cadence

SystemSI – Parallel Bus Analysis Tutorial

Product Version 16.6
July 2014

Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA.

Table of Contents

1 Parallel Bus Analysis Tutorial 6
1.1 Overview 6
1.2 Tutorial Description 6
2 Quick Start 7
2.1 Overview 7
2.2 Open a Template 7
2.3 Editing Components 9
2.3.1 PCB Block 9
2.3.2 Controller / Memory Block 11
2.3.2.1 MCP Generation for Controller and Memory IBIS Blocks 13
2.3.3 EBD Block 16
2.3.3.1 EBD File 16
2.3.3.2 Using EBD Models in System SI 17
2.3.4 S Parameter Block 18
2.3.4.1 Add S Parameter Block 18
2.3.4.2 Load S Parameter File 18
2.3.4.3 Extract the BBS Model 20
2.3.4.4 Load the BBS Model 23
2.3.4.5 Switch the Models 23
2.3.5 VRM Block 24
2.4 Adding New Components 25
2.5 Setting up Probe Points 26
2.6 Setting up the Simulation 29
2.7 Running the Simulation 30
2.8 Custom Templates 31
3 IBIS File Augmentation 32
3.1 Overview 32
3.2 IBIS Resources 32
3.3 Bus Definition 32
3.4 Pin Mapping 32
3.5 IBIS File GUI 33
3.5.1 Open a Template Project 33
3.5.2 Editing the Controller IBIS 34
3.5.3 Pin Mapping 38
3.5.4 Supported IBIS Keywords 39
4 Design Entry 41
4.1 Overview 41
4.2 Parallel Bus Analysis Templates 41
4.2.1 Rank Definition 44
4.3 Parallel Bus Analysis Wizard 46
5 Assigning Device Models 51
5.1 Overview 51
5.2 Design Files 51
5.3 Controller Model 52
5.3.1 Assign the Controller IBIS Model 52
5.3.2 Controller Bus Definition 54
5.4 Adding Memory Device Models 58
6 Interconnect Models 60
6.1 Introduction 60
6.2 Design Files 60
6.3 PCB Model Generation 60

6.3.1 PCB Block Editing 61
6.3.1.1 Layout Extraction Tab 61
6.3.2 PowerSI Model Extraction 63
6.3.3 Assign PowerSI Model 66
6.4 Package Models 69
6.4.1 Controller and Memory Package Model 69
7 Adding a VRM 70
7.1 Overview 70
7.2 Design Files 70
7.3 Define the VRM Block 70
8 Connecting the System 73
8.1 Introduction 73
8.1.1 Example Files 73
8.2 Connection Definitions 74
8.2.1 Memory Component 74
8.2.2 Connecting Controller Component 79
8.2.3 Editing the VRM Block 81
8.2.4 Unmapped Pins in MCP Editor 83
9 Simulation 85
9.1 Introduction 85
9.2 Tutorial Files 85
9.3 Analysis Options 85
9.4 Simulator 86
9.5 Simulation Configuration 87
9.6 Data Bus Write 89
9.6.1.1 Eye Distribution Methods 90
9.6.1.2 BER_Eyes 90
9.7 Stimulus Definition and Model Selection 90
9.7.1 Controller Setup 91
9.7.2 Memory Setup 94
9.7.3 Timing Budget 94
9.7.3.1 Timing Parameters Definition 96
9.7.3.2 Stimulus Offset Options 98
9.8 Restore Defaults 100
9.9 Simulation 100
9.9.1 Explore Sweep Manager 101
9.9.1.1 Sweep Type 101
9.9.1.2 Sweep List 102
9.9.1.3 Total Iterations 103
9.9.1.4 Results 106
9.9.1.5 Grouping Parameters 107
9.9.1.6 Model Sweep 108
9.9.2 Initial Simulation 109
9.9.3 Running Multiple Sweep 111
9.9.4 Data Presentation / 2D Curve Viewer 112
9.9.5 Simulation Directory 115
9.10 Additional Exercises 117
9.10.1 BER Analysis for DDR4 Interfaces 117
9.10.2 Crosstalk 119
9.10.3 Change IO Models 120
9.10.4 Non-Ideal Power 122
9.10.5 Worst Case Timing 123
9.10.5.1 Memory Bus in Write Mode 123
9.10.5.2 Memory Bus in Read Mode 125

9.10.6 Terminate Unconnected Nodes ... 126
9.10.7 Frequency Response ... 127
9.10.8 S-Parameter Extraction ... 129
9.10.8.1 Single-ended Mode ... 130
9.10.8.2 Differential Mode ... 132
9.10.9 S-Parameter View ... 134
9.10.10 Result Browser and 2D Curve Presentation ... 136
9.10.10.1 Result Browser ... 136
9.10.10.2 Curve Pane Context Menu ... 137
9.10.11 General Options ... 138
10 DDR Measurement Report ... 140
10.1 Introduction ... 140
10.2 Tutorial Files ... 140
10.3 Simulation Results ... 141
10.4 DDR Report Settings ... 142
10.5 Setup and Hold Derating Tables ... 146
10.6 DDR4 Threshold ... 147
10.7 LPDDR2 and LPDDR3 Threshold Application ... 148
10.8 Generate the Report ... 149
10.9 Report Contents ... 151
10.9.1 Table of Contents ... 151
10.9.2 Section 1 – General Information ... 153
10.9.3 Section 2 – Simulation Setup ... 153
10.9.4 Section 3 – DDR Measurement Setup ... 154
10.9.5 Section 4 – Results ... 154
10.9.5.1 Waveform Quality Report ... 154
10.9.5.2 Eye Quality Report ... 157
10.9.5.3 Timing Report ... 159
10.9.5.4 Timing Report – Strobe and Clock ... 162
10.9.5.5 Timing Report - Worst Case Condition ... 162
10.9.5.6 DQ Mask Report ... 166
10.9.5.7 Delay Report ... 167
10.9.6 Appendix ... 169
11 SystemSI with Other Tools ... 171
11.1 SystemSI Integration with Allegro Signal Explorer (SigXplorer) ... 171
11.2 SystemSI Integration with TimingDesigner ... 175
11.3 Auto Archive SSI Project ... 177

A APPENDIX: BATCH MODE SUPPORT 182

1 Parallel Bus Analysis Tutorial

1.1 Overview

Sigrity **SystemSI – Parallel Bus Analysis** is a system-level signal integrity (SI) analysis tool, focused on high speed parallel interfaces. It is integrated with Sigrity patented board and package modeling tools including PowerSI for extraction of accurate hardware interconnect models including real power distribution networks.

Parallel Bus Analysis supports HSPICE transistor-level and IBIS behavioral I/O models, including power-aware constructs from BIRD95 and BIRD98, to enable simulations with non-ideal power conditions. Interconnects like connectors and cables can be included with S-parameter data, or SPICE compatible circuit models.

Synchronous design performance metrics including eye diagrams with detailed timing measurements are available as outputs.

1.2 Tutorial Description

This tutorial covers the typical steps a user will go through to enter, edit and simulate a parallel bus interface.

- Chapter 2 has a **Quick Start Guide** that uses one of the simulation templates to introduce the **User Interface** and **Design Flow** for **Parallel Bus Analysis**.
- Chapter 3 provides reference information including IBIS file details.
- Chapters 4 through 9 provide a step-by-step overview on how to use **Parallel Bus Analysis** for a DDR3 system.
- Chapter 10 has advanced topics.

Files can be classified into two types: Templates and Samples.

- The template files for Chapter 2 and 3 are easily accessible from the GUI, through the File menu.
- The Sample files for Chapter 4 through 9 are in the `\Samples\SystemSI\Parallel Bus Analysis\Tutorial` directory, typically located here:

  `<INSTALL_DIR>\SpeedXP\Samples\SystemSI\Parallel Bus Analysis\Tutorial`

Note: *Instead of updating the files in the sample directory, it is recommended that you copy these files to your working directory and work on them.*

This document does not go into detail on theory. Please refer to **Application Notes** for additional documentation.

Contact Cadence support at si-support@cadence.com for more focused application assistance.

The example used in this tutorial is a simple DDR3 memory bus, including controller, PCB and DRAM devices.

Key topics that will be covered can be grouped into 3 broad categories: model creation, design entry, and simulation.

Each chapter is written to be a standalone module for users to follow for specific task-focused activity.

Please feedback any problems or issues found with this document to Cadence support.

2 Quick Start

2.1 Overview

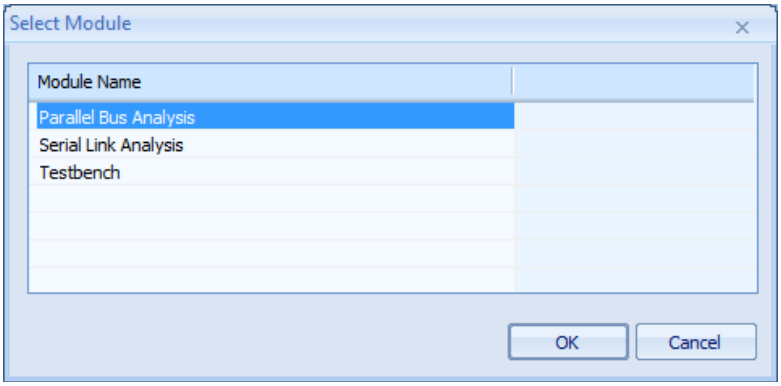
As an introduction to **Parallel Bus Analysis**, we will start by looking at one of the templates that are installed with the tool. This will allow us to look at the GUI, design entry commands, simulation setup and data presentation windows.

2.2 Open a Template

1. Launch SystemSI.
2. To open a new workspace, select *File -- New*.

   Alternatively, you can also click the **New** icon.

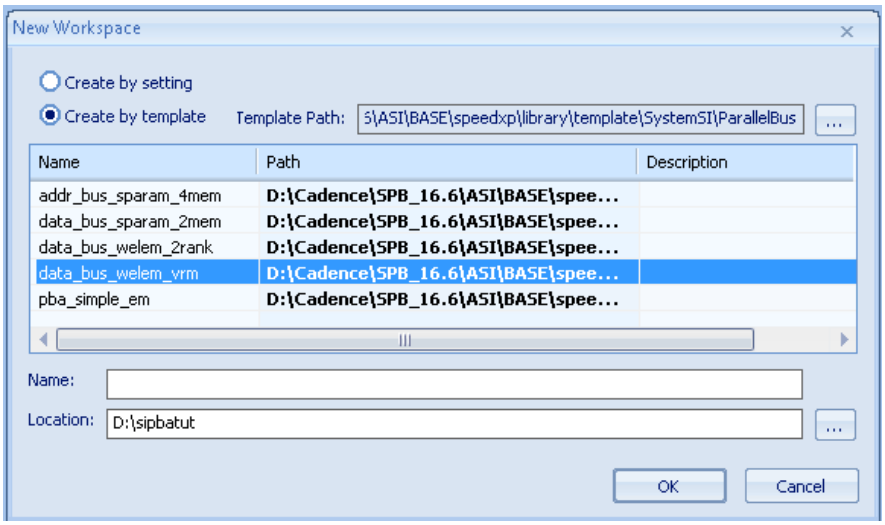
   The **Select Module** window opens.



3. Select **Parallel Bus Analysis**.
4. Click **OK**.

   The **Choose License Suites** window appears.
5. Select the license and click **Close**.

   The **New Workspace** window opens.
6. Select **Create by Template**.
7. Select **data_bus_welem_vrm**.

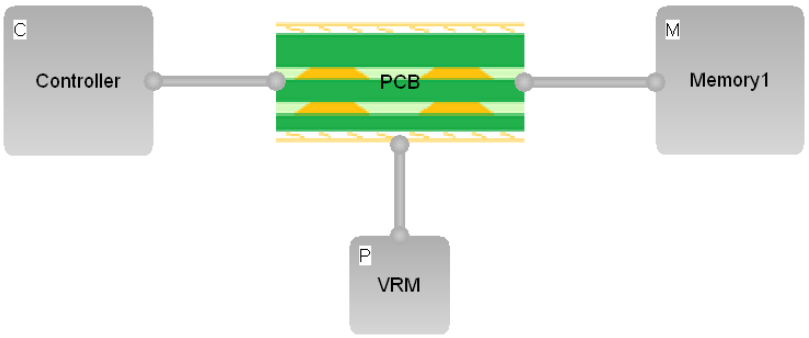


8. Enter a **Name** for the project. For this tutorial, enter the name as ***mysipba.***
9. Enter a **Location**.

NOTE!	A directory with the workspace name is created in the location specified. This directory contains the workspace file, *mysipba.ssix*, as well as all of the models for the memory controller, memory device and other bus blocks that are available in the single device template folder.

10. Click **OK**.

The Block diagram is created.



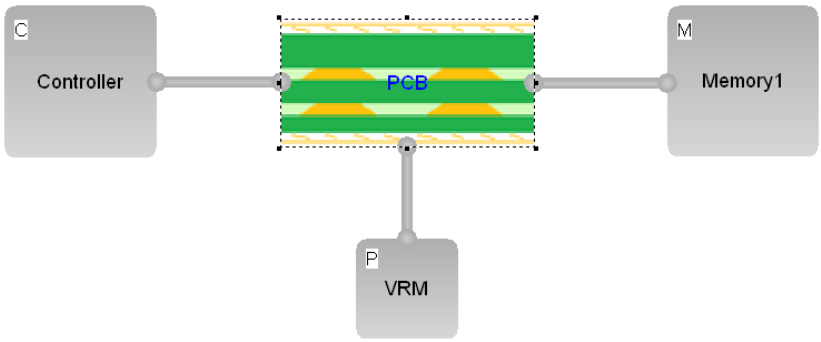
This is a pre-constructed, ready to simulate template, consisting of a controller and a single memory device. IBIS models are defined, edited, and linked to the devices. The PCB interconnects are coupled transmission line models, consisting of HSPICE W-element components. There is also a Voltage Regulator Module (VRM) block that has power supply explicitly defined. This is required for simulations including non-ideal power (i.e. no “ideal power” assumption at the IO models).

2.3 Editing Components

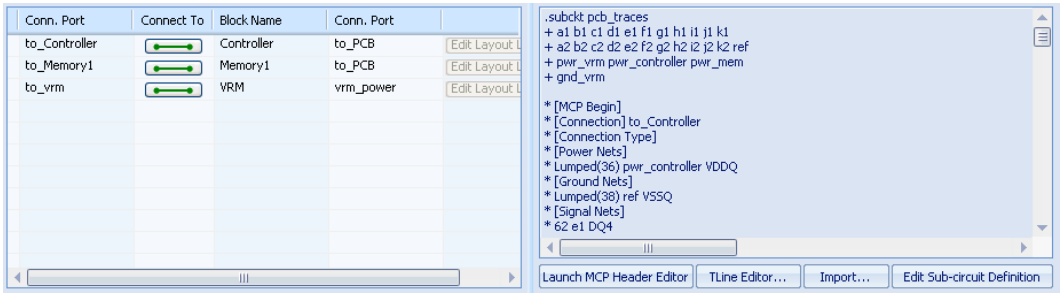
Irrespective of the method used to create the design, if required, you can edit the blocks as well as the connections. Editing a block and connection in the system is done by double-clicking the block.

2.3.1 PCB Block

1. Double-click the **PCB** component.

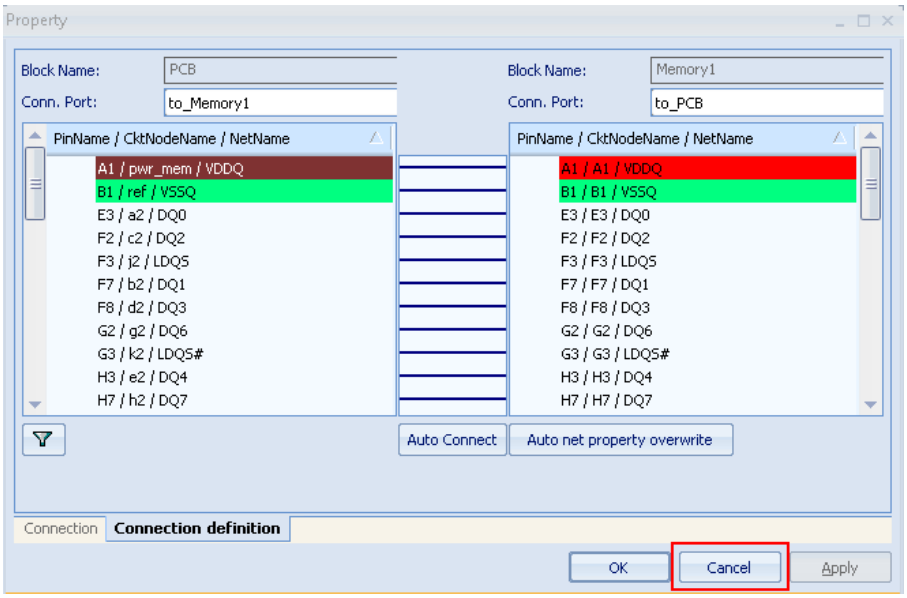


The **Property** dialog box opens. The left side of this dialog defines the **Connections** to this component, using Sigrity model-connection protocol (MCP).



2. Click the Green **Connect To** wires icons.

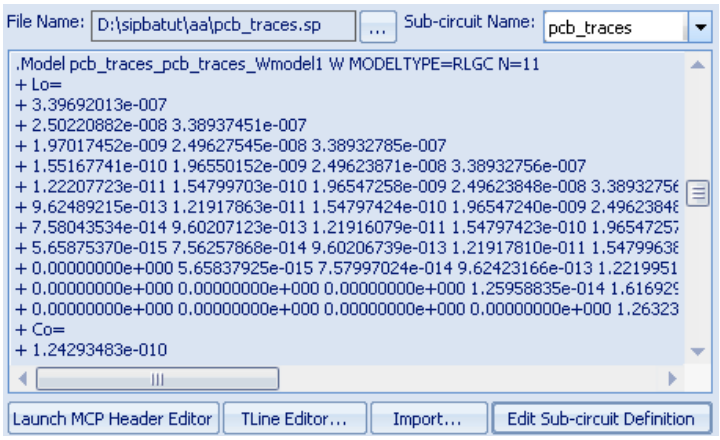
The MCP connectivity between these components, with specific pin to pin mapping, is shown.



3. Click **Cancel** to exit this MCP dialog without making changes.

   The right side of the Property dialog has model or circuit netlist definition for the component. This SPICE netlist file can be swapped for an alternative file, or edited in a text editor.

   The W-element description for the PCB traces in this file can be viewed by scrolling down as shown in the following figure.

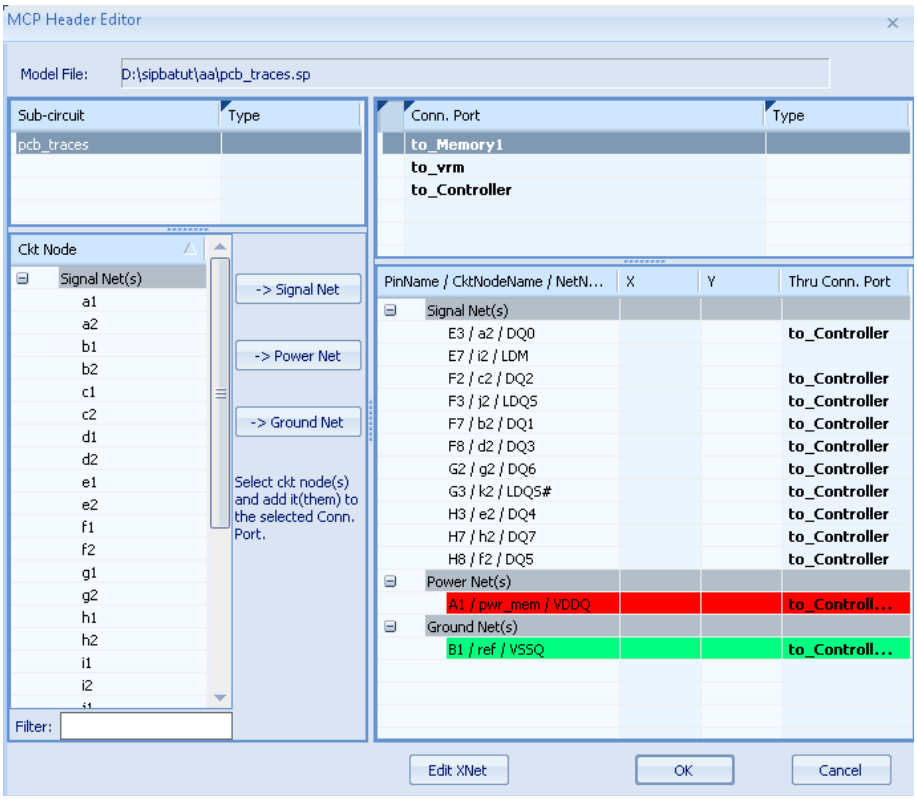


4. In case you need to edit the model or circuit netlist definition, click the **Edit Sub-circuit Definition** button.

   The file opens in editable mode in the Sub-circuit Definition Editor tab. You can make the required modifications to the file and click OK to save the changes.

5. For this design example, exit the Sub-circuit Definition Editor tab without making changes.

6. If required, you can add or modify the MCP header by selecting the **Launch MCP Header File** button.



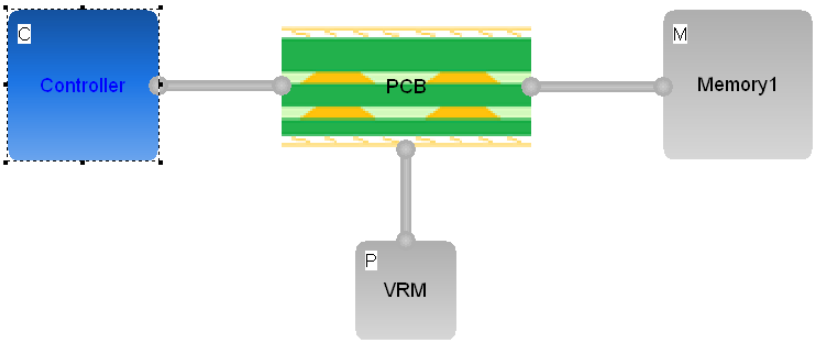
NOTE!	If the `.sp` or the `.ckt` file does not have the MCP header, you need to add the same by editing the MCP header. By default, models generated by Sigrity tools (`.sp` or `.ckt`), contains the MCP header.

7. Exit the MCP Header Editor without making any modifications.

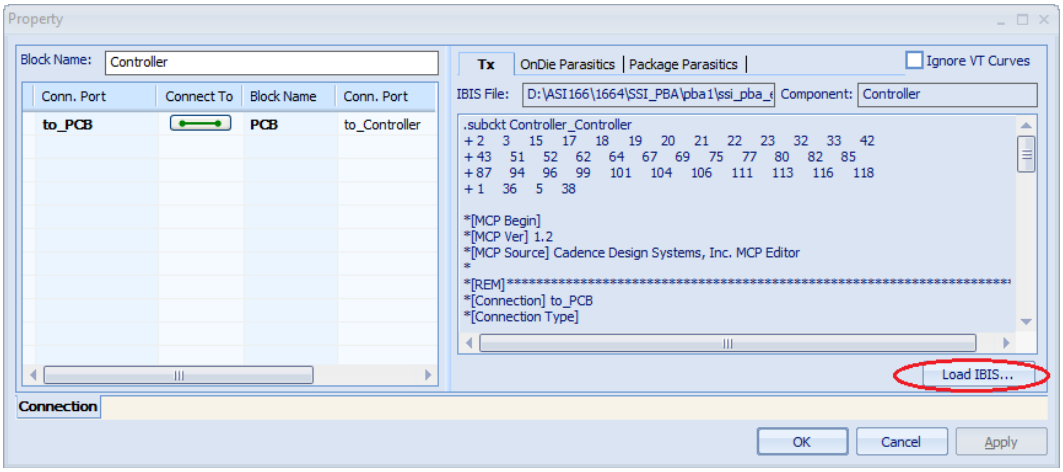
NOTE!	The **TLine Editor** button is used to define the properties of the transmission line block (PCB). Use the **Import** button to import the properties of the Tline block already defined in other project files. To know more about the usage of these buttons, see *Pre-Layout Transmission Line Modeling AppNote:SystemSI*.

2.3.2 Controller / Memory Block

1. Double-click the **Controller** or **Memory** component.

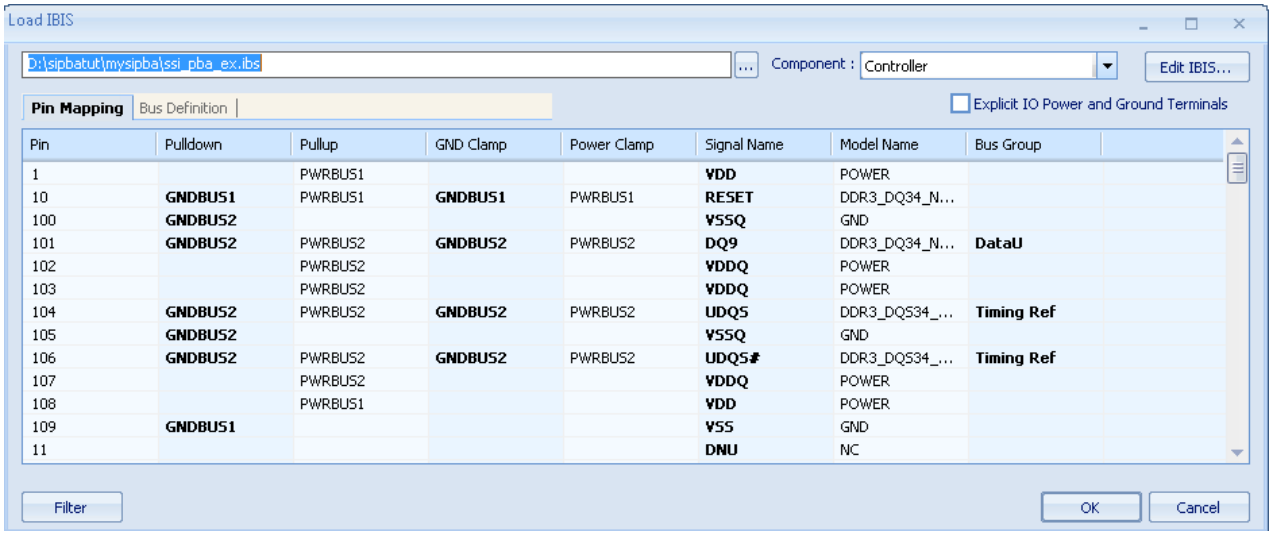


The **Property** dialog box that opens is similar to the one for the PCB, except for the editing function. Instead of a simple text editor, there is a **Load IBIS…** button.



2. Click the **Load IBIS…** button, and view the IBIS model definition for this Controller component.

   In this window, the IBIS models maybe specified, and the **Pin Mapping** and **Bus Definitions** for the model can be viewed and defined. Changes made in this window will update the IBIS file.



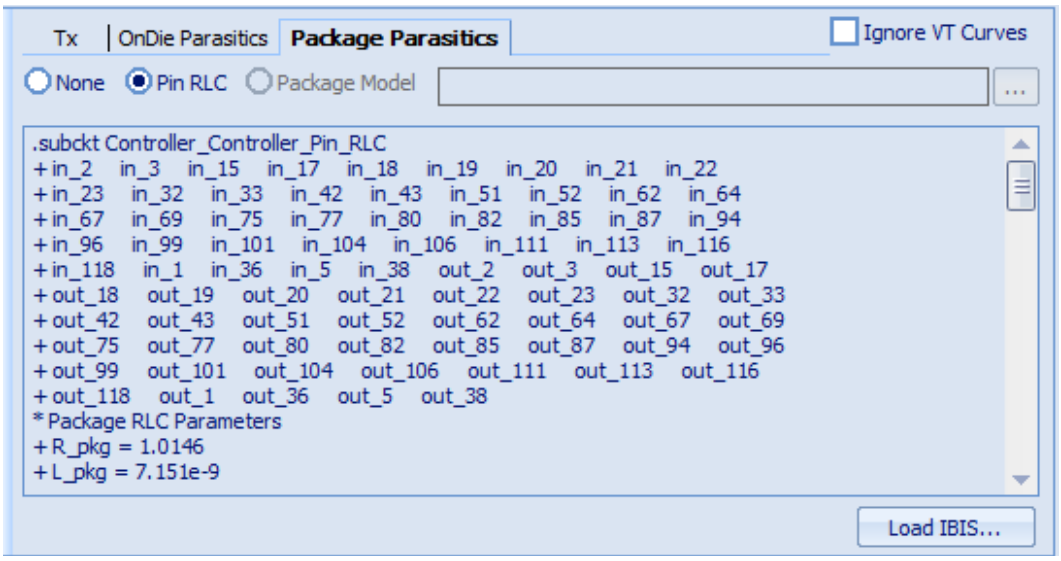
   The MCPs of the Controller and Memory blocks are generated by SystemSI based on the bus definition, Pin Mapping, and setup of the **Load IBIS** GUI. These are not editable.

   To know more about the MCP generation process, see Rules for Generating MCPs.

3. Click **Cancel** to exit without making changes.

   In the **Property** dialog, the OnDie Parasitics and Package Parasitics tabs can be used specify the **Die** and **Package** models, respectively. These can be separate model files, like a PKG file, or simply the RLC package parasitics that are already included in the IBIS model.

4. To view the Pin model in the IBIS file, click the **Package Parasitics** tab and select **Pin RLC**.

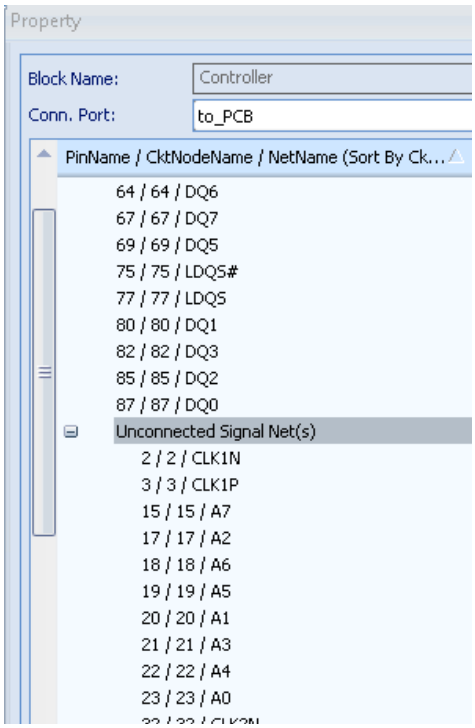


NOTE!	For Parallel Bus Analysis, the **Ignore VT Curves** option is not selected by default. This is to ensure that during analysis, simulator takes into account the non-liner behavior of the drivers by reading the VT data.

2.3.2.1 MCP Generation for Controller and Memory IBIS Blocks

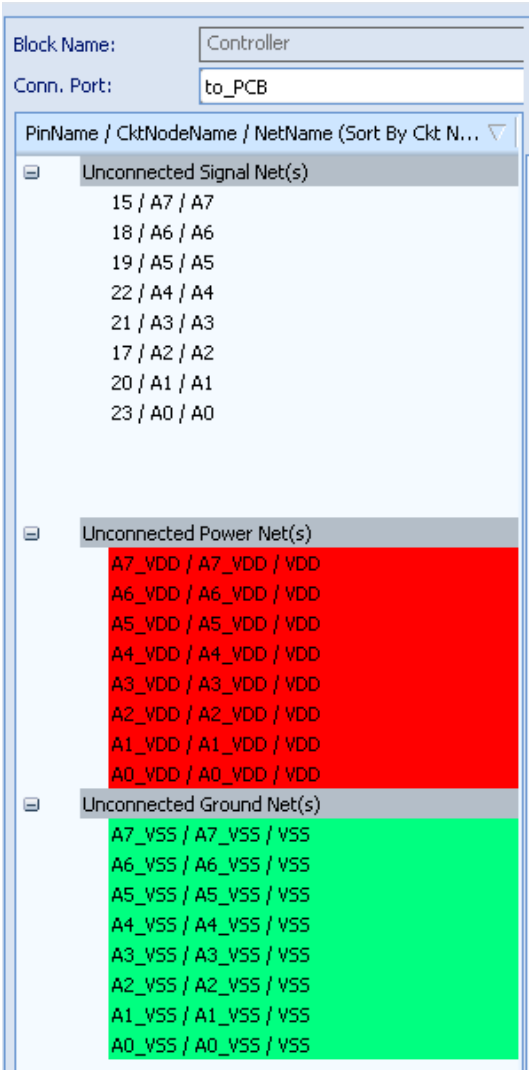
This section details the rules or the guidelines followed by SystemSI for generating MCPs.

1. In the Load IBIS dialog box, if the **Explicit IO Power and Ground Terminals** option is not checked, the **Ckt Node**, **Net,** and **Pin** in the MCP are determined by the **Pin Mapping** section of the IBIS component.



- The values in the **Ckt Node** and **Pin** columns are the same, and are read from the **Pin** column in the **Pin Mapping** section of the Load IBIS dialog box.
- The values in the **Net** column of the MCP Editor are same as the values in the **Signal Name** column in the **Pin Mapping** section of the Load IBIS dialog box.

2. If the Load IBIS dialog box has the **Explicit IO Power and Ground Terminals** option checked, in the generated MCP, each signal has its corresponding power and ground as shown in the following figure.



The MCP is generated using following rules:

- In case of a regular signal, the **Ckt Node** value is same as the **Signal Name** in the **Pin Mapping** section
- For the corresponding power signals, the values in the **Ckt Node** and **Pin** columns are the same, and these are derived by combining the signal Net and power Net.
- Similarly, for the Ground of the signal, the **Ckt Node** and **Pin** values are the same, and these are derived from the signal Net and the ground Net.

You can click **Ckt Node** of the Controller/Memory block to sort the **Ckt Node** for easy connection.

The signal and its power and ground are grouped together for the Controller/Memory block in order to make the manual connecting easy.

NOTE!	When Explicit IO Power and Ground Terminals are utilized, all the powers and grounds of the Controller/Memory must be connected in order to do the non-ideal power bus simulation.

2.3.3 EBD Block

While using SystemSI you can also import the Electrical Board Description (EBD) models that are as per the IBIS specification. EBD models are imported directly into SystemSI and automatically expanded out to include interconnect and referenced component blocks. EBD files are commonly used for modeling DIMMS.

NOTE!	In SystemSI you cannot instantiate an EBD block along with a memory block. This is not supported.

2.3.3.1 EBD File

When you associate an EBD model to an EBD block in SystemSI, the model information is displayed in the four tabs of the Load EBD dialog box.

IBIS Tab

This tab lists all the Memory blocks and Terminators listed as Reference Designators. To view the Pin Mapping of a memory block or a terminator, from the **Reference Designator** drop-down list, select the refdes of the component.

Note: A Terminator can have its own IBIS component; but all the Memory blocks must share the same component of the same ibis file.

Bus Definition

This tab lists the Bus groups defined for the Memory component. If required, use this tab to add and define new bus groups.

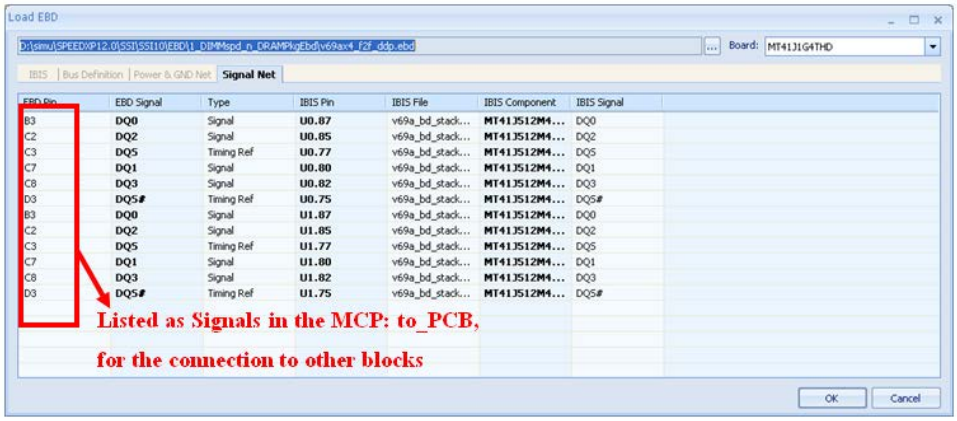
Power & GND net

The EBD model does not have the definitions for the Power and Ground nets. In order for the IBIS components to get the power supply from outside the EBD, each IBIS Power Pin must be linked to one EBD Power Pin that is connected to other blocks. The same is true for the IBIS Ground Pins.

NOTE!	Only the IBIS Power and Ground Pins that are associated with the defined Bus signals are listed in the **Power & GND net** tab.

Signal Net

The IBIS Pins are for the defined Bus signals of the Memory blocks. If a signal is defined in the EBD file, it's IBIS Pin and the EBD Pin will be listed together in the same row. All the EBD pins are listed in the MCP are use for the connection to the other blocks.



2.3.3.2 Using EBD Models in System SI

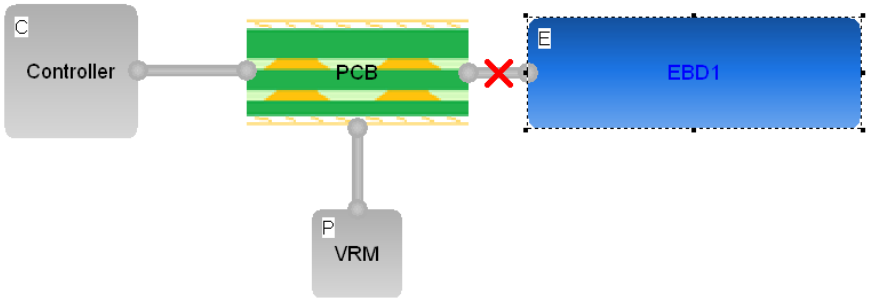
To import an EBD models in SystemSI, you first add an instance of EBD block on the schematic and then associate the EBD file with the EBD block.

To instantiate an EBD block:

1. Click the button.

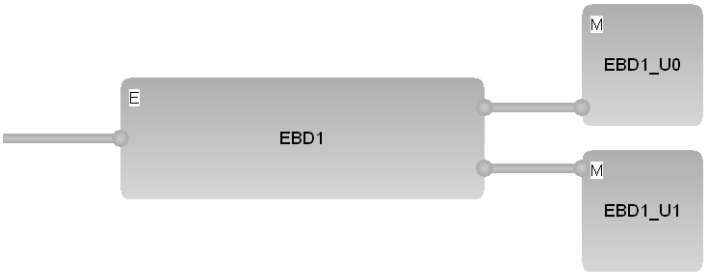
   Note: *This button is enabled only if you do not have a memory block instantiated on the canvas.*

2. Click on the canvas to add an instance of EBD component.
3. To connect EBD block to the PCB block, right-click on the EBD1 instance and from the pop-up menu choose **Add Connection Between – PCB**.



4. To assign an EBD model to the EBD block, open the Property window for the EBD block.
5. Click the Load EBD... button
6. In the Load EBD dialog box, use the Browse (…) button to navigate to the .EBD file.
7. As you load the EBD file, the IBIS, Bus Definition, Power & GND Net and Signal Net for the model can be viewed in the respective tab. If required you can modify these. Changes made in the IBIS and Bus Definition panels will update the IBIS files.
8. Click OK to close the Load EBD dialog box.
9. Click the Apply button to build the EBD model.

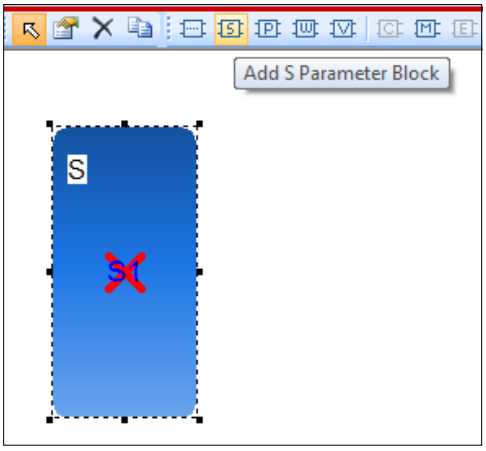
Depending of the EBD model, new Memory blocks are automatically added to the schematic.



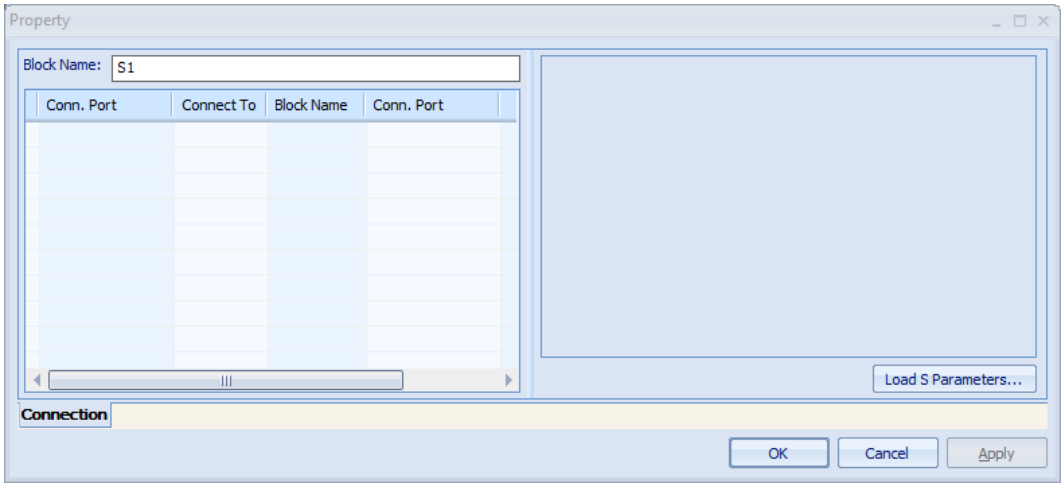
2.3.4 S Parameter Block

2.3.4.1 Add S Parameter Block

1. Click the **Add S Parameter Block** icon on the tool bar, and click in the **Layout** window.
   The S Parameter block **S1** is added.



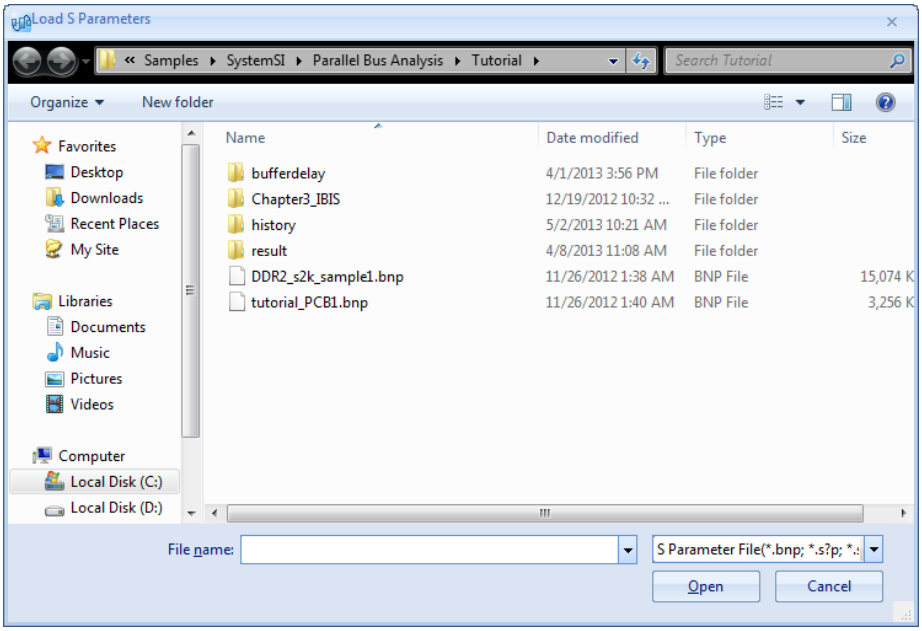
2. Double-click the **S1** block.
   The **Property** pane opens.



2.3.4.2 Load S Parameter File

1. Click the **Load S Parameters…** button.

The **Load S Parameters** window opens.

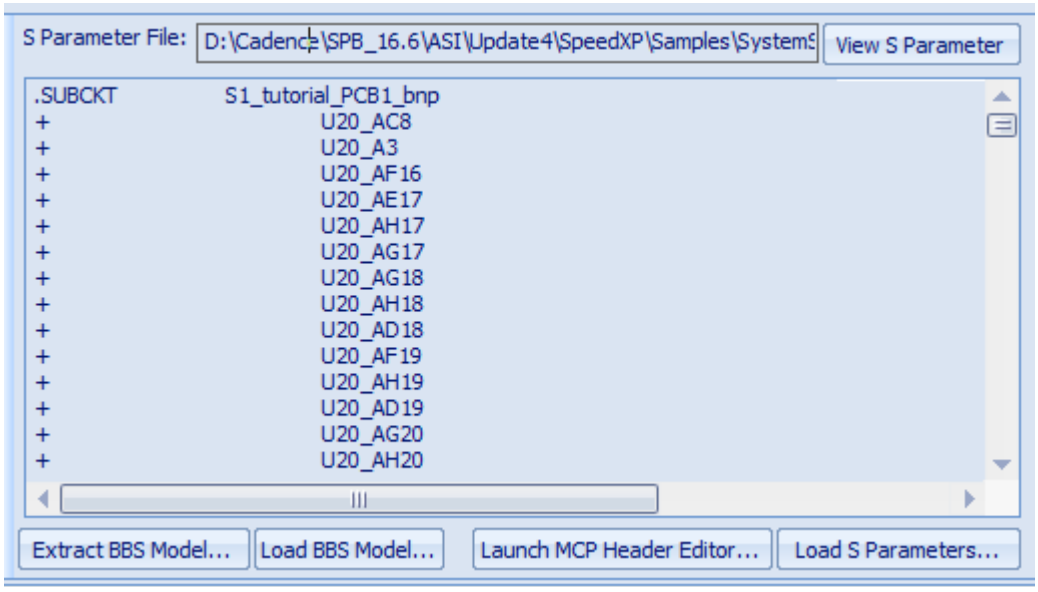


2. Load the S Parameter file (BNP or Touchstone) to the S Parameter block.

   For this tutorial, load the **tutorial_PCB1.bnp** file available at *<install_dir>/SpeedXP/Samples/SystemSI/Parallel Bus Analysis/Tutorial*

   Once the selected S Parameter file is successfully loaded, `.sp` file is automatically generated and loaded to the block.

   - The `.sp` filename is displayed in the **File Name** field.
   - The S Parameter filename is displayed in the **S Parameter File** field.



   - If the BNP file has the MCP information, the MCP section is automatically added to the .sp file for connection.
   - In case the Touchstone or BNP files do not have the MCP information, you are expected to manually add the MCP through the MCP Editor.

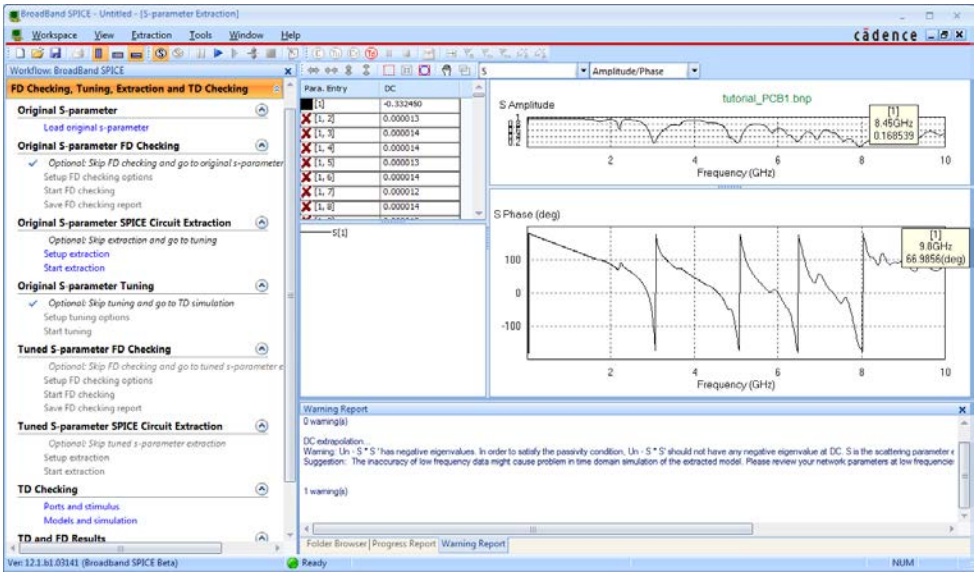
NOTE!	Editing of `.sp` file for the S Parameters is NOT recommended.

2.3.4.3 Extract the BBS Model

To improve convergence and simulation performance, it is sometimes advantageous to convert the S-parameters into a Broadband Spice circuit model. This section covers the steps to generate a circuit model from the S-parameter model used in the previous section.

1. In the Property dialog box for the S-parameter block, click the **Extract BBS Model…** button.

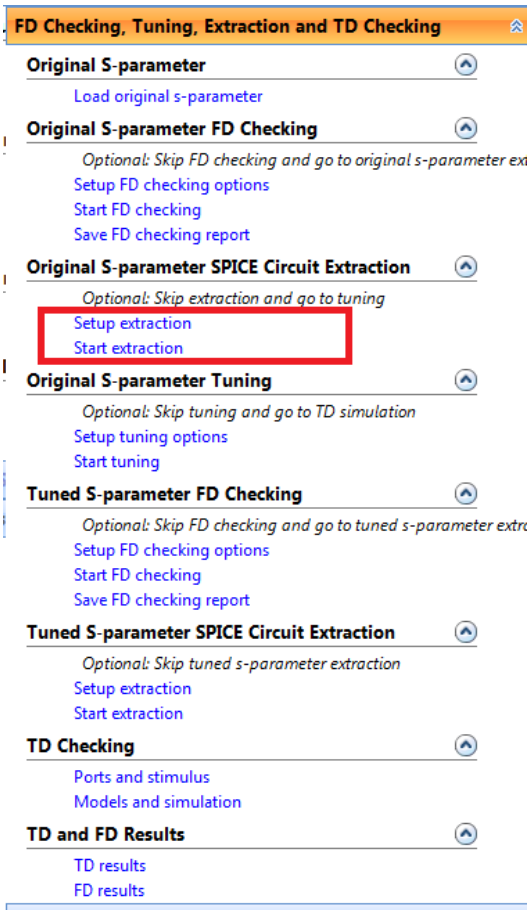
   The BroadbandSPICE application is launched.



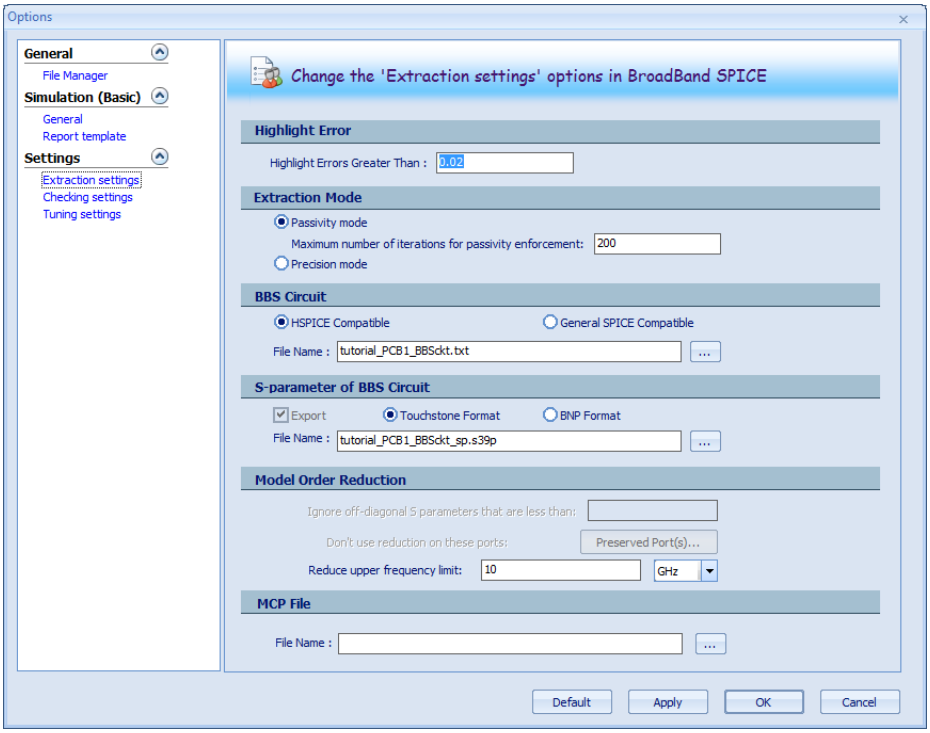
2. Click the **FD Checking, Tuning, Extraction and TD Checking** workflow to check and tune the S Parameters.

   For the application of BroadbandSPICE, refer to ***BroadbandSPICE_Tutorial.pdf*** and ***BroadbandSPICE_UG.pdf***.

3. If the S Parameter checking result looks good to you, click **Setup extraction** in the workflow.



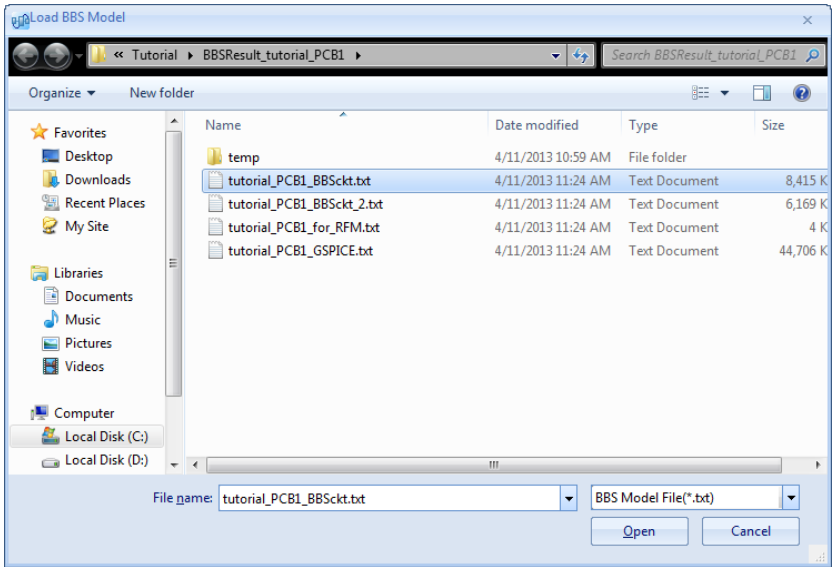
The **Options** window opens.



4. Set up the extraction settings as desired, and click OK to save your settings and to close the Options window.
5. Click **Start extraction** in the workflow to extract the BBS model.

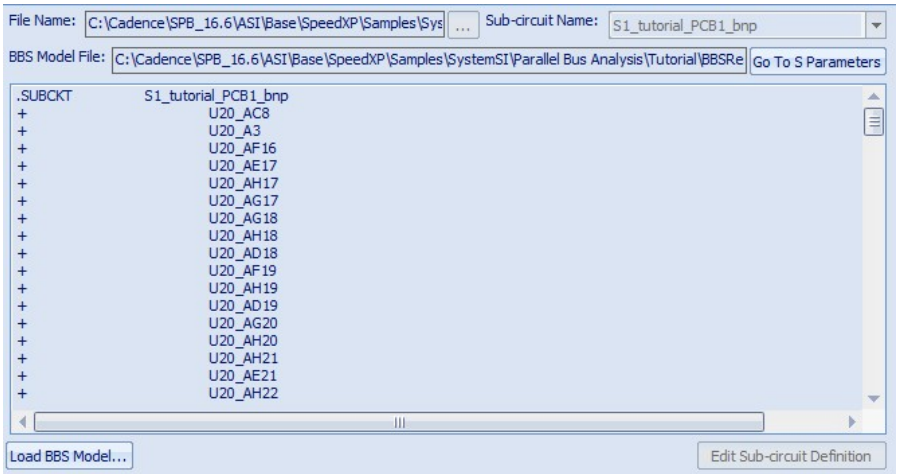
   If the BBS model extraction is successful, a `.txt` file is automatically generated in the folder that has the S-parameter (`.bnp`) file.

   You can now load the generated BBS Model in SystemSI.
6. To load the generated BBS model **tutorial_PCB1_BBSckt.txt**, from the SystemSI GUI, click the **Load BBS Model…** button.
7. Select the circuit file and click Open.



Once the **BBS model is s**uccessfully loaded, an .sp file is **automatically generated** and loaded to the block.

- The `.sp` file is displayed in the **File Name** field
- The BBS model `.txt` file is displayed in the **BBS Model File** field



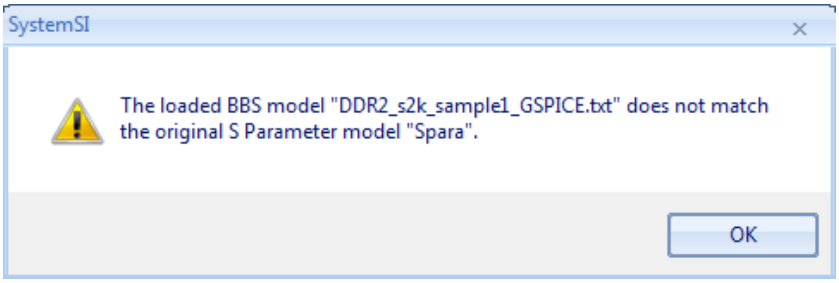
NOTE!	Editing of `.sp` file for the BBS model is NOT recommended.

2.3.4.4 Load the BBS Model

If you have the BBS models for the selected S Parameters, click the **Load BBS Model…** button to directly load the BBS .txt file.

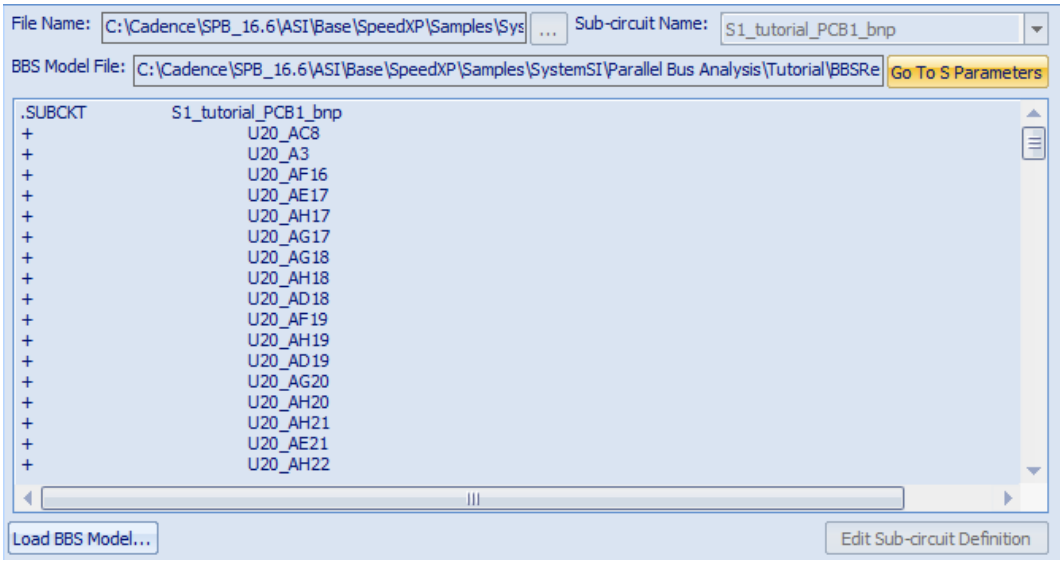
Please refer to **Step 6** in ***Section 2.3.4.3 Extract the BBS Model*** for details.

If the loaded BBS model does not match the original S Parameters, an error message will be issued.

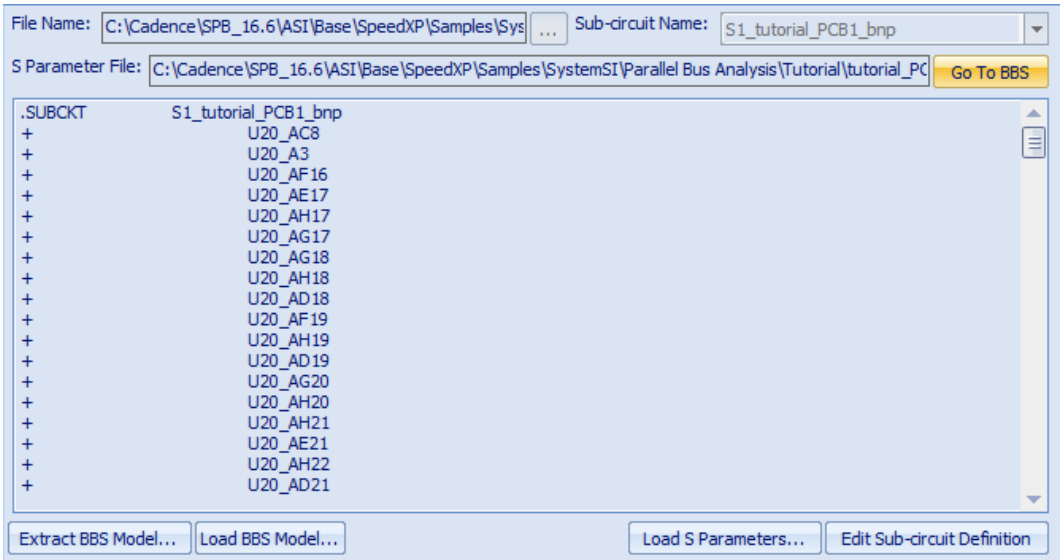


2.3.4.5 Switch the Models

1. Click the **Go To S Parameters** button to switch to the original S Parameters model.



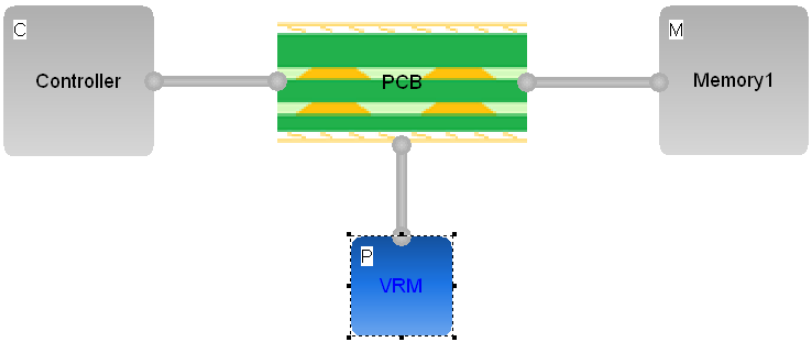
2. Click the **Go To BBS** button to switch back to the BBS model:



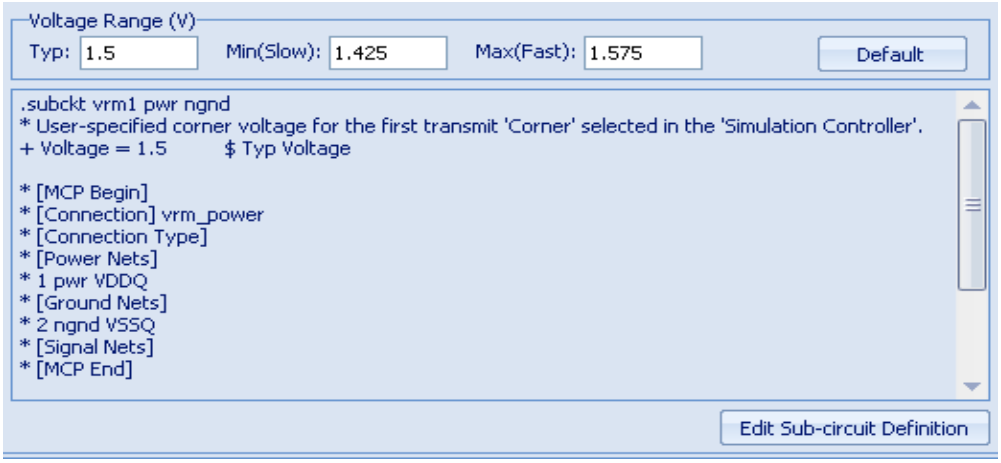
NOTE!	The MCP section will be shared among the BBS models and the original S Parameter model. Any MCP change to one .sp file through the MCP Editor will automatically update the MCP section in other .sp files.

2.3.5 VRM Block

1. Double-click the **VRM** component.



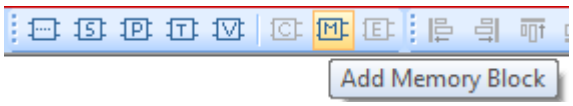
The VRM component for this template is simply an ideal DC supply of 1.5 Volts. Parallel Bus Analysis parameterizes the corner voltages here, including, **Min**, **Typ**, and **Max** VRM voltages, for fast, consistent simulation of the IBIS corner models.



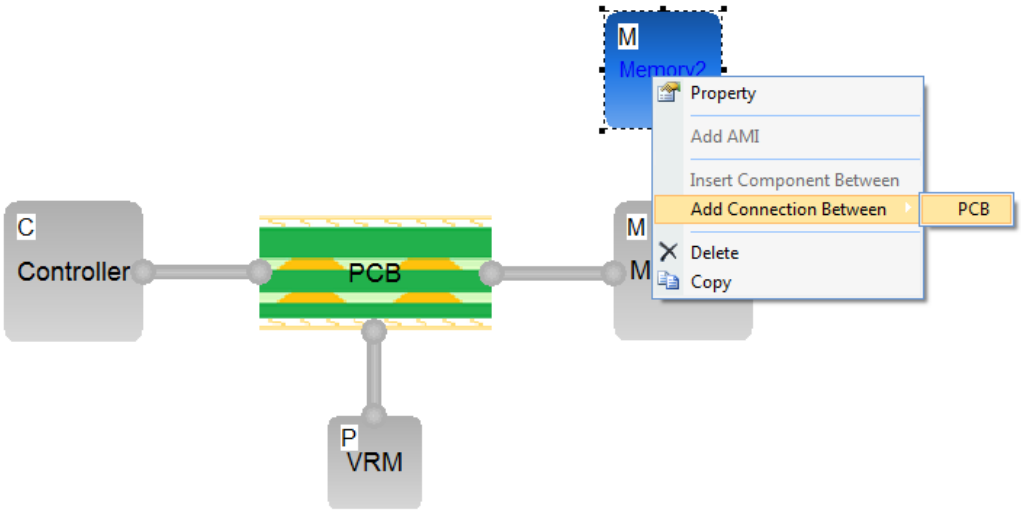
2.4 Adding New Components

You can add new components to the schematic, by selecting appropriate tool button from the Add Block toolbar. This section demonstrates the procedure to add a new memory block to the schematic.

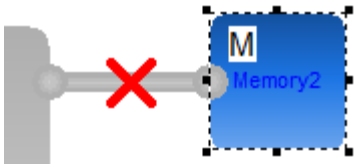
1. To add a new Memory block, click the **Memory Block** component on the toolbar, and place it on the schematic.



2. To add a connection of this block to the **PCB** block, right-click on the Memory2 component, and choose **Add Connection Between -- PCB**.



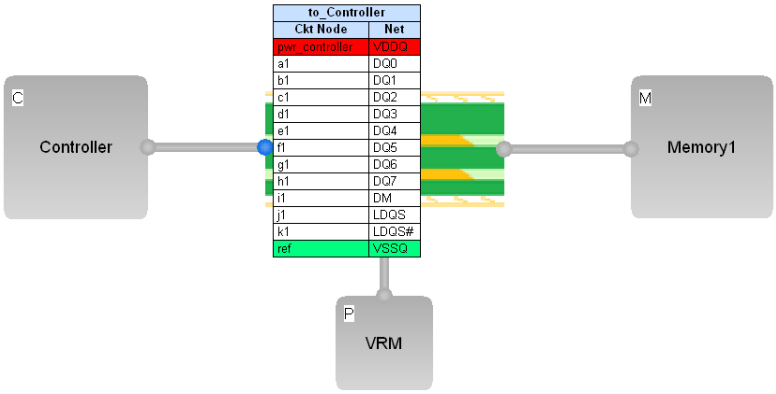
3. Because the MCP connection to the new component has not been defined, there is a Red X on the PCB to Memory2 connection. (Note: Delete Memory2 block before you go on the following steps.)



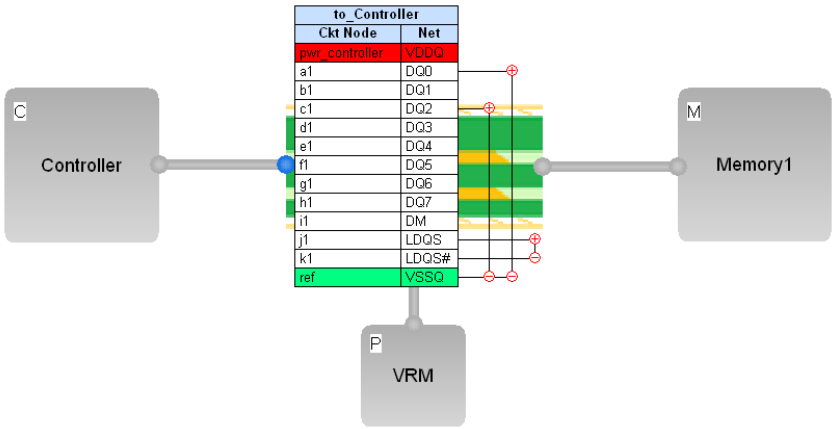
User can use this method to add other components.

2.5 Setting up Probe Points

1. Click the connection point to open the **Ckt Node** menu.



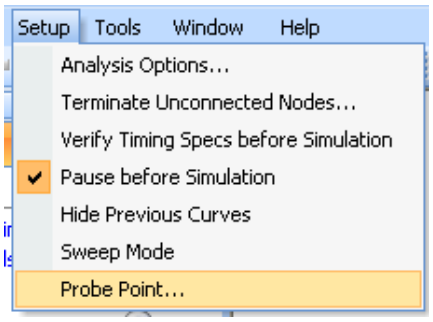
2. To set Probe Point for the simulation, click two Ckt Nodes in the menu.



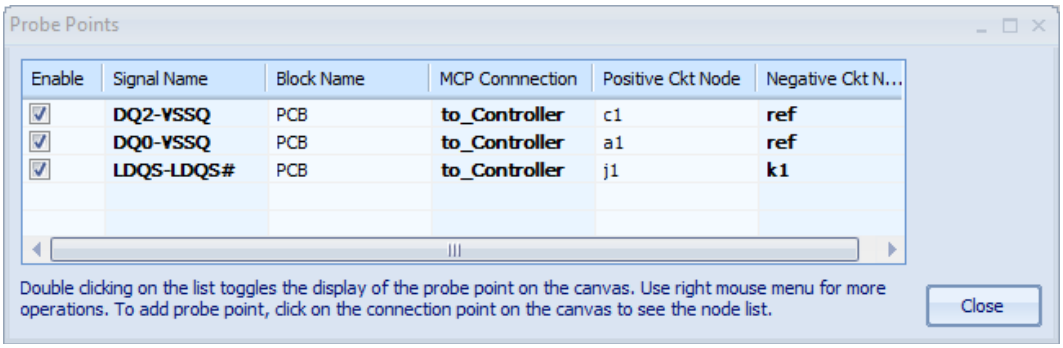
- The two selected nodes are connected.
- The node clicked first is defined as positive, and the second is negative.

  In this section, setup Probe points between c1 and ref, a1 and ref, and j1 and k1.

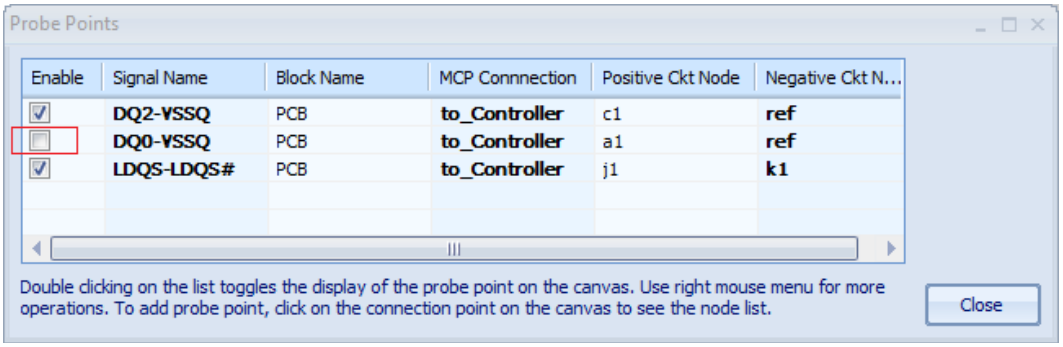
3. To view and edit the defined Probe Points, from the ***Setup*** Menu, select Probe ***Point***….



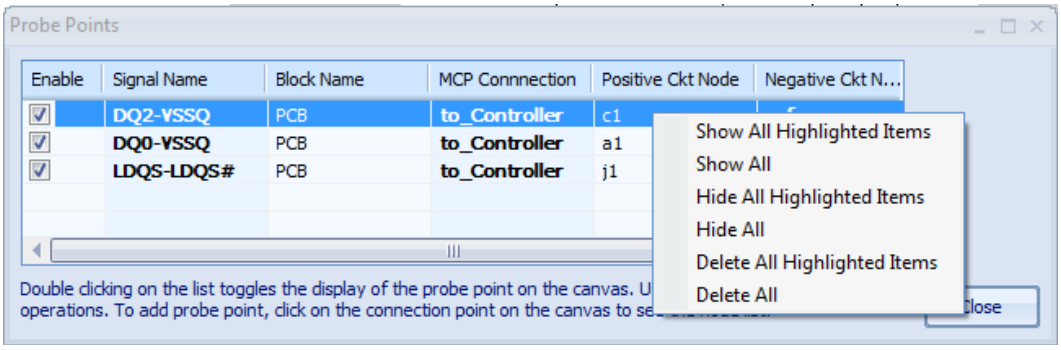
The **Probe Point** window opens as shown in the following figure.



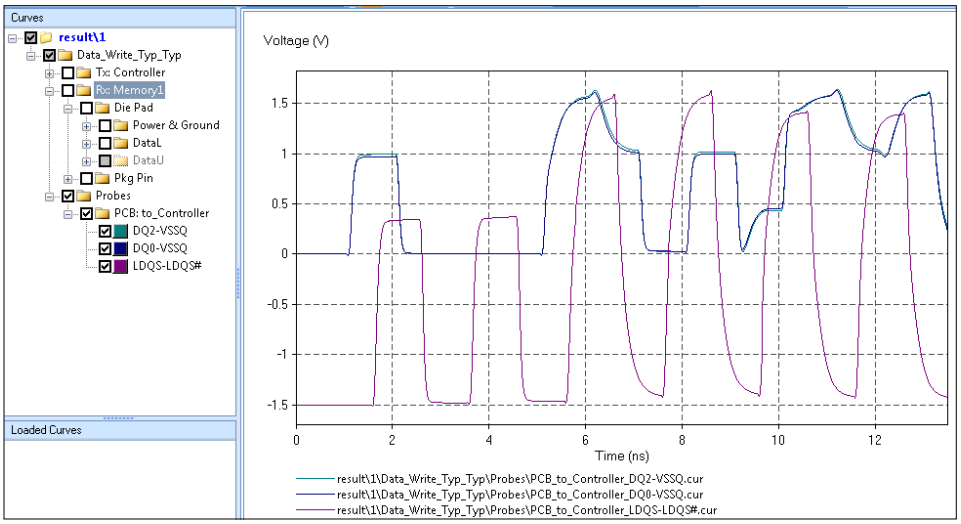
4. By default, all probe points are enabled. To disable a probe, clear the check box next to the probe, as shown.



5. To show or delete the defined Probe Points, right-click and select from the pop-up menu.



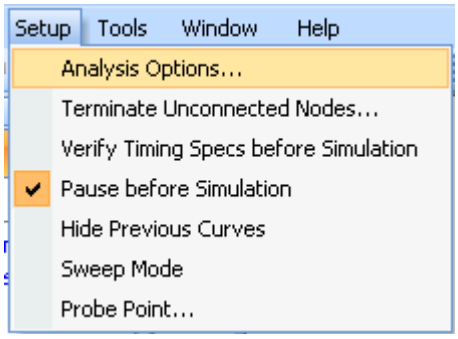
NOTE: After the simulation, the curves of all the defined Probe Points can be viewed from the 2D Curve Window.


Curves
result\1
Data_Write_Typ_Typ
Tx: Controller
Rx: Memory1
Die Pad
Power & Ground
DataL
DataU
Pkg Pin
Probes
PCB: to_Controller
DQ2-VSSQ
DQ0-VSSQ
LDQS-LDQS#
Loaded Curves
Voltage (V)
1.5
1
0.5
0
-0.5
-1
-1.5
0
2
4
6
8
10
12
Time (ns)
result\1\Data_Write_Typ_Typ\Probes\PCB_to_Controller_DQ2-VSSQ.cur
result\1\Data_Write_Typ_Typ\Probes\PCB_to_Controller_DQ0-VSSQ.cur
result\1\Data_Write_Typ_Typ\Probes\PCB_to_Controller_LDQS-LDQS#.cur

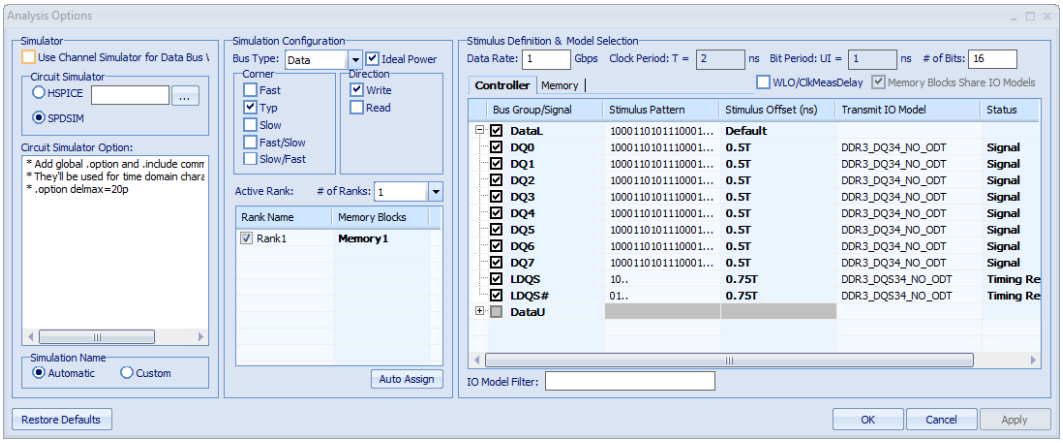
2.6 Setting up the Simulation

The Analysis Options Dialog sets most of the key aspects of the simulation.

1. From the Setup menu, select ***Analysis Options***….

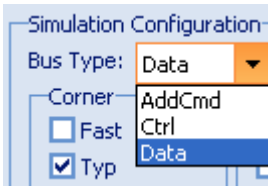


The Analysis Options dialog box displays.

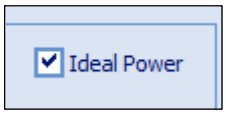


In this dialog box, you can specify the following simulation settings.

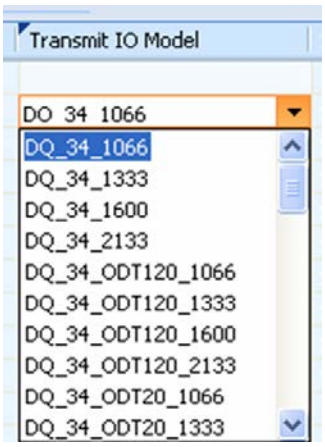
- If the design has more than one type of Bus defined, select the Bus type to be simulated.



- Simulation can be run assuming Ideal or Non-ideal Power.



- IO models for Controller and Memory may be selected through the GUI.

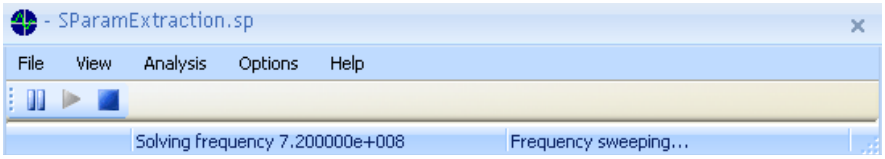


To save Simulation Settings, click on the OK button.

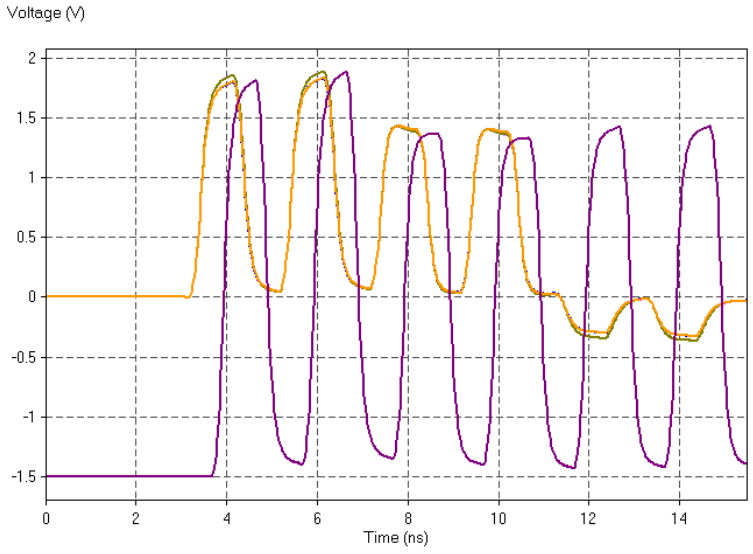
2.7 Running the Simulation

1. Start the simulation with the default settings, by clicking the **Play** icon.

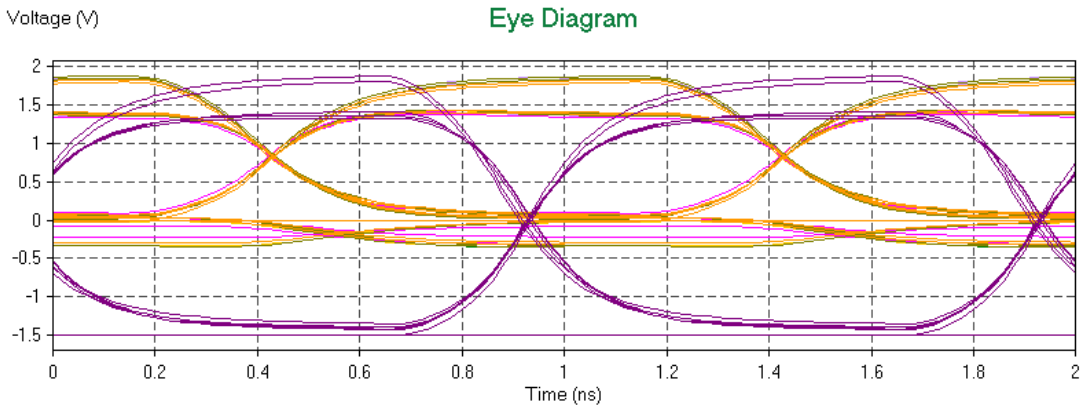
2. The simulation status opens.



3. The time domain waveforms are shown in a 2D curve Display, upon completion.



4. Use the toolbar icons to view the Eye Diagrams.



The paths for the various templates are automatically generated during **SystemSI - Parallel Bus Analysis** installation and cannot be modified.

NOTE!	Do not change the name or location of the template folders, i.e. **`<INSTALL_DIR>\SpeedXP\Library\template\SystemSI\ParallelBus`**.

It is not advisable to change the contents of any of the template folders, since they are used to create the new workspace directory with all of the component models, connectivity, and settings.

2.8 Custom Templates

Advanced users can create and edit their own templates to utilize as a starting point for SystemSI projects. To do this,

1. Create a self-contained SystemSI project, with all required models and connectivity, in a directory, say, `my_new_template`.
2. Within that directory, name the SystemSI project the same name as its directory, `my_new_template.ssix`.
3. Remove any extraneous data, for example result and history folders, from the project and place the directory at the same location as that of default templates **`<INSTALL_DIR>\SpeedXP\Library\template\SystemSI\ParallelBus.`**

The new template will appear upon SystemSI startup for your use.

3 IBIS File Augmentation

3.1 Overview

SystemSI - Parallel Bus Analysis is compatible with the I/O Buffer Information Specification (IBIS), including BIRD95 (composite current) and BIRD98 (gate modulation effect) to allow non-ideal power and ground IO modeling. Interconnects, including boards, connectors, and cables can be modeled with frequency-domain S-parameter data or SPICE-compatible subcircuits, and connected together with explicit power and ground connections to preserve SI and PI behavior related to planes and power/ground pin performance. Transistor-level I/O models in HSPICE format may also be included through the IBIS model interface. Connectivity between components is fast and robust with Model Connection Protocol (MCP).

In order to enable the automation provided by SystemSI - Parallel Bus Analysis, some augmentation of the standard IBIS files is required. Specifically, the definition of bus groups, timing reference signals, and setup/hold specifications are needed. This can all be done through the GUI, and comments are automatically embedded into the IBIS file itself.

In this chapter, the user will learn about key IBIS file dependencies for Parallel Bus Analysis that must be present in the IBIS files for Controllers and Memory. These include the Pin Mapping specification, and the new Bus Definition. We will use a template file to examine and edit a sample IBIS model.

An HSPICE transistor-level driver model in IBIS will be demonstrated.

3.2 IBIS Resources

SystemSI - Parallel Bus Analysis supports the most recent IBIS specification. This popular behavioral modeling format is useful for memory system components like controllers and DRAM devices. To read more about IBIS, please reference the following material.

- IBIS specification

  http://www.eigroup.org
- Pointers to IBIS models of different IC manufacturers

  http://www.eigroup.org/IBIS/ibis%20table/models.htm

3.3 Bus Definition

SystemSI - Parallel Bus Analysis introduces the ability to define signals as a bus and also identify timing reference signals for the bus. This is needed for both Controller and Memory components, and is comprised of a new IBIS file block with associated keywords. The Bus Definition classifies buses into three categories, Clock, Address, or Data. Though bus categorization information can be added to the IBIS file through a text editor, but it is highly recommended that Parallel Bus Analysis GUI is used to create the Bus Definition.

3.4 Pin Mapping

According to the IBIS specification, the [PinMapping] section is used to indicate the power and ground buses to which a given driver, receiver, or terminator is connected. It accepts the following parameters:

pulldown_ref, ***pullup_ref***, ***gnd_clamp_ref***, ***power_clamp_ref***, ***ext_ref***.

According the IBIS specification, the [PinMapping] section is optional, and may not exist in every IBIS file. If this section is not available in any IBIS file, Bus Parallel Bus Analysis is performed

assuming ideal power conditions. It is strongly recommended that a valid pin-mapping specification be obtained from the device manufacturer and included in the IBIS file.

3.5 IBIS File GUI

In this section, we will use the SystemSI user interface to open a template project, edit the controller block, and view the Bus Definition section of the controller IBIS file. A new bus will be added to the Controller. The subsequent changes to the IBIS file will be shown.

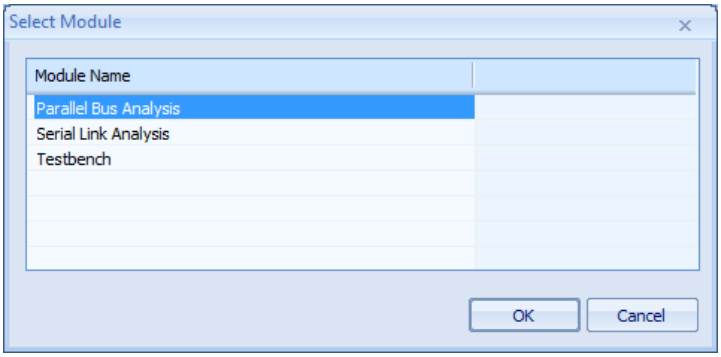
3.5.1 Open a Template Project

1. Launch SystemSI.
2. To open a new workspace, select File -- New.

   or

   Click the **New** button.

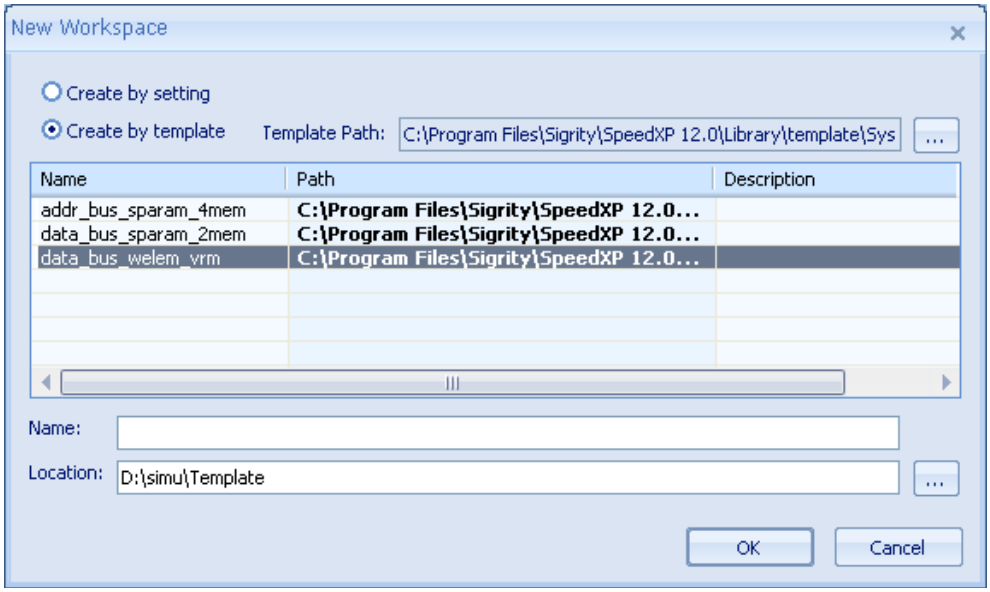
   The **Select Module** window opens.



3. Select **Parallel Bus Analysis**.
4. Click **OK**.

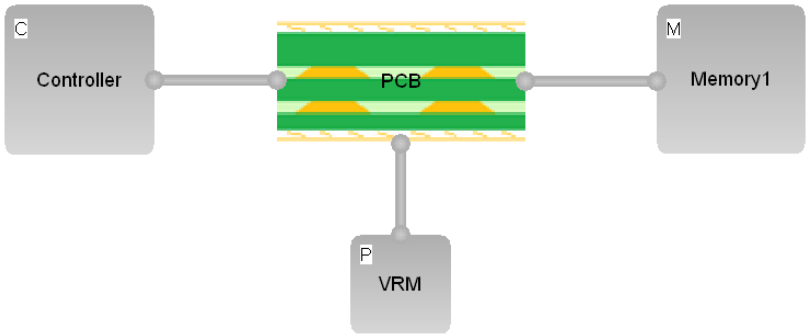
   The **New Workspace** window opens.

5. Select **Create by Template**.
6. Select **data_bus_welem_vrm**.



7. Enter a **Name** for the project.
8. Enter or browse a **Location**.
9. Click **OK**.

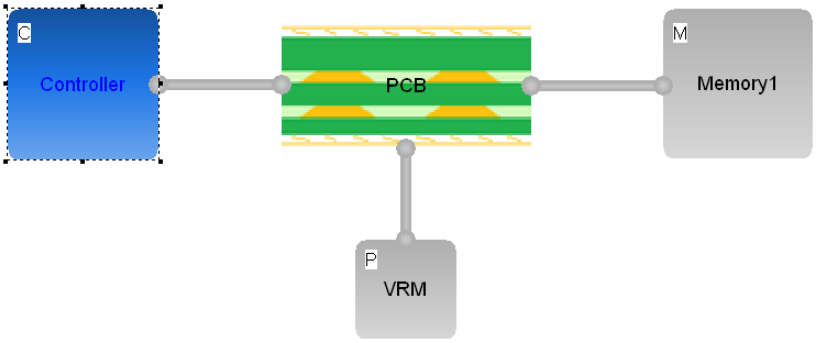
The Block diagram is created as follows.



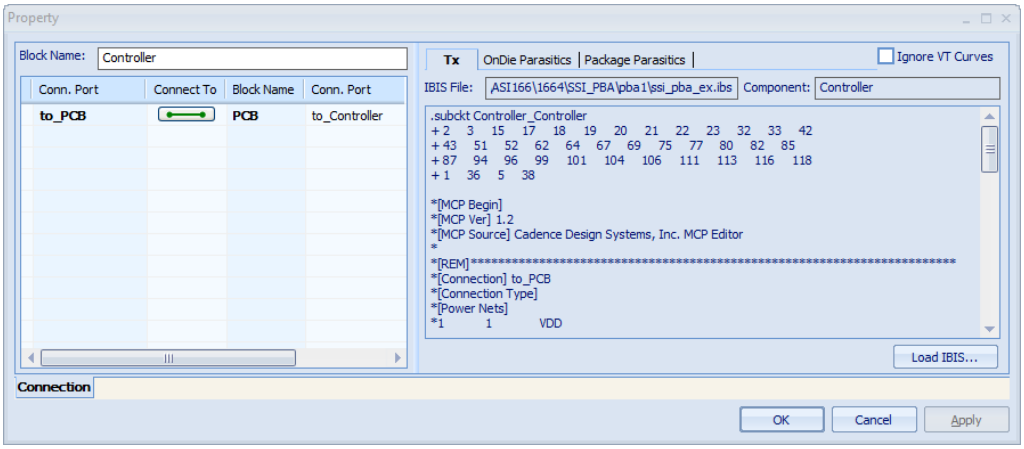
This is a pre-constructed, ready to simulate, template consisting of a controller and a single memory device. IBIS models are defined, edited and linked to the devices. The PCB interconnects are coupled transmission line models consisting of HSPICE W-element components. There is also a VRM supply.

3.5.2 Editing the Controller IBIS

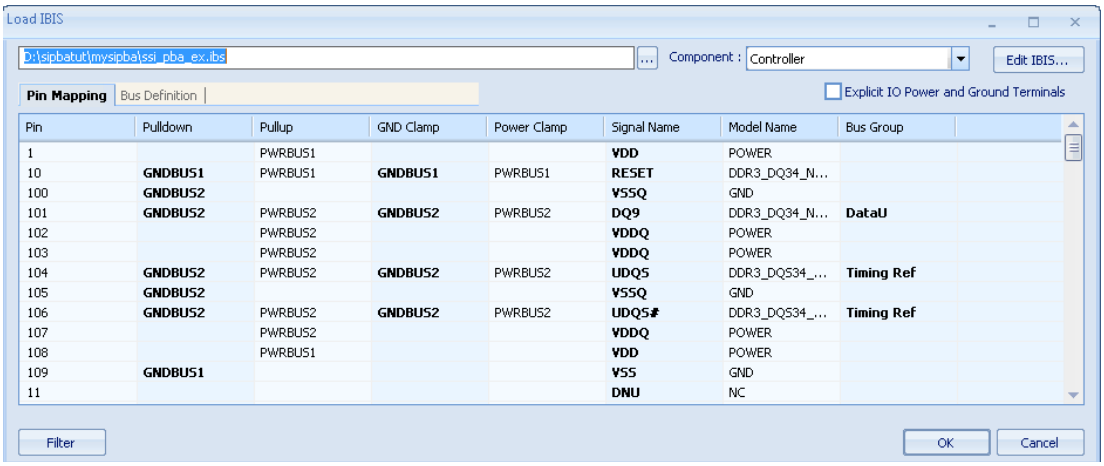
1. Double-click the **Controller** or **Memory** component.



2. In the **Property** dialog box that opens, click the **Load IBIS** button.

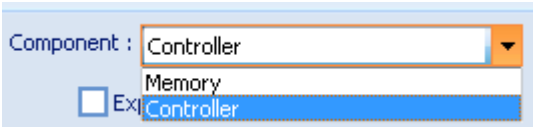


3. View the IBIS model definition for this **Controller** component. In this window, the IBIS model file and component is specified, and the **Pin Mapping** and **Bus Definitions** for the model can be viewed and defined.

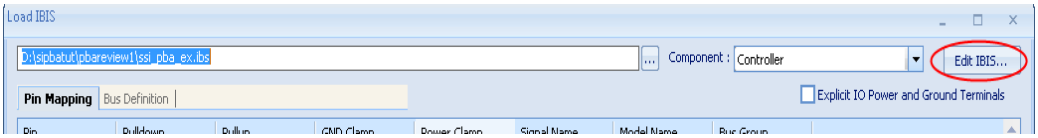


4. Click the file Browser  to see the IBIS files available in this template. Exit the file dialog without making any changes.

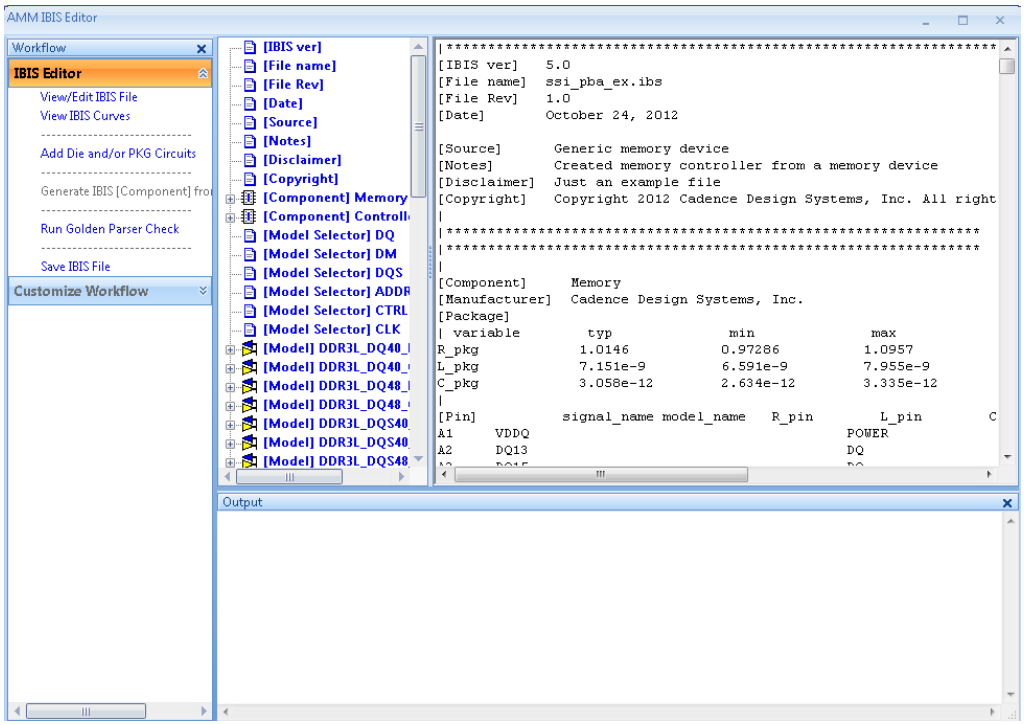
5. Click the **Component** pull-down to see the components in this IBIS file. There is one memory controller and a memory device model in this file. Again, no changes are necessary.



6. To edit the IBIS file, select the Edit IBIS button.

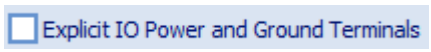


The IBIS file opens in AMM IBIS Editor. If required, you can modify the IBIS file, view IBIS curves, add die or package data and run golden Parser checks.



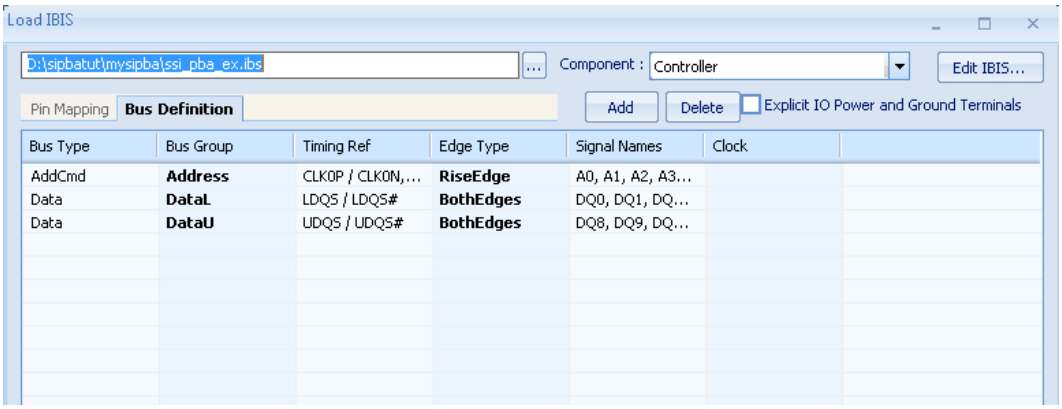
7. For this module, no changes are required, so close the AMM IBIS Editor, without making any modifications.

   Note that, by default, the **Explicit IO Power and Ground Terminals** check box is not selected.



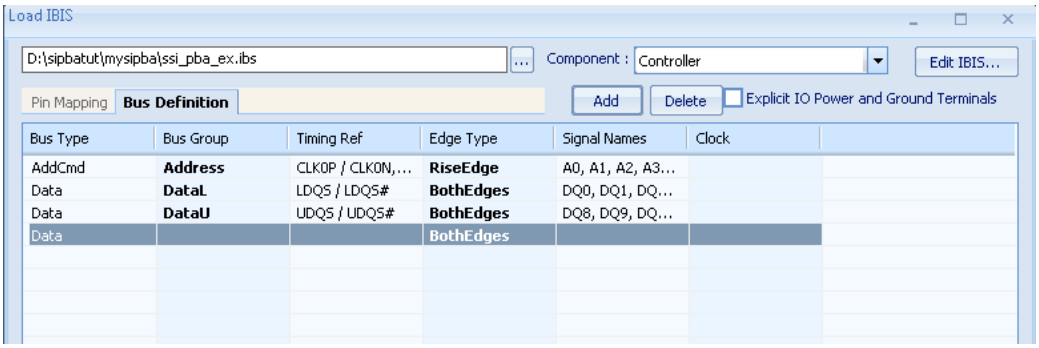
NOTE!	***Section 2.3.2 Controller / Memory Block***, details the impact of selecting this dialog box.

8. Click the **Bus Definition** tab, to see the Buses that have been defined for this controller.



9. Add a new **Bus Group** by clicking **Add**.

   A new **Bus Group** is added.



10. Each field in the new bus definition may be edited, some with pull down menus, others by typing, and yet others by selecting net names.

    Enter the Bus Group as **Calibration**. On completion, the bus definition should be same as shown in the following figure.

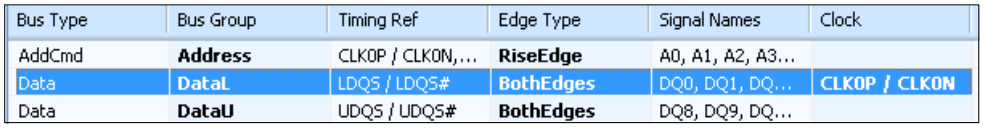
Bus Type	Bus Group	Timing Ref	Edge Type	Signal Names
AddCmd	**Address**	CLK0P / CLK0N,...	**RiseEdge**	A0, A1, A2, A3...
Data	**DataL**	LDQS / LDQS#	**BothEdges**	DQ0, DQ1, DQ...
Data	**DataU**	UDQS / UDQS#	**BothEdges**	DQ8, DQ9, DQ...
Data	**Calibration**	CLK1P / CLK1N	**BothEdges**	BA0, BA1, BA2

    For data buses, you can also specify the Clock. This can be same as the Timing Reference signal of the Ctrl or the AddCmd bus groups.

11. Add the clock signal as CLK0P/CLK0N, as shown in the following figure.

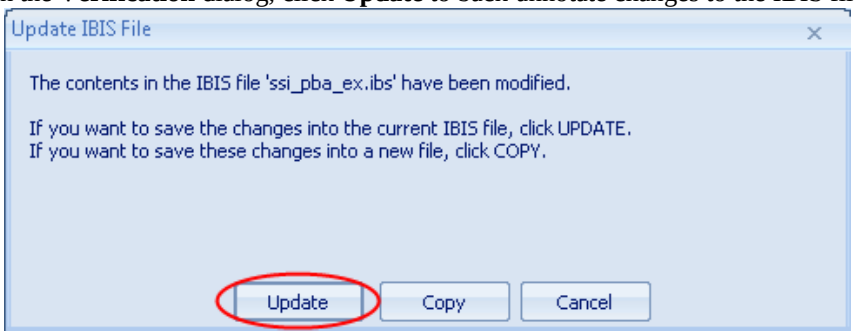
    You can also modify the existing group.

12. Add a clock signal, CLK0P/CLK0N, to the DataL bus, as shown in the following figure.



13. Click **OK** to accept the changes in the Bus Definition tab.

14. In the **Verification** dialog, click **Update** to back-annotate changes to the IBIS file.



    The new Bus can be viewed in the ssi_pba_ex.ibs file, in the Controller Definition section, just after the Pin Mapping.

```
|**********Component Bus Definitions ******************************
|[Component Type]    Controller
|[Bus Type]               AddCmd
|BusGroup                 Address
|TimingRef                CLK0P / CLK0N, CLK1P / CLK1N, CLK2P / CLK2N, CLK3P / CLK3N
|EdgeType                 RiseEdge
|MinTransmitSetup         0.3125UI
|MinTransmitHold          0.3125UI
|SignalName          A0 A1 A2 A3 A4 A5 A6 A7
|
|[Bus Type]               Data
|BusGroup                 DataL
|TimingRef                LDQS / LDQS#
|EdgeType                 BothEdges
|Clock                    CLK0P / CLK0N
|MinTransmitSetup         0.3333UI
|MinTransmitHold          0.3333UI
|MaxReceiveSkew(+)        0.25UI
|MaxReceiveSkew(-)        0.25UI
|SignalName          DQ0 DQ1 DQ2 DQ3 DQ4 DQ5 DQ6 DQ7
|
|[Bus Type]               Data
|BusGroup                 DataU
|TimingRef                UDQS / UDQS#
|EdgeType                 BothEdges
|MinTransmitSetup         0.3333UI
|MinTransmitHold          0.3333UI
|MaxReceiveSkew(+)        0.25UI
|MaxReceiveSkew(-)        0.25UI
|SignalName          DQ8 DQ9 DQ10 DQ11 DQ12 DQ13 DQ14 DQ15
|
|[Bus Type]               Data
|BusGroup                 Calibration
|TimingRef                CLK1P / CLK1N
|EdgeType                 BothEdges
|Clock                    CLK0P / CLK0N
|SignalName          BA0 BA1 BA2
```

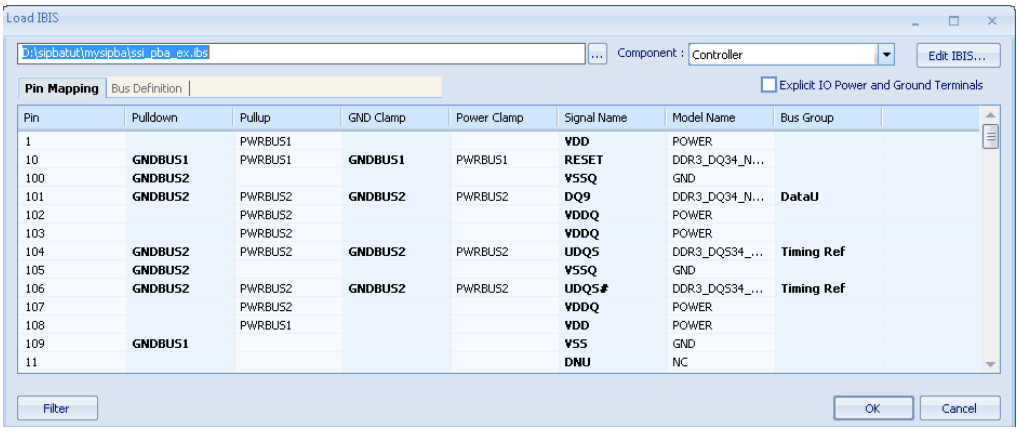
NOTE!	The IBIS model for the Memory component has similar elements. The reader is encouraged to explore the details of the Memory component. Since editing functionality is identical to that for the Controller, details are not reviewed here.

3.5.3 Pin Mapping

The Pin Mapping section is required by SystemSI for simulations with non-ideal power, for example to consider SSN effects. This section is considered optional in the IBIS specification, and some components do not have Pin mapping defined. In such cases, SystemSI automatically defaults to an “Ideal Power” mode for all simulations. To get access to all of the powerful features of the tool, including non-ideal power simulation, it is strongly recommended that a valid Pin Mapping section be added to the IBIS file prior to use in SystemSI.

This section, details the Pin Mapping definition of the controller.

1. From the **Load IBIS** window, view the **Pin Mapping** tab for the Controller.



The Pin Mapping section contains important information required for parallel bus analysis.

- The references, including Pullup, Pulldown and Gnd/Pwr rails for the Bus Signals, are identified in the Pin Mapping. Without this information, Parallel Bus Analysis assumes Ideal Power and Ground for all simulations.
- The Pin Name to Signal Name mapping is shown. These pin names are used to connect components together in Parallel Bus Analysis.

  For details, see the section on MCP Connectivity in Chapter 1.
- The relevant bus group (assuming the bus definitions are already specified) is listed.

2. To close the Pin Mapping without saving, click **Cancel**.
3. Save and close the work space.

3.5.4 Supported IBIS Keywords

Sigrity SPDSIM currently parses the following data sections from an IBIS file:

[Component]	[Voltage Range]	[On]	[Define Package Model]
[Driver Schedule]	[Pullup Reference]	[Off]	[Manufacturer]
[Pin]	[Pulldown Reference]	[R Series]	[Number Of Sections]
[Pin Mapping]	[POWER Clamp Reference]	[L Series]	[Number Of Pins]
[Package]	[GND Clamp Reference]	[R1 Series]	[Pin Numbers]
[Package Model]	[Pulldown]	[C Series]	[Model Data]
[Series Pin Mapping]	[GND_clamp]	[Lc Series]	[End Model Data]
[Series Switch Groups]	[GND Clamp]	[Rc Series]	[Resistance Matrix]
[Model]	[POWER_clamp]	[Series Current]	[Inductance Matrix]
[Model Selector]	[POWER Clamp]	[Series MOSFET]	[Capacitance Matrix]
[Submodel]	[Rgnd]	[Ramp]	[Row]

[Submodel Spec]	[Rpower]	[Rising Waveform]	[Bandwidth]
[GND Pulse Table]	[Rac]	[Falling Waveform]	[End Package Model]
[POWER Pulse Table]	[Cac]	[Composite Current]	[end]
[ISSO PU]	[ISSO PD]		

4 Design Entry

4.1 Overview

SystemSI provides an intuitive schematic entry GUI for capturing system designs as block diagrams. You can base your designs on pre-defined templates shipped with the tool, or you can use a design wizard that comes with Parallel Bus Analysis, to create your designs. In this chapter, both the methods of creating new designs are explained.

NOTE!	Chapters 4 through 9 go through a sequential Bus system design session, from design entry through simulation and data processing. It focuses on an actual DDR3 design, with Controller, Memory devices, PCB and VRM. The user can go through each chapter sequentially, to get a comprehensive understanding of the SystemSI – Parallel Bus Analysis flow, or they may choose to go though selected chapters as needed.

Design files are saved for each chapter in the **\Samples\SystemSI\Parallel Bus Analysis\Tutorial** directory, to allow users to simply open specific files, and review exercises for those sections. In the installation hierarchy, the Tutorial directory is available at:

<INSTALL_DIR>\SpeedXP\Samples\SystemSI\Parallel Bus Analysis\Tutorial

4.2 Parallel Bus Analysis Templates

Two design templates are available, one that uses S-parameter model for PCB, and one that uses W-element model for the PCB. The W-element template, ***data_bus_welem_vrm*** was covered in Chapter 2. Refer to this chapter for an overview of the W-element model template.

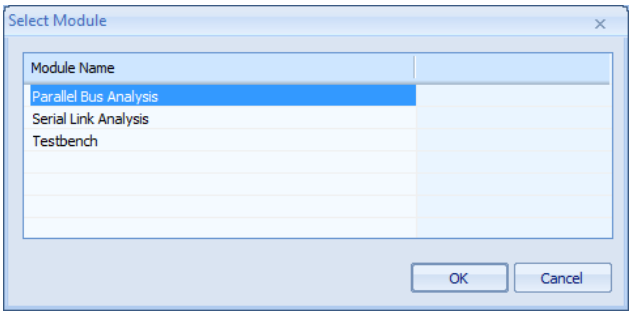
This section demonstrates how to create a design using the S-parameter template.

1. Launch SystemSI.
2. To open a new workspace, select **File -- New**.

   or

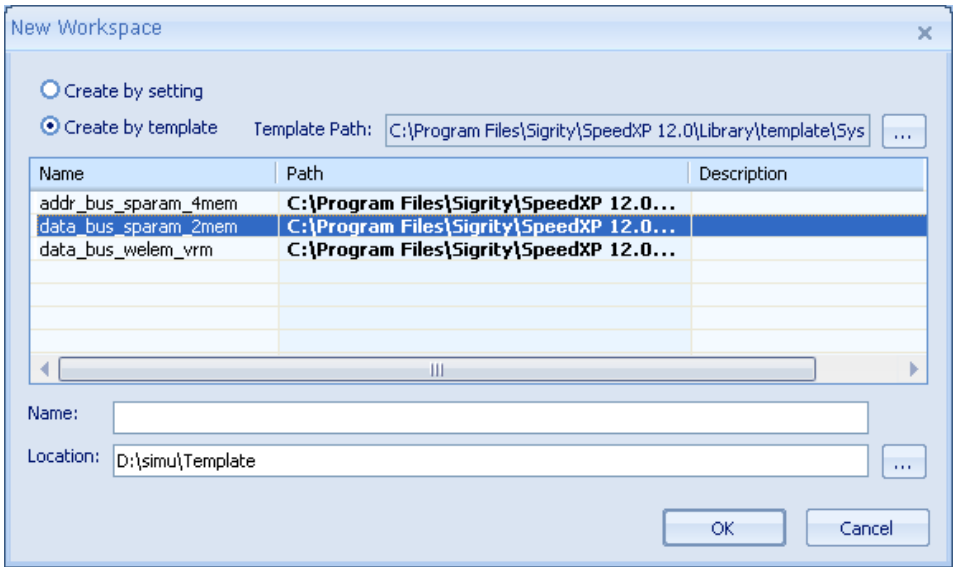
   Click the **New** button.

   The **Select Module** window opens.



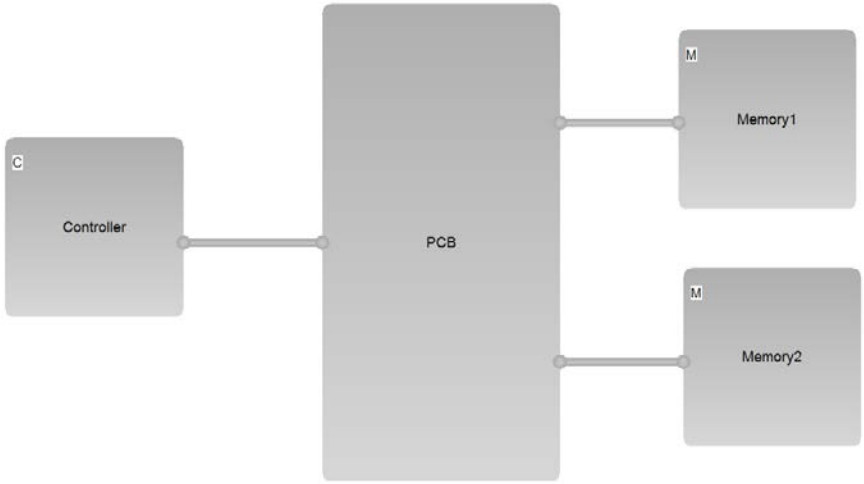
3. Select **Parallel Bus Analysis**, and click **OK**.

   The **New Workspace** window opens.
4. Select **Create by Template**.
5. Select **data_bus_sparam_2mem**.



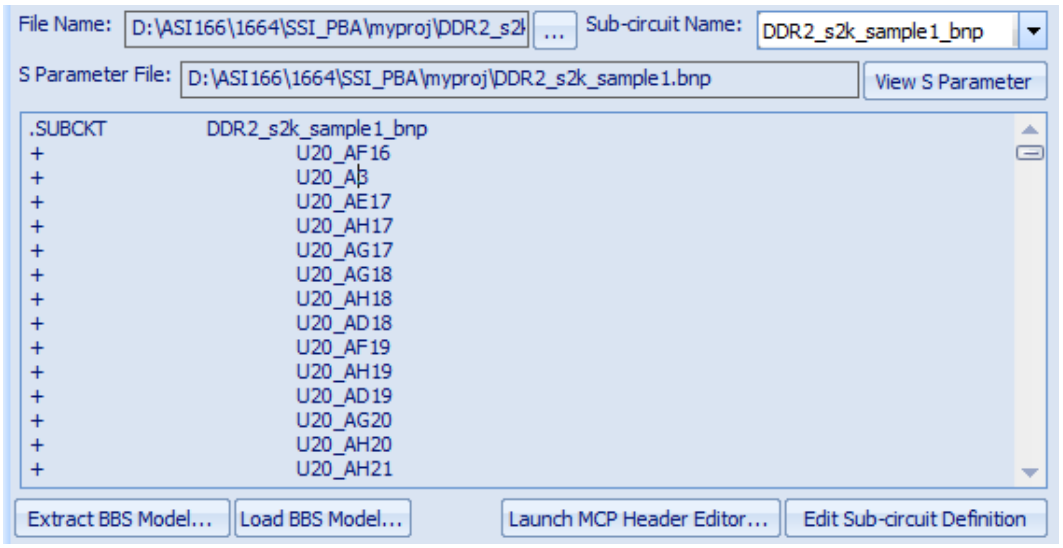
6. Enter a **Name** for the project.
   For this tutorial, specify name as `myproj`.
7. Enter or browse a **Location**.
8. Click **OK**.

   The Block diagram is created as follows.

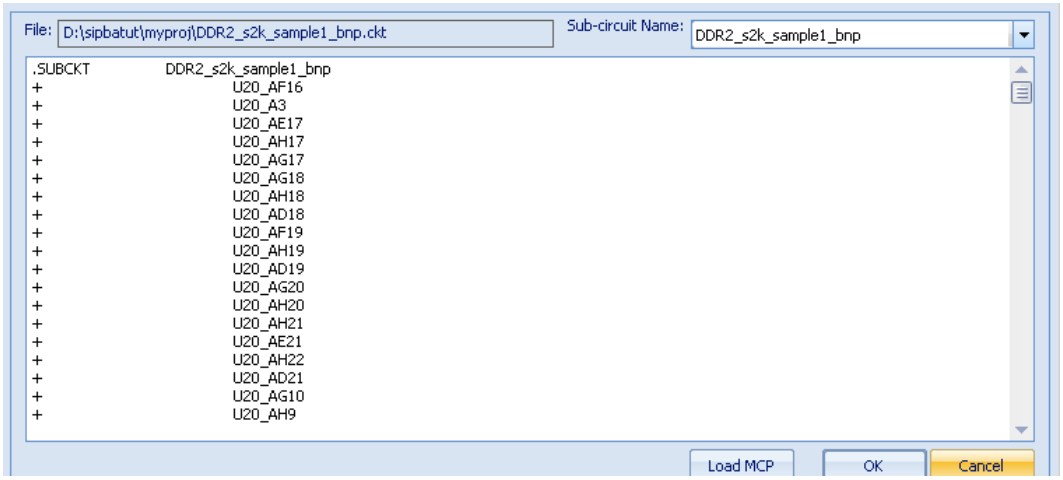


   This design differs from the W-element template in several ways. First, the PCB is defined by an S-parameter extraction completed using Sigrity PowerSI. Second, it has two memory devices, but no VRM defined.

9. Double-click the **PCB** block to open its **Property** dialog.



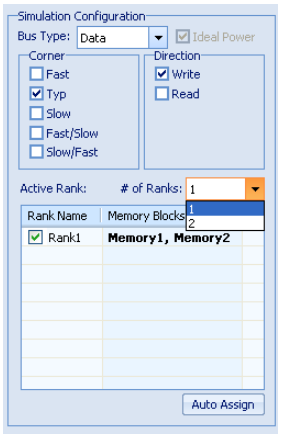
10. Click **Edit Sub-circuit Definition** to see the S-parameter circuit model netlist that is created by PowerSI for the board.



Continue to explore this Template by editing components and simulating, as was done in Chapter 1 for the W-element template.

In this case, the **Analysis Options…** window looks different from the former. It has a section that can be used to specify Ranks.

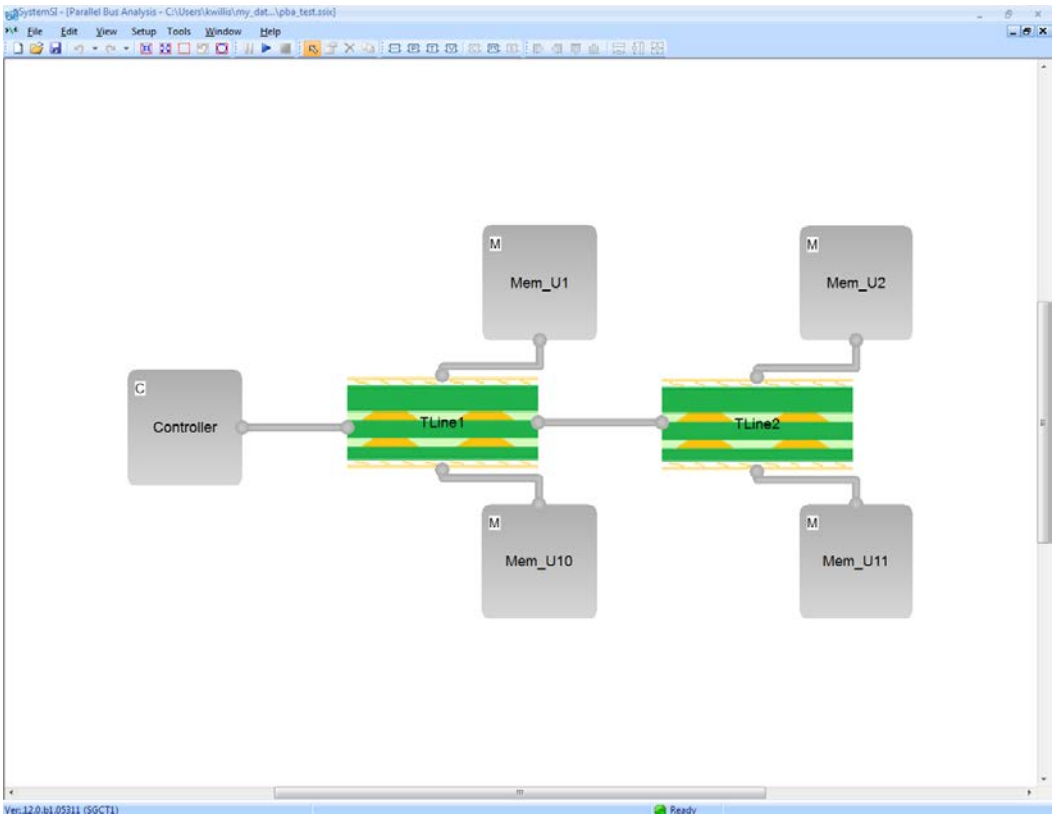
11. Select **1** in the **# of Ranks** drop-down list, and click in the check box before **Rank1** to enable it.



12. Save the settings and simulate the design.

4.2.1 Rank Definition

For Data bus topologies, memory blocks can be organized into “ranks”. A rank is simply a group of memory blocks that are active or inactive all together. Take the example below:

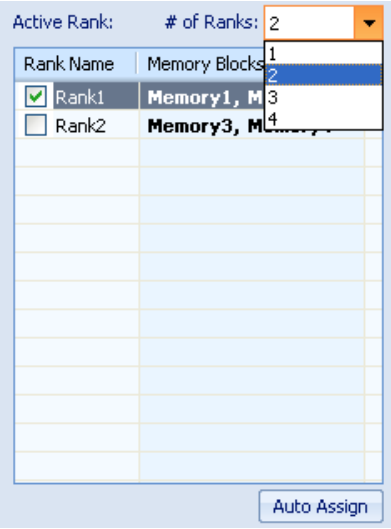


In this case, data signals DQ[7:0] from the controller are connected to Mem_U1 and Mem_U10, and DQ[15:8] are connected to Mem_U2 and Mem_U11. This allows Mem_U1 and Mem_U2 to be grouped as **Rank1**, and Mem_U10 and Mem_U11 to be grouped as **Rank2**. In the case of a Write simulation for this data bus, either **Rank1** or **Rank2** will be active (i.e. receiving on a Write), while the other rank will be placed in the Standby mode. Grouping memory blocks into ranks enables more efficient simulations to be run, eliminating unnecessary combinations of active/inactive memory blocks, and reducing the number of overall simulations to be run.

Active Rank allows a single Data Bus simulation to contain multiple active memory blocks.

1. The memory blocks in an active rank will be active together.
2. If more than one ranks are checked as active, the tool will do multiple simulations for active rank sweeping: each simulation will only have one active rank while others will be considered as standby.

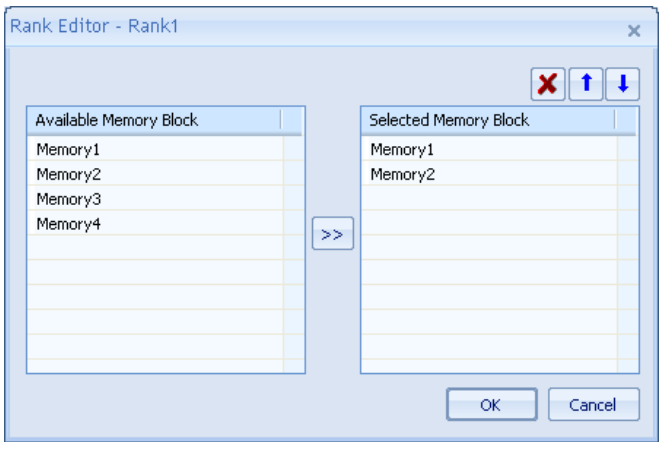
When a number is selected from the **# of Ranks** list, multiple ranks of the selected number will be added to the **Rank Name** column automatically. (The default drop-down list **# of Ranks** contains the max number which is the number of memory blocks.)



3. If the selected # is 1, **Rank1** will contain all memory blocks.
4. If the selected # is max, each rank will contain one memory.
5. If the selected # is neither 1 nor max, the ranks will be assigned randomly.

If the user clicks the **Auto Assign** button, the tool will automatically assign the memory blocks based on Frequency Response.

User can always manually assign the ranks in **Rank Editor** by double-clicking a specific rank.



You can:

6. Select a memory block in the **Available Memory Block** column and click  to add it to the **Selected Memory Block** column.
7. Use the  button to delete the selected memory block.
8. Use the Up arrow() or down arrow () button to move up or down the memory block.

NOTE!	The above instruction on the four-memory data bus is only an explanation of the function of **SystemSI – Parallel Bus Analysis**. This tutorial will not go into this exercise. The remainder of this Chapter and the subsequent sections of this Tutorial go into a new design that has similar components, including an S-parameter based PCB model.

4.3 Parallel Bus Analysis Wizard

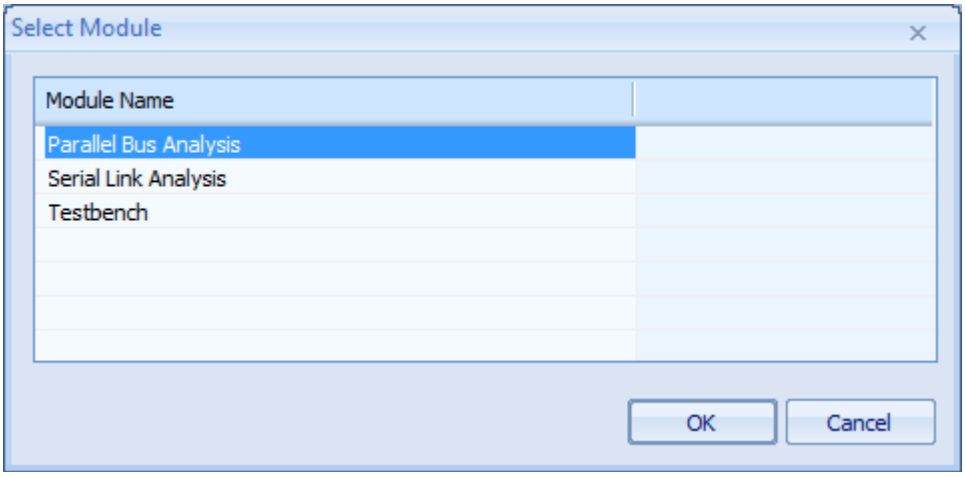
In this section, we'll create a new design using the Wizard. This design will be used for subsequent exercises through Chapter 9.

1. Launch SystemSI.
2. To open a new workspace, select **File > New**

   or

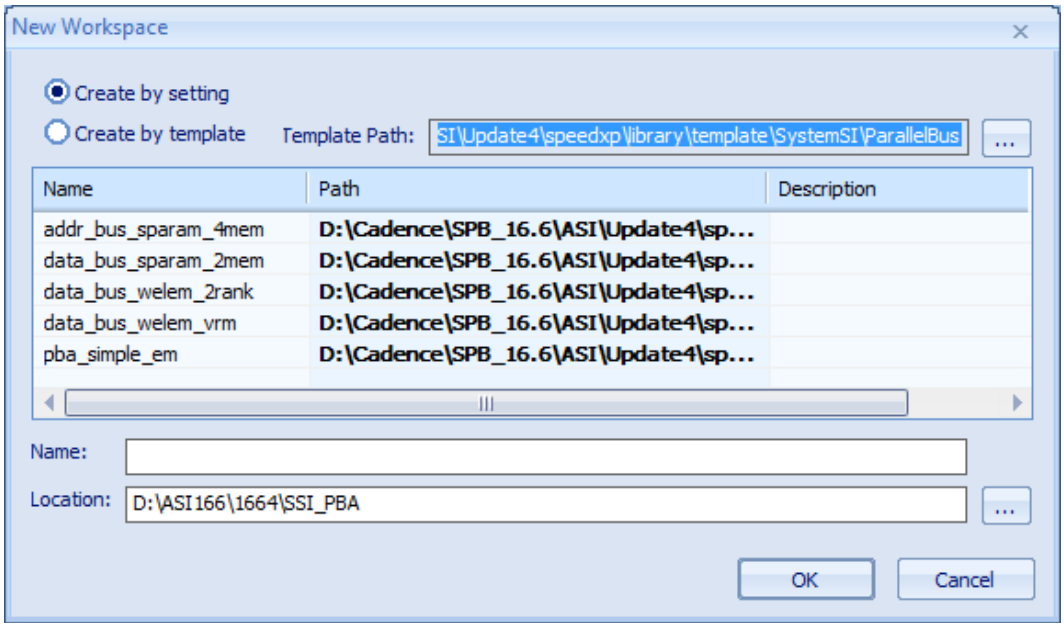
   Click the **New** button.

   The **Select Module** window opens.



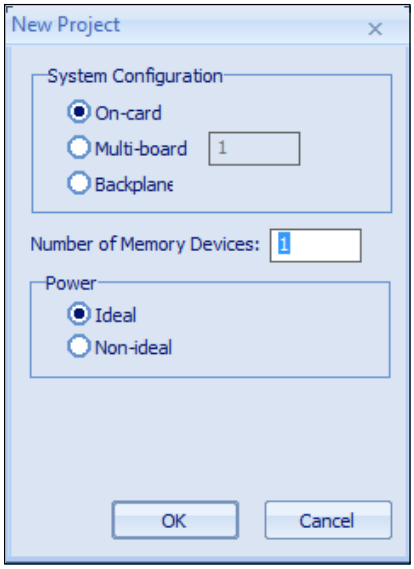
3. Select **Parallel Bus Analysis**.
4. Click **OK**.

   The **New Workspace** window opens.
5. Select **Create by setting**.



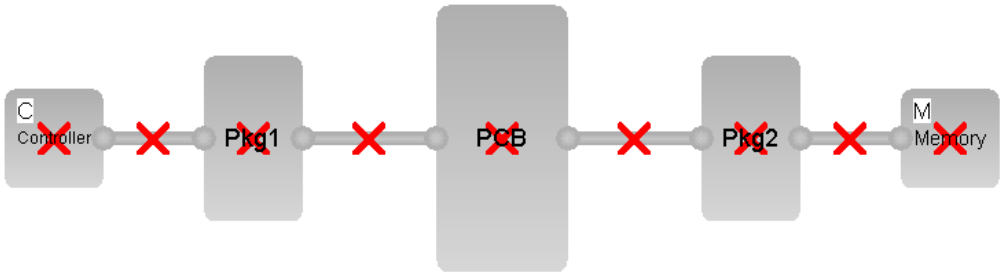
6. Enter a **Name** for the project.
7. Enter or browse a **Location**.
8. Click **OK**.

   The New Project dialog box displays.



9. For this tutorial, use the default selections in the **New Project** dialog, and click **OK**.

   Block diagram is created. Red cross (**X**) on each component block indicates that none of the components have models defined at this point. As a result, "**Connections**" are also a**ll undefined**, as indicated by t**he** red "**X**".

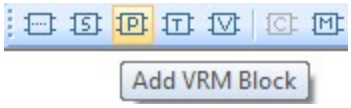


10. Delete the **Pkg1** and **Pkg2** blocks.

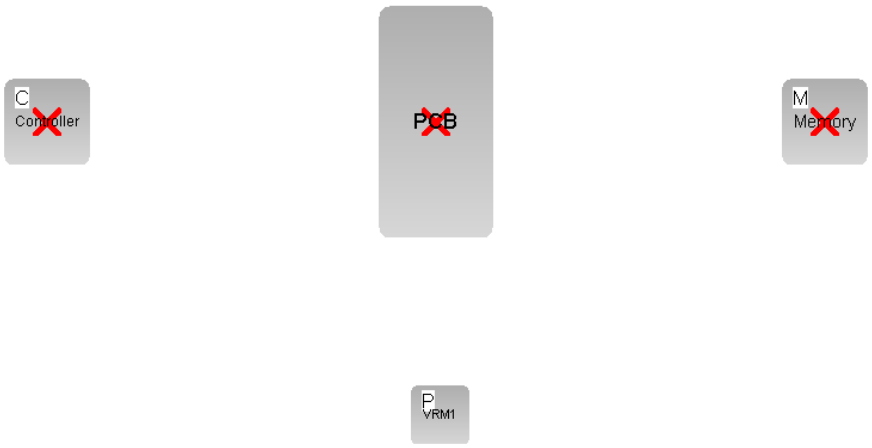
    To delete the blocks, select the blocks and click the Delete button.

    Note that along with the Pkg. blocks, the connections are also deleted.

11. Add a **VRM** Component by clicking the "P" icon in the toolbar.

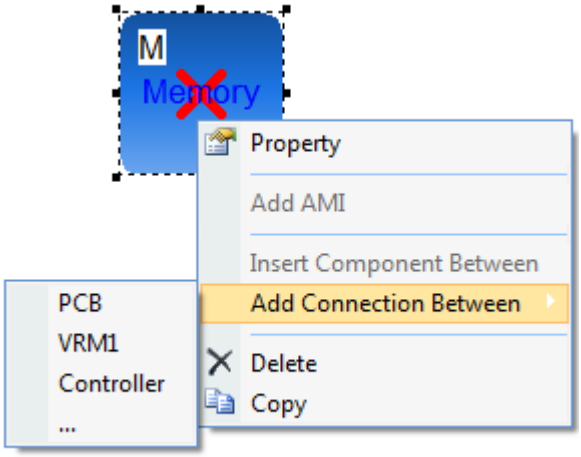


12. Place it below the PCB.

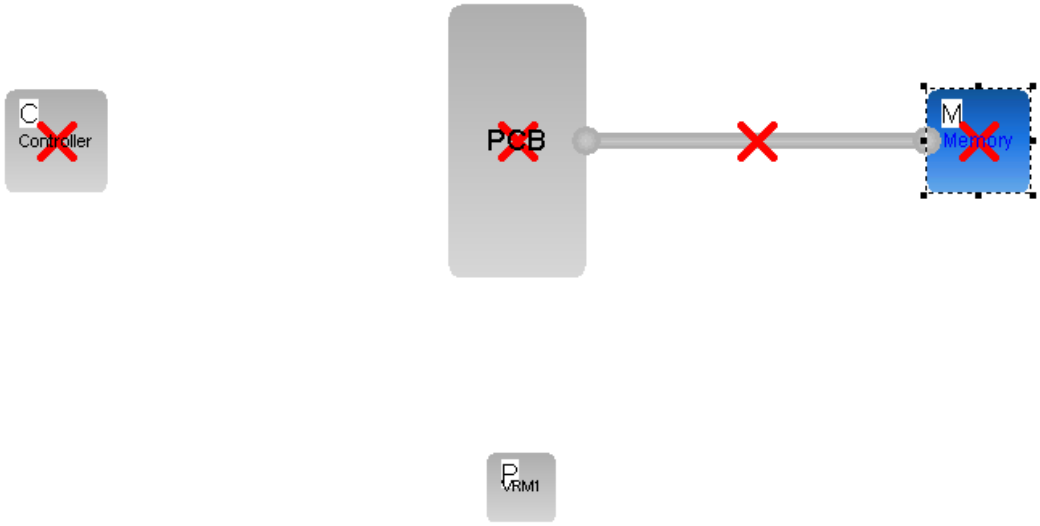


13. Add Connections between each component and the PCB.

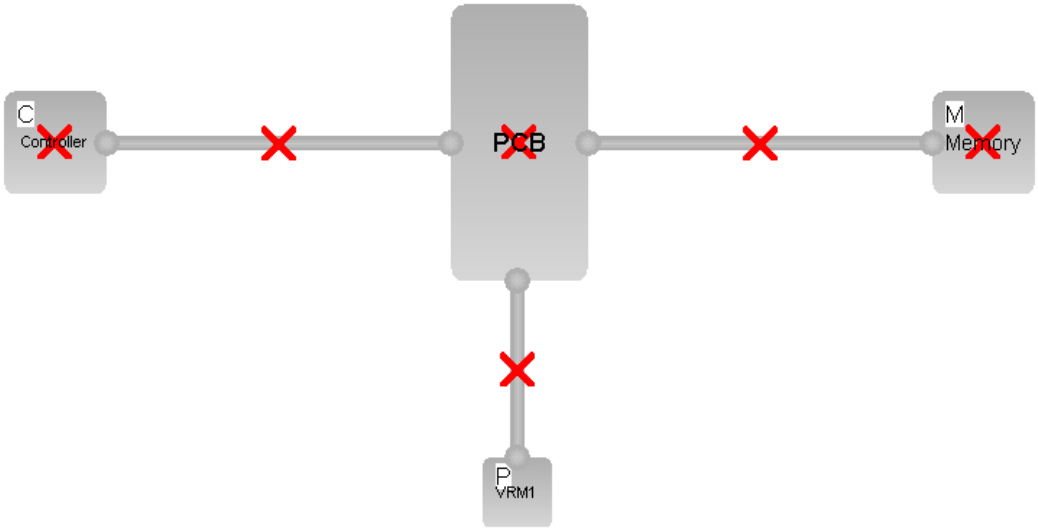
    a) Right-click on the **Memor**y component, and from the popup menu, select *Add Connection Between -- PCB*.



The Connection is made with a Red "X" as shown. The Red Cross (X) on the connection indicates that the connection is not yet complete as the ground signals are not connected.



14. Similarly, connect **VRM and PCB** blocks, and **Controller and PCB** blocks.
    The topology will look like this:



15. Save this design.

Continue to ***Chapter 5*** to specify the IBIS models for Controller and Memory components.

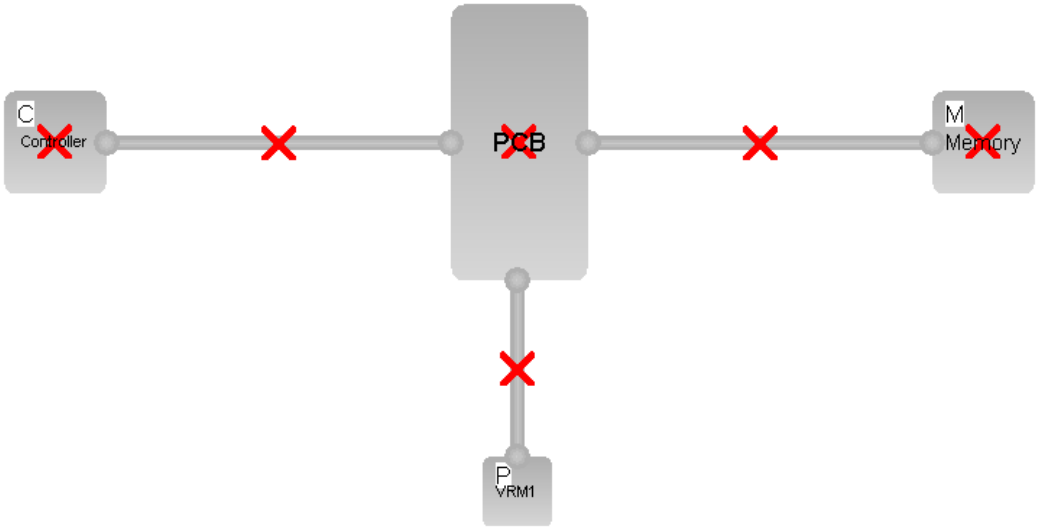
5 Assigning Device Models

5.1 Overview

This chapter demonstrates the procedure for assigning IBIS models to memory controller and memory devices. SystemSI uses IBIS models, and allows both behavioral and transistor-level IO models for these devices. As discussed in Chapter 3, Parallel Bus Analysis adds certain extensions, including keywords, to these IBIS files. And as we will see, Parallel Bus Analysis automates connection of these IBIS files with other components in the system.

5.2 Design Files

The design created using the Design Wizard in ***Chapter 4.3*** is the starting point for this exercise.



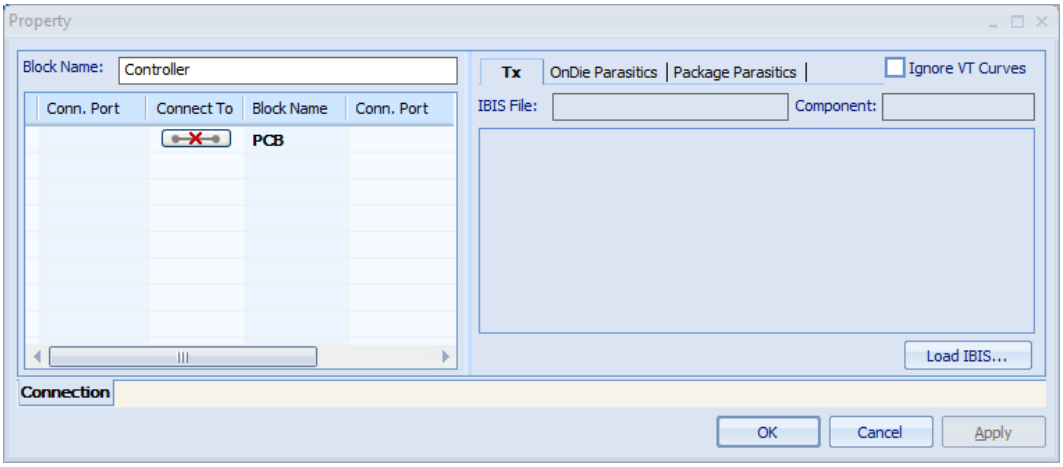
If you have performed the tasks covered in Chapter 4, you can continue with same design files. However, if you are starting from Chapter 5, open the `chap5.ssix` file located at *`<INSTALL_DIR>\SpeedXP\Samples\SystemSI\Parallel Bus Analysis\Tutorial\chap5`*. This directory also contains the IBIS model file for the Controller and the Memory blocks.

5.3 Controller Model

5.3.1 Assign the Controller IBIS Model

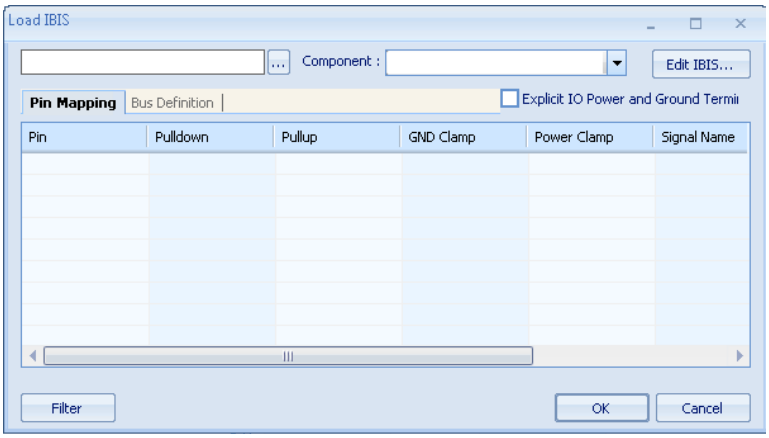
In this section, we will assign the Controller IBIS file to the Controller block.

1. Double-click the Controller block in the schematic, to open the Property dialog box.



2. Click **Load IBIS…**.

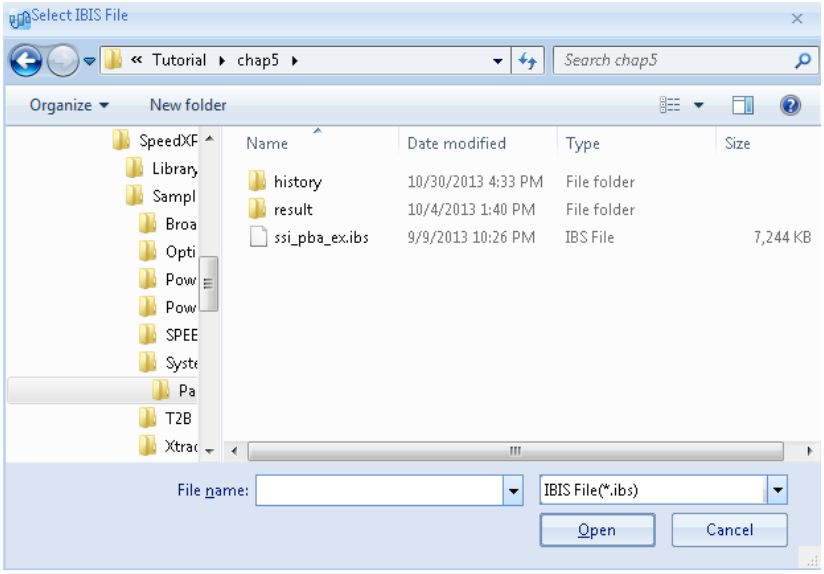
   The Load IBIS dialog box displays.



3. Select the IBIS file by clicking the Browse button ().

   For this tutorial, the IBIS file is available in the `chap5` folder in the `Tutorial` directory.

4. Select the `ssi_pba_ex.ibs` file, and click Open.



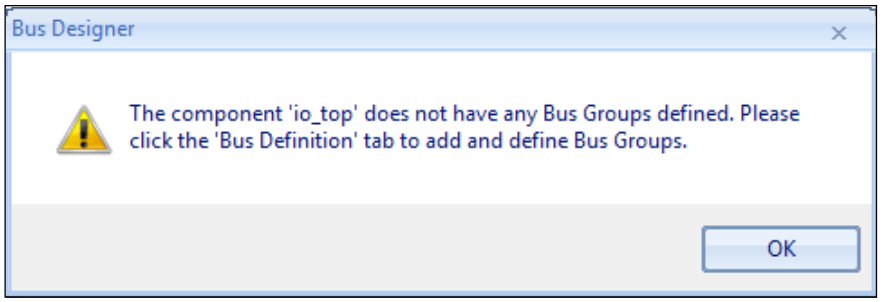
The IBIS model is opened with the **Pin Mapping** tab selected.

5. From the **Component** drop-down list, select **Controller**.

   For this example, the IBIS model has the Pin Mapping from the device manufacturer. However, because the Bus Definition is not yet complete, the Pin Mapping is not fully defined, with only 1 bus group shown.

NOTE!	For more information about the Pin Mapping, Bus Definitions, and other aspects of IBIS models that are needed for Parallel Bus Analysis, see ***Chapter 1*** of this Tutorial.

If you do not have a bus defined, following message is displayed.

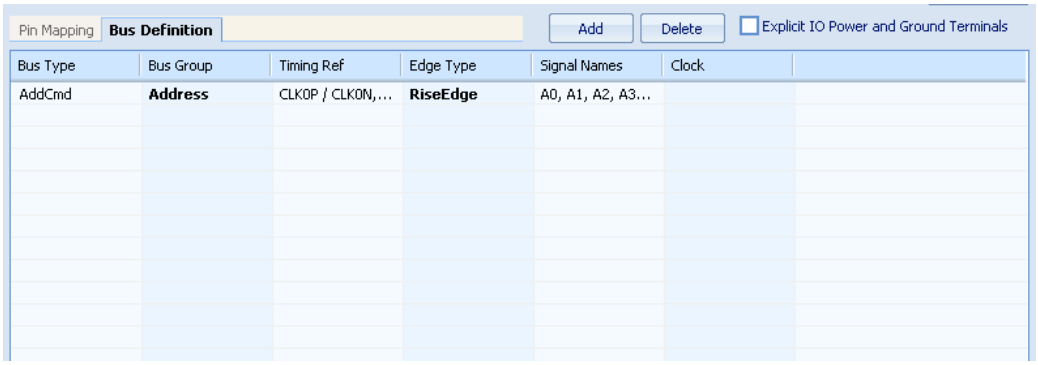


In the next section, we will create the bus definition for the Controller block.

5.3.2 Controller Bus Definition

In this section, we will define two Data buses, made up of some of the DQ signals in the Controller.

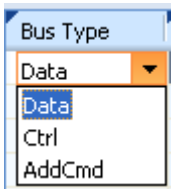
1. Click the **Bus Definition** tab for the IBIS Controller.



2. Click **Add** to create a new bus group.

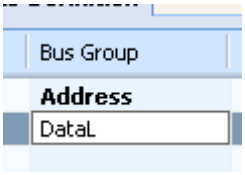
   A new Bus with default Bus Type **Data** and Edge Type **BothEdges** is created. We will define each of the fields that define this bus in the dialog box.

3. Click the **Data** Bus Type, to get the drop-down selection showing the other types of buses, **AddCmd** and **Ctrl**.



4. Leave the Bus Type as **Data**. We will define two Data Buses for this system. Additional buses are present in the IBIS file, but only two will be defined for this Tutorial.

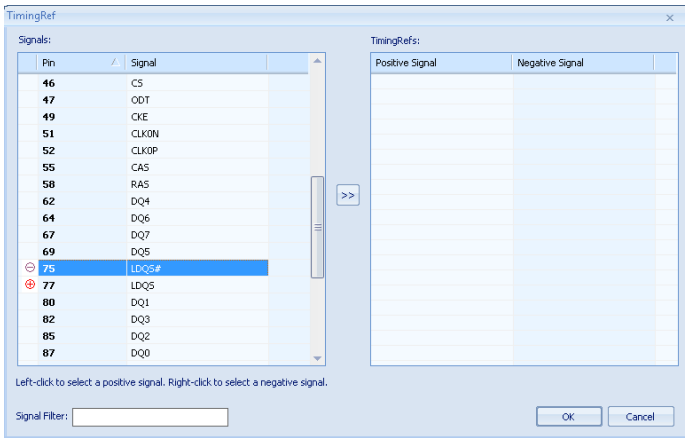
5. Enter **DataL** in the Bus Group by typing the name in the empty field box.



6. Next, define the Timing Reference. Click the **Timing Reference** field.

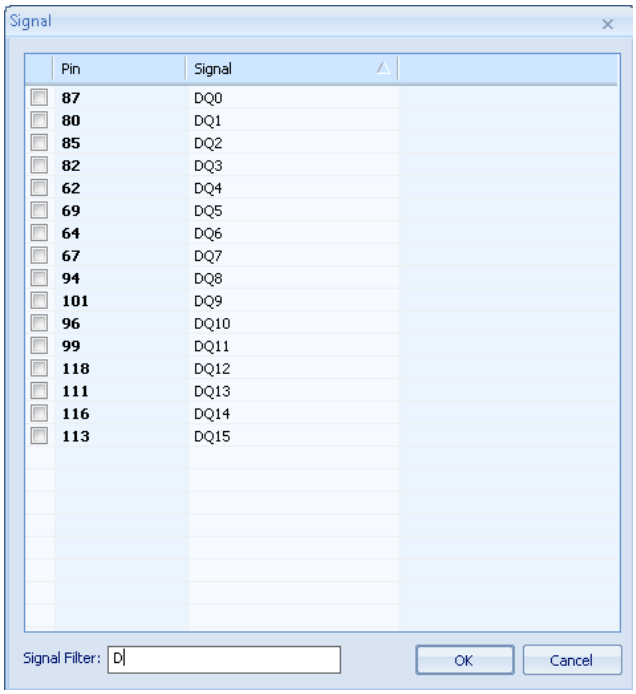
   The Timing Ref dialog box displays, listing all the signals available in the IBIS file.

   - Select the pin **77** connected to the **LDSQ** signal.
   - Right-click on the pin **75** connected to the **LDSQ#** signal.



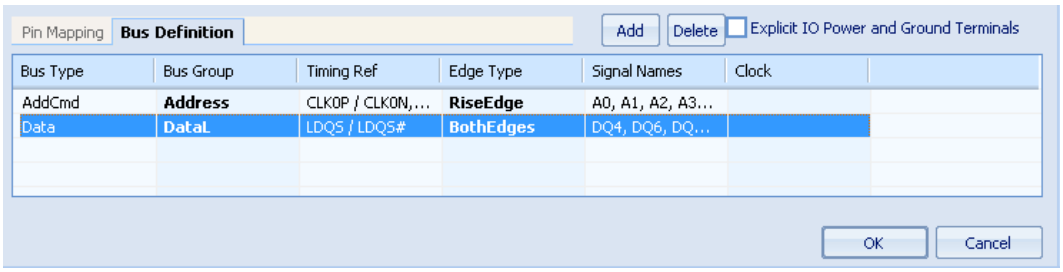
- Click the double arrow button  .
- Click OK.

7. Define the Signals that make up the bus. For the DataL bus, add first 8 DQ signals.
   - Click on the Signal Names grid.
   - In the Signal dialog box, to sort the list by signal names, click the **Signal** field.
   - From the dialog box, select the **DQ0… DQ7** signals.
     You can use the Signal Filter text box to filter the DQ signal.



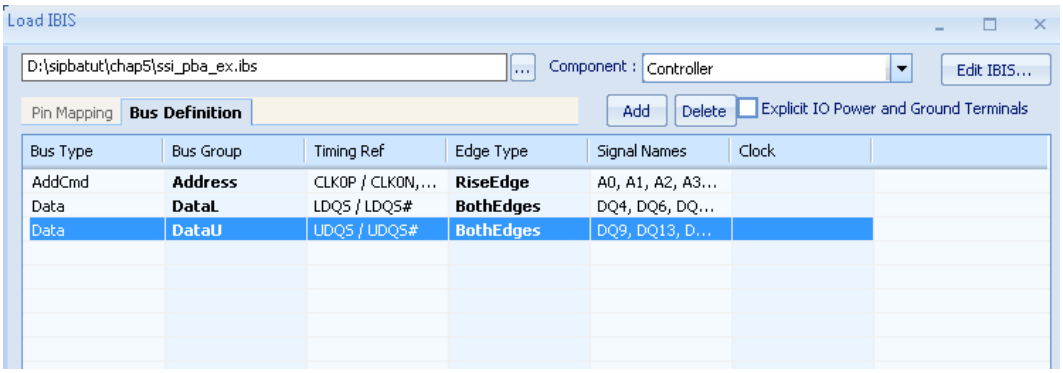
   - Select the check box next to the selected signals, and click OK.

     The **DataL** bus is now setup as shown in the following figure.



8. Repeat Steps 2 through 8 above, to create the second Data Bus with following specifications.
   - Name Group "**DataU**"
   - EdgeType "**BothEdges**"
   - Timing Ref Signals – "**UDQS/UDQS#**"
   - Signal Names – "**DQ8** thru **DQ15**"

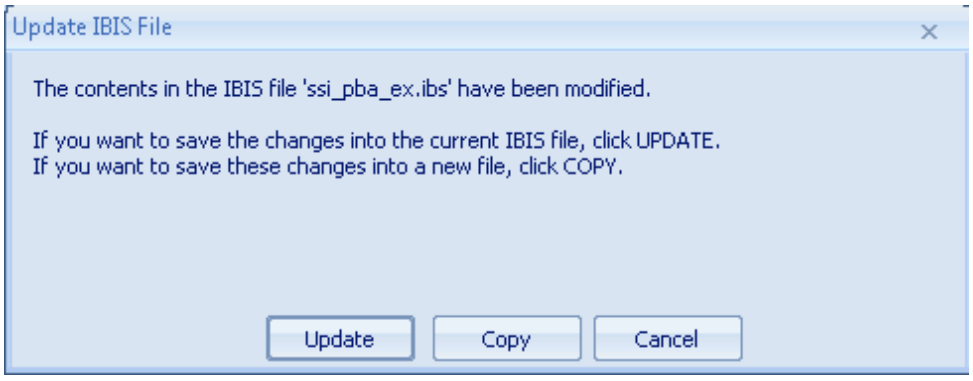
   Following figure shows the **Bus Definition** window after you have successfully created the DataU bus group.



NOTE!	If required, you can specify the Clock signal for the Data bus.

9. Click **OK** to close the Load IBIS dialog box.

   The **Update IBIS File** dialog appears.

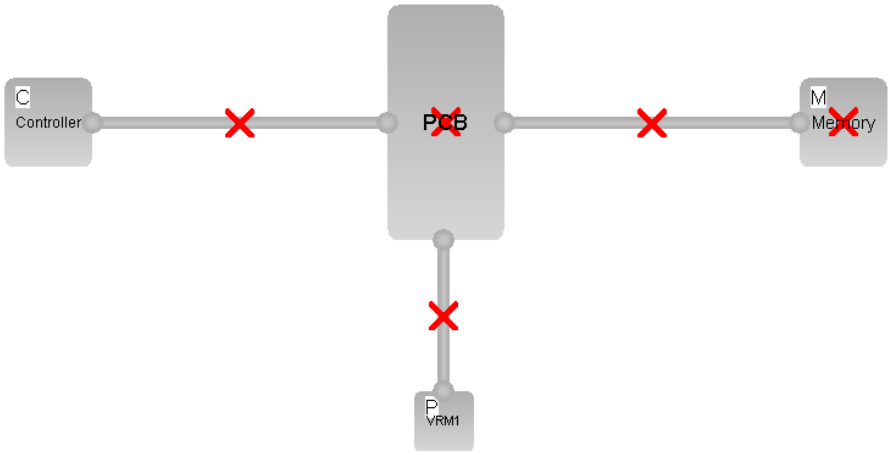


10. Click **Update** to save changes to the IBIS file.

    The Controller IBIS file is now modified.

6. Click Apply to save changes made in the Property dialog box.

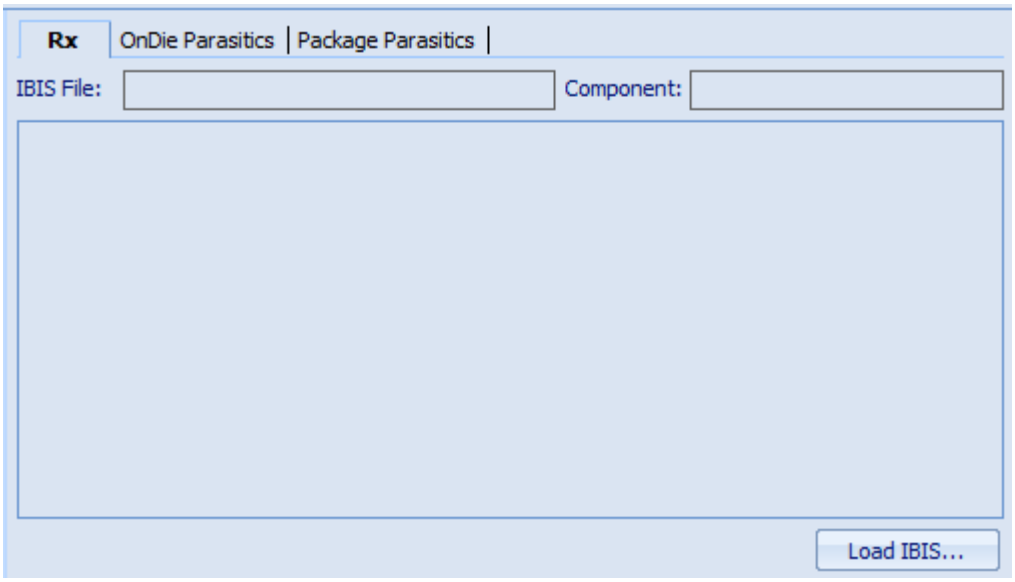
The Red cross on the Controller block is removed, indicating that an IBIS model is attached to the Controller Block.



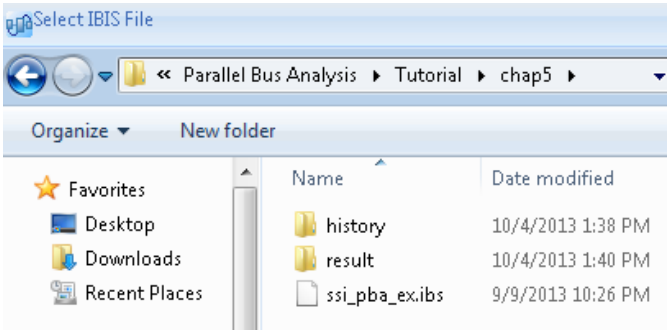
5.4 Adding Memory Device Models

The procedure for assigning IBIS models to memory devices is similar to that for the controller. In this exercise, the Memory component IBIS file has already been defined with correct Pin Mapping. The Bus Definition has not been created, and will be defined in this section of the tutorial.

1. Double-click the Memory block.
2. Click **Load IBIS…** in the dialog box that opens.



3. In the Load IBIS dialog box, select IBIS file by clicking  .
4. In the File Manger, navigate to the Tutorial directory, and double-click the ***ssi_pba_ex.ibs*** file.



5. From the Component drop-down list, select Memory.

   The contents of the file are loaded in the dialog box.

   By default, the Pin Mapping information is displayed for the Memory component
6. Select the **Bus Definition** tab, to view the bus definitions available.

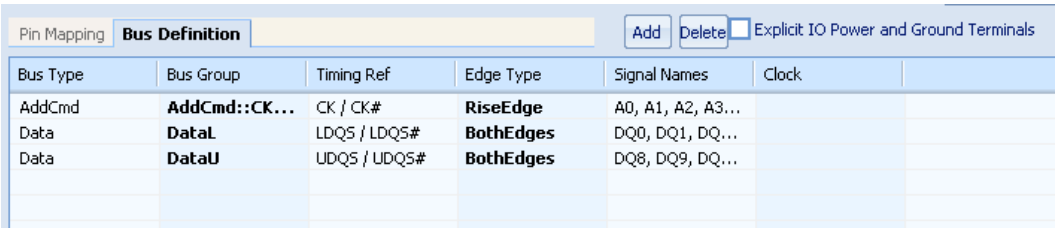
   Two buses have been added to match the Controller Data buses defined in the previous section. The Bus details are:

   - First Bus:
     - Name Group "**DataL**"
     - EdgeType "**BothEdges**"

- Timing Ref Signals – "**LDQS/LDQS#**"
- Signal Names – "**DQ0 thru DQ7**"

- Second Bus
  - Name Group "DataU"
  - EdgeType "BothEdges"
  - Timing Ref Signals – "UDQS/UDQS#"
  - Signal Names – "DQ8 thru DQ15"

The following figure shows the Bus Definition window.



7. Click **OK** to save your changes and to close the Load IBIS dialog box.
8. To Update the IBIS file, click **Update**.

NOTE!	Bus names and Signal names for Controller and Memory are local to the IBIS file, and do not have to match each other, nor do they have to match the names in other components, including the Controller. Pin names, however, are used to automate connectivity between components that have a physical connection. The signal/pin mapping is addressed in the block connections, which will be covered in ***Chapter 8*** on MCP connectivity.

We have now completed the specification of the Controller and Memory devices. In the next chapter, we will consider how to specify the PCB component.

6 Interconnect Models

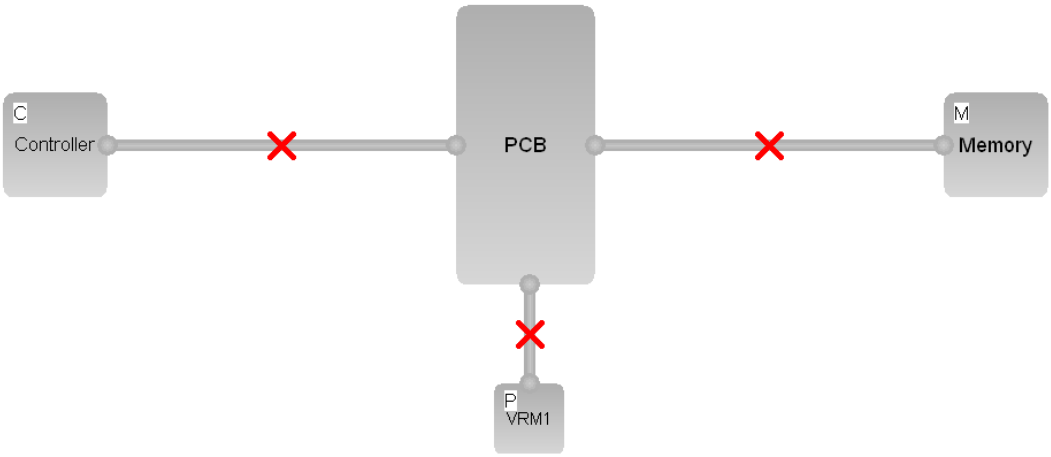
6.1 Introduction

Interconnect components including Printed Circuit Boards (PCBs) and packages, are defined in SPICE netlist definitions. These may be circuit models including lumped elements and distributed transmission lines, W-element models, or S-parameter models in either Touchstone or Sigrity BNP format. Models can be created within Parallel Bus Analysis using an integrated layout extraction utility that can use a variety of extraction engines. Sigrity PowerSI board and package modeling tool is integrated with Parallel Bus Analysis to allow easy extraction of full boards and packages, including non-ideal power and ground nets.

In this chapter, we will see how to assign a CKT file describing a PCB, to the block diagram, and will view the contents of the SPICE file. We will also go through a sample extraction of an S-parameter description of a board using Sigrity PowerSI, and see how the extracted model file is integrated with Parallel Bus Analysis.

6.2 Design Files

The design is the same one that was edited in Chapter 5. The block diagram has models defined for the Controller and Memory blocks. In this chapter, we will extract and assign a model to the PCB block.



If you have performed the tasks covered in Chapter 5, you can continue with same design files. However, if you are starting from Chapter 6, open the `chap6.ssix` file located at *`<INSTALL_DIR>\SpeedXP\Samples\SystemSI\Parallel Bus Analysis\Tutorial\chap6`*. This directory also contains the IBIS model file for the Controller and Memory.

The Tutorial folder contains the PCB file in SPD format for the interconnect model extraction.

6.3 PCB Model Generation

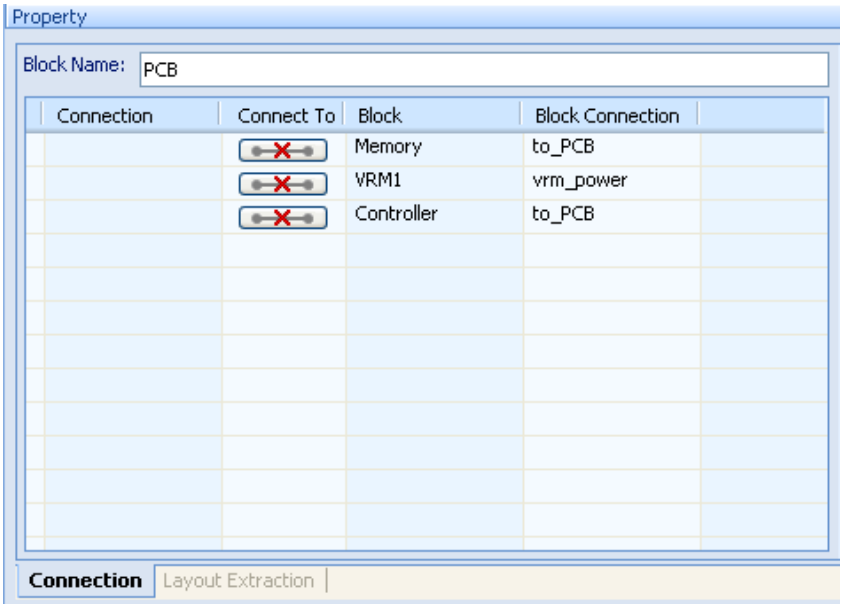
In this section, we will create a model for a DDR3 PCB bus using Sigrity PowerSI simulator. The design file for the PCB bus will be opened and reviewed in PowerSI, and a simulation will be performed. The result of the PowerSI simulation is an S-parameter model, and a SPICE netlist that has the model connection information (MCP) embedded. This will be reviewed in Parallel Bus Analysis.

A simple package model, using a different SPICE interconnects type, will be called in Parallel Bus Analysis, and associated with the Pkg1 component.

NOTE!	PowerSI is a physical board and package simulator, that uses Sigrity patented solution technology to quickly and accurately model high speed interconnects. Both signal integrity and power integrity effects are accurately modeled on parallel bus topologies. For more information, see the PowerSI material available on www.cadence.com.

6.3.1 PCB Block Editing

Double-click the PCB block to view its properties. By default, the **Connection** tab is visible.

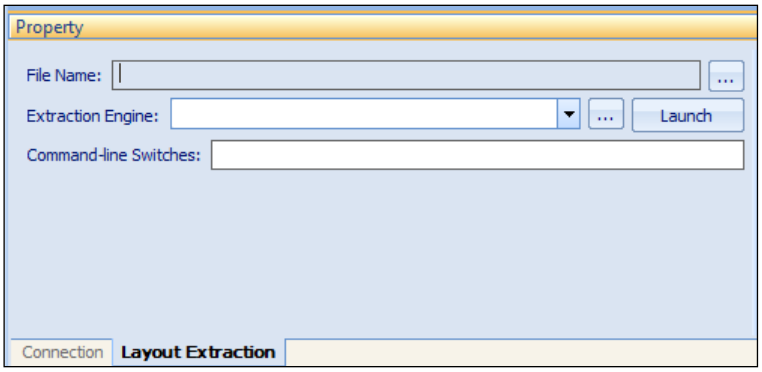


This tab shows information about the connections between the PCB block and other blocks. There are three connections, one each, to the Controller, Memory and VRM. None of the connections have been defined, so the invalid connection is indicated by the red X. Connection details will be covered in ***Chapter 8*** of this Tutorial.

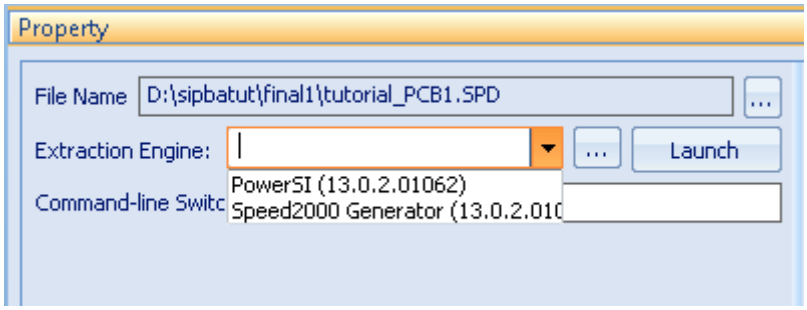
6.3.1.1 Layout Extraction Tab

The second tab in the Property window is the Layout Extraction tab. This tab will be used to perform simulation on the PCB model, using PowerSI. In this section, we will open the PCB design in Sigrity PowerSI modeling tool. We will then simulate the design to extract a bus model, and use the resulting S-parameter model for this PCB block.

1. Click the **Layout Extraction** tab.



2. Click the first  to pick the PCB design File.
3. Select file: ***Tutorial_PCB1.spd***.
4. Click the pull-down arrow next to the Extraction Engine to pick **PowerSI** as the Extraction Engine.



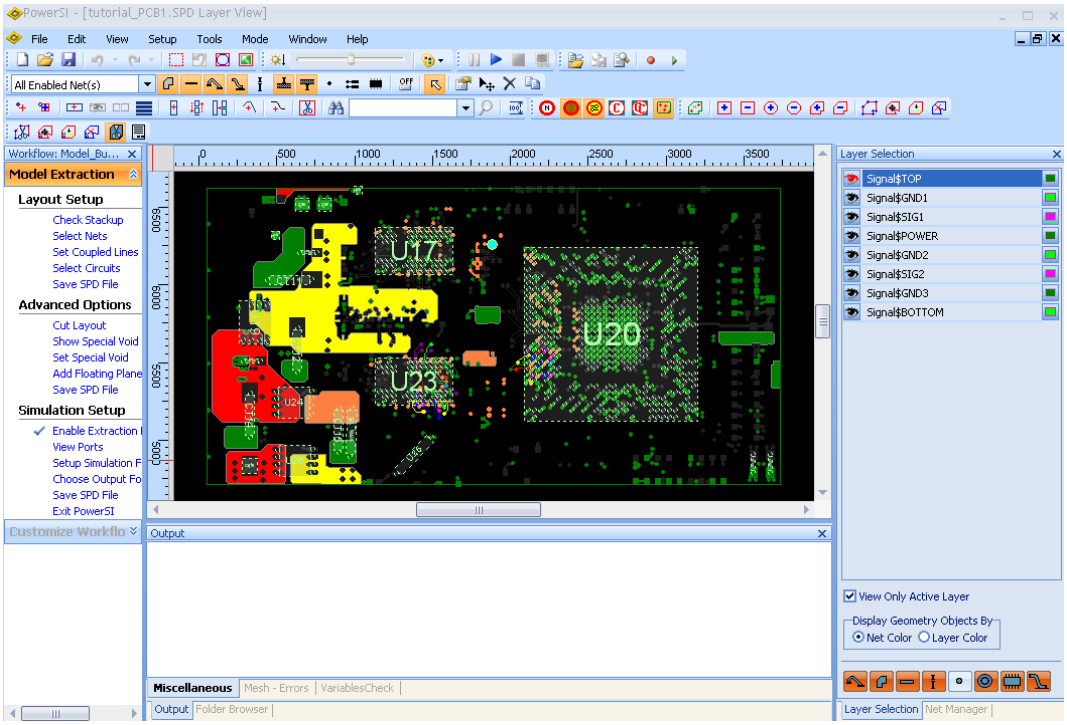
5. Click **Launch**.

Alternatively, you can launch PowerSI outside of SystemSI and complete the steps listed in the next section.

NOTE!	Other tools may be used for interconnect model extraction, but PowerSI is the preferred modeling tool. In general, any tool that can create SPICE compatible interconnect models, including S-parameters, could theoretically be used. The critical model connection is automated when using PowerSI.

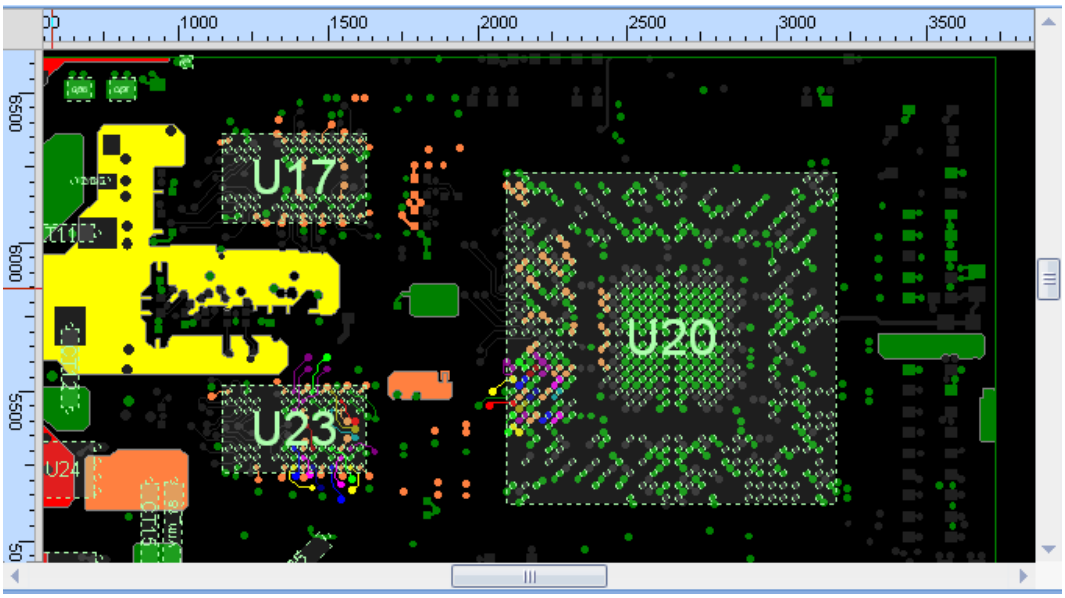
6.3.2 PowerSI Model Extraction

Power**SI** opens with the design, as shown in the following figure.



NOTE!	This design was edited, setup and preliminary simulations performed in PowerSI previously. We will review part of this setup, but details of the PowerSI simulation are beyond the scope of this document. Please contact Cadence support for more information about it, and PowerSI in general.

1. In the Toolbar in PowerSI, click the Select Circuits tool button () to display the circuits in the PCB design.
2. Now click  , to zoom in on part of the window.
3. Zoom in on the middle part of the PCB, with the largest components, U23 and U20.

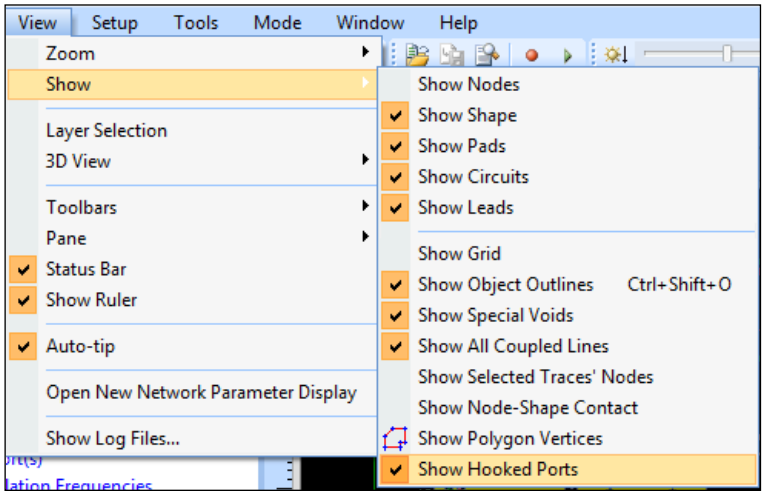


4. The DDR3 bus on this board connects the memory controller, **U20** with two DDR3 devices, **U17** and **U23**. For this tutorial, we will only simulate the Data Bus connecting **U20** and **U23**.

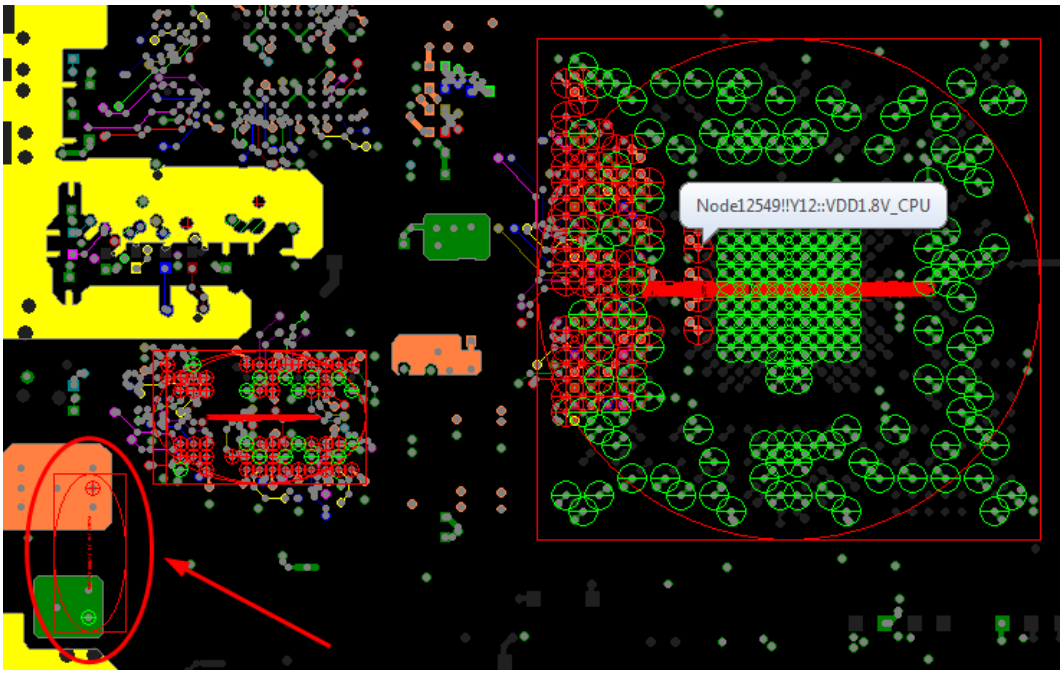
NOTE!	Only circuit connections that are part of the buses to be simulated should be part of the extraction. Additional circuit connections that are unterminated, at Controller and Memory components, will result in unconnected bus signals that cannot be resolved, resulting in an error at run-time in Parallel Bus Analysis.

The PowerSI simulation simulates this bus for its S-parameter performance over frequency.

5. To see the Port Locations for the S-parameters, from the **View** menu, choose **Show -- Show Hooked Ports**.



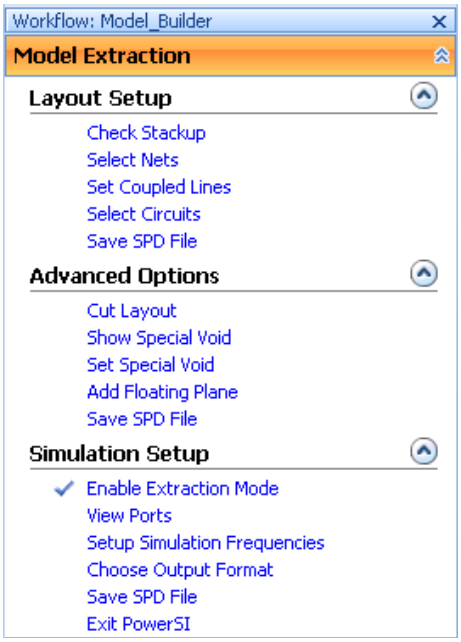
The ports on each of the data lines including strobe are shown on the controller (U20 – right side) and DRAM (U23 – middle) components. Positive port terminals are displayed as **Red** "+" symbols, and negative terminals are **Green** "-" symbols.



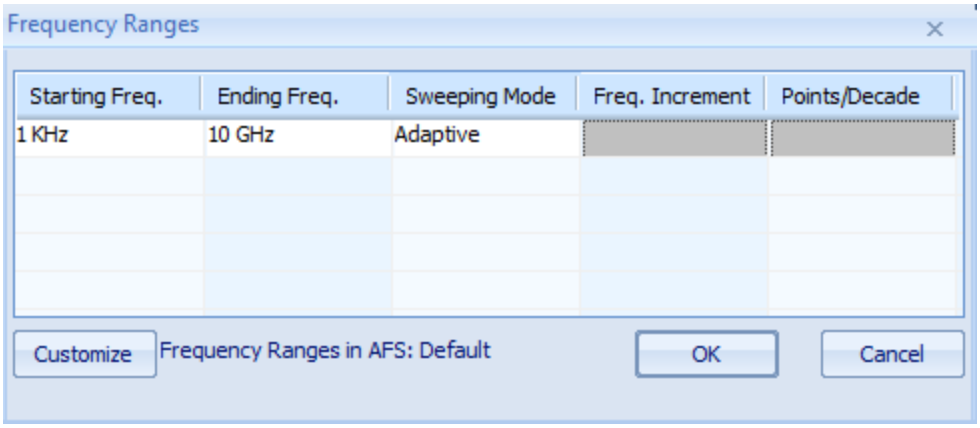
The port in the Bottom left side is where the 1.8 Volt DDR power supply for the PCB. This voltage regulator module (VRM) port is circled in Red.

There are a total of 39 ports defined on the PCB.

6. The Frequency range for the S-parameter simulation can be viewed by clicking the **Setup Simulation Frequencies** command on the left side Workflow Manager:



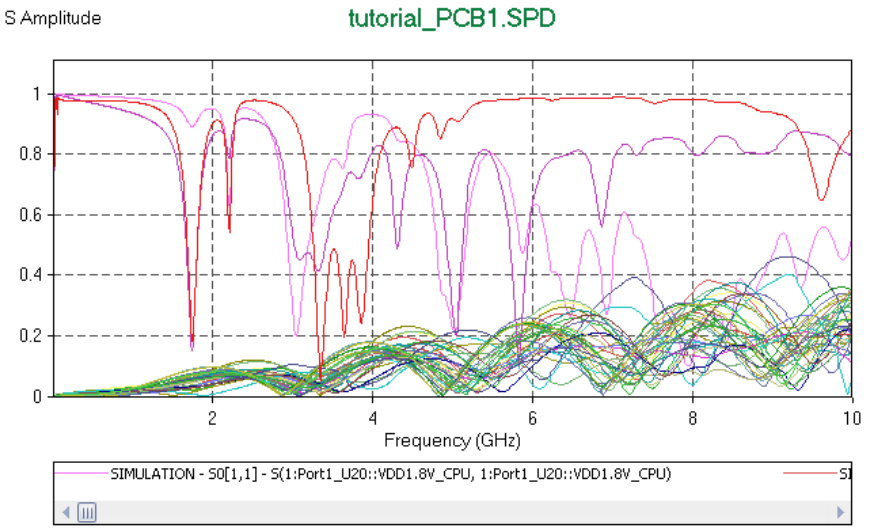
The simulation frequencies should be set as shown below, with a starting frequency of 1 KHz, Ending frequency of 10 GHz and Adaptive frequency sweeping.



7. To Start the PowerSI simulation, click the Play icon .

NOTE!	The PowerSI simulation takes approximately 10 minutes to complete on a dual processor machine.

Upon completion, the S-parameters are automatically plotted in a Network parameter display window.



The results of the PowerSI simulation are automatically saved.

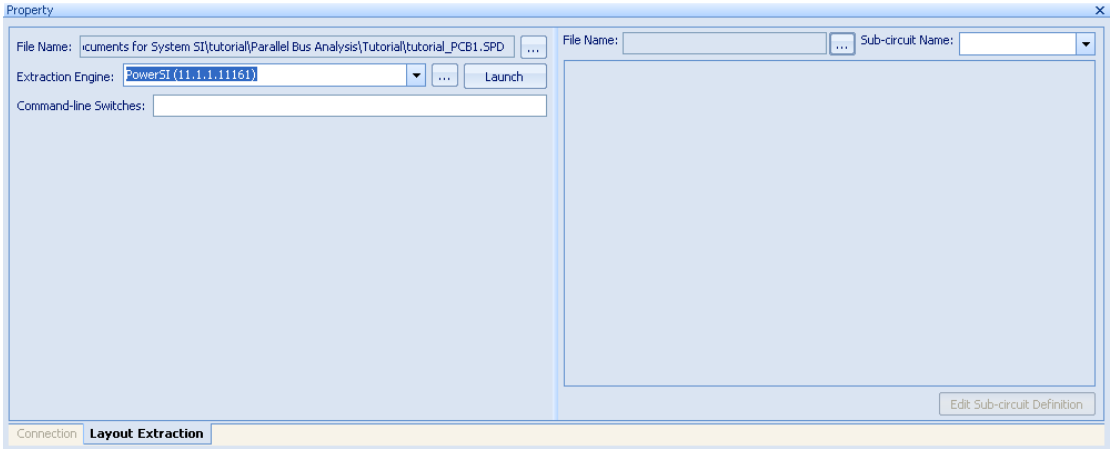
8. To close PowerSI, select *File -- Exit*.

6.3.3 Assign PowerSI Model

Two files are created by PowerSI, a 39-port BNP file that has the S-parameter data in Sigrity compact binary format, and a CKT file with the SPICE netlist that calls this S-parameter data model, and has the connection syntax (MCP) defined.

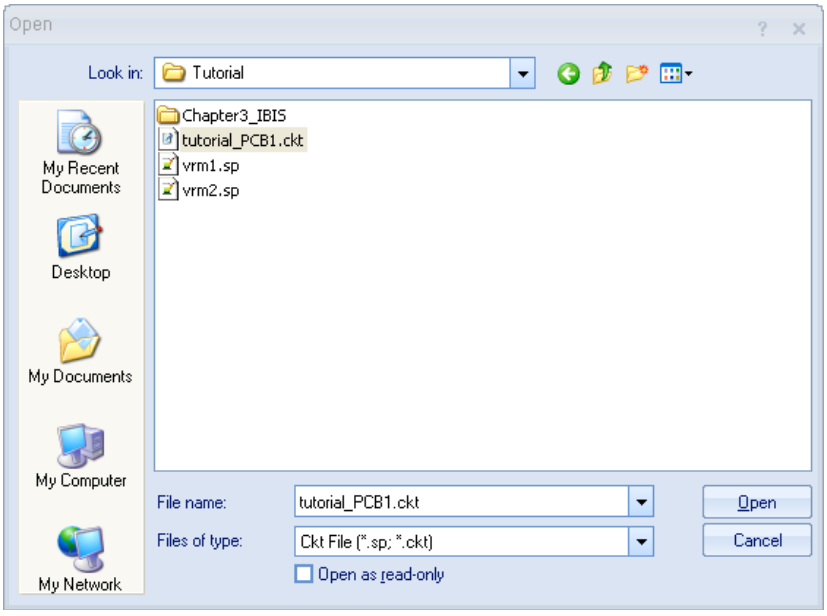
With the PCB model created, we can assign the correct data to the block diagram component in Parallel Bus Analysis.

1. In the **Property** dialog for the PCB block, on the right side, click the **File Name** icon.



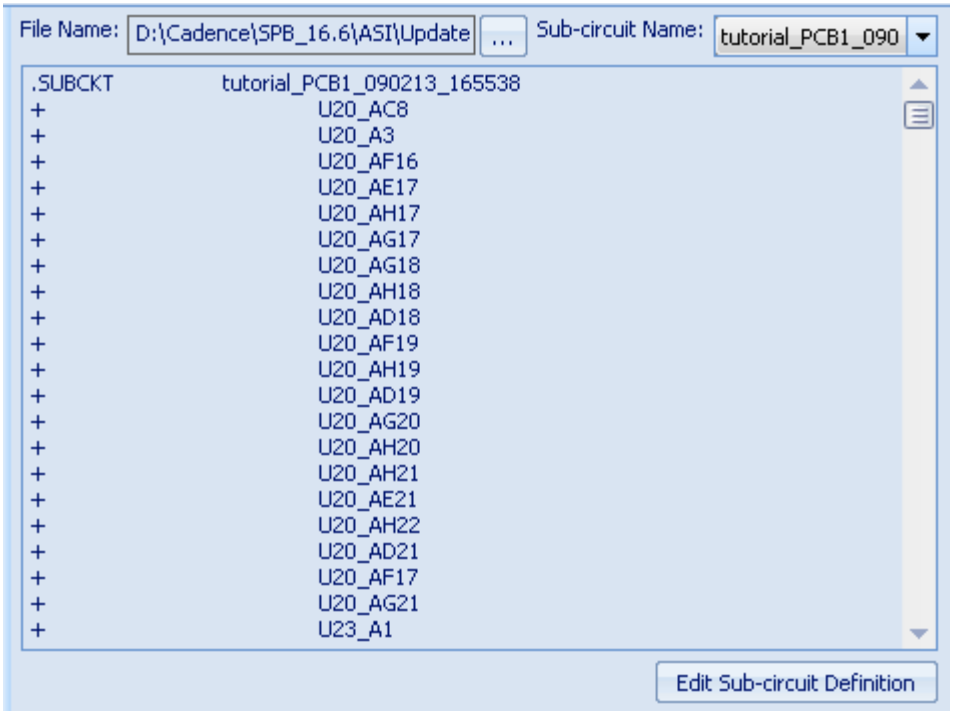
2. Find the correct CKT file, in the Tutorial directory. The file name starts with the name of the SPD design file, and has date appended to it in mmddyy format.

   Note: *If desired, you can use the already generated CKT file,* ***tutorial_PCB1.ckt****, available in the Tutorial directory.*

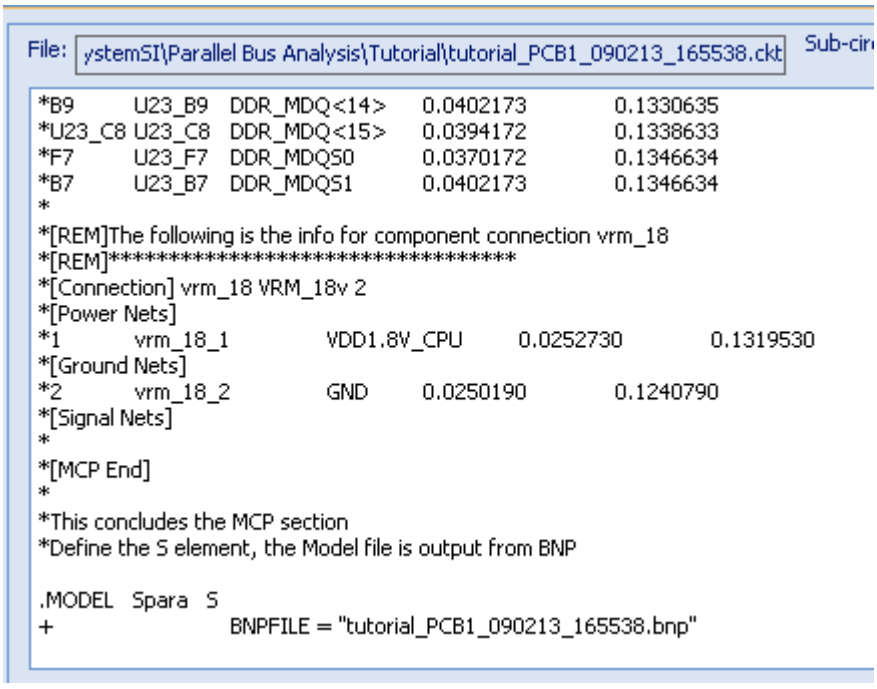


3. Open this file, and the circuit netlist is shown in Parallel Bus Analysis.

   This netlist can be edited.



For example, changing the frequency range or other settings in the PowerSI simulation will result in new S-parameters. A new S-parameter file could be used by editing the BNPFILE line in the CKT file.



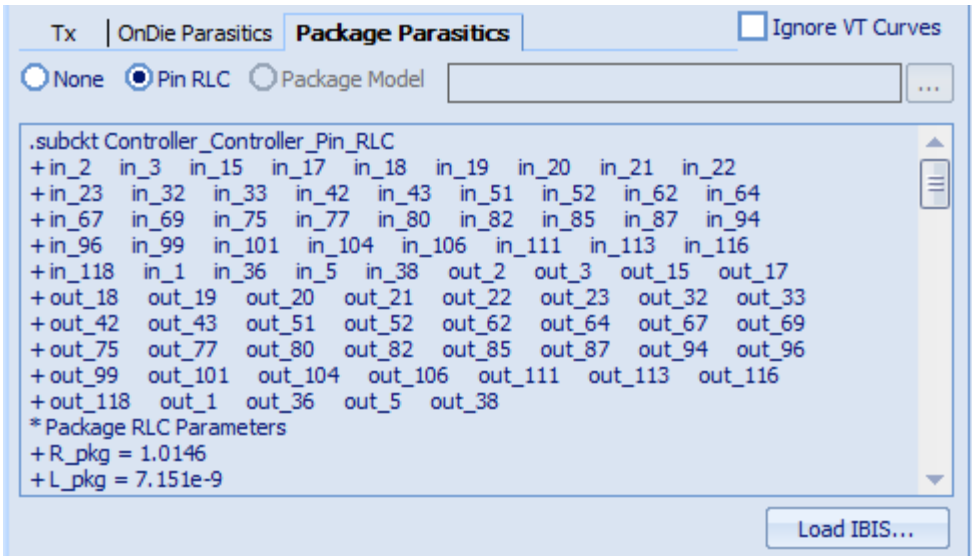
NOTE!	The SPICE file created using the Layout Extraction will usually not require editing. In particular, the pin-naming and MCP sections should not be changed by the user. The MODEL section, including S-parameter filename, can be updated, as long as key aspects of the PowerSI model are not changed. This includes the topology, port location, numbering and naming.

6.4 Package Models

Packages can be modeled in the same way as PCB interconnects, with dedicated extraction using external tools like PowerSI. However, Parallel Bus Analysis also has the flexibility to use the package parasitics defined in IBIS files. For this Tutorial example, we will turn on the RLC parasitic models in the Controller and Memory components.

6.4.1 Controller and Memory Package Model

1. Select the Controller component in the block diagram and open the property dialog box for Controller.
2. Select the **Package Parasitics** tab.
3. Select **Pin RLC.**



4. Click OK.
5. Repeat the steps for the Memory component.

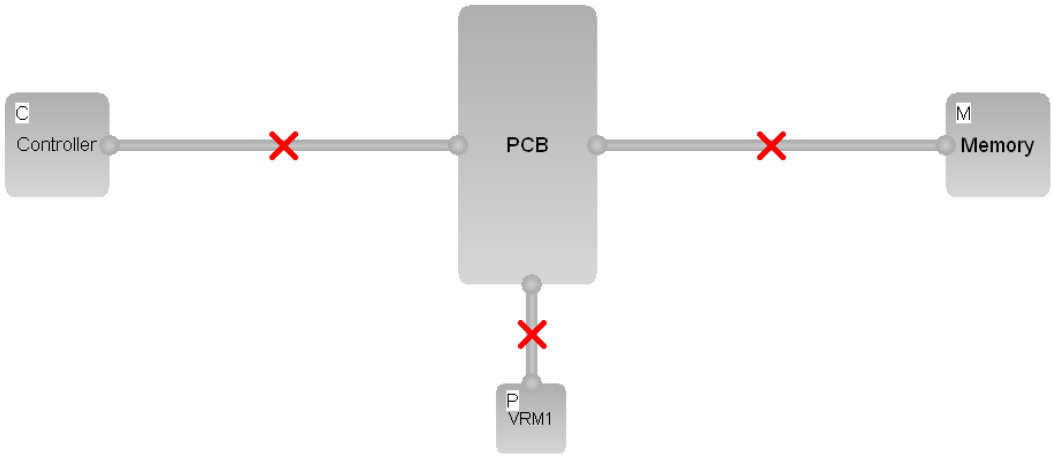
7 Adding a VRM

7.1 Overview

Parallel Bus Analysis allows non-ideal power simulation, including all aspects of power distribution network (PDN) modeling. This includes real power and ground planes in the PCB and package interconnects, as well as the I/O and VRM models. In this section, we will connect a VRM to the PCB and define a simple model for it.

NOTE!	When not-ideal power simulations are performed, on-die parasitics should be included in the Controller and Memory component models, to include the effects of any on-die decoupling capacitance. These can be defined in the **OnDie Parasitics** tab when you double-click a Controller or a Memory block.

7.2 Design Files

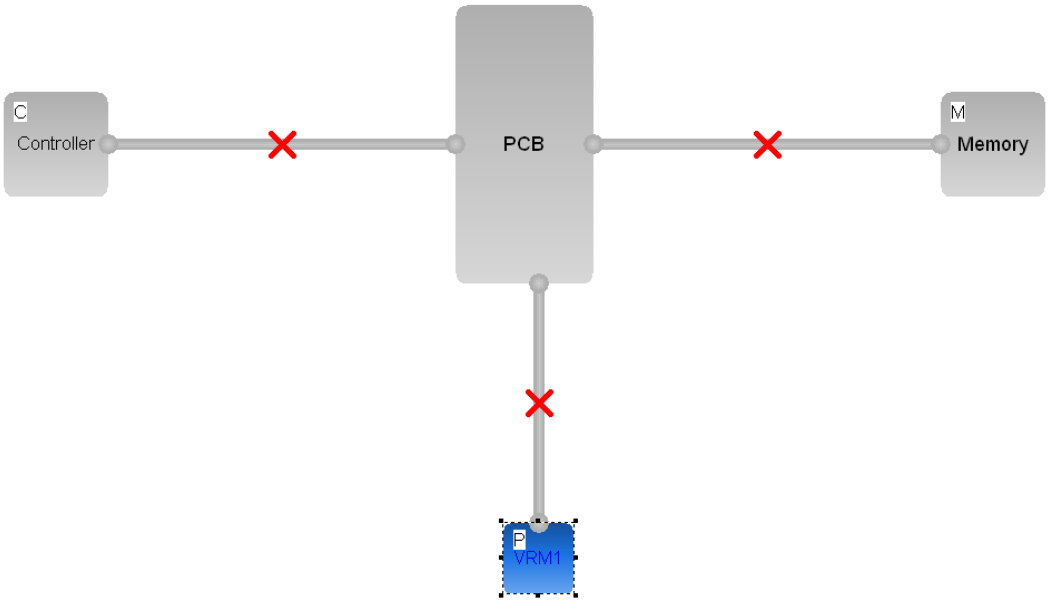


If you have performed the tasks covered in Chapter 6, you can continue with same design files. However, if you are starting from Chapter 7, open the `chap7.ssix` file located at *`<INSTALL_DIR>\SpeedXP\Samples\SystemSI\Parallel Bus Analysis\Tutorial\chap7`*. This directory also contains the IBIS model file for the Controller and Memory.

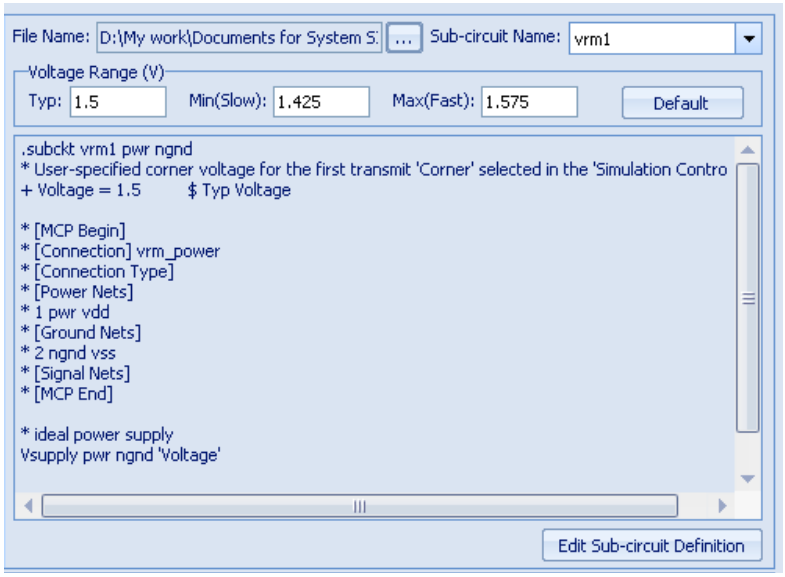
7.3 Define the VRM Block

In this section, we will edit the VRM component and define the model.

1. Select the **VRM** block and open the Property dialog box for the block.



The component has a netlist associated with it. The File Name filed displays the path to the netlist file, `vrm1.sp`. This file was created when the VRM was defined and uses a model with constant DC voltage having Typical, Minimum and Maximum corners as shown.



2. Open the circuit definition netlist by clicking the **Edit Sub-circuit Definition** button. A more complicated VRM model could be defined by entering a model in this netlist in place of the **ideal power supply** entry in this file.

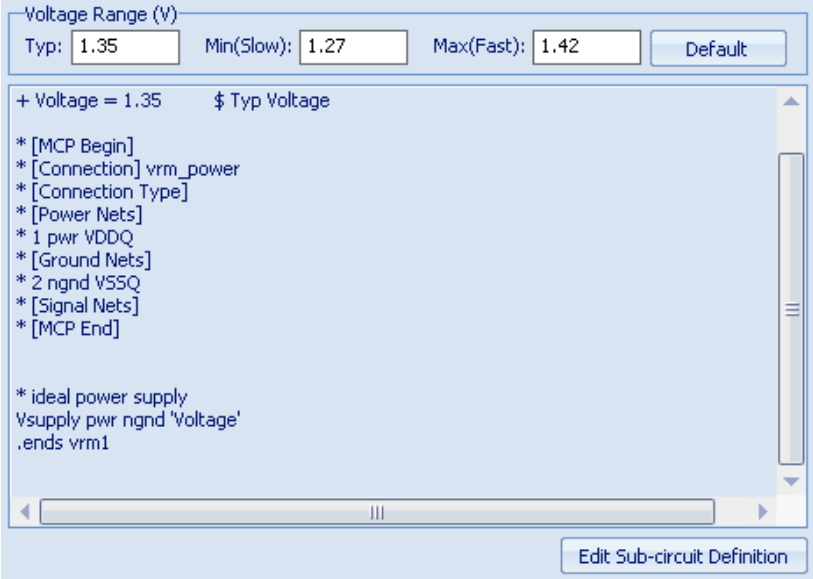
```
.subckt vrm1 pwr ngnd
* User-specified corner voltage for the first transmit 'Corner' selected in the 'Simulation Controller'.
+ Voltage = 1.5        $ Typ Voltage

* [MCP Begin]
* [Connection] vrm_power
* [Connection Type]
* [Power Nets]
* 1 pwr vdd
* [Ground Nets]
* 2 ngnd vss
* [Signal Nets]
* [MCP End]

* ideal power supply
Vsupply pwr ngnd 'Voltage'

.ends vrm1
```

3. Click Cancel to close the dialog box without making any modifications.
4. Modify the DC Voltage source, by changing the values in the Voltage Range group box. Set the new values as :

   Typ=1.35V

   Min=1.27V

   Max=1.42V.



5. To save your modifications click **OK.**
6. Choose **File > Save**.

   At this point all of the components in the Bus are defined. The last step before simulation is to define the connections between each component.

8 Connecting the System

8.1 Introduction

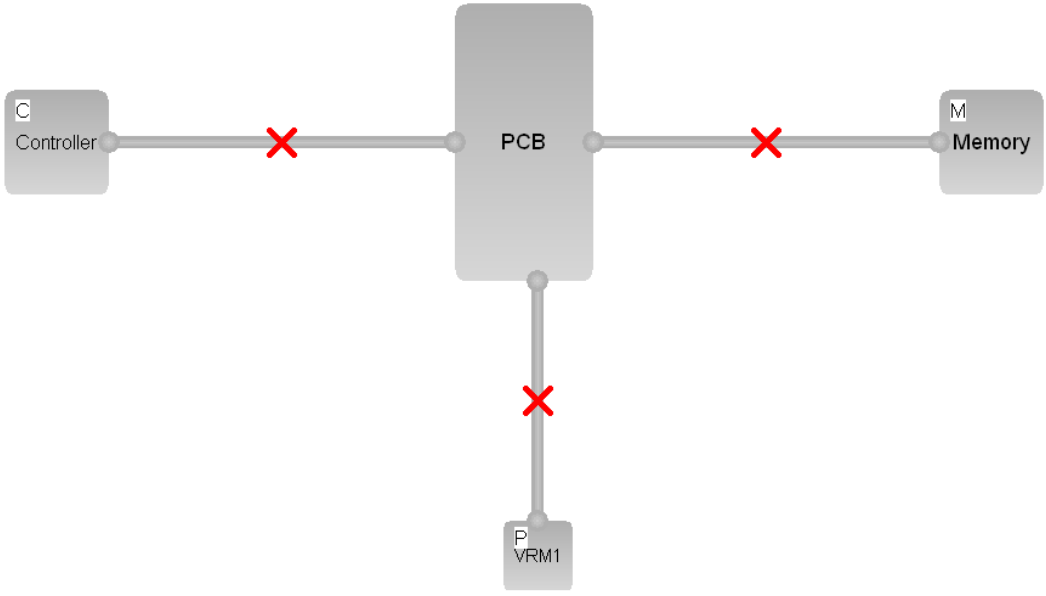
This chapter explains the block connections and Sigrity **Model Connection Protocol (MCP)**. MCP is a netlist description that is either inserted as a header in the circuit netlist for a component, or is separately created for a component. This netlist describes the pins and connections for that block.

For Controller and Memory components, the MCP block is created automatically from the IBIS file by Parallel Bus Analysis. Editing MCP descriptions is not recommended. For the VRM block, the MCP connection is contained in the `.sp` file created by Parallel Bus Analysis. This file includes model information that can be edited, as shown in Chapter 7. For the PCB and other interconnects, the model information is in a **`.CKT`** netlist file. Tools, such as PowerSI, are used for generating .CKT files from the network parameter simulation results.

For most system designs, the MCP connections between components will be created automatically based on common pin names. The exception to this rule is the Power and Ground connections, which usually require manual editing.

8.1.1 Example Files

The block diagram for this chapter consists of a Controller, a PCB, a Memory and a VRM component, as created in Chapter 7. These blocks have been defined and linked to models, but have not been connected together. The design looks like this:



If you have performed the tasks covered in Chapter 7, you can continue using the same design files. However, if you are starting from Chapter 8, open the `chap8.ssix` file located at *`<INSTALL_DIR>\SpeedXP\Samples\SystemSI\Parallel Bus Analysis\Tutorial\chap8`*. This directory also contains the IBIS model file for the Controller and Memory.

8.2 Connection Definitions

The MCP connections between the components are created based on common pin names.

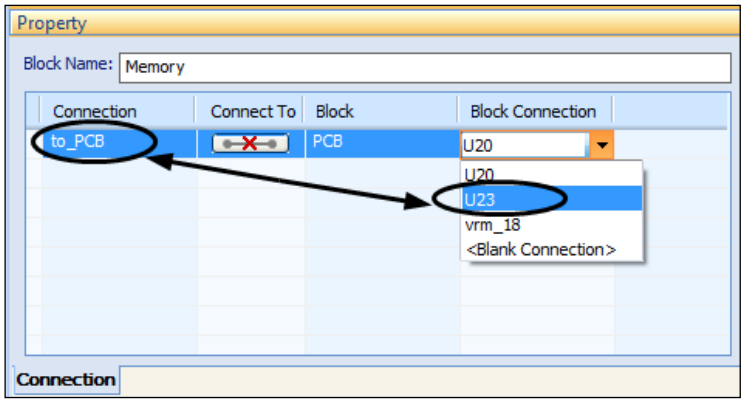
8.2.1 Memory Component

In this section, you will edit the Memory component and define MCP for the Memory block.

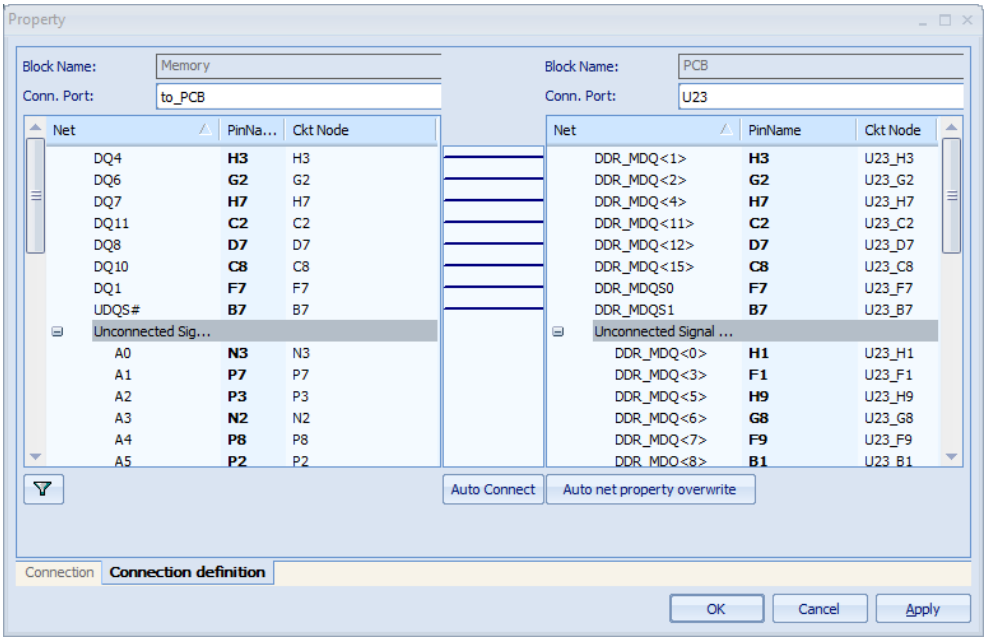
1. Click the **Memory** component Block.

   You need to connect the Memory component to the component **U23** on the PCB.

2. From the Block Connection drop-down list, select **U23**.



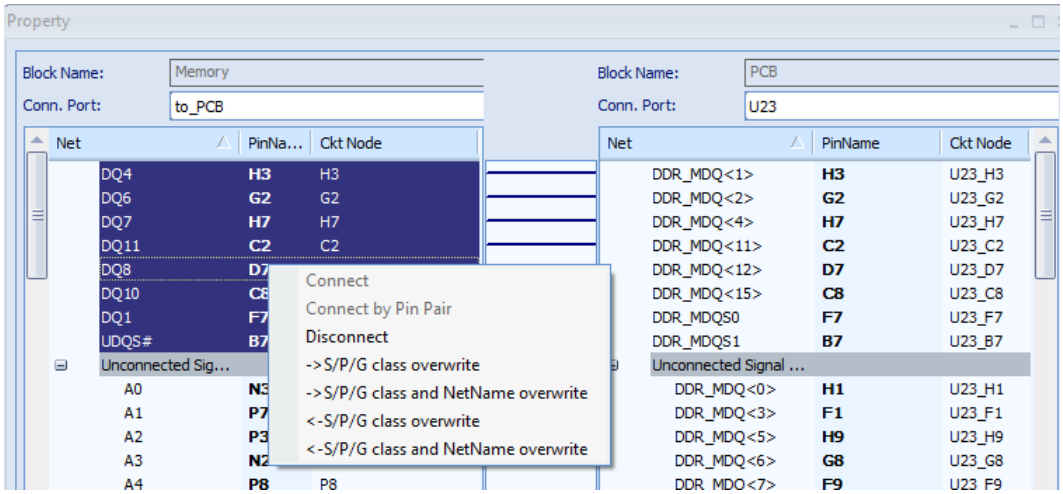
3. To define the connections between the Memory and the PCB, click the wire icon in the ConnectTo column.
   Connections between the memory and PCB blocks are displayed in the Connection Definition tab. By default, signals with same pin names are automatically connected by SystemSI.



To connect the Memory block to PCB block, we need to connect the following:

- Signal DQ1 to DDR_MDQ<1>, DQ2 to DDR_MDQ<2>, and so on.
- LDQS to DDR_MDQS0
- UDQS to DDR_MDQS1
- VDDQ to VDD1.8V_CPU
- VSSQ to GND

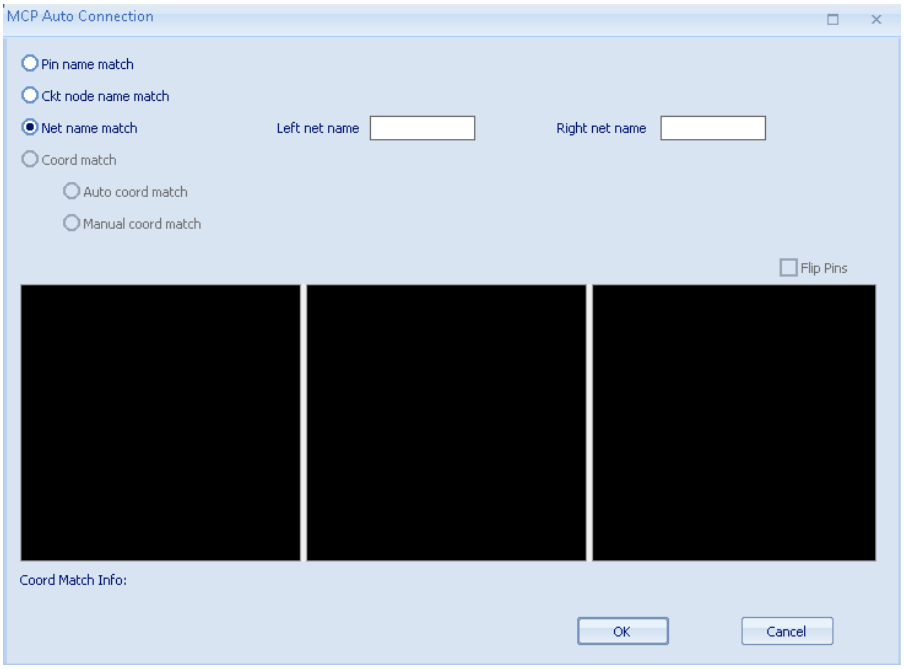
4. Before you start connecting the signals, remove the default connections.
   - Select the connected signals, and right-click.



   - From the pop-up menu, choose *Disconnect*.

5. To connect DQ signals of the Memory block to DDR_MDQ signals of the PCB block, click the Auto Connect button.

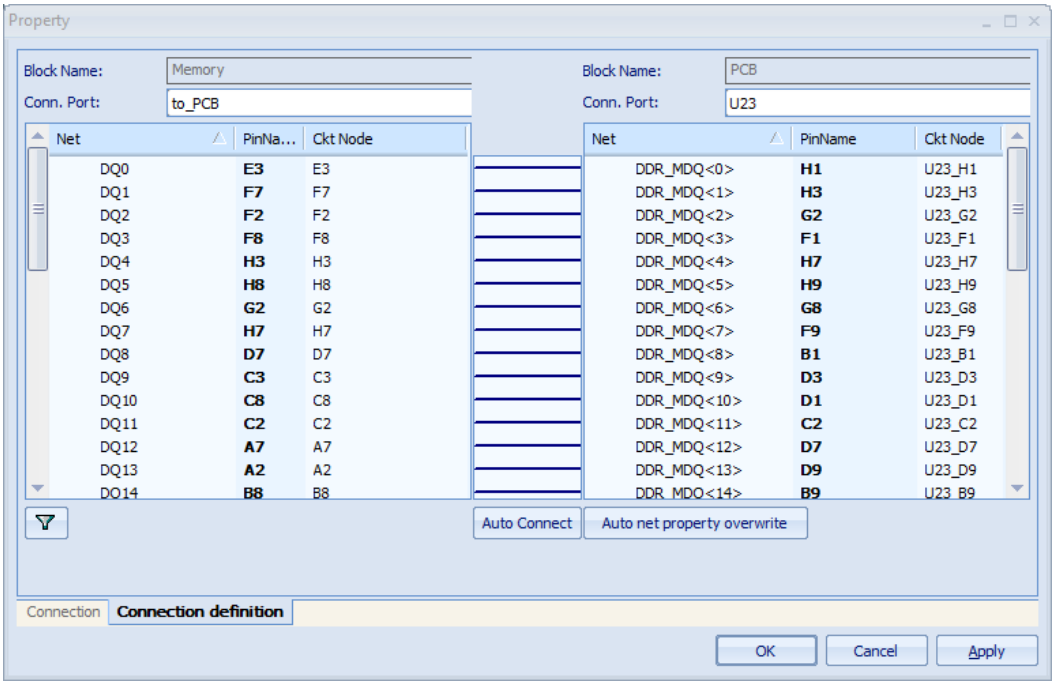
6. In the MCP Auto Connection dialog box, select ***Net name match***.



7. In the Left net name text box, enter DQ*

8. In the Right net name text box, enter DDR_MDQ<*>
9. Click **OK** to close the window.

   The net names are connected as required.

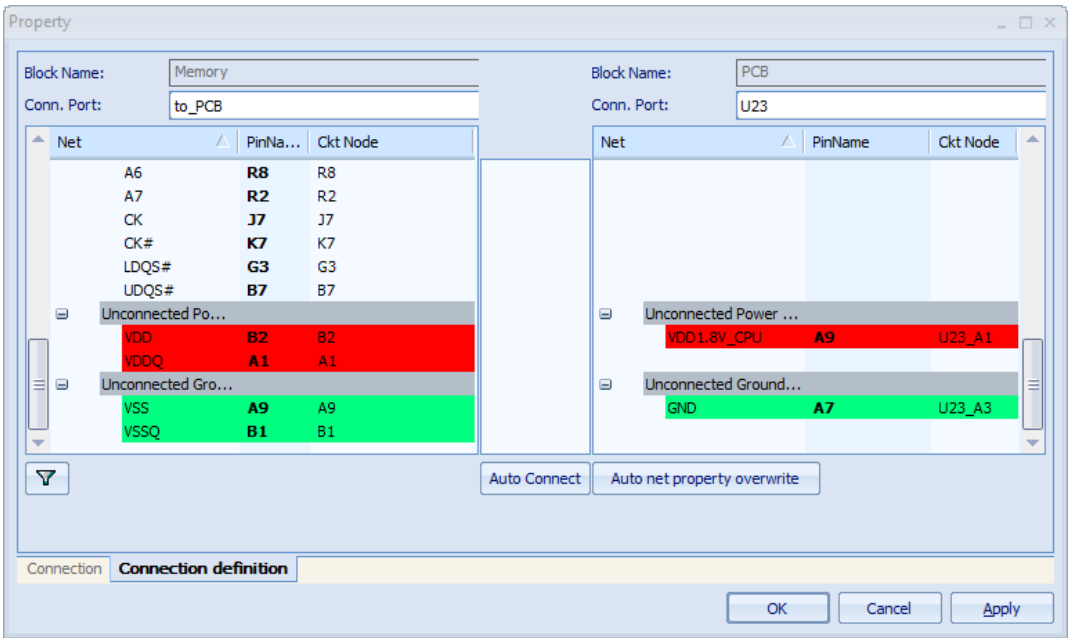


   Next, connect the following nets:

   - LDQS to DDR_MDQS0
   - UDQS to DDR_MDQS1

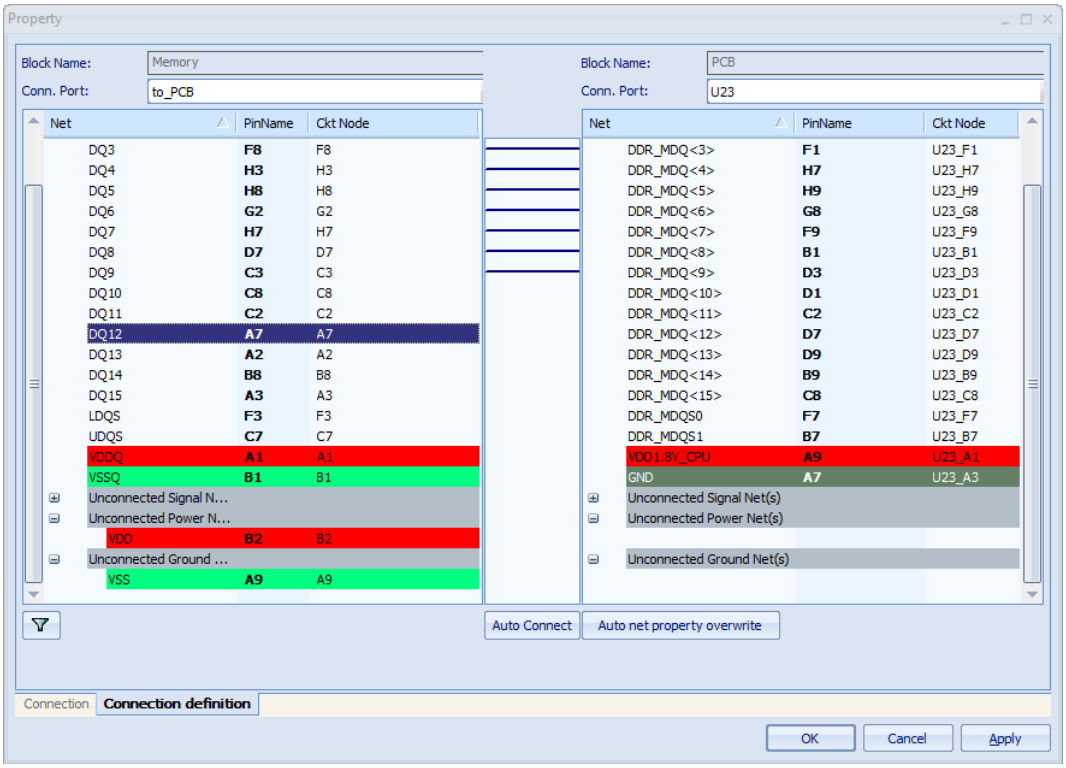
10. Sort the both the columns by NetName.
11. In the left column, select LDQS and UDQS.
12. In the right column, select DDR_MDQS0 and DDR_MDQS1.
13. Right-click and select Connect by Pin Pair.

    This completes the process of connecting Signals of PCB block to the memory block. Next, you need to connect the Power and Ground Signals.
14. To connect the power signals, scroll down the list to display the Unconnected Power Nets.



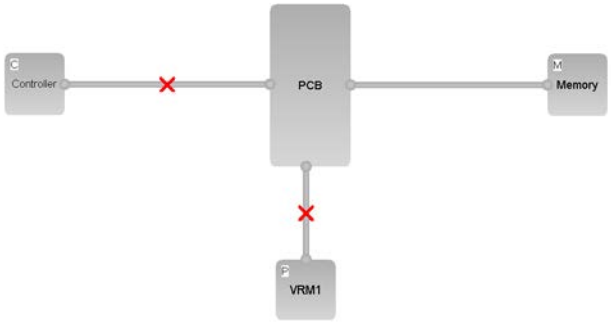
15. From the left column, select VDDQ.
16. From the Right column, select VDD1.8V_CPU and right-click.
17. From the pop-up choose, *Connect*.
18. Similarly, connect VSSQ to GND.

    All PCB signals are now connected. There is no unconnected signal for PCB block as shown in following figure.



19. Click **OK**.

The Property dialog box closes and the **Block** diagram is also updated to show a valid connection between **PCB** and **Memory**.

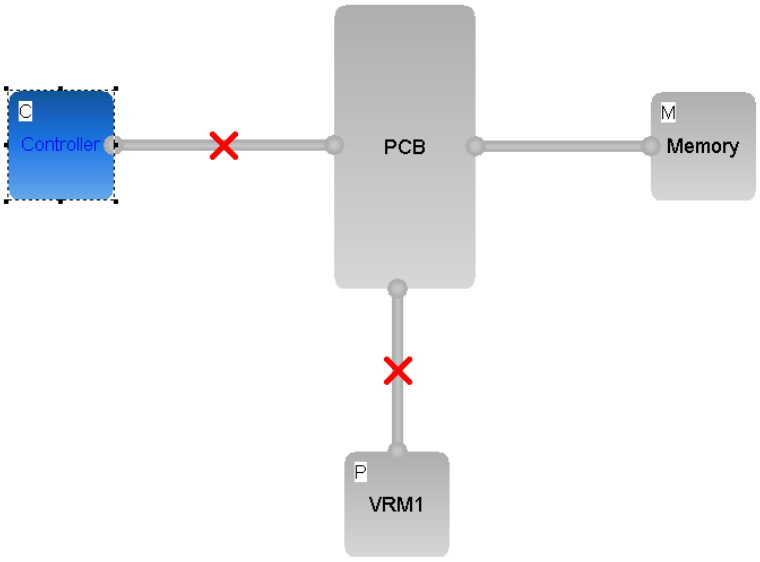


Note!	There can be cases where data nets and some **Power** and **Ground** pins are not connected, or incorrectly connected, because the pin names in the .ckt file do not directly match the pin names in the IBIS files. For such cases, you need to manually map the pins. For procedural details, see Unmapped Pins in MCP Editor.

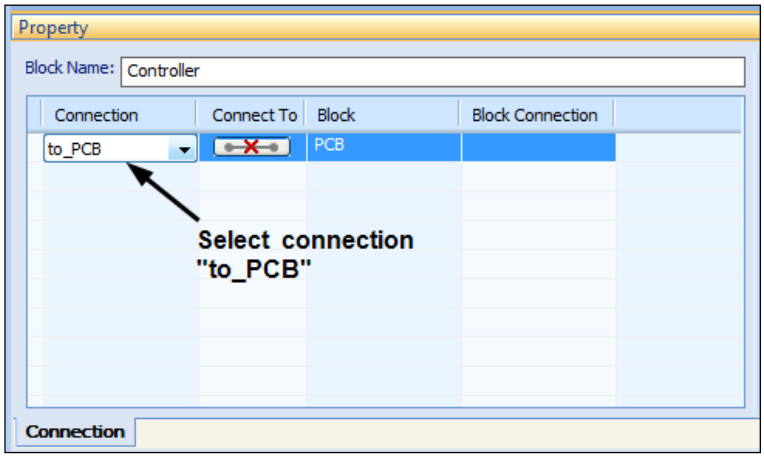
8.2.2 Connecting Controller Component

In this section, you will connect the Controller block to the PCB block. For this, open the Property dialog box for the Controller block.

1. To open the Property dialog box for the Controller block, double-click on the Controller block.



2. Verify that the value in the first Connection Port column is set to **to_PCB**.



3. Next, click second Conn.Port column, and from the drop-down list of PCB connections, select **U20**.

   The Red Cross (X) in the Connect To column indicates unconnected pins.

4. To connect the two blocks, click the Wire symbol with a red Cross, in the Connect To column.

   The Connection Definition tab opens listing all the signals for the block.

5. To enable connecting signals `DQ* to DDR_MDQ<*>`, you can either use the Auto Connect button (see section *8.2.1*,) or perform the following steps.

   a. Select the filter button.

b. Select the *Enable Unconnected Pins Filter* check box.

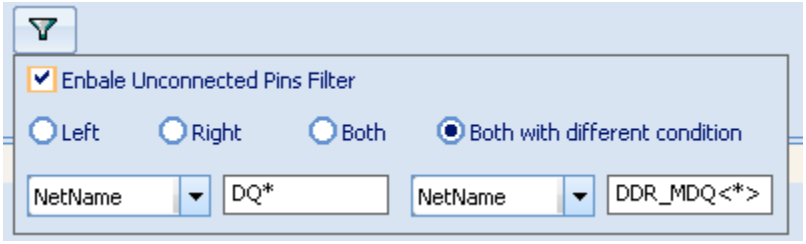
c. Select the ***Both with different condition*** option.

d. From the first drop-down list, select NetName.

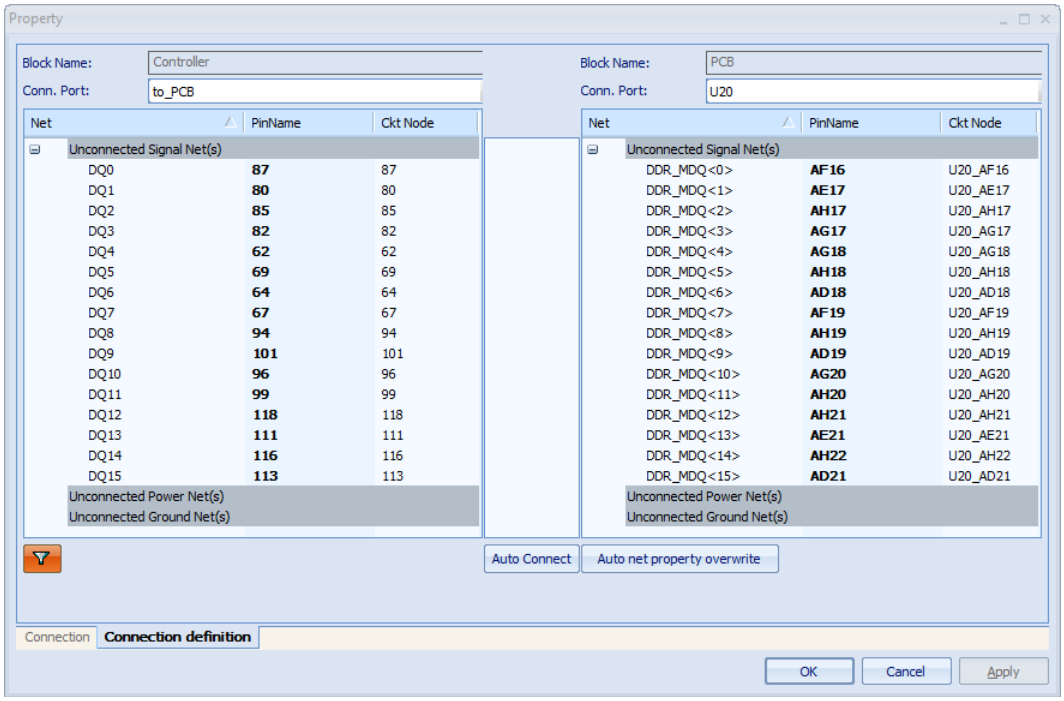
e. In the text box, enter DQ*.

f. From the second drop-down list select NetName.

g. Enter `DDR_MDQ<*>` in the text box and click anywhere outside the filter window.

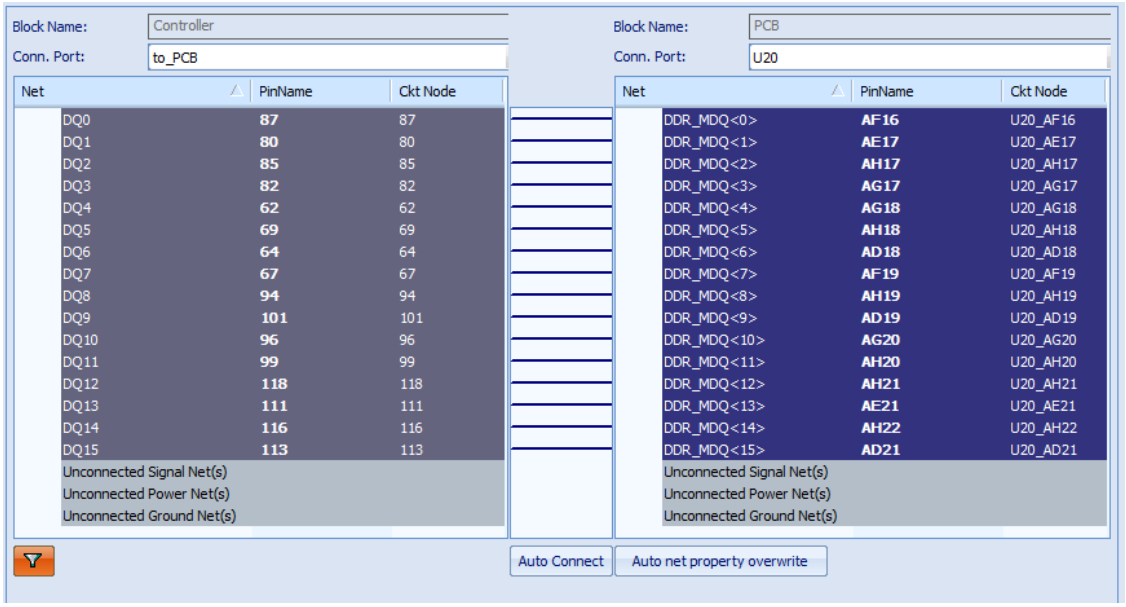


The required signals are filtered and listed in the Property window.



6. Select the nets in columns, right-click and select ***Connect by Pin Pair***.

The signals are connected as shown in the following figure.



7. Clear the filter settings.
8. Connect the following signals.
   a. LDQS to DDR_MDQS0
   b. UDQS to DDR_MDQS1
   c. VDDQ to VDD1.8V_CPU
   d. VSSQ to GND
9. Click Apply.

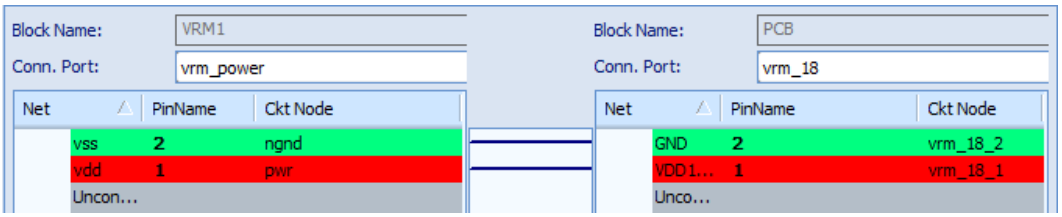
   The Block Diagram is updated and now shows a valid connection between the Controller and PCB, as well. As the ground signal is now connected, the red cross on the connection between Controller PCB block is removed.

Note!	There can be cases where data nets and some **Power** and **Ground** pins are not connected, or incorrectly connected, because the pin names in the .ckt file do not directly match the pin names in the IBIS files. For such cases, you need to manually map the pins. For procedural details, see Unmapped Pins in MCP Editor.

8.2.3 Editing the VRM Block

1. Open the Property dialog box for VRM block.
2. Verify that the value in the first Connection Port column is set to **vrm_power**.
3. Click second Conn.Port column, and from the drop-down list, select **vrm_18**.

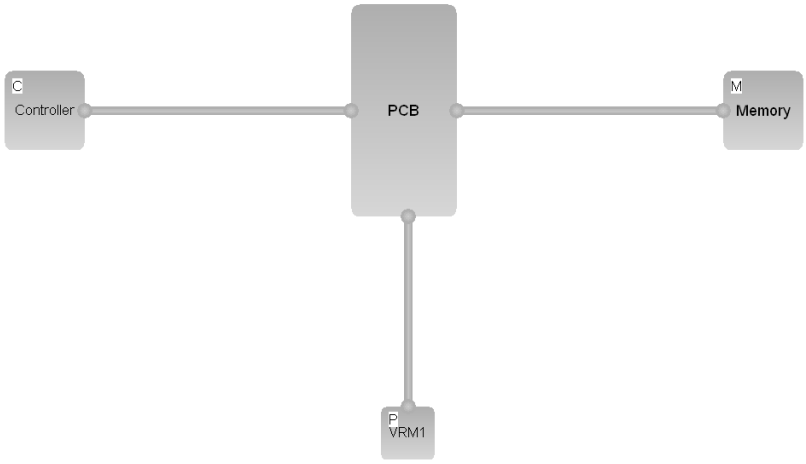
   The wire icon in the Connect To column changes to green wire, indicating the complete connection.
4. To view the MCP connection, click the green wire icon in the ConnectTo column.



Because of the same PinNames, the ground and VCC signals were connected automatically by SystemSI.

5. Click OK.

   The Connector icon changes from a Red **X** to a Green Wire and the block diagram is also updated.



The design is now ready for Simulation Setup.

8.2.4 Unmapped Pins in MCP Editor

Connectivity in the MCP Editor is based on pin names. Signals with same PinNames are connected automatically. For some designs, it may happen that the pin names do not directly match. As a result, MCP Editor will have data nets, and Power and **Ground** pins that are either not mapped, or are incorrectly mapped.

For such scenarios, you need to manually edit the connections in MCP. This section lists the steps to be followed for manual connections.

Connecting Unmapped Pins

Following figure shows the connectivity of a Memory block. The blank column between two blocks indicates that there is no connection for the DQ lines. This is because there are no common pin names between the `.ibis` file and the `.ckt` file, resulting in multiple unconnected data nets.

To see the steps for connecting the pins, see sections 8.2.1 and 8.2.2.

Modifying Incorrect Connections

Following figure shows the connections of a Memory block in the MCP Editor window. The data nets of the Memory block are incorrectly connected to the Power and Ground nets.

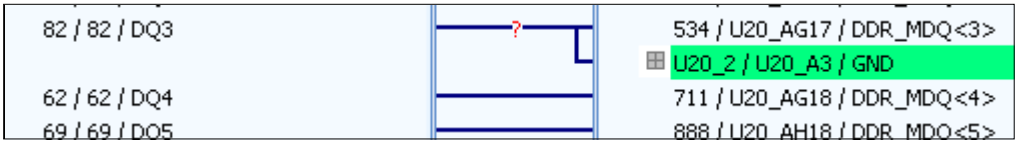


Figure 1: Incorrect Mappings

To fix such connections, you need to follow a 2-step process. First, disconnect the PCB block pins, and then reconnect the pins to correct data nets.

To remove the wrong connections shown in Figure 1, perform the following steps.

- Select all the nodes in the U20 Connection of the PCB block, and right-click.
- From the pop-up menu, select **Disconnect**.

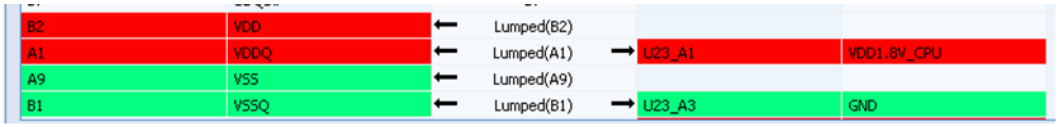
  The connections are removed. The nodes in the PCB block are temporarily floating, so that they can now be matched up to the correct pins in the Memory block.

With the connections on the PCB side clear, we can now connect correct signals. For example, in MCP Editor, connect CKT node **A1-VDDQ** to the **U23_A1,** both in **RED.**

- To connect the nets in the memory block to the Ckt nodes in PCB, first select the net on the memory side, and then select the PCB net.
- To accept the MCP Mapping, click **Apply**.

Similarly, repeat the step to connect Ground, **B1 - VSSQ** on the Memory side to the **U23_A3** on the PCB

Both sets of pins are now be mapped correctly to each other as shown in the following figure.



	In SystemSI - Parallel Bus Analysis, the pin names for the Controller and Memory blocks come from the original IBIS file, and are considered

IMPORTANT!	fixed. So in this particular case, it is necessary to remove and remap the pin names from the U23 Connection, as the pin names in the Memory block are fixed.
Summary!	The methodology for connecting pins through the MCP editor is consistent regardless of type of pin. 1. Disconnect the PCB "side" by selecting "Disconnect". 2. Re-connect the pins by clicking one first, then the other.

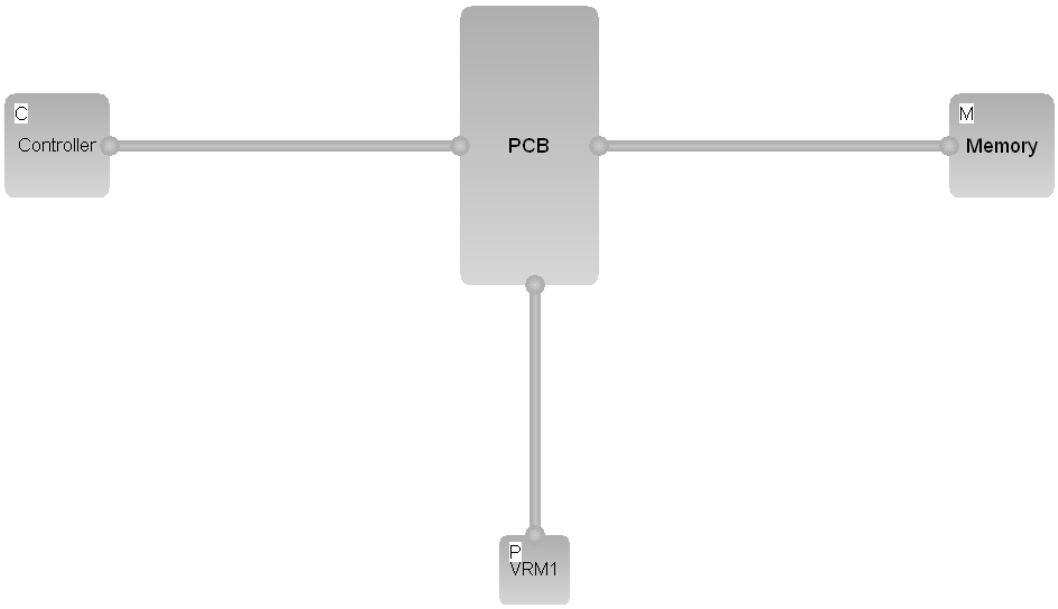
9 Simulation

9.1 Introduction

This chapter introduces the basic simulation settings in the **Analysis Options**, and the data presentation window. You will then perform an initial simulation, then make several changes to the bus design and simulation settings, and perform some experiments on the bus.

9.2 Tutorial Files

The block diagram for this chapter consists of a **Controller**, **Memory**, **PCB,** and **VRM** component, as created in previous chapters. This design is fully defined with models for each component and a valid connection between each. The design looks like this:

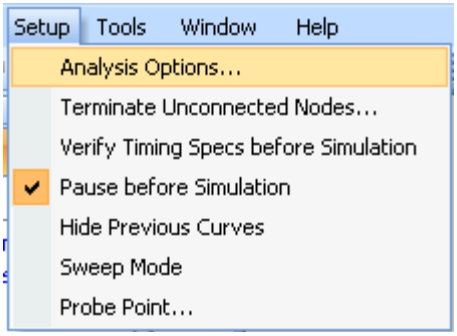


If you have performed the tasks covered in Chapter 8, you can continue with same design files. However, if you are starting from Chapter 9, open the `chap9.ssix` file located at *`<INSTALL_DIR>\SpeedXP\Samples\SystemSI\Parallel Bus Analysis\Tutorial\chap9`*. This directory also contains the IBIS model file for the Controller and Memory blocks.

9.3 Analysis Options

The **Analysis Options** dialog box contains most of the details for the bus simulation. All aspects of the Analysis Options are covered in the next several sections.

1. To open the **Analysis Options** dialog box, from the Setup menu, select **Analysis Options**:



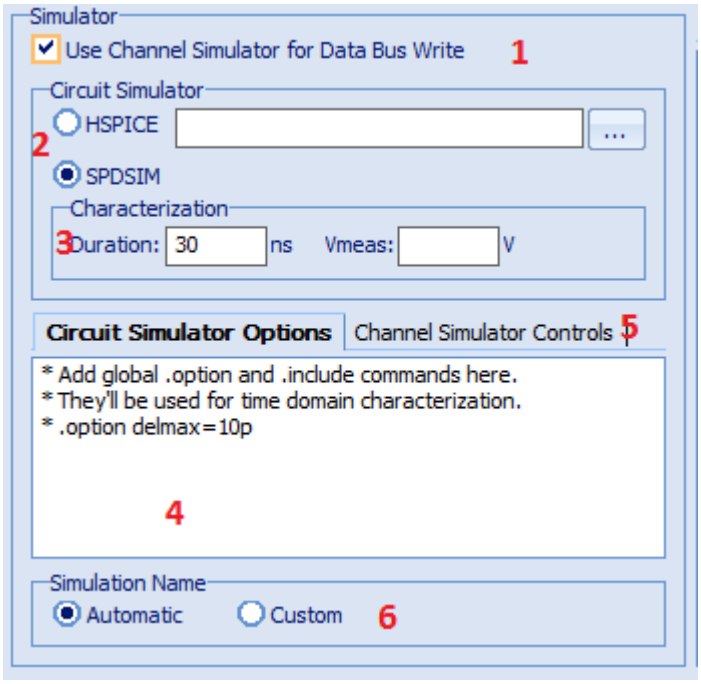
The dialog box has sections:

- Circuit Simulator
- Simulation Configuration
- Stimulus Definition and Model Selection
- Restore Defaults

Each of these options is explained in the following sections.

9.4 Simulator

The default circuit simulator in SystemSI is Sigrity SPDSIM. SPDSIM is the same time domain simulator used in Sigrity SPEED2000. Parallel Bus Analysis can also use HSPICE if available. Some model types, including transistor-level driver models, require HSPICE.



Using the options in the Simulator group box, you can specify the following:

1. Use Channel Simulator for Data Bus Write
   Select this check box if you want to perform BER analysis on DDR4 data buses.
2. Simulator of choice.
   By default, Sigrity SPDSIM is selected. HSPICE can be used if it is available.

3. Characterization Options

   These options are available only if the *Use Channel Simulator for Data Bus Write* check box is selected.

   - **Duration**: Refers to the duration of the Characterization run with the specified Circuit Simulator. The duration value specified should be long enough to allow any reflections to settle out, and to allow the waveforms reach their steady state.
   - **Vmeas**: Refers to the voltage threshold at which delay is measured from the Characterization. This information is included in the channel simulation report as Delay. If Vmeas value is not explicitly called out in an IBIS file associated with the Tx block, Vmeas is taken as the midpoint of the voltage swing seen in the Characterization waveform.

4. Circuit Simulator Options

   Global options, including .OPTION and .INCLUDE cards can be specified in this window. For HSPICE simulation, accurate characterization usually requires the DELMAX option, which sets the maximum allowable transient analysis time step size.

5. Channel Simulator Controls

   This tab is visible only if the *Use Channel Simulator for Data Bus Write* check box is selected.

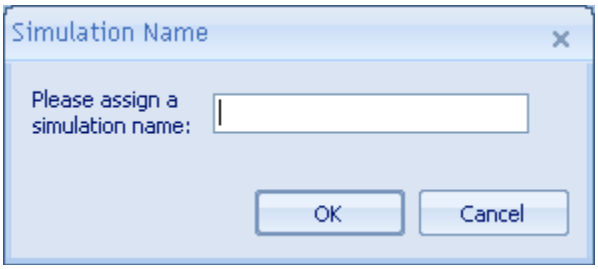
   Use this tab to specify Channel Simulator controls. These controls are reflected in the simulation results.

NOTE!	For detailed explanation of channel simulation options, see *SystemSI – Serial Link Analysis Tutorial*.

6. Simulation Name.

   - If the **Automatic** option is selected, the result folder names are automatically defined according to the simulation times.
   - With the **Custom option** selected, click the **Play** button  . The **Simulation Name** window pops up.

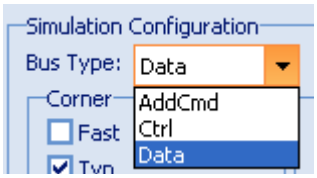
     You can define result folder names manually according to the simulation times.

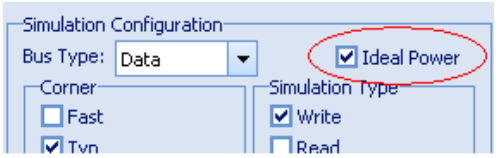
9.5 Simulation Configuration

The **Simulation Configuration** pane allows you to specify key details of the simulation, including selection of the bus, as well as corner model combinations.

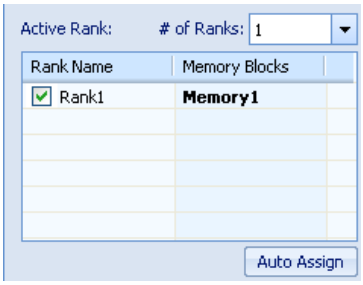
1. Specify the Bus to be simulated. Most common bus types defined for a project are **Address**, **Clock** and **Data** bus. Choose the **Data** bus.



2. The simulation may be run with **Ideal Power** assumed. If this is selected, the VRM model is not used, and an ideal DC voltage is assumed for each device, based on the Corner selected. Also, all power and Ground contacts are assumed to be a single common reference.

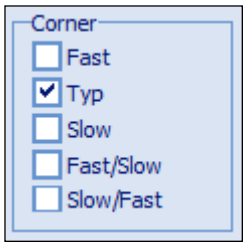


3. For the selected bus, choose the active memory blocks. In this project, there is only one memory block, so enable **Rank1** check box.

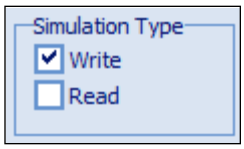


4. The Simulation **Corners** can be specified for the Controller and Memory devices. This allows you to specify the IBIS model **Min/Max/Typ** for specific timing simulations, through the Parallel Bus Analysis GUI. For this project, use the nominal model, or **Typ**.

These corners correspond to device characteristics as well as operating conditions, and are required elements of the IBIS model.



5. The Simulation **direction** can be specified for the bus, either **Write** or **Read**. For this simulation, select **Write**.

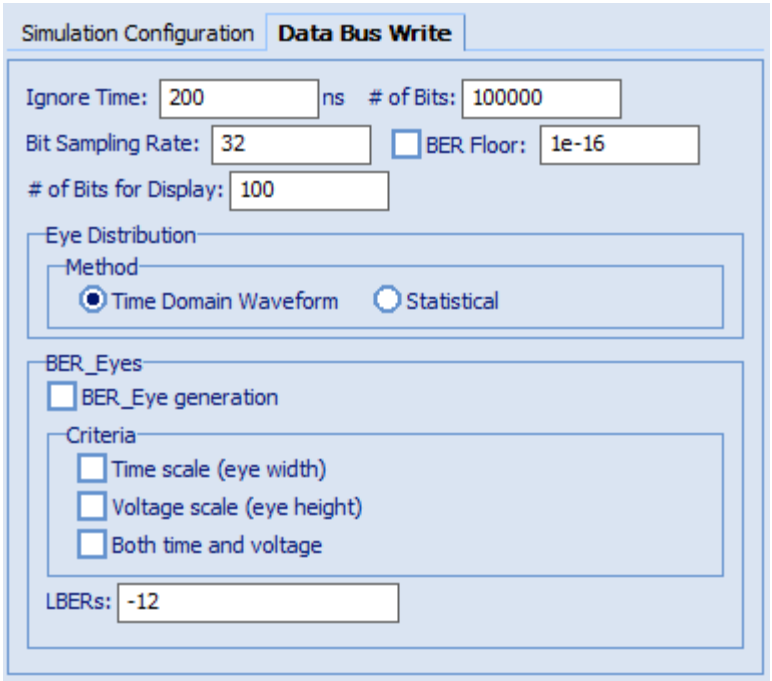


NOTE!	Multiple Corners and Simulation Types can be selected for a given simulation. In this case, successive simulations will be run, with separate results created for each corner.

9.6 Data Bus Write

In this tab, you specify the parameters to run Data Bus Write simulations with the channel simulator to perform BER analysis.

Important!	This tab is visible only if the *Use Channel Simulator for Data Bus Write* check box is selected.



- **Ignore Time** (ns): Specify the initial time to be ignored from the waveform, so that the data is not corrupted with the startup time transients. The default value is 200ns. You can use a lower value such as 100 ns if you do not use adaptive equalizers like adaptive DFE.
- **Number of Bits:** Use this text box to specify the number of bits to be simulated. The default value is set to 100,000 bits, which is the minimum number of bits required for BER computation.
- **Bit Sampling Rate:** This parameter controls the granularity used by the channel simulator to compute the eye density. This is analogous in nature to the timestep control in a traditional circuit simulator. The larger the number is, the longer the simulation time is. Default value is 32 samples/bit.
- **BER Floor:** Specifies the minimum Bit Error Rate (BER) to be used in the simulation. The default value is 1e-20.
- **# of Bits for Display:** The channel simulator generates millions of bits worth of waveforms. Saving all this data, takes up significant disk space and slows down the display performance. To avoid this, SystemSI provides users with an option to specify the number of bits worth of raw waveforms to be saved to disk and displayed. For example, if you enter 100 in the field, SystemSI save last 100 bits of raw waveform data for display.

9.6.1.1 Eye Distribution Methods

Select the method to be use for generating Eye diagrams.

Time Domain Waveform: Uses time domain convolution method for generating Eye diagrams. In this method, real stimulus patterns, including TX jitter effect, are convolved with channel response to produce the final result. Use of this method is recommended, if the design has AMI model with `getwave` function, or if you want to inject TX jitter.

Statistical: This method uses probability density function (PDF) method to produce the final result.

9.6.1.2 BER_Eyes

Select the **BER_Eye generation** check box to generate the statistical eye diagram for a specific BER.

By default, the BER_Eye generation takes into account the time scale as well as voltage scale. However, you can modify the default selection and select either Time scale check box or Voltage scale check box.

LBERs: Log of the bit error ratio at which BER eye is generated.

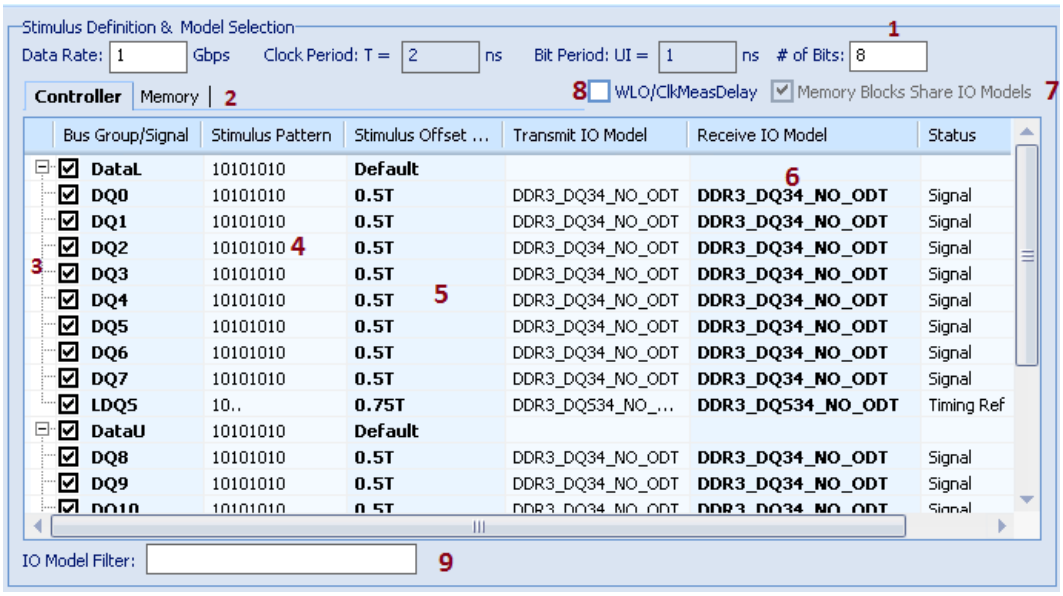
NOTE!	For detailed explanation of channel simulation options, see *SystemSI – Serial Link Analysis Tutorial*.

9.7 Stimulus Definition and Model Selection

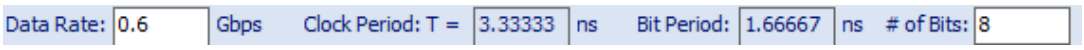
The Stimulus Definition and IO Model Selection pane of the Analysis dialog box is used for source definition, including data rate, delay and pattern. It also allows you to select IO models for both Controller and Memory from the different models present in the IBIS files, based on the [Model Selector] syntax. Lastly, timing can be specified, and is linked to the Worst-case Setup / Hold calculator.

9.7.1 Controller Setup

The following figure shows the simulation settings for the Controller in a Write simulation. Each part of this dialog box is defined and specified below.



1. Basic source parameters can be set in the TOP part of this dialog box as follows:

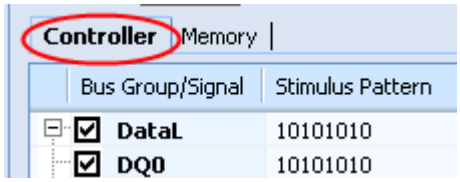


   - Set **Data Rate** to ***0.6 Gbps***

     The **Bit Period** and **Clock Period** (**T** for this data rate) is calculated and displayed, from the value entered.

   - Set **Number of Bits** to be simulated to **8**

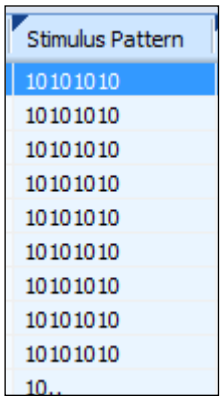
2. Tabs for the Controller and Memory are available. Select the Controller first.



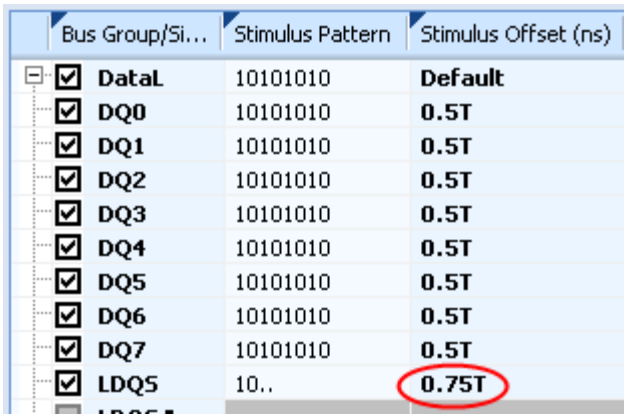
3. Specify the Bus Groups and signals for this simulation. The Controller has two bus groups defined, **DataL**, **DataU**. The single memory device only has two corresponding bus groups, **DataL** and **DataU**, which connects **DQ0** through **DQ15**. We can turn these buses **On** or **Off** by selecting in this dialog box. Make no changes to this setup.

   In case of Data Write simulations, if the data bus has a CLOCK signal defined and connected, the clock signal is also included in this list, and can be selected for simulations as shown in the following figure.

4. A unique stimulus pattern can be defined for each data line and strobe. For this simulation, an identical 1010… pattern stimulus is defined for each data and strobe line.



5. The relative delay can be defined for each data line and strobe. For this simulation, an IDEAL offset of a quarter clock period between Data and Strobe, is used.



NOTE!	Parallel Bus Analysis has a utility for calculating non-ideal, **Worst-case Setup and Hold** values for these delays. See ***Section 9.6.3*** below for more information.

6. The various models defined in the IBIS file may be viewed and a selection made for each signal line. For Transmit (Controller) use the "DDR3_DQ34_NO_ODT" model

NOTE!	If both Write and Read options are selected, then along with Transit IO model, you also see a column for Receive IO model.

7. Memory Block Share IO Model: Select this option if the Memory blocks should share same IO model. If this option is not selected, different IO models can be specified for the Memory blocks.

   For designs with single memory device, this option is selected by default and is disabled.

8. **WLO/ClkMeasDelay**: This option is enabled only for Data buses and is useful for designs with multiple memory blocks. To generate useful values of Write Leveling offset (WLO) and the Clock Delay (ClkMeasDelay), the clock and the Timing Reference signals must be defined and connected for data buses.

   If this option is selected for Data Bus Write simulation, WLO is added to the Stimulus Offset of the data and timing reference signals at the Controller.

   In case of Data Bus Read simulation, the ClkMeasDelay value is added to the "Stimulus Offset" of the data and timing reference signals at the Memory:

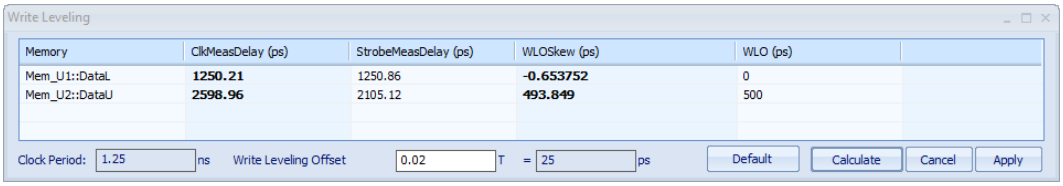
   Depending on the topology, the WLO and ClkMeasDelay values are different for each Data Bus Group. The WLO and ClkMeasDelay values used during simulation are specified or calculated in the Write Leveling Dialog box.

   To invoke the Write Leveling dialog box:

   a) From the **Tools** menu, choose **Write Leveling**.

   b) In the Write Leveling dialog box, you can either specify a value for the Write Leveling Offset Resolution, or use the default value.

   c) Select the Calculate button to auto calculate the WLO and CLK delay values by simulating the design.

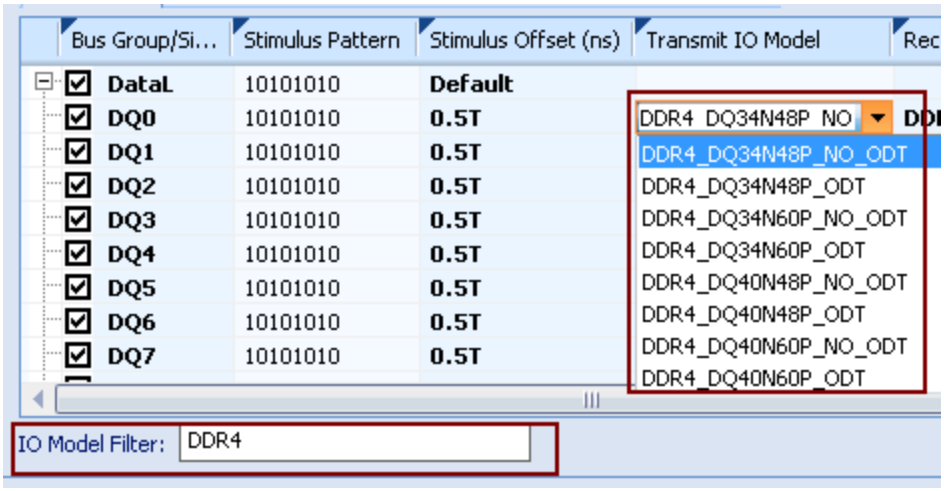
   

   If required, you can manually edit the ClkMeasDelay, StrobeMeasDelay, and WLO values.

   d) Click OK to save the WLO values.

9. The IO Model Filter text box allows you to filter the IO models listed in the drop-down list.

   For example, to view only DDR4 IO models, enter `DDR4` in the IO Model Filter textbox. The drop down list now is modified to display the IO models starting with `DDR4`, as shown in the following figure.

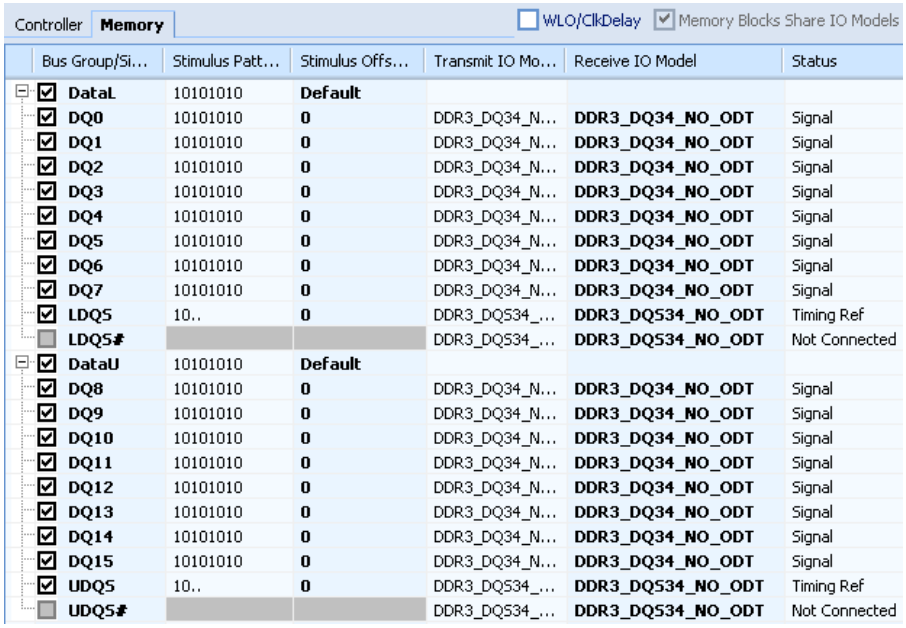


10. Click **OK** to accept all settings.

9.7.2 Memory Setup

Click the **Memory** Tab. Notice that the memory device (from the IBIS model) has two buses defined for it, **DataL** and **DataU**.

1. Click the **Receive Model** field on one of the lines to view the Models. Make no changes to the Memory model at this time.

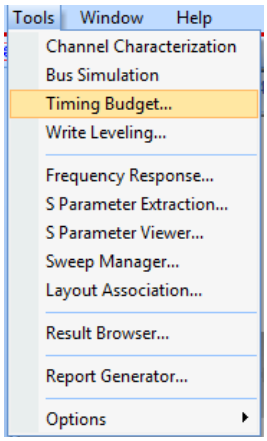


2. Note that there is a single Timing Reference for each bus.

9.7.3 Timing Budget

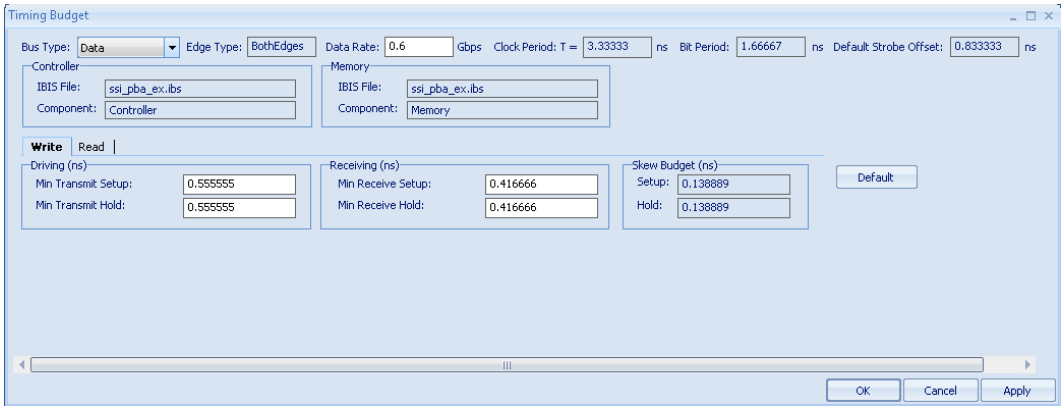
Parallel Bus Analysis comes with an integrated tool that automatically calculates the worst case Setup and Hold values from data set by the user.

1. To launch the **Timing Budget tool**, select Tools > Timing Budget….

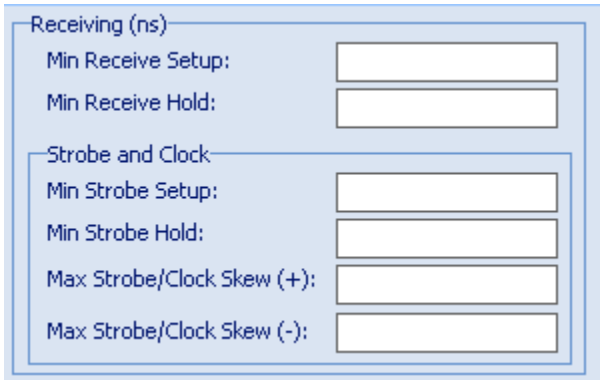


In the **Timing Budget** dialog, note the values updated from the **Analysis Options**, including Data Rate, in addition to IBIS models and Bus type.

2. To set **Driving**, **Receiving** and **Skew Budget** values, click the **Default** button, or you can enter desired values in corresponding fields.



If the data bus in your design, has a clock signal defined and connected, then for all Data write simulations, the Strobe and Clock section is also available. This allows you to measure timing margin between clock signal and data strobe signal.

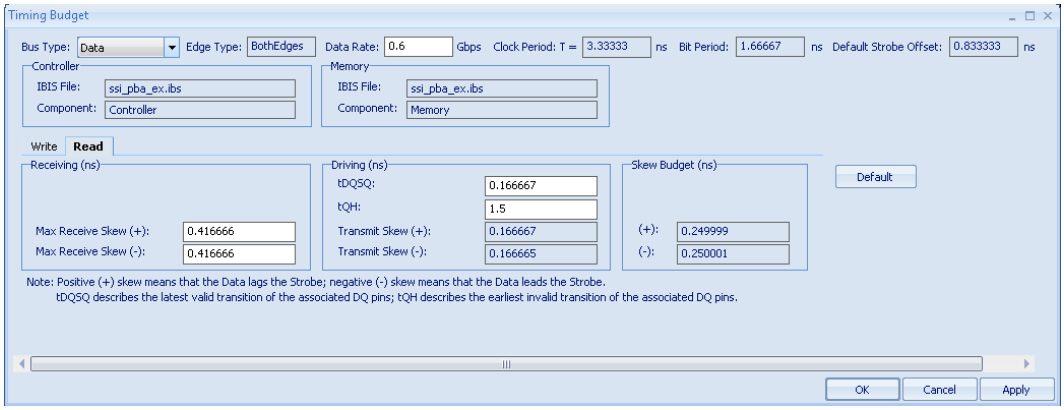


Use the Default button to populate the fields with default values.

3. Click the **Read** tab.
4. Set **Driving**, **Receiving** and **Skew Budget** values. You can either specify custom values or use the default values.

Use the Default button to populate the fields with default values.

Select the **Default** button.



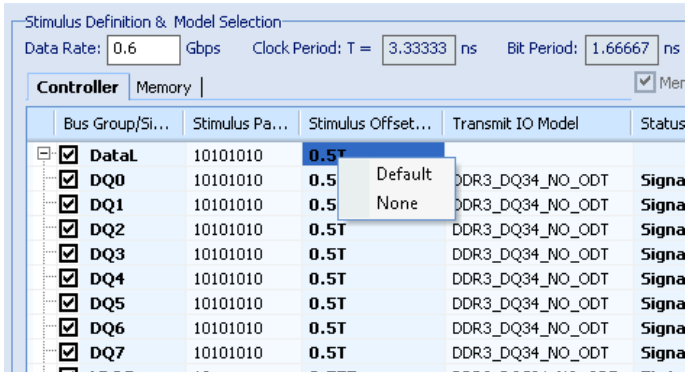
5. Click **OK**.

   The **Timing Budget** dialog closes. It goes back to the **Analysis Options** window. (Go to the **Analysis Options** by selecting Setup > Analysis Options.)

6. In the **Controller** tab, go to **DataL Line**, right-click the **Stimulus Offset** field.

   A pop-up menu appears.

7. Select Default.



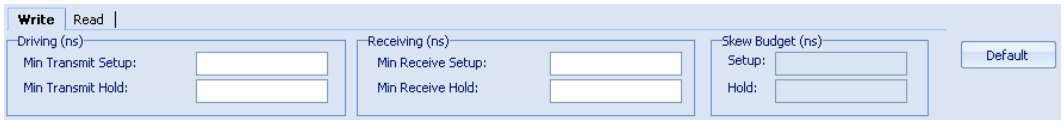
8. Click **OK**.

9.7.3.1 Timing Parameters Definition

The **Timing Budget** form allows you to enter the timing specifications for the project . The timing specifications associated with the transmitting component enables worst-case phase shifts to be applied on the Timing Reference signal to simulate worst case timing conditions. The timing specifications associated with the receiving component allow final timing margins to be computed.

The **Timing Budget** form has different fields depending on **Write** or **Read** case.

- For the **Write** case (also applies to AddCmd and Ctrl buses), the fields appear as follows:



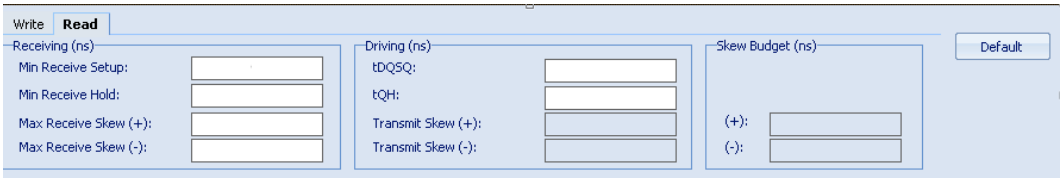
- The **Driving** and **Receiving** parameters are as follows:

- **Min Transmit Setup** — This is the minimum amount of setup time that is guaranteed to exist between the signals and their Timing Reference at the driving component, Controller.
- **Min Transmit Hold** — This is the minimum amount of hold time that is guaranteed to exist between the signals and their Timing Reference at the driving component, i.e. the Controller.
- **Min Receive Setup** — This is the amount of setup time required between the signals and their Timing Reference at the receiving component, i.e. the Memory. This is typically given in data sheets as "tDS(base)" for Data buses and "tIS(base)" for AddCmd buses.
- **Min Receive Hold** — This is the amount of hold time required between the signals and their Timing Reference at the receiving component, i.e. the Memory. This is typically given in data sheets as "tDH(base)" for Data buses and "tIH(base)" for AddCmd buses.

- The **Skew Budget** is automatically calculated from the parameters above, and is intended to show the user how much skew can be introduced by the interconnect while still meeting timing requirements. These are calculated as follows:
  - **Setup** (Skew Budget) = Min Transmit Setup – Min Receive Setup
  - **Hold** (Skew Budget) = Min Transmit Hold – Min Receive Hold
- The **Default** 

button can be used when data sheets or timing specs are not available. Using this button populates these fields with typical values by simply allocating one third of the Signal Bit Period for Transmit Setup/Hold and one quarter of the Signal Bit Period for Receive Setup/Hold.

- For the **Read** case for Data buses, the following fields are available:



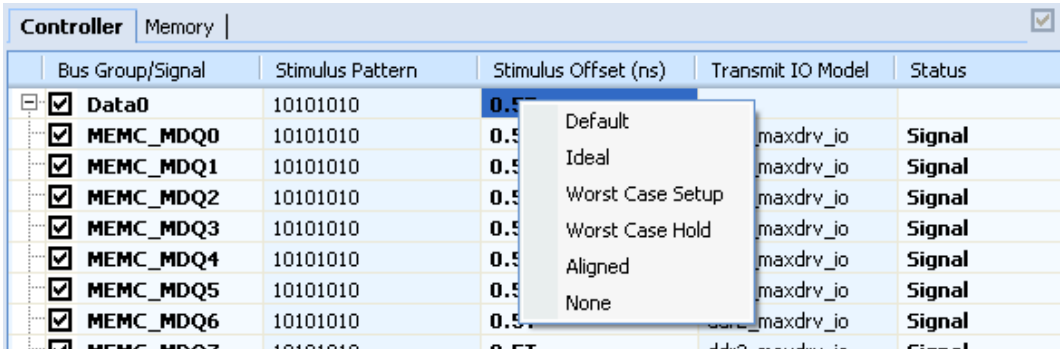
1. The **Receiving** and **Driving** parameters are as follows:
   - **Max Receive Skew (+)** — This is the maximum amount by which the Data is allowed to lag the Strobe at the receiving component, i.e. the Controller.
   - **Max Receive Skew (-)** — This is the maximum amount by which the Data is allowed to lead the Strobe at the receiving component, i.e. the Controller.
   - **Transmit Skew (+)** — This is the maximum amount that the Data will lag the Strobe signal at the driving component, i.e. the Controller. This is derived from the "tDQSQ" parameter commonly found in data sheets for the Memory.
   - **Transmit Skew (-)** — This is the maximum amount that the Data will lead the Strobe signal at the driving component, i.e. the Controller. This is derived from the "tQH" parameter commonly found in data sheets for the Memory
2. The **Skew Budget** is automatically calculated from the parameters above for both the leading and lagging case, and is intended to show the user how much skew can be introduced by the interconnect while still meeting timing requirements. These are calculated as follows:
   - **(+)** (Skew Budget) = Max Receive Skew (+) – Transmit Skew (+)
   - **(-)** (Skew Budget) = Max Receive Skew (-) – Transmit Skew (-)

3. The **Default**  button, to be used when data sheets or timing specs are not available, populates these fields with typical values by simply allocating one tenth of the Signal Bit Period for Transmit Skew and one quarter of the Signal Bit Period for Receive Skew.

9.7.3.2 Stimulus Offset Options

The available **Stimulus Offset** options vary based on whether the **Controller** or the **Memory** is driving the bus.

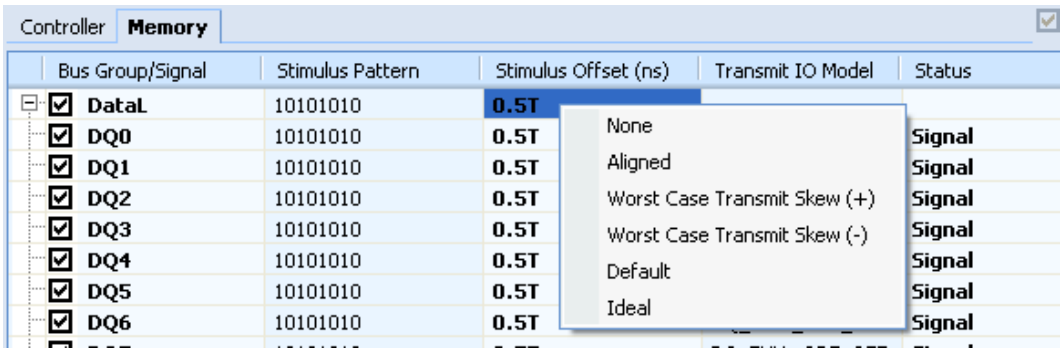
- In cases where the **Controller** is driving the bus, the Stimulus Offset options are listed as the follows:



  - **Default** — The **Timing Reference** signal (ex. strobe) is positioned in the middle of the bus signals' eye. For example, in the case of a **Data** bus, the strobe would be set to lag the data by a quarter clock cycle.
  - **Ideal** — Similar to the **Default** offset, but the buffer delays for the signal are taken into account to make the desired stimulus offset (ex. quarter clock cycle) more exact.

    For a Data Bus group, if the Stimulus Offset is set to either Default or Ideal, following best case timing reports are generated.

    - Timing Report – Best Case Timing
    - Timing Report – Best Case Eye Height
  - **Worst Case Setup** — Stimulus offsets are made to replicate the **Min Transmit Setup** value specified in the **Timing Budget** form, accounting for buffer delays. This should represent the worst case setup condition.
  - **Worst Case Hold** — Stimulus offsets are made to replicate the **Min Transmit Hold** value specified in the **Timing Budget** form, accounting for buffer delays. This should represent the worst case hold condition.
  - **Aligned** — All signals (including **Timing Reference**) are perfectly aligned at the driving component, accounting for buffer delays.
  - **None** — Stimulus offsets are all set to zero.
- If the **Memory** is driving the bus, following Stimulus Offset options are available:



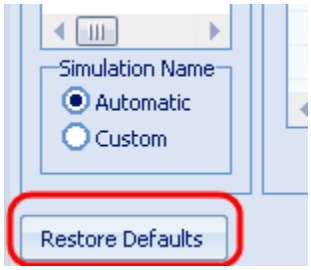
- **None** — Stimulus offsets are all set to zero.
- **Aligned** — All signals (including **Timing Reference**) are perfectly aligned at the driving component, accounting for buffer delays.
- **Worst Case Transmit Skew (+) — Stimulus o**ffsets are made to replicate the **Transmit Skew (+) value specif**ied in the **Timing Budget form, account**ing for buffer delays. This should represent the worst case skew condition, where the data signals lag the strobe.
- **Worst Case Transmit Skew (-)** — Stimulus offsets are made to replicate the **Transmit Skew (-)** value specified in the **Timing Budget** form, accounting for buffer delays. This should represent the worst case skew condition, where the data signals lead the strobe.
- **Default** — Same as above.
- **Ideal** — Same as above.

NOTE!	In the case of a **Data** bus, either can drive, so **Stimulus Offsets** can be defined for the **Controller** on a **Write** simulation, and for the **Memory** on a **Read** simulation.

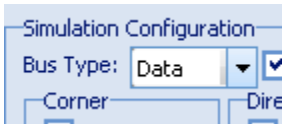
Now we are finally ready to perform some simulations on the bus.

9.8 Restore Defaults

The Restore Defaults button can be used to restore all the changed values in the Analysis Options dialog box to the default values.

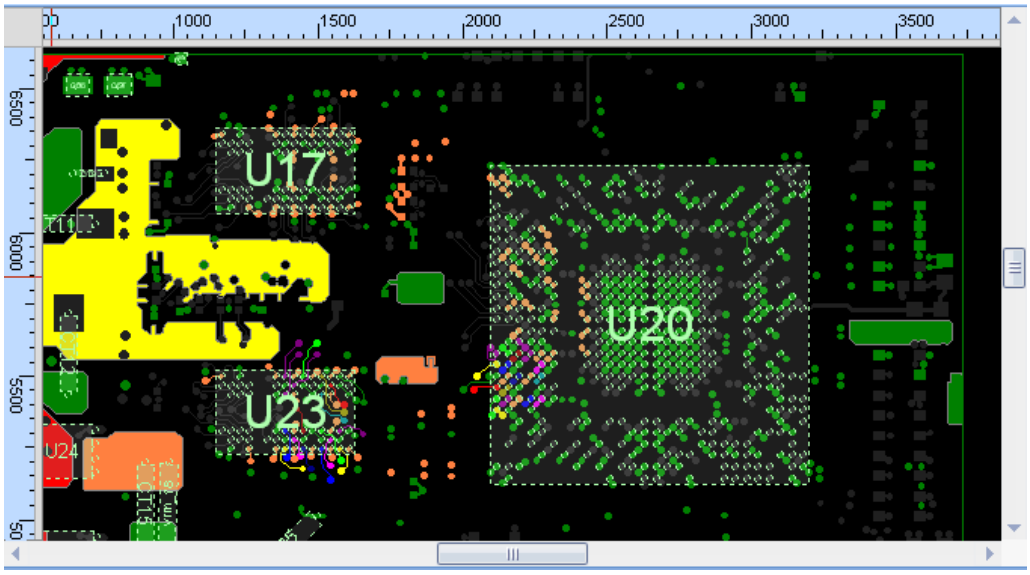


Using Restore Defaults button however, does not change the bus type selected in the Simulation Configuration section.



9.9 Simulation

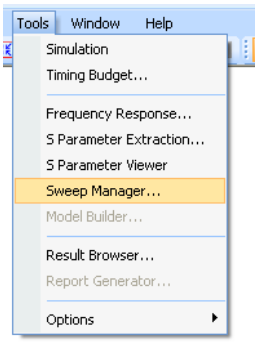
The bus has now been fully defined. Recall that it consists of a Controller (U20), Memory (U23) and VRM on a PCB. The sample looks like this.



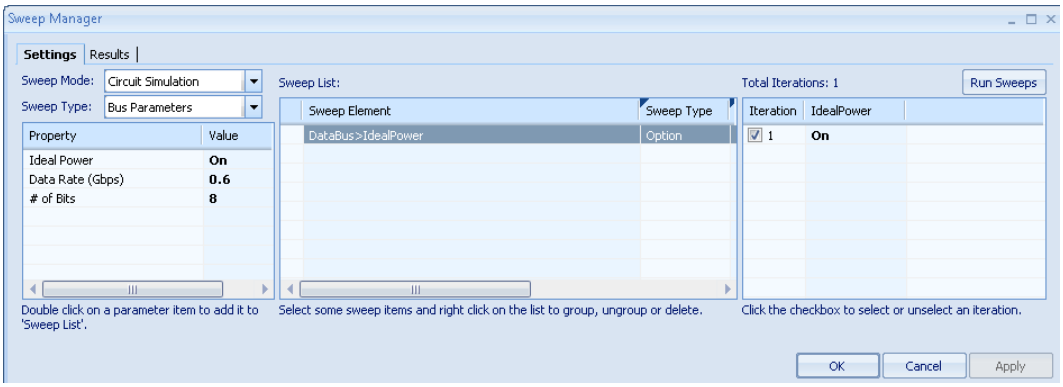
Initially, we will use **Ideal Power**, ignoring the PDN and VRM in this configuration.

9.9.1 Explore Sweep Manager

Select **Tools — Sweep Manager….**



The **Sweep Manager** interface opens.



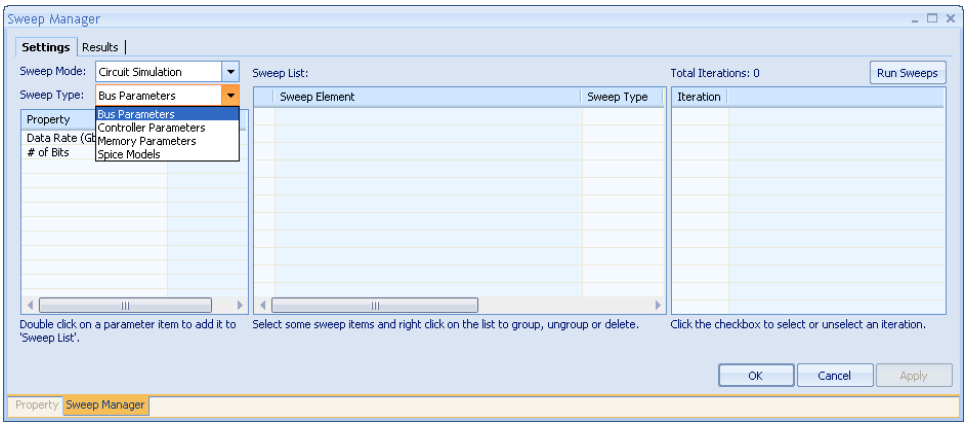
The **Sweep Manager** interface has three sections:

- Adding Parameter
- Sweep List
- Total Iterations

9.9.1.1 Sweep Type

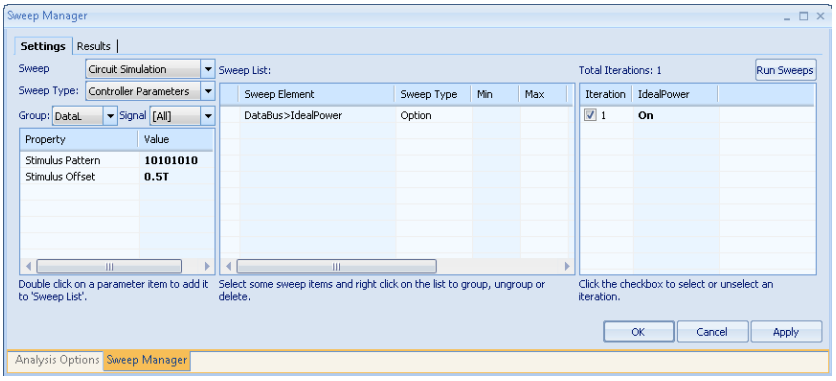
There are four Sweep Types:

- Bus Parameters(selected by default)
- Controller Parameters
- Memory Parameters
- Spice Model



Depending on the **Sweep Type** chosen, you can double-click a parameter to add to the **Sweep List**.

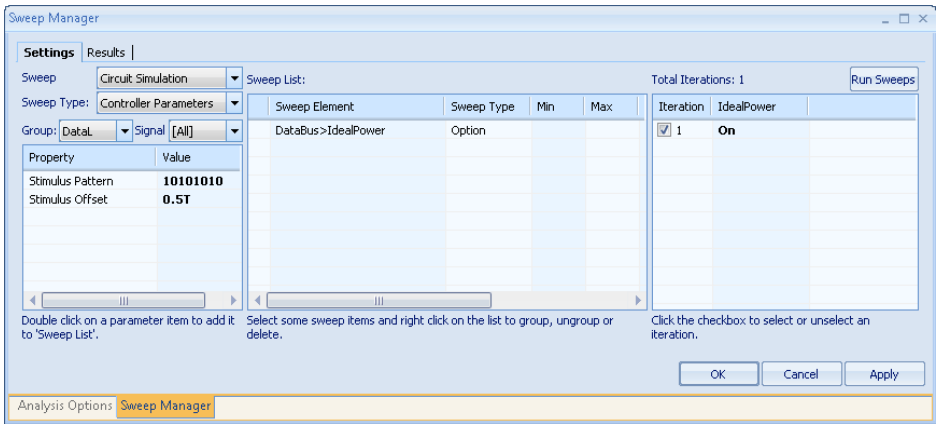
In the following example, the sweep name appears in the **Sweep List** if you double-click **Stimulus Offset**.



9.9.1.2 Sweep List

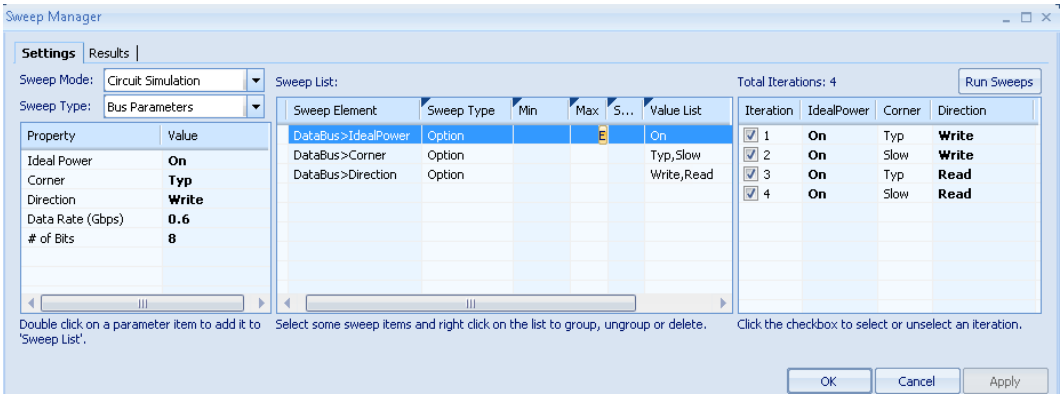
1. To choose parameter values, select the Sweep item from the Sweep List and go to the next section. The **Sweep List** pane contains these items:

   - **Sweep Element** - Identify the Sweep Type and parameter or model selected.
   - **Sweep Type** - Choose from the drop-down menu.
   - **Step Count** - Number of different values for each parameter.
   - **Min**, **Max**, **Step**, and **Value List** - Related element values.



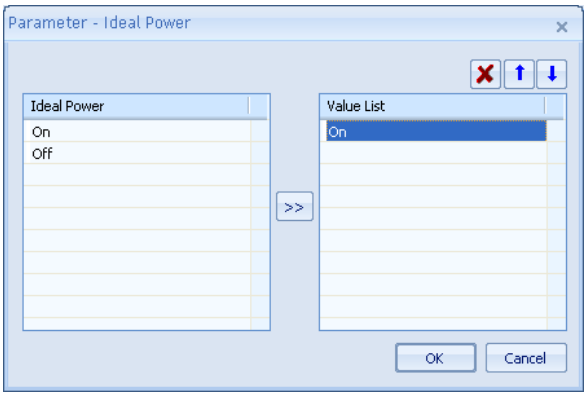
2. Sweep of the **Corner**, **Direction** and **Rank** can only be defined from the **Analysis Options** for sweep.

   For projects with VRM blocks, the **IdealPower** is added to the **Sweep List** automatically.



3. Double-click a box under the **Value List** column (**IdealPower** selected in this example).

   The **Parameter** editing window opens.



   You can:

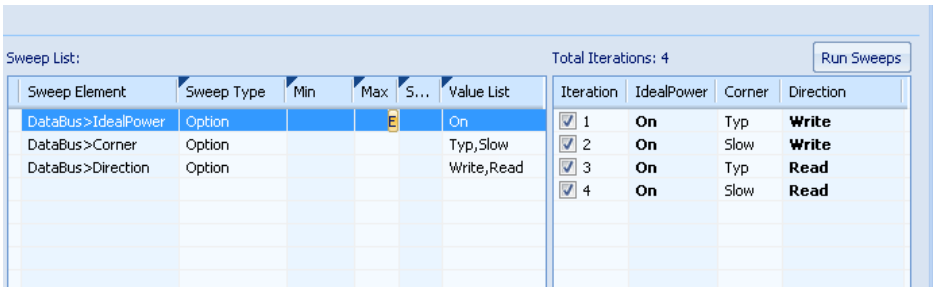
   - Select a value in the **Ideal Power** column and click >> to add it to the **Value List** column.
   - Use the ✘ button to delete the selected value.
   - Use the ↑ or ↓ button to move up or down the value.

4. Click OK to close the window without making any modifications.

9.9.1.3 Total Iterations

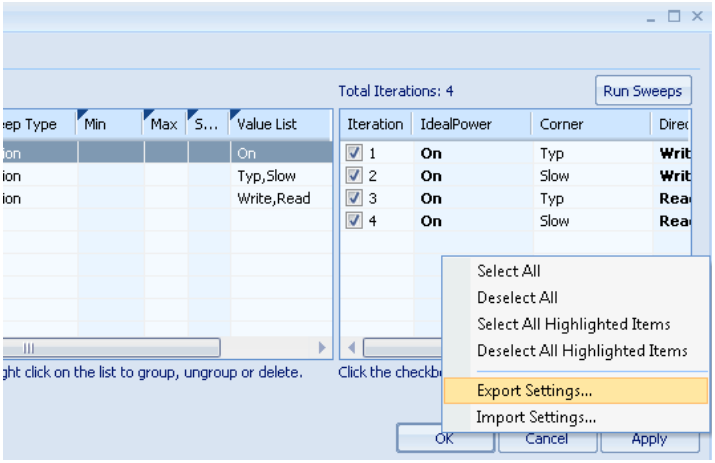
Total Iterations are the total number of simulations to be performed.

In the following example, 4 Iterations will be performed.



1. Right-click on the **Total Iterations** spreadsheet.

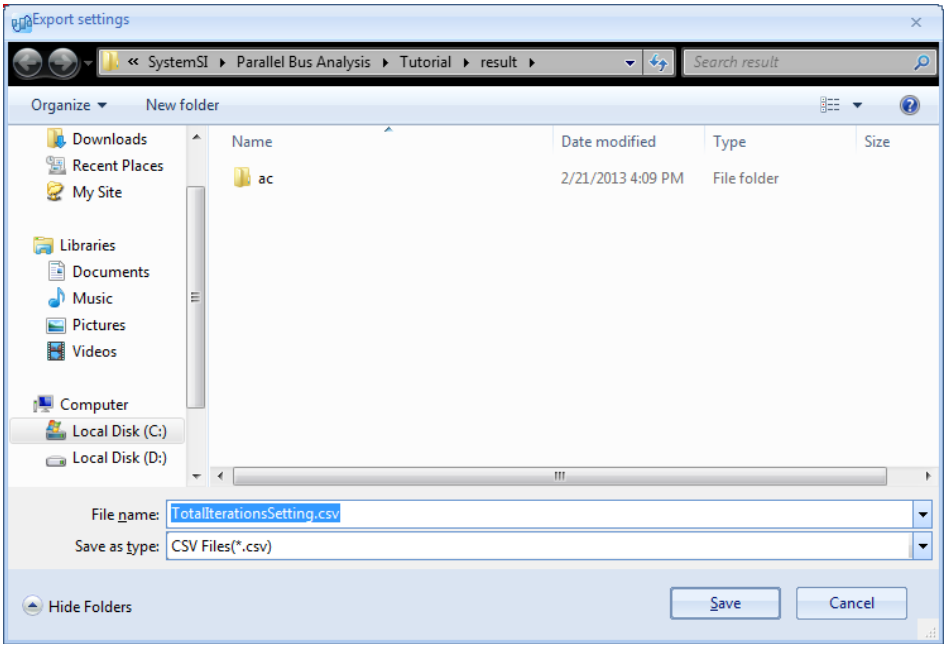
   Two options **Export Settings...** and **Import Settings...** are available in the pop-up menu list.



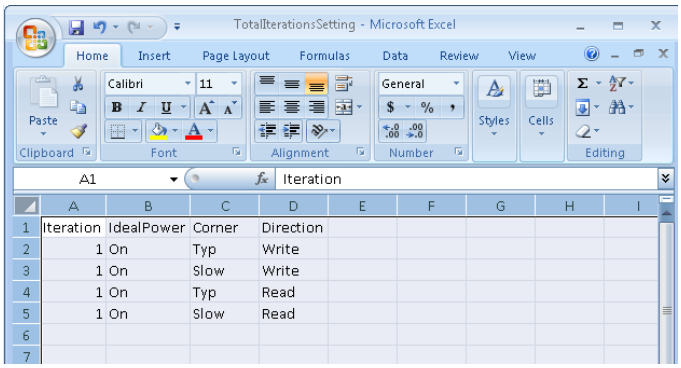
   They are used to export and import the Iteration settings in the csv file.

2. Click **Export Settings...**.

   The **Export settings** window opens.



3. Click **Save** to save a csv file listing all the possible iterations.
4. Open the saved csv file.

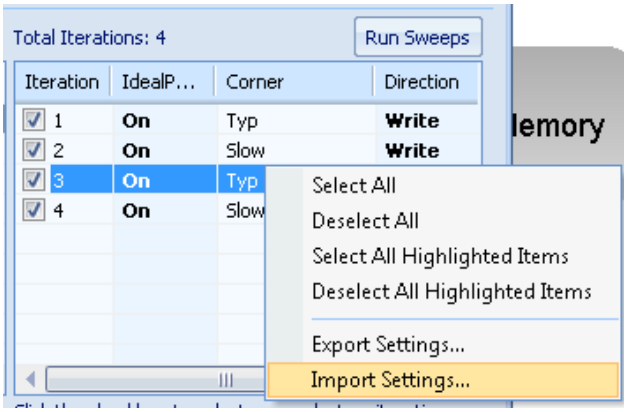


- The number **1** in the **Iteration** column indicates this iteration is enabled in the **Total Iterations** spreadsheet
- The number **0** in the **Iteration** column indicates this iteration is disabled in the **Total Iterations** spreadsheet

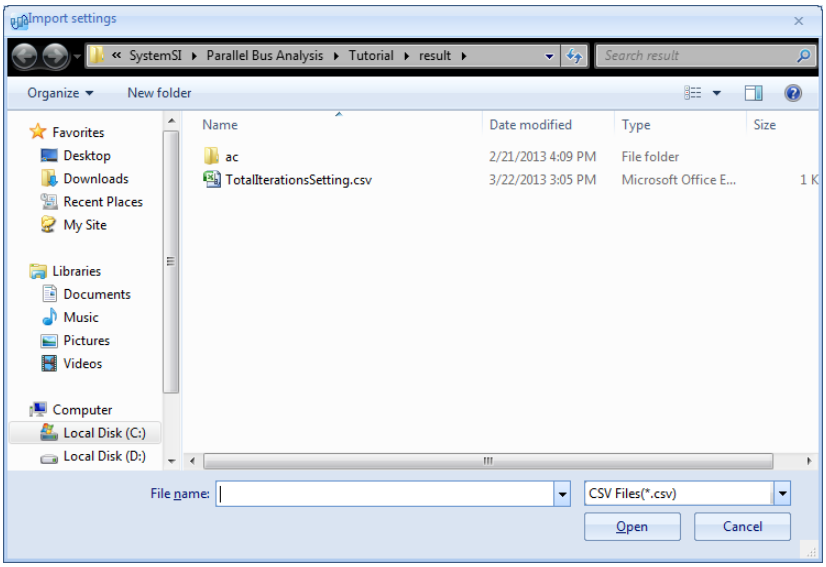
5. Change the **Iteration** value of the first row to **0** as follows, and save the file.

	A	B	C	D
1	Iteration	IdealPower	Corner	Direction
2	0	On	Typ	Write
3	1	On	Slow	Write
4	1	On	Typ	Read
5	1	On	Slow	Read
6				

6. Click **Import Settings...** in the **Sweep Manager** window.

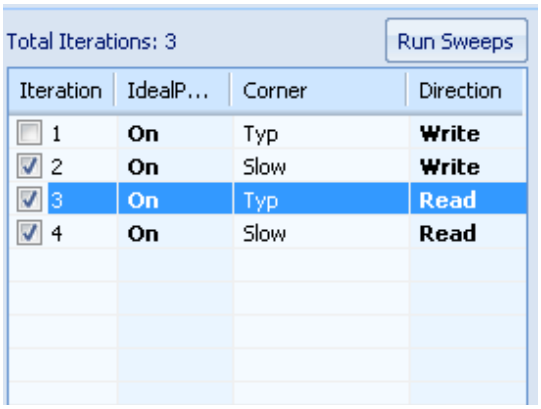


The **Import settings** window opens.



7. Choose the saved csv file, and click **Open**.

   The settings of iterations in the csv file are shown in the **Total Iterations** spreadsheet.

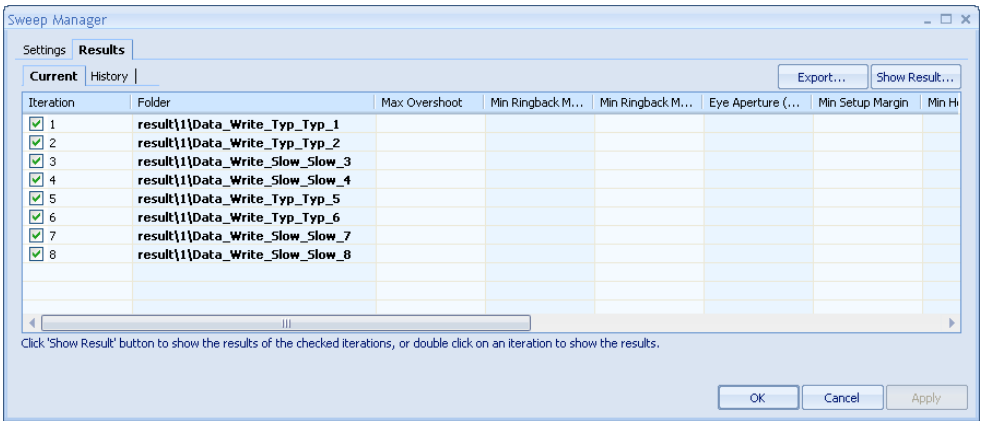


   The first row of **Iteration** is disabled.

NOTE!	The csv file to be imported must match the content of the Total Iterations spreadsheet; otherwise an error message will display to show its failure.

9.9.1.4 Results

- After all simulations are done, click the **Results** tab to display the results.



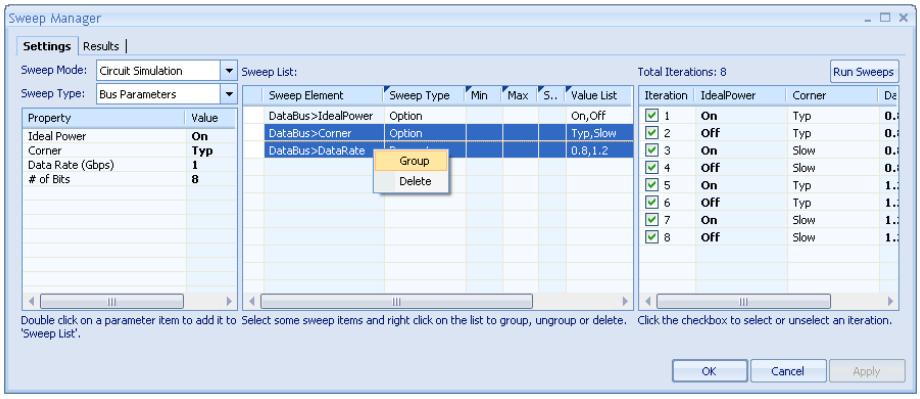
- To view curves such as Eye contour and Bathtub curves, check one or more iterations and click Show Result....

9.9.1.5 Grouping Parameters

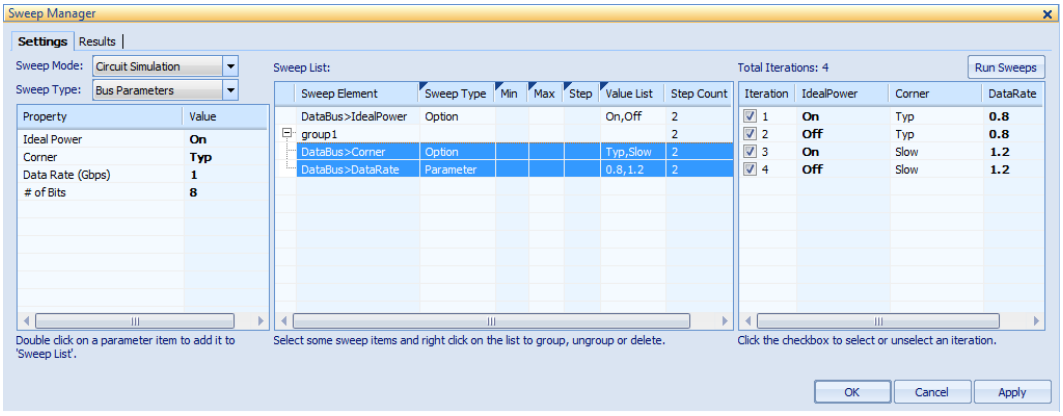
You can group two or more parameters to reduce the number of iterations.

For example, the **Corner** and **Data Rate** can be grouped together.

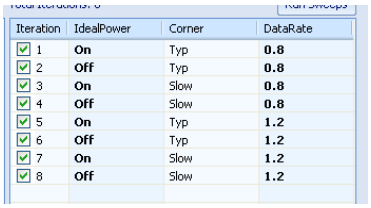
1. Select two rows (**Corner** and **Data Rate** in this example).
2. Right-click the selected rows and select **Group** in the pop-up menu.



The number of **Total Iterations** drops from 8 to 4.

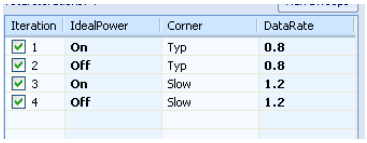


Originally, the iterations look like this:



Iteration	IdealPower	Corner	DataRate
1	On	Typ	0.8
2	Off	Typ	0.8
3	On	Slow	0.8
4	Off	Slow	0.8
5	On	Typ	1.2
6	Off	Typ	1.2
7	On	Slow	1.2
8	Off	Slow	1.2

After grouping the **Corner** and **Data Rate**, the new iterations are:



Iteration	IdealPower	Corner	DataRate
1	On	Typ	0.8
2	Off	Typ	0.8
3	On	Slow	1.2
4	Off	Slow	1.2

When combining two or more parameters in a group:

- The first value for each parameter in the group constitutes one combination.
- The second value for each parameter constitutes the second iteration.

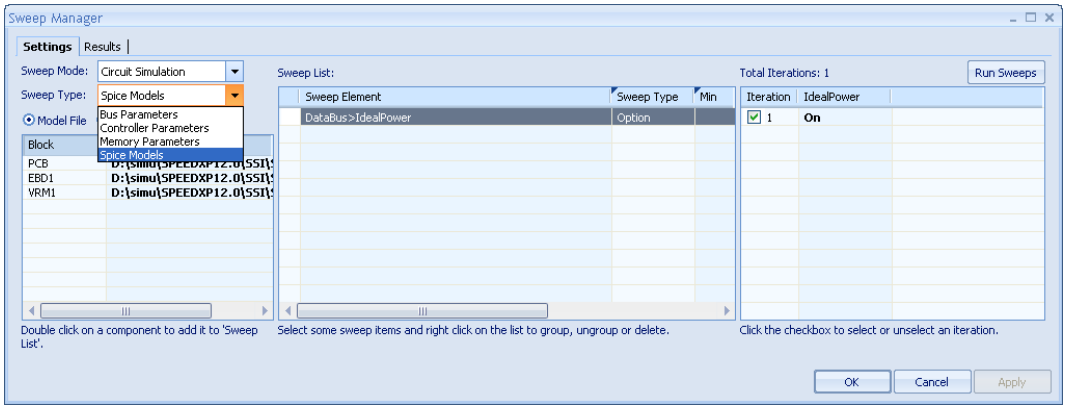
……

NOTE!	Within a group, the value in the **Step Count** column of each parameter must be identical. Or else, the **Total Iterations** of the group will be **0**.

9.9.1.6 Model Sweep

The **Sweep Manager** supports different types of Model Sweep.

Use the Sweep Type drop-down list to select to type of sweep. For this tutorial, select Spice Models.



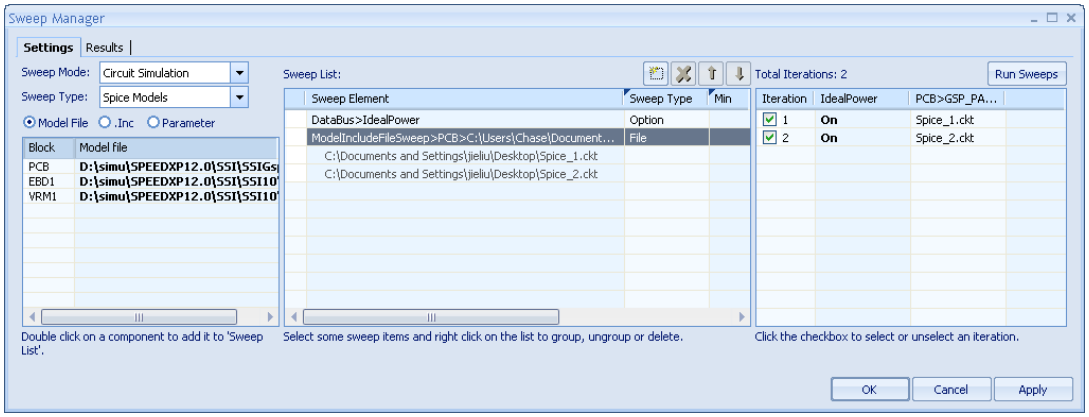
For Spice Models, there are 3 kinds of Model Sweep:

- **Model File** sweep - Sweep different **.sp** or **.ckt** files defined in the blocks (such as the **EBD1** block and **VRM1** block).
- **.Inc** sweep - Sweep multiple circuit models if the original model is called within an **.include** statement (such as the **PCB** block including three **.include** statements).
- **Parameter** sweep – Sweep different parameters defined in the **.sp** file (such as **Voltage** in the **VRM** block).

Model File Sweep

When using the **Model File** sweep, make sure that the connectivity inside each **.sp** file is the same to maintain everything between `*[MCP Begin]` and `*[MCP End]`.

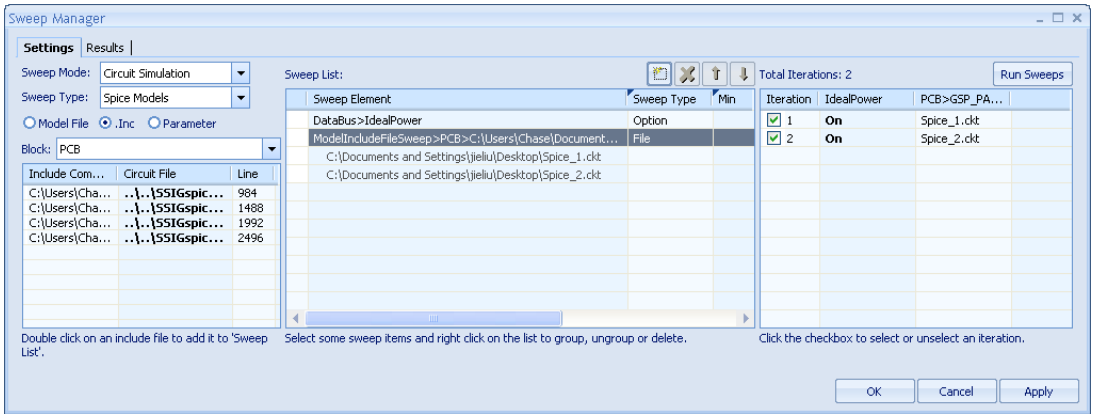
Click the **New** button  to add model files.



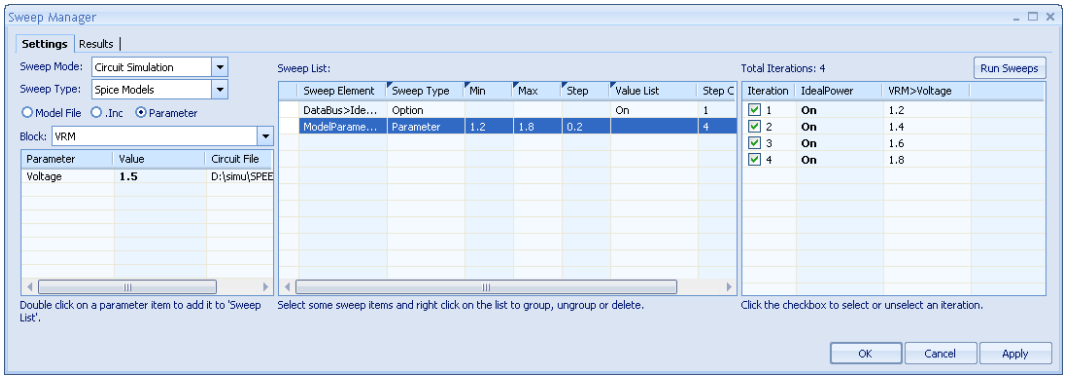
.Inc Sweep

When using **.Inc** sweep, make sure that the sub-circuits inside **.ckt** or **.sp** files all have the same name.

Click the **New** button  to add circuit files.



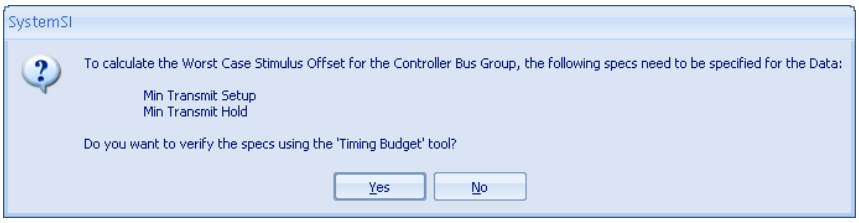
Parameter Sweep



9.9.2 Initial Simulation

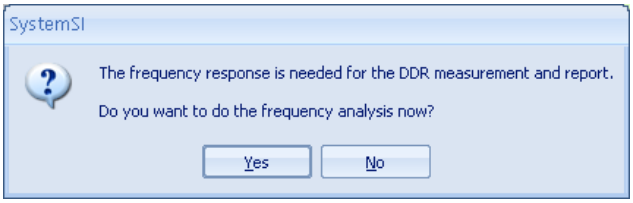
1. Click the **Play** button  .

   The warning message dialog pops up again.



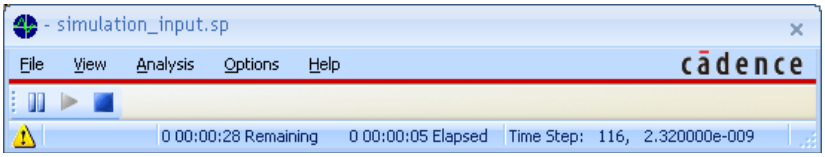
2. Click **No**.

   The frequency response reminding message dialog pops up.

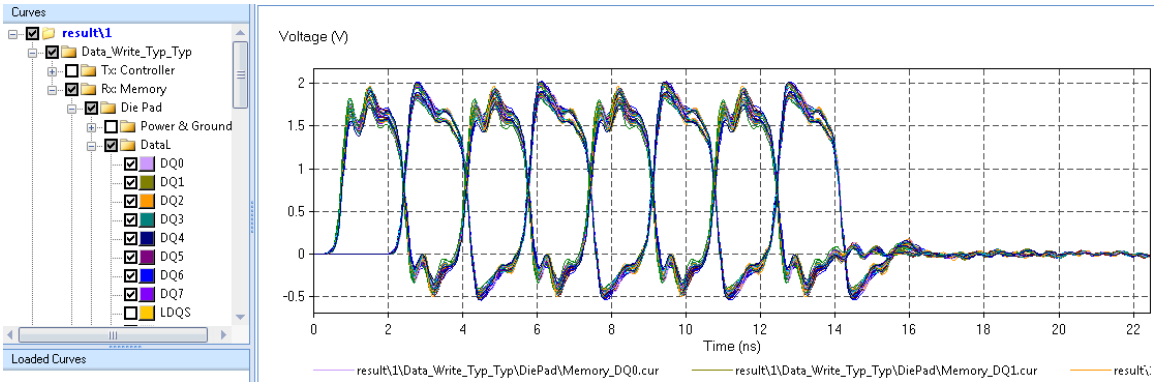


3. Click **No**. (If you want to do the frequency analysis, please refer to ***Section 9.8.6 Frequency Response***.)

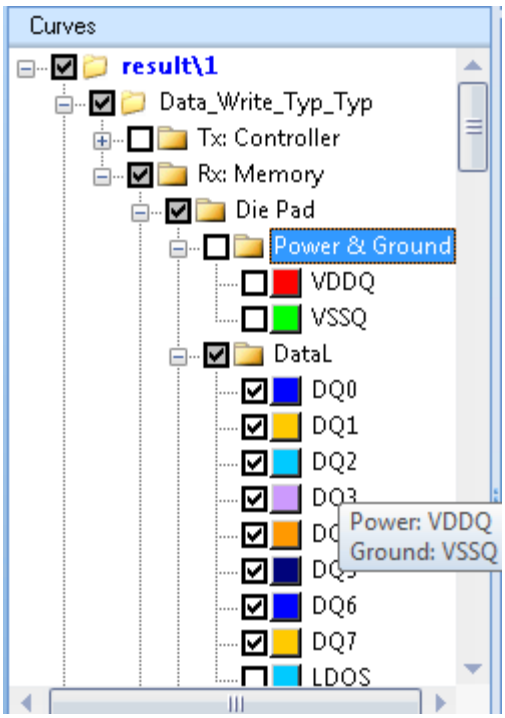
   The simulation starts, showing a **simulation_input.sp** window.



   At the conclusion of the simulation, waveforms are plotted in a 2D Curves (Time Variation) window.



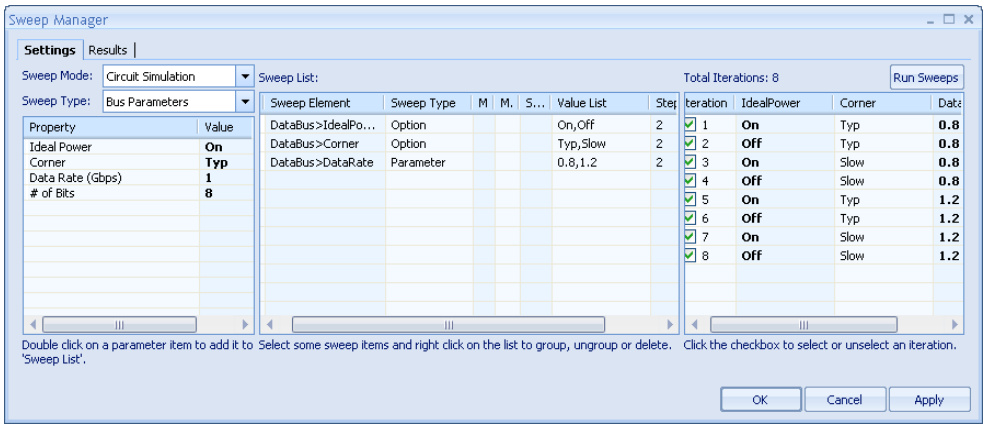
   The powers and grounds of the Controller and Memory blocks will be displayed, and each signal will have a tip to show its power and ground.



9.9.3 Running Multiple Sweep

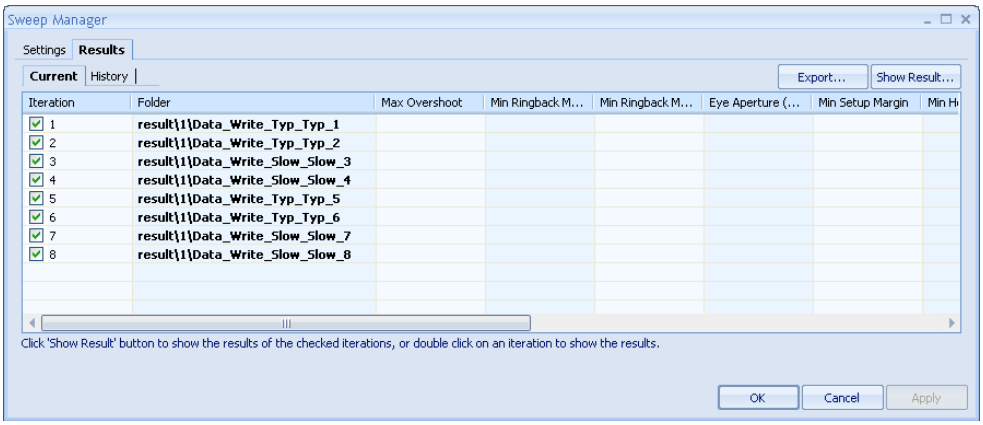
This section describes how to sweep multiple parameters and across capability. The following parameters and values are to be set.

- **IdealPower**: On, Off
- **Corner**: Typ, Slow
- **Data Rate**: 0.8, 1.2



Click Run Sweeps to begin the sweep simulation.

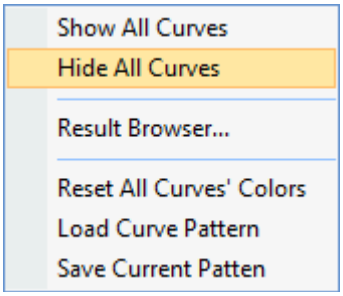
After the simulations are run, the results appear in the **Results** pane. The results from the first sweep are now found under the **History** tab.



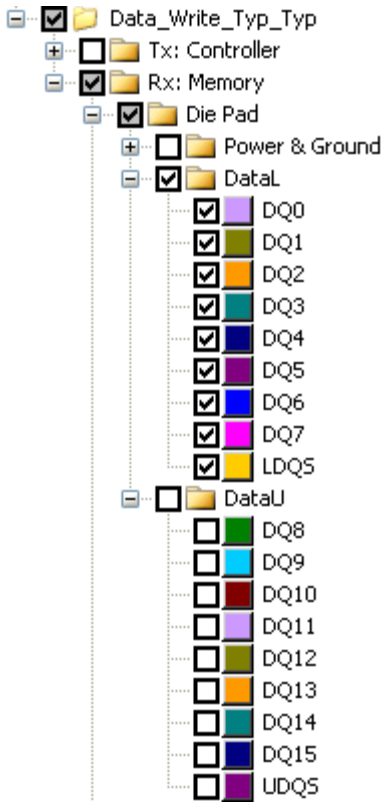
9.9.4 Data Presentation / 2D Curve Viewer

Data for each signal in the bus can be individually selected and viewed as either a raw waveform, or Eye diagram.

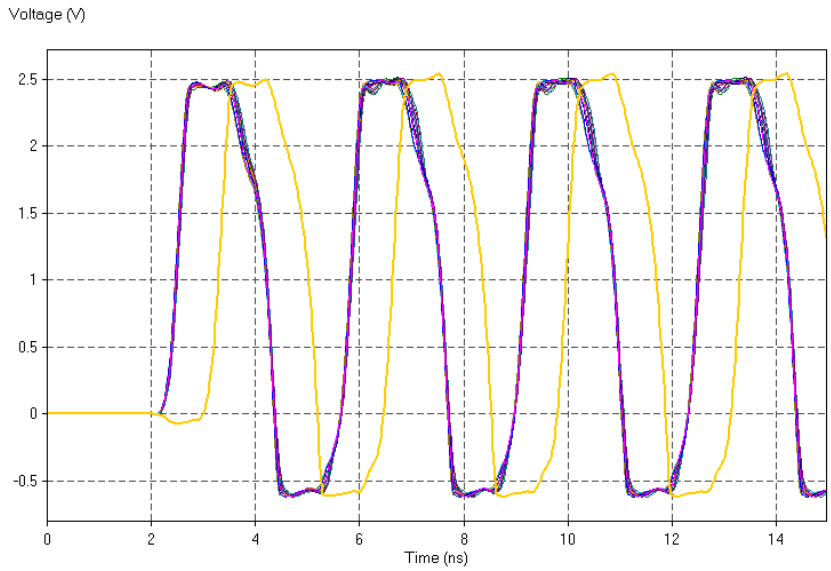
1. In the 2D Curves Window the Left Pane lists simulation results including each signal in our two Data buses. Turn these off by right-click and selecting **Hide All Curves**.



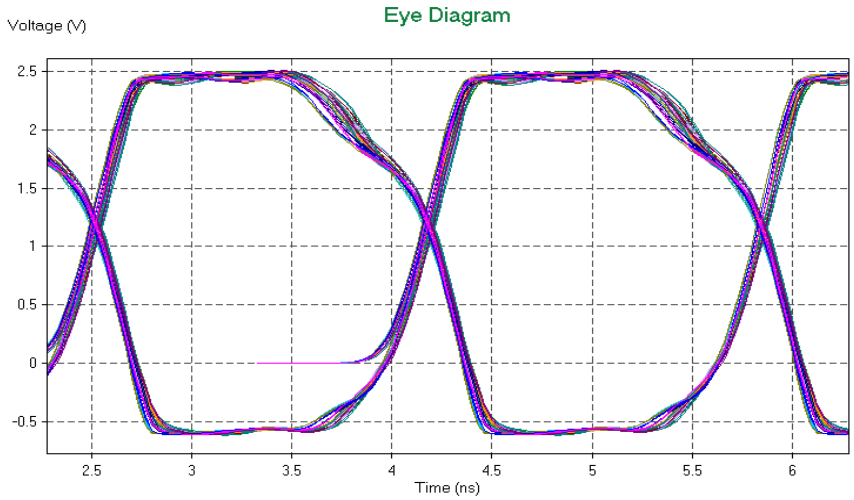
2. Then select only the **DataL** signals.



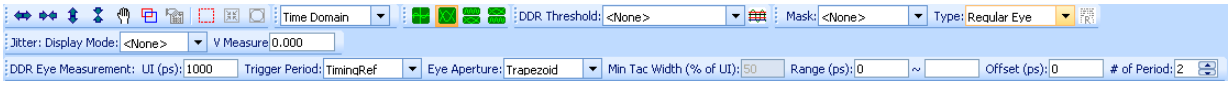
We see the data plotted.



3. Now turn the Strobe trace off, (LDQS).
4. Switch to an Eye diagram plot by selecting the Icon from the Toolbar.

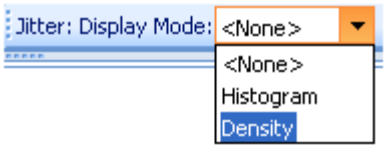


5. With the Eye Diagram visible, the toolbar now has additional measurement and data processing functions available.

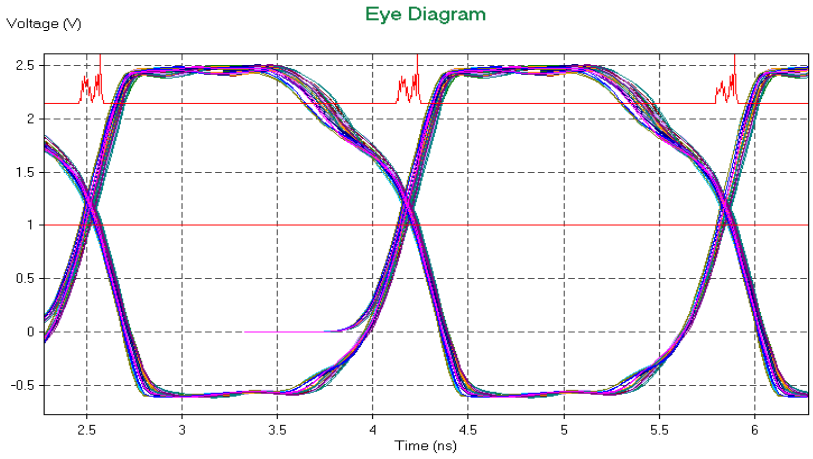


We can specify an Eye Contour or Eye Density plot, overlay an Eye Mask, or measure Eye Opening including Trigger Period, Eye Aperture, Min Tac Width, Offset And No. of Period. We can also look at Jitter.

6. Set **VMeasure** to **1V**, which is the center of the voltage swing.
7. Select **Density** as **Jitter: Display Mode** to overlay a Jitter estimate on the Eye Diagram.



We get a plot with Jitter Density displayed on the screen.



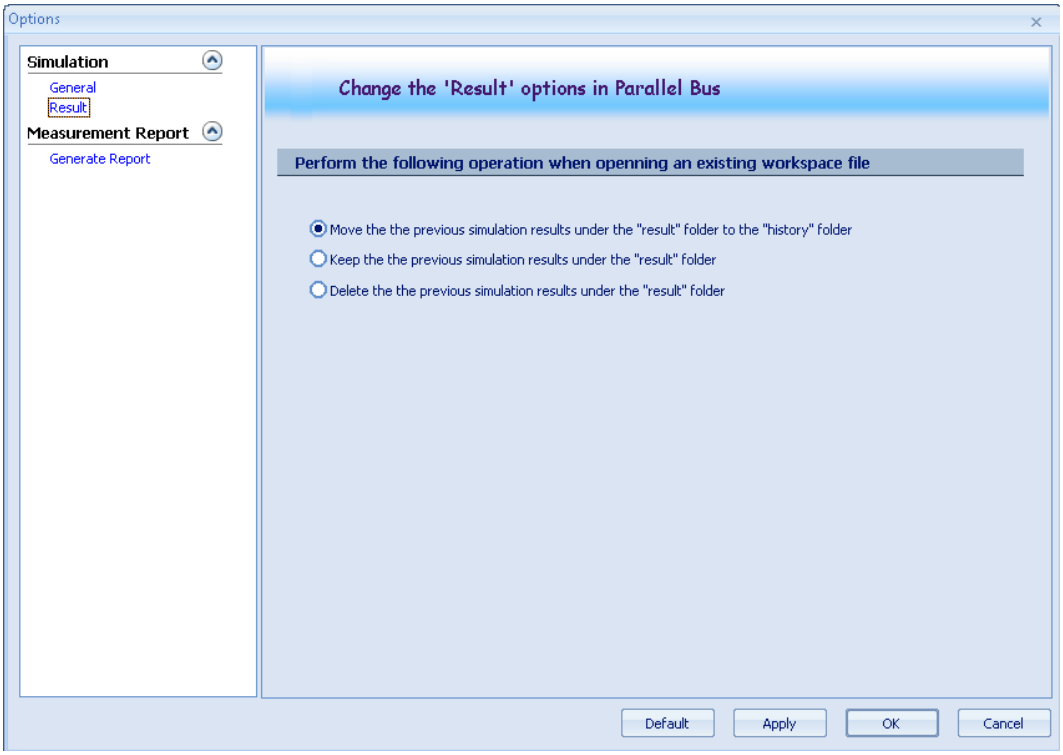
NOTE!	As multiple simulations are run, results are kept in SystemSI and will cause the data curves to add up quickly. (1) Use the "+" and "-" button to minimize and maximize signals in the bus. (2) Right-click to "Hide All Curves" and select

	only the ones that are desired. (3) Close the Curve window to clear the data and start over in a new Curve window if needed.

9.9.5 Simulation Directory

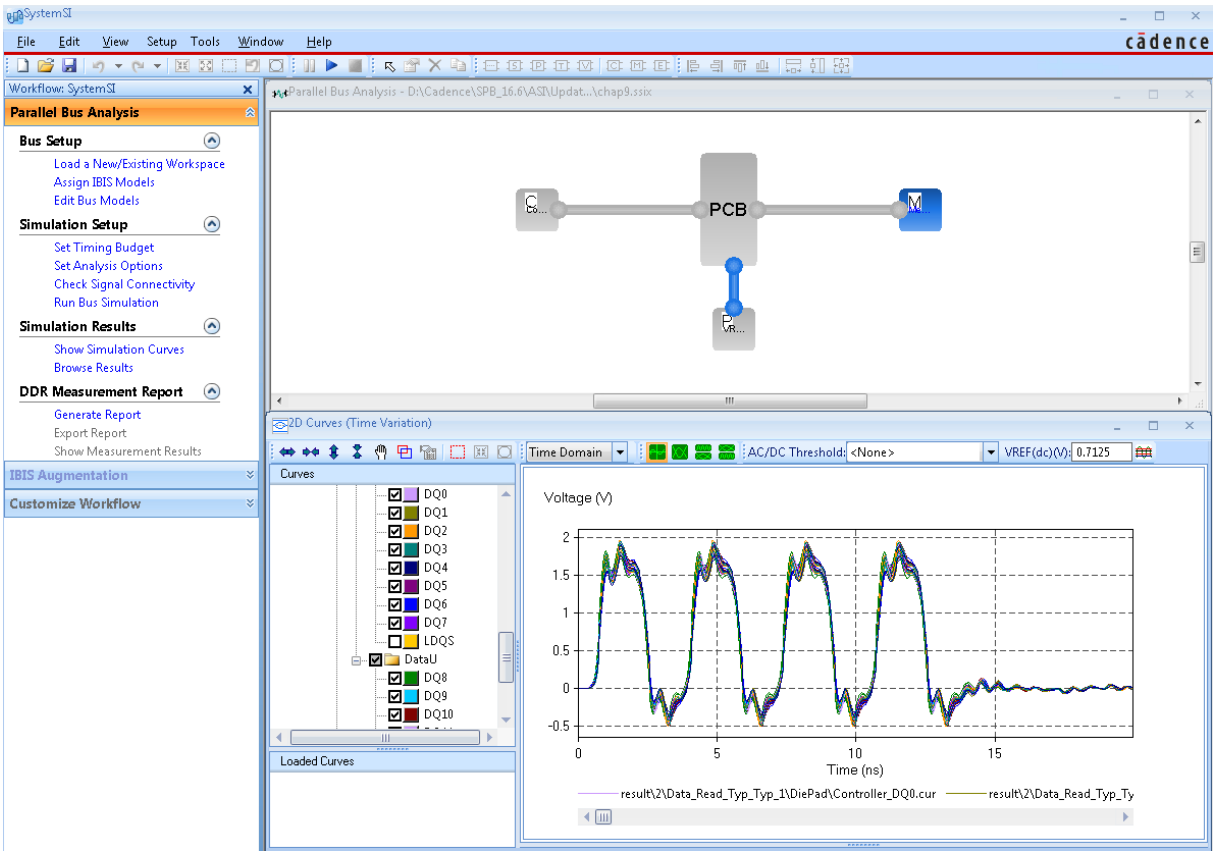
Each simulation creates a unique directory (1, 2, 3 …etc) with all the waveform files under: **`\<case name>\result`**.

By default, these directories are moved to the **history** folder when opening an existing workspace file. However, you can make the following setup by choosing **Tools > Options > Edit Options…**, and clicking **Result** under **Simulation**.



The first one is checked by default.

The following figure shows the waveforms generated by the bus engine. This illustration is the default display.



To show the waveforms, choose

Window > Tile Horizontally/Vertically

or

Window > Cascade

On the **Windows** menu, click **To Default** to arrange all open windows to the default display.

Basically, the `.ssix`, `.html` and `.txt` windows should be put together at the top, and other curve windows should be put together at the bottom.

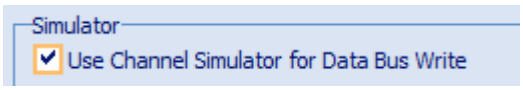
9.10 Additional Exercises

A handful of simple experiments are performed on this sample, to introduce the user to the power of SystemSI - Parallel Bus Analysis.

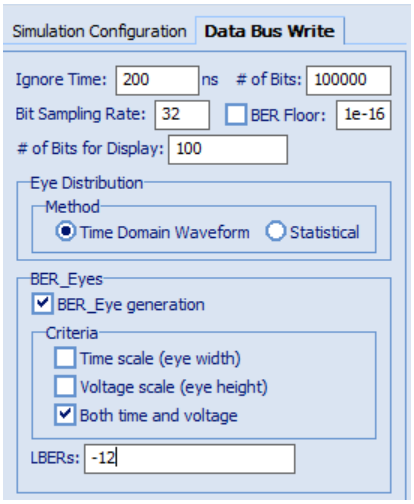
9.10.1 BER Analysis for DDR4 Interfaces

Using SystemSI – Parallel Bus Analysis, you can perform channel simulations for DDR4 Data Bus Write operations.

1. To enable channel simulations, select the *Use Channel Simulation for Data Bus Write* check box in the Analysis Options dialog box.

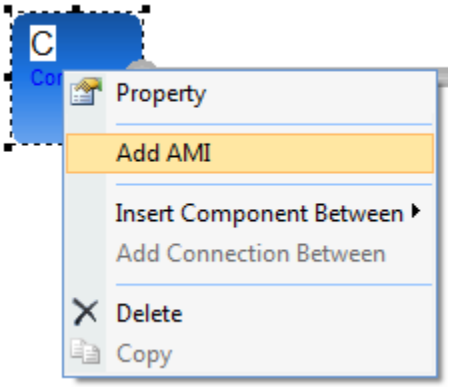


2. Specify the values of channel simulation options that are enabled in the Analysis Options dialog box, and click *Apply*.



3. After you perform Step 2, following options are enabled.

   a) Option to add AMI model to Controller block.

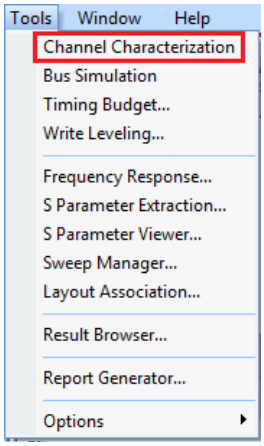


   The AMI model added to the Controller block is applied to all Data signals. Strobe signals are excluded.

   b) Ability to define jitter and noise injection parameters for the strobe and data signals.

Jitter and noise for strobe signals is defined in the *Strobe Jitter and Noise* tab of the Property dialog box for the Controller block. To specify jitter and noise for the Data signals, use the *Data Jitter and Noise* tab, visible in the Property dialog box for the Memory block.

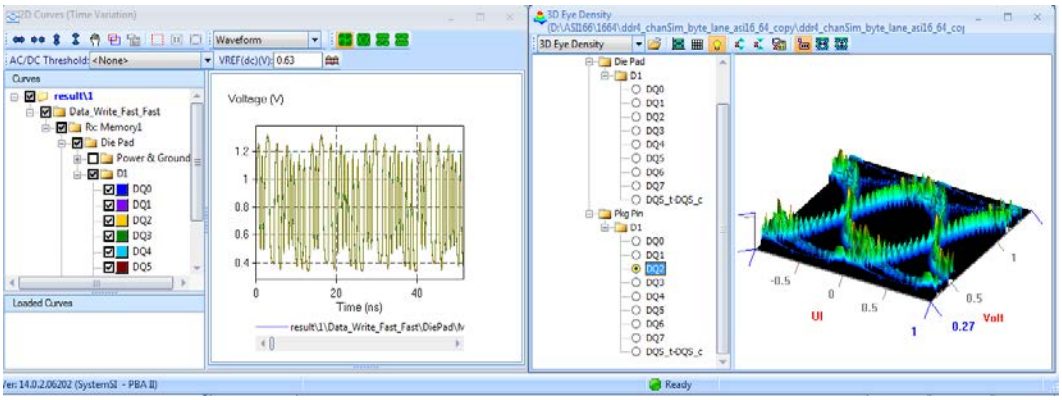
c) The *Channel Characterization* option is visible in the *Tools* menu.



Before you simulate the design, you can characterize the channel to ensure all design modifications are accounted for.

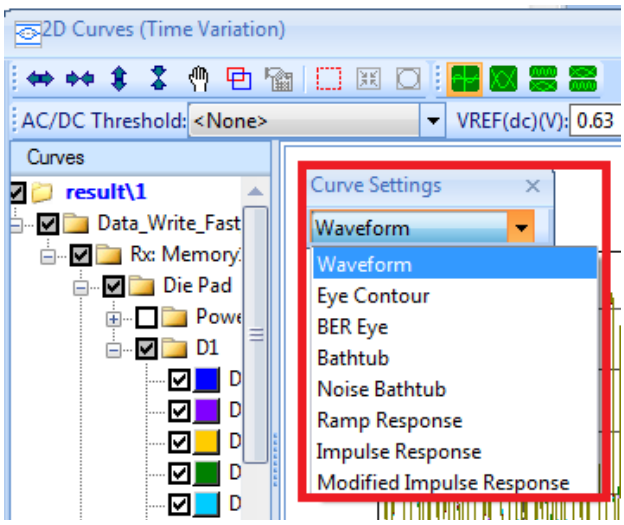
4. Simulate the design.

Channel simulation results are displayed in the 2D Curves and 3D Eye Density windows.



In addition to raw waveforms, channel simulation also generates additional outputs, such as Ramp and impulse responses, Eye contours for the entire high capacity simulation, Jitter and noise bathtub curves, to predict BER performance, and BER_Eyes (if specified in the Analysis Options form), for review.

To view these, select appropriate option from the drop-down list in the *Curve Settings* toolbar of the *2D Curves (Time Variation)* window.



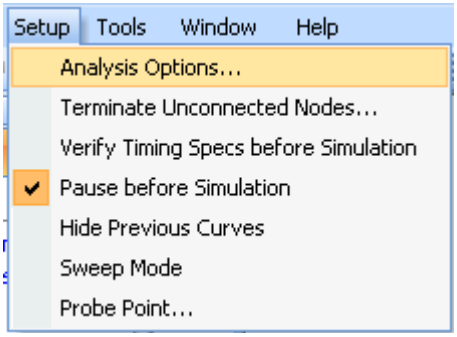
5. Post simulation, you can also generate simulation reports.

9.10.2 Crosstalk

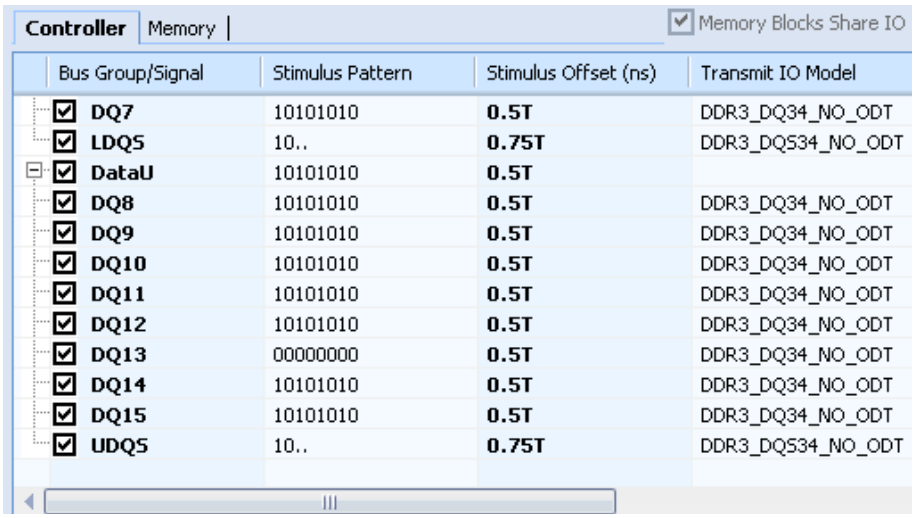
Simulate the crosstalk on a victim line in the bus.

Turn off one of the DQ lines by editing the Stimulus pattern, then re-simulate and see the coupled noise voltage on the victim line.

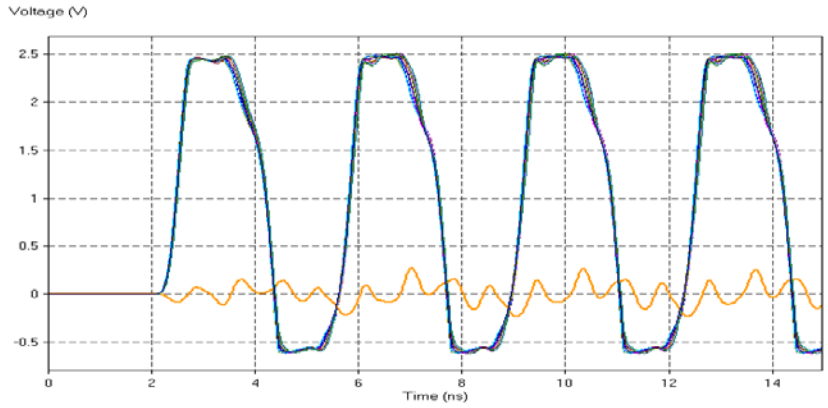
1. Open the **Analysis Options** by selecting Setup —Analysis Options.



2. For the **DataU Bus**, scroll to the **DQ13** signal, and change the pattern to Zero as shown below.



3. Click **Apply** to accept the changes.
4. Re-simulate, to see the noise voltage on the **DQ13** victim line.

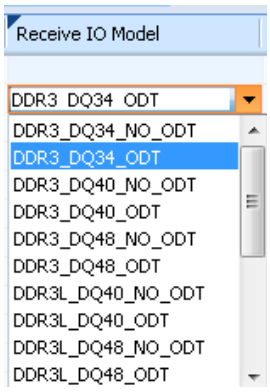


9.10.3 Change IO Models

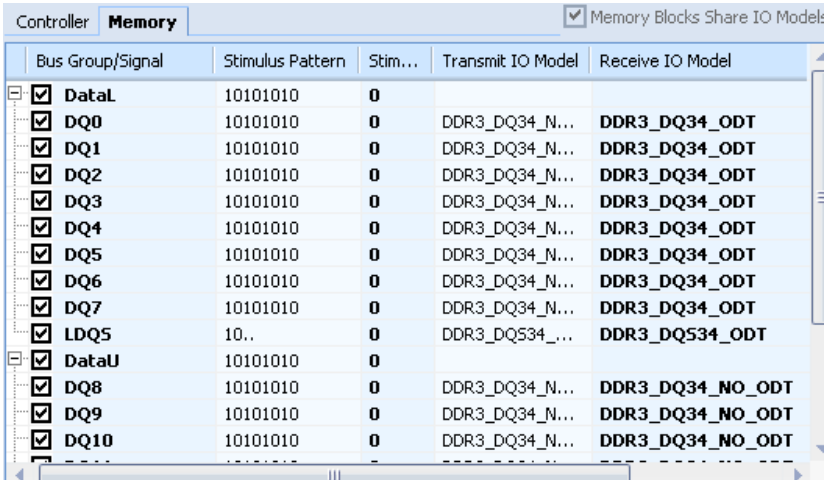
The IO models can be changed with a simple mouse click and the bus re-simulated to compare performance. In this exercise, compare performance of the IBIS corner models. Also, changing the Controller and Memory Transmit/ Receive IO models is simple.

1. Open the **Analysis Options** by selecting Setup > Analysis Options.
2. Select the Memory tab.
3. Change the Receive IO model for the DataL Bus, to **DDR3_DQ34_ODT.**

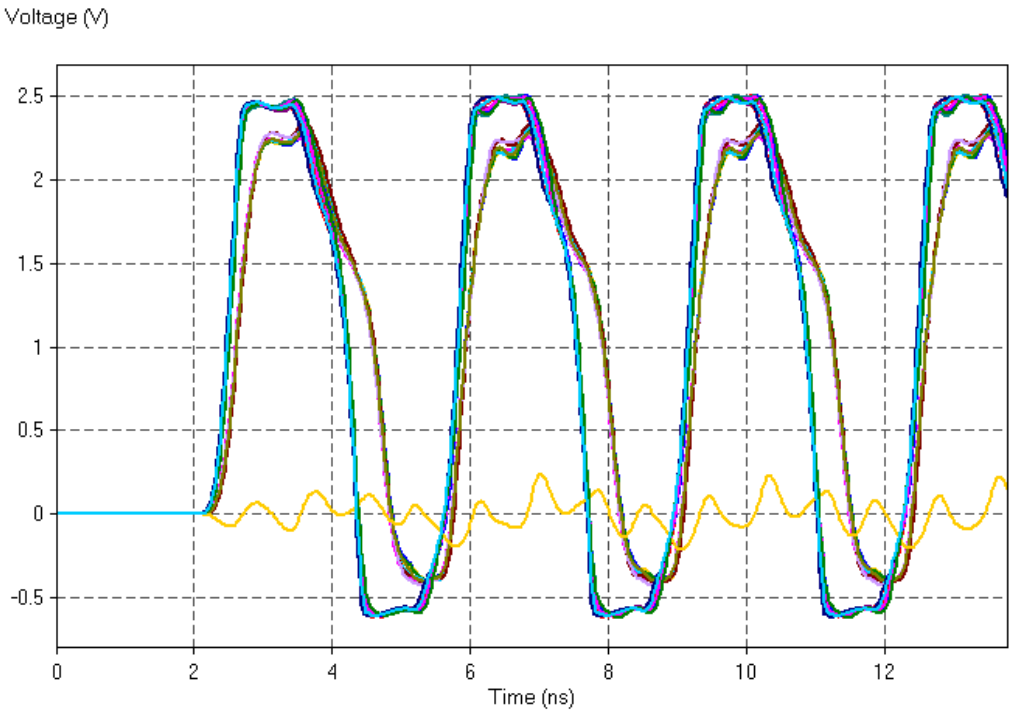
   Changing just one of the signals on the bus will update the model for all lines in the DataL bus.



4. We now have one data bus using a **low-voltage** driver, and one using a **higher voltage** one.



The data shows difference in amplitude and ripple for the two cases.



5. Turn the DQ13 signal back on, and switch back to the Max Drive Transmit model.

☑ MEMC_MDQ13	10101010	0.5T	ddr2_maxdrv_io

9.10.4 Non-Ideal Power

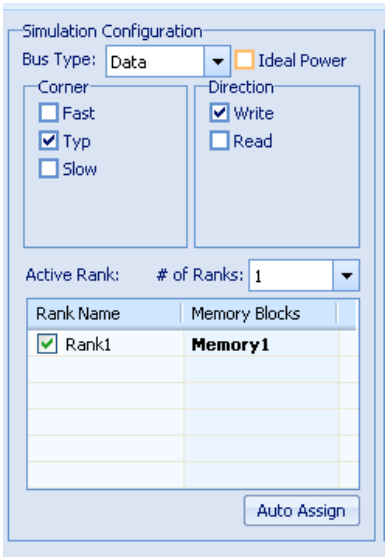
Turn off **Ideal Power** in the Analysis Options. This will effectively "turn on" the 1.8V VCC connections on the PCB that were generated in the PowerSI extraction (Chapter 6), as well as the VRM and their references that are impacted in the Controller and Memory.

NOTE!	If we had defined the devices with HSPICE models, or IBIS models that included the composite pre-driver current (BIRD95), these would also be included in this Non-Ideal Power simulation.

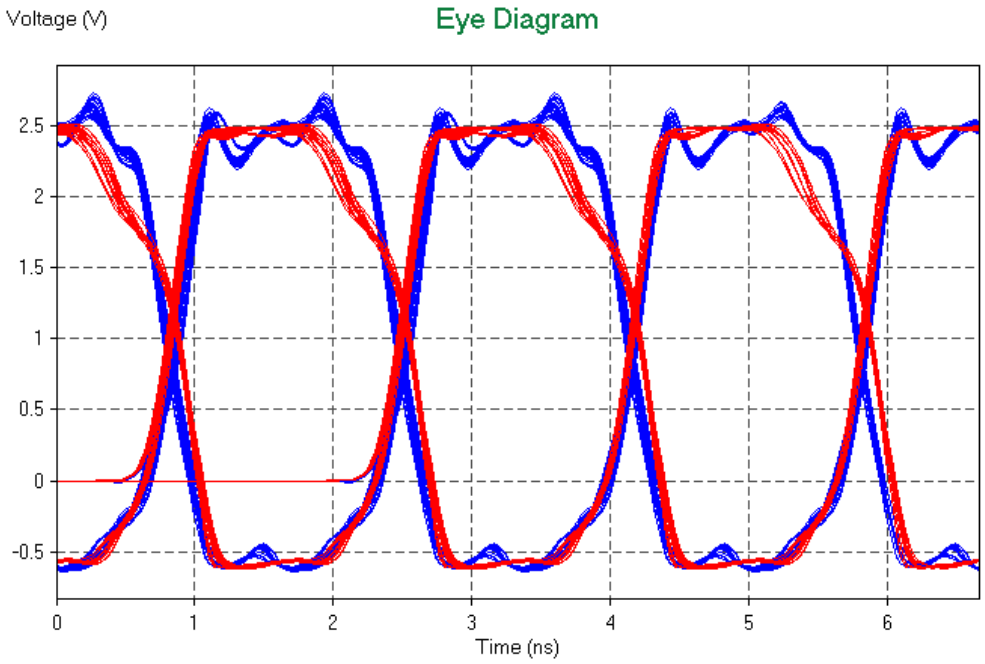
1. To open the **Analysis Options,** select

   Setup > Analysis Options…

2. Turn off **Ideal Power**.



3. Re-simulate and compare results. To simplify the comparison, we plot ONLY the DQ3 signal, with Ideal Power (Red) and Non-ideal Power (Blue) in an Eye Diagram.



We see a big difference, especially in the sharpness of the rising edge in the two waveforms.

More simulations are possible. Please consult with your Sigrity representative for information about SystemSI - Parallel Bus Analysis.

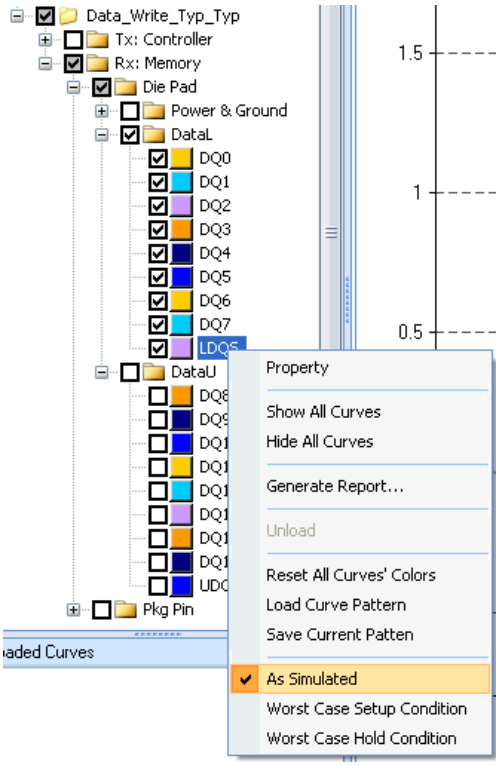
9.10.5 Worst Case Timing

9.10.5.1 Memory Bus in Write Mode

If you set **Stimulus Offset** mode as **Default** in **Analysis Options**, and define worst case timing parameters in **Timing Budget**, Parallel Bus Analysis can simulate both default mode and worst case condition automatically.

1. Enable **LDQS** in **Rx** side and right-click it.

   The pop-up menu lists the Worst Case options.

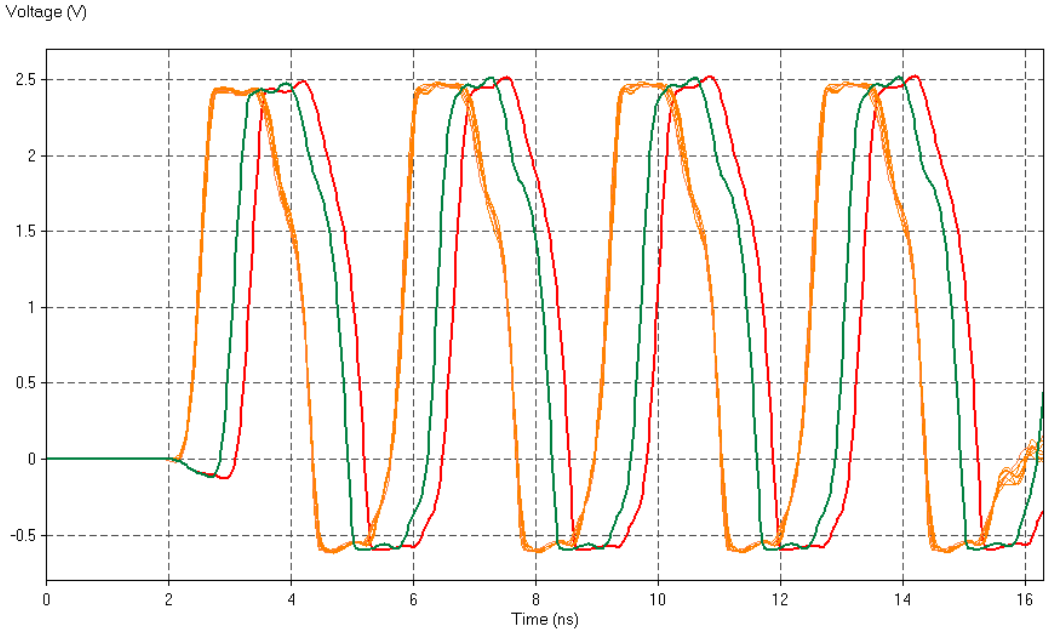


For a Memory bus group in **Write** mode, three options are available:

- **As Simulated** –Shows DQS waveform with Stimulus Offset per the Analysis Options form.
- **Worst Case Setup Condition** - Shows DQS waveform with Stimulus Offset calculated from the Min Transmit Setup parameter in the Timing Budget form.
- **Worst Case Hold Condition** - Shows DQS waveform with Stimulus Offset calculated from the Min Transmit Hold parameter in the Timing Budget form.

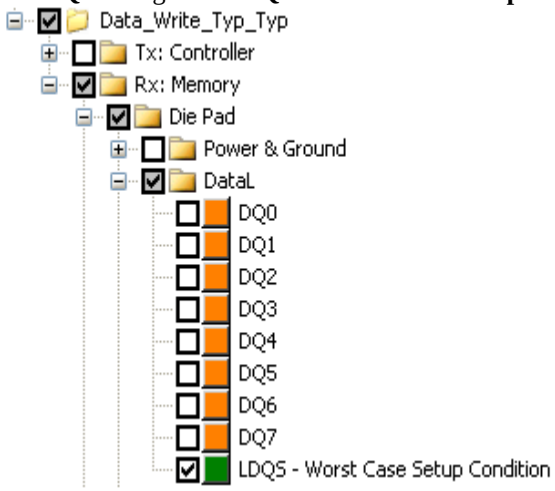
2. Enable **Worst Case Setup Condition**.

The DQS has shifted as the following figure shows.



- **Orange curve** - Data [0:7]
- **Red curve** - LDQS as Simulated
- **Green curve** - Worst Case Setup Condition

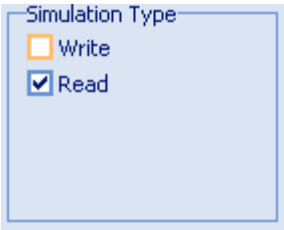
And **LDQS** changes to **LDQS - Worst Case Setup Condition.**



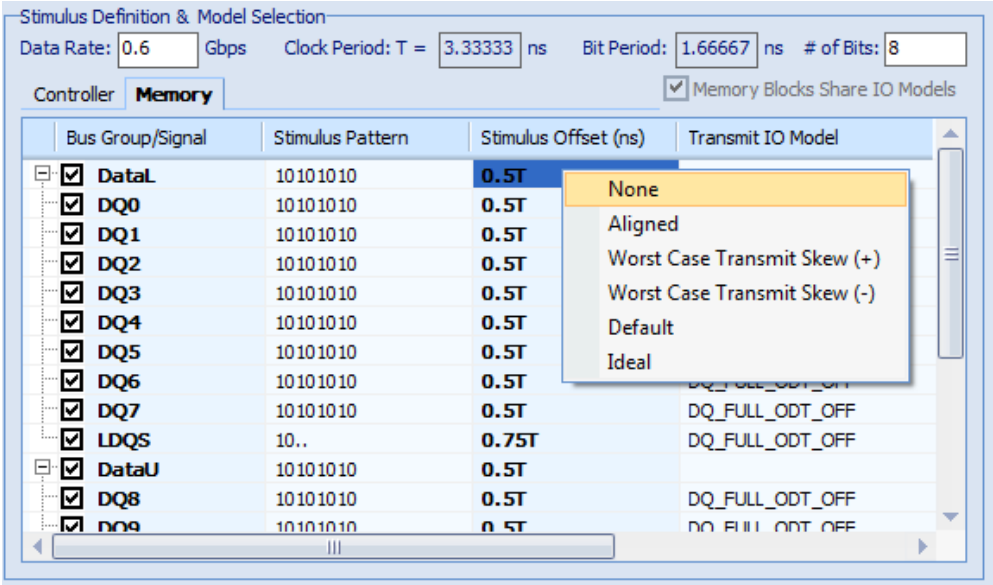
9.10.5.2 Memory Bus in Read Mode

For a Memory bus in **Read** mode, Parallel Bus Analysis can simulate both Default mode and Worst Case Condition automatically too. But it is a little different from **Write** mode.

1. Check **Read** as the simulation type in **Analysis Options**.



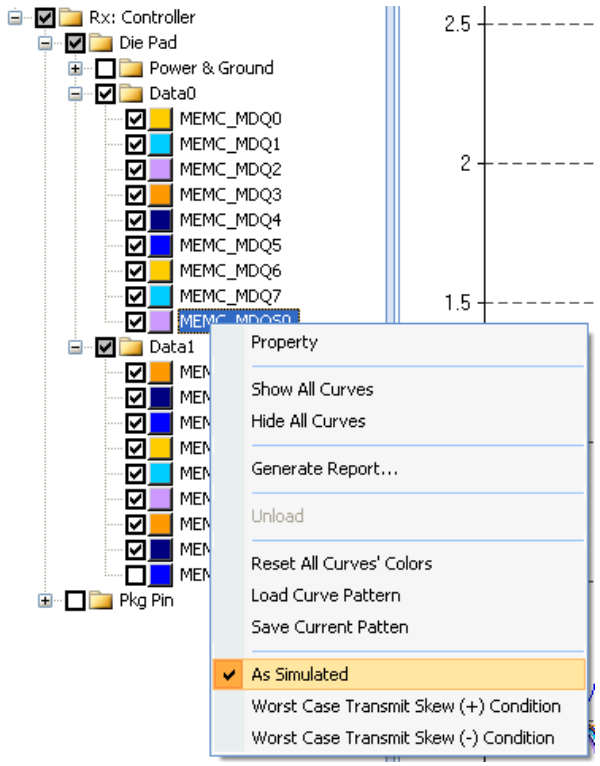
2. In the **Memory** tab pane, right-click the Stimulus Offset value (0.5T) of **DataL** and select **None**.



NOTE!	**Stimulus Offset** must be set to **None** for SystemSI to automatically analyze worst case timing for the **Read** case.

3. After simulation, enable **MEMC_MDQS0** and right-click it.

   The pop-up menu lists the Worst Case options.



   For a Memory bus group in **Read** mode, three options are available:

- **As Simulated –** Shows DQS waveform with Stimulus Offset per the Analysis Options form.
- **Worst Case Transmit Skew(+) Condition** - Shows DQS waveform with Stimulus Offset calculated from the Transmit Skew (+) parameter in the Timing Budget form.
- **Worst Case Transmit Skew(-) Condition** - Shows DQS waveform with Stimulus Offset calculated from the Transmit Skew (-) parameter in the Timing Budget form.

9.10.6 Terminate Unconnected Nodes

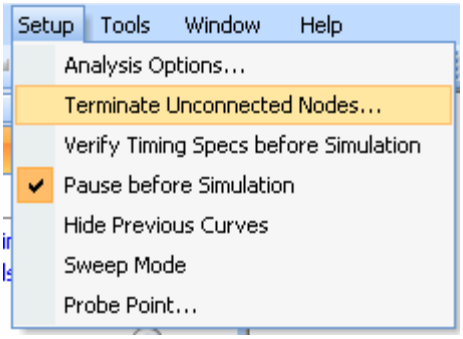
If there are some unused nodes in the block, System SI will automatically terminate the unused nodes.

By default, the termination value for Signal node is 50Ohm, for Power node is 1e+008Ohm and for Ground node is 0Ohm.

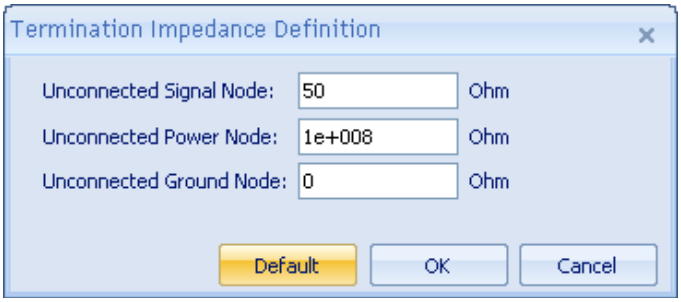
To change the termination value, follow these steps:

1. Select

   Setup > Terminate Unconnected Nodes….



The **Termination Impedance Definition** window pops up.



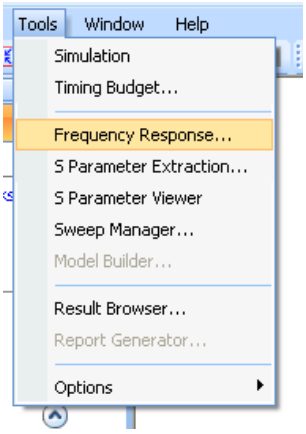
NOTE!	SystemSI will add termination for unused nodes in the blocks except Controller and Memory blocks.

9.10.7 Frequency Response

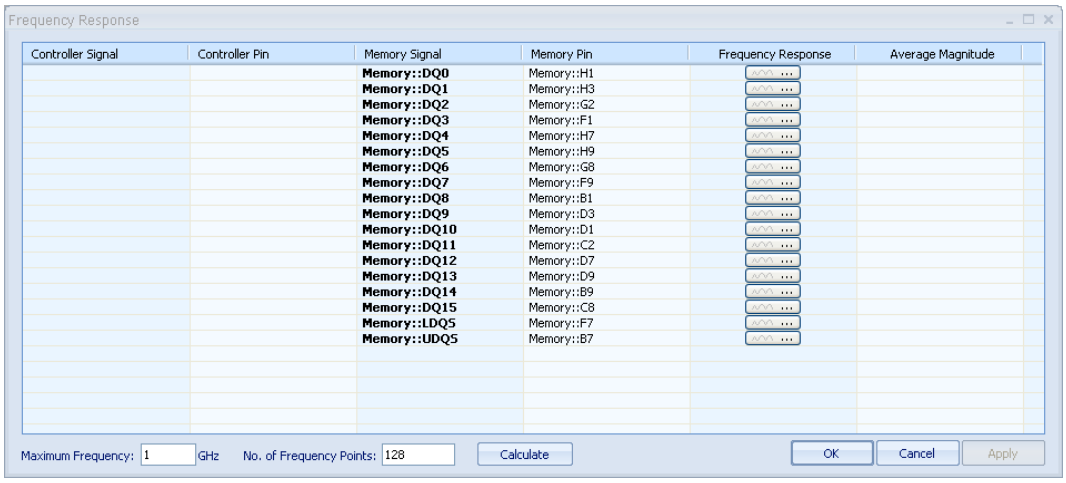
SystemSI - Parallel Bus Analysis provides calculation for frequency response.

1. Select

   Tools > Frequency Response….



The **Frequency Response** window opens.

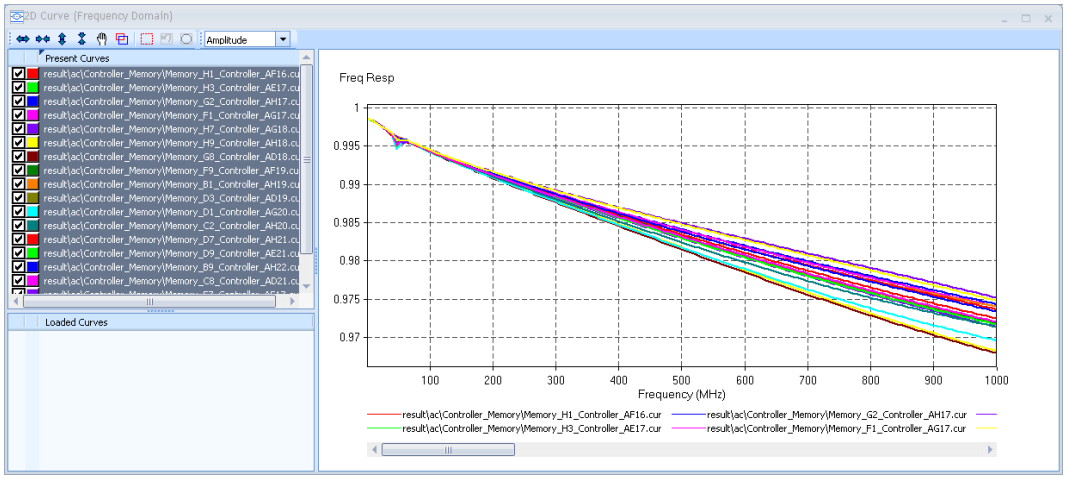


User can define **Maximum Frequency** and **No. of Frequency Points** by entering numbers in the fields next to them. By default:

- **Maximum Frequency** is 1Ghz.
- **No. of Frequency Points** is 128.

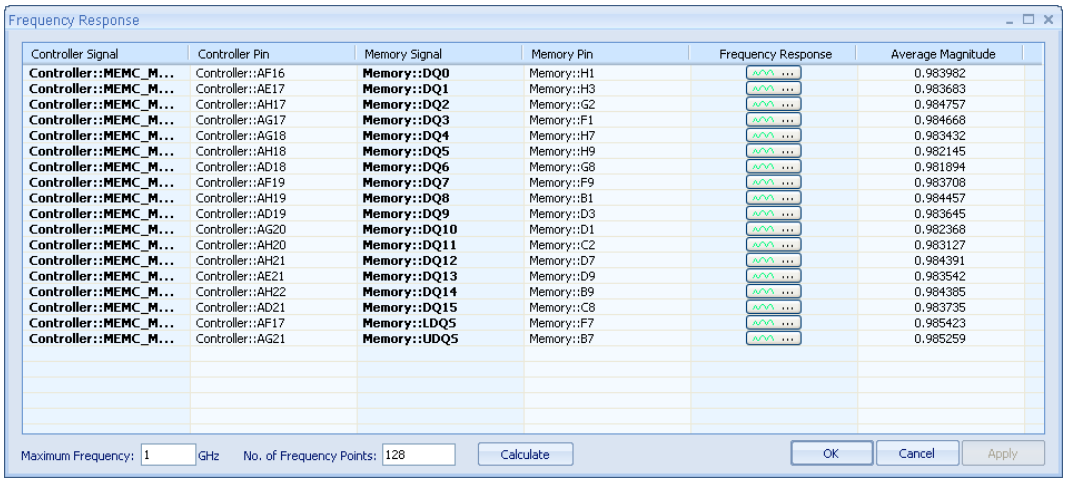
2. Click the **Calculate** button.

When simulation is completed, a result window opens with Frequency Response curves for all signal nets.



The **Frequency Response** window lists detailed information for the controllers. In the **Frequency Response** column, the gray  is turned to green  . It is enabled.

To show **Frequency Response** curve for desired net, just click the corresponding  button.

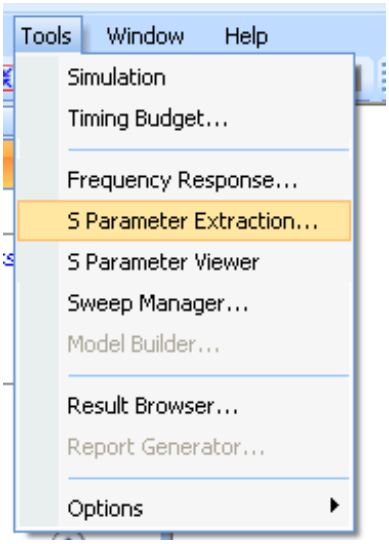


9.10.8 S-Parameter Extraction

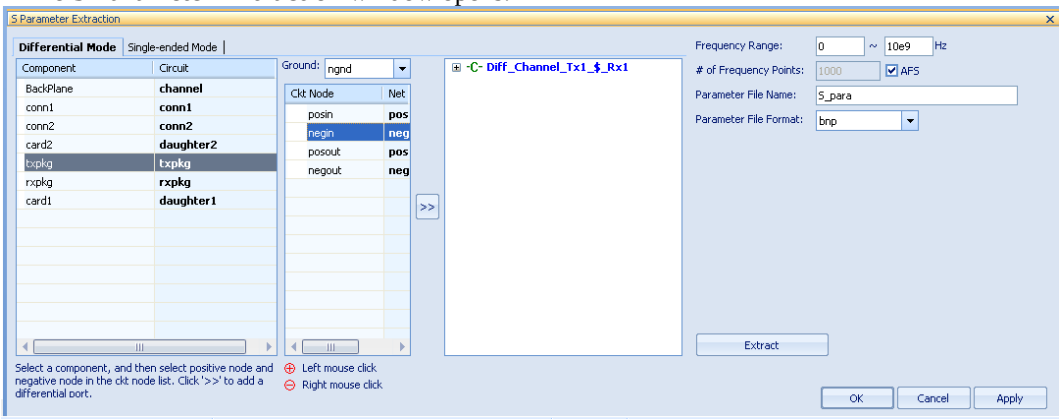
SystemSI - Parallel Bus Analysis supports **S-Parameter Extraction** for both single-ended nets and differential nets.

Select

Tools > S Parameter Extraction….

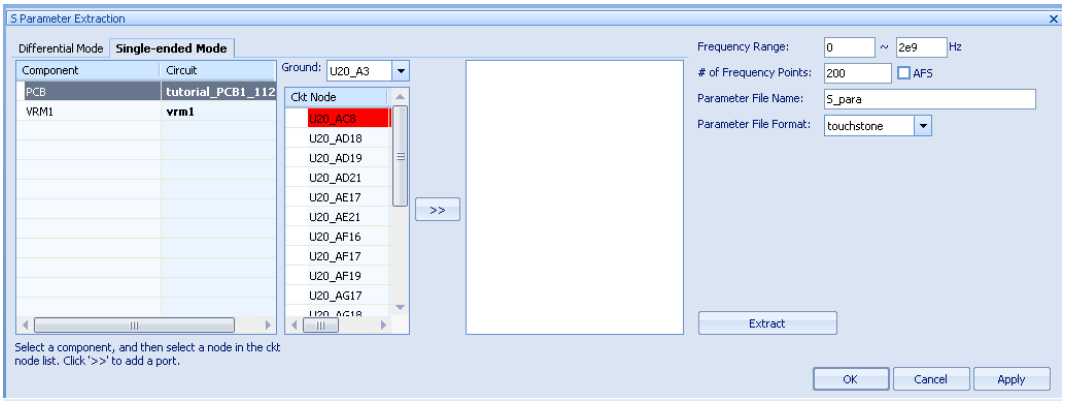


The **S Parameter Extraction** window opens.

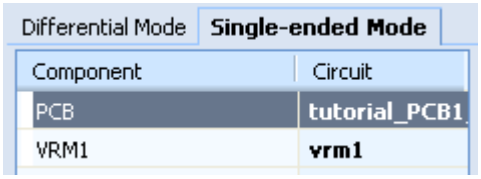


9.10.8.1 Single-ended Mode

1. Click the **Single-ended Mode** tab.



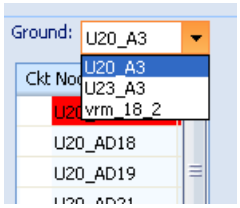
2. Select the **PCB** component to extract S Parameter.



3. Set up ports.

   3.1 Click **U20_AF16** to define positive node.

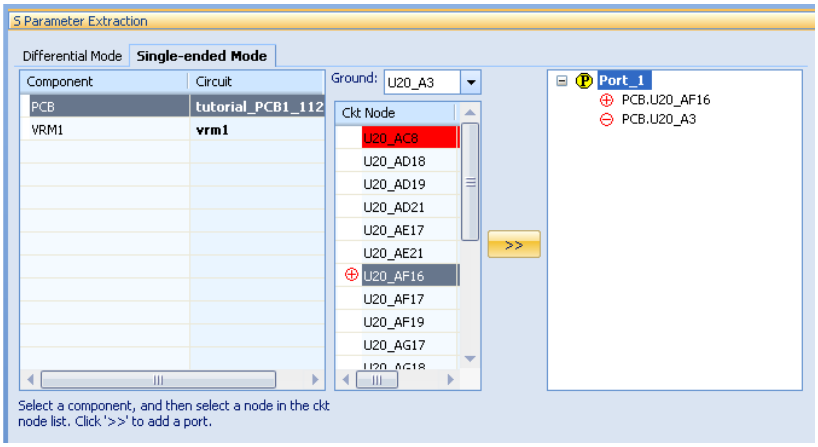
   3.2 Choose **U20_A3** as Ground node.



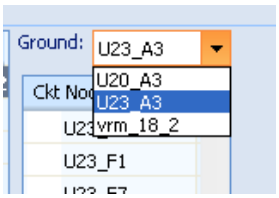
   3.3 Click the 

button.

   Settings for **Port_1** are completed as the following window shows.



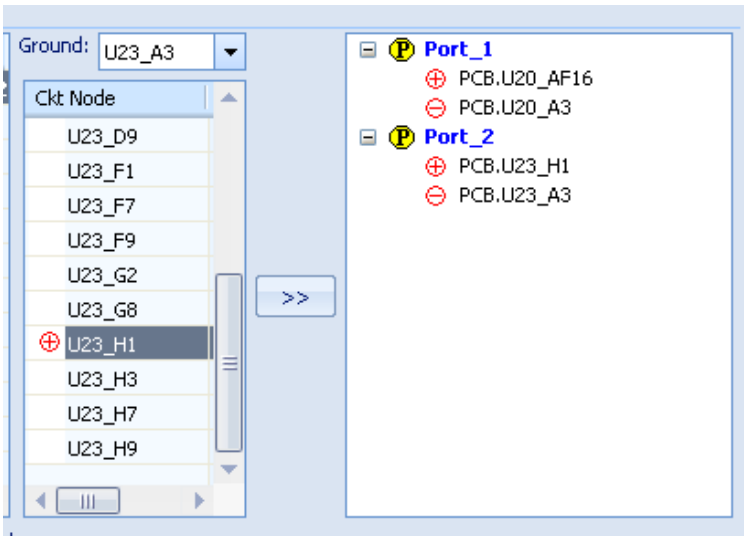
   3.4 Choose **U23_A3** as Ground node.



3.5 Click **U23_H1** to define positive node.

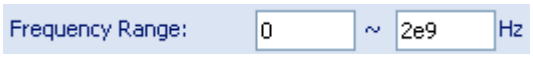
3.6 Click the >> button.

Settings for **Port_2** are completed as the following window shows.



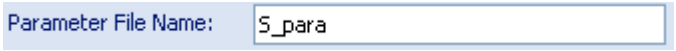
4. Set up Parameters.

4.1 Define **Frequency Range**.



4.2 Define **No. of Frequency Points**.

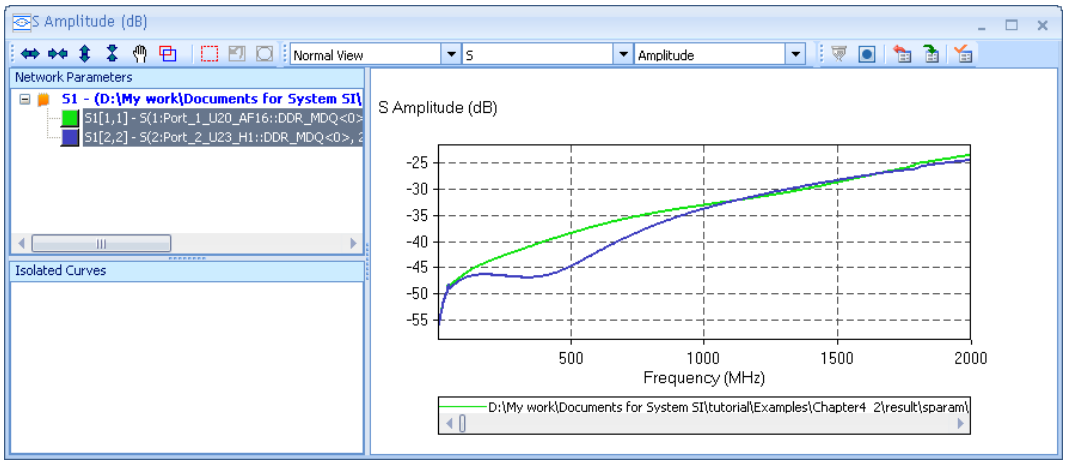
4.3 Input **Parameter File Name**. For example: S_para.



4.4 Select **Parameter File Format**: bnp.

5. Click the **Extract** button.

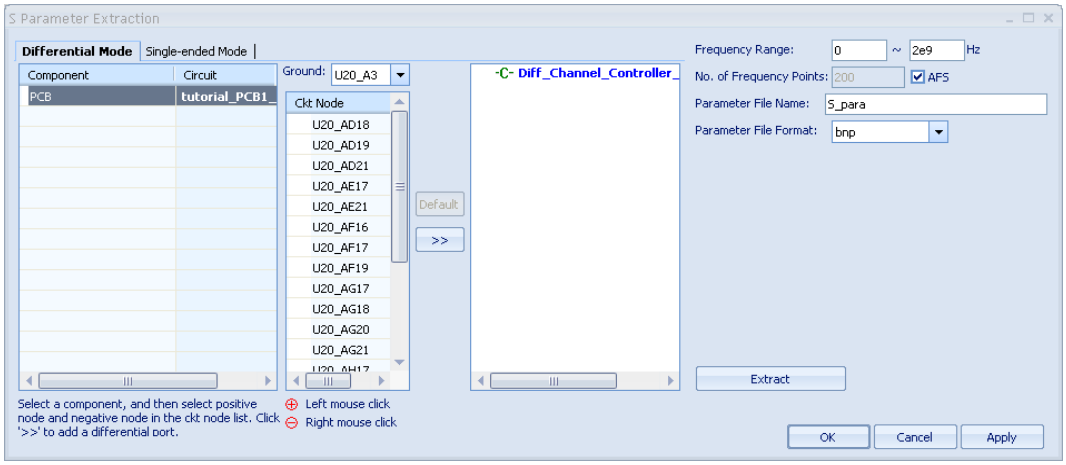
S Parameter curve for signal net **DQ0** shows as the following figure.



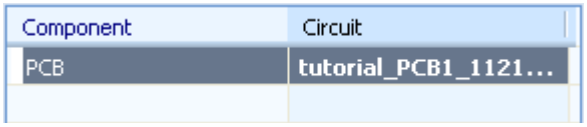
S Parameter document is automatically generated in the folder of <working folder path>\result\sparam\.

9.10.8.2 Differential Mode

1. Click the **Differential Mode** tab.

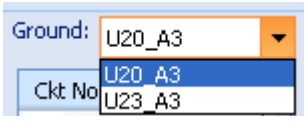


2. Select the **PCB** component to extract S Parameter.



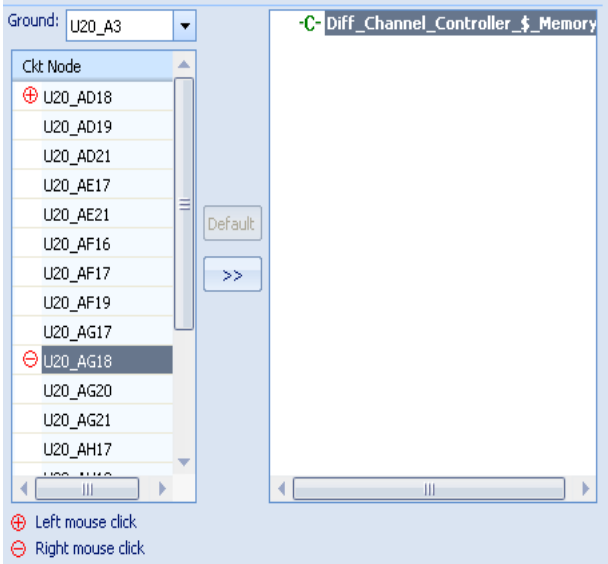
3. Set up ports.

   3.1 Choose **U20_A3** as Ground node.



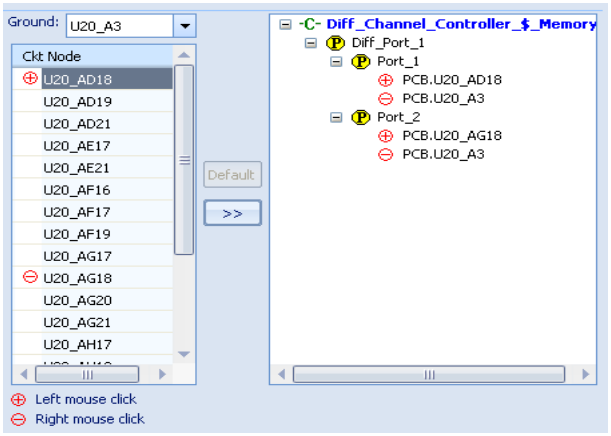
   3.2 Click the circuit node **U20_AD18** to define positive node.

   3.3 Right-click the circuit node **U20_AG18** to define negative node.



3.4 Click the  button.

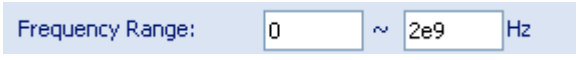
The port **Diff_Port_1** is generated automatically.



NOTE!	If differential TimingRef signals exist in **Controller** and **Memory** blocks, the **Default** button is enabled. You can click the **Default** button to add ports for differential TimingRef signals automatically. The **Default** button is gray if no differential TimingRef signals exist, like in this case.

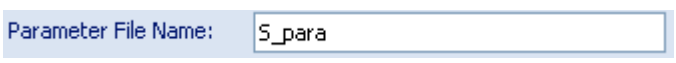
4. Set up parameters.

4.1 Define **Frequency Range**.



4.2 Define **No. Of Frequency Points**.

4.3 Input **Parameter File Name**.



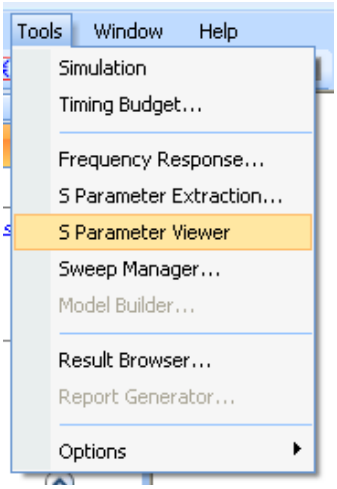
4.4 Select **Parameter File Format**.

5. Click the **Extract** button. (In this example, there are no differential signal nets, so click **Cancel** to exit.)

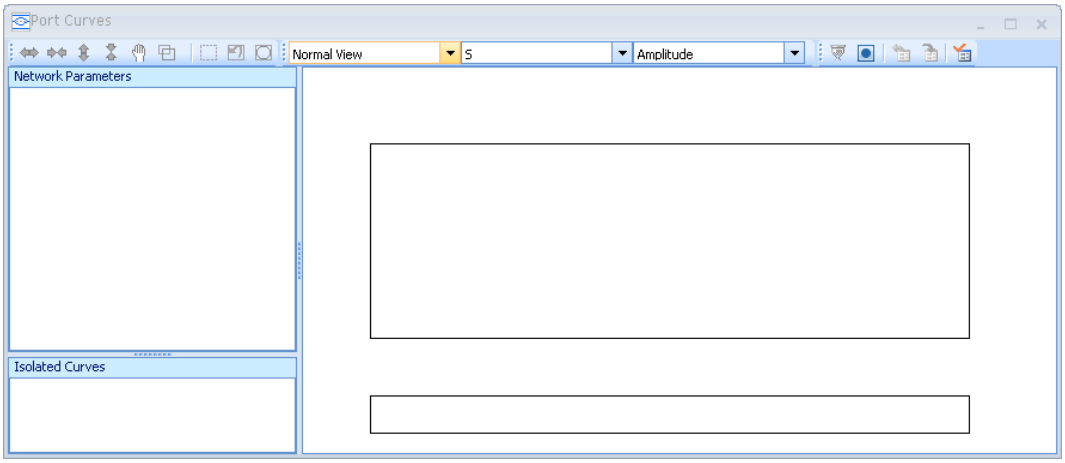
9.10.9 S-Parameter View

1. Select

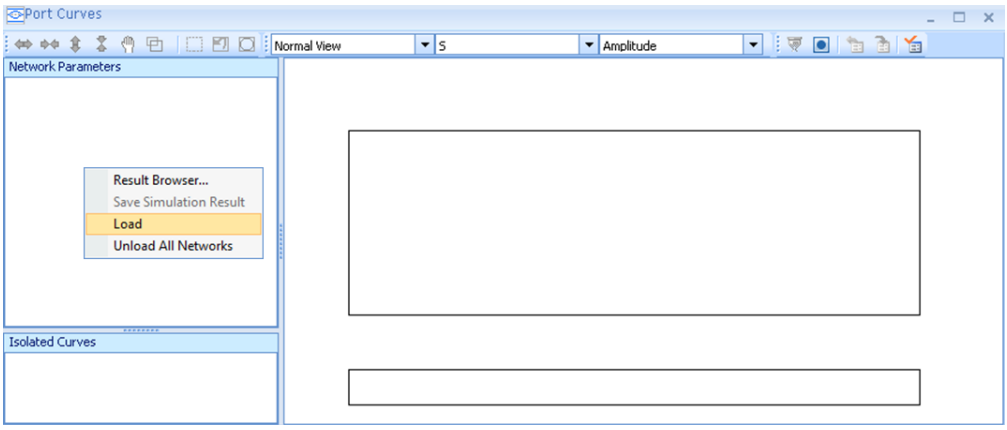
   Tools > S Parameter Viewer.



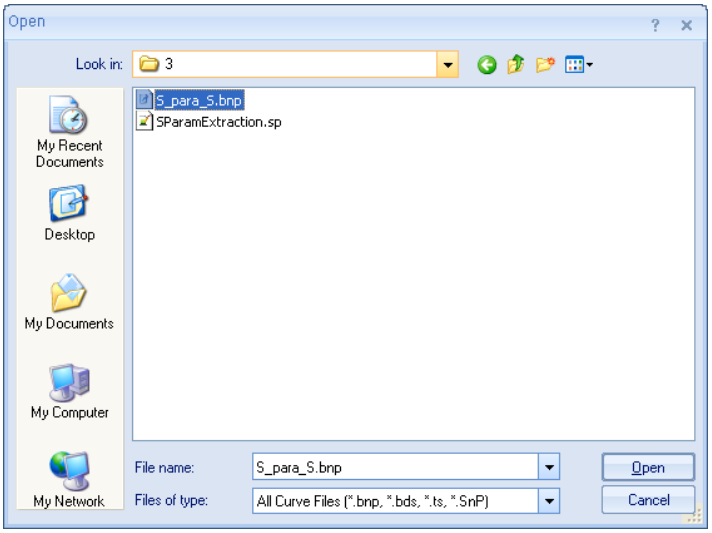
   The **Port Curves** window opens.



2. Right-click the **Network Parameters** pane.
3. Select **Load** in the pop-up menu list.

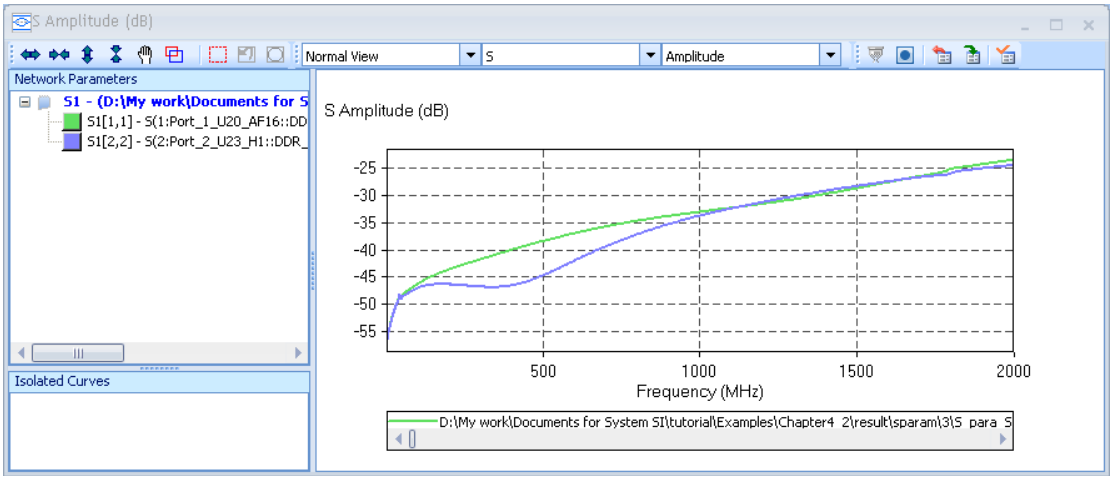


The **Open** window opens.



4. Select an S Parameter, e.g. **S_para_S.bnp**.
5. Click **Open**.

   The curve shows in the result pane like the following figure.

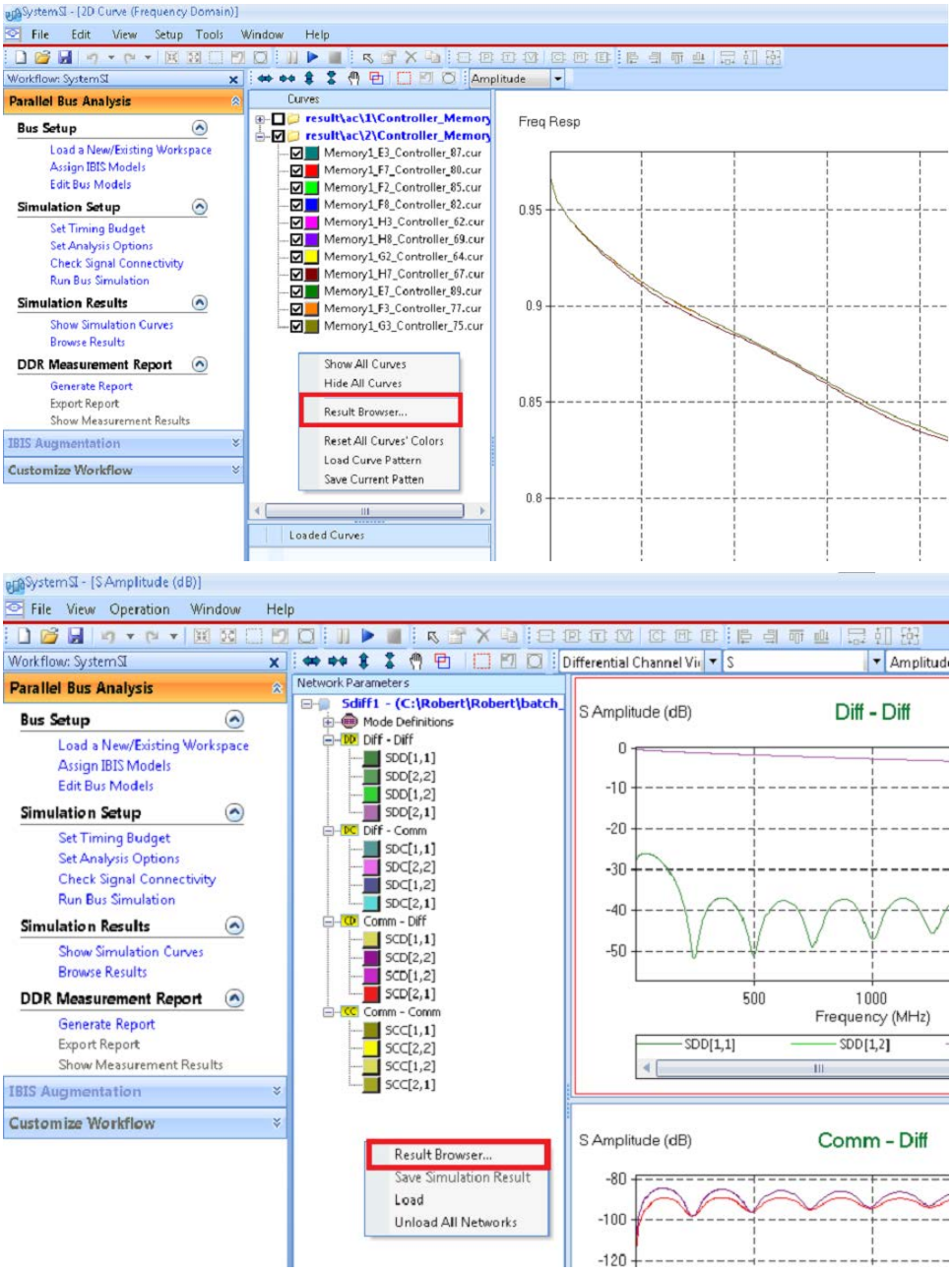


9.10.10 Result Browser and 2D Curve Presentation

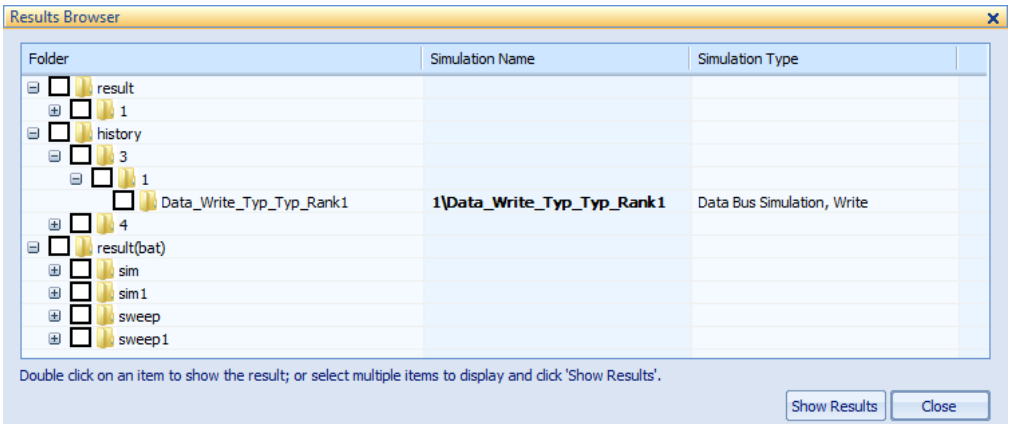
9.10.10.1 Result Browser

The **Result Browser** option is available for all the curve windows, such as 2D Curve (Time Variation), 2D Curve (Frequency Response), and S Parameter Viewer.

You can right-click in the curve panel, and choose **Result Browser…** from the pop-up menu to view the results. As shown in the following figure.



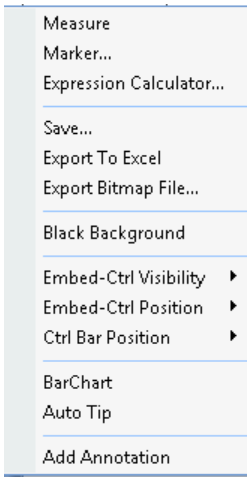
With this option, all the previous results under the **result**, **history** and **result(bat)** folders can be browsed and displayed from various curve windows.



9.10.10.2 Curve Pane Context Menu

Right-click on the Curve window.

A context menu appears.



You can perform the following operations in this menu.

- **Measure** — Toggle the horizontal and vertical measure lines.
- **Marker** — Toggle the horizontal and vertical marker lines.
- **Expression Calculator**—Setup and calculation the expression.
- **Save**—Save the curve.
- **Export To Excel**—Export the curve to Excel.
- **Export to Bitmap File**—Export the curve to Bitmap file.
- **Black / White Background** — Set the background of the curve window to be black or white.
- **Embed-Ctrl Visibility** – Set the visibility of the sub windows (for example, the legend bar) in the display area.
- **Embed-Ctrl Position** – Toggle the sub windows between floating and docking.
- **Ctrl Bar Position** – If a sub window is docked, change the position of the docking.
- **Bar Chart** — Toggle the plot style between a bar chart and a continuous line.
- **Auto Tip** — Show / Hide the tip of the objects in the Curve window when moving the mouse.

- **Add Annotation** — Add a text string in the Curve window.

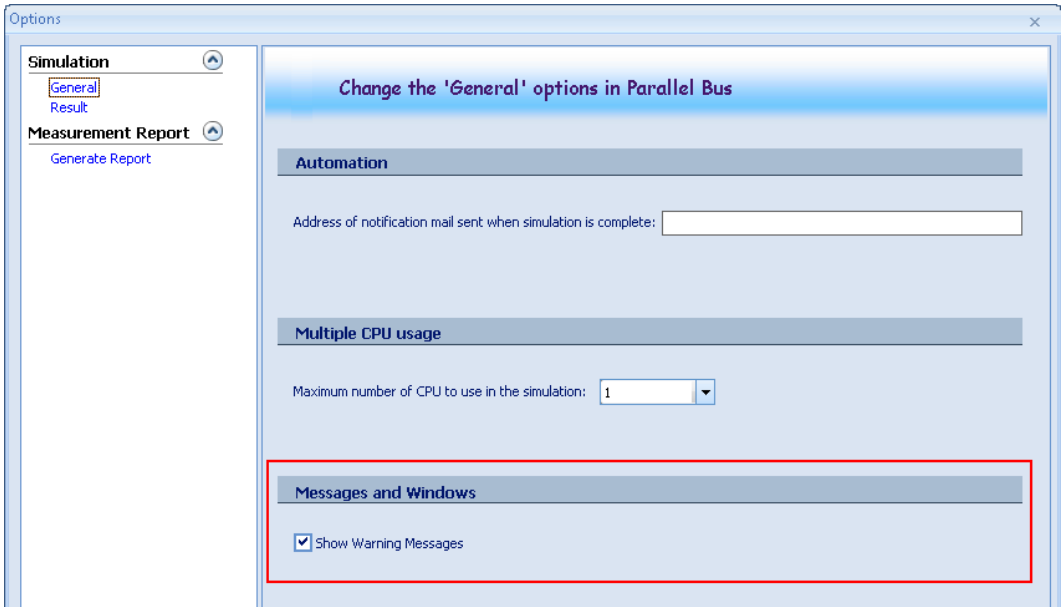
All these common items are available for 2D Curve (Time variation), 2D Curve (Frequency Response), and 2D Curve (DDR Measurement). However, there is an exception. **Expression Calculator…** is not available for 2D Curve (DDR Measurement). Each 2D Curve may have additional items which are unique to the specific 2D Curve window.

9.10.11 General Options

While performing Parallel bus analysis in SystemSI, you can specify some settings in the Options dialog box.

a) Choose ***Tools – Options – Edit Options***.

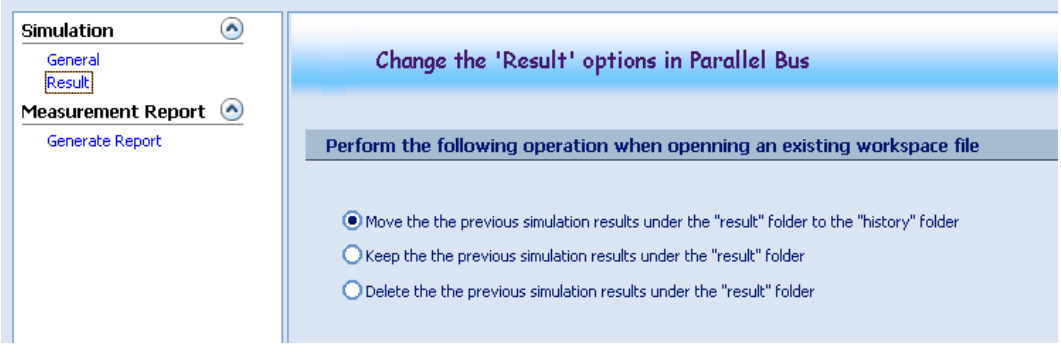
The **Options** window opens.



The **General** page contains three sections:

- **Automation** - user can put email address into the blank dialog box, and then user can get notification email when simulation is complete
- **Multiple CPU usage** - user can set the maximum number of CPU to use in the simulation
- **Messages and Windows –** Provides users with the option to show or hide the warning messages generated during simulations. By default, all warning messages are displayed. To hide the warning messages, clear the Show Warning Messages check box.

b) To specify the display setting of previous simulation results, select Result in the left pane of the Options dialog box.



c) Select the appropriate option. To save the modifications and close the dialog box, click **OK**.

10 DDR Measurement Report

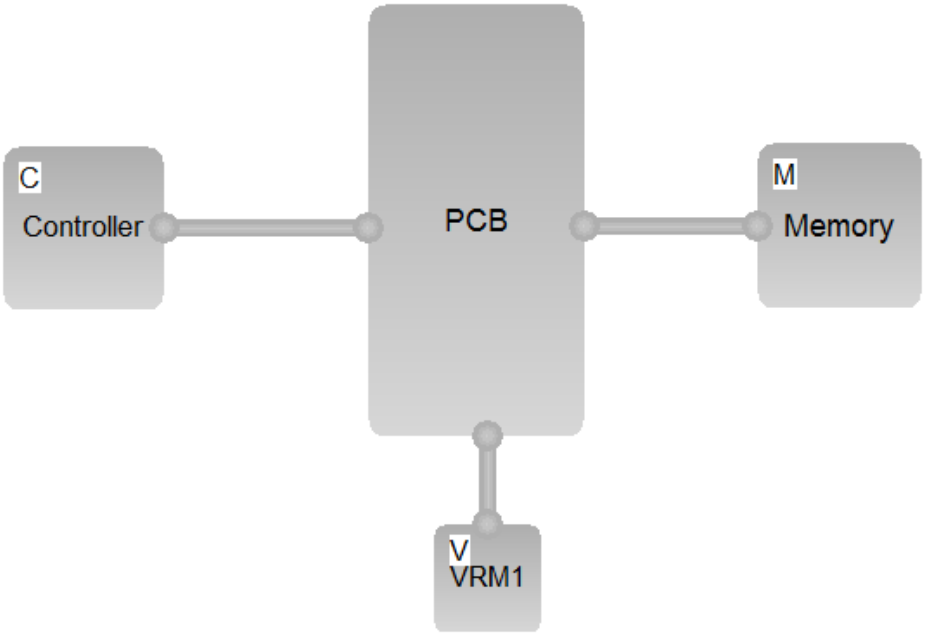
10.1 Introduction

Parallel Bus Analysis has extensive DDR data processing and specification compliance functionality. Standard measurements and specifications from JEDEC standards are included, with user-friendly data presentation and parsing, for unprecedented troubleshooting, all combined with the implicit accuracy that comes with Sigrity unique simulation technology.

This chapter explores the DDR measurement reporting using the completed Tutorial examples.

10.2 Tutorial Files

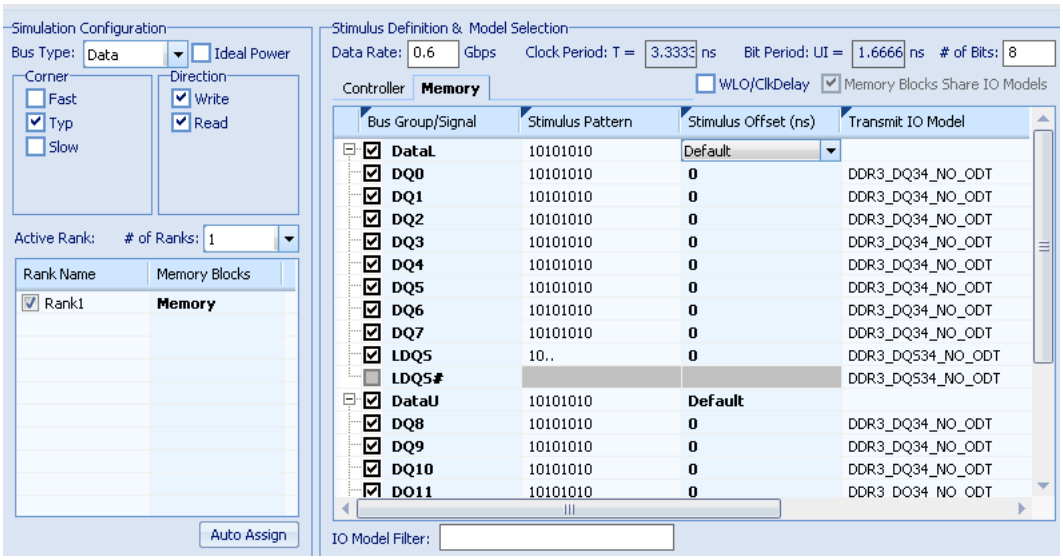
The block diagram for this chapter consists of a Controller, Memory, PCB, and VRM component, as created in Chapter 9. This design is complete and fully functional with models for each component and a valid connection between each. The design looks like this:



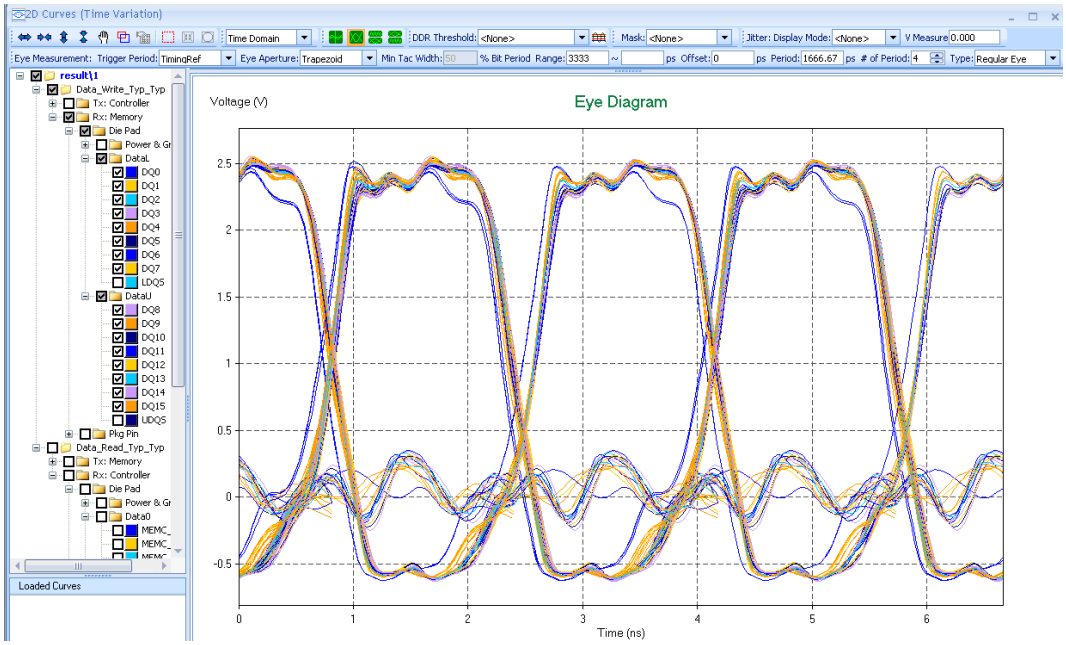
10.3 Simulation Results

Simulate the completed example from ***Chapter 9***. Make sure the following details in the Analysis Options are set.

1. Turn off **Ideal Power**.
2. Make sure the directions **Write** and **Read** are checked.
3. Click the **Memory** tab.
4. Ensure that data buses, **DataL** and **DataU**, are selected and the **Stimulus Offset (ns)** is set to the **Default** value of **0**.
5. Set the **Data Rate** is **0.6** Gbps.



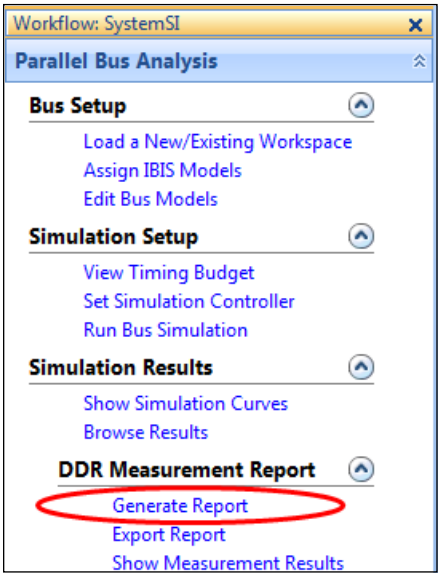
Then, simulate, and plot the Eye Diagram. It should look like this:



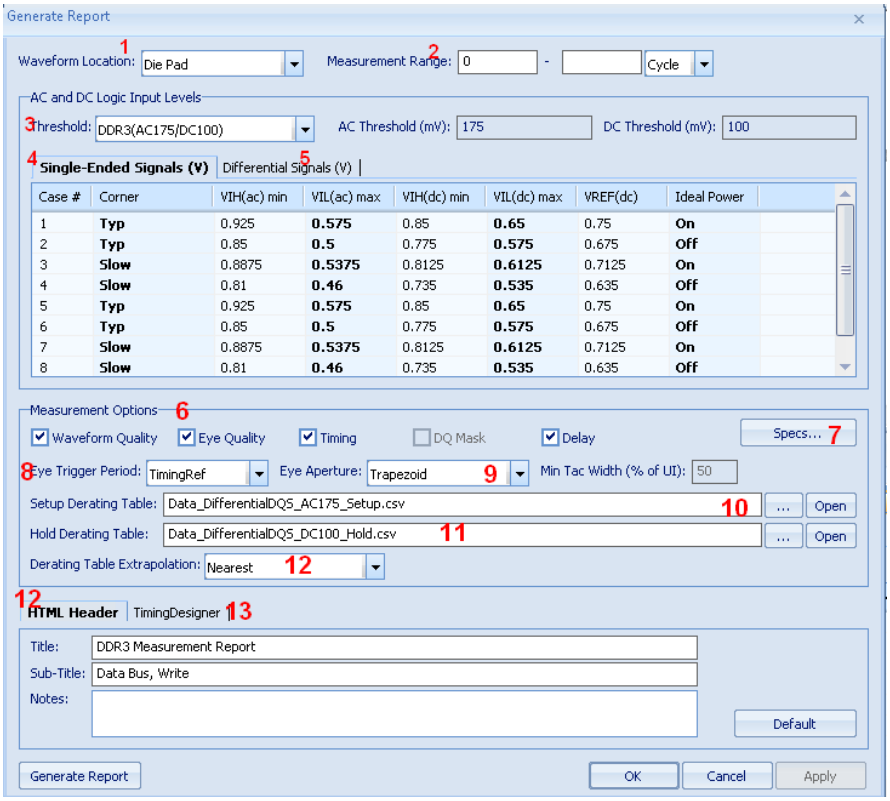
Be sure to set the Offset to approximately 3333 ps, to eliminate the leading “0s”.

10.4 DDR Report Settings

Generate a **DDR Measurement Report** from the **Workflow** selection.



The **Generate Report** dialog box displays.



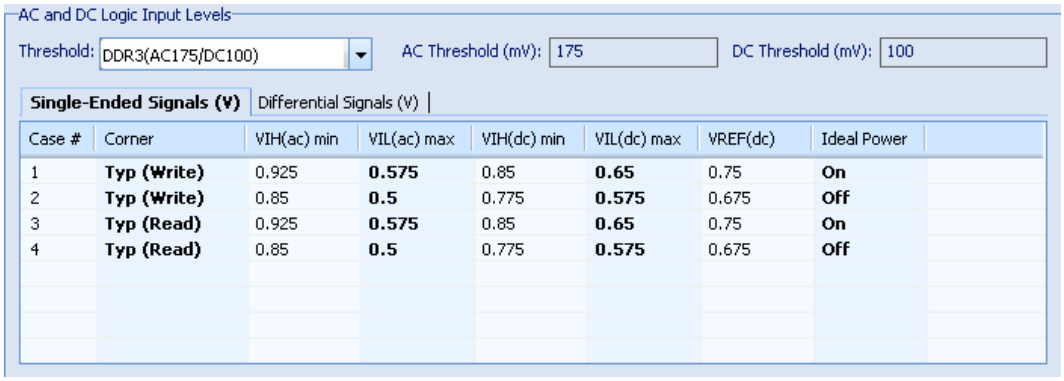
The various parts of this dialog box are discussed below:

1. **Waveform Location**: Specifies whether the output waveform uses Die Pad or PKG Pin data.
2. **Measurement Range**. User can choose anyone of the three units: nano seconds (**ns)**, peco-seconds (**ps)** and **Cycle.**

3. **Threshold** : This drop-down list lists JEDEC specific AC and DC threshold levels for known DDR4, DDR3, DDR2, DDR1, LPDDR3 and LPDDR2 standards are included.

NOTE!	If you are running Channel simulation for Data Bus Write operation, only DDR4 threshold values can be selected.

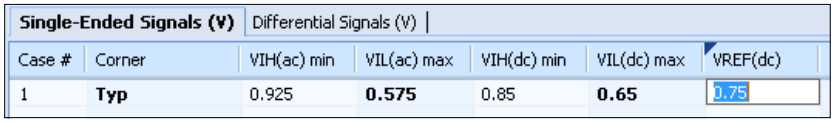
4. Selecting one of the pre-defined threshold values updates the Voltage values in the dialog, including the **AC** and **DC** threshold values, as well as the Single-ended **Typical** corner values for Low and High logic levels, as displayed in the Single-Ended Signals tab.



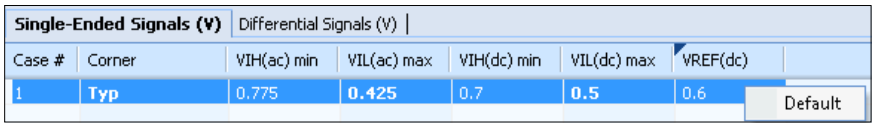
Case #	Corner	VIH(ac) min	VIL(ac) max	VIH(dc) min	VIL(dc) max	VREF(dc)	Ideal Power
1	**Typ (Write)**	0.925	**0.575**	0.85	**0.65**	0.75	**On**
2	**Typ (Write)**	0.85	**0.5**	0.775	**0.575**	0.675	**Off**
3	**Typ (Read)**	0.925	**0.575**	0.85	**0.65**	0.75	**On**
4	**Typ (Read)**	0.85	**0.5**	0.775	**0.575**	0.675	**Off**

You can modify the DC reference Voltage value, *VREF(dc)*.

- To edit the default value, double-click on the $VREF_{(dc)}$ value and enter the new value.



- To restore the default value, right-click and select Default.



Adding User-Defined Threshold Values

If required, you can also add a custom threshold value. To create a user-defined threshold:

a) From the Threshold drop-down list, select *<User Defined..>*

b) In the DDR Threshold dialog box, click *Add*.

c) Specify the Threshold Name, and AC and DC Threshold values.

d) If required, you can also specify the values for Differential AC & DC Thresholds.

e) Click OK.

The Threshold Name is visible in the drop-down list.

5. The Differential Tab displays the differential corner values for Typical Low and High logic levels.

6. The Measurement types currently include **Waveform Quality**, **Eye Quality**, **Timing, DQ Mask,** and **Delay**.

When the DQ Mask option is enabled; Eye Quality And Timing Reports are not generated.

NOTE!	The **DQ Mask** option is enabled only for the DDR4 measurement on the Data Write simulation only. For DQ Mask option to be enabled, following conditions must be met: • Selected **Threshold** is *DDR4(AC100/DC75)* • Simulation Results have Data Write simulations • at least one simulation has the Stimulus Offset set to the Default or Ideal Unless all these conditions are satisfied, the **DQ Mask** option is unchecked and grayed out. Eye Quality and Timing options are not available for channel analysis for Data Bus Write simulations.

7. To set the measurement specifications, click **Specs**.
   - In the Value column, you can set the values for different parameters.
   - To restore the default values, or to use the Timing Specifications from the *Timing Budget* set for this simulation, select the **Default** and **Pull Timing Budget** buttons, respectively.
   - The value specified in the Strobe Adjustment Resolution text box, is used for post-processing of Data Bus simulation results.
8. User can specify **Eye Trigger Period**. User can choose **TimingRef** or **Same As UI**.
9. User can use **Eye Aperture** and **Min Tac Width** for **ApertureWidth** measurement.

   Two options are available for **Eye Aperture**:

   - **Trapezoid** — Selected by default
   - **Tac/Tdc Rectangles** — If selected, eye mask consists of two rectangular shapes:

     - **Tac**: Height of 2*VIH(ac)(from VIL(ac) to VIH(ac)), with a width of Tac.

     - **Tdc**: The second rectangle is from (VIL(dc) to VIH(dc)) and with width of Tdc.

     The overall eye aperture width measurement = Tac + Tdc.

   Min Tac Width can be useful when **Tac/Tdc Rectangles** is selected for **Eye Aperture**. It is a positive number ranging from 0 to 100. The default is 50.

NOTE!	If the ringback occurs after the Tac time point which is higher than ViHDC, it doesn't impact the eye-width measurement hence it is not a failure. If the ringback occurs within the Tac time point which can impact receiver's switching, the first rectangle can be excluded from the eye-width measurement.

10. Derating tables for both **Setup** and **Hold** can be specified, with many standard ones that are available from public-domain sources included.
11. User can choose **Derating Table Extrapolation** from **None** and **Nearest**.
12. HTML Header
    This tab allows you to specify the title and Sub-title of the report. The entries in the Notes text box are appended to the report header.
13. Timing Designer

Select this tab, if you want to Launch Timing Designer on the Timing measurements and configuration files generated by SystemSI.

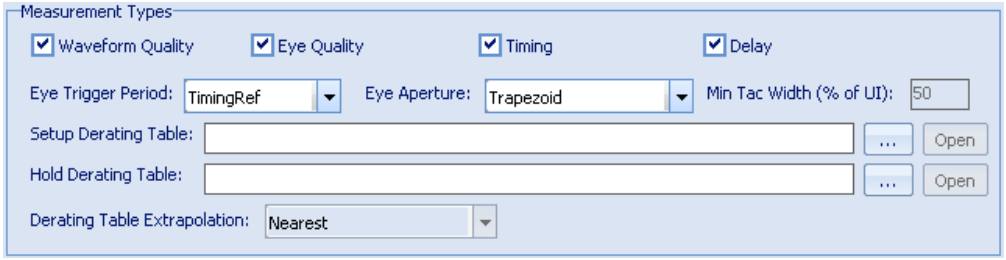
By default, the Setup and Hold measurements are exported to Timing Designer. However, in case of Data Read bus, you can either export the Setup and Hold measurements or export ReadSkew.

NOTE!	This tab is visible only if the interface to the TimingDesigner is enabled for the project. To enable the interface: a) From the ***Tools*** menu, choose ***Options – Edit Options***. b) In the Options dialog box, select ***Generate Report***. c) In the Generate Report page, select the ***Interface to TimingDesigner*** checkbox . d) Click OK. If you now launch the Generate Report dialog box, the TimingDesigner tab is displayed.

10.5 Setup and Hold Derating Tables

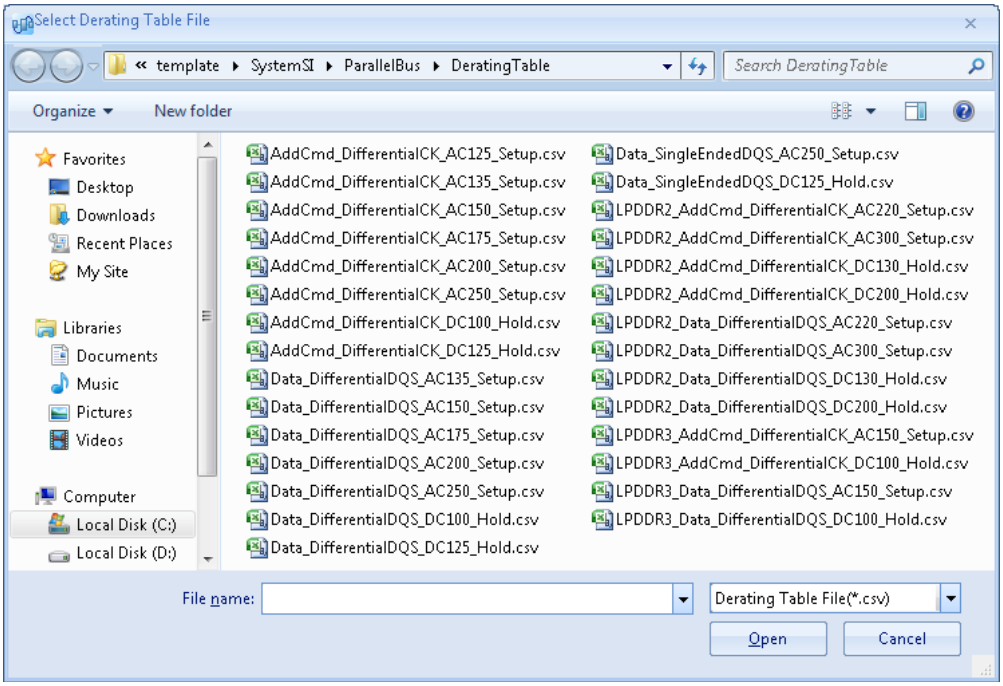
Standard derating tables from JEDEC are included with the tool.

These are delivered in CSV text format, and custom tables can be created by the user. Select a Derating Table by clicking the browse button in the Measurement Types window.



The default location for the derating tables is in the Allegro Sigrity installation directory, typically:

<INSTALL_DIR>\SpeedXP\Library\template\SystemSI\ParallelBus\DeratingTable

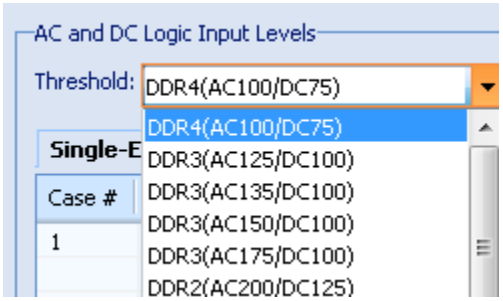


The derating table contains setup or hold scaling constants for specified slew rates. These tables are obtained from JEDEC and other public-domain sources, including device manufacturers. A typical Derating table is shown here:

Delta_tDH (ps) -- DC100 tDH Derating Values for DDR3-800/1066/1333/1600/1866/2133									
		DQS, DQS# Differential Slew Rate (V/ns)							
		4.0 V/ns	3.0 V/ns	2.0 V/ns	1.8 V/ns	1.6 V/ns	1.4 V/ns	1.2 V/ns	1.0 V/ns
		Delta_tDS	Delta_tDS	Delta_tDS	Delta_tDS	Delta_tDS	Delta_tDS	Delta_tDS	Delta_tDS
DQ Slew Rate (V/ns)	2.0	50	50	50					
	1.5	34	34	34	42				
	1.0	0	0	0	8	16			
	0.9		-4	-4	4	12	20		
	0.8			-10	-2	6	14	24	
	0.7				-8	0	8	18	34
	0.6					-10	-2	8	24
	0.5						-16	-6	10
	0.4							-26	-10

10.6 DDR4 Threshold

To support DDR4 interface, the **DDR4(AC100/DC75)** threshold is added to the Threshold drop-down list.

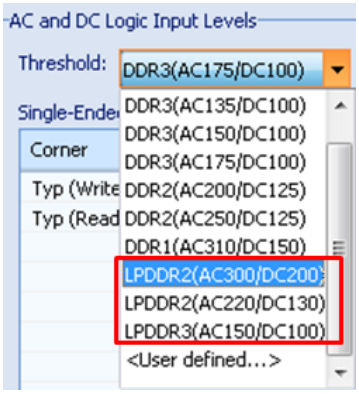


If the **DDR4(AC100/DC75)** is selected, the AC and DC Logic Input Level for the Differential signals are determined using following equations.

$$VIHdiff_{(ac)min} = 2 * (VIH_{(ac)min} - VREF_{(dc)})$$
$$VILdiff_{(ac)max} = 2 * (VIL_{(ac)max} - VREF_{(dc)})$$
$$VIHdiff_{(dc)min} = 0.15$$
$$VILdiff_{(dc)max} = - 0.15$$

10.7 LPDDR2 and LPDDR3 Threshold Application

There are two thresholds for LPDDR2, and one for LPDDR3.



If LPDDR2(AC300/DC200), LPDDR2(AC220/DC130) or LPDDR3(AC150/DC100) is selected, the AC and DC logic input levels for the Differential Signals will be determined by:

VIHdiff(ac) min = 2 * (VIH(ac) min - VREF(dc))

VILdiff(ac) max = 2 * (VIL(ac) max - VREF(dc))

VIHdiff(dc) min = 2 * (VIH(dc) min - VREF(dc))

VILdiff(dc) max = 2 * (VIL(dc) min - VREF(dc))

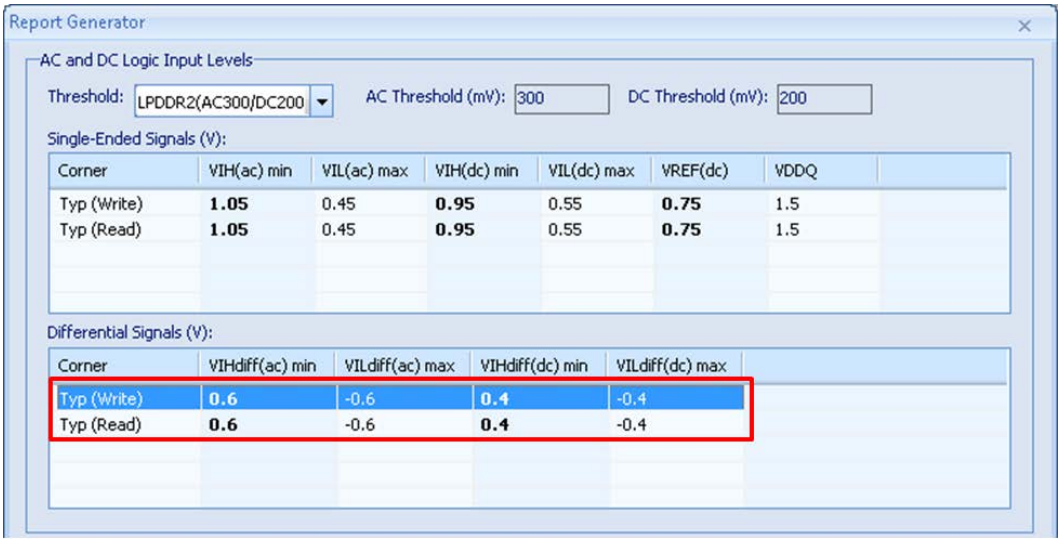
Otherwise, they will be determined by:

VIHdiff(ac) min = 2 * (VIH(ac) min - VREF(dc))

VILdiff(ac) max = 2 * (VIL(ac) max - VREF(dc))

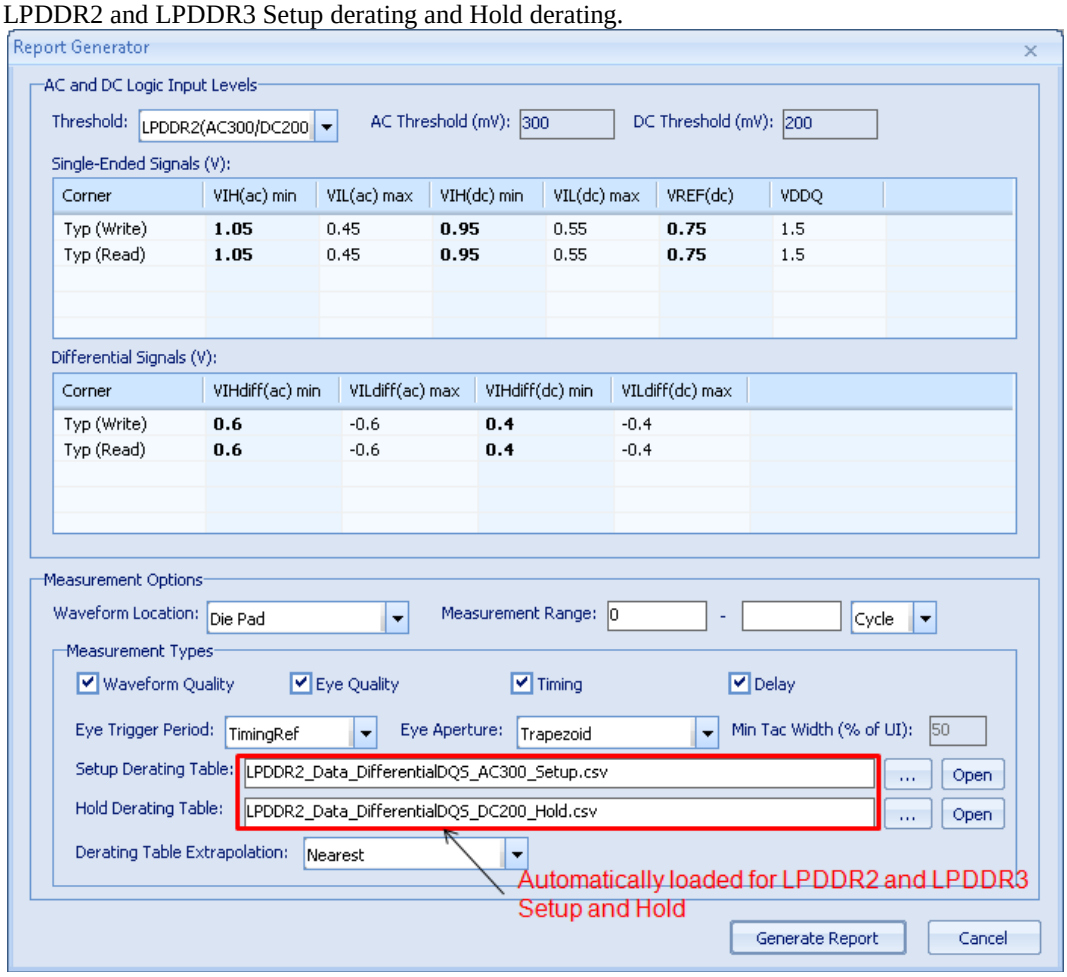
VIHdiff(dc) min = 0.2

VILdiff(dc) max =- 0.2

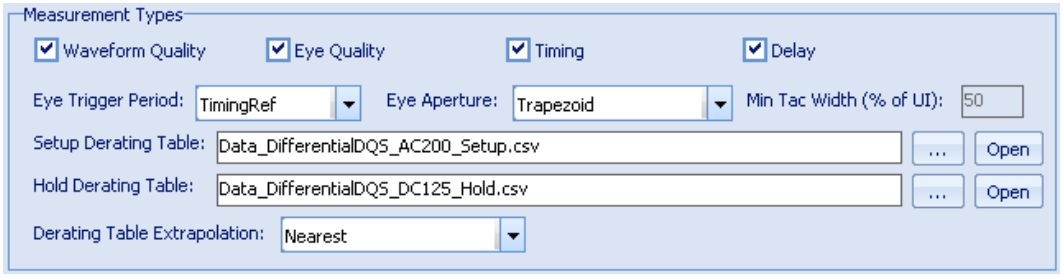


If the TimingRef is differential, the default derating table files will be automatically loaded for

LPDDR2 and LPDDR3 Setup derating and Hold derating.



10.8 Generate the Report



After specifying details of the DDR post-processing, the report is generated by clicking **Generate Report**. The generator returns with the report document itself as well as several data analysis templates for viewing and analyzing the data.

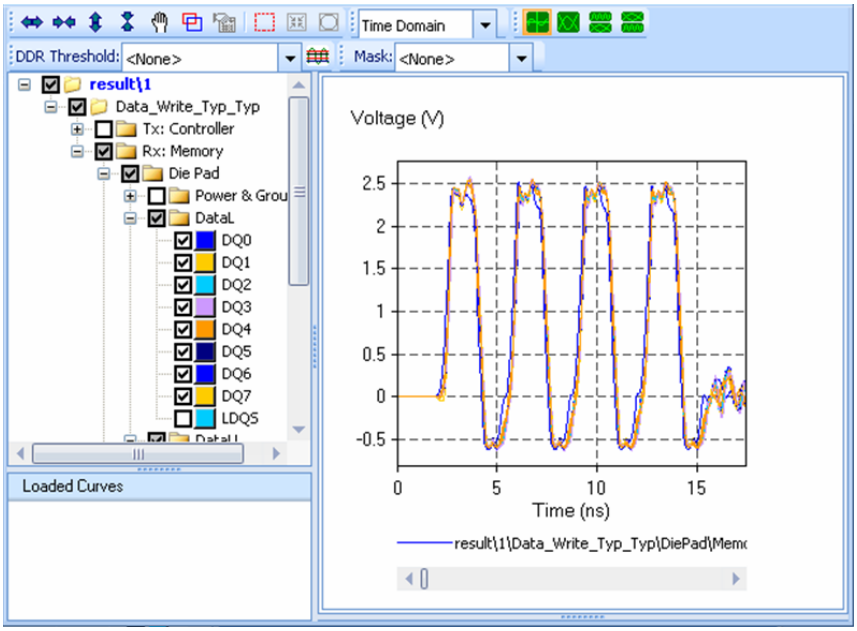
DDR3 Measurement Report

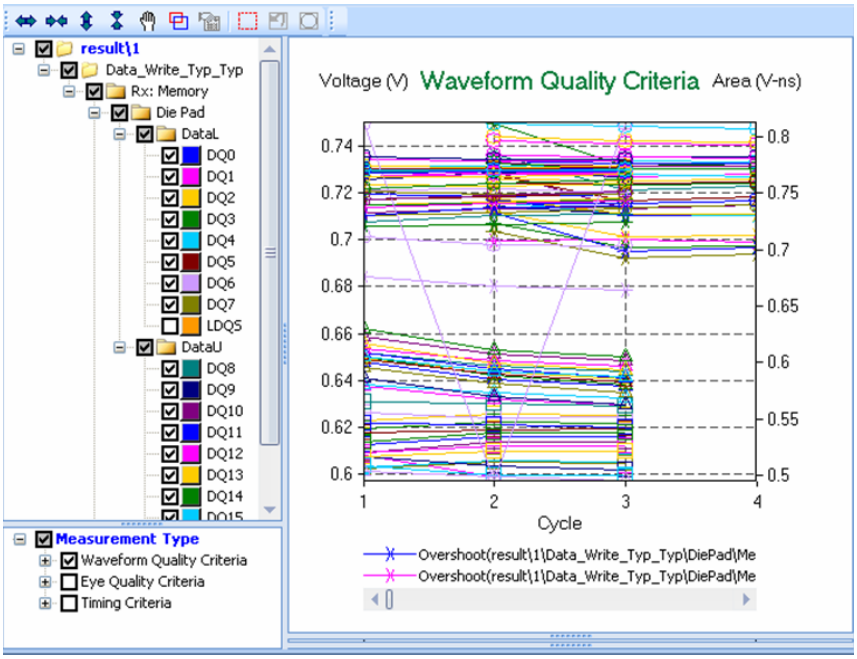
- Data Bus, Write

Generated by Cadence SystemSI, 12.1.2.09061
November 01, 2013

Useful Links:

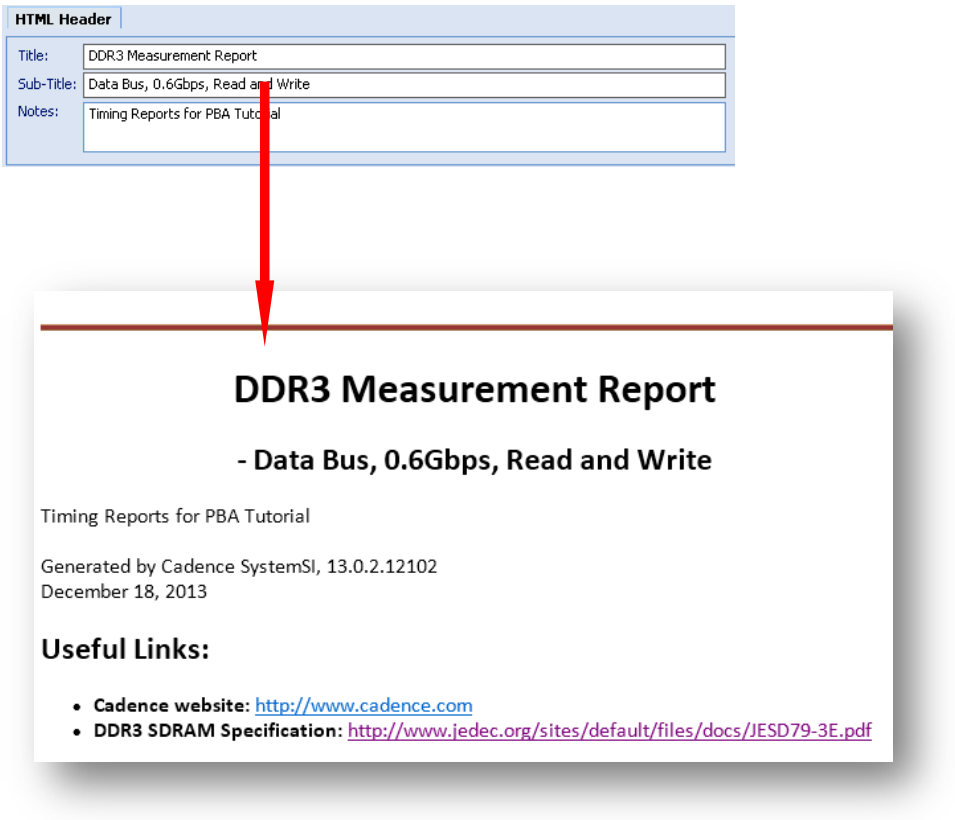
- **Cadence website:** http://www.cadence.com
- **DDR3 SDRAM Specification:** http://www.jedec.org/sites/default/files/docs/JESD79-3E.pdf





10.9 Report Contents

Each report has a header section. This section displays the Title and Subtitle of the Report as entered by you in the HTML Header tab of Generate Tab. The entries made in the Notes section are also appended in this section.



The tool version, and the date on which report was generated, are also listed in this section. Finally, this section includes links to the Cadence website and a link to the DDRx specification.

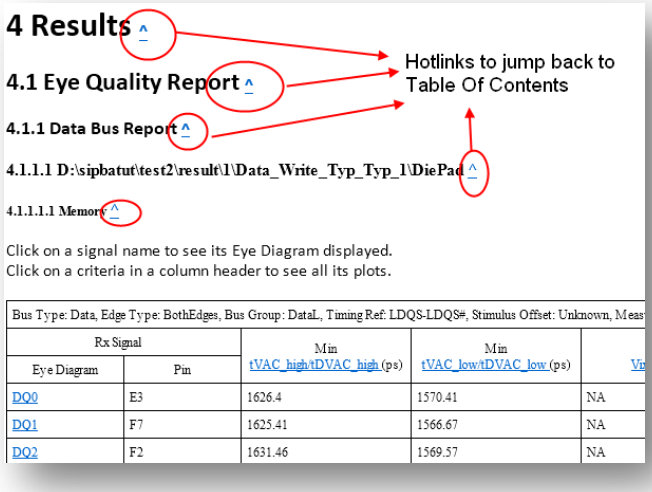
10.9.1 Table of Contents

The TOC provides a link to various sections in the report. To jump to a particular section of the report, click on the appropriate link in the Table of Contents.

Table of Contents

1 General Information
2 Simulation Setup
2.1 Rank Definition
2.2 Model Selection and Stimulus
2.2.1 Controller
2.2.2 Memory
2.3 Signal Connectivity
2.4 Simulation Description
3 DDR Measurement Setup
3.1 AC and DC Logic Input Levels
3.2 Specs
3.2.1 Data Bus Write
3.2.2 Data Bus Read
3.3 Setup Derating Table
3.4 Hold Derating Table
4 Results
4.1 Timing Report
4.1.1 Data Bus Report
4.1.1.1 D:\sipbatut\test2\result\1\Data_Write_Typ_Typ_1\DiePad
4.1.1.1.1 Memory
4.1.1.2 D:\sipbatut\test2\result\1\Data_Write_Typ_Typ_2\DiePad
4.1.1.2.1 Memory
4.1.1.3 D:\sipbatut\test2\result\1\Data_Read_Typ_Typ_3\DiePad
4.1.1.3.1 Controller
4.1.1.4 D:\sipbatut\test2\result\1\Data_Read_Typ_Typ_4\DiePad
4.1.1.4.1 Controller
4.1.2 Worst Case Summary
5 Appendix
5.1 JEDEC DDR Measurement Definitions
5.2 Description of Abbreviations

To enable easy navigation, each section of the report has a link to bring you back to the Table of Contents.



4 Results ^

4.1 Eye Quality Report ^

4.1.1 Data Bus Report ^

4.1.1.1 D:\sipbatut\test2\result\1\Data_Write_Typ_Typ_1\DiePad ^

4.1.1.1.1 Memory ^

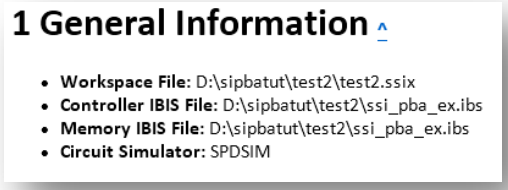
Click on a signal name to see its Eye Diagram displayed.
Click on a criteria in a column header to see all its plots.

Bus Type: Data, Edge Type: BothEdges, Bus Group: DataL, Timing Ref: LDQS-LDQS#, Stimulus Offset: Unknown, Meas

Rx Signal		Min tVAC_high/tDVAC_high (ps)	Min tVAC_low/tDVAC_low (ps)	Vi
Eye Diagram	Pin			
DQ0	E3	1626.4	1570.41	NA
DQ1	F7	1625.41	1566.67	NA
DQ2	F2	1631.46	1569.57	NA

10.9.2 Section 1 – General Information

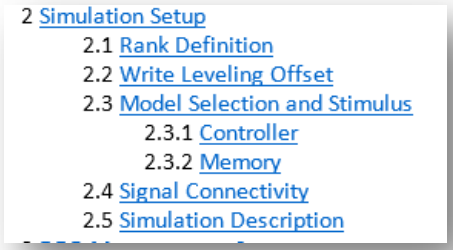
This section includes information about the design files, including the names and locations of the IBIS models used in the simulation.



1 General Information ^

- **Workspace File:** D:\sipbatut\test2\test2.ssix
- **Controller IBIS File:** D:\sipbatut\test2\ssi_pba_ex.ibs
- **Memory IBIS File:** D:\sipbatut\test2\ssi_pba_ex.ibs
- **Circuit Simulator:** SPDSIM

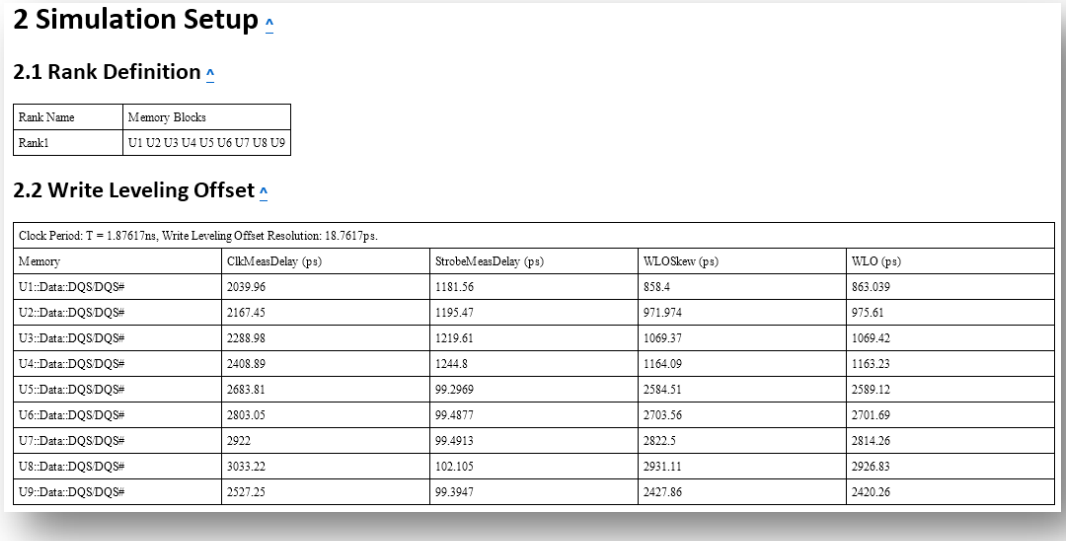
10.9.3 Section 2 – Simulation Setup



2 Simulation Setup
- 2.1 Rank Definition
- 2.2 Write Leveling Offset
- 2.3 Model Selection and Stimulus
  - 2.3.1 Controller
  - 2.3.2 Memory
- 2.4 Signal Connectivity
- 2.5 Simulation Description

The simulation information is listed, including simulator (HSPICE or SPDSIM), Power (Ideal or Non-ideal) Bus identification, the IBIS model selection and Stimulus for Controller and Memory, the Signal Connectivity, and simulation description.

If the WLO/ClkMeasDelay option was selected in the Analysis Options dialog box, the Write Leveling Offset section is added to the report. This section lists the delay values for each memory component, as calculated in the Write Leveling Offset dialog box.



2 Simulation Setup ^

2.1 Rank Definition ^

Rank Name	Memory Blocks
Rank1	U1 U2 U3 U4 U5 U6 U7 U8 U9

2.2 Write Leveling Offset ^

Clock Period: T = 1.87617ns, Write Leveling Offset Resolution: 18.7617ps.				
Memory	ClkMeasDelay (ps)	StrobeMeasDelay (ps)	WLOSkew (ps)	WLO (ps)
U1::Data::DQS/DQS#	2039.96	1181.56	858.4	863.039
U2::Data::DQS/DQS#	2167.45	1195.47	971.974	975.61
U3::Data::DQS/DQS#	2288.98	1219.61	1069.37	1069.42
U4::Data::DQS/DQS#	2408.89	1244.8	1164.09	1163.23
U5::Data::DQS/DQS#	2683.81	99.2969	2584.51	2589.12
U6::Data::DQS/DQS#	2803.05	99.4877	2703.56	2701.69
U7::Data::DQS/DQS#	2922	99.4913	2822.5	2814.26
U8::Data::DQS/DQS#	3033.22	102.105	2931.11	2926.83
U9::Data::DQS/DQS#	2527.25	99.3947	2427.86	2420.26

For each controller and memory block, the report shows the OnDie Parasitics and Package parasitics model used, bus details, Stimulus Offset, and IO model used for each signal.

OnDie Parasitics: None; Package Parasitics: Pin RLC.							
Bus Group	Signal	Pin	Stimulus Pattern	Stimulus Offset (ns)	Transmit IO Model	Receive IO Model	Status

10.9.4 Section 3 – DDR Measurement Setup

This section lists all the details specified in the Generate Report dialog box (see sections ***10.4*** and ***10.5***). These include the AC and DC logic levels, Specifications for each data bus, and selected Derating Tables for Setup/ Hold measurements.

10.9.5 Section 4 – Results

Results for each of the report data types that were selected by the user are listed in tabular format.

10.9.5.1 Waveform Quality Report

The Waveform Quality report includes Maximum Overshoot and undershoot voltage, Maximum Overshoot and Undershoot Area (voltage-time) and Ringback margin for low and high logic levels, for each simulation for each bus.

4.1 Waveform Quality Report ^

4.1.1 Data Bus Report ^

4.1.1.1 D:\sipbatut\test2\result\1\Data_Write_Typ_Typ_1\DiePad ^

4.1.1.1.1 Memory ^

Click on a signal name to see its waveform displayed.
Click on a criteria in a column header to see all its plots.

Bus Type: Data, Edge Type: BothEdges, Bus Group: DataL, Timing Ref: LDQS-LDQS#, Stimulus Offset: Unknown, Measurement Range: [0ps, end].						
Rx Signal		Max Overshoot (mV)	Max OvershootArea (V-ns)	Max Undershoot (mV)	Max UndershootArea (V-ns)	
Waveform	Pin					
DQ0	E3	478.386	0.316066	498.582	0.34409	78
DQ1	F7	490.98	0.317543	509.596	0.344974	78
DQ2	F2	478.859	0.331887	485.879	0.357466	79
DQ3	F8	474.1	0.31383	486.495	0.341201	**77**
DQ4	H3	513.258	0.334189	531.841	0.358504	11
DQ5	H8	514.021	0.35449	524.578	**0.38257**	81
DQ6	G2	**522.416**	**0.355263**	**540.657**	0.378378	81
DQ7	H7	459.789	0.314397	478.431	0.337682	81
LDQS-LDQS#	F3, G3	NA	NA	NA	NA	N

Note: NA = Not Applicable; NMP = No Measurement Possible.

The worst-case values for each of these measurements for the bus are highlighted in BOLD, and a worst case summary sheet is listed below.

4.1.2 Worst Case Summary

Simulation Results:
Case 1: D:\Working\Bugs (SI-SHCN-1008)\01117867_SSI PBA SLA\SSI-09\demo\demo\result\1\Data_Write_Typ_Typ\DiePad
Case 2: D:\Working\Bugs (SI-SHCN-1008)\01117867_SSI PBA SLA\SSI-09\demo\demo\result\1\Data_Read_Typ_Typ\DiePad

Measurement	Max Overshoot (V)	Max OvershootArea (V-ns)	Max Undershoot (V)	Max UndershootArea (V-ns)	Min RBack_marginH (mV)	Min RBack_marginL (mV)	Max Power_Ripple (mVp-p)
Worst Value	0.00873821	0.000682472	0.00778104	0.0011464	624.219	621.936	0
Simulation Result	Case 1	Case 1	Case 1	Case 1	Case 1	Case 1	Case 1
Bus Group	DQ	DQ	DQ	DQ	DQ	DQ	DQ
Rx Signal (Waveform)	DQ1	DQ1	DQ0	DQ7	DQ1	DQ7	VDDQ
Cycle	4	4	4	3	3	2	

If the **Explicit Power and Ground Terminals** option is checked in the **Load IBIS** GUI of the Controller/Memory block (please refer to ***Section 2.3.2Controller / Memory Block*** for details), the Waveform Quality measurement will use the signal's own power and ground for the measurements of the Overshoot, OvershootArea, Undershoot, UndershootArea, and Power_Ripple.

4 Results

4.1 Waveform Quality Report

4.1.1 Data Bus Report

4.1.1.1 D:\Working\Bugs (SI-SHCN-1008)\01117867_SSI PBA SLA\SSI-09\demo\demo\result\2\Data_Write_Typ_Typ\DiePad

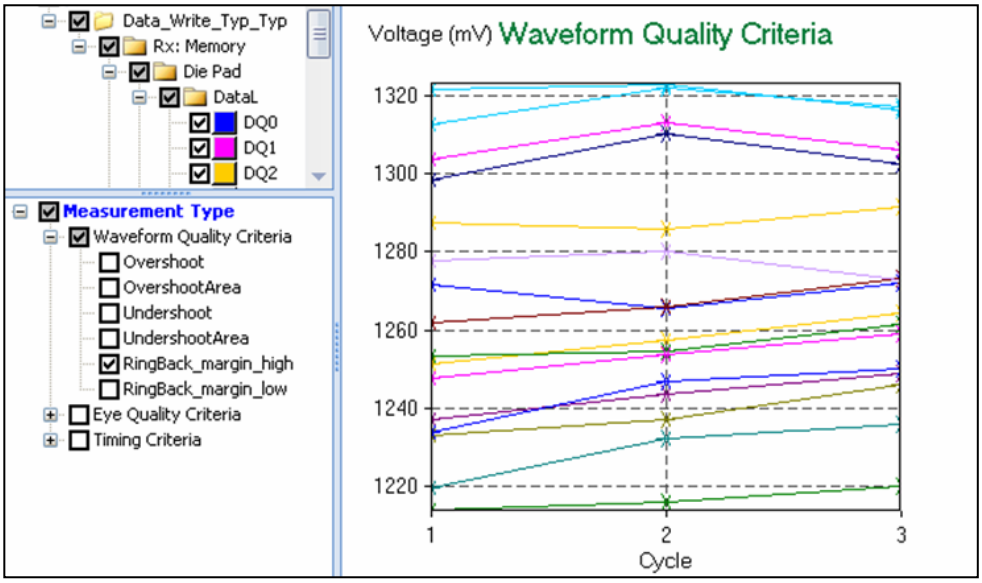
4.1.1.1.1 Memory

Click on a signal name to see its waveform displayed.
Click on a criteria in a column header to see all its plots.

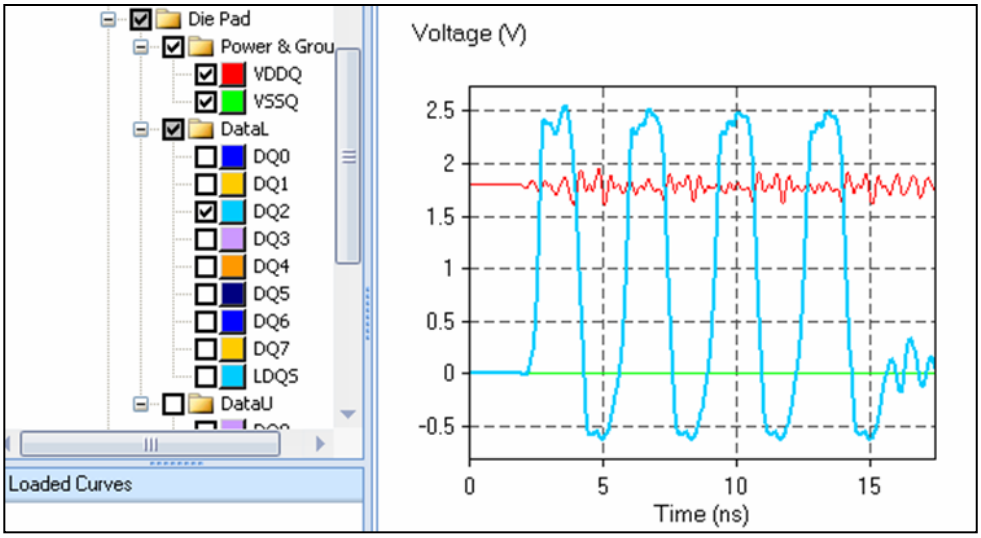
Bus Type: Data, Edge Type: BothEdges, Bus Group: DQ, Timing Ref: DQS-DQS#, Measurement Range: [3727.05ps, end]

Rx Signal		Max Overshoot (V)	Max OvershootArea (V-ns)	Max Undershoot (V)	Max UndershootArea (V-ns)	Min RBack_marginH (mV)	Min RBack_marginL (mV)	Max Power_Ripple (mVp-p)
Waveform	Pin							
DM_TDQS	B7	NMP	NMP	NMP	NMP	NMP	NMP	0
DQ0	B3	0.00840186	0.000448678	**0.00779651**	0.000873248	633.402	623.109	0
DQ1	C7	**0.00877261**	**0.000713627**	0.00109116	1.30255e-005	**624.319**	626.091	0
DQ2	C2	0.00807801	0.000403484	0.00723891	0.000325954	633.078	621.9	0
DQ3	C8	0.00793335	0.000403447	NMP	NMP	632.933	NMP	0
DQ4	E3	NMP	NMP	NMP	NMP	NMP	NMP	0

By clicking the column header link, the cycle-to-cycle variation for that value is plotted for the bus. For example, the Ringback margin (high) link opens this 2D Curve Data Window, showing the Ringback margin at each cycle for every signal in the bus.



Similarly, clicking the Signal Name link in the first (left) column of the data table opens a 2D Curve showing the time-domain waveform for that signal, with the low and high logic levels overlaid.


Die Pad
Power & Grou
VDDQ
VSSQ
DataL
DQ0
DQ1
DQ2
DQ3
DQ4
DQ5
DQ6
DQ7
LDQS
DataU
Loaded Curves
Voltage (V)
2.5
2
1.5
1
0.5
0
-0.5
0
5
10
15
Time (ns)

10.9.5.2 Eye Quality Report

The Eye Quality report has tabular data summarizing results, showing worst-case values in BOLD, and a worst case summary sheet is listed below.

4.2 Eye Quality Report

4.2.1 Data Bus Report

4.2.1.1 C:\SigritySamples\SpeedXP 12.0\SystemSI\Samples\Parallel Bus Analysis\Tutorial\result\1\Data_Write_Typ_Typ\DiePad

4.2.1.1.1 Memory

Click on a signal name to see its Eye Diagram displayed.
Click on a criteria in a column header to see all its plots.

Bus Type: Data, Edge Type: BothEdges, Bus Group: DataL, Timing Ref: LDQS, Measurement Range: [3333ps, end]						
Rx Signal		Min tVAC_high\tDVAC_high (ps)	Min tVAC_low\tDVAC_low (ps)	Max Vix_rise (mV)	Max Vix_fall (mV)	Min ApertureWidth (ps)
Eye Diagram	Pin					
DQ0	H1	1558.18	1654.64	NA	NA	1561.3
DQ1	H3	1540.41	1657.04	NA	NA	1551.34
DQ2	G2	**1534.38**	1658.05	NA	NA	**1544.49**
DQ3	F1	1534.62	1663.66	NA	NA	1544.61
DQ4	H7	1543.23	1651.62	NA	NA	1553.27
DQ5	H9	1536.41	1662.03	NA	NA	1547.48
DQ6	G8	1568.79	**1631.48**	NA	NA	1575.51
DQ7	F9	1547.29	1643.99	NA	NA	1558.28
All Signals		NA	NA	NA	NA	1455.41
LDQS	F7	1727.03	1506.7	NA	NA	1513.83

Note: NA = Not Applicable; NMP = No Measurement Possible.

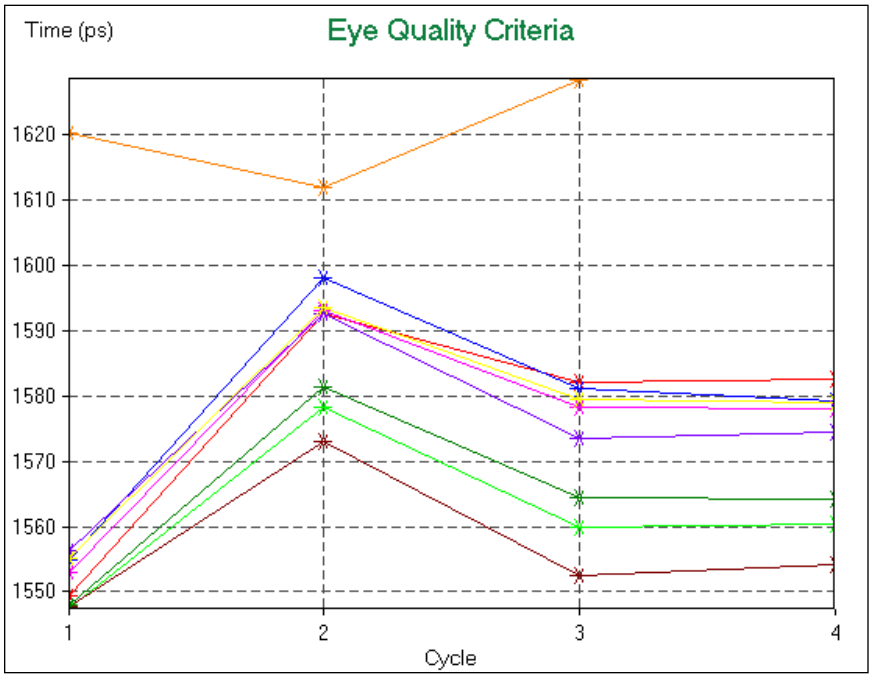
4.2.2 Worst Case Summary

Simulation Results:
Case 1: D:\document update\SSI\SSI_15\case_temp9\result\1\Data_Write_Typ_Typ\DiePad
Case 2: D:\document update\SSI\SSI_15\case_temp9\result\1\Data_Read_Typ_Typ\DiePad

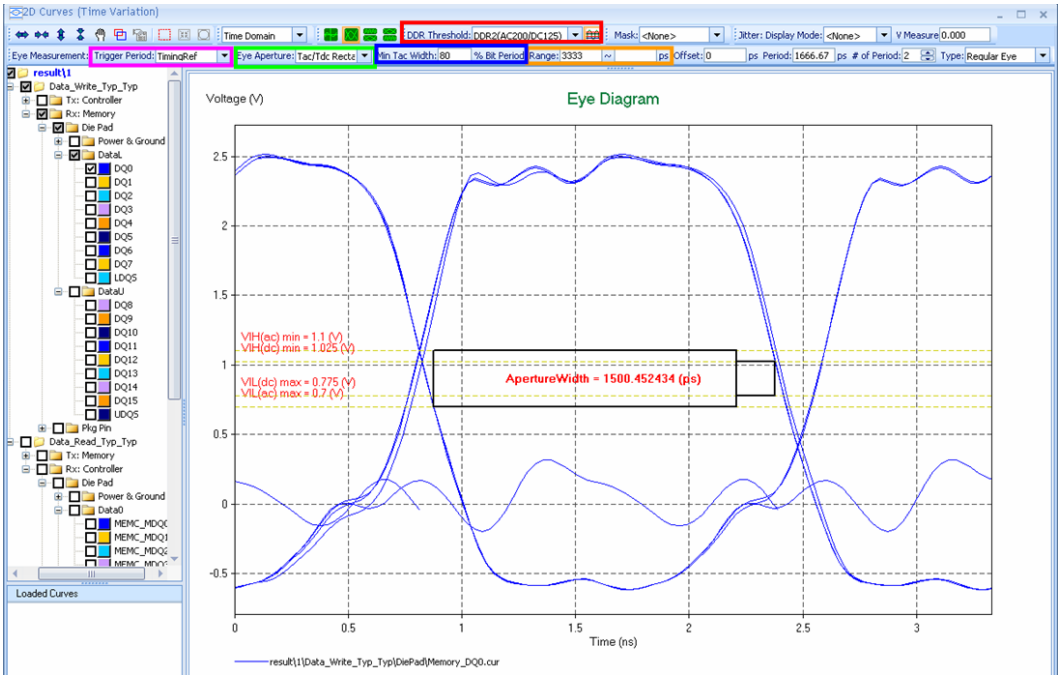
Measurement	Min tVAC_high\tDVAC_high (ps)	Min tVAC_low\tDVAC_low (ps)	Max Vix_rise (mV)	Max Vix_fall (mV)	Min ApertureWidth (ps)	
Worst Value	1395.04	1613.03			1351.85	1336.16
Simulation Result	Case 2	Case 1			Case 2	Case 2
Bus Group	Data0	DataU			Data1	Data1
Rx Signal (Waveform / Eye Diagram)	MEMC_MDQ0	DQ12			MEMC_MDQ13	All Signals
Cycle	2	1				

The V_{ix} parameters are Eye quality metrics for DDR3 that assume a Differential clock signal. The measurement is not applicable for DDR2 systems with a single-ended strobe.

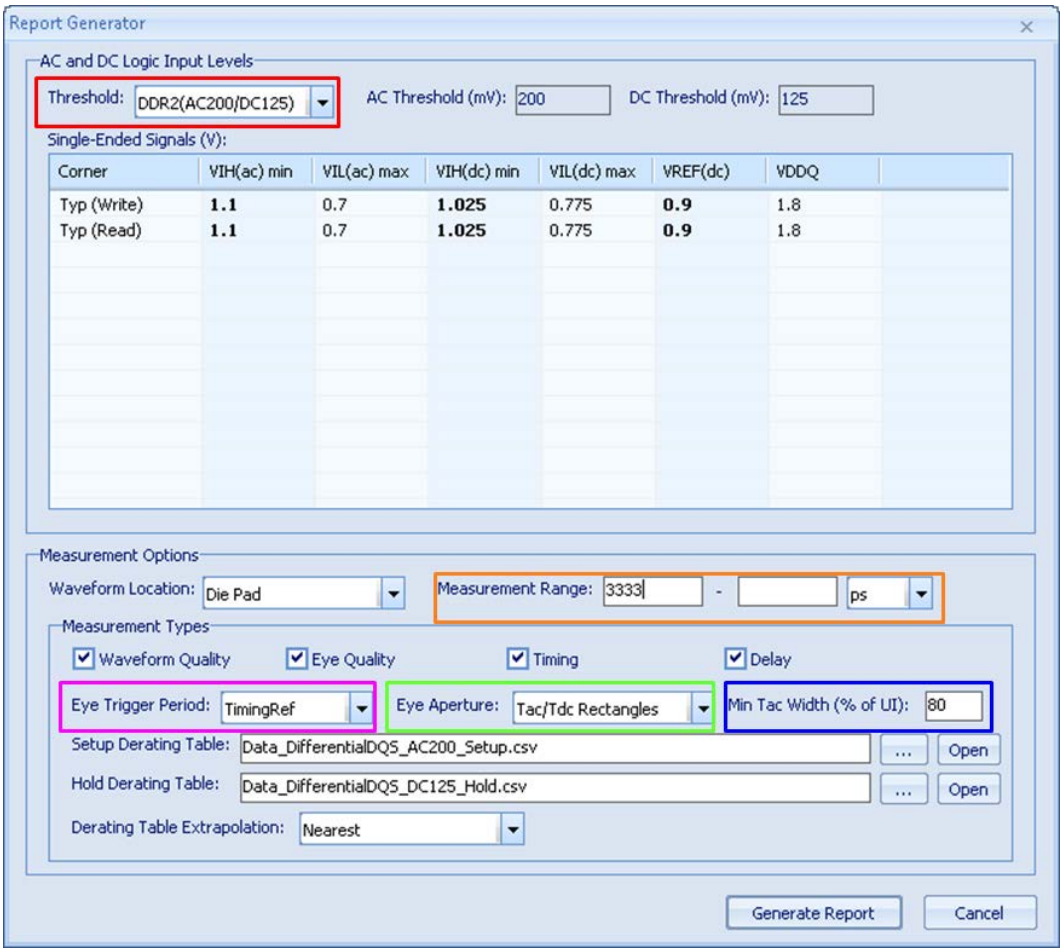
As with the Waveform Quality report, clicking the Column Header for each measurement will open a 2D curve showing the Cycle-to-cycle variation of that parameter. This is the `tVAC_high/tDVAC_low` parameter.



Clicking the Signal name in the first column of the table, opens the Eye diagram for that signal, with the Eye aperture plotted with the data on the plot.



The settings in the marked parts of the above figure are set according to the settings in **Report Generator** as shown below.



10.9.5.3 Timing Report

The timing report has a comprehensive tabular listing of JEDEC setup and hold measurements. Raw and slew-rate adjusted (derated) values are listed, with worst-case values highlighted in bold.

4.3 Timing Report

4.3.1 Data Bus Report

4.3.1.1 C:\SigritySamples\SpeedXP 12.0\SystemSI\Samples\Parallel Bus Analysis\Tutorial\result\1\Data_Write_Typ_Typ\DiePad

4.3.1.1.1 Memory

Click on a signal name to see its waveform displayed.
Click on a criteria in a column header to see all its plots.

Bus Type: Data, Edge Type: BothEdges, Bus Group: DataL, Timing Ref: LDQS, Measurement Range: [3333ps, end]
Rx Specs: Min Receive Setup: 0.416666ns, Min Receive Hold: 0.416666ns.

Rx Signal		[Min, Max] SlewRateTimingRef (V/ns)	Min raw_tDS (ps)	[Min, Max] SlewRateSetup (V/ns)	Max Delta_tDS (ps)	Min adj_tDS (ps)	Min tDS_margin (ps)	Min raw_tDH (ps)
Waveform	Pin							
DQ0	H1	[5.01056, 9.43198]	733.655	[5.21437, 6.85652]	OOR	NMP	316.989	766.909
DQ1	H3	[5.01056, 9.43198]	751.597	[4.83197, 7.80428]	OOR	NMP	334.931	754.59
DQ2	G2	[5.01056, 9.43198]	722.177	[4.91982, 6.70377]	OOR	NMP	305.511	771.214
DQ3	F1	[5.01056, 9.43198]	724.255	[4.99714, 6.98393]	OOR	NMP	307.589	771.037
DQ4	H7	[5.01056, 9.43198]	747.119	[5.07409, 6.65814]	OOR	NMP	330.453	751.541
DQ5	H9	[5.01056, 9.43198]	**717.218**	[4.8304, 6.76983]	OOR	NMP	**300.552**	782.413
DQ6	G8	[5.01056, 9.43198]	837.319	[5.21227, 6.46158]	OOR	NMP	420.653	**654.741**
DQ7	F9	[5.01056, 9.43198]	718.343	[4.7834, 7.142]	OOR	NMP	301.677	783.691
LDQS	F7	NA	NA	NA	NA	NA	NA	NA

4.3.1.2 D:\document update\SSI\SSI_15\case_temp9\result\1\Data_Read_Typ_Typ\DiePad

4.3.1.2.1 Controller

Click on a signal name to see its waveform displayed.
Click on a criteria in a column header to see all its plots.

Bus Type: Data, Edge Type: BothEdges, Bus Group: Data0, Timing Ref: MEMC_MDQS0, Measurement Range: [3333ps, end]
Rx Specs: Max Receive Skew (+): 0.416666ns, Max Receive Skew (-): 0.416666ns.

Rx Signal		[Min, Max] SlewRateTimingRef (V/ns)	Min raw_tDS (ps)	[Min, Max] SlewRateSetup (V/ns)	Max Delta_tDS (ps)	Min adj_tDS (ps)	Min raw_tDH (ps)	[Min, Max] SlewRateHold (V/ns)	Max Delta_tDH (ps)	Min adj_tDH (ps)	Max ReadSkew (ps)	Min ReadSkew_margin (ps)
Waveform	Pin											
MEMC_MDQ0	AF16	[4.97115, 11.7245]	54.1279	[5.318, 5.4601]	OOR	NMP	15.3585	[3.33427, 8.48676]	OOR	NMP	**32.4492**	**384.217**
MEMC_MDQ1	AE17	[4.97115, 11.7245]	35.3475	[5.87836, 8.72584]	OOR	NMP	**0.668045**	[5.05105, 9.53359]	OOR	NMP	-16.9096	399.756

4.3.2 Worst Case Summary

Measurement	[Min, Max] SlewRateTimingRef (V/ns)	Min raw_tDS (ps)	[Min, Max] SlewRateSetup (V/ns)	Max Delta_tDS (ps)	Min adj_tDS (ps)	Min tDS_margin (ps)	Min raw_tDH (ps)
Worst Value		717.218				300.552	654.741
Bus Group		DataL				DataL	DataL
Rx Signal (Waveform)		DQ5				DQ5	DQ6
Cycle		1.5				1.5	2

Out of Range are typically undefined values (Pink) in the derating tables:

Delta_tDS (ps) -- AC200 tDS Derating Values for DDR2-667/800/1066									
		DQS, DQS# Differential Slew Rate (V/ns)							
		4.0 V/ns	3.0 V/ns	2.0 V/ns	1.8 V/ns	1.6 V/ns	1.4 V/ns	1.2 V/ns	1.0 V/ns	0.8 V/ns
		Delta_tDS	Delta_tDS	Delta_tDS	Delta_tDS	Delta_tDS	Delta_tDS	Delta_tDS	Delta_tDS	Delta_tDS
DQ Slew Rate (V/ns)	2.0	100	100	100						
	1.5	67	67	67	79					
	1.0	0	0	0	12	24				
	0.9		-5	-5	7	19	31			
	0.8			-13	-1	11	23	35		
	0.7				-10	2	14	26	38	
	0.6					-10	2	14	26	38
	0.5						-24	-12	0	12
	0.4							-52	-40	-28

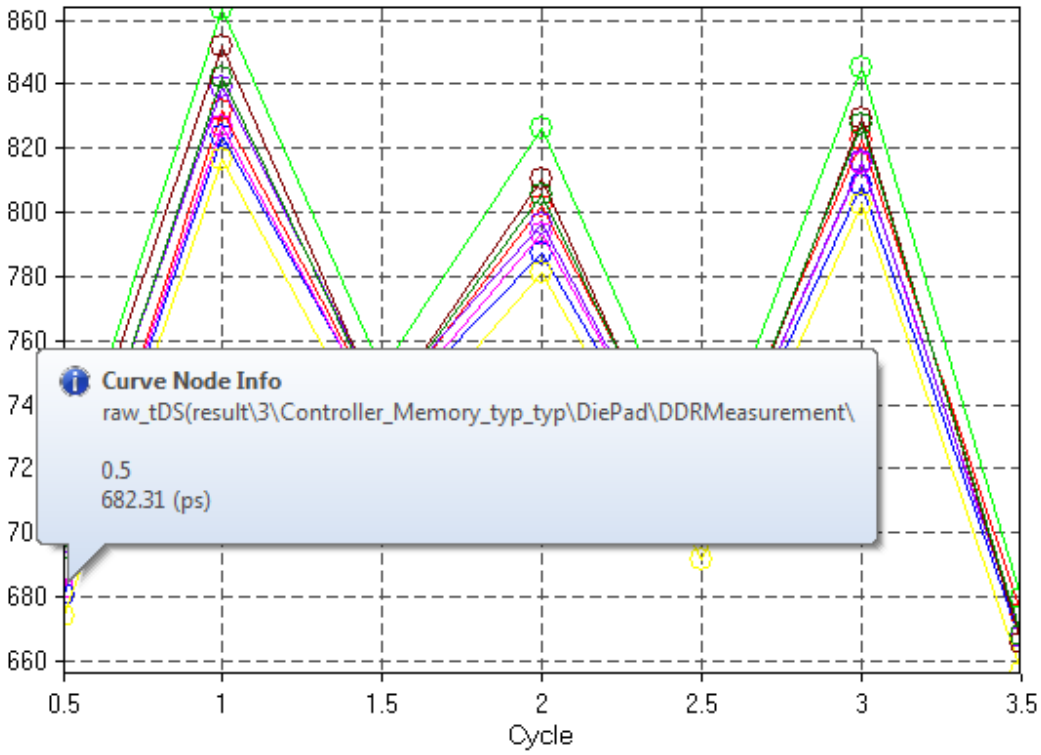
- If the selection for the **Derating Table Extrapolation** is **Nearest**, a note will be added below the derating table:

  Note: Derating Table Extrapolation: Nearest.

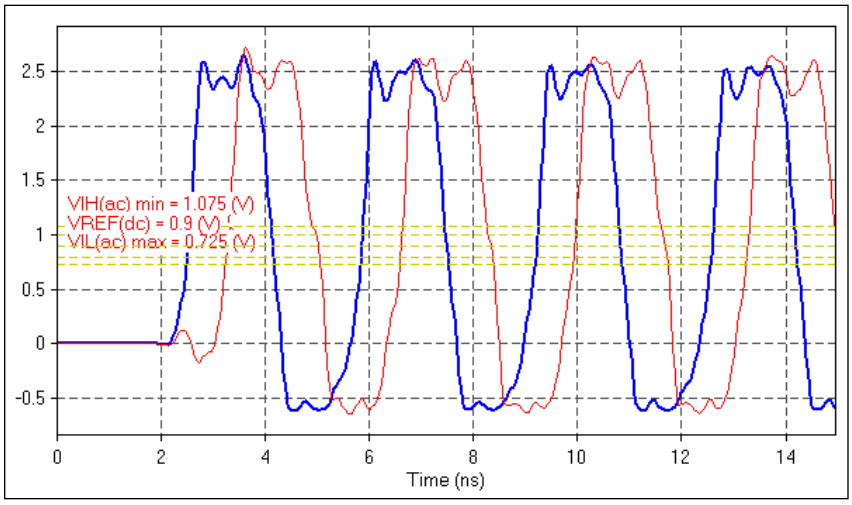
- If the selection for the **Derating Table Extrapolation** is **None**, a note will be added below the derating table:

  Note: cell contents shaded in red are defined as 'not supported'.

Clicking any of the column headers opens a cycle-to-cycle plot in the 2D Curve window, showing the variation of that parameter for each signal in the bus. This the raw tDS (Setup Time), with the mouse Autotip showing the data for one of the DQ lines at the first cycle. Time periods where a data line does not have a transition will not have data for that cycle. Note that the "raw" prefix indicates that the measurement is taken directly from the raw waveforms that were produced.



Clicking the Signal name in the first column of the table opens the Time domain plot for that signal, with its clock/ strobe signal.



10.9.5.4 Timing Report – Strobe and Clock

This report is available only for Data bus Write simulations. In case of Data bus has a clock signal defined and connected, the timing margin between the clock signal and the data strobe signal is measured and include in this report.

4.4 Timing Report - Strobe and Clock ^

4.4.1 Data Bus Report ^

4.4.1.1 D:\sipbatut\dqsclk_copy_copy\dqsclk_copy_copy\result\1\Data_Write_Typ_Typ\DiePad ^

4.4.1.1.1 Memory1 ^

Click on a signal name to see its waveform displayed.
Click on a criteria in a column header to see all its plots.

Bus Type: Data, Clock: CK-CKB, Measurement Range: [3411.54ps, end]
Rx Specs: Min Strobe Setup: 312.5ps, Min Strobe Hold: 312.5ps, Max Strobe/Clock Skew(+): 312.5ps, Max Strobe/Clock Skew(-): 312.5ps.

Rx Signal		Min tDSS (ps)	Min tDSS_margin (ps)	Min tDSH (ps)	Min tDSH_margin (ps)	|Max| tDQSS (ps)	Min tDQSS_margin (ps)
Waveform	Pin						
DQSL-DQSLB	F3, G3	**827.688**	**515.188**	1665.4	1352.9	**429.763**	**-117.263**
DQSU-DQSUB	C7, B7	827.742	515.242	**1665.34**	**1352.84**	417.95	-105.45
CK-CKB	J7, K7	NA	NA	NA	NA	NA	NA

Note: NA = Not Applicable; NMP = No Measurement Possible.

10.9.5.5 Timing Report - Worst Case Condition

If worst case Timing Reference waveforms exist, the DDR report will have additional Timing Report tables for those cases. To generate data for these reports, SystemSI simply shifts the Timing Reference waveform according to the parameters set up in the Timing Budget form, and re-measures all the report criteria. This enables the worst case timing margins to be determined from a single comprehensive time domain simulation run, saving significant simulation time. The shift applied to the Timing Reference signal is documented in each Worst Case report table, and the appropriate shifted waveforms will appear when cross probed from these tables.

1. **Timing Report - Worst Case Setup Condition**

4.4 Timing Report - Worst Case Setup Condition

4.4.1 Data Bus Report

4.4.1.1 C:\SigritySamples\SpeedXP 12.0\SystemSI\Samples\Parallel Bus Analysis\Tutorial\result\1\Data_Write_Typ_Typ\DiePad

4.4.1.1.1 Memory

Click on a signal name to see its waveform displayed.
Click on a criteria in a column header to see all its plots.

Bus Type: Data, Edge Type: BothEdges, Bus Group: DataL, Timing Ref: LDQS, Measurement Range: [3333ps, end]
Rx Specs: Min Receive Setup: 0.416666ns, Min Receive Hold: 0.416666ns. TimingRef was shifted left by 255.378 ps for the Worst Case Setup Condition.

Rx Signal		[Min, Max] SlewRateTimingRef (V/ns)	Min raw_tDS (ps)	[Min, Max] SlewRateSetup (V/ns)	Max Delta_tDS (ps)	Min adj_tDS (ps)	Min tDS_margin (ps)	Min raw_tDH (ps)
Waveform	Pin							
DQ0	H1	[4.80582, 9.51268]	477.87	[5.21437, 6.85652]	NMP	NMP	61.2037	1023.73
DQ1	H3	[4.80582, 9.51268]	495.883	[4.83197, 7.80428]	NMP	NMP	79.2166	1011.41
DQ2	G2	[4.80582, 9.51268]	466.392	[4.91982, 6.70377]	NMP	NMP	49.7257	1028.04
DQ3	F1	[4.80582, 9.51268]	468.47	[4.99714, 6.98393]	NMP	NMP	51.8039	1027.86
DQ4	H7	[4.80582, 9.51268]	491.334	[5.07409, 6.65814]	NMP	NMP	74.6676	1008.36
DQ5	H9	[4.80582, 9.51268]	**461.433**	[4.8304, 6.76983]	NMP	NMP	**44.7668**	1039.24
DQ6	G8	[4.80582, 9.51268]	581.534	[5.21227, 6.46158]	NMP	NMP	164.868	**911.563**
DQ7	F9	[4.80582, 9.51268]	462.557	[4.7834, 7.142]	NMP	NMP	45.8912	1040.51
LDQS	F7	NA	NA	NA	NA	NA	NA	NA

Note: NA = Not Applicable; NMP = No Measurement Possible; OOR = Out of Range.

4.4.2 Worst Case Summary

Measurement	[Min, Max] SlewRateTimingRef (V/ns)	Min raw_tDS (ps)	[Min, Max] SlewRateSetup (V/ns)	Max Delta_tDS (ps)	Min adj_tDS (ps)	Min tDS_margin (ps)	Min raw_tDH (ps)
Worst Value		455.944				39.2775	911.563
Bus Group		DataU				DataU	DataL
Rx Signal (Waveform)		DQ11				DQ11	DQ6
Cycle		1.5				1.5	2

2. Timing Report - Worst Case Hold Condition

4.5 Timing Report - Worst Case Hold Condition

4.5.1 Data Bus Report

4.5.1.1 D:\My work\Documents for System SI\tutorial\Examples\Chapter4_2\result\testqq\Data_Write_Typ_Typ\DiePad

4.5.1.1.1 Memory

Click on a signal name to see its waveform displayed.
Click on a criteria in a column header to see all its plots.

Bus Type: Data, Edge Type: BothEdges, Bus Group: DataL, Timing Ref: LDQS, Number of Cycles Ignored: 0
Rx Specs: Min Receive Setup: 0.416666ns, Min Receive Hold: 0.416666ns. TimingRef was shifted right by 253.918 ps for the Worst Case Hold Condition.

Rx Signal		[Min, Max] SlewRateTimingRef (V/ns)	Min raw_tDS (ps)	[Min, Max] SlewRateSetup (V/ns)	Max Delta_tDS (ps)	Min adj_tDS (ps)	Min tDS_margin (ps)	Min raw_tDH (ps)	[Min, Max] SlewRateHold (V/ns)	Max Delta_tDH (ps)	Min adj_tDH (ps)	Min tDH_margin (ps)
Waveform	Pin											
DQ0	H1	[4.62188, 6.91083]	1041.02	[6.29443, 6.99742]	NMP	NMP	624.355	483.588	[5.44916, 6.451]	NMP	NMP	66.9221
DQ1	H3	[4.62188, 6.91083]	1066.9	[6.42991, 7.42108]	NMP	NMP	650.235	**452.286**	[4.86771, 7.051]	NMP	NMP	**35.6201**
DQ2	G2	[4.62188, 6.91083]	1041.99	[5.66115, 6.83466]	NMP	NMP	625.328	483.556	[4.61503, 6.52821]	NMP	NMP	66.8897
DQ3	F1	[4.62188, 6.91083]	1043.14	[5.92535, 6.85653]	NMP	NMP	626.475	481.283	[4.74608, 6.53304]	NMP	NMP	64.6174
DQ4	H7	[4.62188, 6.91083]	1060.3	[5.52516, 7.05846]	NMP	NMP	643.637	467.551	[4.42351, 6.65107]	NMP	NMP	50.8851
DQ5	H9	[4.62188, 6.91083]	**1030.22**	[5.74919, 6.76733]	NMP	NMP	**613.55**	495.567	[4.9757, 6.46169]	NMP	NMP	78.901
DQ6	G8	[4.62188, 6.91083]	1064.18	[6.15695, 7.30063]	NMP	NMP	647.515	464.747	[4.7879, 6.99058]	NMP	NMP	48.0806
DQ7	F9	[4.62188, 6.91083]	1047.62	[6.33303, 7.23075]	NMP	NMP	630.952	478.717	[5.55791, 6.83187]	NMP	NMP	62.0506
LDQS	F7	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

Note: NA = Not Applicable, NMP = No Measurement Possible, OOR = Out of Range

4.5.2 Worst Case Summary

Simulation Results:
Case 1: D:\My work\Documents for System SI\tutorial\Examples\Chapter4_2\result\testqq\Data_Write_Typ_Typ\DiePad
Case 2: D:\My work\Documents for System SI\tutorial\Examples\Chapter4_2\result\testqq\Data_Read_Typ_Typ\DiePad

Measurement	[Min, Max] SlewRateTimingRef (V/ns)	Min raw_tDS (ps)	[Min, Max] SlewRateSetup (V/ns)	Max Delta_tDS (ps)	Min adj_tDS (ps)	Min tDS_margin (ps)	Min raw_tDH (ps)	[Min, Max] SlewRateHold (V/ns)	Max Delta_tDH (ps)	Min adj_tDH (ps)	Min tDH_margin (ps)
Worst Value		1016.62				599.957	446.269				29.6031
Simulation Result		Case 1				Case 1	Case 1				Case 1
Bus Group		DataU				DataU	DataU				DataU
Rx Signal (Waveform)		DQ14				DQ14	DQ10				DQ10
Cycle		2.5				2.5	3				3

3. Timing Report - Worst Case Transmit Skew (+) Condition

4.6 Timing Report - Worst Case Transmit Skew (+) Condition

4.6.1 Data Bus Report

4.6.1.2 D:\document update\SSI\SSI_15\case_temp9\result\1\Data_Read_Typ_Typ\DiePad

4.6.1.2.1 Controller

Click on a signal name to see its waveform displayed.
Click on a criteria in a column header to see all its plots.

Bus Type: Data, Edge Type: BothEdges, Bus Group: Data0, Timing Ref: MEMC_MDQS0, Measurement Range: [3333ps, end]
Rx Specs: Max Receive Skew (+): 0.416666ns, Max Receive Skew (-): 0.416666ns. TimingRef was shifted left by 162.423 ps for the Worst Case Transmit Skew (+) Condition.

Rx Signal		[Min, Max] SlewRateTimingRef (V/ns)	Min raw_tDS (ps)	[Min, Max] SlewRateSetup (V/ns)	Max Delta_tDS (ps)	Min adj_tDS (ps)	Min raw_tDH (ps)	[Min, Max] SlewRateHold (V/ns)	Max Delta_tDH (ps)	Min adj_tDH (ps)	Max ReadSkew (ps)	Min ReadSkew_margin (ps)
Waveform	Pin											
MEMC_MDQ0	AF16	[4.99257, 11.6996]	**1286.81**	[8.61033, 8.95511]	NMP	NMP	**121.379**	[5.318, 8.48676]	NMP	NMP	**195.064**	**221.602**
MEMC_MDQ1	AF17	[4.99257, 11.6996]	1332.52	[9.13232,	NMP	NMP	133.055	[5.87836,	NMP	NMP	150.171	266.485

4. Timing Report - Worst Case Transmit Skew (-) Condition

4.7 Timing Report - Worst Case Transmit Skew (-) Condition

4.7.1 Data Bus Report

4.7.1.2 D:\document update\SSI\SSI_15\case_temp9\result\1\Data_Read_Typ_Typ\DiePad

4.7.1.2.1 Controller

Click on a signal name to see its waveform displayed.
Click on a criteria in a column header to see all its plots.

Bus Type: Data, Edge Type: BothEdges, Bus Group: Data0, Timing Ref: MEMC_MDQS0, Measurement Range: [3333ps, end]
Rx Specs: Max Receive Skew (+): 0.416666ns, Max Receive Skew (-): 0.416666ns. TimingRef was shifted right by 191.814 ps for the Worst Case Transmit Skew (-) Condition.

Rx Signal		[Min, Max] SlewRateTimingRef (V/ns)	Min raw_tDS (ps)	[Min, Max] SlewRateSetup (V/ns)	Max Delta_tDS (ps)	Min adj_tDS (ps)	Min raw_tDH (ps)	[Min, Max] SlewRateHold (V/ns)	Max Delta_tDH (ps)	Min adj_tDH (ps)	|Max| ReadSkew (ps)	Min ReadSkew_margin (ps)
Waveform	Pin											
MEMC_MDQ0	AF16	[4.96353, 12.0335]	136.087	[2.95668, 8.95511]	NMP	NMP	**1271.79**	[5.318, 8.48676]	NMP	NMP	-207.72	208.946
MEMC_MDQ1	AE17	[4.96353, 12.0335]	167.315	[4.83306, 9.53359]	NMP	NMP	1273.76	[5.87836, 8.90327]	NMP	NMP	-209.241	207.425
MEMC_MDQ2	AH17	[4.96353, 12.0335]	140.206	[4.30116, 8.33027]	NMP	NMP	1296.26	[5.30849, 7.56909]	NMP	NMP	-187.661	229.005

You can click any signals or parameter to open the 2D Curve or Criteria, and the shifted value of Timing reference is shown on the top of the sheet.

Bus Type: Data, Edge Type: BothEdges, Bus Group: DataL, Timing Ref: LDQS, Measurement Range: [3333ps, end]
Rx Specs: Min Receive Setup: 0.416666ns, Min Receive Hold: 0.416666ns. TimingRef was shifted right by 253.738 ps for the Worst Case Hold Condition.

Rx Signal		[Min, Max] SlewRateTimingRef (V/ns)	Min raw_tDS (ps)	[Min, Max] SlewRateSetup (V/ns)	Max Delta_tDS (ps)	Min adj_tDS (ps)	Min tDS_margin (ps)	Min raw_tDH (ps)	[Min, Max] SlewRateHold (V/ns)	Max Delta_tDH (ps)	Min adj_tDH (ps)	Min tDH_margin (ps)
Waveform	Pin											
DQ0	H1	[4.65741, 9.62724]	983.122	[5.20652, 6.85652]	NMP	NMP	566.456	510.1	[5.61526, 6.7903]	NMP	NMP	93.4336
DQ1	H3	[4.65741, 9.62724]	1001.4	[4.83197, 7.80428]	NMP	NMP	584.731	497.175	[5.00619, 6.87401]	NMP	NMP	80.5089

4.6 Timing Report - Worst Case Transmit Skew (+) Condition

4.6.1 Data Bus Report

4.6.1.2 D:\document update\SSI\SSI_15\case_temp9\result\1\Data_Read_Typ_Typ\DiePad

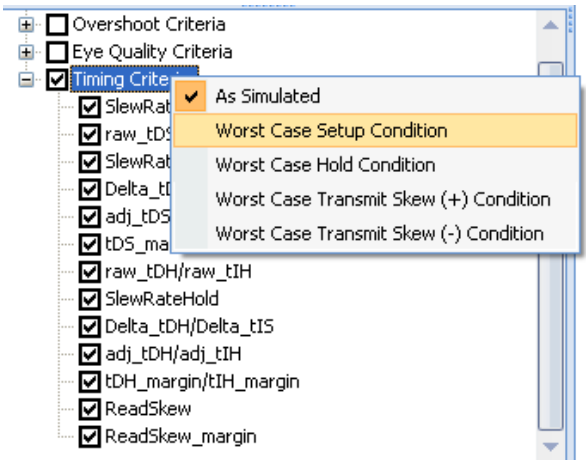
4.6.1.2.1 Controller

Click on a signal name to see its waveform displayed.
Click on a criteria in a column header to see all its plots.

Bus Type: Data, Edge Type: BothEdges, Bus Group: Data0, Timing Ref: MEMC_MDQS0, Measurement Range: [3333ps, end]
Rx Specs: Max Receive Skew (+): 0.416666ns, Max Receive Skew (-): 0.416666ns. TimingRef was shifted left by 162.423 ps for the Worst Case Transmit Skew (+) Condition.

Rx Signal		[Min, Max] SlewRateTimingRef (V/ns)	Min raw_tDS (ps)	[Min, Max] SlewRateSetup (V/ns)	Max Delta_tDS (ps)	Min adj_tDS (ps)	Min raw_tDH (ps)	[Min, Max] SlewRateHold (V/ns)	Max Delta_tDH (ps)	Min adj_tDH (ps)	|Max| ReadSkew (ps)	Min ReadSkew_margin (ps)
Waveform	Pin											
MEMC_MDQ0	AF16	[4.99257, 11.6996]	**1286.81**	[8.61033, 8.95511]	NMP	NMP	**121.379**	[5.318, 8.48676]	NMP	NMP	**195.064**	**221.602**
MEMC_MDQ1	AE17	[4.99257, 11.6996]	1332.52	[9.13232,	NMP	NMP	123.055	[5.87836,	NMP	NMP	150.171	266.495

NOTE!	Right-click **Timing Criteria** in the 2D Curves (DDR Measurement) window, and a pop-up menu lists 5 options available for selection.



10.9.5.6 DQ Mask Report

DQ Mask report is generated on Data Write simulations for DDR4 measurements only. For the DQ Mask to be generated, at least one simulation must have the Stimulus Offset set to the *Default* or *Ideal*.

4.2.1 Data Bus Report ^

4.2.1.1 D:\sipbatut\ddr4_welem_vrm_copy\ddr4_welem_vrm_copy\result\1\Data_Write_Typ_Typ\DiePad ^

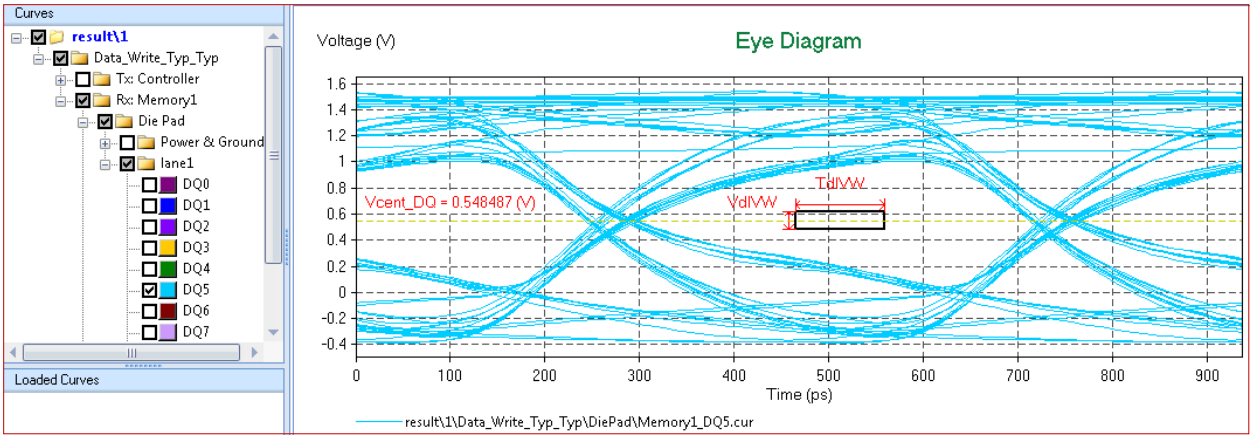
4.2.1.1.1 Memory1 ^

Click on a signal name to see the DQ Mask displayed.

Bus Type: Data, Edge Type: BothEdges, Bus Group: lane1, Timing Ref: DQS_t-DQS_c, Stimulus Offset: Default, Measurement Range: [3369.71ps, end], UI: 468.824ps.
Rx DQ Mask Specs: VdIVW: 136mV, TdIVW: 93.7647ps.
Rx Timing Specs: Max tDQS2DQ: 79.7ps, Max tDQ2DQ: 46.8824ps, Min VIHL_AC: 186mV, Min TdIPW: 271.918ps, Min SlewRate_Mask: 1V/ns, Max SlewRate_Mask: 9V/ns, Min SlewRate_AC_Swing: 0.2V/ns, Max SlewRate_AC_Swing: 9V/ns.

Rx Signal		Vcent_DQ (V)	DQ Compliance Mask	Min Jitter_margin (ps)	Min Noise_margin (mv)	|Max| tDQS2DQ (ps)	Max tDQ2DQ (ps)	Min VIHL_AC (mV)	Min TdIPW (ps)	Min SlewRate_Mask (V/ns)	Max SlewRate_Mask (V/ns)	Min SlewRate_AC_Swing (V/ns)	Max SlewRate_AC_Swing (V/ns)
DQ Mask	Pin												
DQ0	C2	0.549947	Pass	150.358	235.579	-37.0591	NA	607.159	432.301	3.16195	6.30698	3.07902	6.24484
DQ1	B7	0.546425	Pass	**141.314**	216.041	**-47.8327**	NA	568.082	414.653	**2.86127**	6.22312	**2.81228**	6.17491
DQ2	D3	0.545792	Pass	141.967	218.575	-42.6529	NA	573.151	**414.438**	3.06302	6.15219	2.93573	6.12392
DQ3	D7	0.546247	Pass	142.026	218.487	-42.8909	NA	572.975	414.525	3.04976	6.17029	2.93245	6.14046
DQ4	D2	0.545106	Pass	142.526	**215.72**	-47.1462	NA	**567.439**	416.238	2.87598	**6.35295**	2.82585	**6.30357**
DQ5	D8	0.545826	Pass	142.35	216.13	-46.2657	NA	568.26	415.757	2.91078	6.31304	2.85283	6.26928
DQ6	E3	0.546563	Pass	142.188	220.885	-42.6592	NA	577.769	415.162	3.0793	6.1275	2.94517	6.10067
DQ7	E7	0.551867	Pass	148.888	241.404	-34.6444	NA	618.808	430.113	3.27011	5.99994	3.1427	5.94223
All Signals		0.548487	Pass	NA	NA	NA	**13.1884**	NA	NA	NA	NA	NA	NA
DQS_t-DQS_c	C3, B3	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

Note: NA = Not Applicable; NMP = No Measurement Possible.



10.9.5.7 Delay Report

The Delay report includes Rx Signal and Tx Buffer Output information, MeasDealy, BufferDelay, InterconnectDelay, InterconnectSkew, StrobeInterconnectSkew, FirstSwitch, and FinalSettle. The worst-case values are highlighted in Bold, and a worst case summary sheet is listed in the following figure.

The Stimulus offset column is populated for designs that have the WLO/ClkMeasDelay option is selected in the Analysis options dialog box.

4.8 Delay Report

4.8.1 Data Bus Report

4.8.1.1 C:\Cadence\SPB_16.6\ASI\Base\SpeedXP\Samples\SystemSI\Parallel Bus Analysis\Tutorial\result\2\Data_Write_Typ_Typ\DiePad

4.8.1.1.1 Memory

Click on a signal name to see its waveform displayed.
Click on a criteria in a column header to see all its plots.

Bus Type: Data, Edge Type: BothEdges, Bus Group: Data[..., Timing Ref: LDQS-LDQS#, Stimulus Offset: Unknown, Measurement Range: [3937.88ps, end].

Rx			Tx										
Waveform	Pin	IO Model	Signal	Pin	IO Model	Stimulus Offset (ps)	Buffer Delay (ps)	MeasDelay (ps)	Max InterconnectDelay (ps)	Max| InterconnectSkew (ps)	Max| StrobeInterconnectSkew (ps)	Min FirstSwitch (ps)	Max FinalSettle (ps)
DQ0	E3	DDR3_DQ34_NO_ODT						3424.59	3424.59	-513.288	NA	**3392.58**	3523.71
DQ1	F7	DDR3_DQ34_NO_ODT						3423.8	3423.8	-514.073	NA	3393.23	3534.31
DQ2	F2	DDR3_DQ34_NO_ODT						3429.57	3429.57	-508.307	NA	3399.16	3541.27
DQ3	F8	DDR3_DQ34_NO_ODT						3429.58	**3429.58**	-508.296	NA	3399.03	**3541.97**
DQ4	H3	DDR3_DQ34_NO_ODT						3423.73	3423.73	**-514.144**	NA	3393.1	3535.41
DQ5	H8	DDR3_DQ34_NO_ODT						3428.5	3428.5	-509.375	NA	3397.89	3540.8
DQ6	G2	DDR3_DQ34_NO_ODT						3428.97	3428.97	-508.908	NA	3398.46	3539.88
DQ7	H7	DDR3_DQ34_NO_ODT						3425.01	3425.01	-512.868	NA	3392.69	3523.28
LDQS, LDQS#	F3, G3	DDR3_DQS34_NO_ODT						3937.88	3937.88	NA	**0**	NA	NA

Note: NA = Not Applicable; NMP = No Measurement Possible.

If the WLO/ClKDelay option is selected in the Analysis Options dialog box, the Stimulus Offset column includes the WLO values for controller block and and CLKDelay values for memory block.

4.8.1.2 C:\Cadence\SPB_16.6\ASI\Base\SpeedXP\Samples\SystemSI\Parallel Bus Analysis\Tutorial\result\2\Data_Read_Typ_Typ\DiePad

4.8.1.2.1 Controller

Click on a signal name to see its waveform displayed.
Click on a criteria in a column header to see all its plots.

Bus Type: Data, Edge Type: BothEdges, Bus Group: Data0, Timing Ref: MEMC_MDQS0

Rx Signal			Tx Buffer Output			BufferDelay (ps)	MeasDelay (ps)	Max InterconnectDelay (ps)	Max| InterconnectSkew (ps)	Max StrobeInterconnectSkew (ps)	Min FirstSwitch (ps)	Max FinalSettle (ps)
Waveform	Pin	IO Model	Signal	Pin	IO Model							
MEMC_MDQ0	AF16	ddr2_maxdrv_io	DQ0	H1	DQ_FULL_ODT_OFF	701.777	1237.61	535.831	2.68804	NA	453.673	553.393
MEMC_MDQ1	AE17	ddr2_maxdrv_io	DQ1	H3	DQ_FULL_ODT_OFF	684.69	1210.66	525.97	-7.17245	NA	428.509	544.128
MEMC_MDQ2	AH17	ddr2_maxdrv_io	DQ2	G2	DQ_FULL_ODT_OFF	692.819	1232.99	540.168	7.02463	NA	463.335	558.13
MEMC_MDQ3	AG17	ddr2_maxdrv_io	DQ3	F1	DQ_FULL_ODT_OFF	697.512	1237	539.484	6.34067	NA	460.835	557.031
MEMC_MDQ4	AG18	ddr2_maxdrv_io	DQ4	H7	DQ_FULL_ODT_OFF	681.429	1205.85	524.422	-8.72068	NA	**422.813**	544.833
MEMC_MDQ5	AH18	ddr2_maxdrv_io	DQ5	H9	DQ_FULL_ODT_OFF	693.738	1227.51	533.774	0.631409	NA	438.37	552.824
MEMC_MDQ6	AD18	ddr2_maxdrv_io	DQ6	G8	DQ_FULL_ODT_OFF	686.887	1240.89	**553.998**	**20.8553**	NA	442.802	**571.261**
MEMC_MDQ7	AF19	ddr2_maxdrv_io	DQ7	F9	DQ_FULL_ODT_OFF	698.735	1229.67	530.937	-2.20625	NA	439.515	548.121
MEMC_MDQS0	AF17	ddr2_maxdrv_io	LDQS	F7	DQ_FULL_ODT_OFF	681.787	1214.93	533.143	NA	**0**	NA	NA

Note: NA = Not Applicable; NMP = No Measurement Possible.

Bus Type: Data, Edge Type: BothEdges, Bus Group: Data1, Timing Ref: MEMC_MDQS1

Rx Signal			Tx Buffer Output			BufferDelay (ps)	MeasDelay (ps)	Max InterconnectDelay (ps)	|Max| InterconnectSkew (ps)	Max StrobeInterconnectSkew (ps)	Min FirstSwitch (ps)	Max FinalSettle (ps)
Waveform	Pin	IO Model	Signal	Pin	IO Model							
MEMC_MDQ8	AH19	ddr2_maxdrv_io	DQ8	B1	DQ_FULL_ODT_OFF	701.701	1226.84	525.143	-1.27371	NA	446.184	544.815
MEMC_MDQ9	AD19	ddr2_maxdrv_io	DQ9	D3	DQ_FULL_ODT_OFF	686.113	1207.87	521.76	-4.65747	NA	**427.664**	541.102
MEMC_MDQ10	AG20	ddr2_maxdrv_io	DQ10	D1	DQ_FULL_ODT_OFF	704.399	1241.69	537.294	10.8769	NA	465.33	556.493
MEMC_MDQ11	AH20	ddr2_maxdrv_io	DQ11	C2	DQ_FULL_ODT_OFF	693.815	1231.57	537.754	11.3369	NA	464.623	558.289
MEMC_MDQ12	AH21	ddr2_maxdrv_io	DQ12	D7	DQ_FULL_ODT_OFF	683.257	1227.47	**544.217**	**17.8001**	NA	465.262	**564.23**
MEMC_MDQ13	AE21	ddr2_maxdrv_io	DQ13	D9	DQ_FULL_ODT_OFF	701.105	1230.07	528.961	2.54342	NA	451.395	548.53
MEMC_MDQ14	AH22	ddr2_maxdrv_io	DQ14	B9	DQ_FULL_ODT_OFF	699.175	1242.28	543.104	16.687	NA	473.809	562.533
MEMC_MDQ15	AD21	ddr2_maxdrv_io	DQ15	C8	DQ_FULL_ODT_OFF	692.211	1217.22	525.005	-1.41248	NA	437.715	545.687
MEMC_MDQS1	AG21	ddr2_maxdrv_io	UDQS	B7	DQ_FULL_ODT_OFF	684.293	1210.71	526.417	NA	**6.7258**	NA	NA

Note: NA = Not Applicable; NMP = No Measurement Possible.

4.8.2 Worst Case Summary

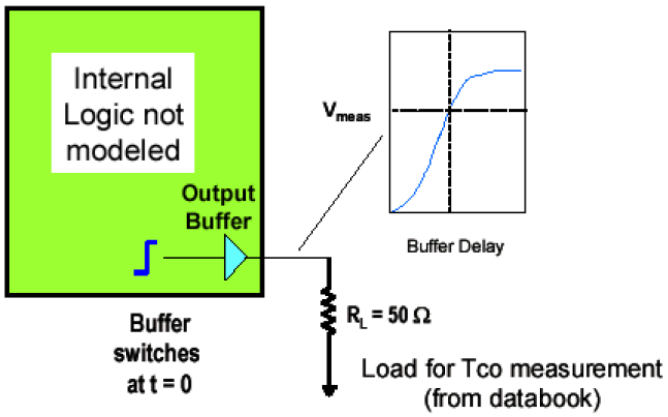
Simulation Results:
Case 1: C:\Cadence\SPB_16.6\ASI\Base\SpeedXP\Samples\SystemSI\Parallel Bus Analysis\Tutorial\result\2\Data_Write_Typ_Typ\DiePad
Case 2: C:\Cadence\SPB_16.6\ASI\Base\SpeedXP\Samples\SystemSI\Parallel Bus Analysis\Tutorial\result\2\Data_Read_Typ_Typ\DiePad

Measurement	BufferDelay (ps)	MeasDelay (ps)	Max InterconnectDelay (ps)	Max| InterconnectSkew (ps)	Max StrobeInterconnectSkew (ps)	Min FirstSwitch (ps)	Max FinalSettle (ps)
Worst Value			553.998	84.6564	19.483	134.276	571.261
Simulation Result			Case 2	Case 1	Case 1	Case 1	Case 2
Receiver			Controller	Memory	Memory	Memory	Controller
Bus Group			Data0	Data1	Data1	Data1	Data0
Rx Signal (Waveform)			MEMC_MDQ6	DQ6	LDQS	DQ4	MEMC_MDQ6
Cycle							0.5

The delay measurements are defined as below:

- **BufferDelay**

  The output buffer model used for signal integrity analysis is connected to the Tco test load and simulated. The delay is measured at the point where the output pin crosses Vmeas. The corresponding delay is then saved and used in flight time computations.



- **MeasDelay**

  Measure delay(MeasDelay) is measured from time 0 in the simulation to when the receiver waveform crosses the Vmeas threshold.

- **InterconnectDelay**

  InterconnectDelay is measured on each cycle, as:

  MeasDelay - BufferDelay

  This produces a min and max value.

- **InterconnectSkew**

  InterconnectSkew is measured on each cycle, as:

  InterconnectDelay_Signal - InterconnectDelay_TimingRef

  The max one of these gets reported in the Delay Report as **Max InterconnectSkew**.

- **StrobeInterconnectSkew**

  StrobeInterconnectSkew is measured on each cycle, as:

InterconnectDelay_Strobe - InterconnectDelay_Clock

The max one of these gets reported in the Delay Report as **Max StrobeInterconnectSkew**.

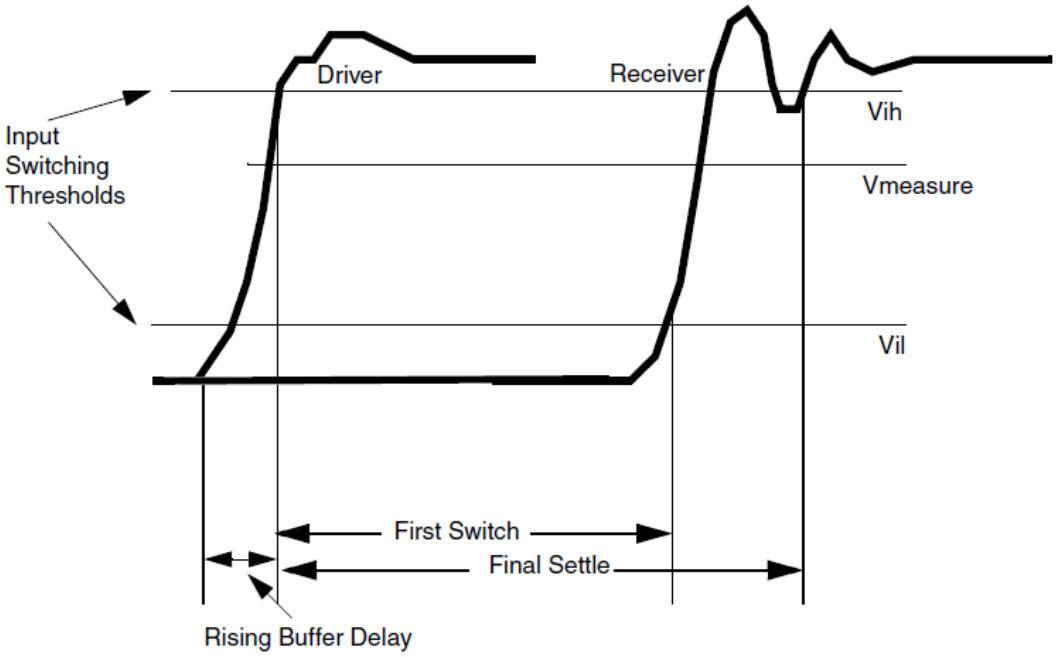
- **FirstSwitch**

  This is determined by subtracting the associated buffer delay from a simulation measurement. For example, on a rising (falling) edge, the simulation measurement is from time zero to when the receiver first crosses its Vil (Vih) DC logic threshold. The associated rising (falling) buffer delay of the driving IOCell is subtracted from this measurement value to produce the reported first switch delay.

- **FinalSettle**

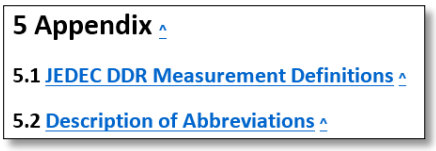
  This is determined by subtracting the associated buffer delay from a simulation measurement. For example, on a rising (falling) edge, the simulation measurement is from time zero to when the receiver crosses its Vih (Vil) AC logic threshold the final time and settles into the high (low) logic state. The associated rising (falling) buffer delay of the driving IOCell is subtracted from this measurement value to produce the reported final settle delay.

  The following diagram illustrates buffer delay, first switch delay, and final settle delay for a rising signal.

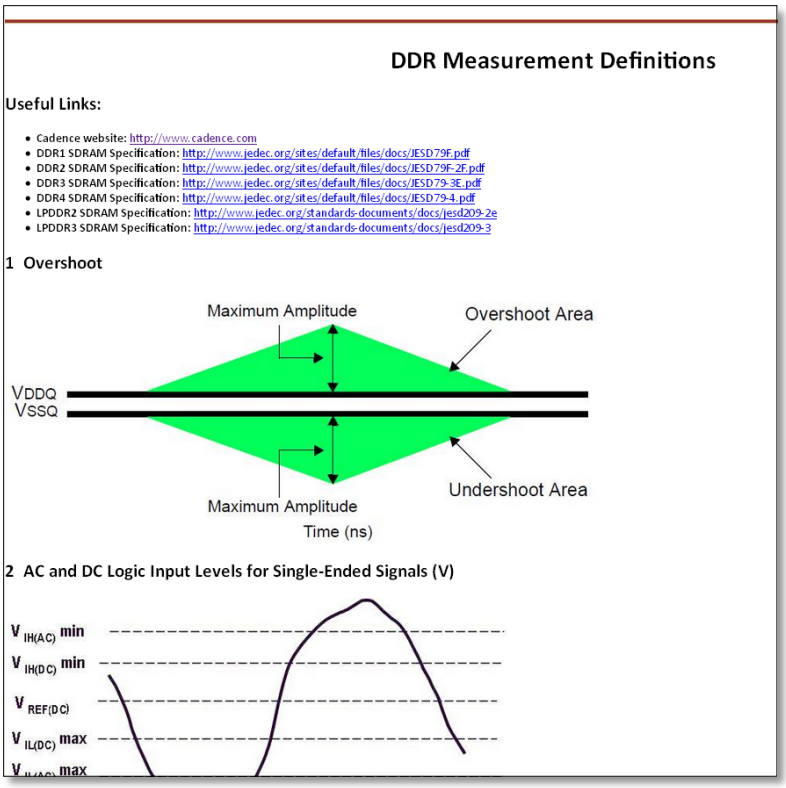


10.9.6 Appendix

This section provides links to the DDR Measurement Definitions and Abbreviations.



Selecting the JEDEC DDR Measurement Definitions link opens another HTML files that explains each of the measurements included in the generated report.



Select the Description of Abbreviation link, displays a table containing the Abbreviation and its meaning.

Description of Abbreviations

ApertureWidth	Aperture Width of Eye Diagram
BufferDelay	Buffer Delay of the Tx IO Model
ClockInterconnectSkew	Skew between Strobe and Clock
DDR	Double Data Rate
DDR SDRAM	Double Data Rate Synchronous Dynamic Random-Access Memory
DM	Write Data Mask Signal
DQ	Data Signal
DQS	Data Strobe Signal
FinalSettle	Final Settle Delay
FirstSwitch	First Switch Delay
IBIS	Input/Output Buffer Information Specification
InterconnectDelay	MeasDelay - BufferDelay
InterconnectSkew	Interconnect Skew between the Signal and TimingRef
JEDEC	Joint Electron Devices Engineering Council
MeasDelay	Measured Delay at Rx crossing of Vmeas (per IBIS)
NA	Not Applicable
NMP	No Measurement Possible
OOR	Out of Range
Overshoot	Max Excursion of Signal over Power Rail

11 SystemSI with Other Tools

This chapter covers following topics.

- SystemSI Integration with Allegro Signal Explorer (SigXplorer)
- SystemSI Integration with Timing Designer
- Auto Archive SSI Project

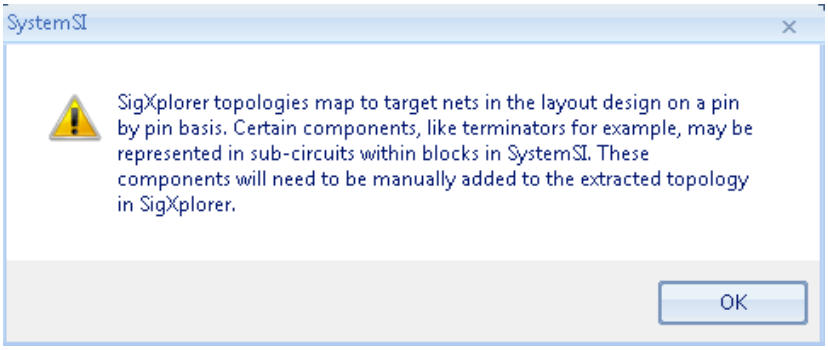
11.1 SystemSI Integration with Allegro Signal Explorer (SigXplorer)

SystemSI provides a way to push a topology for a particular signal into SigXplorer for constraint capture. SigXplorer topologies are then used to drive electrical constraints, including the routing schedule, into Allegro layout. The topology database becomes an Electrical Constraint Set (ECSet) within the Allegro database that can be applied to one or many nets.

To export the results of Parallel bus analysis as Electrical constraints, perform the following steps.

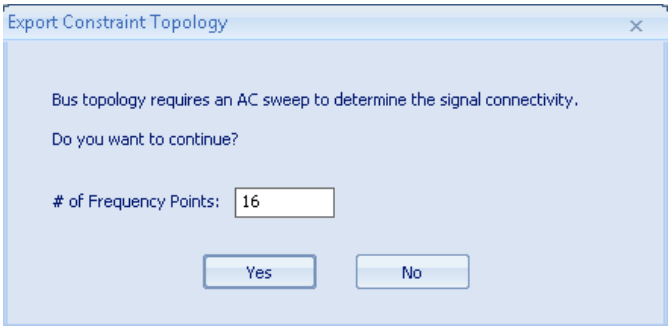
1. From the File menu, choose **Export Constraint Topology**.

   Following message box is displayed.



2. Select **OK**, to close the message box.

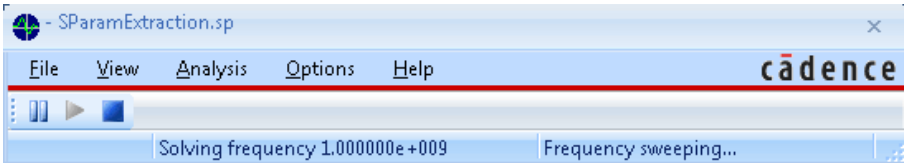
   The Export Constraint Topology dialog box appears.



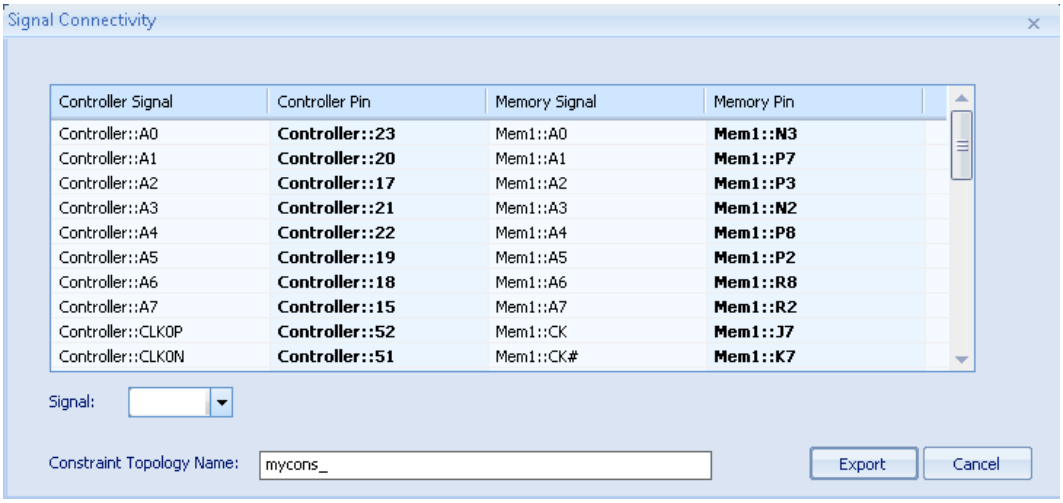
   Before extracting the topology information, System SI runs a connectivity check of the Controller – Memory topology. The connectivity is determined by running the AC Sweep analysis (involves S parameter extracting of the interconnects) and analyzing the strength of each connection. For a design or a topology, this is a one-time validation task. The analysis results are used for all subsequent exports.

3. To continue, click **Yes**.

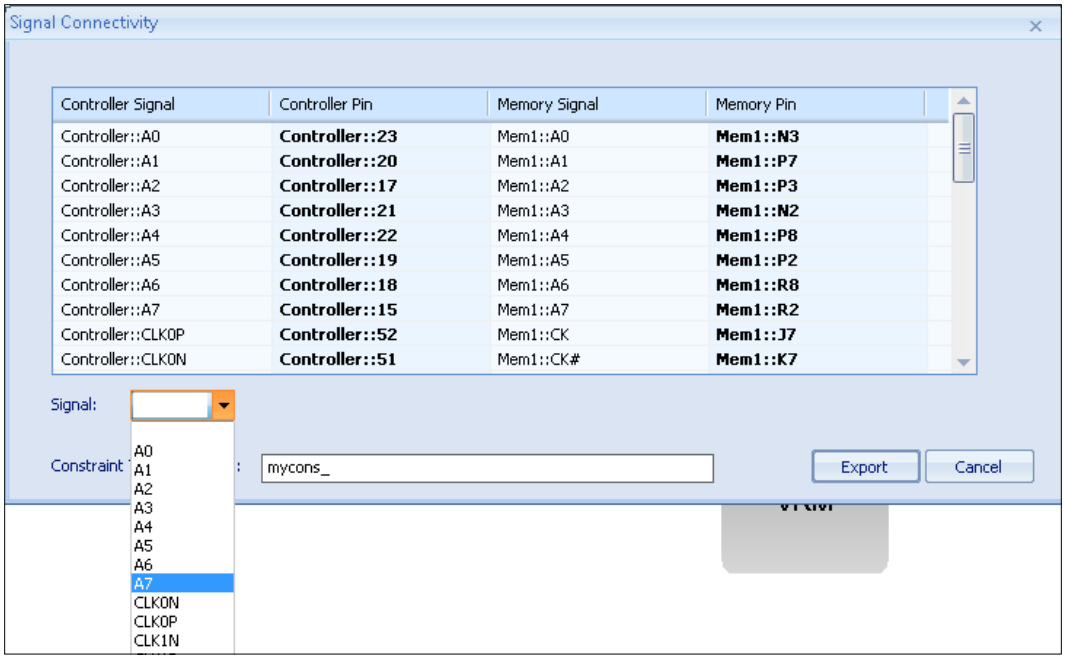
AC sweep analysis starts to determine the signal connectivity.



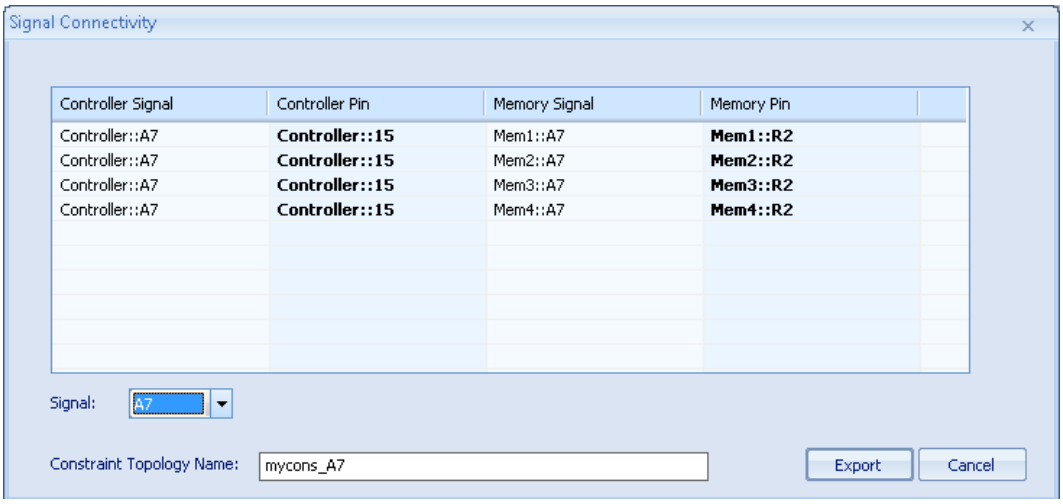
Once the AC Sweep analysis is completed, the connectivity data is displayed in the Signal Connectivity dialog box as shown in the following figure.



4. From the signal drop-down list, select the signal for which topology is to be exported as constraint.



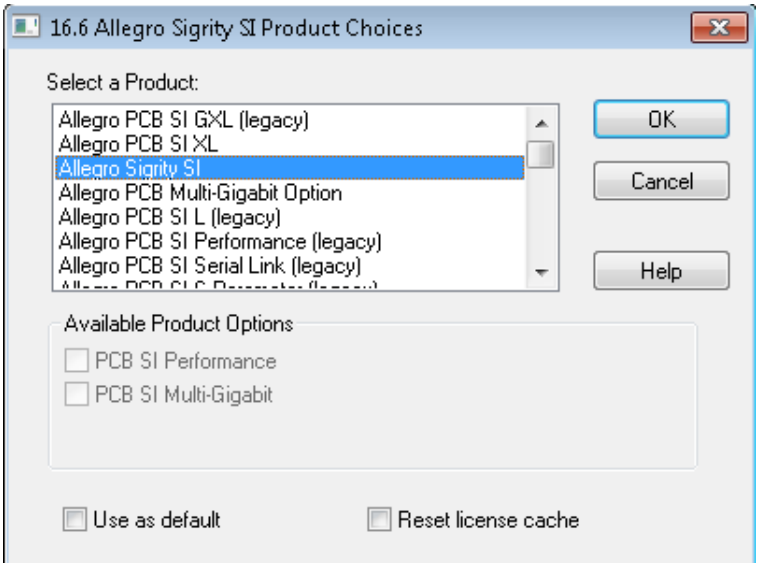
All blocks connected to that signal are displayed in the Signal Connectivity dialog box. The Constraint Topology Name field is also populated by default, *<design name_signalname>*. If required, you can specify a different name for the constraint topology.



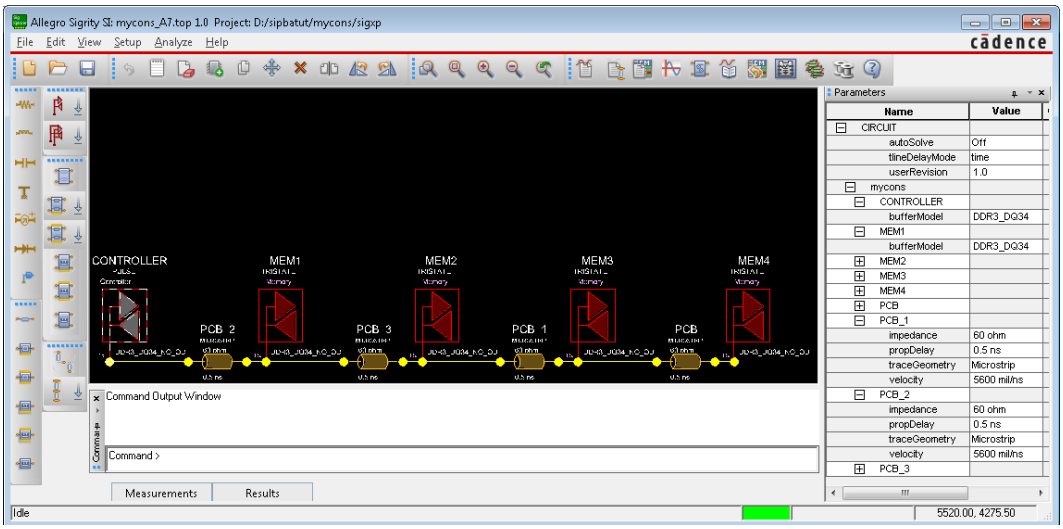
5. Click **Export**.

   Splash screen for Allegro SigXplorer appears, followed by the Product selection dialog box.

6. Select the appropriate license and click **OK**.



The exported topology is displayed in Allegro SigXplorer.



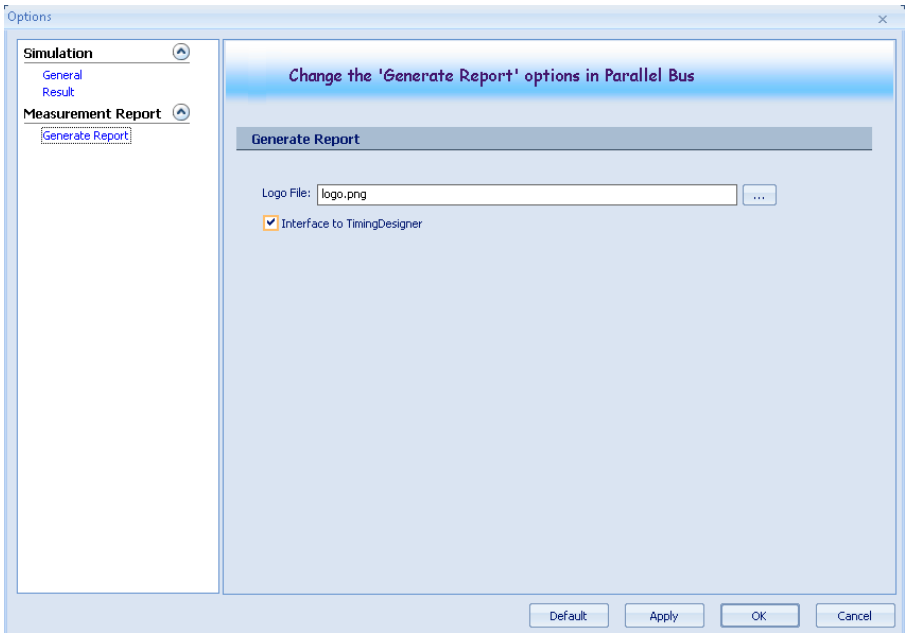
Similarly, you can export the topology information for other nets as well. On the same SystemSI design, when you repeat the steps listed above, you will observe that the **Export Constraint Topology** dialog box is not displayed. This is because the connectivity information generated in the first run is used to populate the Signal Connectivity dialog box.

Caution!	Do not use the constraint topologies in SigXplorer for simulations. This is because during the export process, various SystemSI blocks — such as S-Param, PCB, traces, package blocks, and so on —are translated to ideal T-Lines, so that they are understood by SigXplorer as a constraint topology.

11.2 SystemSI Integration with TimingDesigner

SystemSI provides support for exporting timing measurements to TimingDesigner, a timing analysis tool from EMA. To export the bus data from SystemSI to TimingDesigner, perform the following tasks.

- Enable Integration with TimingDesigner
  a) From the **Tools** menu, choose ***Option -- Edit Options***.
  b) From the left pane of the Options dialog box, select **Generate Report**.
  c) Select the **Interface to TimingDesigner** check box.



  d) To close the dialog box and save the changes, click **OK**.

- Launch Timing Designer

a) From the work flow, select **Generate Report**.

b) In the Timing Designer tab, specify the location of the executable.

c) Select Launch to invoke Timing Designer

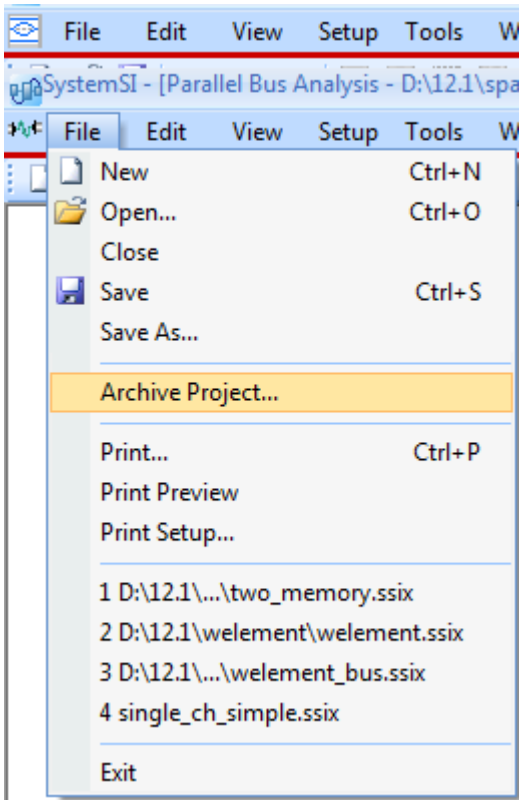
SystemSI generates the configuration file based on the timing measurements and launches the TimingDesigner to draw the timing diagram.

By default, Setup and Hold measurements are exported to the TimingDesigner. However, for Data Read results, you can specify whether the ReadSkew or Setup and Hold measurements are to be exported.

NOTE!	The SystemSI generated configuration files contain information about simulation setup, bus group definition, and timing measurement results. These are required by TimingDesigner to generate timing diagrams.

11.3 Auto Archive SSI Project

1. Choose File > Archive Project….



   The **Archive Project** window opens.

2. Specify **Project Location** and **Project Name** for the copied project.

   By default:

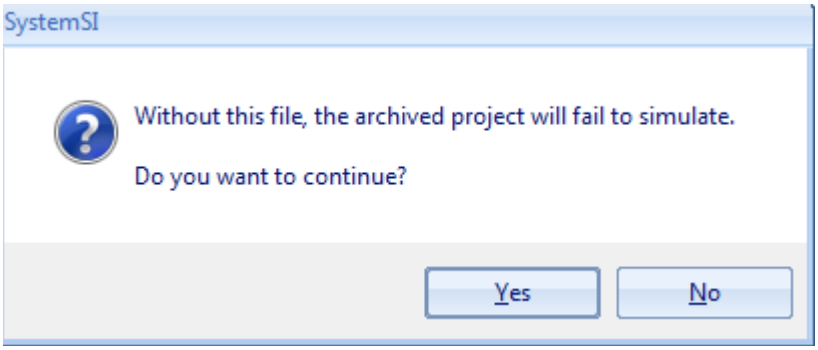
   - The location of the original project is selected for the copied project location
   - **<original_project_name>_copy** is set for the copied project name

3. Check the files that should be included in the copied project.

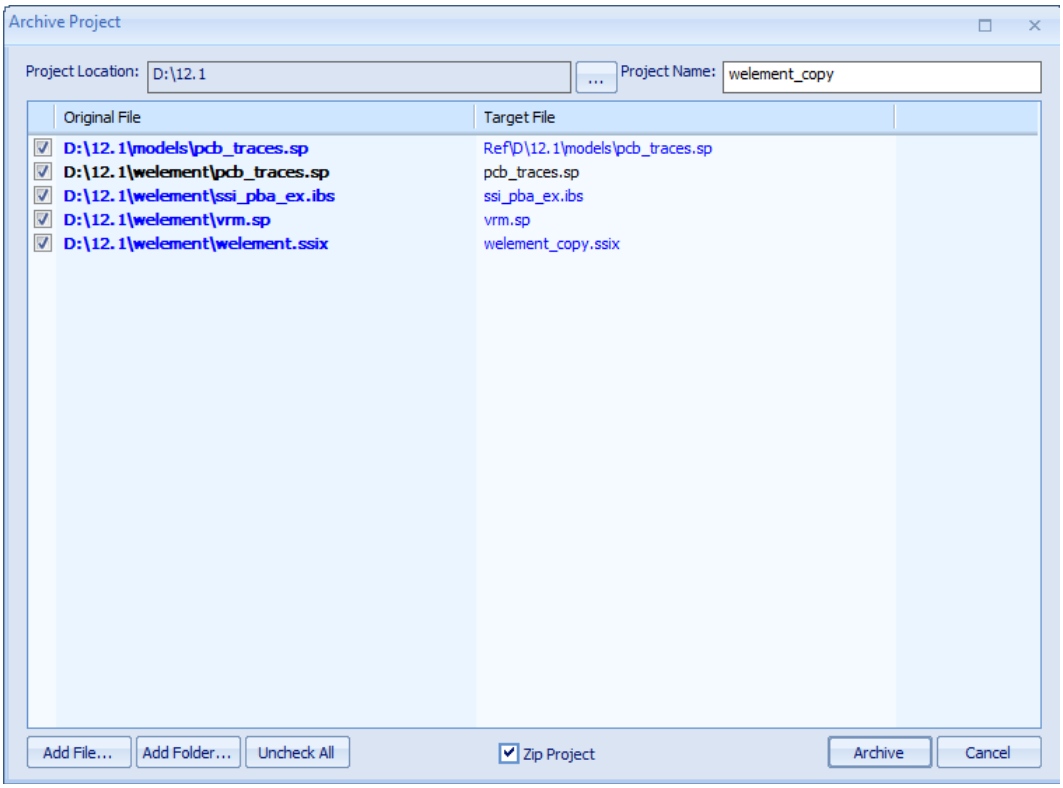
   By default,

   - The files under the original project folder and under the original sub-folders are listed and checked for archiving, and will be copied to the new project folder
   - The files used by the original project but located other than the original project folder are listed and checked, and they will be copied to a sub-folder named **Ref** under the new project folder
   - The files under the **history**, **result**, **result(bat)**, **bufferdelay** and **Report** sub-folders are not listed

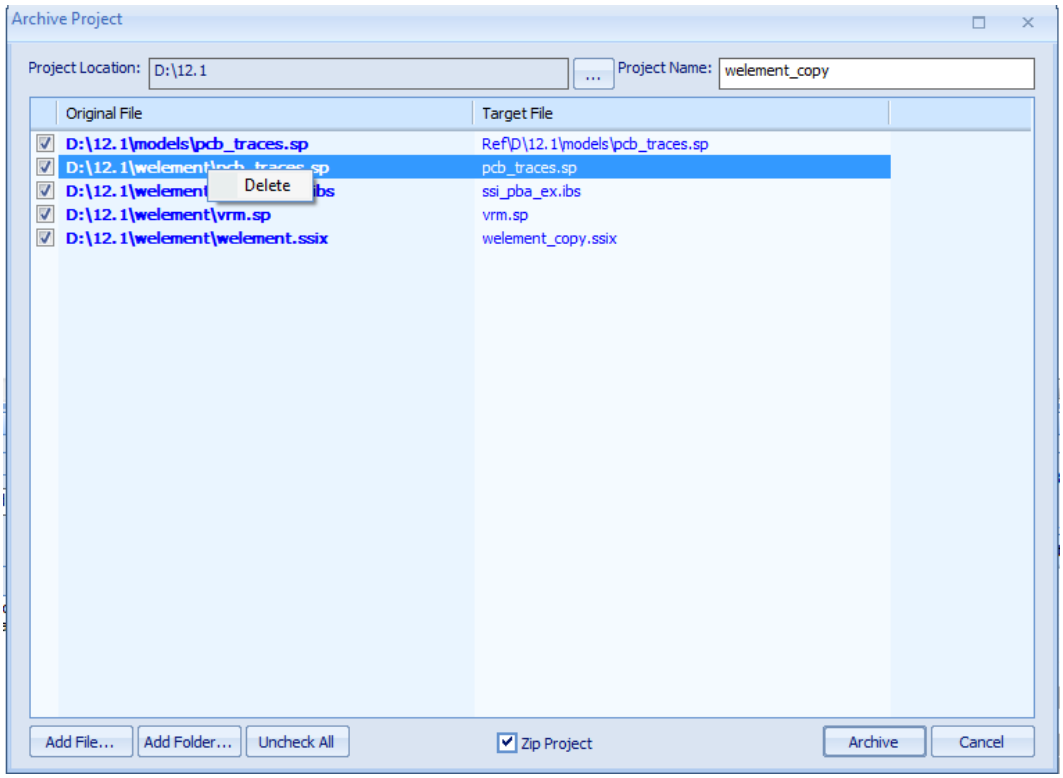
   If you want to uncheck a file used by the original project, the following message opens.



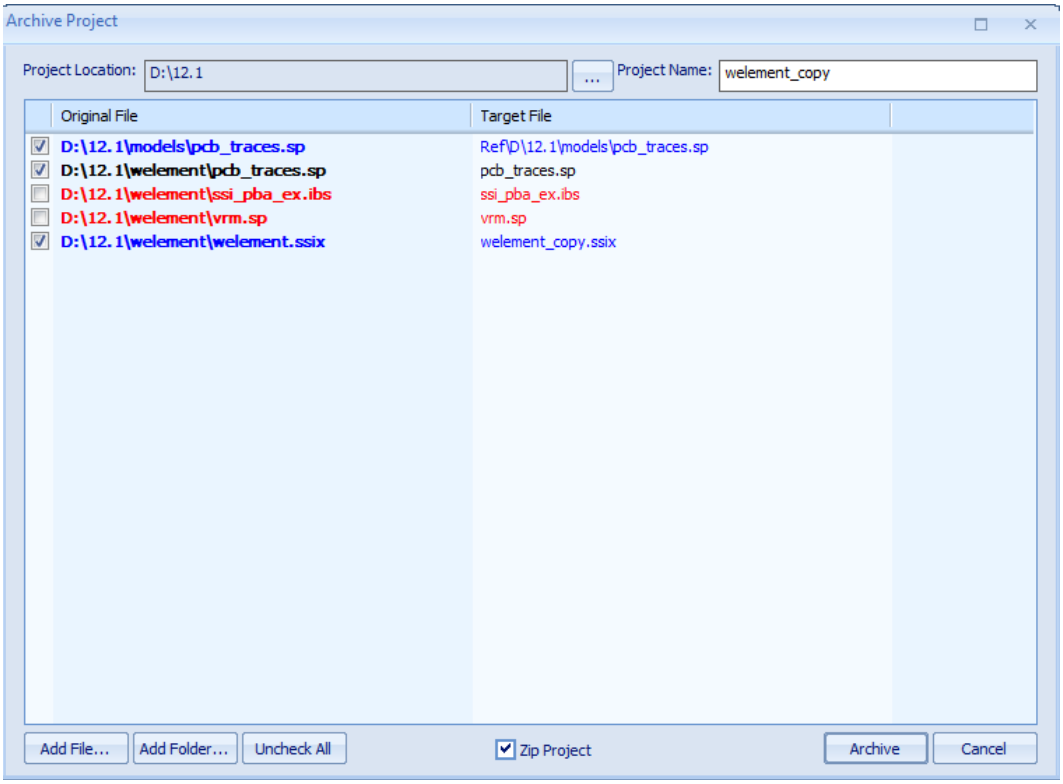
4. To add additional file, click the **Add File…** button.
5. To add additional folder, click the **Add Folder…** button.
    - The files checked and used by the original project are highlighted in blue



    - The files not used by the original project are not highlighted, and they can be deleted from the list



- The files are highlighted in red if they are used by the original project but do not exist or are un-checked

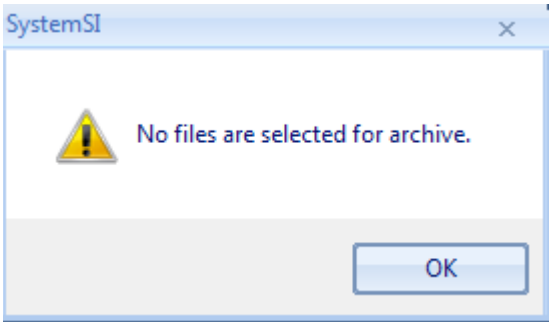


NOTE!	If there are any files highlighted in red, the copied project will fail to simulate.

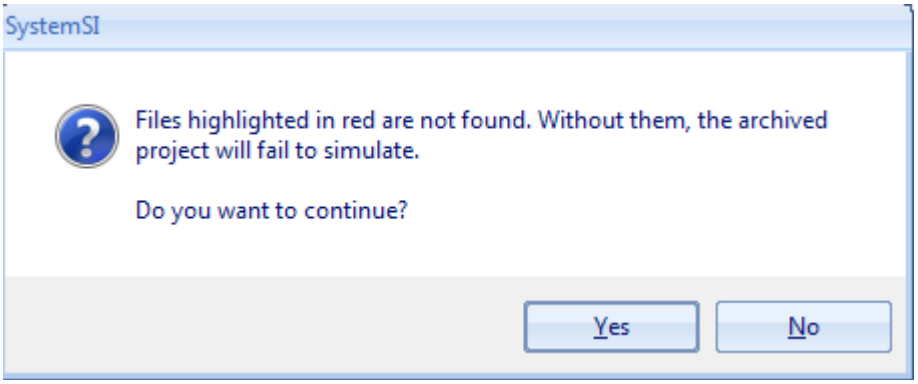
6. Click the **Archive** button to generate the copied project.
   - All the checked files will be copied to the copied project
   - If the **Zip Project** option is checked, a zipped project file will be created. Otherwise, a new project folder will be created for the copied project

   The following messages may appear while archiving:

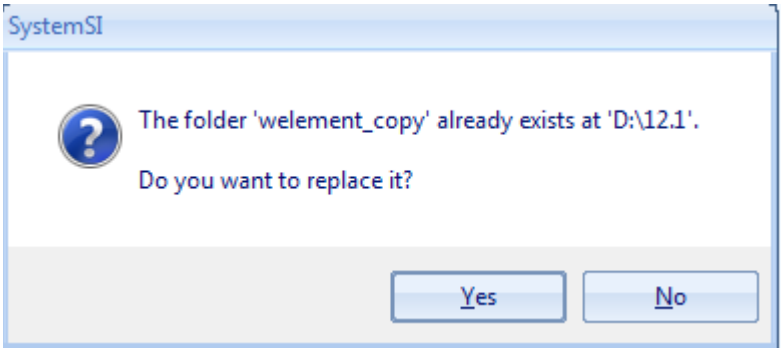
   - If no files are checked, the following message displays.

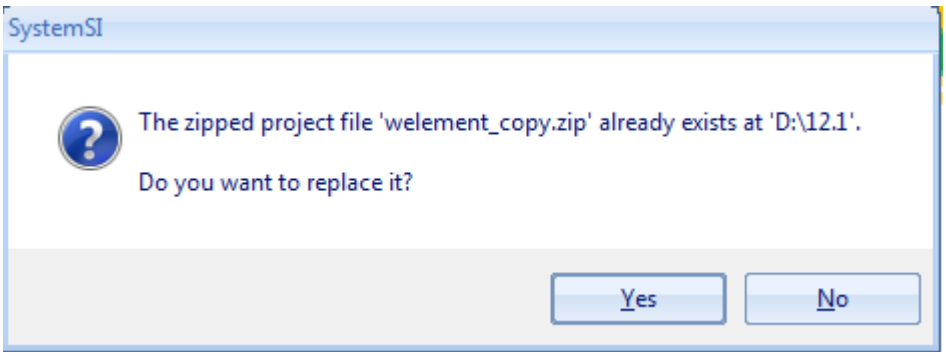


   - If the file used by the original project does not exist, the following message opens

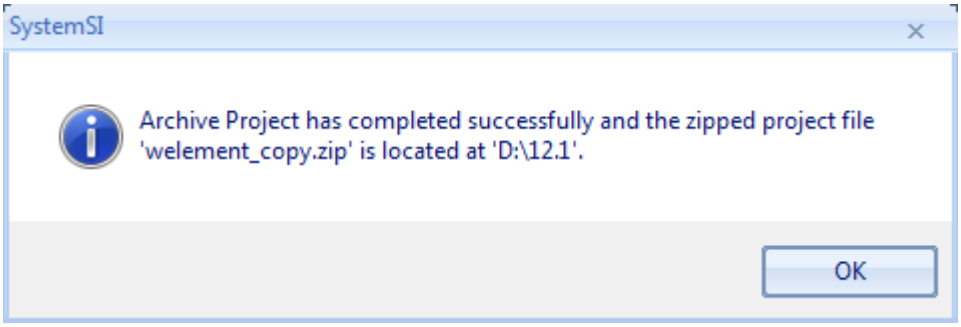


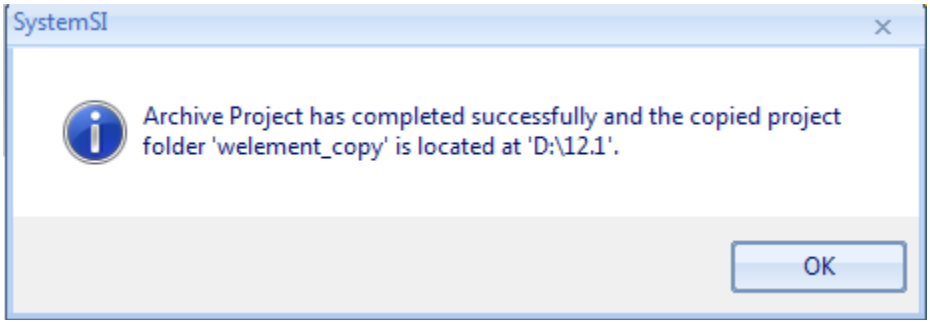
   - If the designated project folder already has the archived project folder or zipped file, the following messages open





- If the archive project process completes successfully, the following messages open





A Appendix: Batch Mode Support

A.1 Batch Mode Command

List of the batch mode commands for SystemSI – Parallel Bus Analysis:

```
<ExeFileFullPath\SystemSI.exe> -b -sim <workspace file>
```

\\ for the default SSI simulation. For a PBA workspace, it will generate the DDR report and export the html file to the default folder

```
<ExeFileFullPath\SystemSI.exe> -b -sim:sweep <workspace file>
```

\\ for the sweep. For a PBA workspace, it will generate the DDR report and export the html file to the default folder

```
<ExeFileFullPath\SystemSI.exe> -b -sim:freq <workspace
file>
```

\\ for the frequency response

```
<ExeFileFullPath\SystemSI.exe> -b -sim:sparam <workspace
file>
```

\\ for the S-parameter extraction

Example for Windows OS:

```
Path\SystemSI.exe -b –sim
“C:\Working\SystemSI\pbs\pbs_1\pbs_1.ssix”
Path\SystemSI.exe -b -sim:sweep
“C:\Working\SystemSI\pbs\pbs_1\pbs_1.ssix”
Path\SystemSI.exe -b -sim:freq
“C:\Working\SystemSI\pbs\pbs_1\pbs_1.ssix”
Path\SystemSI.exe -b -sim:sparam
“C:\Working\SystemSI\pbs\pbs_1\pbs_1.ssix”
```

A.2 Run a .bat file

1. Put down the batch mode commands into a `.bat` file.

   Example for Windows OS:

   (pbs_1.ssix: a PBA workspace)

   ```
   %systemsi% -b –sim “C:\Working\SystemSI\pbs\pbs_1\pbs_1.ssix”
   %systemsi% -b -sim:sweep
   “C:\Working\SystemSI\pbs\pbs_1\pbs_1.ssix”
   %systemsi% -b -sim:freq
   “C:\Working\SystemSI\pbs\pbs_1\pbs_1.ssix”
   ```

```
%systemsi% -b -sim:sparam
“C:\Working\SystemSI\pbs\pbs_1\pbs_1.ssix”
```

To run the tests simultaneously, make the following modification in the tested `.bat` file:

```
set systemsi=start “the full path of the systemsi.exe”
```

NOTE!	Replace the red font part with the full path of SystemSI.exe file.

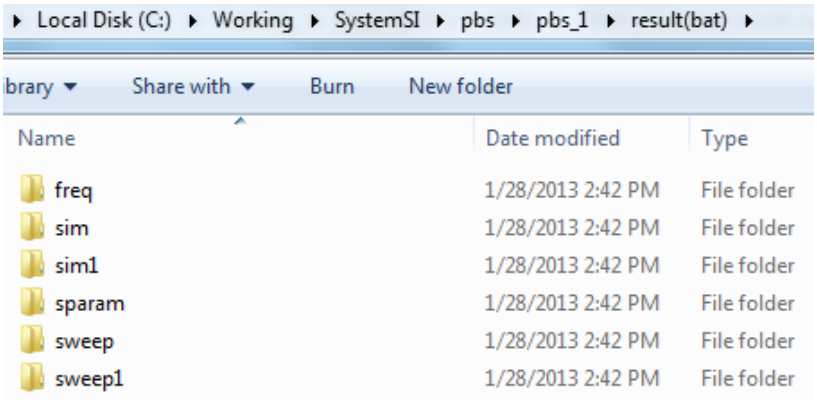
2. Double-click the `.bat` file to run it.

A.3 Result Folder for Batch Mode

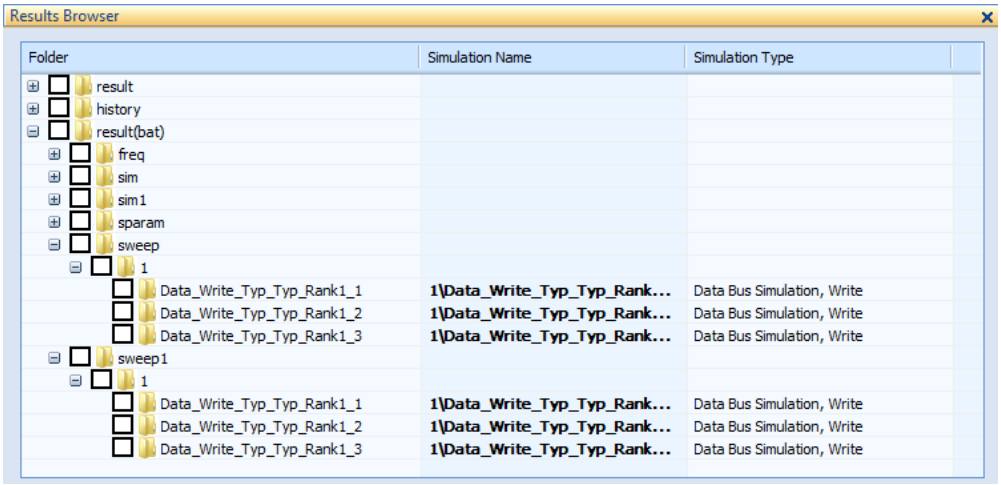
A new sub-folder **result(bat)** will be added for all batch mode results. Each batch mode simulation has its own sub-folder under the **result(bat)** folder.

For a PBA workspace, each result is located in its responding sub-folder.

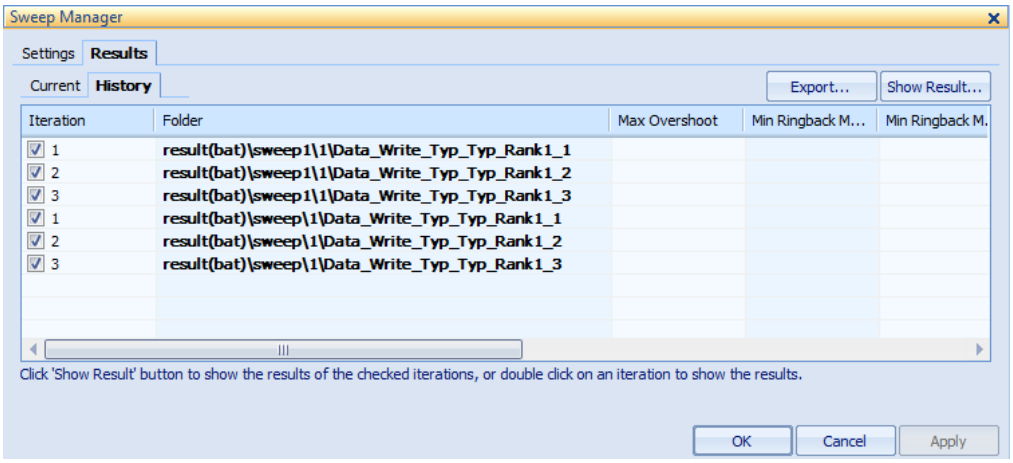
Example



The **result(bat)** folder is available from **Results Browser**.



In **Sweep Manager**, all the sweep results including those in the **result(bat)** folder are listed under the **Results** > **History** tab.



A.4 batch_mode.log File

If a simulation does not run or fails for any reasons, please check the batch_mode.log file located under the **result(bat)** folder.

Error messages are added to batch_mode.log for the following failures:

- The sweep parameter is not defined for the sweep analysis
- The ports are not defined for the S-parameter extraction