Agilent Technologies

Advanced Design System 2011.01

Feburary 2011
Signal Integrity DesignGuide

Acknowledgments
Mentor Graphics is a trademark of Mentor Graphics Corporation in the U.S. and other countries. Mentor products and processes are registered trademarks of Mentor Graphics Corporation. * Calibre is a trademark of Mentor Graphics Corporation in the US and other countries. "Microsoft®, Windows®, MS Windows®, Windows NT®, Windows 2000® and Windows Internet Explorer® are U.S. registered trademarks of Microsoft Corporation. Pentium® is a U.S. registered trademark of Intel Corporation. PostScript® and Acrobat® are trademarks of Adobe Systems Incorporated. UNIX® is a registered trademark of the Open Group. Oracle and Java and registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. SystemC® is a registered trademark of Open SystemC Initiative, Inc. in the United States and other countries and is used with permission. MATLAB® is a U.S. registered trademark of The Math Works, Inc.. HiSIM2 source code, and all copyrights, trade secrets or other intellectual property rights in and to the source code in its entirety, is owned by Hiroshima University and STARC. FLEXlm is a trademark of Globetrotter Software, Incorporated. Layout Boolean Engine by Klaas Holwerda, v1.7 http://www.xs4all.nl/~kholwerd/bool.html . FreeType Project, Copyright (c) 1996-1999 by David Turner, Robert Wilhelm, and Werner Lemberg. QuestAgent search engine (c) 2000-2002, JObjects. Motif is a trademark of the Open Software Foundation. Netscape is a trademark of Netscape Communications Corporation. Netscape Portable Runtime (NSPR), Copyright (c) 1998-2003 The Mozilla Organization. A copy of the Mozilla Public License is at http://www.mozilla.org/MPL/ . FFTW, The Fastest Fourier Transform in the West, Copyright (c) 1997-1999 Massachusetts Institute of Technology. All rights reserved.

The following third-party libraries are used by the NlogN Momentum solver:

"This program includes Metis 4.0, Copyright © 1998, Regents of the University of Minnesota", http://www.cs.umn.edu/~metis , METIS was written by George Karypis (karypis@cs.umn.edu).

Intel@ Math Kernel Library, http://www.intel.com/software/products/mkl

SuperLU_MT version 2.0 - Copyright © 2003, The Regents of the University of California, through Lawrence Berkeley National Laboratory (subject to receipt of any required approvals from U.S. Dept. of Energy). All rights reserved. SuperLU Disclaimer: THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF

Errata The ADS product may contain references to "HP" or "HPEESOF" such as in file names and directory names. The business entity formerly known as "HP EEsof" is now part of Agilent Technologies and is known as "Agilent EEsof". To avoid broken functionality and to maintain backward compatibility for our customers, we did not change all the names and labels that contain "HP" or "HPEESOF" references.

Warranty The material contained in this document is provided "as is", and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this documentation and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license. Portions of this product include the SystemC software licensed under Open Source terms, which are available for download at http://systemc.org/ . This software is redistributed by Agilent. The Contributors of the SystemC software provide this software "as is" and offer no warranty of any kind, express or implied, including without limitation warranties or conditions or title and non-infringement, and implied warranties or conditions merchantability and fitness for a particular purpose. Contributors shall not be liable for any damages of any kind including without limitation direct, indirect, special, incidental and consequential damages, such as lost profits. Any provisions that differ from this disclaimer are offered by Agilent only.

Restricted Rights Legend U.S. Government Restricted Rights. Software and technical data rights granted to the federal government include only those rights customarily provided to end user customers. Agilent provides this customary commercial license in Software and technical data pursuant to FAR 12.211 (Technical Data) and 12.212 (Computer Software) and, for the Department of Defense, DFARS 252.227-7015 (Technical Data - Commercial Items) and DFARS 227.7202-3 (Rights in Commercial Computer Software or Computer Software Documentation).

Common Mode Impedance Simulation

This simulation finds the common mode impedance of the coupled lines using components from the multilayer library, which is based an accurate 2DEM solver. Note that the even mode impedance is twice the common-mode impedance. LineCalc shows 53.94 Ohms as Zeven, which yields a common-mode impedance of 26.97 Ohms. This schematic simulation gives similar results to LineCalc.

DCA File Import (.csv)

Selecting DCA File Import (.csv) from the DesignGuide menu opens a dialog box that lets you import comma separated value (.csv) data files that have been written by an Agilent 86100C Infiniium DCA-J wideband digital communication analyzer oscilloscope .

DDR2 Compliance Measurement

The DDR2 Compliance Measurement application provides a fast and easy way to perform DDR2 related measurements on your designs. These measurements are done using a probe which is shipped with ADS as a design kit. This probe will allow you to perform compliance testing and margin analysis to determine if your design meets a set of given standards. All the implemented measurements are based on the JEDEC [1] standard.

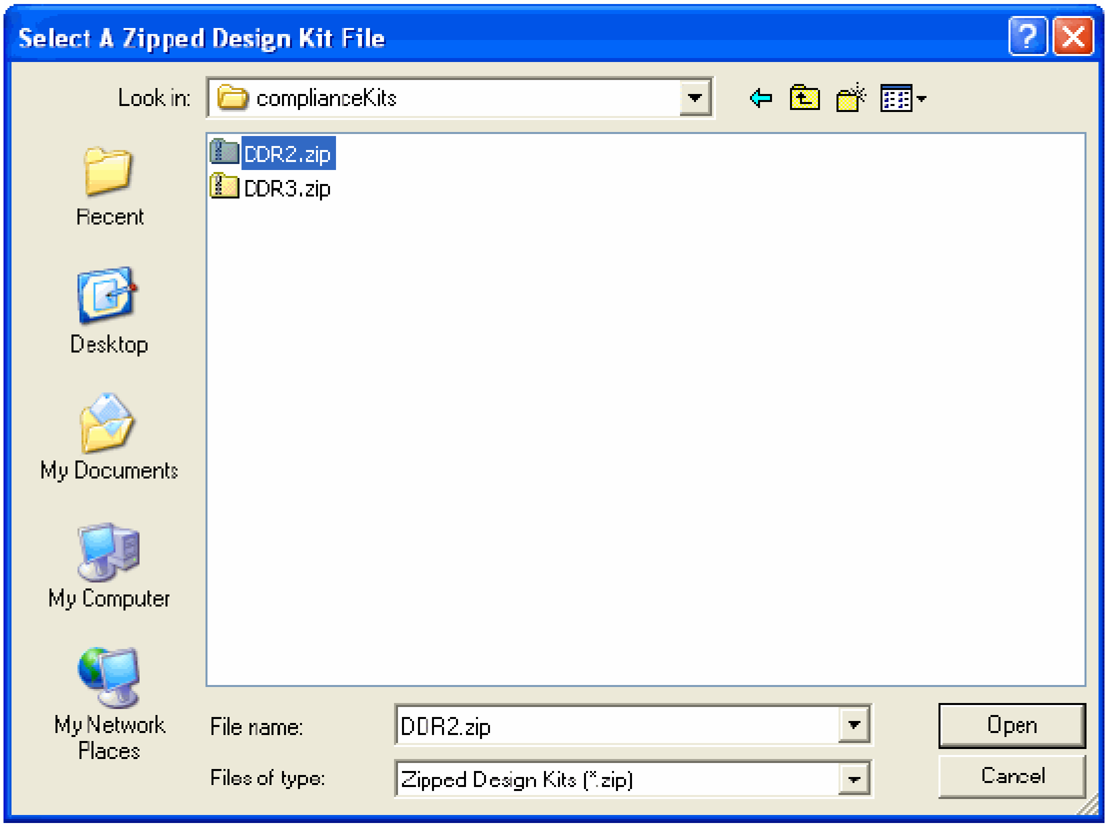
Besides providing a user-friendly interface this application is flexible enough to allow you to customize everything from the base measurement code to the compliance test standards. All the measurements are implemented in C++ which is efficient in terms of processing speed and memory usage. The measurement code is compiled on-demand into a shared library using a compiler that is shipped with ADS. The measurement compilation process is done behind the scene automatically when the source code is modified or when there is a version conflict.

> **Note**
> [1] The JEDEC (Joint Electronic Device Engineering Council) Solid State Technology Association is a semiconductor engineering standardization body of the Electronic Industries Alliance (EIA), a trade association that represents all areas of the electronic industry.

Installation

This section provides a step-by-step procedure to install the DDR2 measurement probe.

1. Open ADS workspace where *DDR2.zip* file needs to be extracted (for example, DDR2Compliance_wrk).
2. From the ADS Main window, select **DesignKits > Manage Favorite Design Kits**. The Manage Favorite Design Kits window appears.
3. Click **Add Zipped Design Kit**. This will bring up another dialog box which will allow you to select the zip file. The *DDR2.zip* file is located in your ADS installation directory under the *complianceKits* directory. Select the *DDR2.zip* file and click **Open**.



Select A Zipped Design Kit File

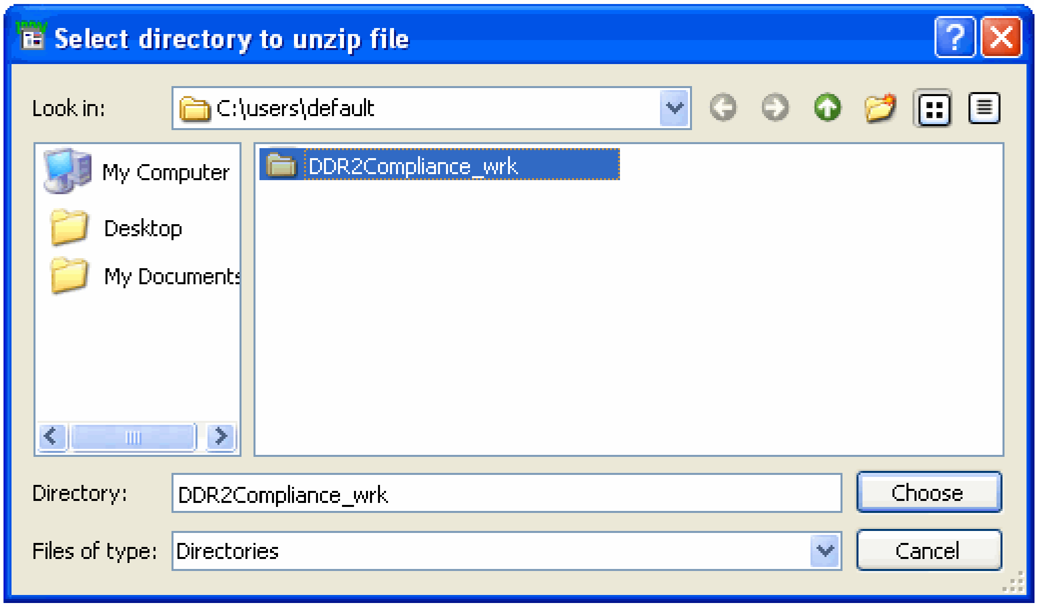
Look in: complianceKits

DDR2.zip
DDR3.zip

File name: DDR2.zip

Files of type: Zipped Design Kits (*.zip)

4. Select the workspace directory and click **Choose** to unzip the DDR2.zip file.



Select directory to unzip file

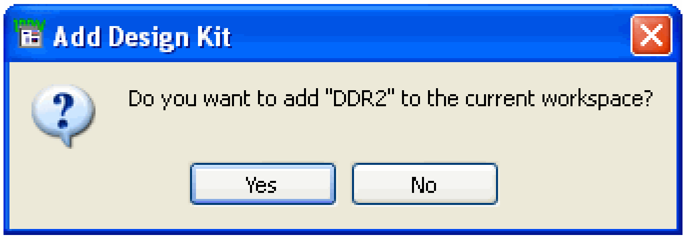
Look in: C:\users\default

DDR2Compliance_wrk

Directory: DDR2Compliance_wrk

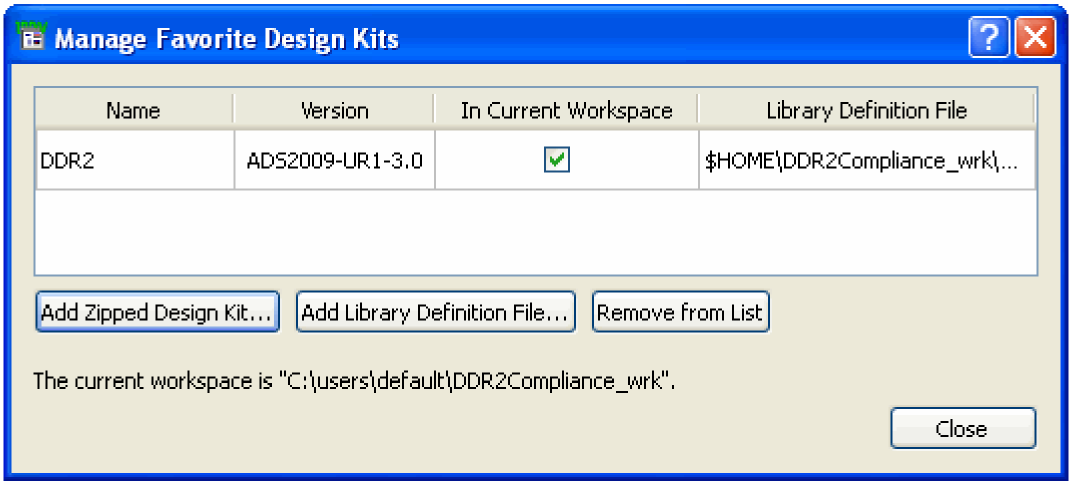
Files of type: Directories

5. Once the *DDR2.zip* file is extracted, you are prompted to add DDR2 to the current workspace. Click **Yes** to add the DDR2 compliance measurement application, else click **No**.



Do you want to add "DDR2" to the current workspace?

6. Click **Close** to close the Manage Favorite Design Kits window. The DDR2 measurement probe is now installed.



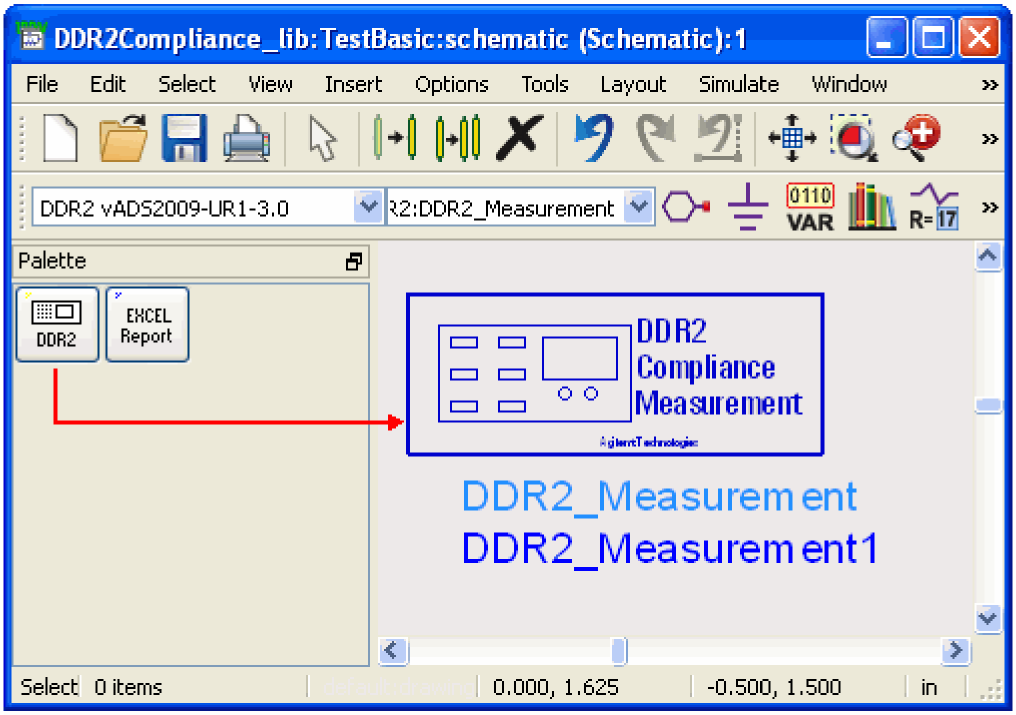
Name	Version	In Current Workspace	Library Definition File
DDR2	ADS2009-UR1-3.0	☑	$HOME\DDR2Compliance_wrk\...

The current workspace is "C:\users\default\DDR2Compliance_wrk".

Setting up for Simulation

Follow the steps below to setup DDR2 probe:

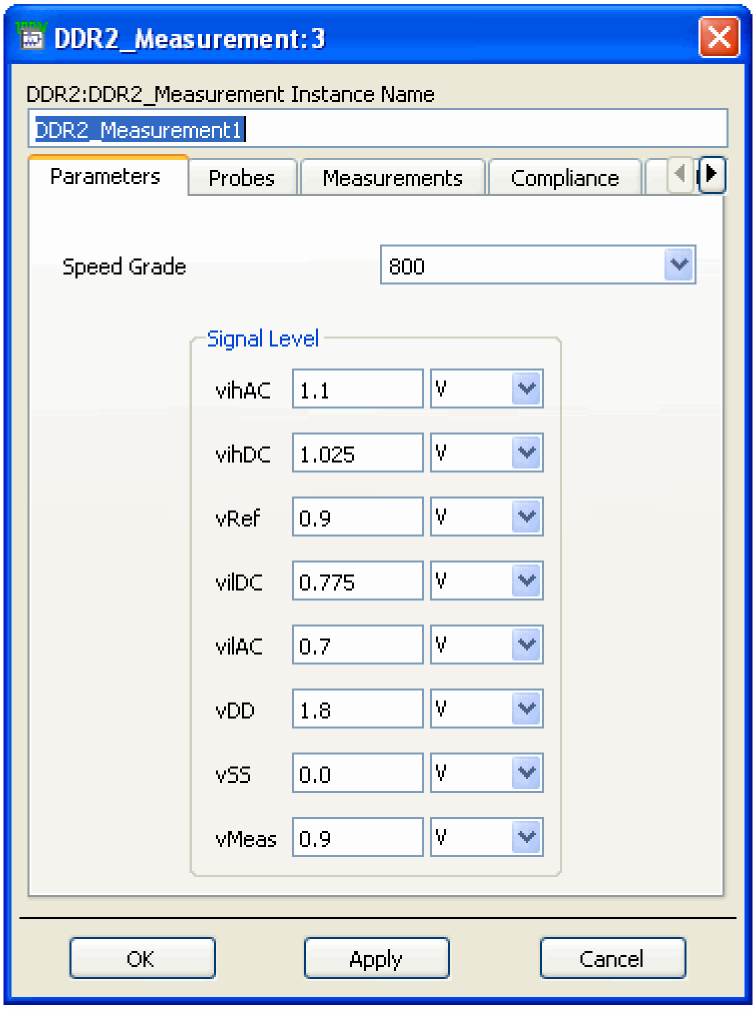
1. In the Schematic window, select the *DDR2 vADS2009-UR1-3.0* palette from the Component list.
2. Drag and drop the **DDR2_Measurement** probe component from the Palette into the Schematic area.



3. Double-click on the DDR2_Measurement probe to setup the probe. This step involves adjusting the parameters, connecting the probe, selecting the measurements, activating the compliance analysis and setting output options.

Adjusting the Parameters

There are two sets of parameters that need to be set. The first is the *Speed Grade* of the DDR2 and the second is the *Signal Level*. A detailed description of these parameters is given in the Parameters section.



DDR2_Measurement:3

DDR2:DDR2_Measurement Instance Name: DDR2_Measurement1

Speed Grade: 800

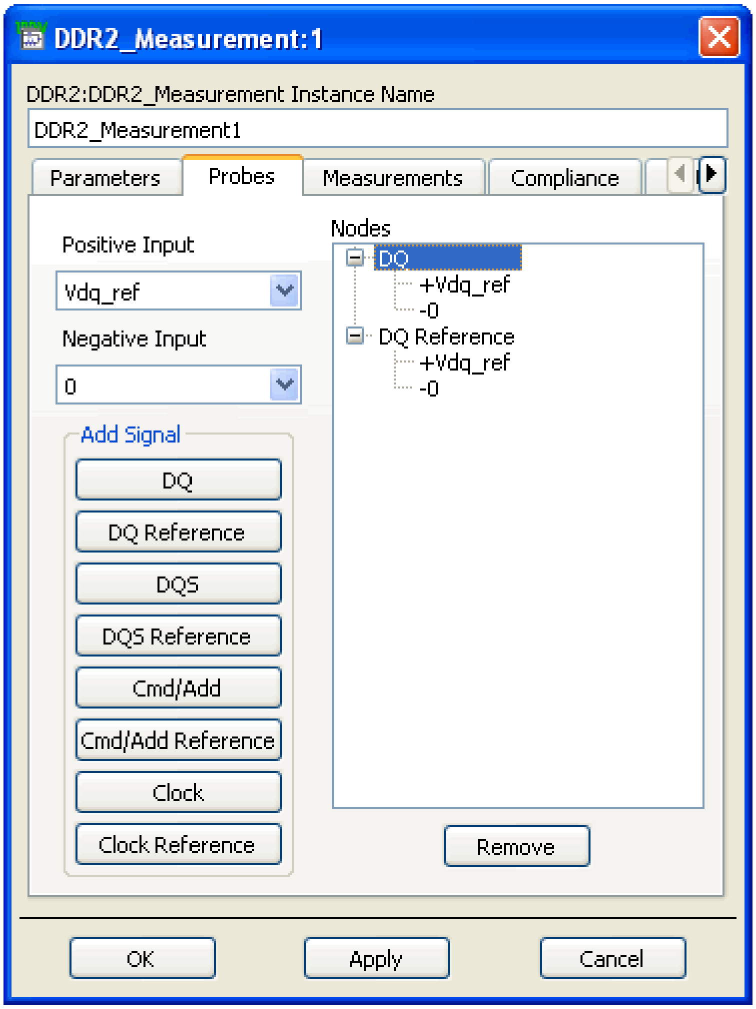
Signal Level

vihAC	1.1	V
vihDC	1.025	V
vRef	0.9	V
vilDC	0.775	V
vilAC	0.7	V
vDD	1.8	V
vSS	0.0	V
vMeas	0.9	V

Connecting the Probe

Once the parameters are set the probe needs to be connected to the circuit.

1. Select the *Positive Input* and *Negative Input* nodes in Probes tab.
2. Depending on the type of signal of the selected nodes, click on one of the buttons from the *Add Signal* pane. This should add the nodes into the tree-like structure on right hand side of the dialog box.
3. You can select the signal type and remove the node set using the **Remove** button.



DDR2:DDR2_Measurement Instance Name: DDR2_Measurement1

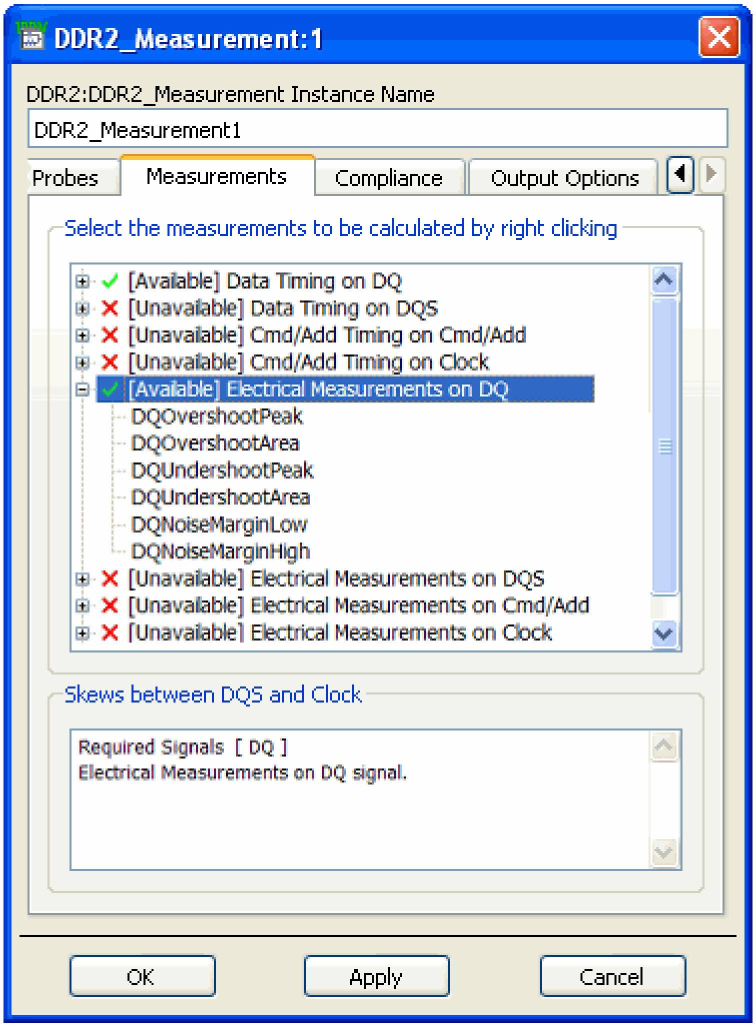
Positive Input: Vdq_ref

Negative Input: 0

Nodes
- DQ
  - +Vdq_ref
  - -0
- DQ Reference
  - +Vdq_ref
  - -0

Selecting the Measurements

Measurements can be selected by selecting the measurement group and right-clicking on the group. Note that only the measurements that are *Available* can be selected. Measurements become available depending on the type of signals that are connected to the probe. In the above example, two signal types **DQ, DQ Reference** have been selected, based on this selection only the *Data Timing on DQ* and *Electrical Measurements on DQ* are available. The required signals for a given measurement group is given in the brief description section, located towards the end of the dialog box. Each measurement group can be expanded and when you select it, the brief description section will contain an appropriate description.


DDR2_Measurement:1
Probes
Measurements
Compliance
Output Options
Select the measurements to be calculated by right clicking
Skews between DQS and Clock
OK
Apply
Cancel

DDR2:DDR2_Measurement Instance Name

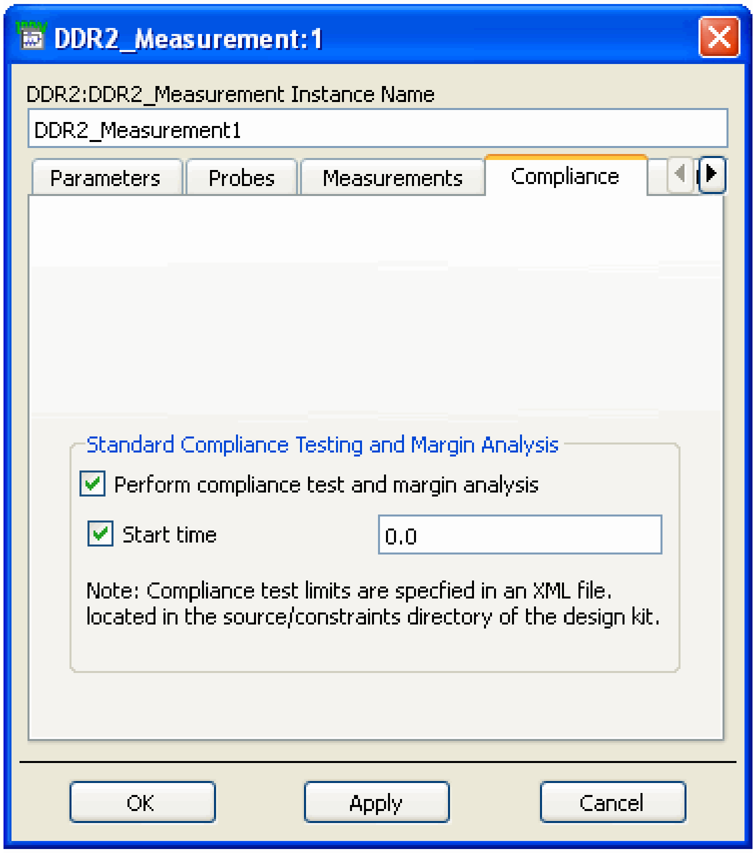
DDR2_Measurement1

- ✓ [Available] Data Timing on DQ
- ✗ [Unavailable] Data Timing on DQS
- ✗ [Unavailable] Cmd/Add Timing on Cmd/Add
- ✗ [Unavailable] Cmd/Add Timing on Clock
- ✓ [Available] Electrical Measurements on DQ
  - DQOvershootPeak
  - DQOvershootArea
  - DQUndershootPeak
  - DQUndershootArea
  - DQNoiseMarginLow
  - DQNoiseMarginHigh
- ✗ [Unavailable] Electrical Measurements on DQS
- ✗ [Unavailable] Electrical Measurements on Cmd/Add
- ✗ [Unavailable] Electrical Measurements on Clock

Required Signals [DQ]
Electrical Measurements on DQ signal.

Activating the Compliance Testing and Margin Analysis

Activating the compliance testing and margin analysis is done by selecting the *Perform compliance test and margin analysis* option. The *Start time* check box will allow you to specify the transient analysis time when measurements will be taken. This can be used to eliminate the transient response of the circuit and base the measurements on the steady state response.



DDR2:DDR2_Measurement Instance Name

DDR2_Measurement1

Standard Compliance Testing and Margin Analysis

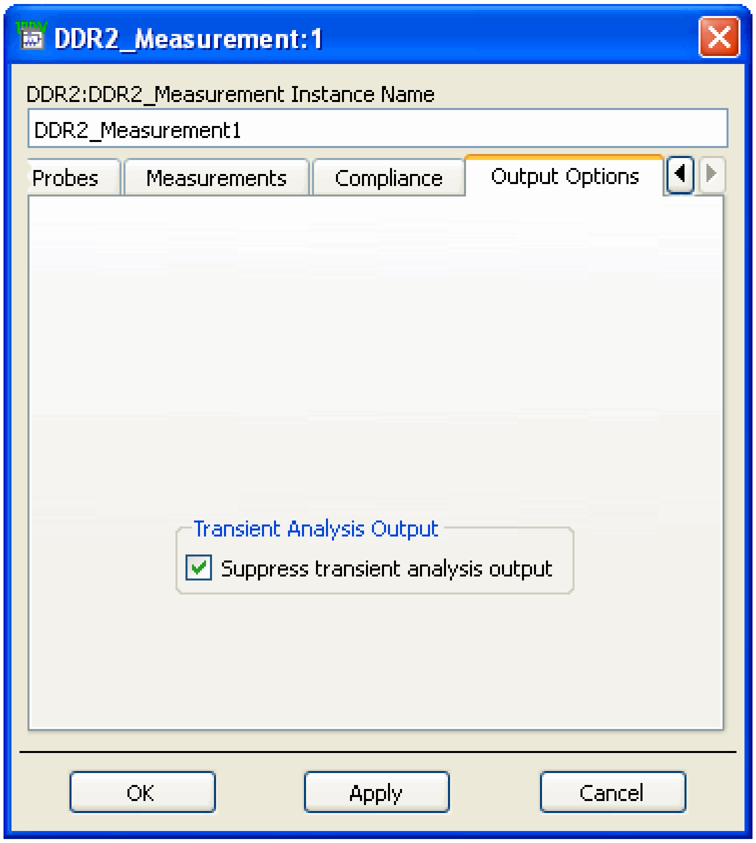
☑ Perform compliance test and margin analysis

☑ Start time 0.0

Note: Compliance test limits are specfied in an XML file.
located in the source/constraints directory of the design kit.

Setting Output Options

When you select the *Suppress transient analysis output* option, you can prevent any transient analysis waveforms from being saved.



DDR2_Measurement1

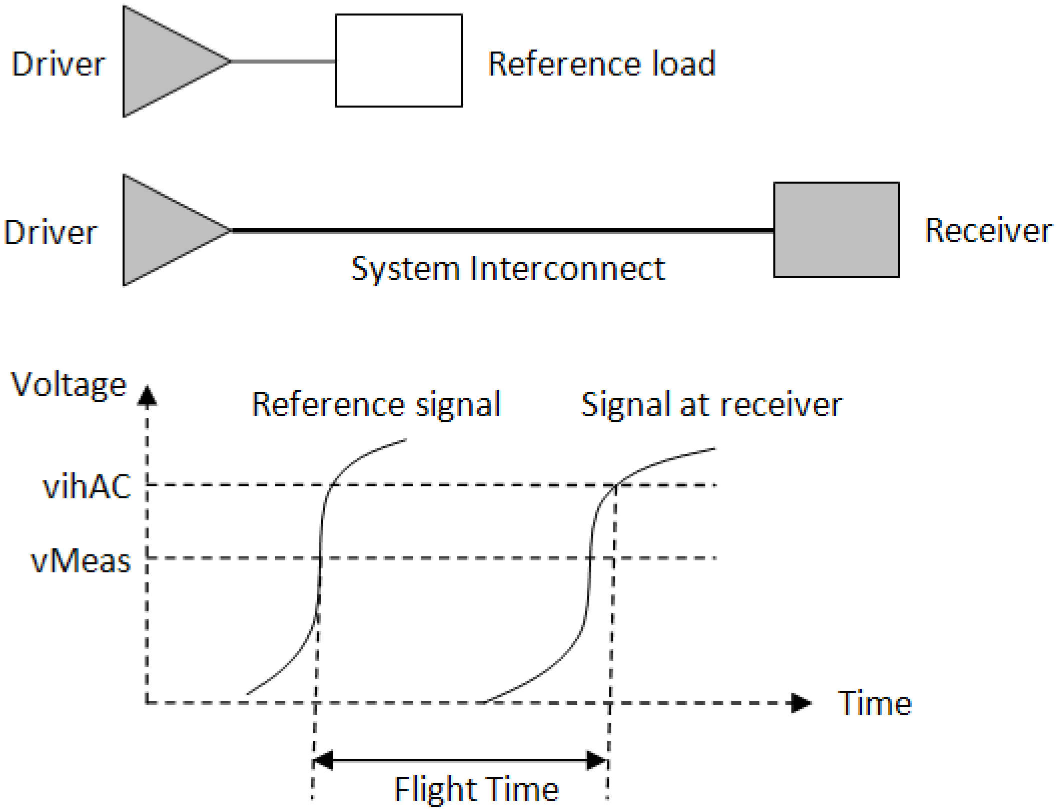
Transient Analysis Output

☑ Suppress transient analysis output

General Measurements

Flight Time

Flight time is defined as the time difference between the reference waveform and the waveform at the receiver. (Stephen H. Hall, Garrett W. Hall, and James A. McCall, *High-Speed Digital System Design*, Wiley, New York, 2000). To be specific, it is the amount of time that elapses between the point where the signal on an output buffer driving a reference load crosses the threshold voltage vMeas (defined in *Parameters* tab of DDR2 probe component) and the point where the signal crosses a threshold voltage (such as vihAC, vihDC, vRef, vilDC, vilAC in the *Parameters* tab) at the receiver in the system for the same falling or rising edge. The following figure illustrates the flight time from vMeas crossing of reference signal and vihAC crossing of signal at receiver for the same rising edge. Flight times are used later on to calculate the skews between signals.



Slew Rate

Slew Rate used in DDR2 measurements is defined in detail in *JEDEC Standard No. 79-2E* section 6 (pp. 85-100) and illustrated in Figures 85-96. In this DDR2 design kit, the accurate tangent line slew rates are measured which are required for derating value.

Derating Value and Table

The setup/hold time base values are specified based on standard slew rates (1.0 for DQ/CmdADD and single-ended DQS and 2.0 for differential DQS/Clock). For different slew rates, these base values need to be compensated by adding the derating values looked up from derating tables. Various derating tables are provided as CSV files and located in *DDR2/source/data* directory. These tables are loaded and looked up by skew measurement code, such as *DDR2/source/measurements/DQ_DQS_Skew.cxx*.

Skew with Derating

Skew is calculated as the difference between flight times of two different signals. For example, DQ_DQS skew is the difference between DQ flight time and DQS flight time. Skew is used in the equation of the most interested timing margin calculation:

```
Timing Margin = controller spec - interconnect skew - (memory spec base + derating value)
             = controller spec - (interconnect skew + derating value) - memory spec base
```

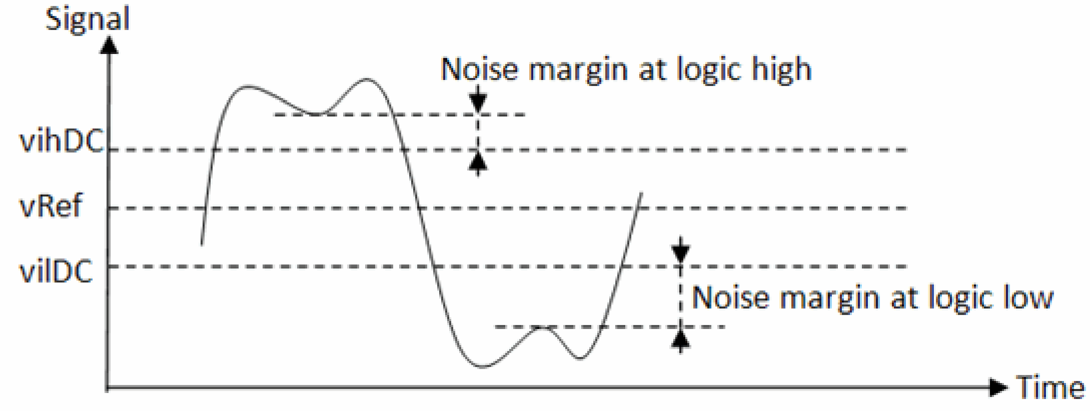
From this equation, we can see that the timing margin can also be calculated by adding the derating value to interconnect skew, instead of memory spec base. The skews calculated in this DDR2 design kit are the ones with derating value taken into account.

Overshoot and Undershoot

The overshoot/undershoot peak and area are defined in *JEDEC Standard No. 79-2E* section 6 and illustrated in Figures 75 and 76. The maximum amplitudes above/below VDD/VSS are defined as overshoot/undershoot peaks. The areas between waveform and VDD/VSS are defined as overshoot/undershoot areas.

Noise Margin

Noise margin measures how close of the signal comes to DC threshold voltage vihDC/vilDC in a logic high/low state. A signal is in a logic high state after the signal has crossed vihDC threshold in a rising edge and before it crosses vihDC in a falling edge. A signal is in a logic low state after the signal has crossed vilDC threshold in a falling edge and before it crosses vilDC in a rising edge. Basically noise margin measures how much noise can be tolerated without the signal falsely crossing vihDC/vilDC threshold in a logic high/low state. Noise margin measurement is further illustrated in the figure below.



VIX

VIX is the differential input cross point voltage and measured from the actual cross point of true and complement signals to the midlevel between VDD and VSS. It is officially defined in *JEDEC Standard No. 79-2E* section 6 and illustrated in Figure 74.

Measurements

Data Timing on DQ

Data Timing on DQ is a measure of DQ signal for Data Timing analysis. Required signals are DQ and DQ_Reference. It includes the following measurements:

Measurement type	Description	Units
DQFlightTimeSetupFall	Flight time between DQ vilAC crossing and DQ_Reference vMeas crossing in a falling edge	ps
DQFlightTimeSetupRise	Flight time between DQ vihAC crossing and DQ_Reference vMeas crossing in a rising edge	ps
DQFlightTimeHoldFall	Flight time between DQ vihDC crossing and DQ_Reference vMeas crossing in a falling edge	ps
DQFlightTimeHoldRise	Flight time between DQ vilDC crossing and DQ_Reference vMeas crossing in a rising edge	ps
DQSlewRateSetupFall	Tangent line slew rate between DQ vRef and vilAC crossings in a falling edge	V/ns
DQSlewRateSetupRise	Tangent line slew rate between DQ vRef and vihAC crossings in a rising edge	V/ns
DQSlewRateHoldFall	Tangent line slew rate between DQ vihDC and vRef crossings in a falling edge	V/ns
DQSlewRateHoldRise	Tangent line slew rate between DQ vilDC and vRef crossings in a rising edge	V/ns
DQFlightTimeReadFallMin	Flight time between the first DQ vRef crossing and DQ_Reference vMeas crossing in a falling edge	ps
DQFlightTimeReadFallMax	Flight time between the last DQ vRef crossing and DQ_Reference vMeas crossing in a falling edge	ps
DQFlightTimeReadRiseMin	Flight time between the first DQ vRef crossing and DQ_Reference vMeas crossing in a rising edge	ps
DQFlightTimeReadRiseMax	Fight time between the last DQ vRef crossing and DQ_Reference vMeas crossing in a rising edge	ps

Data Timing on DQS

Data Timing on DQS is a measure of DQS signal for Data Timing analysis. Required signals are DQS and DQS_Reference. It includes the following measurements:

Measurement type	Description	Units
DQSFlightTimeFallVih	Flight time between single-ended DQS vihDC crossing and DQS_Reference vMeas crossing in a falling edge	ps
DQSFlightTimeFallVref0	Flight time between DQS zero crossing (vRef for single-ended) and DQS_Reference zero crossing (vMeas for single-ended) in a falling edge	ps
DQSFlightTimeFallVil	Flight time between single-ended DQS vilAC crossing and DQS_Reference vMeas crossing in a falling edge	ps
DQSFlightTimeRiseVil	Flight time between single-ended DQS vilDC crossing and DQS_Reference vMeas crossing in a rising edge	ps
DQSFlightTimeRiseVref0	Flight time between DQS zero crossing (vRef for single-ended) and DQS_Reference zero crossing (vMeas for single-ended) in a rising edge	ps
DQSFlightTimeRiseVih	Flight time between single-ended DQS vihAC crossing and DQS_Reference vMeas crossing in a rising edge	ps
DQSSlewRateFall	Slew rate between DQS 2(vihDC-vRef) crossing (vihDC for single-ended) and 2(vilAC-vRef) crossing (vilAC for single-ended) in a falling edge	V/ns
DQSSlewRateRise	Slew rate between DQS 2(vilDC-vRef) crossing (vilDC for single-ended) and 2(vihAC-vRef) crossing (vihAC for single-ended) in a rising edge	V/ns

Cmd/Add Timing on Cmd/Add

Cmd/Add Timing on Cmd/Add is a measure of Command and Address signal for Command and Address Timing analysis. Required signals are CmdAdd and CmdAdd_Reference. It includes the following measurements:

Measurement type	**Description**	**Units**
CmdAddFlightTimeSetupFall	Flight time between CmdAdd vilAC crossing and CmdAdd_Reference vMeas crossing in a falling edge	ps
CmdAddFlightTimeSetupRise	Flight time between CmdAdd vihAC crossing and CmdAdd_Reference vMeas crossing in a rising edge	ps
CmdAddFlightTimeHoldFall	Flight time between CmdAdd vihDC crossing and CmdAdd_Reference vMeas crossing in a falling edge	ps
CmdAddFlightTimeHoldRise	Flight time between CmdAdd vilDC crossing and CmdAdd_Reference vMeas crossing in a rising edge	ps
CmdAddSlewRateSetupFall	Tangent line slew rate between CmdAdd vRef and vilAC crossings in a falling edge	V/ns
CmdAddSlewRateSetupRise	Tangent line slew rate between CmdAdd vRef and vihAC crossings in a rising edge	V/ns
CmdAddSlewRateHoldFall	Tangent line slew rate between CmdAdd vihDC and vRef crossings in a falling edge	V/ns
CmdAddSlewRateHoldRise	Tangent line slew rate between CmdAdd vilDC and vRef crossings in a rising edge	V/ns

Cmd/Add Timing on Clock

Cmd/Add Timing on Cmd/Add is a measure of Clock signal for Command and Address Timing analysis. Required signals are Clock and Clock_Reference. It includes the following measurements.

Measurement type	**Description**	**Units**
ClockFlightTimeFall	Flight time between Clock zero crossing and Clock_Reference zero crossing in a falling edge	ps
ClockFlightTimeRise	Flight time between Clock zero crossing and Clock_Reference zero crossing in a rising edge	ps
ClockSlewRateFall	Slew rate between Clock 0.2V and -0.2V crossings in a falling edge	V/ns
ClockSlewRateRise	Slew rate between Clock -0.2V and 0.2V crossings in a rising edge	V/ns

Electrical Measurements on DQ

Electrical Measurements on DQ requires signal DQ. It includes the following measurements:

Measurement type	Description	Units
DQOvershootPeak	DQ peak amplitude above vDD	V
DQOvershootArea	Area of DQ signal above vDD	V-ns
DQUndershootPeak	DQ peak amplitude below vSS	V
DQUndershootArea	Area of DQ signal below vSS	V-ns
DQNoiseMarginLow	The closeness of DQ coming to vilDC in a logic low state	V
DQNoiseMarginHigh	The closeness of DQ coming to vihDC in a logic high state	V

Electrical Measurement on DQS

Electrical Measurement on DQS requires signal DQ. It includes the following measurement:

Measurement type	Description	Units
DQSVIX	DQS differential input cross point voltage relative to vDD/2	mV

Electrical Measurements on CmdAdd

Electrical Measurements on Command or Address signal requires signal CmdAdd. It includes the following measurements:

Measurement type	Description	Units
CmdAddOvershootPeak	CmdAdd peak amplitude above vDD	V
CmdAddOvershootArea	Area of CmdAdd signal above vDD	V-ns
CmdAddUndershootPeak	CmdAdd peak amplitude below vSS	V
CmdAddUndershootArea	Area of CmdAdd signal below vSS	V-ns
CmdAddNoiseMarginLow	The closeness of CmdAdd coming to vilDC in a logic low state	V
CmdAddNoiseMarginHigh	The closeness of CmdAdd coming to vihDC in a logic high state	V

Electrical Measurement on Clock

Electrical Measurement on Clock requires signal Clock. It includes the following measurement:

Measurement type	Description	Units
ClockVIX	Clock differential input cross point voltage relative to vDD/2	mV

Skews between DQ and DQS

Skews between DQ and DQS calculate skews between DQ and DQS signals. Required signals are DQ, DQ_Reference, DQS, DQS_Reference. It also requires that Data Timing on DQ and Data Timing on DQS measurements are selected on Measurements tab. It includes the following measurements:

Measurement type	Description	Units
DQ_DQSSkewSetupFall	Falling edge setup skew between DQ and DQS (DQ falling edge setup flight time - DQS flight time), derated if DQ/DQS derating table is provided	ps
DQ_DQSSkewSetupRise	Rising edge setup skew between DQ and DQS (DQ rising edge setup flight time - DQS flight time), derated if DQ/DQS derating table is provided	ps
DQ_DQSSkewSetupFallAC150	AC150 falling edge setup skew between DQ and DQS (DQ AC150 falling edge setup flight time - DQS flight time), derated if DQ/DQS AC150 derating table is provided	ps
DQ_DQSSkewSetupRiseAC150	AC150 rising edge setup skew between DQ and DQS (DQ AC150 riseing edge setup flight time - DQS flight time), derated if DQ/DQS AC150 derating table is provided	ps
DQ_DQSSkewHoldFall	Falling edge hold skew between DQ and DQS (DQS flight time - DQ falling edge hold flight time), derated if DQ/DQS derating table is provided	ps
DQ_DQSSkewHoldRise	Rising edge hold skew between DQ and DQS (DQS flight time - DQ rising edge hold flight time), derated if DQ/DQS derating table is provided	ps
DQ_DQSSkewReadSetupFall	Read mode falling edge setup skew between DQ and DQS (DQ read mode falling edge maximum flight time - DQS flight time)	ps
DQ_DQSSkewReadSetupRise	Read mode rising edge setup skew between DQ and DQS (DQ read mode rising edge maximum flight time - DQS flight time)	ps
DQ_DQSSkewReadHoldFall	Read mode falling edge hold skew between DQ and DQS (DQS flight time - DQ read mode falling edge minimum flight time)	ps
DQ_DQSSkewReadHoldRise	Read mode rising edge hold skew between DQ and DQS (DQS flight time - DQ read mode rising edge minimum flight time)	ps

Skews between Cmd/Add and Clock

Skews between Cmd/Add and Clock calculate skews between Command/Address and Clock signals. Required signals are CmdAdd, CmdAdd_Reference, Clock, Clock_Reference. It also requires that Cmd/Add Timing on Cmd/Add and Cmd/Add Timing on Clock measurements are selected on Measurements tab. It includes the following measurements:

Measurement type	Description	Units
CmdAdd_ClockSkewSetupFall	Falling edge setup skew between Cmd/Add and Clock (CmdAdd falling edge setup flight time - Clock flight time), derated if CmdAdd/CLK derating table is provided	ps
CmdAdd_ClockSkewSetupRise	Rising edge setup skew between Cmd/Add and Clock (CmdAdd rising edge setup flight time - Clock flight time), derated if CmdAdd/CLK derating table is provided	ps
CmdAdd_ClockSkewHoldFall	Falling edge hold skew between Cmd/Add and Clock (Clock flight time - CmdAdd falling edge hold flight time), derated if CmdAdd/CLK derating table is provided	ps
CmdAdd_ClockSkewHoldRise	Rising edge hold skew between Cmd/Add and Clock (Clock flight time - CmdAdd rising edge hold flight time), derated if CmdAdd/CLK derating table is provided	ps
CmdAdd_ClockSkewSetupFallAC150	AC150 falling edge setup skew between Cmd/Add and Clock (CmdAdd AC150 falling edge setup flight time - Clock flight time), derated if CmdAdd/CLK AC150 derating table is provided	ps
CmdAdd_ClockSkewSetupRiseAC150	AC150 rising edge setup skew between Cmd/Add and Clock (CmdAdd AC150 rising edge setup flight time - Clock flight time), derated if CmdAdd/CLK AC150 derating table is provided	ps

Skews between DQS and Clock

Skews between DQS and Clock clculate skews between DQS and Clock signals. Required signals are DQS, DQS_Reference, Clock, Clock_Reference. It also requires that Data Timing on DQS and Cmd/Add Timing on Clock measurements are selected on Measurements tab. It includes the following measurements:

Measurement type	Description	Units
DQS_ClockSkewFallMax	DQS falling edge maximum skew to Clock (maximum DQS falling edge flight time - minimum Clock flight time)	ps
DQS_ClockSkewFallMin	DQS falling edge minimum skew to Clock (minimum DQS falling edge flight time - maximum Clock flight time)	ps
DQS_ClockSkewRiseMax	DQS Rising edge maximum skew to Clock (maximum DQS Rising edge flight time - minimum Clock flight time)	ps
DQS_ClockSkewRiseMin	DQS Rising edge minimum skew to Clock (minimum DQS Rising edge flight time - maximum Clock flight time)	ps

Advanced Features

Compliance Kit Directory Structure

The relevant directories of the measurement compliance kit are:

- DDR2
  - doc
  - lib
    - Platform (e.g. linux_x86,win32,etc)
  - source
    - bitmaps
    - constraints
    - data

- doc
- measurements
- symbols

Directory	Contains
doc	A copy of the documentation in pdf format describing this measurement kit.
lib	The measurement shared libraries for different platforms.
source/bitmap	The icons for the measurement component used in the schematic.
source/constraints	Several constraint files used for compliance/margin testing.
source/data	Derating tables.
source/doc	The original documentation in pdf format.
source/measurements	The C++ measurement source code.
source/symbols	The symbol for the measurement component used in the schematic.

Compliance Test Constraint File

The constraint files are located in the directory *DDR2/source/constraints*. There are several constraint files depending on the speed grade that is used in the simulation. These files are XML files which are named *constraints<speed_grade>.xml*.

The generic format for these files contains the name of the measurement to be tested and the corresponding ranges.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<TestSet>
   <TestLimit Name="Measurement1">
       <Min>0</Min>
       <Max>100</Max>
   </TestLimit>
   <TestLimit Name="Measurement2">
       <Min>-100</Min>
       <Max>200</Max>
   </TestLimit>
 </TestSet>
```

In this example there are two measurements named <*Measurement1*> and <*Measurement2*>. An acceptable value for <*Measurement1*> should be between 0 and 100.
To make the minimum value boundless a value of <*-inf*> can be used and similarly <*inf*> for the maximum value.

Derating Tables

Derating Table Formats

The derating table is specified using the CSV (comma-separated values) file format. There are two types of sub-formats that are used, namely a matrix and a set of vectors.

The matrix based files have the following generic format:

```
Type,Matrix
Title,AnyTitle
X,X-axis name
Y,Y-axis name
,1 , 2
3 ,3 , 6
4 ,4 , 8
```

The vector based files have the following generic format:

```
Type,Vector
Title, AnyTitle
x-name,y-name
1,2
2,4
3,6
```

DDR2 Specific Tables

File name	Format	Description
DDR2_derating_table_CmdAdd_Hold.csv	Matrix	Delat_tIH (ps) - derating values of tIH
DDR2_derating_table_DQ_Hold.csv	Matrix	Delat_tDH (ps) - tDH derating values
DDR2_tVAC_constraint_AC150.csv	Vector	tVAC min constraint vs Slew Rate for AC150
DDR2_derating_table_CmdAdd_Setup_AC150.csv	Matrix	Delat_tIS (ps) - derating values of tIS - for AC150
DDR2_derating_table_DQ_Setup_1333_1600.csv	Matrix	Delat_tDS (ps) - derating values of tDS for DDR2-1333/1600
DDR2_tVAC_constraint.csv	Vector	tVAC min constraint vs Slew Rate
DDR2_derating_table_CmdAdd_Setup.csv	Matrix	Delat_tIS (ps) - derating values of tIS
DDR2_derating_table_DQ_Setup_800_1066.csv	Matrix	Delat_tDS (ps) - derating values of tDS for DDR2-800/1066
DDR2_tVAC_constraint_DQ_1333_1600.csv	Vector	tVAC min constraint vs Slew Rate for DQ of DDR2-1333/1600

Measurement Source Code

The measurement source codes are located in the *DDR2/source/measurements* directory. The source code is organized based on the measurements groups. Each measurement group is a C++ class with the following entry points:

Class Method	**Description**
MeasurementGroup::initialize()	Called right before the transient analysis starts up.
MeasurementGroup::event(Trigger* trigger)	Called when a trigger event has occurred.
MeasurementGroup::evaluate(double time)	Called at each transient time point.
MeasurementGroup::finalize()	Called when the transient analysis is done.
MeasurementGroup::checkCompliance()	Called when the transient analysis is done and compliance testing is analyzing the measumrents.

The source code is compiled when the source code is used for the first time by the simulator or when the source code is modified. Once the compilation is done a file named *Compiler.log* is saved in the *DDR2/lib* directory.

DDR3 Compliance Measurement

The DDR3 Compliance Measurement application provides a fast and easy way to perform DDR3 related measurements on your designs. These measurements are done using a probe which is shipped with ADS as a design kit. This probe will allow you to perform compliance testing and margin analysis to determine if your design meets a set of given standards. All the implemented measurements are based on the JEDEC [1] standard.

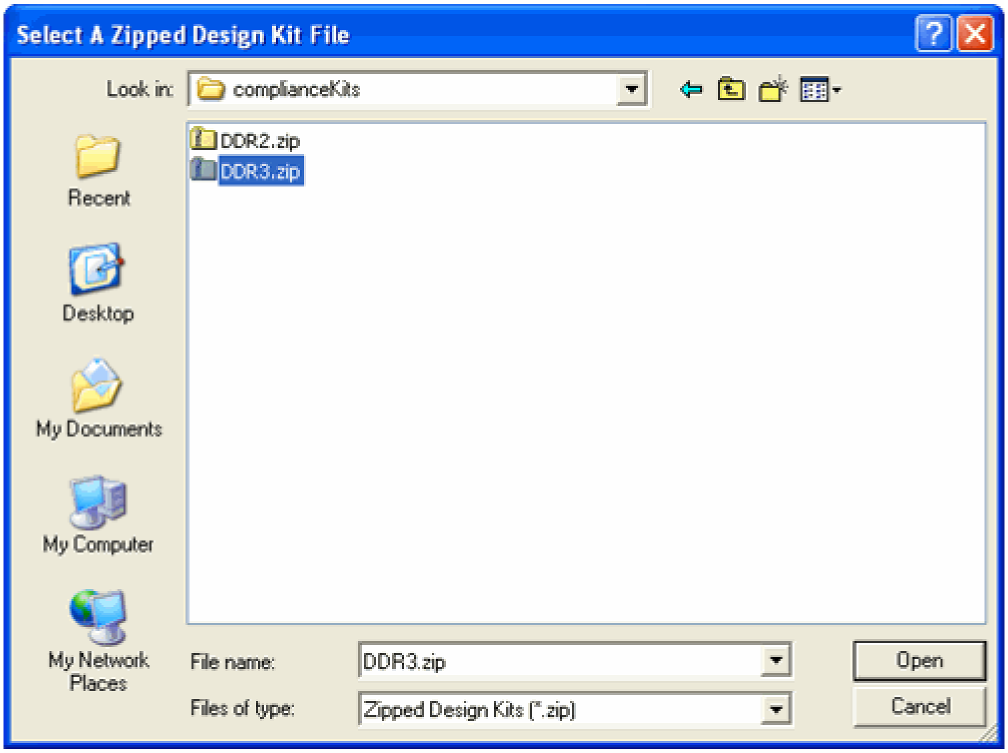
Besides providing a user-friendly interface this application is flexible enough to allow you to customize everything from the base measurement code to the compliance test standards. All the measurements are implemented in C++ which is efficient in terms of processing speed and memory usage. The measurement code is compiled on-demand into a shared library using a compiler that is shipped with ADS. The measurement compilation process is done behind the scene automatically when the source code is modified or when there is a version conflict.

> **Note**
> [1] The JEDEC (Joint Electronic Device Engineering Council) Solid State Technology Association is a semiconductor engineering standardization body of the Electronic Industries Alliance (EIA), a trade association that represents all areas of the electronic industry.

Installation

This section provides a step-by-step procedure to install the DDR3 measurement probe.

1. Open ADS workspace where *DDR3.zip* file needs to be extracted (for example, DDR3Compliance_wrk).
2. From the ADS Main window, select **DesignKits > Manage Favorite Design Kits**. The Manage Favorite Design Kits window appears
3. Click **Add Zipped Design Kit**. This will bring up another dialog box which will allow you to select the zip file. The *DDR3.zip* file is located in your ADS installation directory under the *complianceKits* directory. Select the *DDR3.zip* file and click **Open**.



Select A Zipped Design Kit File

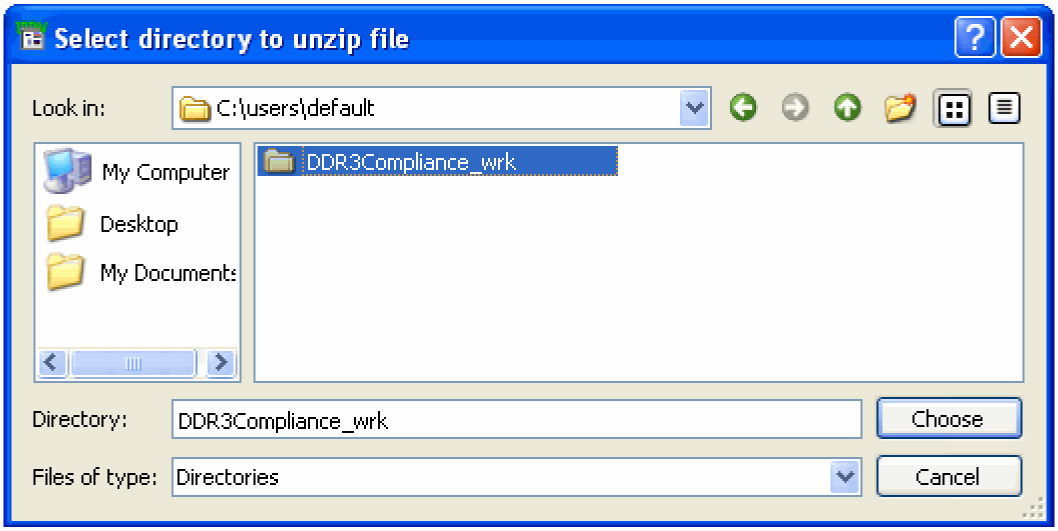
Look in: complianceKits

DDR2.zip
DDR3.zip

File name: DDR3.zip

Files of type: Zipped Design Kits (*.zip)

4. Select the workspace directory and click **Choose** to unzip the DDR3.zip file.



Select directory to unzip file

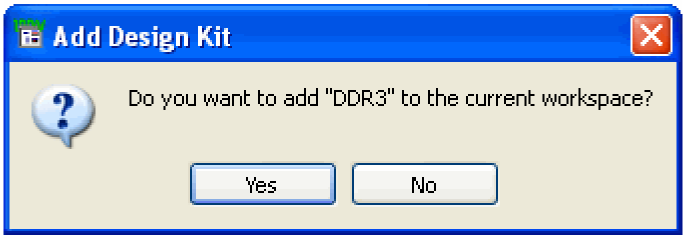
Look in: C:\users\default

DDR3Compliance_wrk

Directory: DDR3Compliance_wrk

Files of type: Directories

5. Once the *DDR3.zip* file is extracted, you are prompted to add DDR3 to the current workspace. Click **Yes** to add the DDR3 compliance measurement application, else click **No**.

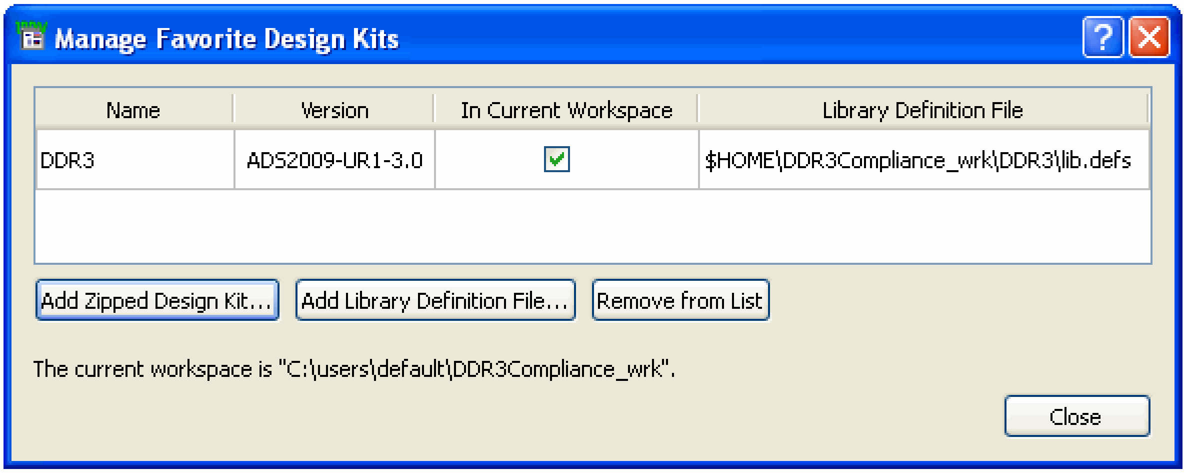


Add Design Kit

Do you want to add "DDR3" to the current workspace?

6. Click **Close** to close the Manage Favorite Design Kits window. The DDR3

measurement probe is now installed.



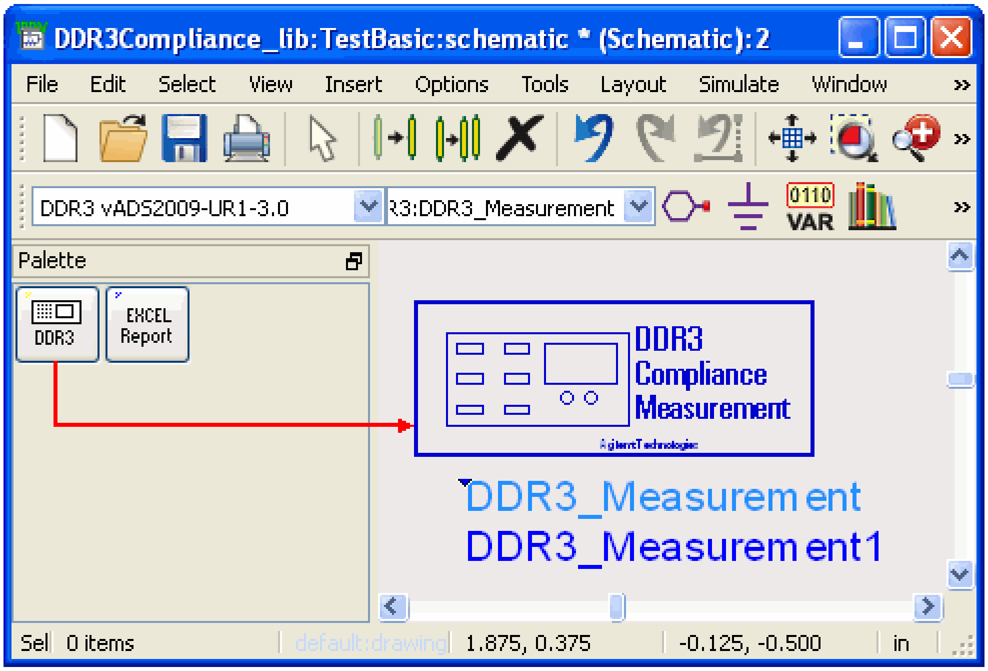
Name	Version	In Current Workspace	Library Definition File
DDR3	ADS2009-UR1-3.0	☑	$HOME\DDR3Compliance_wrk\DDR3\lib.defs

The current workspace is "C:\users\default\DDR3Compliance_wrk".

Setting up for Simulation

Follow the steps below to setup DDR3 probe:

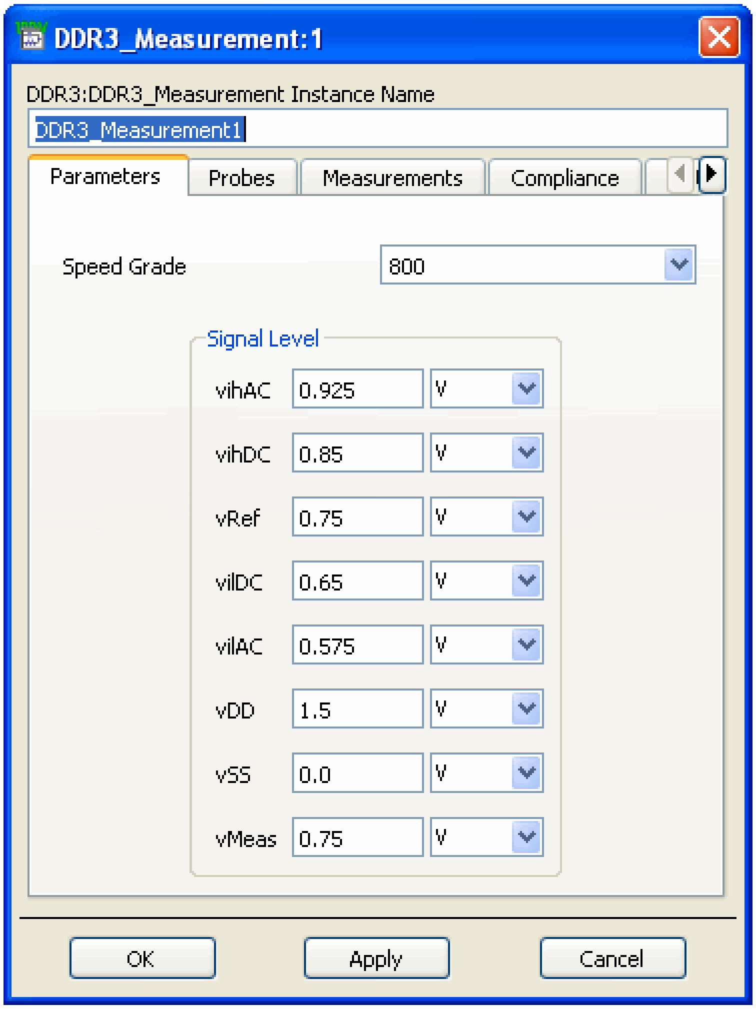
1. In the Schematic window, select the *DDR3 vADS2009-UR1-3.0* palette from the Component list.
2. Drag and drop the **DDR3_Measurement** probe component from the Palette into the Schematic area.



3. Double click on the **DDR3_Measurement** probe to setup the probe. This step involves adjusting the parameters, connecting the probe, selecting the measurements, activating the compliance analysis and setting output options.

Adjusting the Parameters

There are two sets of parameters that need to be set. The first is the *Speed Grade* of the DDR3 and the second is the *Signal Level*. A detailed description of these parameters is given in the Parameters section.



DDR3_Measurement:1

DDR3:DDR3_Measurement Instance Name: DDR3_Measurement1

Speed Grade: 800

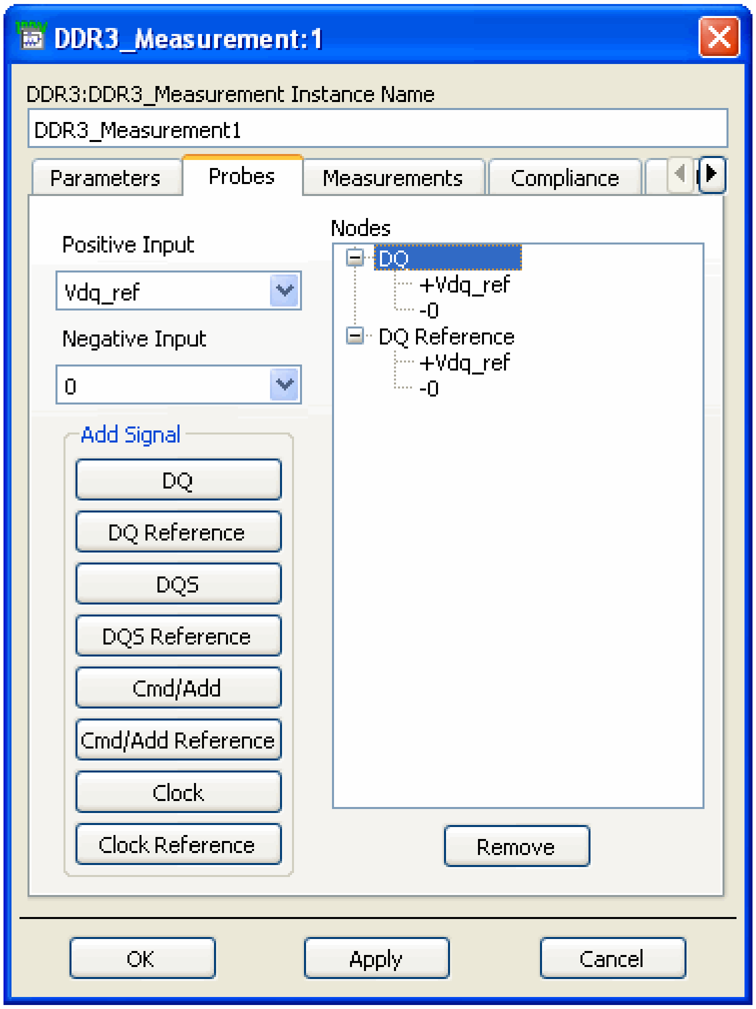
Signal Level

vihAC	0.925	V
vihDC	0.85	V
vRef	0.75	V
vilDC	0.65	V
vilAC	0.575	V
vDD	1.5	V
vSS	0.0	V
vMeas	0.75	V

Connecting the Probe

Once the parameters are set the probe needs to be connected to the circuit.

1. Firstly, select the *Positive Input* and *Negative Input* nodes using the pop-down menu.
2. Depending on the type of signal of the selected nodes, click on one of the buttons from the *Add Signal* on the bottom left hand side of the dialog box. This should add the nodes into the tree-like structure on right hand side of the dialog box.
3. You can select the signal type and remove the node set using the **Remove** button.



DDR3_Measurement:1

DDR3:DDR3_Measurement Instance Name

DDR3_Measurement1

Positive Input: Vdq_ref

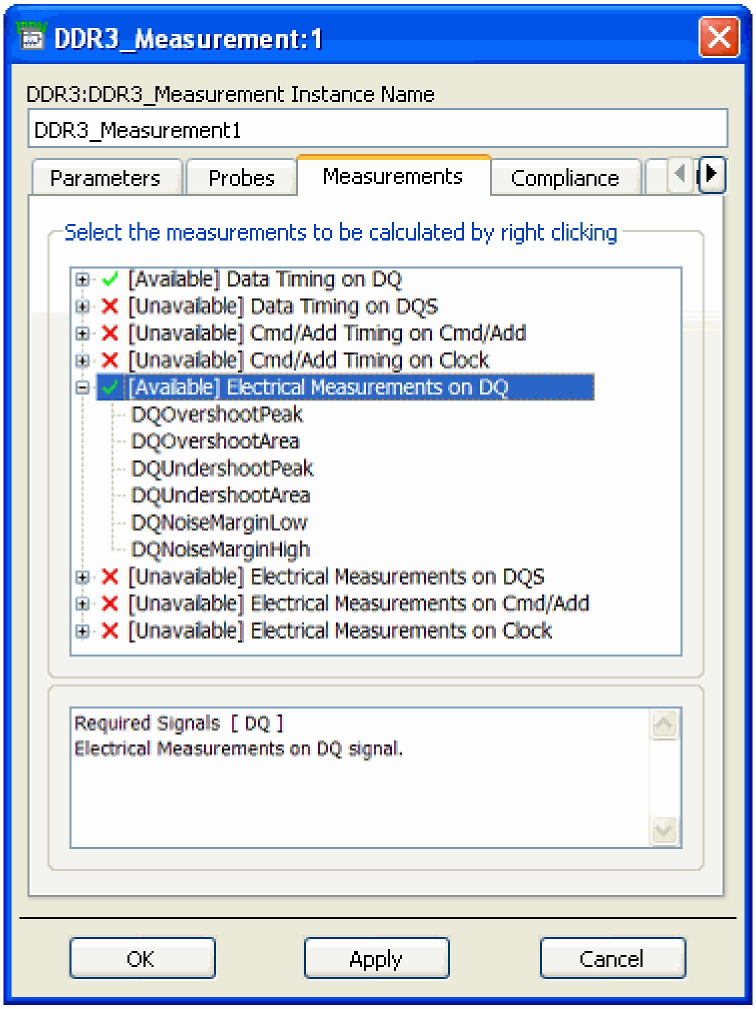
Negative Input: 0

Nodes

```
DQ
    +Vdq_ref
    -0
DQ Reference
    +Vdq_ref
    -0
```

Selecting the Measurements

Measurements can be selected by selecting the measurement group and right-clicking on the group. Note that only the measurements that are *Available* can be selected. Measurements become available depending on the type of signals that are connected to the probe. In the above example, two signal types **DQ, DQ Reference** have been selected, based on this selection only the *Data Timing on DQ* and *Electrical Measurements on DQ* are available. The required signals for a given measurement group is given in the brief description section, located towards the end of the dialog box. Each measurement group can be expanded and when you select it, the brief description section will contain an appropriate description.



DDR3:DDR3_Measurement Instance Name

DDR3_Measurement1

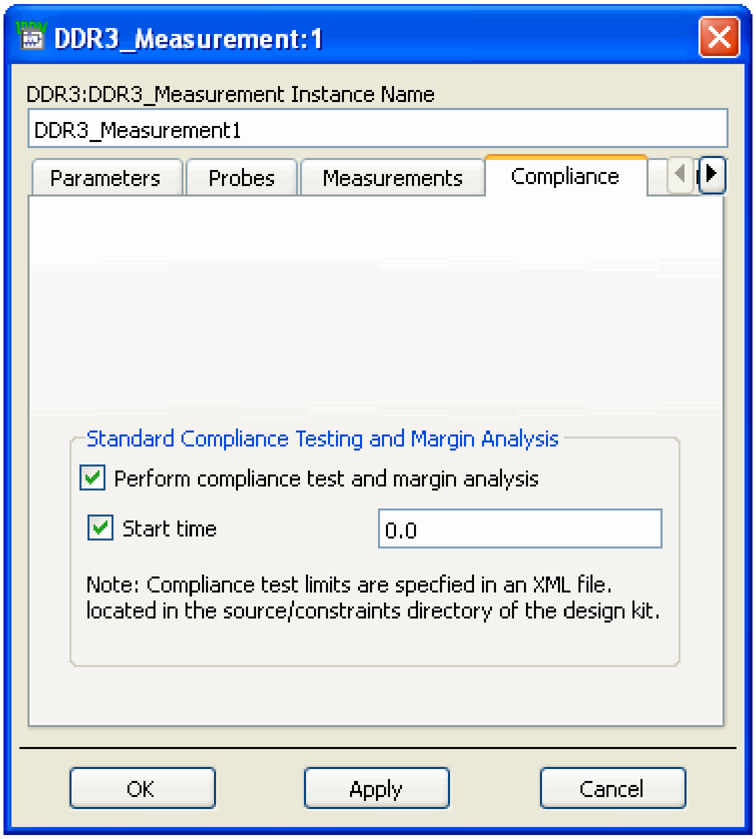
Select the measurements to be calculated by right clicking

- ✓ [Available] Data Timing on DQ
- ✗ [Unavailable] Data Timing on DQS
- ✗ [Unavailable] Cmd/Add Timing on Cmd/Add
- ✗ [Unavailable] Cmd/Add Timing on Clock
- ✓ [Available] Electrical Measurements on DQ
  - DQOvershootPeak
  - DQOvershootArea
  - DQUndershootPeak
  - DQUndershootArea
  - DQNoiseMarginLow
  - DQNoiseMarginHigh
- ✗ [Unavailable] Electrical Measurements on DQS
- ✗ [Unavailable] Electrical Measurements on Cmd/Add
- ✗ [Unavailable] Electrical Measurements on Clock

Required Signals [DQ]
Electrical Measurements on DQ signal.

Activating the Compliance Testing and Margin Analysis

Activating the compliance testing and margin analysis is done by selecting the *Perform compliance test and margin analysis* option. The *Start time* check box will allow you to specify the transient analysis time when measurements will be taken. This can be used to eliminate the transient response of the circuit and base the measurements on the steady state response.



DDR3:DDR3_Measurement Instance Name

DDR3_Measurement1

Standard Compliance Testing and Margin Analysis

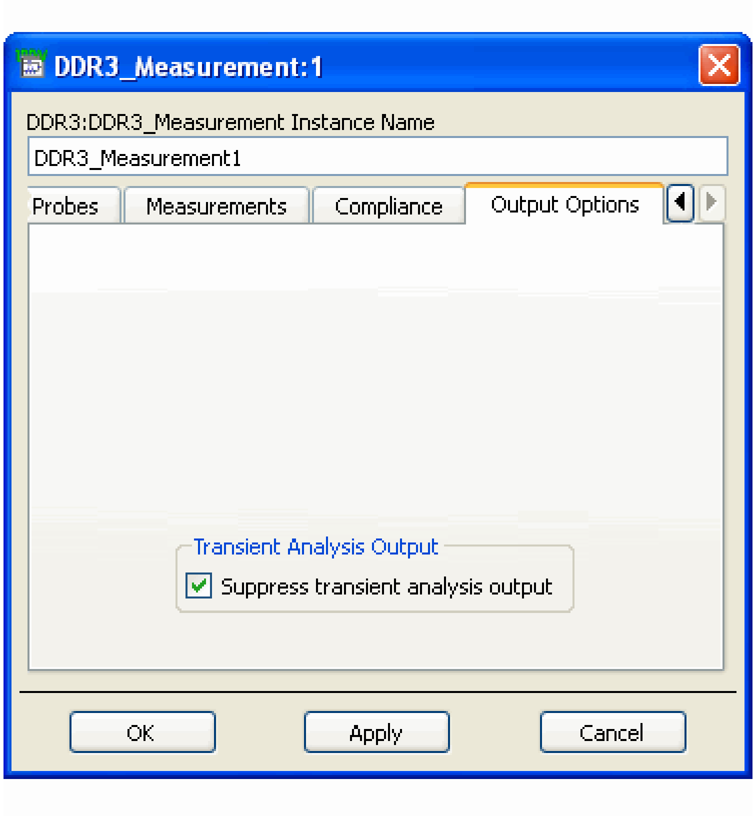
☑ Perform compliance test and margin analysis

☑ Start time 0.0

Note: Compliance test limits are specfied in an XML file.
located in the source/constraints directory of the design kit.

Setting Output Options

When you select the *Suppress transient analysis output* option, you can prevent any transient analysis waveforms from being saved.


DDR3_Measurement:1

DDR3:DDR3_Measurement Instance Name

DDR3_Measurement1

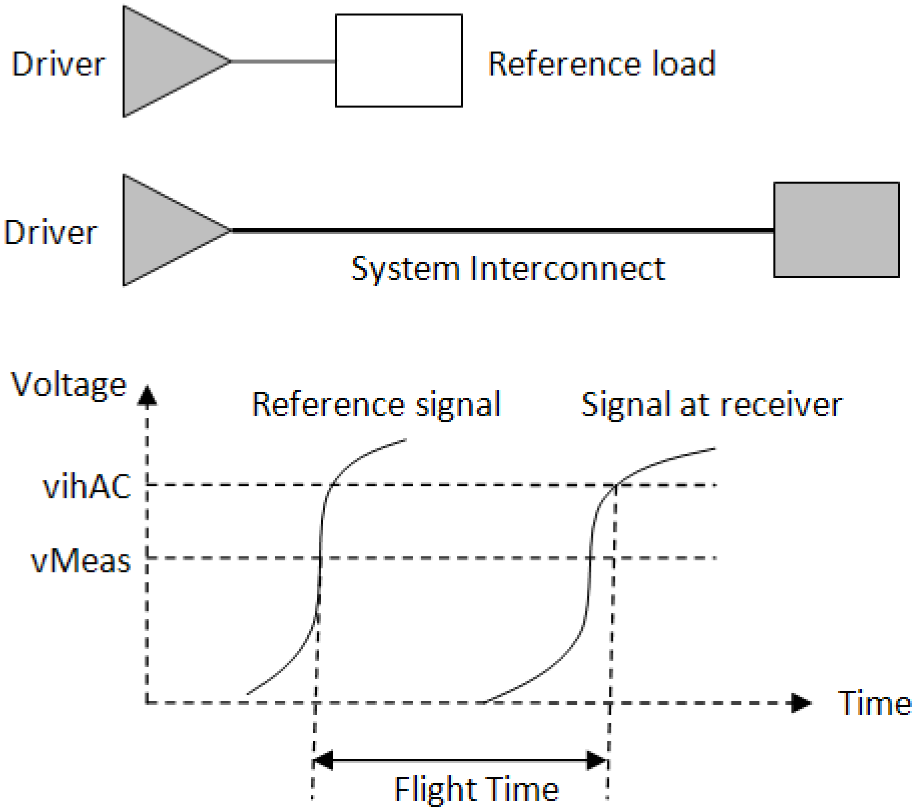
Transient Analysis Output

☑ Suppress transient analysis output

General Measurements

Flight Time

Flight time is defined as the time difference between the reference waveform and the waveform at the receiver. (Stephen H. Hall, Garrett W. Hall, and James A. McCall, *High-Speed Digital System Design*, Wiley, New York, 2000). To be specific, it is the amount of time that elapses between the point where the signal on an output buffer driving a reference load crosses the threshold voltage vMeas (defined in *Parameters* tab of DDR3 probe component) and the point where the signal crosses a threshold voltage (such as vihAC, vihDC, vRef, vilDC, vilAC in the *Parameters* tab) at the receiver in the system for the same falling or rising edge. The following figure illustrates the flight time from vMeas crossing of reference signal and vihAC crossing of signal at receiver for the same rising edge. Flight times are used later on to calculate the skews between signals.



Slew Rate

Slew Rate used in DDR3 measurements is defined in detail in *JEDEC Standard No. 79-3C* section 13.3 and 13.4 (pp. 173-186) and illustrated in Figures 110-117. In this DDR3 design kit, the accurate tangent line slew rates are measured which are required for derating value and tVAC constraint lookups.

Derating Value and Table

The setup/hold time base values are specified based on standard slew rates (1.0 for DQ/CmdADD and 2.0 for DQS/Clock). For different slew rates, these base values need to be compensated by adding the derating values looked up from derating tables. Various derating tables are provided as CSV files and located in *DDR3/source/data* directory. These tables are loaded and looked up by skew measurement code, such as *DDR3/source/measurements/DQ_DQS_Skew.cxx*.

Skew with Derating

Skew is calculated as the difference between flight time of two different signals. For example, DQ_DQS skew is the difference between DQ flight time and DQS flight time. Skew is used in the equation of the most interested timing margin calculation:

```
Timing Margin = controller spec - interconnect skew - (memory spec base + derating value)
  = controller spec - (interconnect skew + derating value) - memory spec base
```

From this equation, we can see that the timing margin can also be calculated by adding the derating value to interconnect skew, instead of memory spec base. The skews calculated in this DDR3 design kit are the ones with derating value taken into account.

tVAC and tVAC margin

For a valid transition, the input signal has to remain above/below AC threshold levels - vihAC/vilAC - for some time. This time span is tVAC which is defined in *JEDEC Standard No. 79-3C* section 13.3 and 13.4 (pp. 173-186) and illustrated in Figures 110-117. As shown in JEDEC79-3C tables 69 and 73 tVAC constraint is slew rate dependent, similar to derating value. For user's convenience tVAC margin measurement is added and calculated as:

```
tVAC margin = tVAC - tVAC constraint value looked up based on slew rate
```

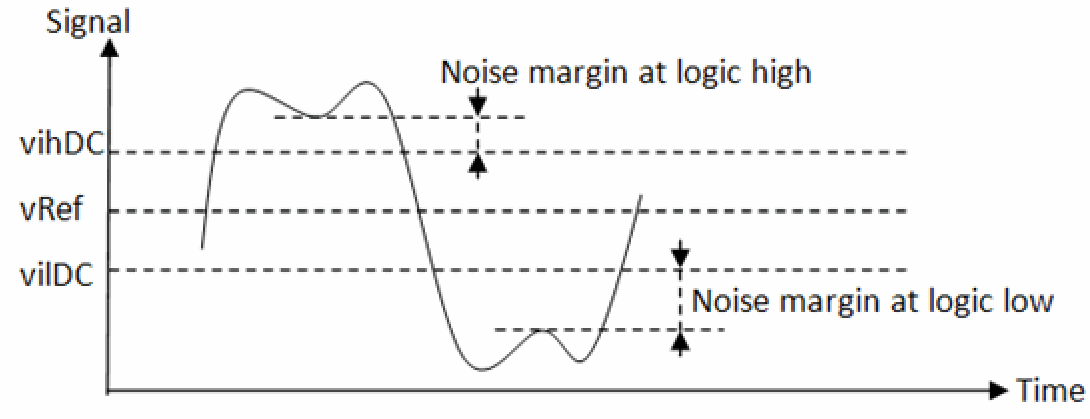
The slew rate dependent tVAC constraint tables are provided as CSV files and located in *DDR3/source/data* directory. They are loaded and used in the similar way as derating tables.

Overshoot and Undershoot

The overshoot/undershoot peak and area are defined in *JEDEC Standard No. 79-3C* section 9.6 and illustrated in Figures 98 and 99. The maximum amplitudes above/below VDD/VSS are defined as overshoot/undershoot peaks. The areas between waveform and VDD/VSS are defined as overshoot/undershoot areas.

Noise Margin

Noise margin measures how close of the signal comes to DC threshold voltage vihDC/vilDC in a logic high/low state. A signal is in a logic high state after the signal has crossed vihDC threshold in a rising edge and before it crosses vihDC in a falling edge. A signal is in a logic low state after the signal has crossed vilDC threshold in a falling edge and before it crosses vilDC in a rising edge. Basically noise margin measures how much noise can be tolerated without the signal falsely crossing vihDC/vilDC threshold in a logic high/low state. Noise margin measurement is further illustrated in the figure below.



VIX

VIX is the differential input cross point voltage and measured from the actual cross point of true and complement signals to the midlevel between VDD and VSS. It is officially defined in *JEDEC Standard No. 79-3C* section 8.4 and illustrated in Figure 93.

Measurements

Data Timing on DQ

Data Timing on DQ is a measure of DQ signal for Data Timing analysis. Required signals are DQ and DQ_Reference. It includes the following measurements:

Measurement type	Description	Units
DQFlightTimeSetupFall	Flight time between DQ vilAC crossing and DQ_Reference vMeas crossing in a falling edge	ps
DQFlightTimeSetupRise	Flight time between DQ vihAC crossing and DQ_Reference vMeas crossing in a rising edge	ps
DQFlightTimeHoldFall	Flight time between DQ vihDC crossing and DQ_Reference vMeas crossing in a falling edge	ps
DQFlightTimeHoldRise	Flight time between DQ vilDC crossing and DQ_Reference vMeas crossing in a rising edge	ps
DQSlewRateSetupFall	Tangent line slew rate between DQ vRef and vilAC crossings in a falling edge	V/ns
DQSlewRateSetupRise	Tangent line slew rate between DQ vRef and vihAC crossings in a rising edge	V/ns
DQSlewRateHoldFall	Tangent line slew rate between DQ vihDC and vRef crossings in a falling edge	V/ns
DQSlewRateHoldRise	Tangent line slew rate between DQ vilDC and vRef crossings in a rising edge	V/ns
DQFlightTimeReadFallMin	Flight time between the first DQ vRef crossing and DQ_Reference vMeas crossing in a falling edge	ps
DQFlightTimeReadFallMax	Flight time between the last DQ vRef crossing and DQ_Reference vMeas crossing in a falling edge	ps
DQFlightTimeReadRiseMin	Flight time between the first DQ vRef crossing and DQ_Reference vMeas crossing in a rising edge	ps
DQFlightTimeReadRiseMax	Fight time between the last DQ vRef crossing and DQ_Reference vMeas crossing in a rising edge	ps
DQFlightTimeSetupFallAC150	Flight time between DQ vilAC150 (vRef-150mV) crossing and DQ_Reference vMeas crossing in a falling edge	ps
DQFlightTimeSetupRiseAC150	Flight time between DQ vihAC150 (vRef+150mV) crossing and DQ_Reference vMeas crossing in a rising edge	ps
DQSlewRateSetupFallAC150	Tangent line slew rate between DQ vRef and vilAC150 (vRef-150mV) crossings in a falling edge	V/ns
DQSlewRateSetupRiseAC150	Tangent line slew rate between DQ vRef and vihAC150 (vRef+150mV) crossings in a rising edge	V/ns
DQtVACHigh	Time of DQ above vihAC for valid transition	ps
DQtVACLow	Time of DQ below vilAC for valid transition	ps
DQtVACMarginHigh	DQ tVAC above vihAC (DQtVACHigh) minus tVAC constraint based on DQ rising edge setup slew rate (DQSlewRateSetupRise)	ps
DQtVACMarginLow	DQ tVAC below vilAC (DQtVACLow) minus tVAC constraint based on DQ falling edge setup slew rate (DQSlewRateSetupFall)	ps
DQtVACHighAC150	Time of DQ above vihAC150 (vRef+150mV) for valid transition	ps
DQtVACLowAC150	Time of DQ below vilAC150 (vRef-150mV) for valid transition	ps
DQtVACMarginHighAC150	DQ tVAC above vihAC150 (DQtVACHighAC150) minus tVAC constraint based on DQ AC150 rising edge setup slew rate (DQSlewRateSetupRiseAC150)	ps
DQtVACMarginLowAC150	DQ tVAC below vilAC150 (DQtVACLowAC150) minus tVAC constraint based on DQ AC150 falling edge setup slew rate (DQSlewRateSetupFallAC150)	ps

Data Timing on DQS

Data Timing on DQS is a measure of DQS signal for Data Timing analysis. Required signals are DQS and DQS_Reference. It includes the following measurements:

Measurement type	**Description**	**Units**
DQSFlightTimeFall	Flight time between DQS zero crossing and DQS_Reference zero crossing in a falling edge	ps
DQSFlightTimeRise	Flight time between DQS zero crossing and DQS_Reference zero crossing in a rising edge	ps
DQSSlewRateFall	Slew rate between DQS 0.2V and -0.2V crossings in a falling edge	V/ns
DQSSlewRateRise	Slew rate between DQS -0.2V and 0.2V crossings in a rising edge	V/ns

Cmd/Add Timing on Cmd/Add

Cmd/Add Timing on Cmd/Add is a measure of Command and Address signal for Command and Address Timing analysis. Required signals are CmdAdd and CmdAdd_Reference. It includes the following measurements:

Measurement type	Description	Units
CmdAddFlightTimeSetupFall	Flight time between CmdAdd vilAC crossing and CmdAdd_Reference vMeas crossing in a falling edge	ps
CmdAddFlightTimeSetupRise	Flight time between CmdAdd vihAC crossing and CmdAdd_Reference vMeas crossing in a rising edge	ps
CmdAddFlightTimeHoldFall	Flight time between CmdAdd vihDC crossing and CmdAdd_Reference vMeas crossing in a falling edge	ps
CmdAddFlightTimeHoldRise	Flight time between CmdAdd vilDC crossing and CmdAdd_Reference vMeas crossing in a rising edge	ps
CmdAddSlewRateSetupFall	Tangent line slew rate between CmdAdd vRef and vilAC crossings in a falling edge	V/ns
CmdAddSlewRateSetupRise	Tangent line slew rate between CmdAdd vRef and vihAC crossings in a rising edge	V/ns
CmdAddSlewRateHoldFall	Tangent line slew rate between CmdAdd vihDC and vRef crossings in a falling edge	V/ns
CmdAddSlewRateHoldRise	Tangent line slew rate between CmdAdd vilDC and vRef crossings in a rising edge	V/ns
CmdAddFlightTimeSetupFallAC150	Flight time between CmdAdd vilAC150 (vRef-150mV) crossing and CmdAdd_Reference vMeas crossing in a falling edge	ps
CmdAddFlightTimeSetupRiseAC150	Flight time between CmdAdd vihAC150 (vRef+150mV) crossing and CmdAdd_Reference vMeas crossing in a rising edge	ps
CmdAddSlewRateSetupFallAC150	Tangent line slew rate between CmdAdd vRef and vilAC150 (vRef-150mV) crossings in a falling edge	V/ns
CmdAddSlewRateSetupRiseAC150	Tangent line slew rate between CmdAdd vRef and vihAC150 (vRef+150mV) crossings in a rising edge	V/ns
CmdAddtVACHigh	Time of CmdAdd above vihAC for valid transition	ps
CmdAddtVACLow	Time of CmdAdd below vilAC for valid transition	ps
CmdAddtVACMarginHigh	CmdAdd tVAC above vihAC (CmdAddtVACHigh) minus tVAC constraint based on CmdAdd rising edge setup slew rate (CmdAddSlewRateSetupRise)	ps
CmdAddtVACMarginLow	CmdAdd tVAC below vilAC (CmdAddtVACLow) minus tVAC constraint based on CmdAdd falling edge setup slew rate (CmdAddSlewRateSetupFall)	ps
CmdAddtVACHighAC150	Time of CmdAdd above vihAC150 (vRef+150mV) for valid transition	ps
CmdAddtVACLowAC150	Time of CmdAdd below vilAC150 (vRef-150mV) for valid transition	ps
CmdAddtVACMarginHighAC150	CmdAdd tVAC above vihAC150 (CmdAddtVACHighAC150) minus tVAC constraint based on CmdAdd AC150 rising edge setup slew rate (CmdAddSlewRateSetupRiseAC150)	ps
CmdAddtVACMarginLowAC150	CmdAdd tVAC below vilAC150 (CmdAddtVACLowAC150) minus tVAC constraint based on CmdAdd AC150 falling edge setup slew rate (CmdAddSlewRateSetupFallAC150)	ps

Cmd/Add Timing on Clock

Cmd/Add Timing on Cmd/Add is a measure of Clock signal for Command and Address Timing analysis. Required signals are Clock and Clock_Reference. It includes the following measurements.

Measurement type	Description	Units
ClockFlightTimeFall	Flight time between Clock zero crossing and Clock_Reference zero crossing in a falling edge	ps
ClockFlightTimeRise	Flight time between Clock zero crossing and Clock_Reference zero crossing in a rising edge	ps
ClockSlewRateFall	Slew rate between Clock 0.2V and -0.2V crossings in a falling edge	V/ns
ClockSlewRateRise	Slew rate between Clock -0.2V and 0.2V crossings in a rising edge	V/ns

Electrical Measurements on DQ

Electrical Measurements on DQ requires signal DQ. It includes the following measurements:

Measurement type	Description	Units
DQOvershootPeak	DQ peak amplitude above vDD	V
DQOvershootArea	Area of DQ signal above vDD	V-ns
DQUndershootPeak	DQ peak amplitude below vSS	V
DQUndershootArea	Area of DQ signal below vSS	V-ns
DQNoiseMarginLow	The closeness of DQ coming to vilDC in a logic low state	V
DQNoiseMarginHigh	The closeness of DQ coming to vihDC in a logic high state	V

Electrical Measurement on DQS

Electrical Measurement on DQS requires signal DQ. It includes the following measurement:

Measurement type	Description	Units
DQSVIX	DQS differential input cross point voltage relative to vDD/2	mV

Electrical Measurements on CmdAdd

Electrical Measurements on Command or Address signal requires signal CmdAdd. It includes the following measurements:

Measurement type	Description	Units
CmdAddOvershootPeak	CmdAdd peak amplitude above vDD	V
CmdAddOvershootArea	Area of CmdAdd signal above vDD	V-ns
CmdAddUndershootPeak	CmdAdd peak amplitude below vSS	V
CmdAddUndershootArea	Area of CmdAdd signal below vSS	V-ns
CmdAddNoiseMarginLow	The closeness of CmdAdd coming to vilDC in a logic low state	V
CmdAddNoiseMarginHigh	The closeness of CmdAdd coming to vihDC in a logic high state	V

Electrical Measurement on Clock

Electrical Measurement on Clock requires signal Clock. It includes the following measurement:

Measurement type	Description	Units
ClockVIX	Clock differential input cross point voltage relative to vDD/2	mV

Skews between DQ and DQS

Skews between DQ and DQS calculate skews between DQ and DQS signals. Required signals are DQ, DQ_Reference, DQS, DQS_Reference. It also requires that Data Timing on DQ and Data Timing on DQS measurements are selected on Measurements tab. It includes the following measurements:

Measurement type	Description	Units
DQ_DQSSkewSetupFall	Falling edge setup skew between DQ and DQS (DQ falling edge setup flight time - DQS flight time), derated if DQ/DQS derating table is provided	ps
DQ_DQSSkewSetupRise	Rising edge setup skew between DQ and DQS (DQ rising edge setup flight time - DQS flight time), derated if DQ/DQS derating table is provided	ps
DQ_DQSSkewSetupFallAC150	AC150 falling edge setup skew between DQ and DQS (DQ AC150 falling edge setup flight time - DQS flight time), derated if DQ/DQS AC150 derating table is provided	ps
DQ_DQSSkewSetupRiseAC150	AC150 rising edge setup skew between DQ and DQS (DQ AC150 riseing edge setup flight time - DQS flight time), derated if DQ/DQS AC150 derating table is provided	ps
DQ_DQSSkewHoldFall	Falling edge hold skew between DQ and DQS (DQS flight time - DQ falling edge hold flight time), derated if DQ/DQS derating table is provided	ps
DQ_DQSSkewHoldRise	Rising edge hold skew between DQ and DQS (DQS flight time - DQ rising edge hold flight time), derated if DQ/DQS derating table is provided	ps
DQ_DQSSkewReadSetupFall	Read mode falling edge setup skew between DQ and DQS (DQ read mode falling edge maximum flight time - DQS flight time)	ps
DQ_DQSSkewReadSetupRise	Read mode rising edge setup skew between DQ and DQS (DQ read mode rising edge maximum flight time - DQS flight time)	ps
DQ_DQSSkewReadHoldFall	Read mode falling edge hold skew between DQ and DQS (DQS flight time - DQ read mode falling edge minimum flight time)	ps
DQ_DQSSkewReadHoldRise	Read mode rising edge hold skew between DQ and DQS (DQS flight time - DQ read mode rising edge minimum flight time)	ps

Skews between Cmd/Add and Clock

Skews between Cmd/Add and Clock calculate skews between Command/Address and Clock signals. Required signals are CmdAdd, CmdAdd_Reference, Clock, Clock_Reference. It also requires that Cmd/Add Timing on Cmd/Add and Cmd/Add Timing on Clock measurements are selected on Measurements tab. It includes the following measurements:

Measurement type	Description	Units
CmdAdd_ClockSkewSetupFall	Falling edge setup skew between Cmd/Add and Clock (CmdAdd falling edge setup flight time - Clock flight time), derated if CmdAdd/CLK derating table is provided	ps
CmdAdd_ClockSkewSetupRise	Rising edge setup skew between Cmd/Add and Clock (CmdAdd rising edge setup flight time - Clock flight time), derated if CmdAdd/CLK derating table is provided	ps
CmdAdd_ClockSkewHoldFall	Falling edge hold skew between Cmd/Add and Clock (Clock flight time - CmdAdd falling edge hold flight time), derated if CmdAdd/CLK derating table is provided	ps
CmdAdd_ClockSkewHoldRise	Rising edge hold skew between Cmd/Add and Clock (Clock flight time - CmdAdd rising edge hold flight time), derated if CmdAdd/CLK derating table is provided	ps
CmdAdd_ClockSkewSetupFallAC150	AC150 falling edge setup skew between Cmd/Add and Clock (CmdAdd AC150 falling edge setup flight time - Clock flight time), derated if CmdAdd/CLK AC150 derating table is provided	ps
CmdAdd_ClockSkewSetupRiseAC150	AC150 rising edge setup skew between Cmd/Add and Clock (CmdAdd AC150 rising edge setup flight time - Clock flight time), derated if CmdAdd/CLK AC150 derating table is provided	ps

Skews between DQS and Clock

Skews between DQS and Clock clculate skews between DQS and Clock signals. Required signals are DQS, DQS_Reference, Clock, Clock_Reference. It also requires that Data Timing on DQS and Cmd/Add Timing on Clock measurements are selected on Measurements tab. It includes the following measurements:

Measurement type	Description	Units
DQS_ClockSkewFallMax	DQS falling edge maximum skew to Clock (maximum DQS falling edge flight time - minimum Clock flight time)	ps
DQS_ClockSkewFallMin	DQS falling edge minimum skew to Clock (minimum DQS falling edge flight time - maximum Clock flight time)	ps
DQS_ClockSkewRiseMax	DQS Rising edge maximum skew to Clock (maximum DQS Rising edge flight time - minimum Clock flight time)	ps
DQS_ClockSkewRiseMin	DQS Rising edge minimum skew to Clock (minimum DQS Rising edge flight time - maximum Clock flight time)	ps

Advanced Features

Compliance Kit Directory Structure

The relevant directories of the measurement compliance kit are:

- DDR3
  - doc
  - lib
    - Platform (e.g. linux_x86,win32,etc)
  - source
    - bitmaps
    - constraints
    - data

- doc
- measurements
- symbols

Directory	Contains
doc	A copy of the documentation in pdf format describing this measurement kit.
lib	The measurement shared libraries for different platforms.
source/bitmap	The icons for the measurement component used in the schematic.
source/constraints	Several constraint files used for compliance/margin testing.
source/data	Derating tables.
source/doc	The original documentation in pdf format.
source/measurements	The C++ measurement source code.
source/symbols	The symbol for the measurement component used in the schematic.

Compliance Test Constraint File

The constraint files are located in the directory *DDR3/source/constraints*. There are several constraint files depending on the speed grade that is used in the simulation. These files are XML files which are named *constraints<speed_grade>.xml*.

The generic format for these files contains the name of the measurement to be tested and the corresponding ranges.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<TestSet>
   <TestLimit Name="Measurement1">
       <Min>0</Min>
       <Max>100</Max>
   </TestLimit>
   <TestLimit Name="Measurement2">
       <Min>-100</Min>
       <Max>200</Max>
   </TestLimit>
 </TestSet>
```

In this example there are two measurements named *<Measurement1>* and *<Measurement2>*. An acceptable value for *<Measurement1>* should be between 0 and 100.
To make the minimum value boundless a value of *<-inf>* can be used and similarly *<inf>* for the maximum value.

Derating Tables

Derating Table Formats

The derating table is specified using the CSV (comma-separated values) file format. There are two types of sub-formats that are used, namely a matrix and a set of vectors.

The matrix based files have the following generic format:

```
Type,Matrix
Title,AnyTitle
X,X-axis name
Y,Y-axis name
,1 , 2
3 ,3 , 6
4 ,4 , 8
```

The vector based files have the following generic format:

```
Type,Vector
Title, AnyTitle
x-name,y-name
1,2
2,4
3,6
```

DDR3 Specific Tables

File name	Format	Description
DDR3_derating_table_CmdAdd_Hold.csv	Matrix	Delta_tIH (ps) - derating values of tIH
DDR3_derating_table_DQ_Hold.csv	Matrix	Delta_tDH (ps) - tDH derating values
DDR3_tVAC_constraint_AC150.csv	Vector	tVAC min constraint vs Slew Rate for AC150
DDR3_derating_table_CmdAdd_Setup_AC150.csv	Matrix	Delta_tIS (ps) - derating values of tIS - for AC150
DDR3_derating_table_DQ_Setup_1333_1600.csv	Matrix	Delta_tDS (ps) - derating values of tDS for DDR3-1333/1600
DDR3_tVAC_constraint.csv	Vector	tVAC min constraint vs Slew Rate
DDR3_derating_table_CmdAdd_Setup.csv	Matrix	Delta_tIS (ps) - derating values of tIS
DDR3_derating_table_DQ_Setup_800_1066.csv	Matrix	Delta_tDS (ps) - derating values of tDS for DDR3-800/1066
DDR3_tVAC_constraint_DQ_1333_1600.csv	Vector	tVAC min constraint vs Slew Rate for DQ of DDR3-1333/1600

Measurement Source Code

The measurement source codes are located in the *DDR3/source/measurements* directory. The source code is organized based on the measurements groups. Each measurement group is a C++ class with the following entry points:

Class Method	**Description**
MeasurementGroup::initialize()	Called right before the transient analysis starts up.
MeasurementGroup::event(Trigger* trigger)	Called when a trigger event has occurred.
MeasurementGroup::evaluate(double time)	Called at each transient time point.
MeasurementGroup::finalize()	Called when the transient analysis is done.
MeasurementGroup::checkCompliance()	Called when the transient analysis is done and compliance testing is analyzing the measumrents.

The source code is compiled when the source code is used for the first time by the simulator or when the source code is modified. Once the compilation is done a file named *Compiler.log* is saved in the *DDR3/lib* directory.

Differential Impedance Simulation

This simulation finds the differential impedance of the coupled lines using components from the multilayer library, which uses an accurate 2DEM solver. Note that the odd mode impedance is half of the differential impedance. LineCalc shows 43.66 Ohms as Zodd, which yields a differential impedance of 87.32 Ohms. This schematic simulation gives similar results to LineCalc.

Eye Closure Measurements

Note
Much of the functionality in this exercise has been superceded by the *Eye Probe* (ccsim) and *EyeDiff Probe (Differential Eye Measurement Probe)* (ccsim) components, which we recommend for new designs.

This example shows the use of AEL-based eye calculation functions.

The AEL code for the functions are listed on the data display pages.

These functions are:

- eye_amplitude - essentially takes a vertical histogram of the eye voltages, and subtracts the '0' level mean from the '1' level mean within a given measurement window.
- eye_height - also using histograms, computes the inner bounds of the eye opening.
- eye_closure - a ratio of the eye height to the eye amplitude.
- eye_rise_time - provide 20% to 80% rise time.
- eye_fall_time - provide 80% to 20% fall time.

Eye Diagram Front Panel

ℹ **Note**
Much of the functionality of Eye Diagram Front Panel has been superceded by the *Eye Probe* (ccsim) and *EyeDiff Probe (Differential Eye Measurement Probe)* (ccsim) components, which we recommend for new designs.

This menu item is simply a convenient way to open the *Eye Diagram FrontPanel* (data) user interface. (This tool can also be access from the Data Display window menu bar by selecting **Tools > FrontPanel > Eye**).

Eye Diagram Jitter Histogram Measurements

Note
Much of the functionality in this exercise has been superceded by the *Eye Probe* (ccsim) and *EyeDiff Probe (Differential Eye Measurement Probe)* (ccsim) components, which we recommend for new designs.

The `cross_hist()` function will compute and plot jitter histogram within the boundary (time1, time2, m1, m2). The AEL code for this function is listed on page AEL_Code of the data display for this exercise.

Linear Differential TDT

Inserts a subcircuit consisting of a differential three-line linear test component into your schematic. For an example of its use, refer to *Linear Differential TDT Simulation* (sigint).

Linear Differential TDT Simulation

Demonstration of the *Linear Differential TDT* (sigint) subcircuit.

Mixed Mode Simulations

3-port Simulations

The schematic provides 3-port circuits that have a single-ended input and a differential-mode output. The schematic contains a template for simulating full mixed-mode S-parameters using a single-ended 3-port simulation. The simulated results are displayed in the 3port_template.dds, which includes the appropriate equations to look at the single-ended S-parameters at the input and the mixed-mode (differential-mode, common-mode, and mode conversion) S-parameters at the output.

4-port Simulations

The simulation provides a 4-port schematic that have a differential-mode input and a differential-mode output. The schematic contains a template for simulating full mixed-mode S-parameters using a single-ended 4-port simulation. The simulated results are displayed in the 4port_template.dds, which includes the appropriate equations to look at the mixed-mode (differential-mode, common-mode, and mode conversion) S-parameters.

Mixed Mode Simulation Using Measured Data

3-port Simulation Using Measured Data

This case is for 3-port circuits that have a single-ended input and a differential-mode output. The schematic contains a template for importing multiple 2-port S-parameter measurements to create a 3-port simulation. The simulated results are displayed in the `Create_3port.dds`, which includes the appropriate equations to look at the single-ended S-parameters at the input and the mixed-mode (differential-mode, common-mode, and mode conversion) S-parameters at the output.

4-port Simulation Using Measured Data

This case is for 4-port circuits that have a differential mode input and a differential mode output. The schematic contains a template for importing multiple 2 port S-parameter measurements to create a 4 port simulation. The simulated results are displayed in the `Create_4port.dds`, which includes the appropriate equations to look at the mixed mode (differential-mode, common-mode, and mode conversion) S-parameters.

Mixed Mode S-Parameter Basics

Data display with derivation of mixed mode differential S-parameters from the single-ended S-parameters.

Non Linear Differential TDT

Inserts a subcircuit consisting of a differential three-line non-linear test component into your schematic. For an example of its use, please see *Non Linear Differential TDT Simulation* (sigint).

Non Linear Differential TDT Simulation

Demonstration of the *Non Linear Differential TDT* (sigint) subcircuit.

Pre-Emphasis and Equalization Co-simulation

Note
In many cases *Channel Simulation* (cktsimchan) will be a more convenient way of modeling pre-emphasis and equalization. It is recommended for all new designs.

An example of co-simulation of Ptolemy and Transient Convolution Simulator on a signal integrity channel transmitter pre-emphasis and receiver equalization.

This behavioural circuit demonstrates a pre-emphasis technique used by designers to help overcome frequency dependant loss in a channel. It is designed to take bipolar NRZ (PAM) input data. The circuit functions by boosting the first clock period after any level change from unity gain by a number applied at P3. The first clock period after a level change is detected by comparing the signal with a one clock period delayed version of itself with a NOT EQUAL test. When the signals are changing on a clock period to period basis the two inputs to the TestNE are always different, giving an output of 1 which switches the Mux to look at P3. When the signals are the same, as they would be for any state held longer that one clock period, the output of the TestNE becomes 0 and the Mux changes to the internal Const, and an overall gain between P1 and P2 is then 1.

Single Ended TDR and TDT

Inserts a subcircuit consisting of a single ended TDR and TDT test component into your schematic. For examples of its use, refer to *Single Ended TDR and TDT Simulation* (sigint) and *Single Ended TDR and TDT Impulse Simulation* (sigint).

Single Ended TDR and TDT Impulse Simulation

This is a demonstration design schematic that illustrates usage of the *Single Ended TDR and TDT* (sigint) component.

This compares measured and simulated results for a microstrip with step discontinuities. Both TDR (time domain reflectometry) and S-parameter simulations are shown. TDR_transient is used to simulate the TDR response. The TDR measurements were made using the Agilent Technologies 54120 scope and normalizing the step source to a 45 picosecond rise time. Results are displayed in `TDR_measVSmod.dds`.

Single Ended TDR and TDT Simulation

This compares measured and simulated results for a microstrip with step discontinuities. Both TDR (time domain reflectometry) and S-parameter simulations are shown. TDR_transient_NEW makes use of the *Single Ended TDR and TDT* (sigint) component. The results are shown in `TDR_transient_NEW.dds` where there is also a new function shown as `tdr_step_imped`. This function simply calculates impedance from reflection coefficient.

S-Parameter to Time Transform

This compares measured and simulated results for a microstrip with step discontinuities. Both TDR (time domain reflectometry) and S-parameter simulations are shown. SP_freqsweep is used to calculate the frequency response. The schematic shows the S-parameter test setup. The layout window shows the test line. The test line is a microstrip fabricated on 59mil FR-4 material, and has step discontinuities. Results are displayed in `SP_measVSmod.dds`. `SP_converted_to_TDR.dds` uses port-simulation processing to convert the linear frequency domain results into the time domain. Plots derived from S11 show the impulse reponse, TDR response and impedance response versus time. `SP_measVSmod_NEW.dds` shows two AEL functions that simplify the calculation of time domain reflection coefficient and impedance from S-parameter data.