Application Report
SPRABW3–October 2014

TI Keystone DSP PCIe SerDes IBIS-AMI Models

David Liu

ABSTRACT

This document describes the organization, structure, and proper usage of the TI serializer and deserializer (SerDes) IBIS-AMI models for Keystone DSP PCIe interface.

Contents

List of Figures

List of Tables

1 Overview

The TI SerDes IBIS-AMI models for Keystone Hyperlink interface, referred to throughout the remainder of the document as 'the model', are intended by SerDes customers for system-level modeling and verification. This document assumes that you are familiar with the relevant IBIS-AMI modeling specifications.

1.1 Charter of the SerDes IBIS-AMI Models

The model is designed in accordance with the IBIS-AMI standard and attempts to model the significant characteristics of most components in the SerDes. The models are not intended to be an exact representation of SerDes components implemented. Rather, the models seek to provide as high a degree of accuracy as is feasible outside of Spice-based models and simulations.

Table 1. Terms and Abbreviations

BER	Bit Error Rate. A metric of link performance equal to the number of errors divided by the number of bits received.
Channel	General term for a link between a Tx and Rx
EDA Simulator	General term for various commercially available IBIS-AMI simulators.
FIR	Finite Impulse Response. Often used in conjunction with the Tx filter settings.
FIR map	A matrix of Tx FIR filter tap settings spanning multiple pre-cursor and multiple post-cursor tap settings.
Model	*model* refers to TI's SerDes IBIS-AMI model
PRBS	Pseudo-random bit sequence
s4p	*s4p* refers to a four port S-parameter in touchstone format
SEEH	Single-ended eye height. A figure of merit in the context of SerDes simulations. It is related to the single-ended height of the post-equalized statistical eye at a particular BER/probability. An equivalent interpretation is the amount of single-ended noise that can be added to the signal at the input to an ideal sampler and still achieve the specified BER.
SerDes	Serializer and deserializer. General term for IP macros of this type.
SerDes Simulator	SerDes Simulator refers to TI's SerDes simulator platform
Rx	Receiver. The deserializer part of a SerDes.
Tx	Transmitter. The serializer part of a SerDes.

1.2 IBIS Interconnect Modeling Specification (IBIS) Standard Version

The model release conforms to IBIS standard version 5.0.

1.3 Software Required to Run the Model

The models require that the following software be installed on the computer. Note that these are one-time installation steps that set up the computer to run any current and future releases of the TI's IBIS-AMI models, unless otherwise specified.

- Matlab Compiler Runtime (MCR): The MCR enables you to use all the core functionality of MATLAB while executing the Platform without having to install MATLAB. Since TI's AMI model executes compiled MATLAB code, this software is required to be installed on the computer. For detailed instructions on MCR installation, see Appendix A.
- EDA Simulator: EDA Simulators capable of simulating IBIS-AMI models are required to be installed on the target system. For installation steps, see the installation guide provided by the EDA simulators. A list of EDA simulators used to verify this model release is provided in Section 1.4. Appendix B provides instructions on how to provide and set up the TI IBIS-AMI models in the different EDA Simulators.

1.4 EDA Simulators Qualification

This model release was verified with the following EDA Simulators:

- SiSoft™, Quantum Channel Designer™ 2010.06-SP8 Build 4
- Agilent, Advanced Design System 2011.01
- Sigrity, Channel Designer Version 10.1.5.03141

1.5 OS Platforms Supported

Windows 32 bit

1.6 Opal Quality Statement

This model release is compliant with Opal requirements. For more details on this, see Appendix C.

1.7 AMI Model Correlation Statement

- This release of TI's AMI model had been correlated to TI's SerDes simulator platform model. TI's SerDes simulator platform model has been correlated to silicon measurement data.
- The correlation has been performed over multiple data rates and across multiple channels. The correlation has been performed using silicon from nominal and worst-case process corners.
- The method used for determining correlation involves measuring link performance over a transmit FIR map and comparing the result from silicon measurements and simulation results from an equivalent setup in the TI's SerDes simulator platform model.

2 Release Contents

The release contents are available in a zip file named ti_draco_40nm_ami_v1p9.zip. This file contains the following folders:

- tx_ibis_ami: Contains the tx IBIS, AMI, Dynamic Link Library (DLL) files and the tx analog model as s4p files
- rx_ibis_ami: Contains the rx IBIS, AMI, DLL files and the rx analog model as s4p/s2p files
- examples: Contains two folders: one named channels_and_packages that contains example channel and package models in s4p format and the other named project_kits that contains demo project kits for various EDA Simulators.

Table 2 describes the key files and folders delivered with the model release as part of the compressed archive.

Table 2. Files and Folders Included With the Model Release

File Name	Folder	Type	Description
draco_40nm_tx.ibs	tx_ibis_ami	IBIS	Top-level IBIS wrapper for the Tx AMI model
draco_40nm_tx.ami	tx_ibis_ami	AMI	Parameters file for the Tx model as required by the IBIS-AMI standard. This is a text file that is common for all OS/execution platforms.
draco_40nm_tx.dll	tx_ibis_ami	DLL	Windows 32-bit compiled shared library for the Tx model. This shared library includes the AMI_Init, AMI_GetWave, and AMI_Close functions defined in the IBIS-AMI standard.
draco_40nm_rx.ibs	rx_ibis_ami	IBIS	Top-level IBIS wrapper for the Rx AMI model
draco_40nm_rx.ami	rx_ibis_ami	AMI	Parameters file for the Rx model as required by the IBIS-AMI standard. This is a text file that is common for all OS/execution platforms.
draco_40nm_rx.dll	rx_ibis_ami	DLL	Windows 32-bit compiled shared library for the Rx model. This shared library includes the AMI_Init, AMI_GetWave, and AMI_Close functions defined in the IBIS-AMI standard.
draco_40nm_* _txdac_*.s4p	tx_ibis_ami/tx_ analog_model	s4p	Four port single-ended touchstone files representing analog front-end characteristics of the transmitter.
draco_40nm_* _rxterm_*.s4p	rx_ibis_ami/rx_ analog_model	s4p	Four port single-ended touchstone files representing analog front-end characteristics of the receiver.
draco_40nm_* _rxterm *.s2p	rx_ibis_ami/rx_ analog_model	s2p	Two port single-ended touchstone files representing analog front-end characteristics of the receiver. The choice of AMI simulation tool dictates whether to use s2p or s4p for simulations.
example_thru_ channel.s4p	examples/channels_ and_packages	s4p	Example channel model file that can be used to run TI's IBIS-AMI model. This file models a typical 6 inch FR4 trace.
example_xtalk_ channel.s4p	examples/channels_ and_packages	s12p	Example channel model file that can be used to run TI's IBIS-AMI model in xtalk mode. This file can be used to setup a link simulation with one victim and two aggressors.
example_tx_pkg.s4p	examples/channels_ and_packages	s4p	Example transmit package model file that can be used with TI's TX IBIS-AMI model.
example_rx_pkg.s4p	eexamples/channels_ and_packages	s4p	Example receive package model file that can be used with TI's RX IBIS-AMI model.
sisoft_qcd_project	examples/ project_kits	Folder	Contains an example IBIS-AMI model using TI's Tx and Rx. This project is set up with the example channel and package and is set up to run with default AMI parameter settings.
mentor_hyperlynx_ project	examples/ project_kits	Folder	Contains an example IBIS-AMI model using TI's Tx and Rx. This project is set up with the example channel and package and is set up to run with default AMI parameter settings.
sigrity_cd_project	examples/ project_kits	Folder	Contains an example IBIS-AMI model using TI's Tx and Rx. This project is set up with the example channel and package and is set up to run with default AMI parameter settings.
agilent_ads_project	examples/ project_kits	Folder	Contains an example IBIS-AMI model using TI's Tx and Rx. This project is set up with the example channel and package and is set up to run with default AMI parameter settings.

3 Model Usage

This section documents the various parts of the IBIS-AMI model and how to choose the parameters to run it appropriately. The EDA simulator should read the AMI file and display the parameter selections. You can then select the appropriate configuration for the simulation.

3.1 Signal Flow Diagram

This section provides an overview of a typical TI IBIS-AMI model. Figure 1 shows the signal flow diagram. A typical signal chain consists of the Tx IBIS and AMI model followed by the Tx Analog model, Tx package, channel, Rx package, Rx analog model and the Rx IBIS and AMI model. Section 3.1.1 and Section 3.1.2 provide additional details on how to choose and instantiate the Tx and Rx analog models based on the AMI parameter settings. The release package comes with an example channel and package models and projects kits for various EDA simulators. Section 3.1.3 explains differences between the typical signal flow diagram shown in Figure 1 and the implementation currently followed in various EDA simulators. This signal flow diagram is applicable only to the EDA simulators that are listed in Section 1.3. If you are using an EDA simulator that is not listed in Section 1.3, then this signal flow graph may not be applicable.

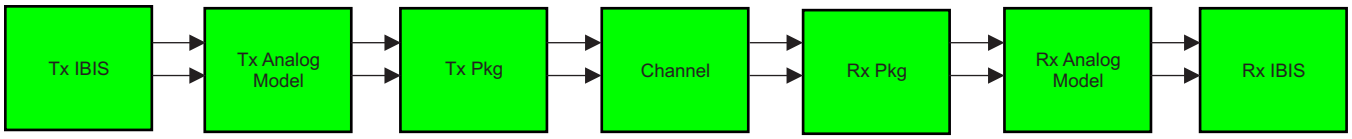

Figure 1. IBIS-AMI Model Signal Flow Diagram

3.1.1 Transmitter Analog Model

The transmit analog model is a s4p file located in the tx_ibis_ami/tx_analog_model folder of the release. This file models the transmitter bandwidth limits and the termination characteristics. This file follows the following naming convention:

```
draco_40nm_*_ txdac_<corner>.s4p
```

The selection of this model depends on the model specific parameter *corner* selected in the Tx AMI model. As an example, if the field *corner* is selected to be ‘n_0p90_1p50_027_nomR’, then the transmit analog model s4p file should be:

```
draco_40nm_*_ txdac_nom_0p900V_25deg.s4p
```

The port ordering for the transmit analog model is shown below. Sometimes the default port ordering assigned by the EDA simulator may be incorrect and needs to be manually changed to reflect the correct port order.

```
     1 ----xxxxxxxxxxxxxxxxx---- 3
           xxxxxxxxxxxxxxxxx
(Input)    xxxxxxxxxxxxxxxxx  (Output)
           xxxxxxxxxxxxxxxxx
     2 ----xxxxxxxxxxxxxxxxx---- 4
```

3.1.2 Receiver Analog Model

The receive analog model is a s4p/s2p file located in the rx_ibis_ami/rx_analog_model folder of the release. This file models the receiver termination characteristics and uses the following naming convention:

```
draco_40nm_*_ rxterm_<corner>.s4p/s2p
```

The selection of this model depends on the model-specific parameter *corner* selected in the Rx AMI model. For example, if the field *corner* is selected to be *n_0p90_1p50_027_nomR*, then the receive analog model file should be:

```
draco_40nm_*_ rxterm_n_0p90_1p50_027_nomR.s4p/s2p
```

The port ordering for the 4 port receive analog model is shown below. Sometimes the default port ordering assigned by the EDA simulator may be incorrect and needs to be manually changed to reflect the correct port order.

```
     1 ----xxxxxxxxxxxxxxxxxx---- 3
           xxxxxxxxxxxxxxxxxx
(Input)    xxxxxxxxxxxxxxxxxx  (Output)
           xxxxxxxxxxxxxxxxxx
     2 ----xxxxxxxxxxxxxxxxxx---- 4
```

3.1.3 EDA Simulator Specific Information

The signal flow implemented by various EDA Simulators may differ in one or more ways from the one shown in Figure 1. Only the differences that are currently known are stated below. If no statement is made about a specific EDA Simulator listed in Section 1.3, then it can be assumed to follow the signal flow diagram shown in Figure 1.

- SiSoft QCD does not require the Tx Analog model to be explicitly provided in the schematic. The Tx Analog model gets defined in the IBIS file and can be selected using the model selector in the schematic. For more details on how to make this choice, see Appendix B.

3.2 Transmitter AMI Model Parameters

This section provides a detailed description of each transmit AMI parameter, their range of values and its default value. The default value for each parameter is chosen to reflect the normal mode of operation of the SerDes. Parameters that describe design impairments, such as the process *corner* are chosen to default its worst performing value. Table 3 shows the various Tx AMI model parameters and their description.

Table 3. Tx AMI Model Parameters

Parameter Name	Range of Values	Default Value	Description
corner	nom_0p900V_25deg st_1p000V_m40deg w_0p765V_125deg	w_0p765V_ 125deg	Process, voltage, temperature and poly resistance corner affecting the Tx design impairment. The naming convention is: <process>_<VDDT>_<VDDR>_<temperature>_<resistance> nom_0p900V_25deg: nominal process, voltage, temperature st_1p000V_m40deg: strong process, high voltage, low temperature w_0p765V_125deg: weak process, low voltage, high temperature
de	Codes [0:15] = values [0%: -4.76% :-71.42%]	0	This parameter specifies the TX de-emphasis tap settings. Valid range of settings is codes 0 to 15. These values correspond to pre-cursor tap weights from 0 to -71.42%
swing	0,1,3,5,8,10,12,13,14,15	15	Transmitter output swing. The mapping between the swing code and the transmitter output signal amplitude is shown below: **Swing Code** / **Max_Rail (mv) (Single Ended Peak-Peak)** 0 / 120/2 1 / 200/2 2 / 280/2 3 / 360/2 4 / 440/2 5 / 530/2 6 / 610/2 7 / 690/2 8 / 770/2 9 / 850/2 10 / 960/2 11 / 1010/2 12 / 1090/2 13 / 1170/2 14 / 1230/2 15 / 1330/2

3.3 Receiver AMI Model Parameters

This section provides a detailed description of each receive AMI parameter, their range of values and its default value. The default value for each parameter is chosen to reflect the normal mode of operation of the SerDes. Parameters that describe design impairments, such as process *corner* and *RX_jitter_DCD* are chosen to default its worst performing value. Table 4 shows the various Rx AMI model parameters and their description.

Table 4. Rx AMI Model Parameters

Parameter Name	Range of Values	Default Value	Description
corner	n_0p90_1p50_027_nomR s_0p90_1p50_125_minR s_1p10_1p98_m40_minR	s_0p90_1p50_125_minR	Process, voltage, temperature and poly resistance corner affecting the Rx design impairment. The naming convention is: <process>_<VDDT>_<VDDR>_<temperature>_<resistance> n_0p90_1p50_027_nomR : nominal process, voltage, temperature and poly resistance s_0p90_1p50_125_minR : strong process, nominal voltage, high temperature and low poly resistance s_1p10_1p98_m40_minR : strong process, high voltage, low temperature and low poly resistance
invpair	Auto Detect Normal Polarity Invert Polarity	Normal Polarity	'invpair' value: Schematic Connection 'Normal Polarity': txp -- rxp; txn -- rxn 'Invert Polarity': txp -- rxn; txn -- rxp In case the invpair value and schematic connection are set incorrectly, TI will print out a warning *Check RX input polarity* in the *.report file
CDR_phase	Adapt or 0:48	Adapt	Rx CDR Phase code. The default value is set to Adapt (-1), which lets the Rx CDR adapt to the incoming data. If you choose any value between 0 and 48, the Rx CDR is locked to the chosen phase code. 24 picks the center of the eye.
RX_jitter_DCD	4ps	4ps.	Rx Duty Cycle Distortion (DCD) value. Default value is worst case DCD value derived based on circuit simulations.
RX_jitter_RJ	0.8ps	0.8 ps.	RX Random Jitter. Defines the standard deviation of the guassian distribution. Default value is worst case random jitter value derived based on circuit simulations.
anlg_eql_gaincode	Adapt or 0:16	Adapt	Rx Analog equalizer gaincode. The default value is set to Adapt which lets the analog equalizer adapt to the optimal setting. If the user chooses any value between 0 and 16, then the equalizer will NOT adapt and the chosen gain code value will be applied. 0 is Max Equalization and 16 is No Equalization.
anlg_eql_zerocode	0:7	0	0 is Max Zero Frequency (365 MHz) and 7 is Min (50 MHz) Zero Frequency.
anlg_eql_gainboost	0,1	0	0 is gain boost turned off and 1 is gain boost turned on
DLLid	ti_draco_40nm_ami_rx	ti_rincewind_40nm_ami_rx	This name is used to generate a file named *ti_draco_40nm_ami_rx.report*, which contains the simulation results.

3.3.1 Statistical Mode of Simulation

You need to enable the statistical mode of simulation in the EDA Simulator. The statistical mode of simulation performs the following:

- TX Model: Applies a 3-tap FIR filter with transmit bandwidth and return loss
- RX Model: Applies an analog equalizer (peaking filter) transfer function and its non-linear behavior along with the receiver input return loss. The choice of the analog equalizer transfer function is made using a statistical optimization algorithm. This algorithm has been correlated to the time domain analog equalizer adaption algorithm implemented in silicon.

This mode of simulation is available to help you perform fast simulations over a wide range of simulation conditions and channels and then pick appropriate macro settings to perform more detailed and accurate time domain simulation. This simulation mode is not intended to replace the time domain simulation.

3.4 Simulation Settings

The common simulation settings required for each time domain simulation are listed below. These are values:

- Over sampling ratio/Samples per bit: 8
- Block size (number of bits per block): 2000
- Minimum Ignore Bits: 10000
- Total number of simulated bits: $2E^6$

3.5 Interpreting Simulation Results

The simulations results are available in the results dir, in the file named *ti_draco_40nm_ami_rx.report*. The name of this file is different for SiSoft QCD simulator, which typically follows the format of *qcd_*.*.dll.report*. This file is generated every time you run a time domain simulation. The location of the results directory varies with each EDA simulator. You can check Section B.6 to see where the results directory gets created for each EDA Simulator.

```
Report "Adapted EQ GainCode" 3
```

This means that the receiver analog equalizer adapted to a gain code of 3.

In case the invpair value and schematic connection are set incorrectly, TI will print out the warning *Check RX input polarity* in the *.report file.

3.6 Reporting Errors

In the event that you receive errors while running the model, you should contact TI at serdes_matlab@list.ti.com. When contacting TI, you should include the following information:

- The name of the release archive, ti_draco_40nm_ami_v1p9.zipand where the release was downloaded from (iCDDS, FTP, and so forth)
- The error message that pops out of the EDA Simulator. A snapshot of the windows screen with the error message would be useful.
- Log files that are generated in the results directory. If any or none of these files are not generated, then you should report that. The location of the files is EDA Simulator dependent. For tool-specific information, see Section B.6. The files that you should send TI are:
  - draco_AMI_TX_Init_logfile
  - draco_AMI_RX_Init_logfile
  - ti_draco_40nm_ami_rx.report

- Additional files generated specific to each EDA Simulator. For tool-specific information, see Section B.6.

4 Model Limitations

The model has the following limitations and known issues:

- For meaningful simulation results, the number of bits simulated per block should be at least 2000.
- For meaningful simulation results, the `ignore_bits` parameter should be set to 10000 bits to allow time for the various adaptive loops in the receiver to converge.
- Every block should be an integer number of bits.

5 Frequently-Asked Questions (FAQ)

For help with the Platform, see the following list of questions and answers.

Question:

What is an MCR, and do I need to use it?

Answer:

MATLAB Compiler Runtime (MCR) library enables you to use all the core functionality of MATLAB while executing the Platform without having to install MATLAB. If you do not have MATLAB version installed on your machine, and if you have not yet installed the MCR for MATLAB version R2008b (included in this release), then you DO need to perform this one-time installation of the MCR. For the steps to install the MCR, see Appendix A.

Question:

I see this error pop up when I try to run TI's AMI models. How do I fix it?

Figure 2. Unable to Locate Component

Answer:

The most likely reason for this to occur is due to incorrect installation of MATLAB or the MCR. For detailed steps on the MCR installation or to check the MATLAB installation, follow the detailed procedure in Section 3.1.

NOTE: For all other questions, contact the SerDes Simulation Platform customer support list at serdes_matlab@list.ti.com.

Appendix A MATLAB Compiler Runtime Installation

This section guides you through the installation of the MCR. This is a one-time installation step that is required to run any TI IBIS-AMI models. If you have already installed the MATLAB R2008b MCR library, you can skip this step.

A.1 General Installation Steps

1. Open the installation archive using the Winzip utility in Windows.

   The AMI model executes compiled MATLAB code. Therefore, the MATLAB Component Runtime (MCR) library is required to be installed on the target system before executing the AMI model. Note that the MCR version must match with the version for which the AMI model is compiled.

   The current MATLAB MCR version required is R2008b. If you do not have this version of the MCR installed, see Section A.2 for installation steps.

   It is possible to share the MCR installation for multiple AMI models. However, all these AMI models must use the same MCR/MATLAB version.
2. Add the MCR path [mcr root]\v79\runtime\win32 to the windows environment variable PATH. This is a one-time installation step and need not be done for model updates with the same MCR version. This can be done by:

   (a) Opening the control panel.

   (b) Select *System* → *Advanced* tab → *Environment Variables* button. Look for the PATH variable under the system variables and add this directory to it.

A.2 Installing the MATLAB Compiler Runtime (MCR) Library

If you have never installed the MATLAB Compiler Runtime (MCR) library on the machine you use to run your simulations, then this must be done once prior to running your IBIS-AMI simulator. To install the MCR, follow these steps:

1. Run MCRInstaller.exe on the machine you use to run your simulations. This brings up an installation wizard, which looks similar to Figure 3. Click *Next* to continue.

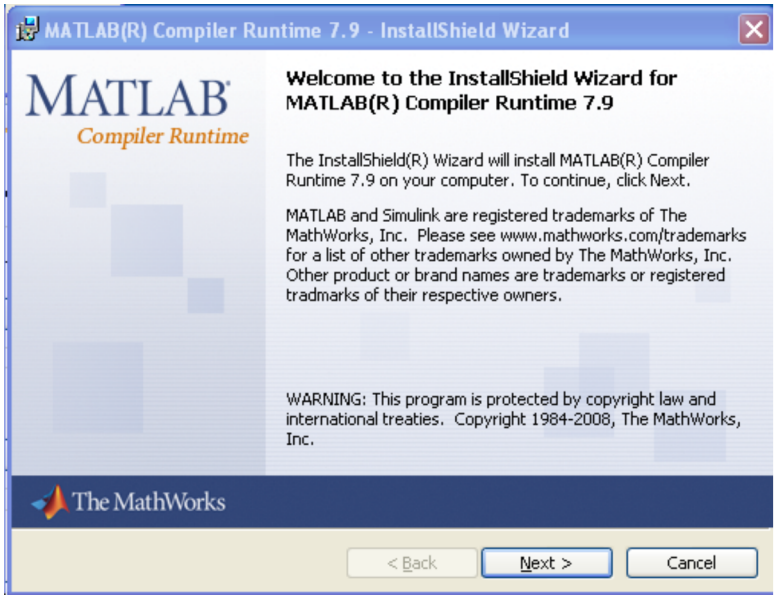

Figure 3. MCR Installer GUI

2. Enter your customer information, as shown in Figure 4. Click *Next* to continue.

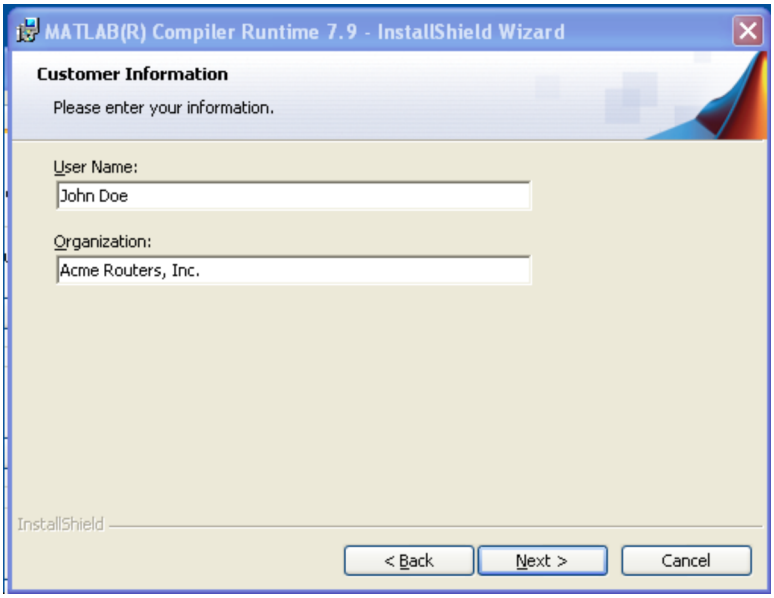

Figure 4. MCR GUI, Specifying User Information

3. Specify where you want to install the MCR library, as shown in Figure 5. Click *Next* to continue.

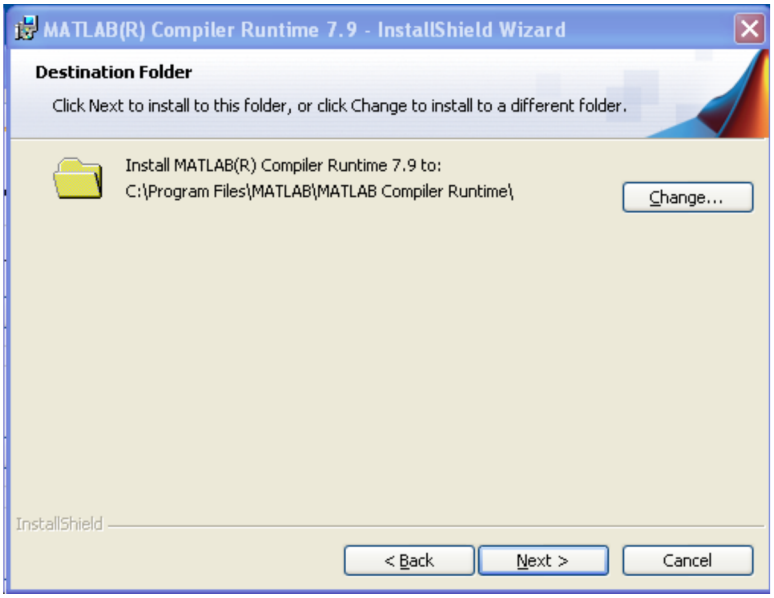

Figure 5. MCR GUI, Specifying Install Location

4. Click *Install* or *Back* to change your installation settings, as shown in Figure 6.

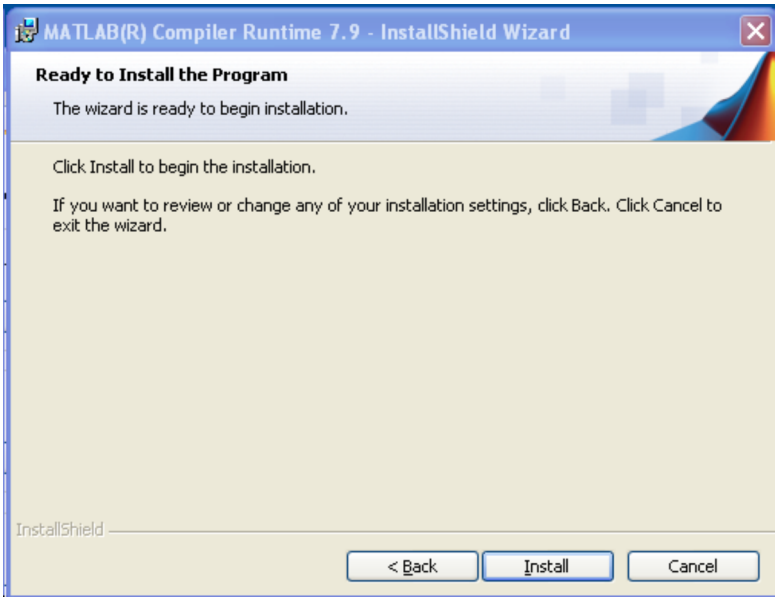

Figure 6. MCR GUI, Ready to Begin Installation

The MCR Installer will begin installing in the specified directory. This may take around 10 minutes to complete. During the installation, the GUI will show the installation status with a progress bar.

5. You will receive a *Finish* screen similar to Figure 7 when the Installer GUI and the MCR installation is complete.

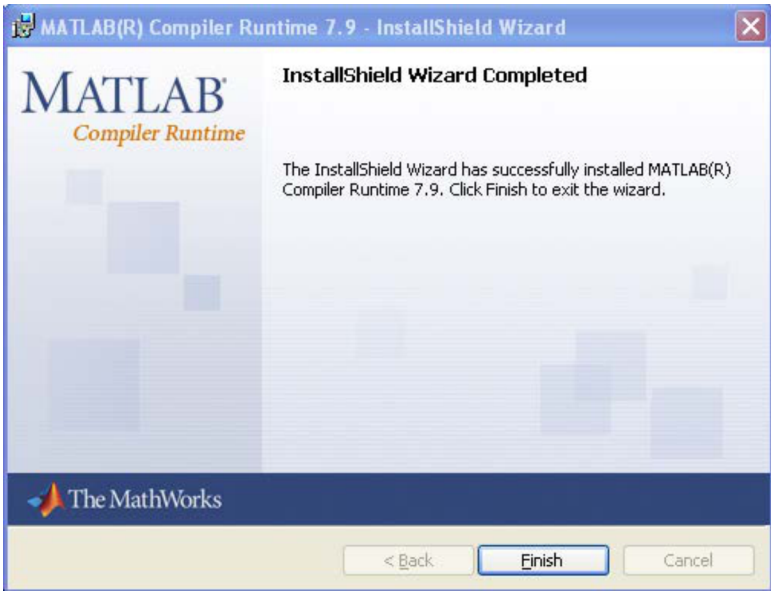

Figure 7. MCR GUI, Finish Screen

Appendix B Guide to Run TI IBIS-AMI Models With SiSoft Quantum Channel Designer

This section provides an overview of running the TI IBIS-AMI models using Sisoft's Quantum Channel Designer software. For instructions on how to install the software, see the Sisoft's website. A demo project kit named *sisoft_qcd_project* is also included as part of the release contents in the examples and project_kits folder of the release.

B.1 Creating a New Project

To create a new project in QCD:

1. Go to *File* → *Project* → *New Project*.
2. Fill in the project details in the GUI that pops up. An example is shown in Figure 8.
3. Enter the Project Name as *QCD_Project*.
4. Enter the Interface Name as *ti_serdes*.
5. Select *OK*.

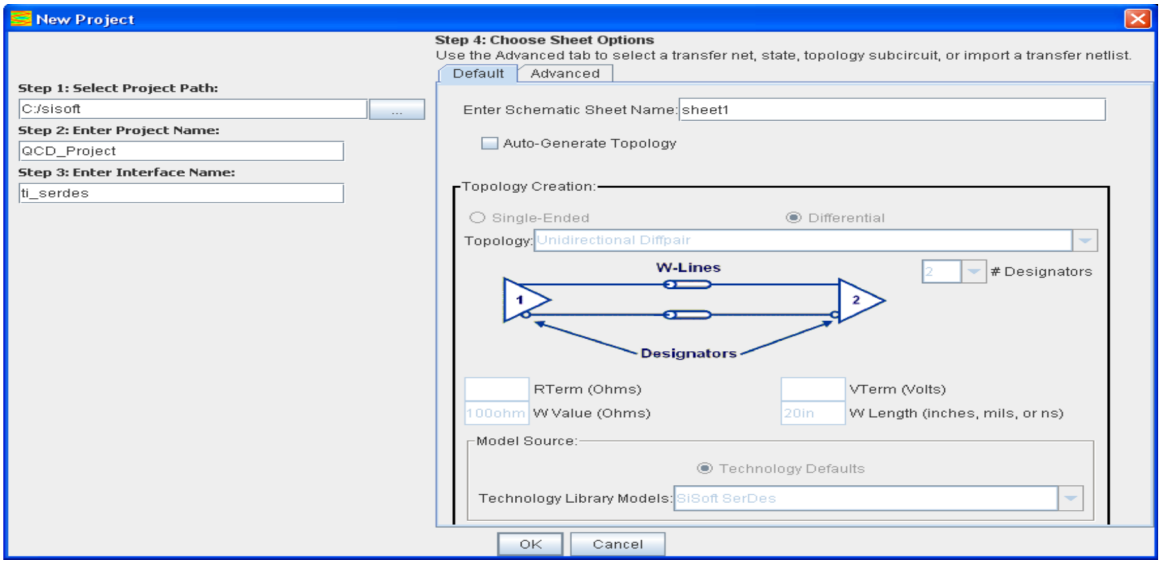

Figure 8. SiSoft QCD -- Creating a New Project

B.2 Importing IBIS and AMI Files

To import the IBIS files in QCD:

1. Go to *Libraries* → select *IBIS*.
2. Navigate to the model release directory, `ti_draco_40nm_ami_v0p96`.
3. Select the *draco_40nm_tx.ibs* file from the tx_ibis_ami folder and select *Import*.
4. Similarly, select the *draco_40nm_rx.ibs* file from the rx_ibis_ami folder and select *Import*.
5. QCD automatically copies the *.ami and *.dll files associated with the *.ibs files. These files can be found in QCD_Project\si_lib\ibis.

B.3 Importing S-Parameter Models

To import the S-parameter files in QCD:

1. Go to *Libraries → Import S-Parameter*.
2. Navigate to the model release directory, ti_draco_40nm_ami_v1p9.
3. Select all the *draco_40nm_*_txdac_*.s4p* files from the tx_ibis_ami/tx_analog_model folder and select *Import*.
4. Ignore the port order shown in the GUI. QCD assigns the correct port order of the transmit analog model based on the *nodemap* value defined in the draco_40nm_tx.ami file.
5. Similarly, select all of the *draco_40nm_*_rxterm_*.s4p* files from the rx_ibis_ami/rx_analog_model folder and select *Import.* QCD assigns the port order of the receive analog model based on the *nodemap* value defined in the draco_40nm_rx.ami file.
6. Rearranging the port order of the Tx and Rx analog models is not required as the correct port order is assigned in the ami file.
7. All of the S-parameters have now been imported into the project.

B.4 Schematic Creation

The following sections show the steps to create a schematic for simulations with and without crosstalk.

B.4.1 Simulations Without Cross Talk

1. Add a new Differential Buffer element to the panel.
2. Add a new S-Parameter element to the panel.
3. Connect the elements as shown in Figure 9.
4. Choose the appropriate simulation settings, data rate, corner, and so forth.
5. Make sure that the *Type* is set to SerDes in the T-Net properties.

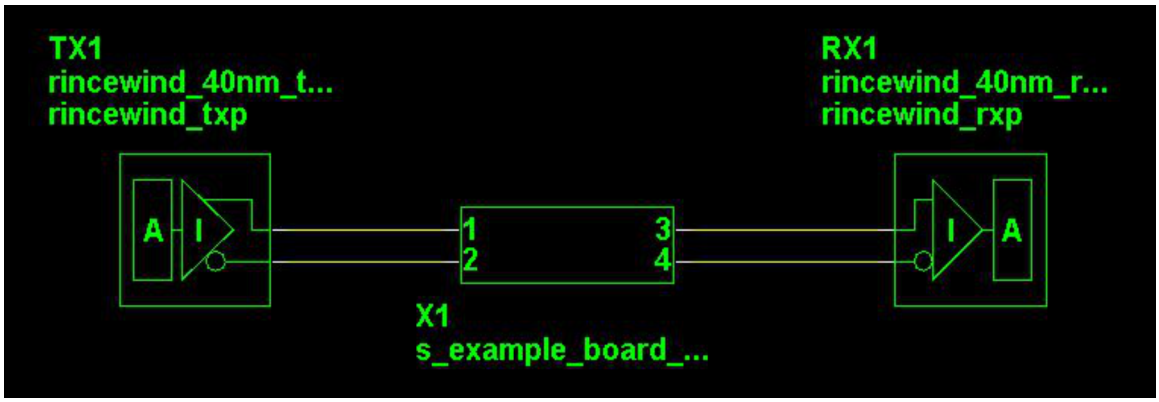

Figure 9. SiSoft QCD – Thru Simulation Schematic

B.4.2 Simulations With Cross Talk

1. Create individual schematics sheets without cross talk as shown in Section B.4.1. The number of individual schematic sheets should be equal to the sum of victims and crosstalkers intended for simulation.
2. Make sure that the transmitter and receiver in each individual schematic sheet has a unique instance name.
3. Create a new crosstalk schematic as shown in Figure 10. This schematic shows a crosstalk simulation setup with 1 victim and two aggressors in the link setup.

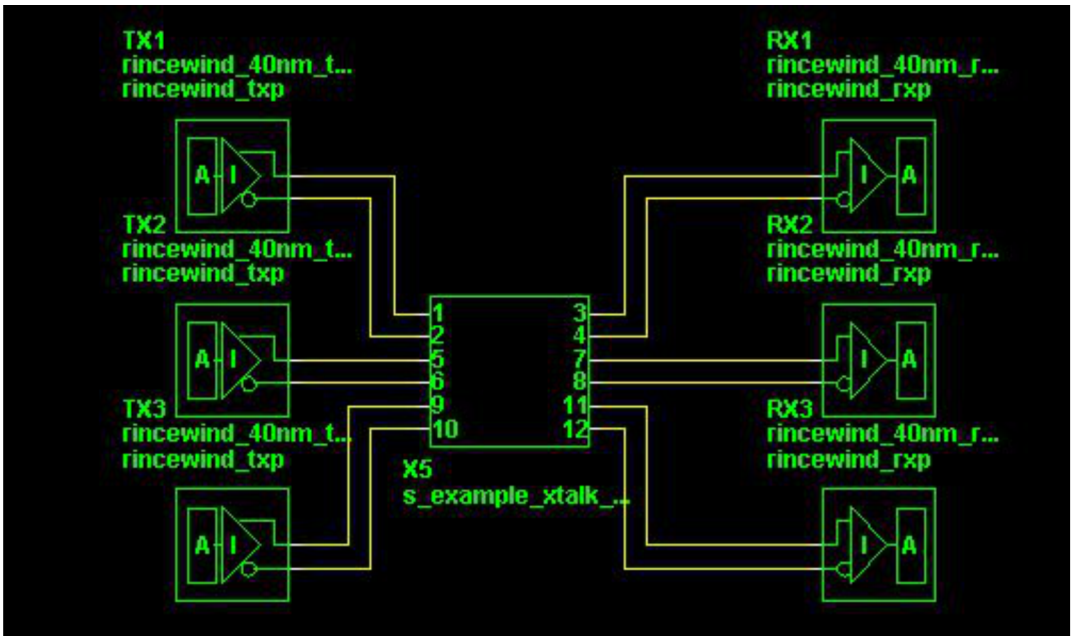

Figure 10. SiSoft QCD – Thru Simulation Schematic

4. Change the *Type* to Widebus in the T-Net properties.
5. Edit the Transfer net properties and set the transfers from the source to their corresponding targets.

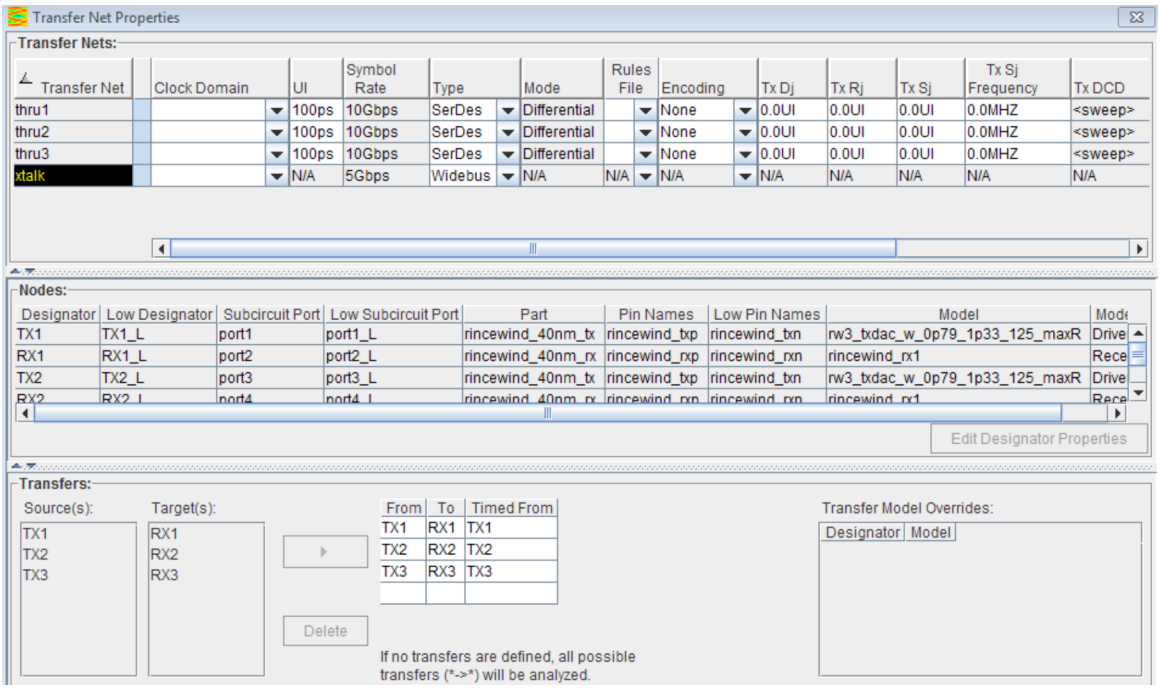

Figure 11. SiSoft QCD – Setting the Transfers in the Transfer Net Properties

B.5 Process Corner Selection in QCD

This section explains the selection of the process corner in the QCD simulator. The process corner string defined in the code in Section 3.1.1 and Section 3.1.2 are automatically mapped onto the QCD GUI as follows. Selecting the 'SS' Corner in the QCD GUI implies that you are selecting the *w_0p765V_125deg* corner for the Tx and the *s_0p90_1p50_125_minR* for the Rx. The complete mapping of the QCD Corner with the Tx and Rx corner is provided in Table 5 and Figure 12.

Table 5. SiSoft QCD – Selecting Process Corner

QCD Corner	Tx Corner	Rx Corner
SS	w_0p765V_125deg	s_0p90_1p50_125_minR
TT	nom_0p900V_25deg	n_0p90_1p50_027_nomR
FF	st_1p000V_m40deg	s_1p10_1p98_m40_minR

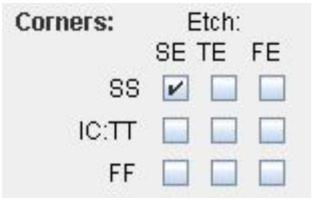

Figure 12. SiSoft QCD – Selecting Process Corner

B.6 Simulation Settings

This section explains how to modify some global simulation parameters in the EDA simulator. Some of these parameters are also explained in Section 3.4. To access the simulation settings GUI choose *Setup → Simulation Parameters*. Table 6 provides the list of parameters that need to be modified. Values not mentioned in Table 6 can be left in their default state. Note that these are recommended values and can be changed based on your judgment. There is an upper limit to the number of waveform bits that can be collected by QCD. To check this limit, see the *SiSoft QCD Software User's Guide* **(website or literature number????)**.

Table 6. SiSoft QCD Simulation Settings

Parameter Name	Recommended Value	Description
Samples Per Bit	8	Number of samples per Bit
Minimum Ignore Bits	10000	Number of bits required to reach steady state.
Record Start	1997500	Number of bits after which waveform collection begins
Record Stop	2000000	Number of bits after which waveform collection ends
Time Domain Stop	2000000	Number of bits after which time domain simulation ends
Block Size	16000	Number of samples per block. The number of bits per block = Block Size and Samples per Bit.
Clock Mode	Normal	In this mode, the data eye and clock PDF are calculated and presented separately.
Time domain Crosstalk mode	Explicit	This value should be set to explicit during the cross talk simulations

B.7 Simulation Results

The simulation results can be found in the .\QCD_Project\interfaces\ti_serdes\pre_sims\sheet1\default.ssm\qcd\qcd_*.*.dll.report file. The contents of this file are explained in Section 3.5.

B.8 Reporting Errors

In the event that you receive errors while running the model, you should contact TI at serdes_matlab@list.ti.com. When contacting TI, you should include the following information:

- The name of the release, archive ti_draco_40nm_ami_v1p9.zip and where the release was downloaded from (CDDS, FTP, and so forth).
- The error message that pops up from the EDA Simulator. A snapshot of the windows screen with the error message would be useful.
- The log files that are generated in the results directory. If any or none of these files are not generated, then you should report that. The location of the files is *.\QCD_Project\interfaces\ti_serdes\pre_sims\sheet1\default.ssm\qcd*. The files that you should send TI are:
  - draco_AMI_TX_Init_logfile
  - draco_AMI_RX_Init_logfile
  - all files in the draco_RX_figures directory
  - qcd_*.*.dll.report
  - qcd_*.info

Appendix C Opal Quality Statement

This section provides a summary of the Opal requirements and recommendations and TI's compliance to them. Opal is a set of guidelines for the quality, usability, portability, performance, and functionality of IBIS-AMI models. For more detailed information, see the Opal website.

Table 7. Opal Requirements

Requirement Number	Page Number	Requirement Description	TI Compliance
R2.1_A	pg 9	All files are distributed in a single archive	Y
R2.1_F	pg 10	Model installation directory independent of execution directory	Y
R2.2.1_A	pg 11	Support Windows and Linux	Y
R2.2.2_A	pg 11	Multiple instances of one model in one simulation/analysis	Y
R2.2.2_B	pg 11	Multiple instance of multiple models in one simulation/analysis	Y
R2.2.2_C	pg 12	Multiple simultaneous simulations/analyses	Y
R2.3_E	pg 18	Support Samples_Per_Bit parameter, if necessary	Y
R2.5_A	pg 21	Complete parameter declaration	Y
R2.5_C	pg 21	Useful parameter description	Y
R2.5_D	pg 21	Parameter names in model same as model names in .ami file	Y
R2.5_E	pg 21	Unrecognized parameters do not cause failure	Y
R2.5_L	pg 23	Labels consistent with List	Y
R2.7_A	pg 26	Model correlated to another behavior description	Y
R2.7_B	pg 26	Correlation conditions defined	Y
R2.7_C	pg 26	Correlation method defined	Y
R2.7_D	pg 26	Correlation criteria defined	Y
R2.8_A	pg 27	Minimum documentation requirements	Y
R3.0_A	pg 30	All parameters in dependency table declared before table	Y
R3.0_E	pg 31	Column header and all rows in dependency table have same length	Y
R3.0_F	pg 31	Dependency row value type convertible to all column types	Y
R4.0_A	pg 34	Fully IBIS compliant analog model available	Y
R4.2_B	pg 37	S parameter file ports and organization	Y
R4.2_F	pg 38	Node map consistent with S parameter file7.0 Table of Requirements	Y

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