TMS320VC5421

Digital Signal Processor

Silicon Errata

SPRZ170C
April 2000
Revised October 2001

TEXAS INSTRUMENTS

Contents

1 Introduction . . . 3
1.1 Quality and Reliability Conditions . . . 3
TMX Definition . . . 3
TMP Definition . . . 3
TMS Definition . . . 3
1.2 Revision Identification . . . 4
2 Known Design Marginality/Exceptions to Functional Specifications . . . 5
SPI EEPROM Boot Mode . . . 5
DMPREC . . . 6
HPI/DMA Doubleword Accesses . . . 7
Round (RND) Instruction Clears Pending Interrupts . . . 7
Far Branches/Calls/Interrupts from Active Repeat Blocks (BRAF) . . . 8
NMI . . . 8
WRITA/MVDP . . . 9
HPI Hint . . . 9
100-MHz Performance . . . 10
READY Input Latch Timing for DMA_XIO Accesses . . . 10
CPU B XIO . . . 10
Shared Memory Access in Emulation Mode . . . 11
3 Documentation Support . . . 12

1 Introduction

This document describes the silicon updates to the functional specifications for the TMS320VC5421. The advisories are applicable to:

- TMS320VC5421 (144-pin LQFP, PGE suffix)
- TMS320VC5421 (144-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

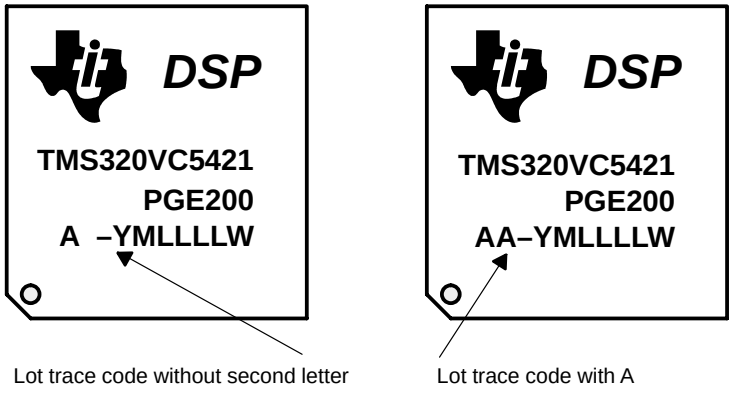
Fully-qualified production device.

MicroStar BGA is a trademark of Texas Instruments.

1.2 Revision Identification

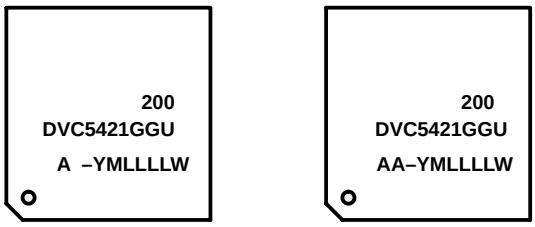
The device revision can be determined by the lot trace code marked on the top of the package. The locations for the lot trace codes for the PGE and the GGU packages are shown in Figure 1 and Figure 2, respectively. The location of other markings may vary per device.

Figure 1. Example, Typical Lot Trace Code for TMS320VC5421 (PGE)



Lot Trace Code	Silicon Revision	Comments
Blank (no second letter in prefix)	Initial Silicon	This silicon revision is available as TMX only.
A (second letter in prefix is A)	Silicon Revision A	This silicon revision is TMS (production).

Figure 2. Example, Typical Lot Trace Code for TMS320VC5421 (GGU)



NOTE: Qualified devices in the PGE package are marked with the letters “TMS” at the beginning of the device name, while nonqualified devices in the PGE package are marked with the letters “TMX” or “TMP” at the beginning of the device name. Similarly, 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
SPI EEPROM Boot Mode	Initial and Revision A Silicon	5
DMPREC	Initial and Revision A Silicon	6
HPI/DMA Doubleword Mode Accesses	Initial and Revision A Silicon	7
Round (RND) Instruction Clears Pending Interrupts	Initial and Revision A Silicon	7
Far Branches/Calls/Interrupts from Active Repeat Blocks (BRAF)	Initial and Revision A Silicon	8
NMI	Initial and Revision A Silicon	8
WRITA/MVDP	Initial and Revision A Silicon	9
HPI Hint	Initial and Revision A Silicon	9
100-MHz Performance	Initial Silicon	10
READY Input Latch Timing for DMA_XIO Accesses	Initial Silicon	10
CPU B XIO	Initial Silicon	10
Shared Memory Access in Emulation Mode	Initial Silicon	11

Advisory *SPI EEPROM Boot Mode*

Revision(s) Affected: Initial Silicon and Revision A Silicon

Details: The SPI EEPROM boot mode does not function.

Workaround: None.

Advisory *DMPREC*

Revision(s) Affected: Initial Silicon and Revision A Silicon

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:

Channel 0 transfer completes and DMA logic clears DE0 internally.

User code attempts to enable another channel (e.g., ORM #2, DMPREC)

The user code will re-enable channel 0 (DMPREC value written = 00C3h), and an additional, unintended transfer will begin on channel 0.

Workaround: There are a few use conditions under which this problem does not occur. If all active DMA channels are configured in ABU mode or in auto initialization 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 re-enabling 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 *HPI/DMA Doubleword Accesses*

Revision(s) Affected: Initial Silicon and Revision A Silicon

Details: HPI accesses can corrupt the doubleword mode bit (DBLW) of an active DMA channel. When a HPI access occurs, it takes priority over the DMA and the active DMA channel transfer is temporarily paused. When the DMA operation continues after the HPI access, the active DMA channel uses the doubleword context of the next channel. If the doubleword context for the next channel is different than the active channel, the DMA transfer will be incorrectly executed (i.e., wrong doubleword mode).

Workaround: To prevent this, all DMA channels should be programmed with the same doubleword mode. That is, all DMA transfers should use either the 16-bit mode or they should all use the 32-bit mode. Note that each DMA channel has a unique DBLW bit in its DMA Sync Event and Frame Count (DMSFCn) register, so each of these registers must be programmed.

Advisory *Round (RND) Instruction Clears Pending Interrupts*

Revision(s) Affected: Initial Silicon and Revision A Silicon

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: Replace the RND instruction with an ADD instruction as follows:

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

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

Revision(s) Affected: Initial Silicon and Revision A Silicon

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(s):

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 bug 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	*NMI*

Revision(s) Affected: Initial Silicon and Revision A Silicon

Details: An NMI can be ignored if the internal CPU interrupt logic is not adequately prepared.

Workaround: Avoid generating an NMI during the time when other interrupts are being serviced. Alternatively, use one of the other external interrupts, appropriately enabled, to serve the NMI function.

Advisory — *WRITA/MVDP*

Revision(s) Affected: Initial Silicon and Revision A Silicon

Details: If a WRITA or MVDP instruction executing from a SARAM block performs a write to any SARAM block that is immediately followed by any read (including DMA or instruction fetch) of the same address that is written to, the read data may be corrupted.

Workaround: Avoid using WRITA/MVDP to write to an area in memory that will be executed as code immediately following the WRITA/MVDP.

Rearrange the code so that a read access does not immediately follow the WRITA/MVDP instruction with an address that is identical to the last address written to. Use a dummy write if necessary.

Avoid DMA reads in an area of program SARAM that is written to by WRITA/MVDP.

Advisory — *HPI Hint*

Revision(s) Affected: Initial Silicon and Revision A Silicon

Details: The HPI will become locked up, with HRDY stuck low, if both the host processor and the 5421 CPU write a one (1) to HINT at the same time.

Workaround: Do not perform 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 *100-MHz Performance*

Revision(s) Affected: Initial Silicon

Details: At the rated core supply voltage of 1.8 V, neither CPU can function reliably at the maximum speed of 100 MHz across the full commercial case temperature range of 0°C to 85°C.

Workaround(s):

1. At the core supply voltage of 1.8 V, reduce the internal CPU clock speed to 80 MHz for functionality across the full commercial case temperature range of 0°C to 85°C.
2. TI has implemented a revised test flow beginning in April of 2000. This flow screens devices for CPU functionality at 100 MHz with a 2.0-V core supply voltage at case temperature of 50°C.

None. This problem is corrected in Revision A silicon.

Advisory *READY Input Latch Timing for DMA_XIO Accesses*

Revision(s) Affected: Initial Silicon

Details: READY is latched on the rising edge of CLKOUT (instead of the falling edge of CLKOUT).

Workaround: None. This problem is corrected in Revision A silicon.

Advisory *CPU B XIO*

Revision(s) Affected: Initial Silicon

Details: CPU B loses control of XIO when the device is put in HOLD.

Workaround: None. This problem is corrected in Revision A silicon.

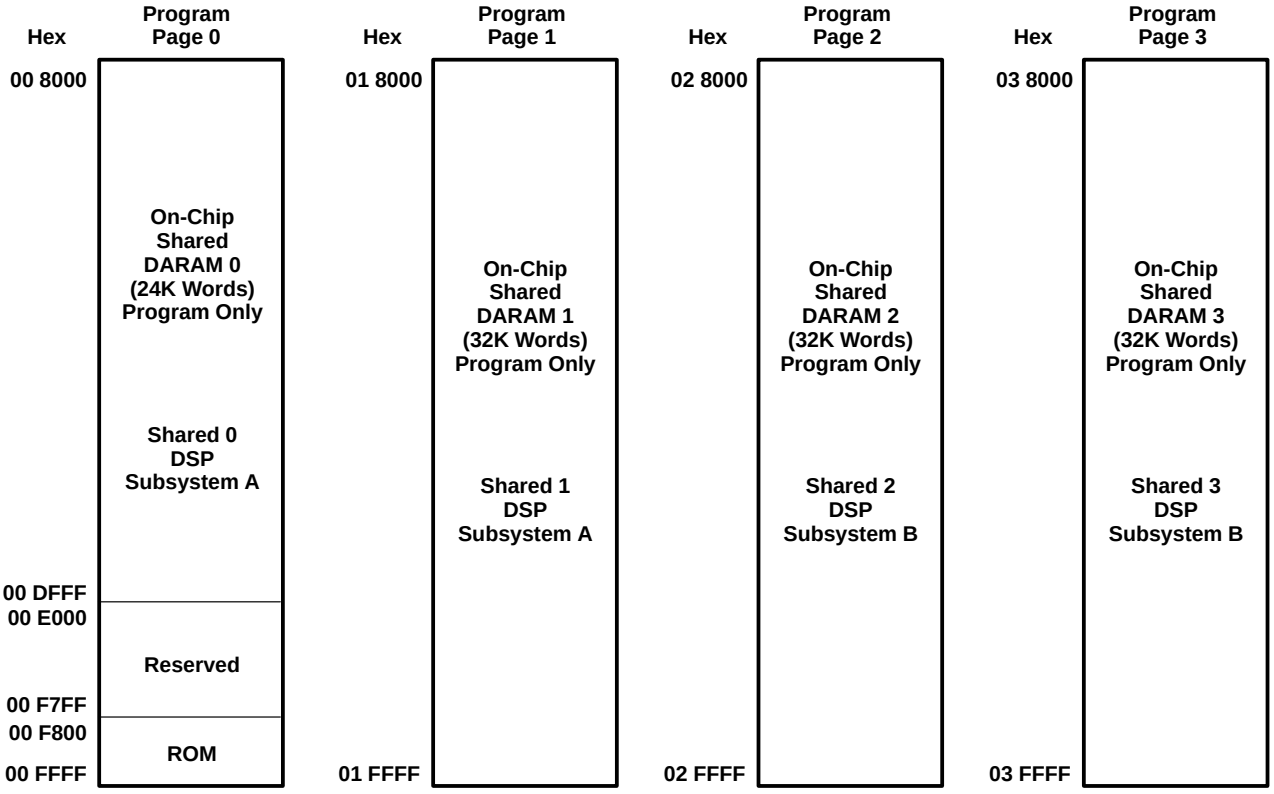
Advisory *Shared Memory Access in Emulation Mode*

Revision(s) Affected: Initial Silicon

Details: When a 5421 subsystem is halted in emulation mode, the other subsystem is unable to access the shared memory associated with the halted subsystem (see Figure 3 for the shared-memory map). This also affects the use of software and hardware breakpoints in the shared memory. Note that this issue only affects the TMX320VC5421 during emulation, and there are no issues with shared memory accesses during normal functional mode.

Workaround: None. This problem is corrected in Revision A silicon.

Figure 3. Shared-Memory Map



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. Open the "**Products**" dialog box and select "**Digital Signal Processors**"
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 TMS320VC5421, please refer to:

- *TMS320VC5421 Fixed-Point Digital Signal Processor* data sheet*,* literature number SPRS098
- *TMS320C54™ 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.

TMS320VC5421

Digital Signal Processor

Silicon Errata

SPRZ170C
April 2000
Revised October 2001

TEXAS INSTRUMENTS

Contents

1 Introduction . . . 3
1.1 Quality and Reliability Conditions . . . 3
TMX Definition . . . 3
TMP Definition . . . 3
TMS Definition . . . 3
1.2 Revision Identification . . . 4
2 Known Design Marginality/Exceptions to Functional Specifications . . . 5
SPI EEPROM Boot Mode . . . 5
DMPREC . . . 6
HPI/DMA Doubleword Accesses . . . 7
Round (RND) Instruction Clears Pending Interrupts . . . 7
Far Branches/Calls/Interrupts from Active Repeat Blocks (BRAF) . . . 8
NMI . . . 8
WRITA/MVDP . . . 9
HPI Hint . . . 9
100-MHz Performance . . . 10
READY Input Latch Timing for DMA_XIO Accesses . . . 10
CPU B XIO . . . 10
Shared Memory Access in Emulation Mode . . . 11
3 Documentation Support . . . 12

1 Introduction

This document describes the silicon updates to the functional specifications for the TMS320VC5421. The advisories are applicable to:

- TMS320VC5421 (144-pin LQFP, PGE suffix)
- TMS320VC5421 (144-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

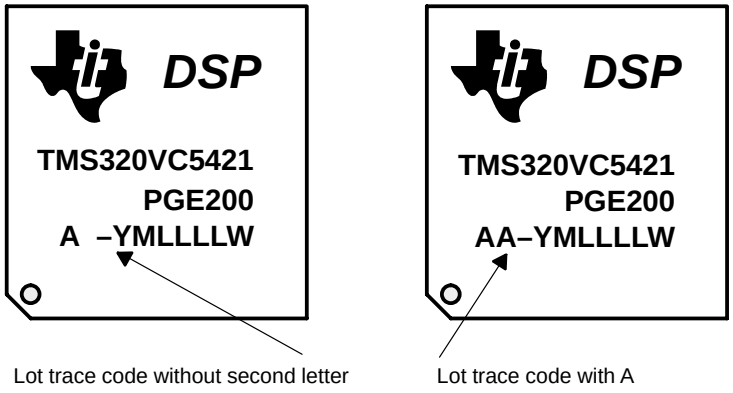
Fully-qualified production device.

MicroStar BGA is a trademark of Texas Instruments.

1.2 Revision Identification

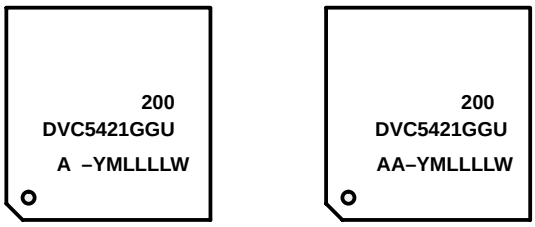
The device revision can be determined by the lot trace code marked on the top of the package. The locations for the lot trace codes for the PGE and the GGU packages are shown in Figure 1 and Figure 2, respectively. The location of other markings may vary per device.

Figure 1. Example, Typical Lot Trace Code for TMS320VC5421 (PGE)



Lot Trace Code	Silicon Revision	Comments
Blank (no second letter in prefix)	Initial Silicon	This silicon revision is available as TMX only.
A (second letter in prefix is A)	Silicon Revision A	This silicon revision is TMS (production).

Figure 2. Example, Typical Lot Trace Code for TMS320VC5421 (GGU)



NOTE: Qualified devices in the PGE package are marked with the letters “TMS” at the beginning of the device name, while nonqualified devices in the PGE package are marked with the letters “TMX” or “TMP” at the beginning of the device name. Similarly, 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
SPI EEPROM Boot Mode	Initial and Revision A Silicon	5
DMPREC	Initial and Revision A Silicon	6
HPI/DMA Doubleword Mode Accesses	Initial and Revision A Silicon	7
Round (RND) Instruction Clears Pending Interrupts	Initial and Revision A Silicon	7
Far Branches/Calls/Interrupts from Active Repeat Blocks (BRAF)	Initial and Revision A Silicon	8
NMI	Initial and Revision A Silicon	8
WRITA/MVDP	Initial and Revision A Silicon	9
HPI Hint	Initial and Revision A Silicon	9
100-MHz Performance	Initial Silicon	10
READY Input Latch Timing for DMA_XIO Accesses	Initial Silicon	10
CPU B XIO	Initial Silicon	10
Shared Memory Access in Emulation Mode	Initial Silicon	11

Advisory *SPI EEPROM Boot Mode*

Revision(s) Affected: Initial Silicon and Revision A Silicon

Details: The SPI EEPROM boot mode does not function.

Workaround: None.

Advisory *DMPREC*

Revision(s) Affected: Initial Silicon and Revision A Silicon

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:

Channel 0 transfer completes and DMA logic clears DE0 internally.

User code attempts to enable another channel (e.g., ORM #2, DMPREC)

The user code will re-enable channel 0 (DMPREC value written = 00C3h), and an additional, unintended transfer will begin on channel 0.

Workaround: There are a few use conditions under which this problem does not occur. If all active DMA channels are configured in ABU mode or in auto initialization 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 re-enabling 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 *HPI/DMA Doubleword Accesses*

Revision(s) Affected: Initial Silicon and Revision A Silicon

Details: HPI accesses can corrupt the doubleword mode bit (DBLW) of an active DMA channel. When a HPI access occurs, it takes priority over the DMA and the active DMA channel transfer is temporarily paused. When the DMA operation continues after the HPI access, the active DMA channel uses the doubleword context of the next channel. If the doubleword context for the next channel is different than the active channel, the DMA transfer will be incorrectly executed (i.e., wrong doubleword mode).

Workaround: To prevent this, all DMA channels should be programmed with the same doubleword mode. That is, all DMA transfers should use either the 16-bit mode or they should all use the 32-bit mode. Note that each DMA channel has a unique DBLW bit in its DMA Sync Event and Frame Count (DMSFCn) register, so each of these registers must be programmed.

Advisory *Round (RND) Instruction Clears Pending Interrupts*

Revision(s) Affected: Initial Silicon and Revision A Silicon

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: Replace the RND instruction with an ADD instruction as follows:

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

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

Revision(s) Affected: Initial Silicon and Revision A Silicon

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(s):

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 bug 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	*NMI*

Revision(s) Affected: Initial Silicon and Revision A Silicon

Details: An NMI can be ignored if the internal CPU interrupt logic is not adequately prepared.

Workaround: Avoid generating an NMI during the time when other interrupts are being serviced. Alternatively, use one of the other external interrupts, appropriately enabled, to serve the NMI function.

Advisory — *WRITA/MVDP*

Revision(s) Affected: Initial Silicon and Revision A Silicon

Details: If a WRITA or MVDP instruction executing from a SARAM block performs a write to any SARAM block that is immediately followed by any read (including DMA or instruction fetch) of the same address that is written to, the read data may be corrupted.

Workaround: Avoid using WRITA/MVDP to write to an area in memory that will be executed as code immediately following the WRITA/MVDP.

Rearrange the code so that a read access does not immediately follow the WRITA/MVDP instruction with an address that is identical to the last address written to. Use a dummy write if necessary.

Avoid DMA reads in an area of program SARAM that is written to by WRITA/MVDP.

Advisory — *HPI Hint*

Revision(s) Affected: Initial Silicon and Revision A Silicon

Details: The HPI will become locked up, with HRDY stuck low, if both the host processor and the 5421 CPU write a one (1) to HINT at the same time.

Workaround: Do not perform 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 *100-MHz Performance*

Revision(s) Affected: Initial Silicon

Details: At the rated core supply voltage of 1.8 V, neither CPU can function reliably at the maximum speed of 100 MHz across the full commercial case temperature range of 0°C to 85°C.

Workaround(s):

1. At the core supply voltage of 1.8 V, reduce the internal CPU clock speed to 80 MHz for functionality across the full commercial case temperature range of 0°C to 85°C.
2. TI has implemented a revised test flow beginning in April of 2000. This flow screens devices for CPU functionality at 100 MHz with a 2.0-V core supply voltage at case temperature of 50°C.

None. This problem is corrected in Revision A silicon.

Advisory *READY Input Latch Timing for DMA_XIO Accesses*

Revision(s) Affected: Initial Silicon

Details: READY is latched on the rising edge of CLKOUT (instead of the falling edge of CLKOUT).

Workaround: None. This problem is corrected in Revision A silicon.

Advisory *CPU B XIO*

Revision(s) Affected: Initial Silicon

Details: CPU B loses control of XIO when the device is put in HOLD.

Workaround: None. This problem is corrected in Revision A silicon.

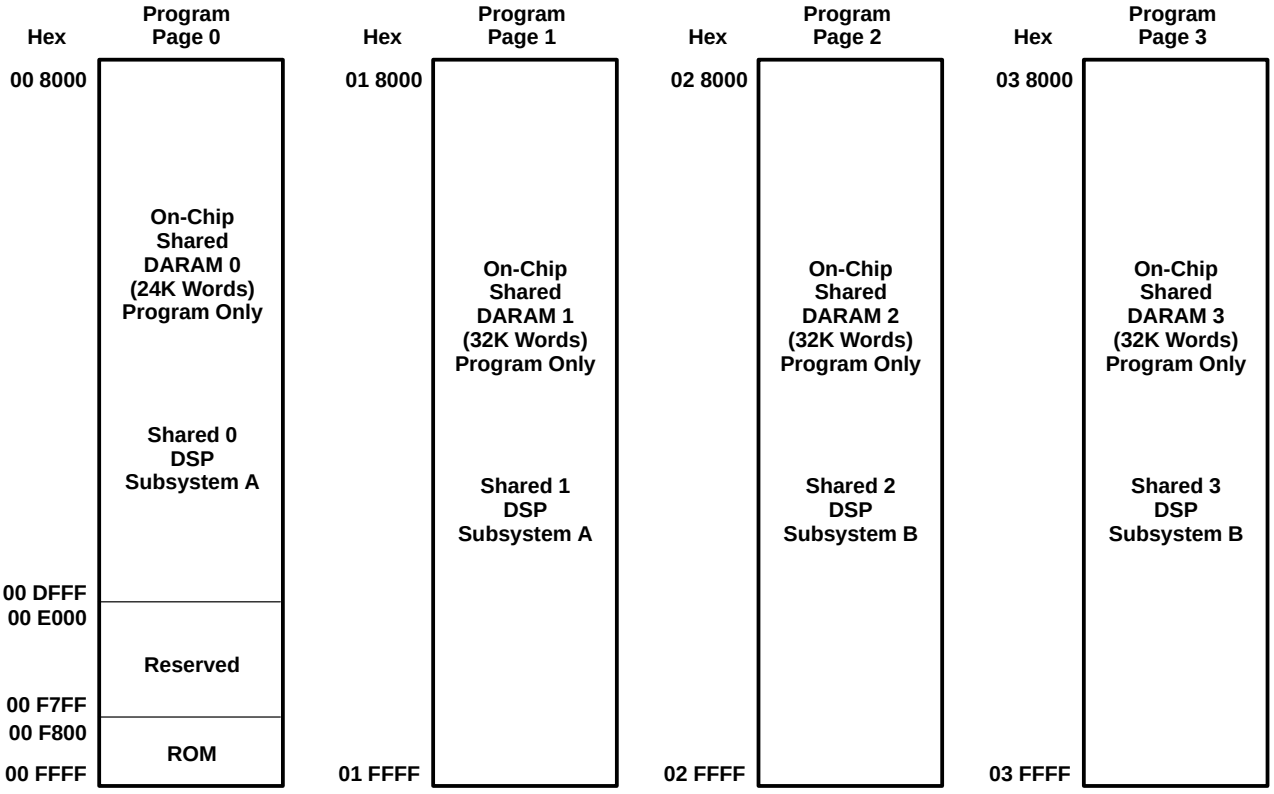
Advisory *Shared Memory Access in Emulation Mode*

Revision(s) Affected: Initial Silicon

Details: When a 5421 subsystem is halted in emulation mode, the other subsystem is unable to access the shared memory associated with the halted subsystem (see Figure 3 for the shared-memory map). This also affects the use of software and hardware breakpoints in the shared memory. Note that this issue only affects the TMX320VC5421 during emulation, and there are no issues with shared memory accesses during normal functional mode.

Workaround: None. This problem is corrected in Revision A silicon.

Figure 3. Shared-Memory Map



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. Open the "**Products**" dialog box and select "**Digital Signal Processors**"
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 TMS320VC5421, please refer to:

- *TMS320VC5421 Fixed-Point Digital Signal Processor* data sheet*,* literature number SPRS098
- *TMS320C54™ 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.

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 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.

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.

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265