TMS320C6A816x Integra DSP+ARM Processors Revision 1.0

Silicon Errata

Literature Number: SPRZ328
November 2010

List of Figures

List of Tables

C6A816x Integra DSP+ARM Processors Revision 1.0

1 Introduction

This document describes the known exceptions to the functional specifications for the C6A816x Integra™ DSP+ ARM® Processors. The updates are applicable to the CYG package. For additional information, see the *TMS320C6A816x Integra DSP+ARM Processors* data manual (literature number SPRS680).

The advisory numbers in this document may not be sequential. Some advisory numbers may be moved to the next revision and others may have been removed and documented in the device-specific data manual or peripheral user's guide. When items are moved or deleted, the remaining numbers remain the same and are not resequenced.

This document also contains Usage Notes. Usage Notes highlight and describe particular situations where the device's behavior may not match presumed or documented behavior. This may include behaviors that affect device performance or functional correctness. These notes will be incorporated into future documentation updates for the device (such as the device-specific data manual), and the behaviors they describe will not be altered in future device revisions.

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 DSP devices and support tools. Each DSP commercial family member has one of three prefixes: X, P, or TMS (e.g., (e.g., **X**320C6A8168CYG)). 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 (X/TMDX) through fully qualified production devices/tools (TMS/TMDS).

Device development evolutionary flow:

X — Experimental device that is not necessarily representative of the final device's electrical specifications and may not use production assembly flow.

P — Prototype device that is not necessarily the final silicon die and may not necessarily meet final electrical specifications.

TMS — Production version of the silicon die that is fully qualified.

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.

X and P devices and TMDX development-support tools are shipped against the following disclaimer:

"Developmental product is intended for internal evaluation purposes."

Production 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 (X or P) 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.

1.2 Package Symbolization and Revision Identification

Figure 1 shows an example of the C6A816x processor package symbolization. The device revision can be identified by the markings on the top of the CYG package; the blank space between the device number and the package identifier indicates the initial device revision (1.0), as noted in Figure 1.

Table 1 lists the device revisions for the C6A816x processor.

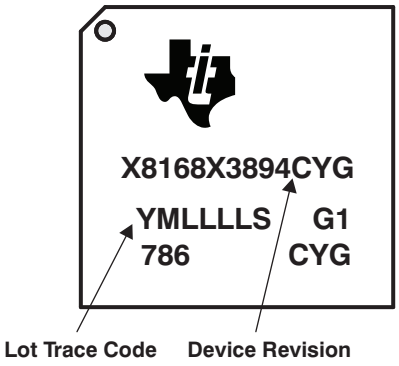


Figure 1. Package Symbolization

Table 1. Identifying Device Revision From Package Symbolization

DEVICE REVISION IDENTIFIER	DEVICE REVISION	COMMENTS
Blank	1.0	Initial device revision

2 Revision 1.0 Usage Notes and Known Design Exceptions to Functional Specifications

2.1 Usage Notes for Revision 1.0

Usage Notes highlight and describe particular situations where the device's behavior may not match presumed or documented behavior. This may include behaviors that affect device performance or functional correctness. These notes will be incorporated into future documentation updates for the device (such as the device-specific data manual), and the behaviors they describe will not be altered in future device revisions.

TBD

2.2 Revision 1.0 Known Design Exceptions to Functional Specifications

Table 2. Revision 1.0 Advisory List

Advisory 1.0.1 *Error Interrupt for PCIe EP Mode Not Generated*

Revisions Affected: 1.0

Details: PCI Express® (PCIe®) errors observed on the device configured as endpoint (EP) do not result in interrupts to the local host (A8). As per the PCIe specification, a non-posted PCIe request from the initiator, which is interpreted as an unsupported request at the completer, results in an error indication being sent to the initiator. Currently, the device in EP mode does not generate an interrupt to the local host (A8) on receiving such an error. The same behavior is applicable for completer abort (CA) errors. This issue only impacts non-posted transactions. Memory writes do not have this issue since they are always posted.

Workaround: Software running on EP should check the error status bit after completing every request that would be initiated as a non-posted request over the PCIe link. This is particularly applicable to all PCIe reads initiated from A8 on EP.

To be specific, all reads initiated from A8 on EP over PCIe need to be followed by error status checks to ensure the sanity of read data. Note that this is not continuous polling and the error status should be checked only at the end of transaction. This should not impact performance in data transfer scenarios where EDMA is used to initiate transfers and the software running on EP relies on DMA completion interrupts.

Advisory 1.0.2 *VIP Parser Output FIFO Overflow Bit is Set/VIP Port Locks Up When Capture is Run With Other Drivers (e.g., SC, Display)*

Revisions Affected: 1.0

Details: When running capture with other HDVPSS drivers such as SC and Display, the VIP parser port sometimes gets locked or hung up and the output FIFO overflow bit in the VIP parser gets set (in the VINx_PARSER FIQ Status register).

This condition occurs when the VPDMA inbound and outbound descriptor priority for Capture, SC, or Display and other drivers is set to 0 (highest priority). Due to this issue, capture data does not get written to DDR memory because the capture VPDMA stalls and causes the output FIFO to become full. Once the output FIFO becomes full, the VPDMA drops the request to the VIP parser and causes the VIP port to lock up.

Workaround: In the VPDMA data descriptor, set the highest priority for capture, followed by display, and then memory-to-memory (M2M) drivers such as SC, DEI, and NSF. For example:

1. Set Capture VPDMA descriptors to priority 0 (highest).
2. Set Display VPDMA descriptors to priority 1.
3. Set M2M VPDMA descriptors to priority 7 (lowest).

With the above priority settings, the output FIFO overflow condition does not occur.

Advisory 1.0.3 *Spurious Descriptors are Output for Capture When No Descriptors are Programmed to VPDMA for Capture*

Revisions Affected: 1.0

Details:

The Capture driver works as follows:

- A list is programmed with two data descriptors for every capture channel.
- When a frame for given channel is captured, the frame data goes to the buffer address programmed in the descriptor and the programmed descriptor is written back to a specific memory location.
- The software then checks the specific memory location and, for every that is written in this memory location, it re-programs a new data descriptor.

When software does not program a descriptor, the expectation is that any new frame data arriving for that channel should be dropped **and** when software does not program a descriptor, there should be no descriptor output to the memory.

However when the software does not program a descriptor for a channel, the new frame data gets dropped since there is no descriptor and, therefore, no buffer address to output this data. But, the VPDMA generates a *spurious* descriptor and writes it to memory.

The spurious descriptor that gets written out has the following properties:

- The width x height reported in the descriptor is 1x1.
- The buffer address in the descriptor is same as the last received descriptor.

Workaround:

1. Ensure that spurious descriptors do not occur; i.e., make sure a capture VPDMA channel is never starved of descriptors. A capture VPDMA channel can be starved of descriptors in the following scenarios:
   - When the CPU is halted due to breakpoints.
   - When the CPU is halted due to a Code Composer Studio (CCStudio) printf.
   - When a high-priority task or ISR continuously consumes CPU MHz for more than 1 frame or field interval.
2. Thus, once a capture is started, the software should not perform any CCStudio printf (the software can perform a UART printf since the UART printf does not block the CPU).
   - Do not set any breakpoints.
   - The capture driver thread should be the highest priority.
3. If, after the above, spurious descriptors (that meet the above-described properties; i.e., 1x1 size and buffer address equals previously received descriptor buffer address), do not re-submit a new descriptor for this received descriptor since it is a spurious descriptor.

Advisory 1.0.4 *Clear Descriptor Count in VPDMA Descriptor Status Control Register Does Not Perform as Expected*

Revisions Affected: 1.0

Details:

The Capture driver working mechanism is briefly explained in Advisory 1.0.3. In addition:

- Every few milliseconds, the capture driver software:
  - Examines this memory location for received descriptors.
  - Replaces the memory location with new descriptors.
  - Re-submits the new list of descriptors to the VPDMA.
- While capture is checking the received descriptors, new descriptors could be generated and, in order to make sure no descriptors output by VPDMA are missed, the capture driver sets the clear descriptor count bit in the VPDMA Descriptor Status Control register.
- This bit performs the following:
  - Switches the descriptor writing to a new memory location.
  - Tells the software, via the last descriptors written field in the Descriptor Status Control register, how many descriptors were received at the instant the clear descriptor count bit was set.

However, when the clear descriptor count bit is set, sometimes some descriptors were never written to memory. Due to this issue, the software does not replace those descriptors causing frame-rate drops for those channels.

Workaround:

- Do not use the clear descriptor count bit. Instead, use the Current Descriptor register in the VPDMA to keep track of how many descriptors were written since last check.
- The Descriptors Top and Descriptors Bottom registers are programmed to write the received descriptors in a circular manner in the buffer that starts at Descriptors Top and ends at Descriptors Bottom.
- The descriptor received at any given time is the *current descriptor* (the value of the current descriptor in the previous check). The software takes care to handle wraparound when reading descriptors in a circular manner.

Advisory 1.0.5 *Descriptor With Erroneous Frame Size 1x1 Written by VPDMA When a Descriptor is Posted For the Corresponding Channel For Capture*

Revisions Affected: 1.0

Details: The Capture driver working mechanism is briefly explained in Advisory 1.0.3.

Sometimes, the VPDMA outputs descriptors that report the incorrect frame size 1x1 even though the descriptors were programmed to the VPDMA. Here the frame data, itself, is written to the correct programmed memory address.

The erroneous descriptor that gets written out has the following properties:

- The width x height reported in the descriptor is 1x1.
- The buffer address in the descriptor is the correct buffer address for the current frame.

Workaround: Detect this condition based on the above described properties of the descriptor and replace the received descriptors as usual.

Advisory 1.0.6 *Camera Removal and Reconnection Causes VIP Parser to Lock Up*

Revisions Affected: 1.0

Details:

If the camera is removed and reconnected while capture is running, then the VIP parser locks up and causes the capture driver to hang up.

This issue occurs because when the camera is removed or reconnected, the F-bit and V-bit transitions are not consistent. For example, the V-bit transitions to a new frame every few lines and, thus, causes many short frames to be received at the VIP port which causes the VIP port to lock.

In the multichannel case, the internal hardware FIFO that is used for capture is common for all channels at a given port. So if one channel is removed or reconnected, the whole FIFO at the VIP port locks up and causes all channels at that port to lock up.

Workaround:

For multichannel case, with TVP5158:

- Operate TVP5158 in pixel-mux mode.
- Load TVP5158 firmware patch v2.1.21 or above.
- Disable fast lock detect:
  - TVP5158 REG_FV_CTRL = 0x06.
- Disable auto-switch mode; i.e., be in force NTSC or force PAL mode:
  - TVP5158 REG_VID_STD_SELECT = 0x1 (NTSC) or 0x2 (PAL).
- Always decode F and V bits from line count (TVP5146 compatible):
  - TVP5158 REG_FV_DEC_CTRL = 0x03.
- TVP5158 REG_OP_MODE_CTRL = 0.

The above workaround does ***not*** work TVP5158 in line mux mode and is ***not*** guaranteed to work with non-TVP5158 multichannel video decoders.

For single-channel case:

- The software should detect cable removal by querying the external video decoder.
- When sync loss is detected, the VIP port should be stopped.
- When the cable is reconnected, the software should:
  - Perform a VIP reset sequence, as mentioned in Advisory 1.0.8.
  - Start the VIP port.

Advisory 1.0.7 *CHR_DS Hangs When it Receives a Frame With an Odd Number of Lines*

Revisions Affected: 1.0

Details: When CHR_DS receives a frame with an odd number of lines and then an EOF signal, it hangs up waiting for the even numbered line to arrive.

When using CHR_DS in memory-to-memory mode this is not an issue, since user software can ensure that CHR_DS always receives an even number of lines. However, when CHR_DS is used in capture path, this cannot be guaranteed.

Workaround: To resolve this issue:

1. Ensure that the external video decoder always gives even number of lines.
2. If Step 1 cannot be guaranteed, then use the SC to trim the input to an even number of lines and then send the lines to CHR_DS.
3. If Steps 1 and 2 cannot be performed (because SC is used in some other path), then do not use CHR_DS and use either YUV422S or YUV422I output mode.

If CHR_DS locks up, you can recover it by using the VIP reset sequence mentioned in Advisory 1.0.8.

This issue could also occur when the cable is disconnected. In this situation, refer to the workaround in Advisory 1.0.6 for single-channel case.

Advisory 1.0.8 *VPDMA List Stalls and Hangs When VIP Port is Re-enabled After Disabling With CHR_DS and/or SC and/or CSC Enabled in Non-Mux Mode*

Revisions Affected: 1.0

Details:

When capturing in non-mux mode with CHR_DS and/or SC and/or CSC enabled in the capture path, the capture runs correctly for the first run.

If capture is stopped by disabling the VIP port and then capture is started again with CHR_DS and/or SC and/or CSC in the path, then the VPDMA list for capture stalls and hangs up, causing the capture driver to hang up as well.

This is applicable for both embedded-sync mode as well as discrete-sync mode, except that in discrete-sync mode the hang occurs on the first run.

Workaround:

For 8-/16-bit, YUV422 embedded-sync mode:

1. Disable the VIP port.
2. Assert VIP reset in the HDVPSS CLKC register.
3. Assert the Async FIFO reset in the VIP parser register.
4. Make a list of abort descriptors for all VIP VPDMA channels.
5. Post this list and wait for it to complete.
6. De-assert the VIP reset in the HDVPSS CLKC register.
7. De-assert the Async FIFO reset in the VIP parser register.

For 8/16/24-bit discrete-sync mode, in addition to the above workaround, the following additional sequence of descriptor programming **must** be performed in order to use discrete-sync mode with CSC, SC, and CHR_DS. For the first list that is posted when discrete-sync mode is started, perform the following:

1. To the list, add a data descriptor such that video data from the VIP parser goes directly to VPDMA, with DROP_DATA = 1, WRITE_DESC = 0 in the data descriptor (e.g., PORTB_RGB VPDMA channel).
2. To the list, add a sync-on-client descriptor with the VPDMA channel as the one set in the above data descriptor (e.g., PORTB_RGB VPDMA channel and event = EOF).
3. To the list, add a configuration descriptor with the actual required VIP mux settings which enables CHR_DS, CSC, and SC.
4. To the list, add the actual data descriptors as before (e.g., YUV420 output data descriptors).
5. Program the VIP mux such that data from the VIP parser goes directly to VPDMA.
6. Enable the VIP port so that data starts flowing.
7. Post the list that is prepared above.
8. The list stalls until it receives an end of frame. In the blanking period, the configuration descriptor programs the mux to use the CSC, SC, and CHR_DS and then capture continues as usual.

Advisory 1.0.9 *Noise Filter (NSF) Outputting Frames of 64x32 Though Input Data is of Larger Size*

Revisions Affected: 1.0

Details: When the noise filter (NSF) is used along with capture and display after a few frames of execution, the NSF starts outputting frames of size 64x32, even though the input size is larger.

Workaround: Use the noise filter in external bypass mode.

This issue will be fixed in revision 1.1.

Advisory 1.0.10 *Discrete Sync Capture Style Line ACTVID Does Not Work*

Revisions Affected: 1.0

Details: The VIP port supports a discrete sync capture mode where the ACTVID signal is used to determine the active region of the video and then the VPDMA writes this active video frame to memory. However, with some external video decoders, this mode does not work by default.

Workaround: The VIP parser requires that the ACTVID signal is active for all lines of the incoming data, including the vertical blanking region. The vertical blanking region, itself, is not written to memory by the VPDMA. Figure 2 illustrates this.

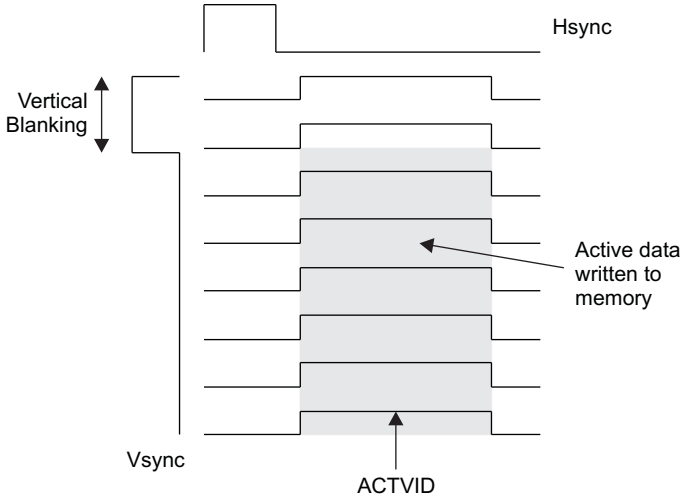


Figure 2. ACTVID Signal Active for All Lines of Incoming Data

Advisory 1.0.11 *In Single-Channel Capture No Descriptor May be Output Even Though a Descriptor was Previously Programmed*

Revisions Affected: 1.0

Details: The Capture driver working mechanism is briefly explained in Advisory 1.0.3.

In single-channel non-mux mode, sometimes, even though a descriptor is programmed, it is not output to memory even though the frame itself is captured. Due to this issue, the software does not reprogram a descriptor for that VIP port and the capture frame rate drops.

Workaround: Program a dummy descriptor on one of the VPDMA MULT_SRCx channels for the port, even though no data is received on that channel. This always causes the VPDMA to output a descriptor to memory for the channel on which data is being received. In this case, Advisory 1.0.5 is also applicable.

Advisory 1.0.12 *Enabling Compression in DEIH and/or DEI Paths Results in List Hang*

Revisions Affected: 1.0

Details: When the compression and decompression blocks are enabled for the DEIH and DEI paths, the VPDMA list hangs after a random duration. This issue is not root caused.

Workaround: Do not enable compression.

Advisory 1.0.13 *Dynamic Mosaic Layout Change From 3x3 Frame to 3x4 Frame Results in Display Hang*

Revisions Affected: 1.0

Details: When the VPDMA stalls on an actual channel and then the previous frame actual channel completes, the client misses the next channel descriptor given by the list manager. In this event, the list stalls.

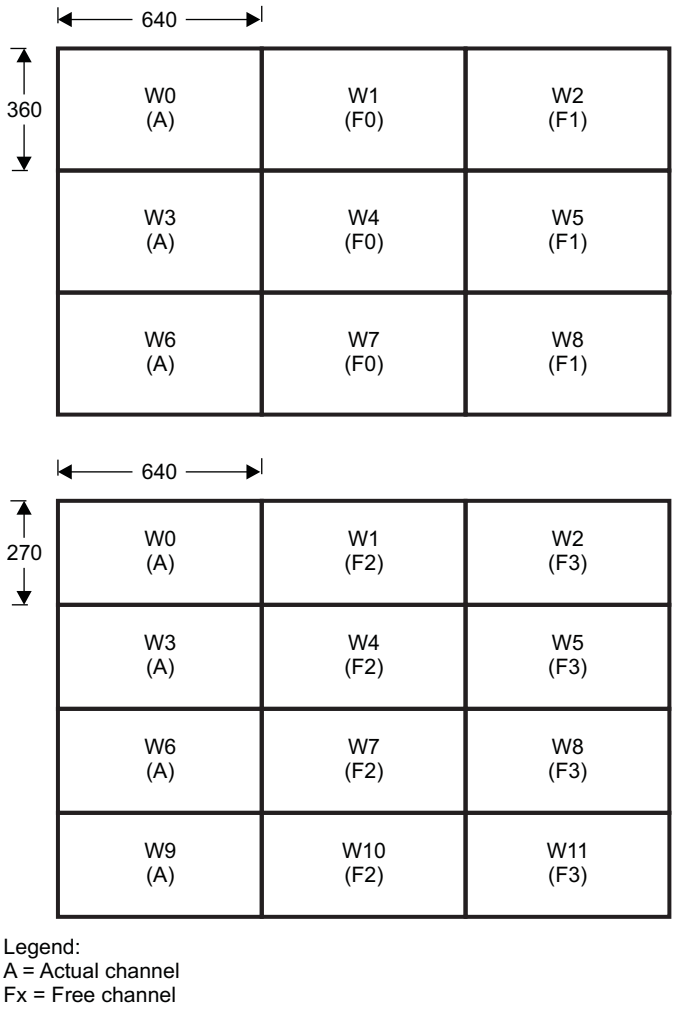
This issue affects the following scenario:

- Changing from display layout 3x3 (each window size is 640x360) to 3x4 (each window size is 640x270) while the display is going on.
- The VPDMA free channels used in 3x3 and 3x4 layouts are different.
- Other settings assumed:
  - Display resolution: 1920x1080
  - Display path: bypass path 0/1.

Layout Description

In the case of 3x3-to-3x4 layout change, the free channels used for the two layouts are different. Therefore, when the layout changes the VPDMA does not block on the free channels (F2 and F3) of the 3x4 descriptors because they are not used earlier. Then, the VPDMA stalls on the actual channel (A) for window W0. At the same time, the last row of 3x3 descriptors is being transferred. When the last line of the first window in the last row (W6) completes, the actual channel (A) is unblocked. But, the client is busy transferring the last line of the remaining windows (W7 and W8). This causes the above mentioned condition, which results in a list hang and the display starts displaying the COMP background color.

This hang does not happen for layouts if there is a dummy descriptor for that layout because it stalls the VPDMA list until the last pixel of the frame. Dummy descriptors are used when the last window in a frame does not end with the frame's bottom-right boundary (see Figure 3 and Figure 4).



Legend:
A = Actual channel
Fx = Free channel

Figure 3. Descriptor Layout

3x3 Layout
W2 (F1)
W1 (F0)
W0 (A)
W5 (F1)
W4 (F0)
W3 (A)
W8 (F1)
W7 (F0)
W6 (A)

3x4 Layout
W2 (F3)
W1 (F2)
W0 (A)
W5 (F3)
W4 (F2)
W3 (A)
W8 (F3)
W7 (F2)
W6 (A)
W11 (F3)
W10 (F2)
W9 (A)

Figure 4. Descriptor Layout in Memory

Workaround: Put an SOCH descriptor on the free channel of the last window before switching to the next frame (see Figure 5). This need not be there if a dummy descriptor is present or if there is no free channel used for the last row.

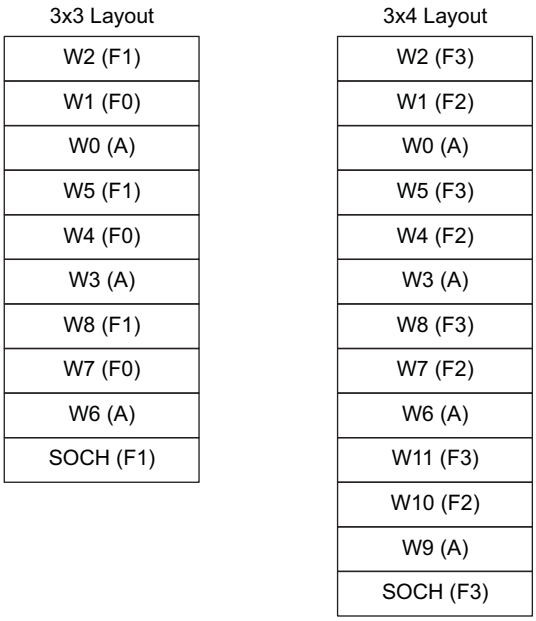


Figure 5. Descriptor Layout in Memory With SOCH Descriptor

Advisory 1.0.14 *When using HDMI Port, HD_VENC_D_CLK Needs to be Doubled for Normal Operation*

Revisions Affected: 1.0

Details: HD VENC_D (DVO1) and HDMI should operate with the same pixel clock frequency since HD VENC_D supplies the video and control signals for HDMI.

The pixel clock selector (pixel clock for HDMI and HD VENC_D 1X clock), HD_VENC_D_CLK1X_SELECT, always selects HD_VENC_D_CLK / 2. This ensures that HD_VENC_D_CLK is always divided by 2; e.g., if the desired display is a resolution of 720p60, the HD_VENC_D_CLK would be 74.25 MHz. Since the pixel clock selector always selects HD_VENC_D_CLK / 2, the HDMI and HD VENC_D pixel clock would be 37.125 MHz.

Workaround: Clock HD_VENC_D_CLK at twice the required pixel clock and configure HD VENC_D to use 1X input; e.g., in the above example HD_VENC_D_CLK should be clocked at 148.5 MHz.

HDMI and HD VENC_D have been verified for HD_VENC_D_CLK input up to 297 MHz for 1080p60 resolution.

Advisory 1.0.15 *USB Host Mode Cannot Perform Write Operation to TxFunction/TxHubPort/TxHubAdr Controller Addresses*

Revisions Affected: 1.0

Details: The USB controller does not support devices connected through a hub when operating as USB host. Due to this, writing to the TxHubPort/TxHubAdr results in the write being ignored.

Workaround: There is no workaround for this issue.

Advisory 1.0.16 *Spurious RX_SOP_STARVATION Interrupt on the First CPPI DMA Rx Descriptor Following USB Module Reset*

Revisions Affected: 1.0

Details: A spurious RX_SOP_STARVATION interrupt (bit 0 of IRQ_STATUS_RAW) occurs on the first CPPI DMA Rx descriptor fetch following a USB module reset. This issue is not dependent on the free queue number used or the RX DMA channel number. The spurious interrupt occurs only after the first CPPI DMA Rx descriptor fetch and is not repeated.

Workaround: The software should ignore only the first RX_SOP_STARVATION interrupt.

Advisory 1.0.17 *GPMC Uses Bad Generator Polynomial in t=4 BCH Mode (t - number of correctable errors)*

Revisions Affected: 1.0

Details: In mode t = 4, GPMC uses the wrong generator polynomial (0x14523043AB86A9) instead of a *good* generator polynomial (0x14523043AB86AB), where bit 1 is incorrect. This results in the following:

- On page write, it generates incorrect ECC parity.
- On page read, it generates an incorrect syndrome.

Workaround: There is no workaround for this issue. It is recommended to use the NAND flashes that need 8-bit or 16-bit ECC.

Advisory 1.0.18 *CPGMAC 1Gbps Mode Does Not Work When GMII_MT_CLK is Not Running*

Revisions Affected: 1.0

Details: Although GMII_MT_CLK is specific to the 10/100Mbps clock, if it is not running, then the 1Gbps mode does not work.

Workaround: Use the PHY chip that outputs the transmit clock to MAC (e.g., ET1011C PHY).

Advisory 1.0.19 *Kernel Crashes in Software While Accessing UART RHR Register*

Revisions Affected: 1.0

Details: In the software while accessing the UART receive holding register (RHR) register, the kernel crashes. If the Rx FIFO is empty before a UART RHR read and the Tx FIFO is full before a UART THR write, an error response results.

Workaround: The software should ensure that the Rx FIFO is not empty before reading RHR and the Tx FIFO not full before writing to THR.

The RHR read on receive FIFO empty can be avoided as follows:

1. Read the UART Line Statue Register (LSR).
2. Check to see if the Receiver Data Ready (DR) bit (bit 0) is set to 1.
3. If yes, proceed to read RHR, otherwise there is no data to be read and the RHR read should be skipped.

The THR write on transmit FIFO full can be avoided as follows:

1. Read the UART Line Statue Register (LSR).
2. Check to see if the Transmit Hold Register Empty (THRE) bit (bit 5) is set to 1.
3. If yes, proceed to write to THR, otherwise loop until THRE is set before writing to THR.
4. It may be desirable to implement a few milliseconds timeout for the loop to poll to avoid long busy waits for THR.

Advisory 1.0.20 *DDR3 Does Not Function*

Revisions Affected: 1.0

Details: DDR3 cannot be used.

Workaround: There is no workaround for this issue; we recommend using DDR2 until the DDR3 issue is fixed in a future revision.

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
DLP® Products	www.dlp.com	Communications and Telecom	www.ti.com/communications
DSP	dsp.ti.com	Computers and Peripherals	www.ti.com/computers
Clocks and Timers	www.ti.com/clocks	Consumer Electronics	www.ti.com/consumer-apps
Interface	interface.ti.com	Energy	www.ti.com/energy
Logic	logic.ti.com	Industrial	www.ti.com/industrial
Power Mgmt	power.ti.com	Medical	www.ti.com/medical
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Space, Avionics & Defense	www.ti.com/space-avionics-defense
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Video and Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless-apps

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