DLP® Discovery™ 4100 - Applications FPGA Pattern Generator Design

User's Guide

Literature Number: DLPU045
September 2016

Contents

DLP® Discovery™ 4100 - Applications FPGA Pattern Generator Design

This document will discuss the design of the Applications FPGA pattern generator.

Note on format:

This document contains an overview, block diagrams and operation scenario sections that provide an overall picture of the block functionality and describes the various modes of operation that the block may operate in.

1 General Overview

The Pattern Generation Applications FPGA was created with the sole purpose of exercising the basic functions of the DLPC410 Controller FPGA on the DLP Discovery 4100 kit and displaying the results on the digital micromirror device (DMD).

1.1 IO List

The APPSFPGA has the following input and output ports:

Table 1. IO List

Signal Name	Input/Output	Description
clk_i	Input	Input 50MHz reference clock
reset_i	Input	Input reset
finished_iv_o	Output	Configuration complete output
arstz_o	Output	Output reset (PLL reset)
clk_r_o	Output	Output 50MHz reference clock
dout_ap_o	Output(15:0)	Channel A output data
dout_an_o	Output(15:0)	Channel A output data
dout_bp_o	Output(15:0)	Channel B output data
dout_bn_o	Output(15:0)	Channel B output data
dout_cp_o	Output(15:0)	Channel C output data
dout_cn_o	Output(15:0)	Channel C output data
dout_dp_o	Output(15:0)	Channel D output data
dout_dn_o	Output(15:0)	Channel D output data
dclk_ap_o	Output	Channel A clock
dclk_an_o	Output	Channel A clock
dclk_bp_o	Output	Channel B clock
dclk_bn_o	Output	Channel B clock
dclk_cp_o	Output	Channel C clock
dclk_cn_o	Output	Channel C clock
dclk_dp_o	Output	Channel D clock
dclk_dn_o	Output	Channel D clock
dvalid_ap_o	Output	Channel A data valid

Table 1. IO List (continued)

Signal Name	Input/Output	Description
dvalid_an_o	Output	Channel A data valid
dvalid_bp_o	Output	Channel B data valid
dvalid_bn_o	Output	Channel B data valid
dvalid_cp_o	Output	Channel C data valid
dvalid_cn_o	Output	Channel C data valid
dvalid_dp_o	Output	Channel D data valid
dvalid_dn_o	Output	Channel D data valid
rowmd_o	Output(1:0)	Row Mode
rowad_o	Output(10:0)	Row Address
stepvcc_o	Output	Unused
comp_data_o	Output	Complement data
ns_flip_o	Output	North/South flip
blkad_o	Output(3:0)	Block Address
blkmd_o	Output(1:0)	Block mode
wdt_enablez_o	Output	Watchdog time enable
ddc_version_i	Input(2:0)	DLPC410 Version number
dmd_type_i	Input(3:0)	DMD type designator
pwr_floatz_o	Output	Power Float
apps_cntl_an	Input	Unused
apps_cntl_ap	Input	Unused
rst2blkz_o		Dual block reset mode select
in_rst_active_i	Input	Reset active indicator
in_init_active_i	Input	Initialization active indicator
in_dip_sw_i	Input(7:0)	Switch inputs
in_pb_sw_i	Input	Input pushbutton
apps_logic_rstn	Input	Input reset
apps_testpt	Output(30:0)	Test point outputs

2 APPSFPGA Top Level

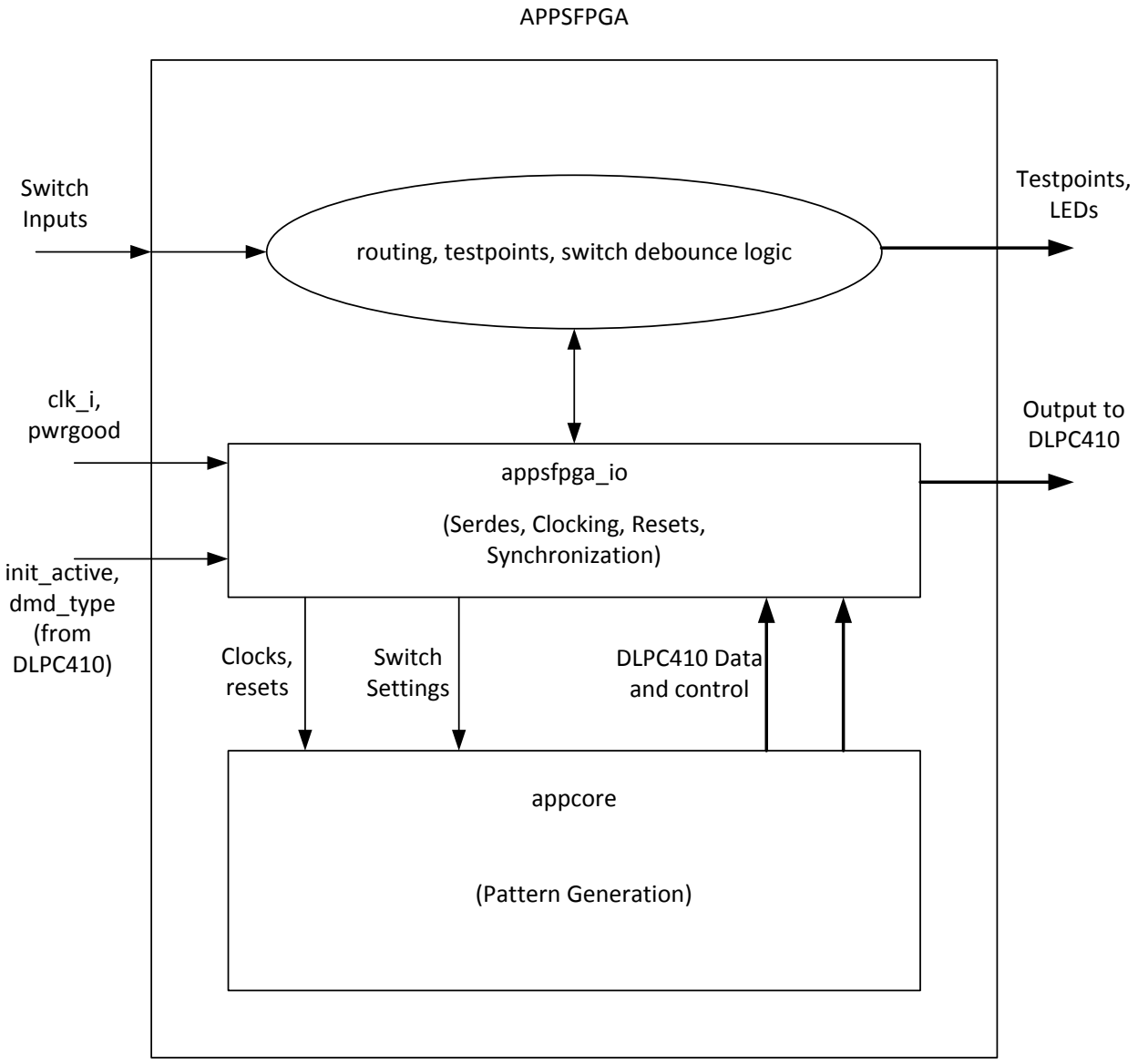


Figure 1. APPSFPGA Top Level

The top level of the APPSFPGA design provides some routing, contains the incoming DIP switch and pushbutton debounce logic, test point output assignments, and instantiation of the two major blocks of the design. These are the "appsfpga_io" block and the "appcore" block.

2.1 Input Switch Functions

There are several input switch functions that control the various operational modes of the APPSFPGA. They are:

- in_pb_sw_i : Power Float [active low (0)]

  Sends a sequence of commands that places the DMD mirrors in a flat or “parked” state (must power cycle to recover from this command).
- in_dip_sw_i(7) : Watchdog Timer enable [active high (1)]

  Enables a timer that will force a mirror reset to the DMD approximately every 10 seconds (if one has not been issued).
- in_dip_sw_i(6) : Row Address mode [low (0) = automatic increment/decrement mode, high (1) = row addressing mode]

  Selects between addressing specific rows and a row increment/decrement mode.
- in_dip_sw_i(5:4) : Reset Type [encoded value]

  Determines if global, single, dual, or quad resets are being used.
  - Single block -- 0,0
  - Dual block -- 0,1
  - Global -- 1,0
  - Quad block -- 1,1
- in_dip_sw_i(3) : North/South flip [active high (1) -- flips the image top to bottom]

  Determines if the row and reset addressing starts at the “top” or “bottom” of the DMD.
- in_dip_sw_i(2) : Complement Data [active high (1) -- inverts the image data]

  Causes the DMD to invert the incoming image data such that 0's become 1's and 1's become 0's
- in_dip_sw_i(1) : Count Halt [active high (1) -- halts the main counter]

  “Freezes” the load operation at a particular point in the pattern count cycle. See Section 4.1.
- in_dip_sw_i(0) : IO Float [active high (1) -- sets V_Bias to 0]

  Recoverable version of the “Power float” command (this command will recover if de-asserted).

For additional information see the DLPC410 datasheet.

3 "appsfpga_io" Block

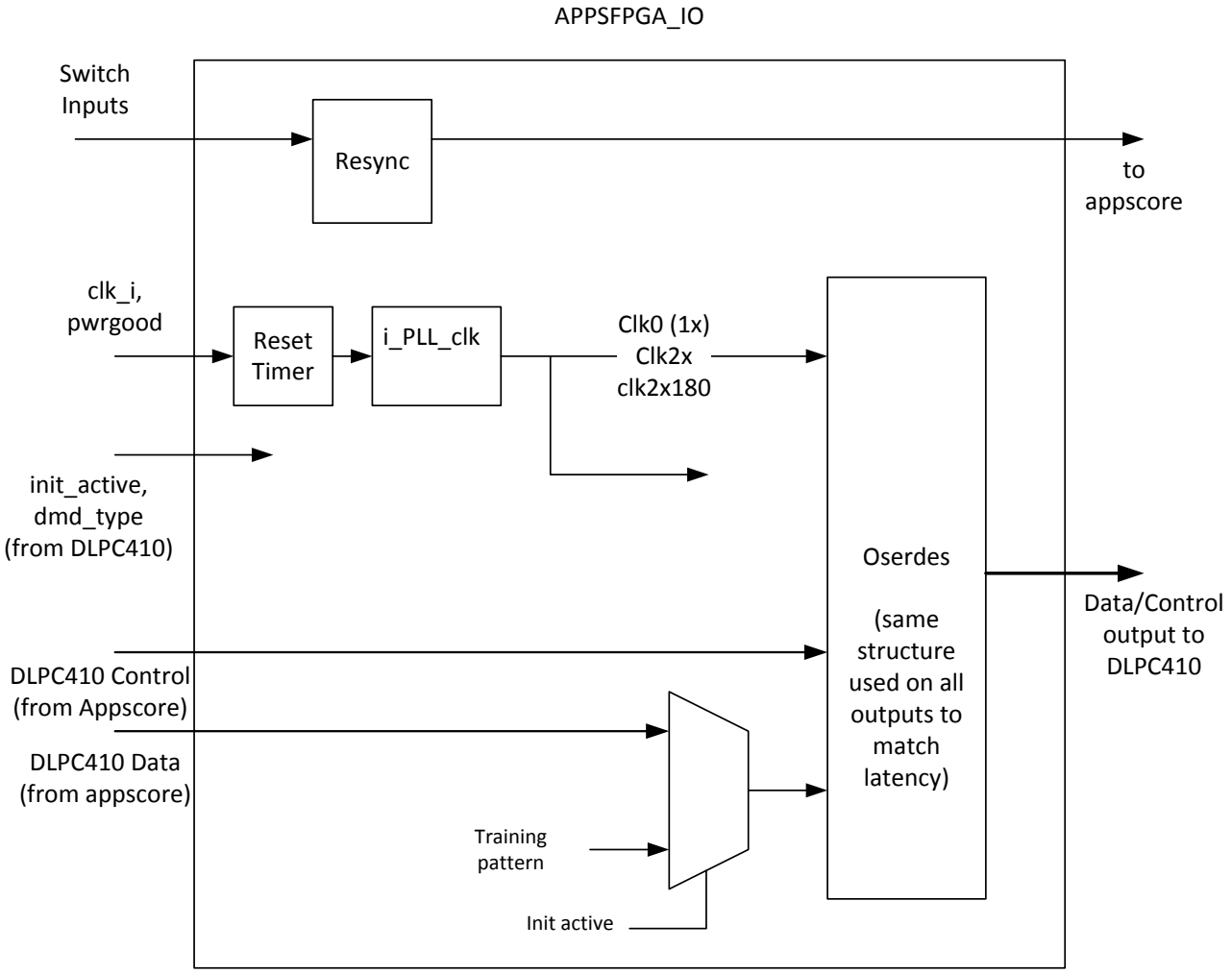


Figure 2. "appsfpga_io" Block Diagram

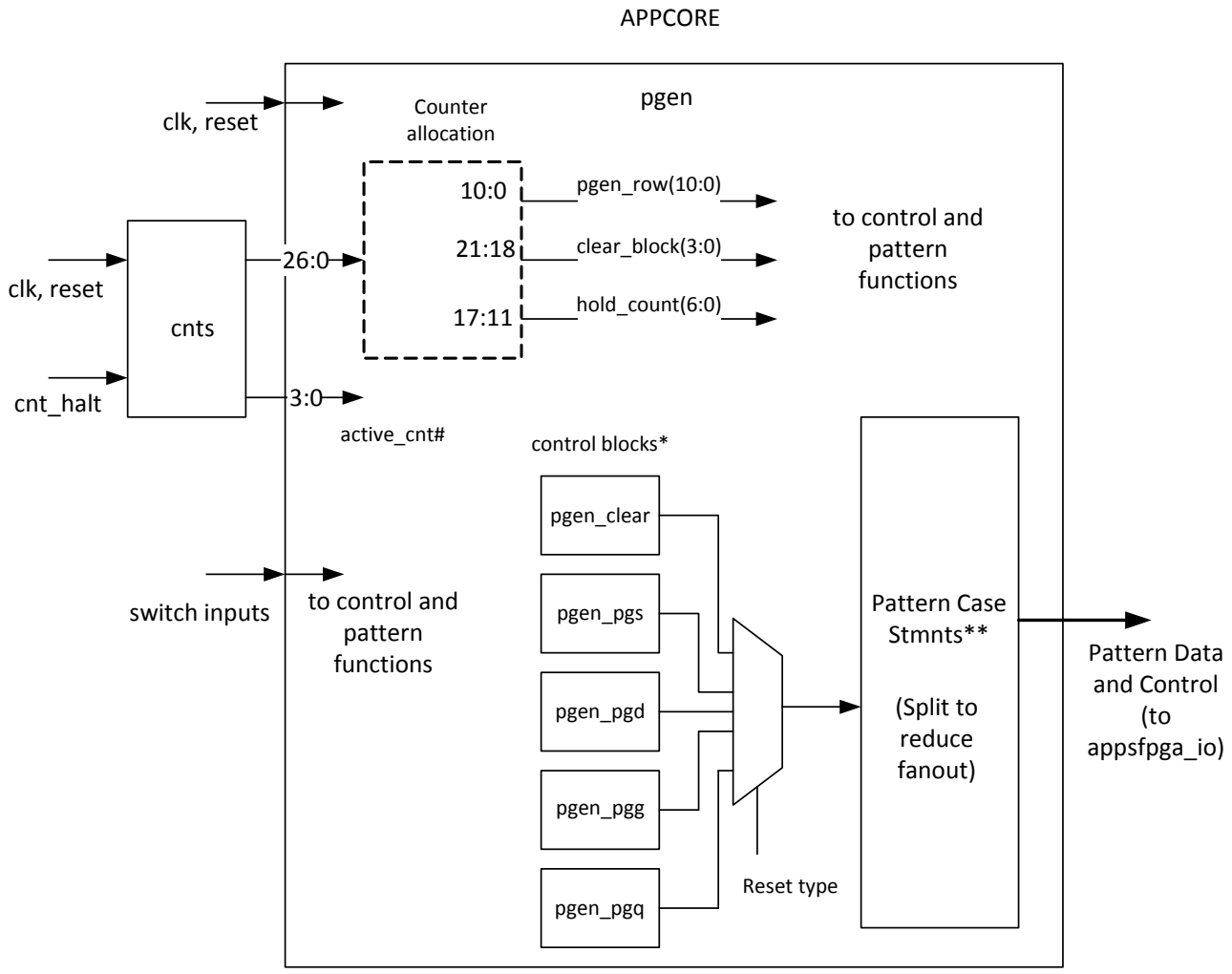
The "appsfpga_io" block performs the following functions:

It receives the input reference clock and uses the PLL to create the other clocks used in the design. The incoming clock is multiplied up to a clock that is one half the base clock frequency (100 MHz for the 200 MHz DDR data rate design) which is used in the rest of the design, and a phase aligned clock at the base clock frequency (100 MHz for the 200 MHz DDR data rate design) which is only used in the output section. Since the APPSFPGA does not include the capability to operate at various clock frequencies, three versions of the "appsfpga_io" block are provided, the appsfpga_io_200_a.vhd, the appsfpga_io_320_a.vhd, and the appsfpga_io_400_a.vhd, which operate at 200 MHz, 320 MHz, and 400 MHz respectively.

The test pattern data and control signals are sent from the "appcore" block and clocked out through the "appsfpga_io" block. A simple multiplexer will substitute the required training pattern data whenever the incoming top level “init_active” signal is detected.

The output data is reformatted and sent to the DLPC410 using the Xilinx 4:1 oserdes primitives. All of the output signals (even the lower speed outputs) use the output serdes structure. Since the oserdes are implemented in the hardened IO ring of the FPGA, this ensures that the clock-output timing will be very close across all of the outputs.

4 "appcore" Block



* These blocks generate control signals for the various reset and clear modes

** This process generates the data for the various test patterns

\# This count defines the various segments of active rows

Figure 3. "appcore" Block

The "appcore" block handles the pattern and control generation functions of the design and is made up of the "cnts" and "pgen" sub-blocks.

4.1 "cnts" Sub-block

The “cnts” sub-block consists of two counters that drive all of the other functions in the design. A small 4-bit counter (cnts_active_cnt) is used to tell the PGEN logic which segment of the currently active line it should be generating. A larger 27-bit counter (cnts_pattern_cnt) is used to keep track of the pattern and row counts, generate clears and resets, create correct spacing, etc. Activating the “count halt” switch will freeze this counter, effectively freezing the image. Three variations of the “cnts” block are provided that generate different spacing between the patterns/active rows/resets.

cnts_a_1and0clks.vhd places zero idle clocks between lines.

cnts_a_1and2clks.vhd places two idle clocks between lines.

cnts_a_1and16clks.vhd places sixteen idle clocks between lines.

It should be noted that, since the internal clock rate is one half of the base DDR clock rate, any spacing placed between lines/data must be an even number of clocks.

4.2 "pgen" Sub-block

The "pgen" sub-block uses these counter values to generate the pattern data and control signals that are sent to the DLPC410. The 27 bit count value is received from the cnts block and is split into three separate fields:

- cnts_pattern_cnt(10:0) = pgen_row(10:0)
- cnts_pattern_cnt(17:11) = hold_count(6:0)
- cnts_pattern_cnt(21:18) = clear_block(3:0)

  NOTE: (26:22) are not used.

When looking at any of the counter values used in these blocks, it is important to remember that the clock is running at half of the base clock frequency, so many count values will be half of what might be expected.

The clear and reset commands are generated by five blocks. The outputs of these blocks feed into a multiplexer and the “reset type” selection (and the pattern type) determines which block’s output is used. The blocks and their functions are:

- pgen_clear : generates global clear instructions
- pgen_pgs : generates single block reset instructions
- pgen_pgd : generates dual block reset instructions
- pgen_pgq : generates quad block reset instructions
- pgen_pgg : generates global reset instructions

A large process that includes two of case statements is used to generate the actual output pattern data for the A, B, C, and D data buses. The case statement was split into two sections to reduce fan out and help reduce timing closure issues. The active_cnt and a pipelined version of pgen_row signals are used to guide the data generation process.

As previously mentioned, it is important to remember that the clock being used in the data generation processes is half of the target clock frequency. Since the data output is DDR at the target frequency, this means that the data generated on the half speed clock must be four times as wide as the data presented at the outputs. Ex: 64 bits at 100MHz = 16 bits DDR at 200MHz.

The patterns displayed by the pgen block are:

- Large box with smaller boxes in the corners
- Thin vertical lines
- Small checkerboard
- Alternating thin and thick vertical lines
- Diagonal lines
- Black screen (with global clear)

5 VHDL Libraries

There are two VHDL libraries, a “synchronizers” and a “ddc4100” library, needed to compile the APPSFPGA code in addition to the default “work” library. The “synchronizers” library is needed for all of our default synchronizer code and packages which are located in the “<installation directory>/Software/synchronizers/” directory. A “ddc4100” library is also required for a design specific package (“<installation directory>/Software/appsfpga_dmd_types_pkg.vhd" file) that is called in the PGEN code.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

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

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license 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 significant portions 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. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

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

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

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