTEST that causes a change in the direction of the bus. This will prevent drive contention.
3. Modify pin definitions where possible to test pins as inputs-only or outputs-only based on functional use.

Method 2a or 2b must be applied to all devices on the affected bus, not just the DSP. These workarounds should be applied to each of the following cases where the bus functions change:

- Between the preload of the boundary register and the application of EXTEST (or any other bus transitions from normal operation mode to boundary-scan test mode)
- Between EXTEST operations that will cause the bus direction to change
- Between EXTEST and a return to SAMPLE/PRELOAD, BYPASS, or any other bus transitions from boundary-scan test mode to normal operation mode.

The practices mentioned above can be considered best practices for boundary-scan tests in general (not withstanding the advisory mentioned here) because they also account for avoiding normal contention due to differing output switching characteristics between devices on the board and differing propagation delays due to board trace lengths. Some boundary-scan ATPG products already implement this type of capability. Contact your tool vendor for specific information on how this capability is supported on your tool.

3.5 Clock Generator Advisory Descriptions

Advisory CLKGEN_1

CLKOUT Pin is Set to DSP Input Clock for a Certain Number of Clock Cycles After Reset

Revision(s) Affected: See Table 1

Details: The CLKOUT pin can be set to reflect three of the clock generator internal clocks (SYSCLK1, SYSCLK2, and SYSCLK3) by setting the CLKOSEL bit of the CLKOUT Selection Register (CLKOUTSR) (see the System Clock Generator section of the device-specific data manual for more information on this feature). After reset, the value of the CLKOSEL bits is set to ‘1001b’ such that SYSCLK1 drives the CLKOUT pin. The CLKOUT pin will toggle at one-fourth the input clock frequency since after reset, the divider that generates SYSCLK1 is programmed to divide the DSP input clock by four.

After the $\overline{\text{RESET}}$ pin is brought high, the CLKOUT pin will be set to the DSP input clock for a certain number of clock cycles based on the state of the GPIO4 pin during reset. If GPIO4 is high during reset (oscillator disabled), CLKOUT will be set equal to the DSP input clock for 70 input clock cycles. If GPIO4 is low during reset (oscillator enabled), CLKOUT will be set to the DSP input clock for 41,102 input clock cycles. After these clock cycles have expired, CLKOUT will be set to SYSCLK1 and it will toggle at one-fourth the speed of the DSP input clock.

Assembler Notification: N/A

Workaround: None

3.6 Miscellaneous Advisory Descriptions

Advisory MISC_1	*PSENSE Pin Must be Connected to Ground*

Revision(s) Affected: See Table 1

Details: PGF and GGW devices containing both original silicon and the original bootloader revision have a PSENSE pin which was used during device qualification. The PSENSE pin is located at pin 61 on PGF devices and at ball U8 on GGW devices. On these devices, the PSENSE pin must be connected to ground.

The PSENSE pin becomes a “No Connect” pin in GGW and PGF devices with other silicon and bootloader revisions and can be left unconnected. GZZ devices do not have a PSENSE pin; therefore, ball U8 can be left unconnected.

Assembler Notification: N/A

Workaround: None

4 Information Regarding GGW Package and Change to GZZ Package

The MicroStar BGA option for the TMS320VC5502 and TMS320VC5501 was changed from the 176-terminal GGW package to the 201-terminal GZZ package. The GZZ package is the same as the GGW package with the exception of 25 thermal balls located in a 5x5 array at the center of the package. The 25 thermal balls were included to improve thermal dissipation of the device. Figure 4 and Figure 5 show the GGW package and the GZZ package, respectively.

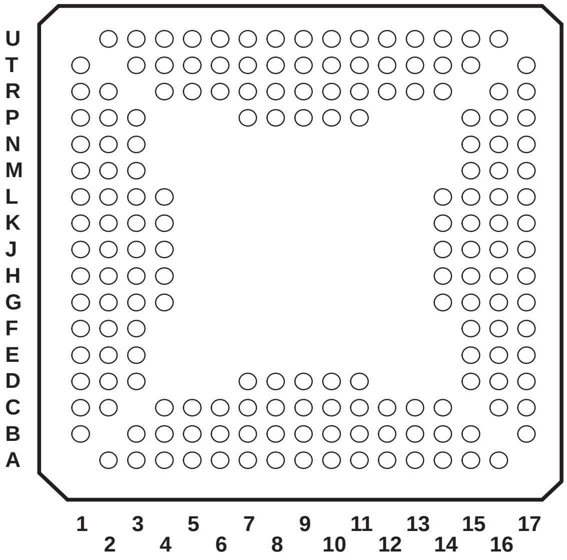


Figure 4. 176-Terminal GGW Package (Bottom View)

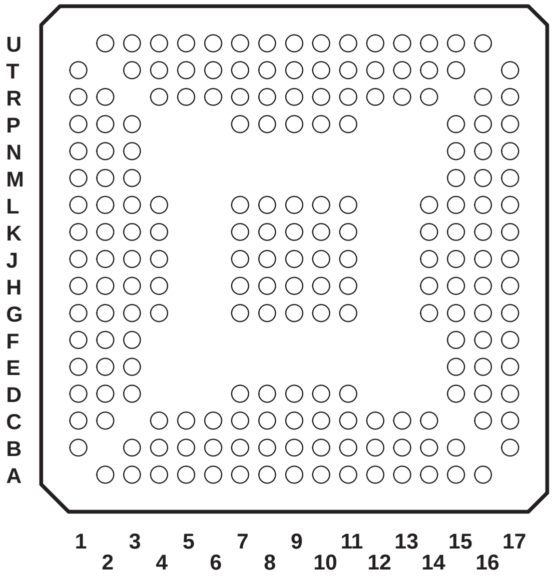


Figure 5. 201-Terminal GZZ Package (Bottom View)

All mechanical data and thermal resistance information for the GZZ package can be obtained from the device data sheets (see Section 5 for literature numbers). The device data sheets also contain information on how to connect the 25 thermal balls. Information for the GGW package is given below. Please note that the GGW package option will not be supported in the future. Instead, all MicroStar BGA devices for both the 5502 and the 5501 will only be available in the GZZ package. The information here is only presented since some TMX samples were shipped in the GGW package.

4.1 Package Thermal Characteristics for the GGW Package

Table 5 and Table 6 provide the thermal resistance characteristics for the GGW package that was used on some 5502 and 5501 TMX devices.

Table 5. Thermal Resistance Characteristics (Ambient)

PACKAGE	$R_{\theta JA}$ (°C/W)	BOARD TYPE†	AIRFLOW (LFM)
GGW	125	High-K	0
	125	High-K	150
	122	High-K	250
	118	High-K	500
	151	Low-K	0
	149	Low-K	150
	143	Low-K	250
	134	Low-K	500

† Board types are as defined by JEDEC. Reference JEDEC Standard JESD51−9, Test Boards for Area Array Surface Mount Package Thermal Measurements.

Table 6. Thermal Resistance Characteristics (Case)

PACKAGE	$R_{\theta JC}$ (°C/W)	BOARD TYPE†
GGW	27.2	2s JEDEC Test Card

† Board types are as defined by JEDEC. Reference JEDEC Standard JESD51−9, Test Boards for Area Array Surface Mount Package Thermal Measurements.

4.2 Mechanical Data for the GGW Package

To view or to download the 176-terminal GGW package mechanical drawing, go to the Texas Instruments web site at http://www.ti.com.

5 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 TMS320VC5502 and the TMS320VC5501, please refer to:

- *TMS320VC5502 Fixed-Point Digital Signal Processor* data manual for -200 and -300 devices (literature number SPRS166C or later)
- *TMS320VC5501 Fixed-Point Digital Signal Processor* data manual (literature number SPRS206)

For additional information, see the latest versions of:

- *TMS320C55x DSP CPU Reference Guide* (literature number SPRU371)
- *TMS320C55x DSP Mnemonic Instruction Set Reference Guide* (literature number SPRU374)
- *TMS320C55x DSP Algebraic Instruction Set Reference Guide* (literature number SPRU375)
- *TMS320C55x DSP Programmer's Guide* (literature number SPRU376)
- *TMS320VC5501/5502 DSP Instruction Cache Reference Guide* (literature number SPRU630)
- *TMS320VC5501/5502 DSP Timers Reference Guide* (literature number SPRU618)
- *TMS320VC5501/5502/5503/5507/5509 DSP Inter-Integrated Circuit (I2C) Module Reference Guide* (literature number SPRU146)
- *TMS320VC5501/5502 DSP Host Port Interface (HPI) Reference Guide* (literature number SPRU620)
- *TMS320VC5501/5502 DSP Direct Memory Access (DMA) Controller Reference Guide* (literature number SPRU613)
- *TMS320VC5501/5502/5503/5507/5509/5510 DSP Multichannel Buffered Serial Port (McBSP) Reference Guide* (literature number SPRU592)
- *TMS320VC5501/5502 DSP External Memory Interface (EMIF) Reference Guide* (literature number SPRU621)
- *TMS320VC5501/5502 DSP Universal Asynchronous Receiver/Transmitter (UART) Reference Guide* (literature number SPRU597)
- *TMS320C55x DSP CPU Programmer's Reference Supplement* (literature number SPRU652)
- *Using the TMS320VC5501/C5502 Bootloader* application report (literature number SPRA911)

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

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

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

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI 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 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. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

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

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

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
RFID	www.ti-rfid.com	Telephony	www.ti.com/telephony
Low Power Wireless	www.ti.com/lpw	Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

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