

UltraScale & UltraScale+ MPSoC DDR Controller Settings and IBIS Simulation

This page provides detailed information regarding DDR configuration and setting up IBIS simulations for the UltraScale and UltraScale+ family of FPGAs and MPSoCs.

Table of Contents

Introduction

Xilinx has determined through extensive simulation and characterization, the FPGA and DRAM configuration settings including Drive Strength, ODT, and Vref. These values are used when the IP is generated. This information is captured in the sections below. The Custom IBIS models capture this information and can be used for board-level simulation.

This guide provides the following details and guidance

- Supported PL and PS DRAMs
- ODT and V_{REF} configuration settings for each supported DRAM interface
- IBIS Models for use in simulations
- Available resources for HyperLynx and ADS simulation tools
- HyperLynx DDRx tips

FPGA and MPSoC Supported DRAM

The UltraScale and UltraScale+ families of FPGAs and MPSoCs support several different DRAM technologies and configurations. This chapter provides an overview of the supported DRAMs and configurations.

Supported DRAM technologies

- FPGA, Programmable Logic (PL), DRAM IP
 - DDR4
 - DDR3
- MPSoC, Processing System (PS), Configured with Processor Configuration Wizard (PCW)
 - DDR4
 - LPDDR4
 - DDR3
 - LPDDR3
- MPSoC devices support both PL and PS DRAM interfaces

Supported DRAM configurations

- Table 1 outlines the supported PL DRAM configurations, PL DRAM performance, and configurations found in DS923, Maximum Physical Interface (PHY) Rate for Memory Interfaces Table.
- Table 2 outlines the supported PS DRAM performance and configurations, PS DRAM configurations found in DS925, Configuration and Security Unit Performance Table.

Table 1: Supported PL DRAM Configurations

Configuration	DDR3/3L	DDR4
Component, 1 rank	x4, x8, x16	x4, x8, x16
Component, 2 rank	x4, x8, x16	x4, x8, x16
1 slot, 1 rank	RDIMM, UDIMM, SODIMM	RDIMM, UDIMM, SODIMM, LRDIMM
1 slot, 2 rank	RDIMM, UDIMM, SODIMM	RDIMM, UDIMM, SODIMM, LRDIMM
1 slot, 4 rank	RDIMM	LRDIMM
2 slot, 1 rank	RDIMM, UDIMM, SODIMM	RDIMM, UDIMM, SODIMM, LRDIMM
2 slot, 2 rank	RDIMM, UDIMM, SODIMM	RDIMM, UDIMM, SODIMM, LRDIMM
2 slot, 4 rank		LRDIMM (UltraScale+ Only)
Configuration	LPDDR3	
1 rank, Component DRAM	x16, x32	

Table 2: Supported PS DRAM Configurations

Configuration	DDR3/3L	DDR4
Component, 1 rank	x8, x16	x8, x16
Component, 2 rank	x8, x16	x8, x16
1 slot, 1 rank	RDIMM, UDIMM, SODIMM	RDIMM, UDIMM, SODIMM
1 slot, 2 rank	RDIMM, UDIMM, SODIMM	RDIMM, UDIMM, SODIMM
1 slot, 4 rank	RDIMM, UDIMM, SODIMM	RDIMM, UDIMM, SODIMM
2 slot, 1 rank	RDIMM, UDIMM, SODIMM	RDIMM, UDIMM, SODIMM
2 slot, 2 rank	RDIMM, UDIMM, SODIMM	RDIMM, UDIMM, SODIMM
2 slot, 4 rank	RDIMM, UDIMM, SODIMM	RDIMM, UDIMM, SODIMM
Configuration	LPDDR3	LPDDR4
Component, 1 rank	x32 or x64, ECC Option	x16 or x32, ECC Option
Component, 2 rank	x32 or x64, ECC Option	x16 or x32, ECC Option

PL DRAM IP Drive Strength, ODT, and V_{REF} Configuration

The PL DRAM IP has been characterized and tested to identify the optimal drive strength, ODT, and V_{REF} settings. This chapter provides the values that will always be used for the PL DRAM IP UltraScale and UltraScale+ DDR3 and DDR4 DRAM interfaces. The Memory Controller supports the following calibration routines. For more details on these routines, please see [PG150](#).

- Write Leveling
 - Write DQS to DQ Deskew
- Read Leveling
 - Per-Bit Deskew

- Read DQS Centering
- Write Latency Calibration
 - Write DQS to CK alignment

UltraScale PL DDR4

- Table 3 provides PL DDR4 FPGA drive strength and ODT configurations
 - FPGA Slew Rate is always FAST
- Table 4 provides PL DDR4 DRAM drive strength and ODT configurations
- Table 5 provides PL DDR4 V_{REF} configurations

Table 3: PL DDR4 FPGA drive strength & ODT configurations

UltraScale PL DDR4	FPGA Driver Strength, Ohm	FPGA ODT, Ohm
Component, 1 or 2 rank	40	40
1 slot, 1 rank	40	40
2 slot, 1 rank	40	40
1 slot, 2 rank	40	40
2 slot, 2 rank	40	60
1 slot, 4 rank	40	40

Table 4: PL DDR4 DRAM drive strength and ODT configurations

UltraScale PL DDR4	DRAM Strength, Ohm	R _{TT} (nom), Ohm	R _{TT} (park), Ohm
Component, 1 or 2 rank	34	40	N/A
1 slot, 1 rank	34	40	N/A
2 slot, 1 rank	34	60	40
1 slot, 2 rank	34	120	60
2 slot, 2 rank	34	240	60
1 slot, 4 rank	34	40	60

Table 5: PL DDR4 V_{REF} configurations

UltraScale PL DDR4 (Vcc = 1.2V)	WRITE V _{REF} , V	READ V _{REF} , V	WRITE V _{REF} , %	READ V _{REF} , %
Component, 1 or 2 rank	0.88	0.88	73%	73%
1 slot, 1 rank	0.93	0.93	78%	78%
2 slot, 1 rank	0.97	1.01	81%	84%
1 slot, 2 rank	0.93	0.97	78%	81%
2 slot, 2 rank	1.00	0.99	83%	83%

1 slot, 4 rank	0.93	0.97	78%	81%
----------------	------	------	-----	-----

UltraScale PL DDR3

- Table 6 provides PL DDR3 FPGA drive strength & ODT configuration
 - FPGA Slew Rate is always FAST
- Table 7 provides PL DDR3 DRAM drive strength and ODT configuration

Table 6: PL DDR3 FPGA drive strength & ODT configuration

UltraScale PL DDR3	FPGA Driver Strength, Ohm	FPGA ODT, Ohm
Component, 1 or 2 rank	40	40
1 slot, 1 rank	40	40
2 slot, 1 rank	40	40
1 slot, 2 rank	40	40
2 slot, 2 rank	40	60
1 slot, 4 rank	40	40

Table 7: PL DDR3 DRAM drive strength and ODT configuration

UltraScale PL DDR3	DRAM Strength, Ohm	R _{TT(nom)} , Ohm	R _{TT(wr)} , Ohm
Component, 1 or 2 rank	40	40	Disabled
1 slot, 1 rank	40	40	Disabled
2 slot, 1 rank	40	40	60
1 slot, 2 rank	40	120	60
2 slot, 2 rank	40	60	120
1 slot, 4 rank	40	120	60

UltraScale+ PL DDR4

- Table 8 provides PL DDR4 FPGA drive strength & ODT configurations
 - FPGA Slew Rate is always FAST
- Table 9 provides PL DDR4 DRAM drive strength and ODT configurations
- Table 10 provides PL DDR4 V_{REF} configurations

Table 8: PL DDR4 FPGA drive strength & ODT configurations

UltraScale+ PL DDR4	FPGA Driver Strength, Ohm	FPGA ODT, Ohm
Component, 1 or 2 rank	40	40

1 slot, 1 rank	40	60	
2 slot, 1 rank	40	60	
1 slot, 2 rank	40	60	
2 slot, 2 rank	40	60	
1 slot, 4 rank	40	60	
2 slot, 4 rank	40	60	

Table 9: PL DDR4 DRAM drive strength and ODT configurations

UltraScale+ PL DDR4	DRAM Strength, Ohm	R _{TT} (nom), Ohm	R _{TT} (park), Ohm
Component, 1 or 2 rank	34	40	N/A
1 slot, 1 rank	34	40	N/A
2 slot, 1 rank	34	60	40
1 slot, 2 rank	34	120	60
2 slot, 2 rank	34	240	60
1 slot, 4 rank	34	40	60
2 slot, 4 rank	34	40	60

Table 10: PL DDR4 V_{REF} configurations

UltraScale+ PL DDR4 (Vcc = 1.2V)	WRITE V _{REF} , V	READ V _{REF} , V	WRITE V _{REF} , %	READ V _{REF} , %
Component, 1 or 2 rank	0.88	0.88	73%	74%
1 slot, 1 rank	0.93	0.87	78%	73%
2 slot, 1 rank	0.97	0.98	81%	82%
1 slot, 2 rank	0.93	0.92	78%	77%
2 slot, 2 rank	1.00	0.99	83%	83%
1 slot, 4 rank	0.93	0.89	78%	74%
2 slot, 4 rank	0.93	0.95	78%	79%

UltraScale+ PL DDR3

- Table 11 provides PL DDR3 FPGA drive strength & ODT configurations
 - FPGA Slew Rate is always FAST

- Table 12 provides PL DDR3 DRAM drive strength and ODT configurations

Table 11: PL DDR3 FPGA drive strength & ODT configurations

UltraScale+ PL DDR3	FPGA Driver Strength, Ohm	FPGA ODT, Ohm
Component, 1 or 2 rank	40	40
1 slot, 1 rank	40	40
2 slot, 1 rank	40	40
1 slot, 2 rank	40	40
2 slot, 2 rank	40	60
1 slot, 4 rank	40	40

Table 12: PL DDR3 DRAM drive strength and ODT configurations

UltraScale+ PL DDR3	DRAM Strength, Ohm	R _{TT} (nom), Ohm	R _{TT} (wr), Ohm
Component, 1 or 2 rank	40	40	Disabled
1 slot, 1 rank	40	40	Disabled
2 slot, 1 rank	40	40	60
1 slot, 2 rank	40	120	60
2 slot, 2 rank	40	60	120
1 slot, 4 rank	40	120	60

Manually finding the DRAM Configuration Settings

The settings from the previous configuration tables can be manually found in the following locations

- DRAM Configuration, drive strength, ODT, and V_{REF} (read & write)
 - ddrX_0 IP --> Synthesis --> ddrX_0_ddrX.sv
 - Search for the configuration settings of the DRAM type being used
- FPGA IO Standard, drive strength and ODT value
 - XDC: write_xdc
 - IBIS: write_ibis
 - **FPGA Slew Rate is always FAST**

Configuration Settings

DDR4

- Output Driver Impedance: MR1 [2:1]
- R_{TT}(nom): MR1 [10:8]

- Dynamic ODT $R_{TT}(wr)$: MR2 [11:9]
- $R_{TT}(park)$: MR5 [8:6]
- V_{REF} Write: MR6 [6:0] (V_{REF} DQ Training Values: JESD79-4B, Table 16 or JESD79-4C, Table 34)
- Vref Read: RD_ V_{REF} _VAL (UG571, v1.12, Table 1-11)

DDR3

- DRAM Settings
 - Output Drive Strength, MR1[5,1]
 - $R_{TT}(\text{nom})$, MR1 [9,6,2]
 - $R_{TT}(wr)$, MR2[10:9]

FPGA Configuration Settings

Finding IO details in the xdc

- Perform write_xdc from an elaborated design
 - Review the following properties
 - IOSTANDARD
 - OUTPUT_IMPEDANCE
 - SLEW
 - ODT
- Example configuration: PL DDR4, 1dpc (1 DIMM per Channel), dual-rank,
- DQ Signals (DQ0)
 - IOSTANDARD
 - set_property IOSTANDARD POD12_DCI [get_ports {c0_ddr4_dq[0]}]
 - OUTPUT_IMPEDANCE
 - set_property OUTPUT_IMPEDANCE RDRV_40_40 [get_ports {c0_ddr4_dq[0]}]
 - SLEW
 - set_property SLEW FAST [get_ports {c0_ddr4_dq[0]}]
 - ODT: R_{TT_60}
 - set_property ODT R_{TT_60} [get_ports {c0_ddr4_dq[0]}]
 - Equalization
 - set_property EQUALIZATION EQ_LEVEL3 [get_ports [list {c0_ddr4_dq[0]}]]
 - Pre-Emphasis
 - set_property PRE_EMPHASIS RDRV_240 [get_ports [list {c0_ddr4_dq[0]}]]
- ADDR Signals (ADDR0)
 - IOSTANDARD
 - set_property IOSTANDARD SSTL12_DCI [get_ports {c0_ddr4_adr[0]}]
 - OUTPUT_IMPEDANCE
 - set_property OUTPUT_IMPEDANCE RDRV_40_40 [get_ports {c0_ddr4_adr[0]}]
 - SLEW
 - set_property SLEW FAST [get_ports {c0_ddr4_adr[0]}]
 - ODT
 - N/A, Address is output only

Finding details in custom IBIS Model

Perform write_ibis from an elaborated design (See Vivado Generated, Custom IBIS Models section)

Locate the desired signal under the [Pin] keyword, the IBIS model name is called out to the right of the signal name. This IBIS model name contains all the active settings.

Example configuration: PL DDR4, 1dpc, dual rank

- DQ Signals (DQ0)
 - c0_ddr4_dq[0] HP_POD12_DCI_F_OUT40_IN60_PE2400
 - IOSTANDARD = POD12 (POD12_DCI)
 - OUTPUT_IMPEDANCE = 40 (OUT40)
 - SLEW = FAST (F)
 - ODT = 60 (IN60)
- Address Signals (ADDR0)
 - c0_ddr4_adr[0] HP_SSTL12_DCI_F_OUT40
 - IOSTANDARD = SSTL12 (SSTL12_DCI)
 - OUTPUT_IMPEDANCE = 40 (OUT40)
 - SLEW = FAST (F)

Zynq UltraScale+ MPSoC DDR Subsystem Drive Strength, ODT and V_{REF} Configuration

The Zynq MPSoC PS DDR subsystem Memory Controller has been characterized and tested to identify the optimal drive strength, ODT and V_{REF} (initial value) settings. This chapter provides the values that will always be used for the Zynq MPSoC PS Memory Controller with DDR3, LPDDR3, DDR4 and LPDDR4 DRAM interfaces. PS DDR drivers do not have discrete settings for drive strength or slew rate. The drive strength and slew rate can't be adjusted or read. The drive strength and slew rate settings have been derived from characterization. The Memory Controller supports the following calibration routines.

- Write Leveling
 - Write DQS to DQ Deskew
- Read Leveling
 - Per-Bit Deskew
 - Read DQS Centering
- Write Latency Calibration
 - Write DQS to CK alignment
- V_{REF} training
 - The DRAM Write V_{REF} is calibrated, initial values are listed, the initial values represent the typical value.

Zynq MPSoC PS DDR4

- Table 13 provides PS DDR4 FPGA drive strength & ODT configurations
 - FPGA Slew Rate can't be adjusted and is always FAST
- Table 14 provides PS DDR4 DRAM drive strength and ODT configurations
- Table 15 provides PS DDR4 V_{REF} (initial value) configurations

Table 13: PS DDR4 FPGA drive strength & ODT configurations

UltraScale+ PS DDR4	FPGA Driver Strength, Ohm	FPGA ODT, Ohm

Component, 1 or 2 rank	34	40
1 slot, 1 rank	34	40
1 slot, 2 rank	34	40

Table 14: PS DDR4 DRAM drive strength and ODT configurations

UltraScale+ PS DDR4	DRAM Strength, Ohm	R _{TT} (nom), Ohm	R _{TT} (park), Ohm
Component, 1 rank	34	40	40
Component, 2 rank	34	48	240
1 slot, 1 rank	34	40	40
1 slot, 2 rank	34	48	240

Table 15: PS DDR4 V_{REF} (initial value) configurations

UltraScale+ PS DDR4	DC Calculation (Vcc = 1.2V)	
	WRITE V _{REF} , V	WRITE V _{REF} , %
All configurations	0.92	76%

Zynq MPSoC PS LPDDR4

- Table 16 provides PS LPDDR4 FPGA drive strength & ODT configurations
 - FPGA Slew Rate can't be adjusted and is always FAST
- Table 17 provides PS LPDDR4 DRAM drive strength and ODT configurations
- Table 18 provides PS LPDDR4 V_{REF} (initial value) configurations

Table 16: PS LPDDR4 FPGA drive strength & ODT configurations

UltraScale+ PS LPDDR4	FPGA Driver Strength, Ohm	FPGA ODT, Ohm
Component, 1 or 2 rank	40	40
Component, 1 or 2 rank ECC	40	40

Table 17: PS LPDDR4 DRAM drive strength and ODT configurations

UltraScale+ PS LPDDR4	DRAM Driver Strength, Ohm	CA R _{TT} , Ohm	DQ R _{TT} , Ohm
Component, 1 or 2 rank	40	48	40

Component, 1 or 2 rank ECC	40	48	40	
----------------------------	----	----	----	--

Table 18: PS LPDDR4 V_{REF} (initial value) configurations

UltraScale+ PS LPDDR4	DC Calculation ($V_{CC} = 1.1V$)			
	WRITE V_{REF} CA, V	WRITE V_{REF} DQ, V	WRITE V_{REF} CA, %	WRITE V_{REF} DQ, %
Component, 1 or 2 rank	0.34	0.45	31%	41%
Component, 1 or 2 rank ECC	0.34	0.44	31%	40%

Zynq MPSoC PS DDR3/3L

- Table 19 provides PS DDR3 FPGA drive strength & ODT configurations
 - FPGA Slew Rate can't be adjusted and is always FAST
- Table 20 provides PS DDR3 DRAM drive strength and ODT configurations

Table 19: PS DDR3 FPGA drive strength & ODT configurations

UltraScale+ PS DDR3/3L	FPGA Driver Strength, Ohm	FPGA ODT, Ohm
Component, 1 or 2 rank	40	40
1 slot, 1 rank	40	40
1 slot, 2 rank	40	40
1 slot, RDIMM, 1 rank	40	40
1 slot, RDIMM, 2 rank	40	60

Table 20: PS DDR3 DRAM drive strength and ODT configurations

UltraScale+ PS DDR3/3L	DRAM Strength, Ohm	$R_{TT}(\text{nom})$, Ohm	$R_{TT}(\text{wr})$, Ohm
Component, 1 rank	40	60	Disabled
Component, 2 rank	40	120	60
1 slot, 1 rank	40	60	Disabled
1 slot, 2 rank	40	120	60
1 slot, RDIMM, 1 rank	40	60	Disabled
1 slot, RDIMM, 2 rank	40	120	60

UltraScale+ PS LPDDR3

- Table 21 provides PS LPDDR3 FPGA drive strength & ODT configurations
 - FPGA Slew Rate can't be adjusted and is always FAST
- Table 22 provides PS LPDDR3 DRAM drive strength and ODT configurations

Table 21: PS LPDDR3 FPGA drive strength & ODT configurations

UltraScale+ PS LPDDR3	FPGA Driver Strength, Ohm	FPGA ODT, Ohm
Component, 1 or 2 rank	40	120

Table 22: PS LPDDR3 DRAM drive strength and ODT configurations

UltraScale+ PS LPDDR3	DRAM Driver Strength, Ohm	R _{TT} (nom), Ohm
Component, 1 or 2 rank	34.3	120

Manually finding the DRAM Configuration Settings

The settings from the previous configuration tables can be manually found in the following locations

- DRAM Configuration, drive strength, ODT and V_{REF} (if applicable)
 - Export Hardware --> design.hdf --> psu_int.c
- FPGA IO Standard, drive strength and ODT value
 - IBIS: write_ibis: See Zynq PS DDR IBIS Decoder

DDR Configuration Settings

DDR4

- DRAM Settings
 - Output Driver Impedance MR1 [2:1]
 - R_{TT}(nom) MR1 [10:8]
 - Dynamic ODT R_{TT}(wr) MR2 [11:9]
 - R_{TT}(park) MR5 [8:6]
- V_{REF} Calibration
 - DRAM V_{REF}_DQ (Writes)
 - Each byte is independently trained
 - The algorithm searches through one V_{REF}_DQ range between 60% and 90% of Vddq and will pick the center value of the observed valid range
 - DRAM V_{REF}_DQ (Reads)
 - Each byte is independently trained
 - The algorithm only searches through one V_{REF}_DQ range between 45.78% and 91.89% of Vcco and will pick the center value of the observed valid range

LPDDR4

- DRAM Settings
 - Drive Strength: MR3[5:3]
 - CA ODT: MR11[6:4]
 - DQ ODT: MR11[2:0]
 - V_{REF} CA Range: MR12[6:0]: JESD209-4, Table 12
 - V_{REF} DQ: MR14[5:0]: JESD209-4, Table 13
 - SOC ODT: MR22[2:0]
- V_{REF} Calibration
 - V_{REF_CA} fixed at 30.8%
 - DRAM V_{REF_DQ} (Writes)
 - Both channels share the same value
 - The V_{REF} training algorithm searches for the optimal V_{REF_DQ} between 10% and 30% Vddq and will pick the center value. Likely most designs will pass for this entire range and use a value of 20%
 - PS DDR V_{REF_DQ} (Reads)
 - Each channel can have an independent value
 - The V_{REF} training algorithm searches from 7.73% to 53.82% of Vcco and will find the center of the observed valid range

DDR3

- DRAM Settings
 - Output Drive Strength, MR1[5,1]
 - $R_{TT}(\text{nom})$, MR1 [9,6,2]
 - $R_{TT}(\text{wr})$, MR2[10:9]

LPDDR3 Configuration Settings

- DRAM Settings
 - Drive Strength: MR3[3:0]
 - DQ ODT: MR11[1:0]

IBIS Models

Xilinx provides **I/O Buffer Information Specification (IBIS)** models for all supported I/O standards in FPGA and MPSoC devices. This chapter provides guides on how to obtain custom and generic IBIS models. To aid in reviewing the IBIS models, name decoders have been included for both PL and PS models.

Generic and Custom IBIS models can be obtained or generated through the following methods.

1. Generic IBIS models from Xilinx.com
2. Generate a custom IBIS model for PL designs
 - a. Generate a custom PL IBIS File from the I/O Pin Planner
 - b. Creating a Custom IBIS File from an Implemented Design
3. Generate a custom IBIS model for the Zynq MPSoC PS DDR

DDR specific IBIS model name decoders have been included for PL and PS models.

1. PL I/O Standards
2. PS DDR I/O Standards

These represent a subset of the IBIS models provided by Xilinx. For a complete IBIS decoding guide, please see the blog post "[Xilinx PL and PS IBIS Model Decoders](#)"

Generic IBIS Models from [Xilinx.com](#)

The generic IBIS model file contains models for all IO Standards supported by the selected family. The generic IBIS model is package and die size agnostic. A global RLC package model is set for every available IO. The generic IBIS model contains every available IO model with no specific pin assignments. The generic models are recommended for a small number of signals in a schematic level simulation. It is not recommended to use the generic models for board level simulations.

Figure 1: Capture of available UltraScale+ IBIS Models from [Xilinx.com](#)

UltraScale+ SelectIO IBIS Models

Below is the latest version of the UltraScale+™ SelectIO IBIS Models

Description	Filename	Size	Date
Kintex UltraScale+	kintexusplus.zip	30 MB	12/3/2019
Virtex UltraScale+	virtexplus.zip	33 MB	12/3/2019
Virtex UltraScale+ 58G	virtexusplus58g.zip	27 MB	12/3/2019
Virtex UltraScale+ HBM	virtexplusHBM.zip	27 MB	12/3/2019
Zynq UltraScale+ MPSoC	zynquplus.zip	35 MB	12/3/2019
Zynq UltraScale+ RFSoC	zynquplusRFSoC.zip	33 MB	12/3/2019

Figure 1: Available UltraScale+ IBIS Models from [Xilinx.com](#)

Generic IBIS models can be found [here](#)

Vivado Generated, Custom IBIS Models

The Vivado generated, custom IBIS models are device, package and design specific. The Vivado generated IBIS models can be used for Board Level and Schematic Level simulations. The custom IBIS models include RLC package details for each individual package pin.

When generating a Custom IBIS model with a DDR IP (PL or PS) you must use the default IO settings like slew, equalization, ODT and drive strength. These parameters are set by the memory configuration (FPGA, DRAM Type and Topology). This information is summarized in the PL Controller and PS Controller Drive Strength, ODT and V_{REF} Configurations.

How to generate custom IBIS Files from Vivado

The following guides can be used to create custom PL or PS IBIS file(s) from within Vivado:

- Creating a Custom PL IBIS File from the I/O Pin Planner
- Creating a Custom PL IBIS File from an Implemented Design
- Creating a Custom IBIS File for Zynq PS DDR

Generate a custom PL IBIS File from the I/O Pin Planner

Creating an IBIS file from the I/O Pin Planner is useful for performing signal integrity simulations even if full Vivado design is not yet available. If the pin-out of the Xilinx device is known, along with the required I/O standards and parameters, a custom IBIS file can be created.

The procedure is as follows:

- Open Vivado and double-click "Create New Project" (Figure 2)



Figure 2: Create Project

- On the “Create a New Vivado Project” window (Figure 3), click Next:

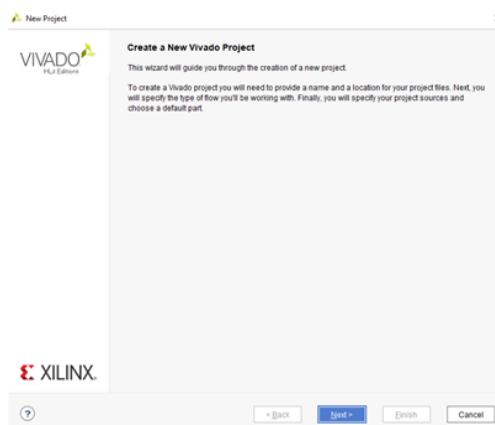


Figure 3: Create New Vivado Project

- Enter a project name and directory (Figure 4), then click Next:

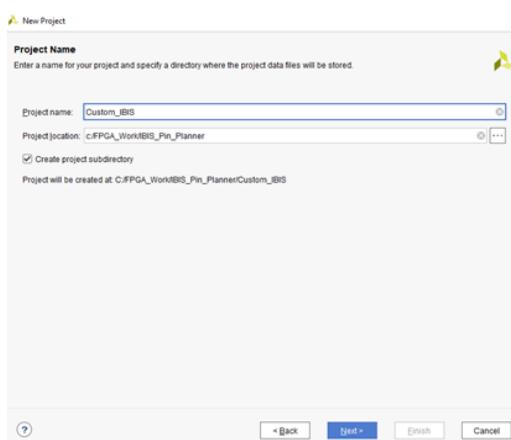


Figure 4: Project Name

- Choose “I/O Planning Project” (Figure 5), then click Next:

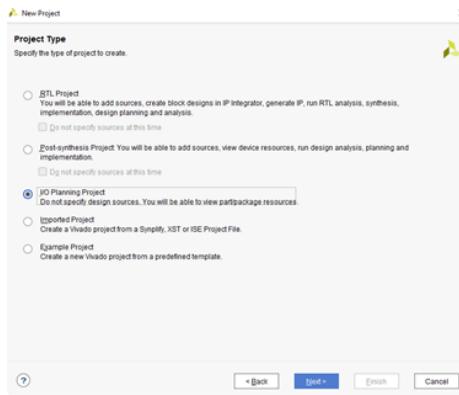


Figure 5: Project Type

- If you already have a CSV (or XDC file) that defines the pins and I/O standards, you can link to it now, or wait until later. For this document, we will import a CSV file at a later stage.
 - Choose “Do not import I/O ports at this time” (Figure 6) and click Next:

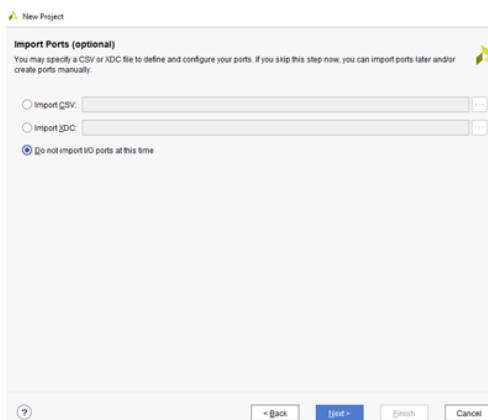


Figure 6: Import Ports option

- Choose the relevant Xilinx Part (Figure 7) and click Next:

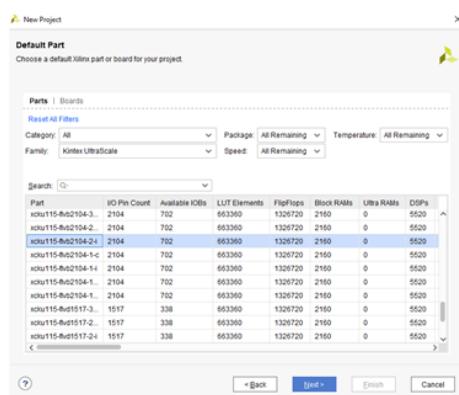


Figure 7: Part Selection

- New Project Summary (Figure 8) Click Finish:

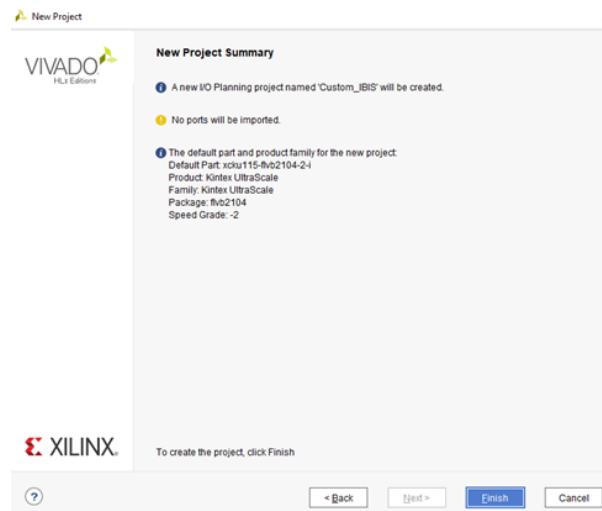


Figure 8: New Project Summary

- When the project is created, a device view will be shown (Figure 9):

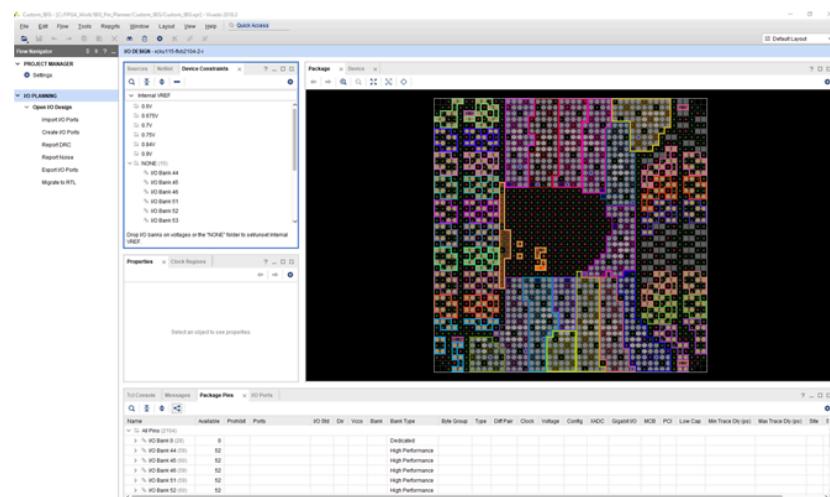


Figure 9: Device View

- At this point, we are ready to begin the process of importing the ports from a CSV file. There is a way to manually create ports using the "Create I/O Ports" option, but this process takes longer and will not be covered at this time. Instead, it is recommended to import a CSV file containing the port definitions.
- Before importing the CSV file, it must be created.
- The CSV file has a particular format that is explained in greater detail in [UG899: Vivado Design Suite User Guide, I/O and Clock Planning](#), though an example is shown below that pertains to a DDR4 memory design.

Example CSV File

- Figure 10 shows a complete CSV file that contains the relevant column headers, and examples of values that are suited for a DDR4 memory design containing 64 data bits (with Data Masks). Columns H, and I are used for differential signals such as clocks and DQS strobes.

Pin	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
					Slew												
1	Name	Signal Name	Direction	IO Standard	Output Impedance	ODT	Type	Supply Voltage	Drive	PRE_EMP	EQUALIZAT	DIFFPAIR	DIFFPAIR	DQS_BIA	DIFF_TER	OFFCHIP	
2	W43	c0_ddr4_dqs[17]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	VDDQ	RDRV_240_EQ_LEVEL2								FP_VCCO_50
3	W33	c0_ddr4_dqs[16]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dqs[16]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
4	A131	c0_ddr4_dqs[15]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dqs[15]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
5	AC31	c0_ddr4_dqs[14]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dqs[14]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
6	AH34	c0_ddr4_dqs[13]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dqs[13]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
7	AH31	c0_ddr4_dqs[12]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dqs[12]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
8	AJ27	c0_ddr4_dqs[11]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dqs[11]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
9	AJ26	c0_ddr4_dqs[10]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dqs[10]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
10	AJ33	c0_ddr4_dqs[9]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dqs[9]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
11	AN12	c0_ddr4_dqs[8]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dqs[8]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
12	BA34	c0_ddr4_dqs[7]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dqs[7]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
13	AW35	c0_ddr4_dqs[6]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dqs[6]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
14	BC34	c0_ddr4_dqs[5]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dqs[5]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
15	BU34	c0_ddr4_dqs[4]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dqs[4]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
16	BP34	c0_ddr4_dqs[3]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dqs[3]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
17	BD40	c0_ddr4_dqs[2]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dqs[2]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
18	AP31	c0_ddr4_dqs[1]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dqs[1]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
19	AM29	c0_ddr4_dqs[0]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dqs[0]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
20	AA33	c0_ddr4_dqs[17]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	N	c0_ddr4_dqs[17]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
21	Y31	c0_ddr4_dqs[16]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	N	c0_ddr4_dqs[16]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
22	AZ31	c0_ddr4_dqs[15]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	N	c0_ddr4_dqs[15]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
23	AJ31	c0_ddr4_dqs[14]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	N	c0_ddr4_dqs[14]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
24	AJ34	c0_ddr4_dqs[13]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	N	c0_ddr4_dqs[13]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
25	AH32	c0_ddr4_dqs[12]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	N	c0_ddr4_dqs[12]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
26	AK27	c0_ddr4_dqs[11]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	N	c0_ddr4_dqs[11]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
27	AH29	c0_ddr4_dqs[10]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	N	c0_ddr4_dqs[10]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
28	AJ34	c0_ddr4_dqs[9]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	N	c0_ddr4_dqs[9]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
29	AK31	c0_ddr4_dqs[8]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	N	c0_ddr4_dqs[8]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
30	AK34	c0_ddr4_dqs[7]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	N	c0_ddr4_dqs[7]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
31	AW36	c0_ddr4_dqs[6]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	N	c0_ddr4_dqs[6]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
32	BD34	c0_ddr4_dqs[5]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	N	c0_ddr4_dqs[5]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
33	BC37	c0_ddr4_dqs[4]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	N	c0_ddr4_dqs[4]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
34	BF40	c0_ddr4_dqs[3]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	N	c0_ddr4_dqs[3]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
35	BD40	c0_ddr4_dqs[2]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	N	c0_ddr4_dqs[2]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
36	AR31	c0_ddr4_dqs[1]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	N	c0_ddr4_dqs[1]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
37	AK34	c0_ddr4_dqs[0]	INPUT	DIFF_POD12_DQ	FAST	RDRV_40_40	RTT_40	N	c0_ddr4_dqs[0]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
38	Y30	c0_ddr4_dq[7]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40			RDRV_240_EQ_LEVEL2							FP_VCCO_50
39	Y30	c0_ddr4_dq[6]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40			RDRV_240_EQ_LEVEL2							FP_VCCO_50
40	AJ34	c0_ddr4_dq[5]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40			RDRV_240_EQ_LEVEL2							FP_VCCO_50
41	AJ34	c0_ddr4_dq[6]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40			RDRV_240_EQ_LEVEL2							FP_VCCO_50

Pin	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
					Slew												
1	W43	c0_ddr4_dq[6]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[6]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
43	W33	c0_ddr4_dq[5]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[5]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
44	Y33	c0_ddr4_dq[4]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[4]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
45	Y33	c0_ddr4_dq[3]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[3]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
46	AJ34	c0_ddr4_dq[2]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[2]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
47	AJ34	c0_ddr4_dq[1]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[1]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
48	AJ33	c0_ddr4_dq[0]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[0]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
49	AJ33	c0_ddr4_dq[60]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[60]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
50	AD34	c0_ddr4_dq[59]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[59]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
51	AC34	c0_ddr4_dq[58]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[58]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
52	AC33	c0_ddr4_dq[57]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[57]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
53	AJ34	c0_ddr4_dq[56]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[56]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
54	AJ34	c0_ddr4_dq[55]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[55]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
55	AJ34	c0_ddr4_dq[54]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[54]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
56	AJ33	c0_ddr4_dq[53]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[53]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
57	AJ33	c0_ddr4_dq[52]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[52]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
58	AJ33	c0_ddr4_dq[51]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[51]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
59	AJ32	c0_ddr4_dq[50]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[50]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
60	AJ32	c0_ddr4_dq[49]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[49]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
61	AJ33	c0_ddr4_dq[48]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[48]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
62	AJ30	c0_ddr4_dq[47]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[47]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
63	AJ29	c0_ddr4_dq[46]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[46]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
64	AJ28	c0_ddr4_dq[45]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[45]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
65	AJ28	c0_ddr4_dq[44]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[44]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
66	AJ31	c0_ddr4_dq[43]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[43]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
67	AJ31	c0_ddr4_dq[42]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[42]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
68	AJ30	c0_ddr4_dq[41]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[41]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
69	AJ30	c0_ddr4_dq[40]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[40]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
70	AP34	c0_ddr4_dq[39]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[39]	RDRV_240_EQ_LEVEL2							FP_VCCO_50
71	AN34	c0_ddr4_dq[38]	INPUT	POD12_DQ	FAST	RDRV_40_40	RTT_40	P	c0_ddr4_dq[38]	RDRV_240_EQ_LEVEL2							

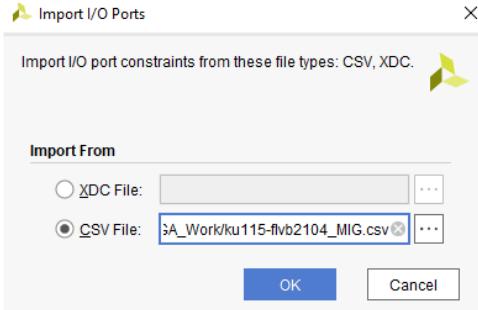


Figure 12: Import I/O Ports popup window

- The file will import, and package pins can be seen via the “I/O Ports” tab (Figure 13).
 - Any errors or warnings will be listed in the “Tcl Console” tab:

The screenshot shows the "I/O Ports" tab of a software interface. The table lists numerous package pins with the following columns: Name, Direction, Neg Diff Pair, Package Pin, Fixed, Bank, IO BM, Vtob, Vref, Drive Strength, Slew Type, Pull Type, and Off-Chip Termination. Many rows have "FP_VTT_50" listed under Off-Chip Termination. The "Name" column contains entries such as "c0_d94_a@{12}" through "c0_d94_a@{21}", "c0_d94_b@{12}" through "c0_d94_b@{21}", and "c0_d94_c@{12}" through "c0_d94_c@{21}".

Figure 13: I/O Ports tab

- Now, the IBIS file can be exported (Figure 14).
 - Choose File->Export->Export IBIS Model

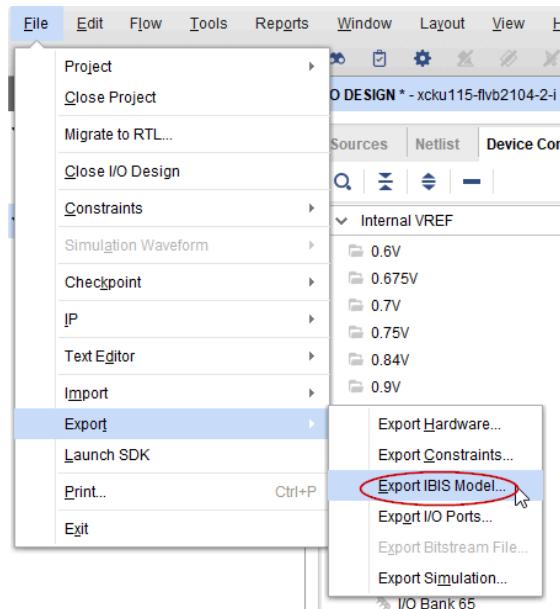


Figure 14: Export IBIS Model

- Export IBIS Model (Figure 15)
 - Choose an output file name and location (or keep default)
 - Keep “Include all models” UNCHECKED to keep the file size small
 - Keep “Disable per pin modeling” UNCHECKED: This will combine the package data with the IBIS models
 - Choose a Component Name, not required. For this example, it has been left blank.

- Click OK to export the file

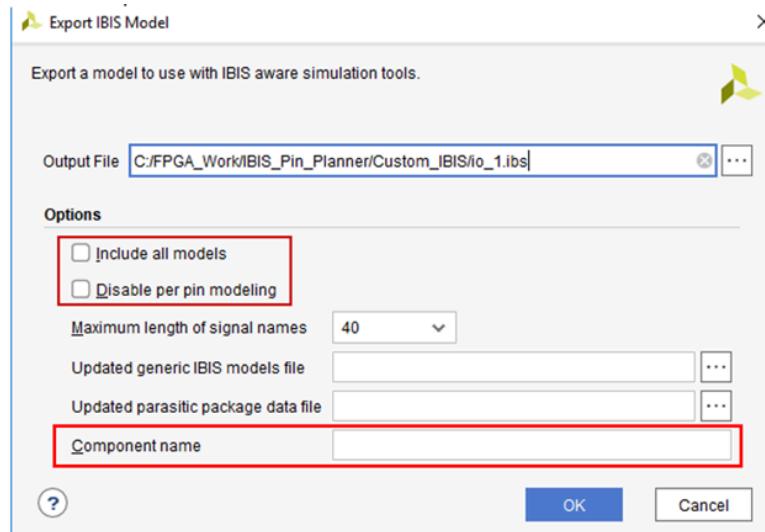


Figure 15: Export IBIS Model, I/O Planner

- The IBIS file is now ready for use in simulation.

Creating a Custom PL IBIS File from an Implemented Design

The procedure for creating a custom IBIS file from an implemented design is very straight-forward, as all pin assignments have been completed as part of the design implementation.

- Begin by opening the project and then click “Open Implemented Design” (Figure 16)

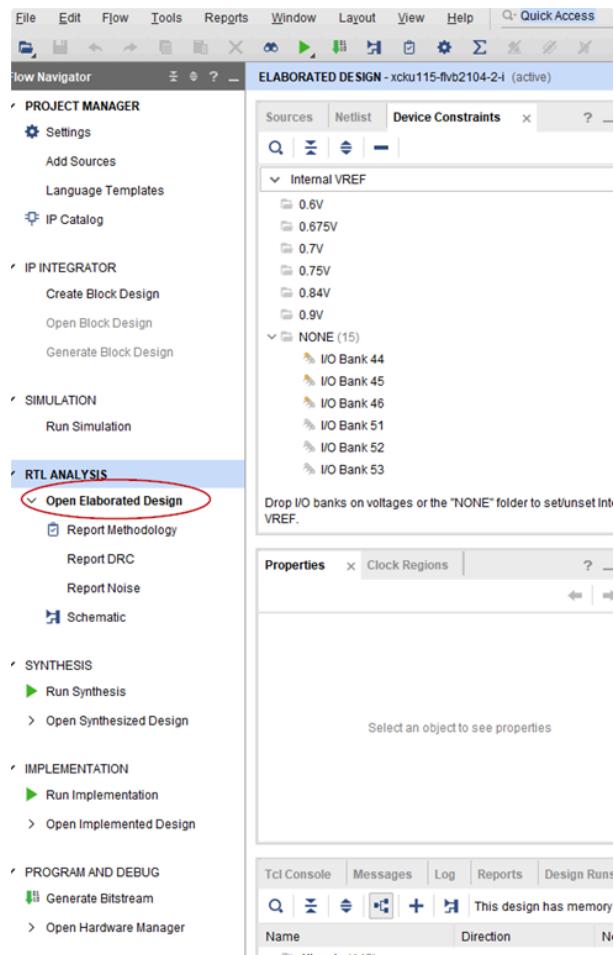


Figure 16: Implement Design

- Choose File->Export->Export IBIS Model

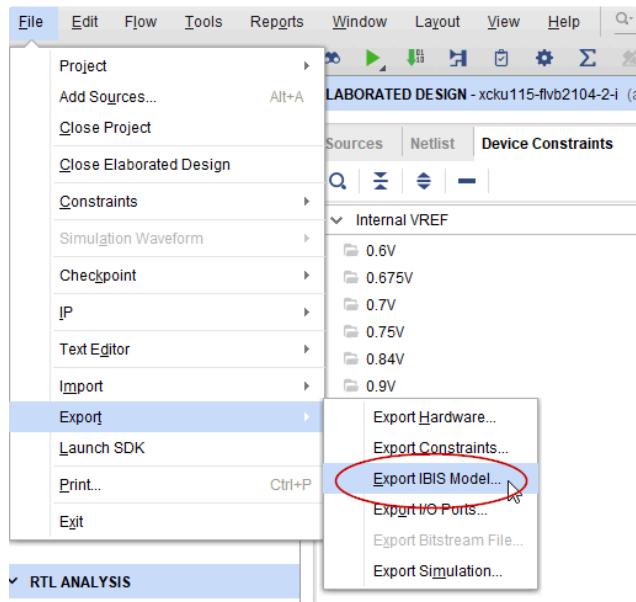


Figure 17: Export IBIS Model from Implemented Design

- Export IBIS Model (Figure 18)
 - Choose an output file name and location (or keep default)
 - Keep “Include all models” UNCHECKED to keep the file size small
 - Keep “Disable per pin modeling” UNCHECKED: This will combine the package data with the IBIS models
 - Choose a Component Name, not required. This example calls out the FPGA device.
 - Click OK to export the file

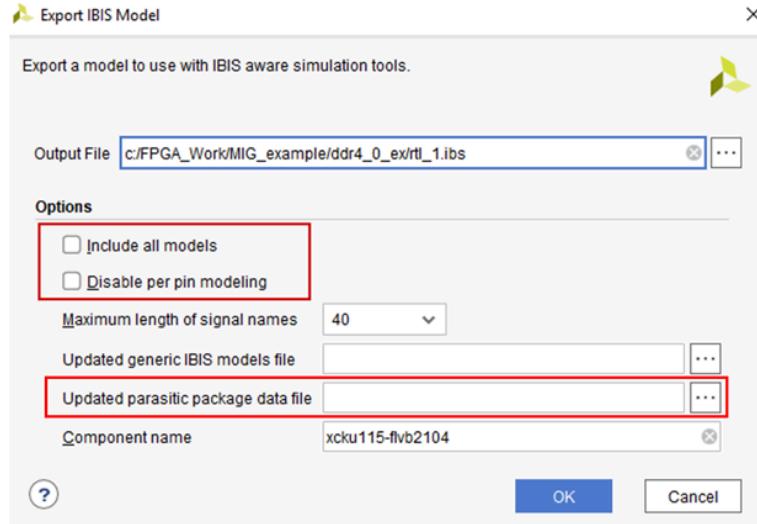


Figure 18: Export IBIS Model, Implemented Design

- The IBIS file is now ready for use in simulation.

How to Generate Custom IBIS File for Zynq PS DDR

To generate a custom IBIS file for the Zynq PS DDR launch a new RTL Project in Vivado.

- Select the target Zynq device and package.
- When the new project is ready, create a block design in the IP Integrator (Figure 19)

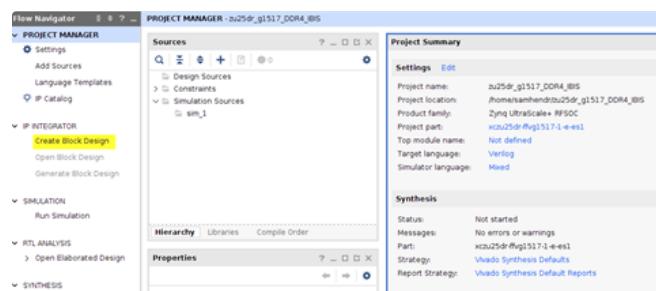


Figure 19: Create Block Design

- Click on the “+” button to add IP, in the IP selector window, select “Zynq UltraScale+ MPSoC”

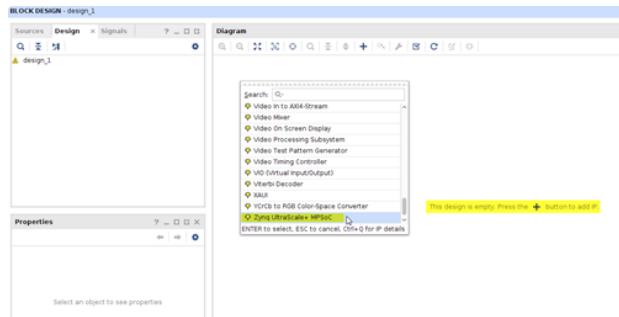


Figure 20: Add Zynq UltraScale+ MPSoC Block

- The zynq_ultra_ps_e_0 will be added to the diagram window



Figure 21: Zynq UltraScale+ MPSoC Block

- Double click on the Zynq block to open the Re-customize IP window (Figure 22). Within the Re-customize IP window select “DDR Configuration” in the Page Navigator. Select the appropriate DDR technology, Bus width, and ECC option. It is not necessary to select anything else for the IBIS model. In the example below the speed, bin has been updated to DDR4 2400P.

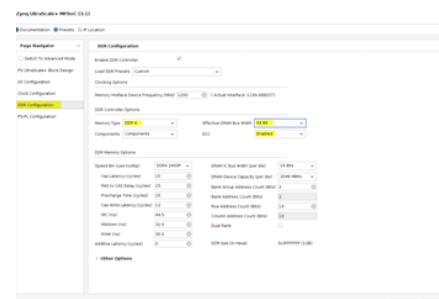


Figure 22: Zynq Re-customize IP Window (PCW)

- If needed, “Other options” can be expanded so “Parity” can be selected.

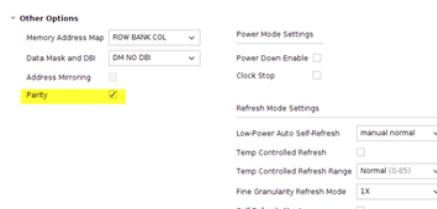


Figure 23: Other Memory Options

- Click ok, the IP will be updated. Right-click on the Zynq IP block, select “Make External.”

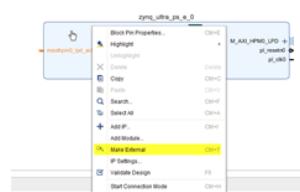


Figure 24: Make Ports External

- Vivado will automatically add external signals (Figure 25)



Figure 25: Zynq Block with External Ports

- To avoid receiving a warning about an un-assigned address click on the "Address Editor" tab (Figure 26) and select "Auto Assign Address"

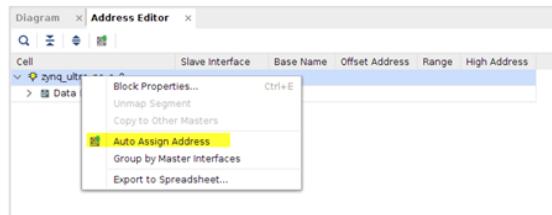


Figure 26: Address Editor

- In the Hierarchy window, click on the "Sources" tab. Right-click on the design and select "Create HDL Wrapper" (Figure 27)

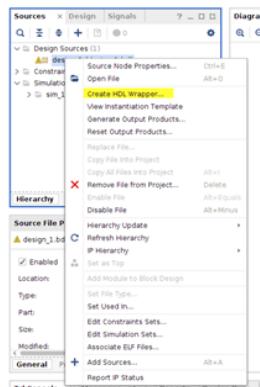


Figure 27: Create HDL Wrapper

- Select "Let Vivado manage wrapper and auto-update," (Figure 28) press OK.

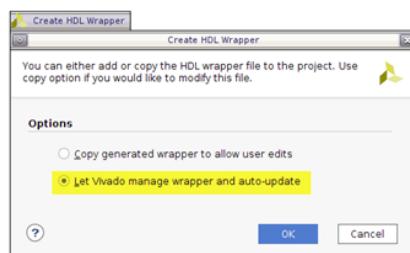


Figure 28: Create HDL Wrapper options

- When the wrapper is complete select "Open Elaborated Design" (Figure 29) in the Flow Navigator.

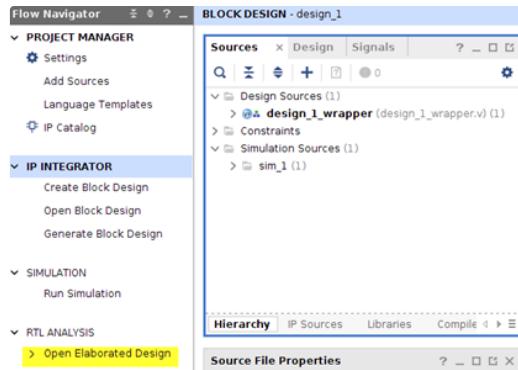


Figure 29: Open Elaborated Design

- A dialog box will open with details about the Elaborated design (Figure 30). Press OK.

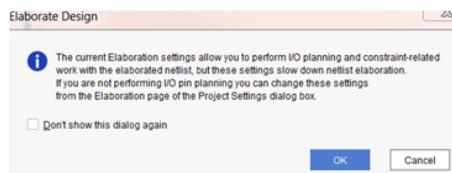


Figure 30: Elaborated Design Popup

- When the Elaborated Design is open, Select File à Export à Export IBIS Model (Figure 31).

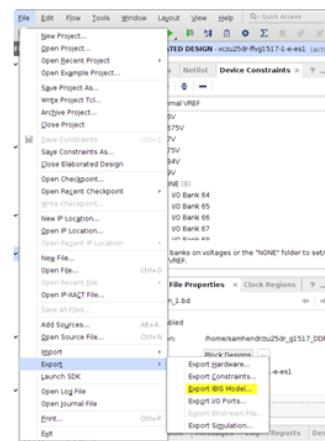


Figure 31: Export PS IBIS Model

- The Export IBIS Model window will open (Figure 32). Set the output path, ensure “Include all models” and “Disable per pin modeling” boxes are not selected. Component name can be assigned, but not necessary.

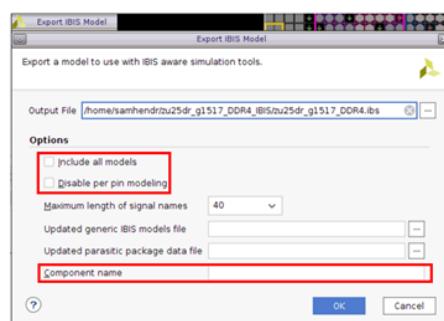


Figure 32: Export IBIS Model Options

- The custom PS IBIS file will now be generated.

Xilinx PL IBIS Decoder

The PL IBIS Decoder (Table 23) can be used to decode PL IBIS models for all PL I/Os. This applies to the Zynq MPSoC PL I/Os.

Table 23: PL IBIS Decoder

Xilinx PL IBIS Model Settings		
Model Setting	Options	Available Documentation
Base Model	BANK-TYPE_IOSTANDARD_SLEW_OUTPUT-IMPEDANCE*_INPUT-ODT_PRE-EMPHASIS • OUTPUT-IMPEDANCE or OUTPUT-DRIVE-STRENGTH depending on IOSTANDARD	
BANK-TYPE	HP, HR, HD	UG571, Ch 1, I/O Tile Overview
IOSTANDARD	See supported I/O standard for BANK-TYPE	HP & HR: UG571, Ch 1, Supported I/O Standards and Terminations HD: UG571, Ch 3, HD I/O Supported Standards
SLEW	FAST, MEDIUM, SLOW	UG571, Ch 1, Output Slew Rate Attributes
OUTPUT-IMPEDANCE (Ohm)*	40, 48, 60 or None	UG571, Ch 1, Source Termination Attribute (OUTPUT_IMPEDANCE)
OUTPUT-STRENGTH (mA)*	4, 8, 12 or 16	UG571, Ch 1, Output Drive Strength Attributes
INPUT-ODT (Ohm)	40, 48, 60, 120, 240 or None	UG571, Ch 1, On-Die Termination (ODT) Attribute
PRE-EMPHASIS	PE1600 or PE2400	UG571, Ch 1, Transmitter Pre-Emphasis

All models (except for LVDS*) will contain Bank Type, IOStandard, Slew Rate and Output Impedance/Drive Strength.

- LVDS models will contain Bank Type, LVDS IOStandard and Digital Termination.

Internal 100-ohm Differential Termination is only available in banks powered at 1.8V (LVDS) or 2.5V (LVDS_25). See [\(UG571\), v1.12, p.130](#) for more details.

Note: not all model settings will be present in every IBIS model. If a setting is not in the model name, that setting is not supported by the model.

Table 24 provides an example of the PL DDR4 IBIS Models.

Table 24: PL DDR4 IBIS Models

Example DDR4 IBIS Models		
Signal Name	Model Name	Model Settings
DQ, DQS, DM	HP_POD12_DCI_F_OUT40_IN40_PE240 0	Bank type: HP IO Standard: POD Bank Voltage: 1.2V Digitally Controlled Impedance (DCI) Slew rate: Fast Output Impedance: 40 ohm

		Input ODT: 40 Ohm Pre-Emphasis enabled
Clock, Address & Command	HP_SSTL12_DCI_F_OUT40	Bank type: HP IO Standard: SSTL Bank Voltage: 1.2V DCI Slew rate: Fast Output Impedance: 40 ohm

Xilinx Zynq PS DDR IBIS Decoder

The Zynq MPSoC PS DDR IBIS signals are unique from all other signals.

Table 25 is a decoder for the Zynq MPSoC PS DDR IBIS models.

Examples of each DDR memory type IBIS model are provided in Tables 26 through 30

Table 25: Zynq PS DDR IBIS Decoder

Zynq PS DDR IBIS Decoder	
Base Model	DWC_D5MXY_Z_MS
X	IO Type
C	Clock, Command, Control, Address
P	Data, Data Mask
Q	Data Strobe
Y	Memory Technology
3	DDR3
3L	DDR3L
L3	LPDDR3
4	DDR4
L4	LPDDR4
Z*	Output Impedance and/or Input Termination <ul style="list-style-type: none"> Output and termination impedances are not adjustable on the PS DDR controller.
xx	Output impedance xx = 34 ohms (DDR4) xx = 40 ohms (DDR3, DDR3L, LPDDR3, LPDDR4)
ODTxx	Input termination xx = 40 ohms (DDR3, DDR3L)
xxODTy	Output Impedance with input termination (DDR4, LPDDR3 & LPDDR4) xx = 34 ohms (DDR4) xx = 40 ohms (LPDDR3) xx = 80 ohms (LPDDR4)

	yy = 40 ohms (DDR4, LPDDR4) yy = 120 ohms (LPDDR3)	
MS	Model Selector	
	The DQ and DQS models support the Model Selector feature. This is indicated by the "_MS" suffix. This will point the simulator to the [Model Selector] keyword in the IBIS file. The [Model Selector] keyword defines the ODT behavior.	

Zynq PS DDR IBIS Examples

Table 26: PS DDR4

DDR4		
Signal Name	Model Name	Notes
PS_DDR4_CK_P/N (OUT)	DWC_D5MC4_34	Clock
PS_DDR4_A* (OUT)	DWC_D5MC4_34	Applies to Address, Command and Control
PS_DDR4_DQ* (IN/OUT)	DWC_D5MP4_34ODT_MS	See Model Selector Example
PS_DDR4_DQS* (IN/OUT)	DWC_D5MQ4_34ODT_MS	See Model Selector Example

Model Selector Example		
DWC_D5MP4_34ODT_MS	DWC_D5MP4_34 or DWC_D5MP4_34ODT40	
DWC_D5MQ4_34ODT_MS	DWC_D5MP4_34 or DWC_D5MP4_34ODT40	

Table 27: PS LPDDR4

LPDDR4		
Signal Name	Model Name	Notes
PS_LPDDR4_CK_P/N (OUT)	DWC_D5MCL4_40	Clock
PS_LPDDR4_A* (OUT)	DWC_D5MCL4_40	Applies to Address, Command and Control
PS_LPDDR4_DQ* (IN/OUT)	DWC_D5MPL4_80ODT_MS	See Model Selector Example
PS_LPDDR4_DQS* (IN/OUT)	DWC_D5MQL4_80ODT_MS	See Model Selector Example

Model Selector Example		
DWC_D5MPL4_80ODT_MS	DWC_D5MPL4_40 or DWC_D5MPL4_80ODT40	
DWC_D5MQL4_80ODT_MS	DWC_D5MQL4_40 or DWC_D5MQL4_80ODT40	

Table 28: PS DDR3

DDR3		
Signal Name	Model Name	Notes
PS_DDR3_CK_P/N (OUT)	DWC_D5MC3_40	Clock
PS_DDR3_A* (OUT)	DWC_D5MC3_40	Applies to Address, Command and Control
PS_DDR3_DQ* (IN/OUT)	DWC_D5MP3_ODT_MS	See Model Selector Example
PS_DDR3_DQS* (IN/OUT)	DWC_D5MQ3_ODT_MS	See Model Selector Example

Model Selector Example		
DWC_D5MP3_ODT_MS	DWC_D5MP3_40 or DWC_D5MP3_ODT40	
DWC_D5MQ3_ODT_MS	DWC_D5MQ3_40 or DWC_D5MQ3_ODT40	

Table 29: PS DDR3L

DDR3L		
Signal Name	Model Name	Notes
PS_DDR3L_CK_P/N (OUT)	DWC_D5MC3L_40	Clock
PS_DDR3L_A* (OUT)	DWC_D5MC3L_40	Applies to Address, Command and Control
PS_DDR3L_DQ* (IN/OUT)	DWC_D5MP3L_ODT_MS	See Model Selector Example
PS_DDR3L_DQS* (IN/OUT)	DWC_D5MQ3L_ODT_MS	See Model Selector Example

Model Selector Example		
DWC_D5MP3L_ODT_MS	DWC_D5MP3L_40 or DWC_D5MP3L_ODT40	
DWC_D5MQ3L_ODT_MS	DWC_D5MQ3L_40 or DWC_D5MQ3L_ODT40	

Table 30: PS LPDDR3

LPDDR3		
Signal Name	Model Name	Notes
PS_LPDDR3_CK_P/N (OUT)	DWC_D5MCL3_40	Clock
PS_LPDDR3_A* (OUT)	DWC_D5MCL3_40	Applies to Address, Command and Control
PS_LPDDR3_DQ* (IN/OUT)	DWC_D5MPL3_40ODT_MS	See Model Selector Example
PS_LPDDR3_DQS* (IN/OUT)	DWC_D5MQL3_40ODT_MS	See Model Selector Example

Model Selector Example		
DWC_D5MPL3_40ODT_MS	DWC_D5MPL3_40 or DWC_D5MPL3_40ODT120	
DWC_D5MQL3_40ODT_MS	DWC_D5MQL3_40 or DWC_D5MQL3_40ODT120	

Available Resources for HyperLynx and ADS Simulation Tools

Xilinx provides the following resources to aid in performing DDR interface simulations. The UltraScale+ content is available through the UltraScale+ Signal and Power Integrity Lounge or upon request. Please submit requests through your FAE.

UltraScale+

UltraScale+ Signal and Power Integrity Lounge

- DDR4 ADS Simulation Kit
- HyperLynx DDRx Timing models
 - The PL timing models are applicable for all devices across Virtex UltraScale+, Kintex UltraScale+ and Zynq UltraScale+ families
 - PL DDR4 @ 2667Mbps
 - PL DDR3L @ 1866Mbps
 - The PS models are only applicable for the Zynq UltraScale+ family
 - PS DDR4 @ 2400Mbps
 - PS LPDDR4 @ 2400Mbps
 - PS DDR3L @ 1866Mbps

HyperLynx DDRx Tips

This section provides some tips on how to properly setup the HyperLynx DDRx wizard. This is not a comprehensive guide on how to use the DDRx wizard.

ODT Models

ODT Models need to be set in the DDRx Wizard. Use the appropriate impedance and ODT values for the PL or PS Controller and DDR type. The capture below is the setup for the Xilinx ZCU104, 64-bit component DDR4 interface.

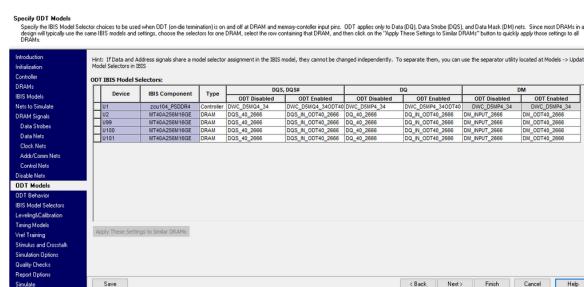


Figure 33: DDRx ODT Models

Leveling and Calibration

The PL and PS DDR controllers support the leveling and calibration settings in the capture below. It is recommended to select "Automatically run simulation 2 times: to estimate skews and to apply them"

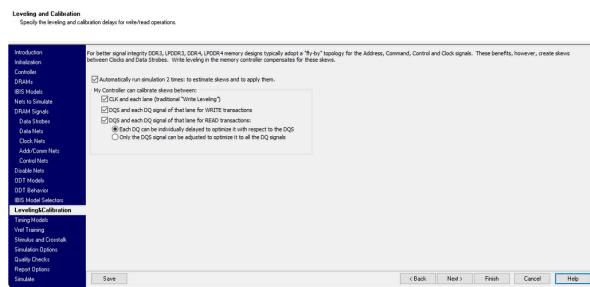


Figure 34: DDRx Leveling and Calibration

Timing Models

For the most accurate DDRx simulation, it is recommended to use the Xilinx timing models.

- The Xilinx timing model numbers will need to be entered using the “TM Wizard”
- In the capture below, the TM Wizard was used to create “PS_DDR4_2400Mbps_DDRx.v”

If Xilinx timing models for the target data rate are not available, two options

- Use Xilinx timing models for the closest available data rate and simulate at that data rate
- Use the generic timing HyperLynx DDRx timing model for your target data rate

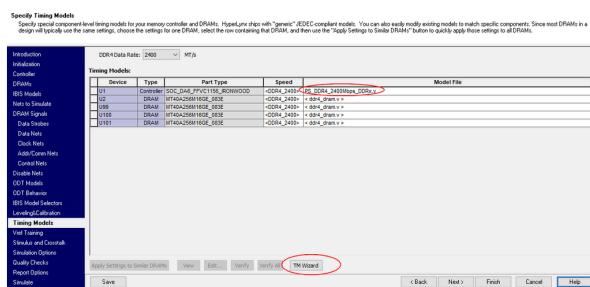


Figure 35: DDRx Timing Models

V_{REF} Training

For PL DDR4, PS DDR4 and PS LPDDR4 select “Single V_{REF} for all the signals”.

- PL DDR4 V_{REF} is fixed: For details on the PL DDR4 V_{REF} see PL Controller ODT/ V_{REF} Configuration, DDR Configuration Settings for DDR4
- PS DDR4 and PS LPDDR4 V_{REF} is calibrated: For details on the PS DDR4 and PS LPDDR4 V_{REF} see PS Controller ODT/ V_{REF} Configuration, DDR Configuration Settings for DDR4 and LPDDR4.



Figure 36: DDRx V_{REF} Training

Simulation Options

Set desired IC model corners. In HyperLynx version 2.4 and newer, the measurement location can be set on the Simulation options page.

- The DRAM specs call for timing analysis to be performed at the pin of the DRAM
- The FPGA has relatively long package signal lengths, thus probing should be done at the die of the FPGA

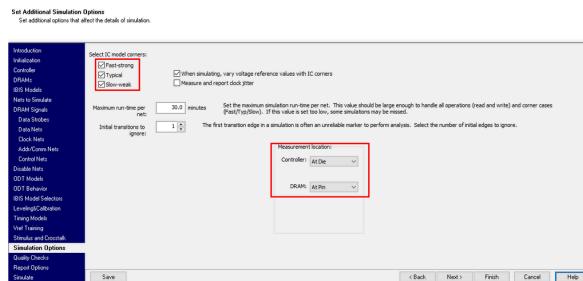


Figure 37: DDRx Simulation Options

Figure 38 shows an example of probing at the Zynq pin vs Zynq die on ZCU104, PS DDR4, DQ4

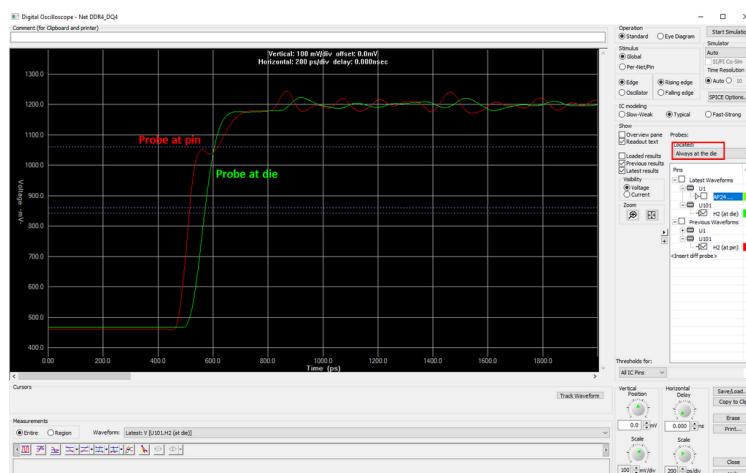


Figure 38: Probe Location Example

If using a HyperLynx version before v2.4, the Xilinx Custom IBIS model will need to be modified to indicate the timing location. If no timing location is in the IBIS model, timing will be performed at the pin.

- Mentor KB Article ID# MG83246: HyperLynx SI: Probing at the Die for Batch and Interactive Simulations
- Virtex UltraScale Plus IBIS example (Figure 39)



Figure 39: IBIS File Modification for Probe Point