Application Report
SPRAC16A–January 2016–Revised January 2018

DSS Bit Exact Output

Prasad Konnur, Sivaraj R, Brijesh Jadav

ABSTRACT

The display subsystem (DSS) in TDA2xx, TDA2Ex and TDA3xx platform is used for displaying video data to external devices. But DSS can also be used as a high-speed data transfer port for any non video data. One of the basic requirements of the data transfer port is that the data that is present in the memory should not be modified before the data is sent out, which means the data sent should be bit exact with the data in memory.

This application report explains how to use the DSS as a data transfer port.

Contents

List of Figures

List of Tables

1 Display Subsystem (DSS) Overview

The DSS provides the logic to interface display peripherals. DSS integrates a DMA engine as part of display controller (DISPC) module, which allows direct access to the memory frame buffer. Various pixel processing capabilities are supported, such as: color space conversion, filtering, scaling, blending, color keying, and so forth.

The display controller mainly consists of DMA Engine, Video Pipelines (VID/GFX), Overlay Managers (OVLY), and Video Ports (VP). Figure 1 and Figure 2 show the different processing blocks present in the Video Pipeline and the Overlay, respectively. The DMA Engine supplies data from system memory to Video Pipeline. The Video Pipeline processes the data coming from the DMA Engine and outputs to the Overlay. The Overlay can manage/blend output from multiple pipelines, after processing sends out on the video port which is a 24-bit parallel video data interface.

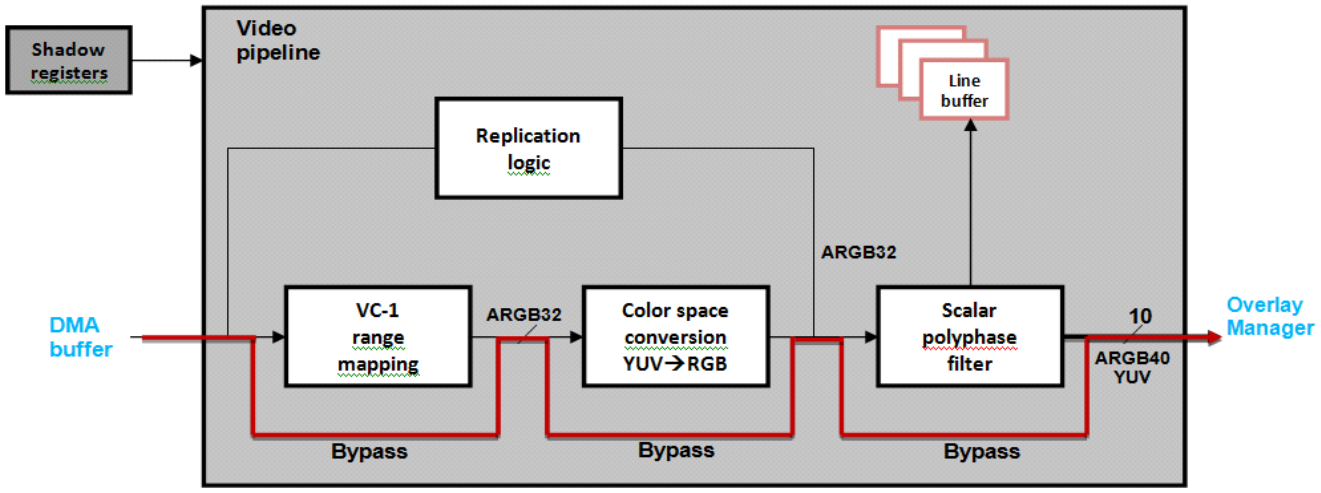

Figure 1. DSS Video Pipeline Block Diagram

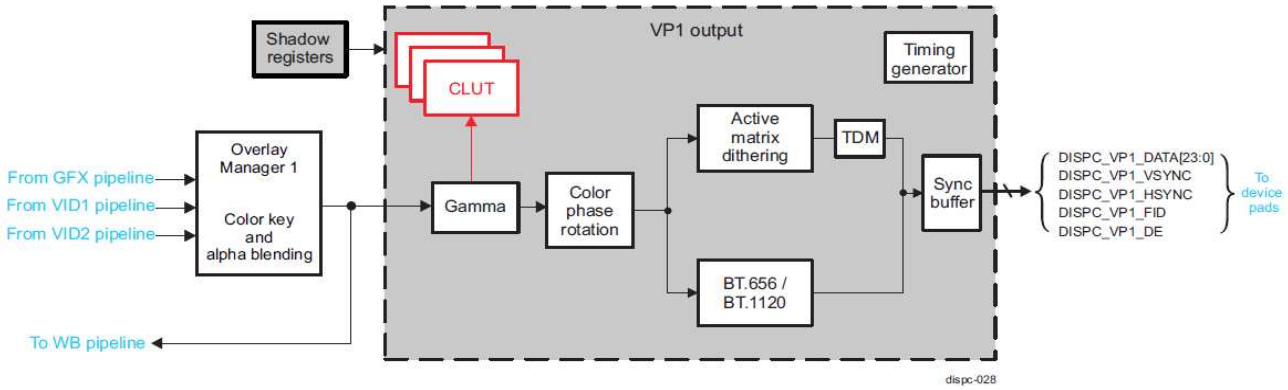

Figure 2. DSS Overlay and Video Port Block Diagram

2 DSS Bit Match Output

The DSS can output data in 24-bit, 16-bit or 8-bit depending on the external device connected to the video port. This section explains the configuration to get bit exact output from DSS video port for the different interface sizes.

2.1 Bit Matching: 24-bit Output Interface

This section describes the configuration when sending the bit matched output when 24 data lines of the VP are used for data transfer.

2.1.1 Video Pipe Settings

Each of the processing blocks present in the video pipelines has a bypass path (see Figure 1), which can be selected to send the data unaltered by the video pipeline. Input data format of the video pipeline should be configured as RGB24-888. Also input line offset/pitch requires to be 3-byte aligned. VC-1 range mapping and CSC blocks of the video pipelines (see Figure 1) will be bypassed by default for this format. Input and output width and height of the video pipeline should be configured to same values to bypass the scaling block.

With the above configuration the data coming to video pipeline from DMA engine will reach the overlay unaltered.

2.1.1.1 Register Configuration

TDA2xx/TDA2Ex:

`DISPC_VID1_ATTRIBUTES.FORMAT = 0x9;`	RGB24-888 (24-bit container)
`DISPC_VID1_ATTRIBUTES.RESIZEENABLE = 0x0;`	Disable both horizontal and vertical resize processing

Also make sure the conditions below are set.

```
DISPC_VID1_SIZE.SIZEX == DISPC_VID1_PICTURE_SIZE.MEMSIZEX
DISPC_VID1_SIZE.SIZEY == DISPC_VID1_PICTURE_SIZE.MEMSIZEY
```

TDA3xx:

`DISPC_VID_ATTRIBUTES.FORMAT = 0xB;`	RGB24-888 (24-bit container)
`DISPC_VID_ATTRIBUTES.RESIZEENABLE = 0x0;`	Disable both horizontal and vertical resize processing

Also make sure the conditions below are set.

```
DISPC_VID_SIZE.SIZEX == DISPC_VID_PICTURE_SIZE.MEMSIZEX
DISPC_VID_SIZE.SIZEY == DISPC_VID_PICTURE_SIZE.MEMSIZEY
```

Driver Programming:

In the `IOCTL_VPS_DISP_SET_DSS_PARAMS`, ioctl pass the parameter `Vps_DispDssParams` with the settings below.

```
Vps_DispDssParams.inFmt.dataFormat = FVID2_DF_BGR24_888;
Vps_DispDssParams.vidCfg->pipeCfg.scEnable = FALSE;
```

Also make sure the conditions below are set.

```
Vps_DispDssParams.inFmt.width == Vps_DispDssParams.tarWidth;
Vps_DispDssParams.inFmt.height == Vps_DispDssParams.tarHeight;
```

2.1.2 Overlay Settings

Each of the processing blocks present in the Overlay and Video Port (see Figure 2) can be bypassed to send the data unaltered.

There are two types of alpha blending supported in the overlay: pixel level and global. Global alpha blending is supported only when more than one pipeline is enabled. Enable only one video pipeline for bit matching. Pixel level alpha blending is supported when input of video pipe is of ARGB data formats. As mentioned in Section 2.1.1, configure data format to RGB24-888 to disable pixel level alpha blending.

The color phase rotation (CPR) block is used for converting the RGB output to YUV (in case of BT656 formats). CPR can be disabled using display controller configuration register. The time division multiplexing (TDM) block can also be disabled using display controller control register. The driver disables the TDM by default unless it is enabled explicitly using the `IOCTL_VPS_DCTRL_DSS_SET_ADV_VENC_TDM_PARAMS` IOCTL with `tdmEnable` variable set to `TRUE`.

Register Configuration:

```
DISPC_CONFIG1.TCKLCDENABLE = 0;     Disable the transparency color key for the LCD
DISPC_CONFIG1.CPR = 0;              Color Phase Rotation Disabled
DISPC_CONTROL1.TDMENABLE = 0;       TDM disabled
```

Driver Programming:

In the `IOCTL_VPS_DCTRL_SET_OVLY_PARAMS`, IOCTL passes the parameter `Vps_DssDispcOvlyPanelConfig` with the setting below to disable color keying.

```
Vps_DssDispcOvlyPanelConfig.transColorKey = 0;
```

In the `IOCTL_VPS_DCTRL_SET_VENC_OUTPUT`, ioctl passes the parameter `Vps_DctrlOutputInfo` with the setting below for disabling CPR.

```
DISPC_CONFIG1.TCKLCDENABLE = 0;
Vps_DctrlOutputInfo.dataFormat = FVID2_DF_RGB24_888;
Vps_DctrlOutputInfo.dvoFormat = VPS_DCTRL_DVOFMT_GENERIC_DISCSYNC;
```

In the `IOCTL_VPS_DCTRL_DSS_SET_ADV_VENC_TDM_PARAMS` IOCTL passes the parameter `Vps_DssDispcAdvLcdTdmConfig` with the setting below to disable the TDM.

```
Vps_DssDispcAdvLcdTdmConfig.tdmEnable = 0;
```

Note that DSS outputs blue color component on D[7:0] data lines, green color component on D[15:8] data lines and red color component on D[23:16] data lines for 24-bit output interface. Also DSS expects memory organization for RGB888 packed input as shown in Figure 3.

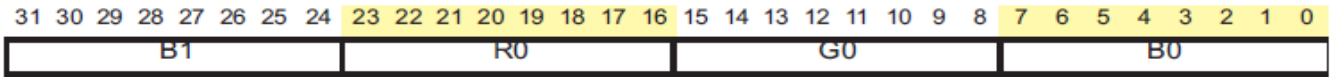

Figure 3. Memory Organization for RGB888 Packed Input Format

If different color ordering is required on the output lines, CPR (color phase rotation) module can be used. CPR module is a 3x3 matrix, which is typically used in the RGB to YUV format conversion, but it can also be enabled to reorder the color components to get the expected color components on the data lines.

For example, in order to swap the Red and Blue color component on the output data lines, configure the CPR module as shown below and enable it:

Register Configuration:

```
DISPC_CPR1_COEF_R.RR = 0;
DISPC_CPR1_COEF_R.RG = 0;
DISPC_CPR1_COEF_R.RB = 256;
DISPC_CPR1_COEF_G.GR = 0;
DISPC_CPR1_COEF_G.GG = 256;
DISPC_CPR1_COEF_G.GB = 0;
DISPC_CPR1_COEF_B.BR = 256;
DISPC_CPR1_COEF_B.BG = 0;
DISPC_CPR1_COEF_B.BB = 0;
DISPC_CONFIG1.CPR = 1;
```

Driver Configuration:

In the `IOCTL_VPS_DCTRL_SET_CPR_COEFF` `ioctl` pass, configure the parameter `Vps_DssVencCprCoeff` with the following settings:

```
Vps_DssVencCprCoeff.rr = 0;
Vps_DssVencCprCoeff.rg = 0;
Vps_DssVencCprCoeff.rb = 256;
Vps_DssVencCprCoeff.gr = 0;
Vps_DssVencCprCoeff.gg = 256;
Vps_DssVencCprCoeff.gb = 0;
Vps_DssVencCprCoeff.br = 256;
Vps_DssVencCprCoeff.bg = 0;
Vps_DssVencCprCoeff.bb = 0;
Vps_DssVencCprCoeff.enableCpr = TRUE;
Vps_DssVencCprCoeff.enableCsc = FALSE;
Vps_DssVencCprCoeff.vencId = VPS_DCTRL_DSS_VENC_LCD1;
```

2.2 Bit Matching: 16-Bit Output Interface

This section describes the configuration when sending the bit matched output when 16 data lines of the VP are used for data transfer.

DSS supports RGB16-565 16-bit data format and the DISPC also can output data in RGB16-565 format. So if input and output is configured as RGB16-565 format, you can get bit-matched data at the output of DSS.

Overlay in DSS works on ARGB48 data format, so the data coming from pipelines should be converted to ARGB48 data format. Table 1 shows the replication logic used to convert from RGB16-565 to ARGB48 data format.

Table 1. DSS DISPC Video Replication Logic

	A[11:0]	R[11:0]	G[11:0]	B[11:0]
Formats	**MSB LSB**	**MSB LSB**	**MSB LSB**	**MSB LSB**
xRGB12-4444	111111111111	R[3:0]R[3:0]R[3:0]	G[3:0]G[3:0]G[3:0]	B[3:0]B[3:0]B[3:0]
RGBx12-4444	111111111111	R[3:0]R[3:0]R[3:0]	G[3:0]G[3:0]G[3:0]	B[3:0]B[3:0]B[3:0]
RGB16-565	111111111111	R[4:0]R[4:0]R[4:3]	G[5:0]G[5:0]	B[4:0]B[4:0]B[4:3]

DSS is configured to output on 16-bit interface, ARGB48 data will be converted to RGB16-565 by truncating the LSB bits. Since the repeated pixels are truncated, the resultant data will be same as original one.

Register Configuration:

TDA2xx/TDA2Ex:

`DISPC_VID1_ATTRIBUTES.FORMAT = 0x6;`	RGB16-565
`DISPC_CONTROL1.TFTDATALINES = 0x1;`	16-bit output aligned on the LSB of the pixel data interface

TDA3xx:

`DISPC_VID_ATTRIBUTES. FORMAT = 0x3;`	RGB16-565
`DISPC_VP1_CONTROL. DATALINES = 0x1;`	16-bit output aligned on the LSB of the pixel data interface

Driver Programming:

In the `IOCTL_VPS_DISP_SET_DSS_PARAMS` IOCTL, passes the parameter `Vps_DispDssParams` with the setting below to configure input format to RGB565.

```
Vps_DispDssParams.inFmt.dataFormat = FVID2_DF_BGR16_565;
```

In the `IOCTL_VPS_DCTRL_SET_VENC_OUTPUT`, IOCTL passes the parameter `Vps_DctrlOutputInfo` with the setting below to configure output width to 16-bits.

```
Vps_DctrlOutputInfo.videoIfWidth = FVID2_VIFW_16BIT;
```

Other settings for bypassing the processing blocks in video pipeline and the overlay remain the same as in Section 2.1.

2.3 Bit Matching: 8-Bit Output Interface

This section describes the configuration when sending the bit matched output when eight data lines of the VP are used for data transfer.

There are two ways to get bit matching output over 8-bit interface: one uses RGB888 input and the other uses RGB565 input.

2.3.1 RGB888 Input

DSS supports TDM mode, you can configure the DSS to send out 24-bit data (RGB24) on 8-bit interface. In this case, one pixel (RGB24-bit) is sent in three different clock cycles.

The video pipeline configuration remains the same as in Section 2.1.1.

Enable TDM mode in the Overlay configuration and configure it in 8-bit interface mode to transmit each pixel in three cycles. Other settings, except TDM configuration, remain the same as in Section 2.1.2.

Register Configuration:

`DISPC_VP1_CONTROL.TDMENABLE = 0x1:`	TDM enabled
`DISPC_VP1_CONTROL. TDMPARALLELMODE = 0x0:`	8-bit parallel output interface selected
`DISPC_VP1_CONTROL. TDMCYCLEFORMAT = 0x2:`	3 cycles for 1 pixel
`DISPC_DATA1_CYCLE1 = 0x8`	
`DISPC_DATA1_CYCLE2 = 0x8`	
`DISPC_DATA1_CYCLE3 = 0x8`	

Driver Programming:

In the `IOCTL_VPS_DCTRL_DSS_SET_ADV_VENC_TDM_PARAMS`, IOCTL passes the parameter `Vps_DssDispcAdvLcdTdmConfig` with the setting below for enabling the TDM mode.

```
Vps_DssDispcAdvLcdTdmConfig.tdmEnable = TRUE;
Vps_DssDispcAdvLcdTdmConfig.tdmCycleFormat = 0x2;
Vps_DssDispcAdvLcdTdmConfig.tdmParallelMode = 0x0;
Vps_DssDispcAdvLcdTdmConfig.noBitsPixel1Cycle1 = 0x8;
Vps_DssDispcAdvLcdTdmConfig.noBitsPixel2Cycle1 = 0x8;
Vps_DssDispcAdvLcdTdmConfig.noBitsPixel3Cycle1 = 0x8;
```

The DSS expects memory organization for RGB888 packed input as shown in Figure 2. TDM mode outputs Red color component on the first clock cycle, followed by green and blue color components as shown in Figure 4.

	24-bpp		
	1st cycle	2nd cycle	3rd cycle
Data[7]	R0[7]	G0[7]	B0[7]
Data[6]	R0[6]	G0[6]	B0[6]
Data[5]	R0[5]	G0[5]	B0[5]
Data[4]	R0[4]	G0[4]	B0[4]
Data[3]	R0[3]	G0[3]	B0[3]
Data[2]	R0[2]	G0[2]	B0[2]
Data[1]	R0[1]	G0[1]	B0[1]
Data[0]	R0[0]	G0[0]	B0[0]

Figure 4. TDM 8-Bit Output for RGB24 Bit Input

If the different color ordering is required in the output, as explained in the section 2.1.2, CPR module in DSS can be used to reorder color components.

2.3.2 RGB16 Input

DSS supports TDM (Time division multiplexing) mode and RGB16-565 16-bit data format for input and output. So, using the TDM mode, two bytes of the RGB565 data format can be sent out in two different clock cycles over 8-bit interface.

Video pipeline configuration will remain the same as that in Section 2.2.

In the Overlay configuration, the TDM mode needs to be enabled and configured in 8-bit interface mode to transmit each pixel in 2 cycles. Other settings except TDM configuration remain the same as in Section 2.2.

Register Configuration:

```
DISPC_VP1_CONTROL.TDMENABLE = 0x1: TDM enabled
DISPC_VP1_CONTROL. TDMPARALLELMODE = 0x0: 8-
bit parallel output interface          selected
DISPC_VP1_CONTROL. TDMCYCLEFORMAT = 0x1: 2 cycles for 1 pixel
DISPC_DATA1_CYCLE1 = 0x8
DISPC_DATA1_CYCLE2 = 0x8
DISPC_DATA1_CYCLE3 = 0x0
```

Driver Programming:

In the `IOCTL_VPS_DCTRL_DSS_SET_ADV_VENC_TDM_PARAMS IOCTL` pass the parameter `Vps_DssDispcAdvLcdTdmConfig` with below setting for enabling the TDM mode.

```
Vps_DssDispcAdvLcdTdmConfig.tdmEnable = TRUE;
Vps_DssDispcAdvLcdTdmConfig.tdmCycleFormat = 0x1;
Vps_DssDispcAdvLcdTdmConfig.tdmParallelMode = 0x0;
Vps_DssDispcAdvLcdTdmConfig.noBitsPixel1Cycle1 = 0x8;
Vps_DssDispcAdvLcdTdmConfig.noBitsPixel2Cycle1 = 0x8;
Vps_DssDispcAdvLcdTdmConfig.noBitsPixel3Cycle1 = 0x0;
```

Note that DSS expects memory organization for RGB888 packed input as shown in Figure 5.

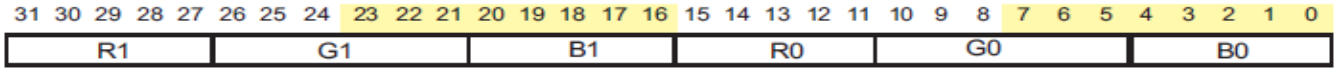

Figure 5. Memory Organization for RGB565 Input

TDM mode outputs Red color component and upper three bits of the green color component on the first clock cycle and blue component and remaining bits of the green color component as shown in Figure 6.

	16-bpp	
	1st cycle	2nd cycle
Data[7]	R0[4]	G0[2]
Data[6]	R0[3]	G0[1]
Data[5]	R0[2]	G0[0]
Data[4]	R0[1]	B0[4]
Data[3]	R0[0]	B0[3]
Data[2]	G0[5]	B0[2]
Data[1]	G0[4]	B0[1]
Data[0]	G0[3]	B0[0]

Figure 6. TDM 8-Bit Output for RGB16 Bit Input

3 References

1. *Display Subsystem* chapter in the *TDA3x SoC for Advanced Driver Assistance Systems (ADAS) Technical Reference Manual* (SPRUHQ1)
2. *Display Subsystem* chapter in the *TDA2x SoC for Advanced Driver Assistance Systems (ADAS) Technical Reference Manual* (SPRUI29)

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Original (January 2016) to A Revision	Page
• Updates were made in Section 2.1.2.	3
• Update was made in Section 2.3.	6
• Updates were made in Section 2.3.1.	6
• Added new Section 2.3.2.	7

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ('TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources 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.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

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