SubChip Design Example

Abstract

A SubChip is a gate-level module that has been tested and optimized for size, timing and function and then placed and routed in a target technology. It can then be instantiated into any other design in the same target technology in the same manner as any other gate.

This application report demonstrates the method for creating a SubChip using the Texas Instruments Design Support Software (TIDSS™) flow. It is intended for ASIC designers learning how to implement the SubChip capability. It assumes that you already have experience with TIDSS tools and third party design tools. This application report is not designed for those learning to use the Texas Instruments (TI™) ASIC design tools.

This application report contains two example flow modules:

- SubChip Create Flow. This module introduces each tool and its part in the SubChip flow.
- SubChip Use Flow. This module demonstrates what is needed to properly instantiate a SubChip into a design. The standard TIDSS design flow is utilized with few changes.

Each module steps you through each tool.

Contents

Figures

Tables

Conventions Used in this Document

- *path* refers only to the fully qualified path.
- *pathname* refers to the combination of the fully qualified path and the filename.
- % denotes the UNIX prompt.
- ~ denotes the user's home directory used such as /user/fred.

SubChip Overview

Basic SubChip Flow

The SubChip flow utilizes the TIDSS design flow with very few differences, as can be seen in Figure 1. Once you have created the gate-level netlist for the target SubChip, you proceed through the normal flow to create the SubChip GOOD™.

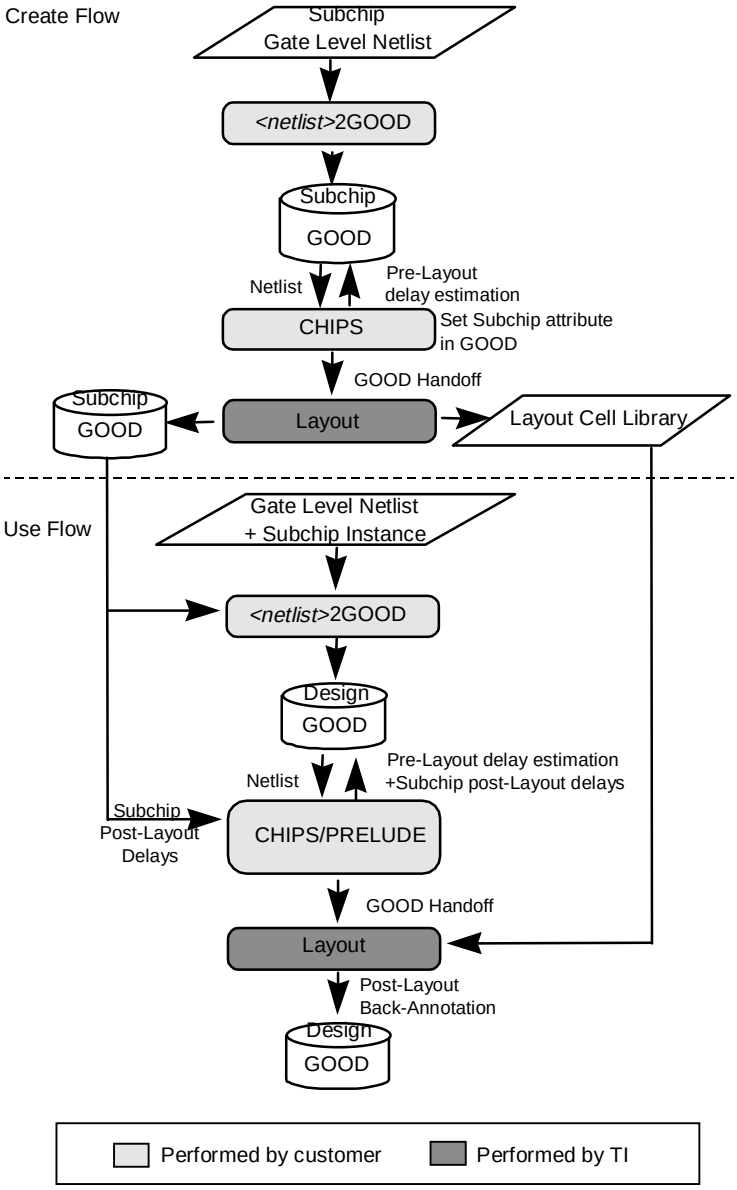
TI's SubChip create flow begins with a <netlist>2GOOD translator. For the purpose of this design example, we use VERILOG2GOOD. From the gate level netlist, the SubChip GOOD is created by VERILOG2GOOD.

You then run FIZZ2GOOD to set input slews and output loads for SubChip I/O ports and set the SubChip base array (see the *SubChip Base Arrays* section for details). Due to the nature of a SubChip, the TESTER, PACKAGE, and PIN statements are not allowed by FIZZ2GOOD.

From here, you must use CHIPS™ to floorplan the SubChip. Some design requirements that must be met for the SubChip create flow can only be done by using CHIPS. CHIPS sets the SubChip flag in the GOOD to be accessed by tools in the use flow. When floorplanning, CHIPS contains the SubChip components in a highly utilized block. (The SubChip components are not scattered across the base array, but placed in a compact area.)

You can use utilization and aspect ratio settings to decrease or increase the area of the SubChip. The I/O ports are placed outside of the base array for manual placements. Automatic placement can be accomplished by specifying their placement in the .fp file (see the *CHIPS Details* section).

Figure 1. Basic SubChip Flow



DETECTOR II™ checks to see that the design conforms to the SubChip methodology. The common SubChip errors include:

- Cells Not Allowed. Both peripheral macros and clock distribution macros are illegal in a SubChip.
- Fanin/fanout. The fanin and fanout cannot be greater than 1.
- Self-contained Clock Trees. Any clock tree within a SubChip cannot drive anything external to that SubChip.
- SubChip Nesting. A SubChip cannot be instantiated within another SubChip.

When performing checks on a SubChip, the core net errors and warnings, load/slew rate errors and warnings, and SubChip errors must be verified. DETECTOR adds a completion code to the SubChip GOOD according to the results of these rule checks. The completion code is accessed by DETECTOR in the use flow to determine if an error or warning should be issued for the SubChip.

GOOD2DELAY can then be used to create SDF for pre-layout timing simulation. After pre-layout delay estimation, the designer uses simulation to verify that these estimated timings meet design requirements.

The SubChip GOOD is then handed-off for place and route as in any other design. A back-annotated GOOD is given back to the designer for post-layout timing verification. The designer must then verify the post-layout SubChip GOOD and notify TI when the SubChip meets all requirements. The SubChip post-layout GOOD is not archived by TI for use in other designs until the designer notifies TI that the SubChip meets specification. The designer is now ready to enter the use flow.

Once it has been verified, the SubChip can be instantiated within a design (such as a NA210) and run through the normal flow to create the Design GOOD.

During <netlist>2GOOD, a SubChip module is required by the translator. This SubChip module must contain only the input and output declarations. All internal connectivity information is available to the translator from the SubChip GOOD. Executing FIZZ2GOOD on a design with an instantiated SubChip requires that the design VTP be the same as that of any instantiated SubChip. When Prelude or CHIPS is run, the SubChip post-layout timing is transferred to the Design GOOD. When calculating delays for a design with an instantiated SubChip, this post-layout SubChip timing is then used by GOOD2DELAY for pre-layout and post-layout modes.

During the design place and route, the SubChip is treated as a black box. The routing and internal timing are not modified. Only the timing for the peripheral cells will change, depending on the actual load and slew.

TIDSS Changes

- FIZZ2GOOD:

  – *subchip*

  This switch causes FIZZ2GOOD to execute in the SubChip create mode. In SubChip create mode, FIZZ2GOOD does not require PACKAGE, TESTER, or PIN statements in the fizz file. Warnings are issued if these statements are specified.

- CHIPS:

  $$-\ subchip\{true|false\}$$

  The –subchip true switch causes CHIPS to execute in the SubChip create mode. In the SubChip create mode, CHIPS treats the SubChip as any other design, placing the top level interface cells as peripheral cells. The default is “false.”

- DETECTOR:

  – *subchip*{*true*|*false*}

  This switch causes DETECTOR to execute in the SubChip create mode. In the SubChip create mode, DETECTOR executes the SubChip rule set. The default is “false.”

SubChip Base Arrays

In the SubChip create flow, one of the SubChip base arrays must be used. A regular design base array is not valid for SubChip create.

You should choose the base array with the target design in mind. You have the choice of creating a very dense or fairly porous SubChip. The two base arrays available in TGC4000 are shown in Table 1.

Table 1. TGC4000 Base Arrays

Base Array	Metal Level	Resulting SubChip Characteristics
TGC4SUB	QLM	Very dense, compact layout
		Completely obstructed in use flow
		Must be placed along the edge of the design in use flow
		No ability to route clock signals across in use flow
		May cause routing congestion during use flow
TGC4SUB_TLM	TLM	Fairly porous layout
		50% of Metal3 available for clock and through routing in use flow
		All of Metal4 available for routing in use flow
		Can be placed almost anywhere, except over Clock Distribution macro
		Less dense than if layout is in a QLM base array

For each metal level, only one SubChip base array is available. Each SubChip base array represents the maximum SubChip size. During floorplanning and layout, the SubChip is optimized for density by compacting the SubChip to the smallest size possible at the target utilization of the technology. The remaining area not used by the SubChip is discarded.

CHIPS Details

During the SubChip create flow, CHIPS must be used for floorplanning. CHIPS processes the SubChip the same way it processes any other design.

The utilization and aspect ratio can be set to influence compactness. The maximum utilization is that of the technology. If a greater utilization is specified, CHIPS gives a warning but proceeds with that utilization. This may cause unroutable designs.

Like full designs, CHIPS allows the SubChip I/O ports to be automatically placed with entries in the floorplan file (.fp). The entry has the following format:

```
SUBCHIP_IO <instance name> TOP|BOTTOM|LEFT|RIGHT) <x>
```

The instances listed must be inputs or outputs to the SubChip.

TOP, BOTTOM, LEFT, and RIGHT indicate on which side of the SubChip a peripheral cell should be placed.

<x> is a positive integer that is required for each SubChip peripheral cell. This number allows ordering of all the peripheral cells on a side. The ordering is always top to bottom, or left to right. The SubChip peripheral cells are evenly spaced in the order indicated.

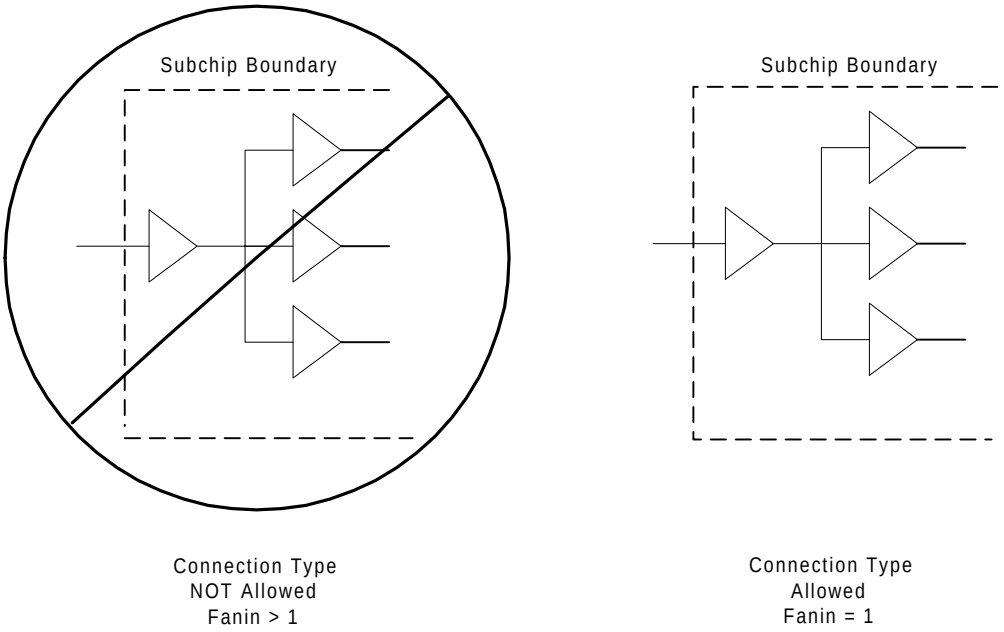
If the peripheral cells are not listed in the floorplan file, they are placed outside the SubChip. It is up to you to manually place each peripheral cell along the boundary of the SubChip.

During the SubChip use flow, PRELUDE™ or CHIPS treats any instantiated SubChip similarly to a RAM. Internal routing of the SubChip is not changed. The SubChip post-layout timing is copied into the design GOOD from the SubChip GOOD without being recalculated. PRELUDE/CHIPS will only recalculate the timing of nets that cross the SubChip boundary.

SubChip Limitations

- Peripheral port placement is fixed.
- SubChip may cause routing blockages in full chip routing.
- Fanin must be 1, not greater, as illustrated in Figure 2.
- Fanout must be 1, not greater, as illustrated in Figure 3.
- The SubChip must consume a rectangular area only (L-shape is not allowed).
- The SubChip orientation is limited by the most constraining macro.
- Placement of the SubChip on the base array is restricted to either side of the Clock Distribution Macro (when in use).
- No bidirectional macros or I/O macros are allowed.
- All peripheral macros must be at the top level within the SubChip hierarchy.
- Embedding is not allowed.

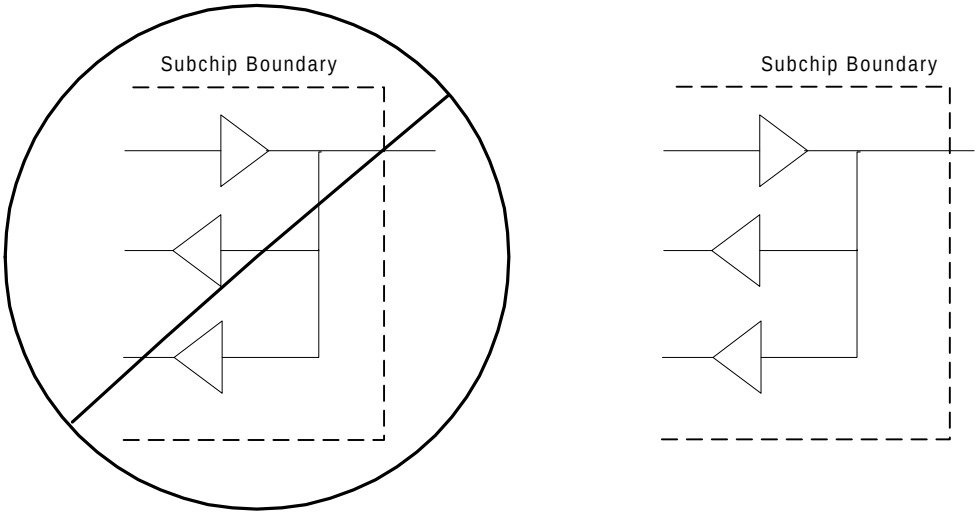
Figure 2. Fanin Limitation



Connection Type
NOT Allowed
Fanin > 1

Connection Type
Allowed
Fanin = 1

Figure 3. Fanout Limitation



Connection Type
NOT Allowed
Fanout > 1

Connection Type
Allowed
Fanout = 1

Setting Up the Example

Requirements Checklist

The answers to the following questions are needed to complete the example. If you do not know the answers, ask another user.

- ❒ Do you have the example files? (If not, see the *Obtaining the Example Files* section.)
- ❒ In which directory are the TIDSS tools stored?
- ❒ Do you have CHIPS? (CHIPS must be used in the Design Create Flow.)

Software Tools

This module demonstrates the method for creating a SubChip using the TIDSS flow. For further information on a tool, see the *Submicron ASIC Products Design Software Manual* (SRGA003).

In the design example, the following tools will be utilized:

- ❒ TIDE
    - ❑ VERILOG2GOOD
    - ❑ FIZZ2GOOD
    - ❑ CHIPS*
    - ❑ PRELUDE
    - ❑ DETECTOR II
    - ❑ GOOD2DELAY
    - ❑ TDL2VERILOG
    - ❑ Verilog™*

*These tools are not part of TIDSS standard software.

Obtaining the Example Files

TI customers can obtain a complete copy of the example files by contacting their CDC or applications representative.

For TI employees, the design example files are available on the TI network. If you have access to TI internal sources, follow these steps:

1) FTP to asic.sc.ti.com (you must have an account on this machine.) The file you need is:
   /sw_releases/SubChip_dsgn_example.tar
2) In binary mode, download the file to your computer.
3) Extract the tar file to the current working directory on your computer.

If you received your example files on tape media, use the command **`tar -xvf <tape_device_name>`** to unpack the files. This copies the files into the current working directory.

NOTE:
For TGC4000 1.0, CTS is available tactically only. A CHIPS patch is required to properly copy CTS timing from the SubChip GOOD into the Design GOOD.

Setting Up the Directory Structure

The locations of programs used in this example are based on the TI Dallas applications environment. Your file locations and paths may vary due to your network configuration. Contact your system administrator for local system information.

Before you begin, you need to obtain the files needed for the example. See the *Obtaining the Example Files* section.

At this point your files should be available in a directory called `SubChip_Example`.

Hints for Resolving Errors

In performing this example, errors may occur due to improper pathname settings, modifications in the tools, or for other reasons. Most errors can be solved easily. Here are some hints for resolving errors:

- Make sure tide_info is sourced.
  - Check pathnames.
  - Verify that the files requested exist.
  - Read the directions again.
  - Examine the error messages for clues as to the root of the problem.

In most cases this example can be done in one sitting. If you must stop, stop at the end of any tool. To continue work, change your directory to `~/Create` (or `~/Use`) and source `tide_info;` then begin with the next tool.

SubChip Create Flow

This module demonstrates the method for creating a Subchip using the TIDSS flow. For further information on any tool, see the *Submicron ASIC Products Design Software Manual* (SRGA003).

In this example, a clock phase divider is created as a SubChip. The flow chart of this example is illustrated in Figure 1. The clock divider in this example is only one of many uses of SubChip. This example has been designed for ease of use and learning.

Task 1: Set Up the Design Directory

Once you have accessed the SubChip Create example directory, you need to set all your environment variables using TIDE.

1) Run TIDE:

   %/<ti_home_dir>/bin/tide

   This starts the TIDE script. Point it to TIDSS Release 4.1. You will be creating a TLM SubChip.

   Set the environment variables as follows:

unix shell	**C shell**
design directory	***use default***
library	**TGC4000 R1.0**
synthesis	**synopsys**
simulator	**verilog**
timing verification	**veritime**
test pattern gen	**synopsys**
floor planner	**TI**
timing	**com**
path of Design GOOD	**<your_root_dir>/ SubChip_Example/ Create/US0001**
metal level	**TLM**
company	**TI**
part number	**US0001**
TI division	**USA-ASIC**

2) Source the tide info file:

   %source tide_info

Task 2: Create the SubChip GOOD

1) Run VERILOG2GOOD:

```
%verilog2good -f v2g.cmd
–scope US0001
```

VERILOG2GOOD creates the SubChip design GOOD without any changes to the regular design flow.

2) Run FIZZ2GOOD:

```
%fizz2good -fizz US0001.fizz
–err f2g.err -subchip
```

FIZZ2GOOD sets the slew and load for I/O ports and specifies the SubChip base array.

Task 3: Floorplan the SubChip with CHIPS

NOTE:
CHIPS is not part of the standard TIDSS software package and needs to be licensed and installed separately. Contact your CDC if you do not have CHIPS installed.

1) Run CHIPS:

```
%chips -subchip true
```

With the -subchip true switch, CHIPS floorplans the SubChip.

CHIPS will finish with 3 warnings, which indicate that an empty IO list was encountered. You may ignore these warnings, the SubChip base array was purposely designed without I/O slots.

For an extra assignment, rename the US0001.fp file; then run CHIPS without a file placement file. This causes the I/O ports to be placed to the side. Manual placement is required.

Task 4: Verify the Design

NOTE:
In a real SubChip design, you would want to run DETECTOR in post-layout mode in the event that any slew rates, etc., had changed. Simulation and timing analysis would also be run on the post-layout design. When the simulation and timing meets the specification, you would notify TI that the SubChip is acceptable. This design example assumes you are familiar with simulation and timing analysis.

1) Run DETECTOR II:

```
%detector -subchip true -pre
```

2) From the MAIN MENU, execute the RULES submenu.

```
>R
```

3) Turn off all rules except the following:

```
core_net_errors
core_net_warn
load_slew
subchip_errors
```

You may leave stats on if you wish.

The design should pass all checks without errors. If errors do occur or any one of the specified rules is turned off, DETECTOR stores a bad completion code in the SubChip GOOD.

4) Run GOOD2DELAY:

```
%good2delay -out US0001.sdf
–target verilog -pre
–scope US0001
–sdfversion ignore
–noconstraint
```

GOOD2DELAY is run as normal to create pre-layout SDF. You may ignore the warnings in this example.

5) Run TDL2VERILOG:

```
%tdl2verilog -tdl US0001.tdl
–vtdl US0001.vtdl
–scf US0001.scf -scfmod main
–dut US0001 -dutinst US0001
–strobe 10 -mis 0
–sdf US0001.sdf
–scope main.US0001
–vcd verilog.dump
–tivcd US0001.tivcd
–incr 10 -finish
```

TDL2VERILOG creates the Verilog stimulus file (US0001.scf).

6) Run Verilog simulation:

```
%verilog -f US0001.verl
```

The Verilog simulation requires a version of Verilog with the TDL2VERILOG PLIs bound in. Ask your system administrator for the location of this executable.

Product Support

Related Documentation

Submicron ASIC Products Design Software Manual (SRGA003).

World Wide Web

Our World Wide Web site at www.ti.com contains the most up to date product information, revisions, and additions. Users registering with TI&ME™ can build custom information pages and receive new product updates automatically via email.

TI Contact Numbers

INTERNET

TI Semiconductor Home Page
www.ti.com/sc

TI Distributors
www.ti.com/sc/docs/distmenu.htm

PRODUCT INFORMATION CENTERS

Americas
Phone +1(972) 644-5580
Fax +1(972) 480-7800
Email sc-infomaster@ti.com

Europe, Middle East, and Africa
Phone
- Deutsch +49-(0) 8161 80 3311
- English +44-(0) 1604 66 3399
- Español +34-(0) 90 23 54 0 28
- Francais +33-(0) 1-30 70 11 64
- Italiano +33-(0) 1-30 70 11 67

Fax +44-(0) 1604 66 33 34
Email epic@ti.com

Japan
Phone
- International +81-3-3344-5311
- Domestic 0120-81-0026

Fax
- International +81-3-3344-5317
- Domestic 0120-81-0036

Email pic-japan@ti.com

Asia
Phone
- International +886-2-23786800
- Domestic
  - Australia 1-800-881-011
    - TI Number -800-800-1450
  - China 10810
    - TI Number -800-800-1450
  - Hong Kong 800-96-1111
    - TI Number -800-800-1450
  - India 000-117
    - TI Number -800-800-1450
  - Indonesia 001-801-10
    - TI Number -800-800-1450
  - Korea 080-551-2804
  - Malaysia 1-800-800-011
    - TI Number -800-800-1450
  - New Zealand 000-911
    - TI Number -800-800-1450
  - Philippines 105-11
    - TI Number -800-800-1450
  - Singapore 800-0111-111
    - TI Number -800-800-1450
  - Taiwan 080-006800
  - Thailand 0019-991-1111
    - TI Number -800-800-1450
  - Fax 886-2-2378-6808
  - Email tiasia@ti.com

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgement, including those pertaining to warranty, patent infringement, and limitation of liability.

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

CERTAIN APPLICATIONS USING SEMICONDUCTOR PRODUCTS MAY INVOLVE POTENTIAL RISKS OF DEATH, PERSONAL INJURY, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE (“CRITICAL APPLICATIONS"). TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. INCLUSION OF TI PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE FULLY AT THE CUSTOMER'S RISK.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. 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 of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty, or endorsement thereof.