SYNOPSYS®

DesignWare Cores DDR4 multiPHY

Verification Environment Quickstart Guide

DWC DDR4 multiPHY

June 28, 2017

Copyright Notice and Proprietary Information Notice

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at

http://www.synopsys.com/Company/Pages/Trademarks.aspx.

All other product or company names may be trademarks of their respective owners.

Synopsys, Inc.
700 E. Middlefield Road
Mountain View, CA 94043

www.synopsys.com

Revision History

Version	Date	Note
1.3	June 28, 2017	Updated: Design Kit Installation o Chaged reference to Installation instruction Configuration Example Design
1.2	October 31, 2013	Updated: The directory structure used in the PHY deliverables
1.1	October 15, 2013	Updated: Design Kit Installation Running Basic Verification
1.0	January 14, 2013	Initial release

Table of Contents

1 Design Kit Installation

This guide is for use with:

The DesignWare Cores DDR4 multiPHY PUB
The DesignWare Cores SDRAM Universal DDR Enhanced Memory Controller (uMCTL2)

For the suggested design kit installation, refer to the 'Installation Instructions' *Appendix of the DWC DDR4 multiPHY.* Portions of this document assume the suggested installation has been followed.

After installation, a directory structure is created. For details on the PHY IP structure and deliverables, refer to the *DWC DDR4 multiPHY Databook*.

Note: It is recommended that all the intermediate simulation or other verification tools be stored outside of the IP structure

2 Example Design

The installation kit contains example design files for use with the verification environment.
<IP_dir>/synopsys/ddr4_multiphy<process>/Latest/macro/Latest/example/rtl

DW_tap.v
DWC_DDRPHY_CHIP.v

<IP_dir>/synopsys/ddr4_multiphy<process>/Latest/macro/Latest/example/rtl

DWC_DDRPHY.v
DWC_DDRPHY_define.v
DWC_DDRPHY_top.v
DWC_DDRPHYAC_io.v
DWC_DDRPHYDAT8_io.v
DWC_DDRPHYAC_top.v
DWC_DDRDATX8_top.v
DWC_DDRPHY_atpg_define.v

3 Running Basic Verification

The "runtc" file allows the user to run basic verification and is located:
./synopsys/ddr4_multiphy<process>/Latest/macro/Latest/sim/

After the IP files are installed, users should prepare the working environment as following:

1. Copy the full verification environment directory to your working directory:

```
mkdir <IP_dir>/sim
cp <IP_dir>/synopsys/ddr4_multiphy<process>/Latest/macro/Latest/sim /* ./sim/
```

2. Go to the verification environment sim directory:

```
cd <IP_dir>/sim
```

"sim_vcs" is an example of batch simulation of different modes. Launch specific "runtc" specifying the options or launch "sim_vcs"

Note: Some components that are customer specific maybe added or edit in "runtc" copy in working directory

3.1 Configuration

The configuration of the user application is performed through a number of options that are specified with "runtc" :

Table 1: Configuration Options range/default value

Option	Description	Range	Default
dw	size in bits	8,16,24,32,40,48,56,64,72	32
ranks	No of ranks	1,2,3,4	2
mbps	mega-bits-per-second	667 to 2133	1600
-ddr3	DDR3 mode	ddr3, ddr4, lpddr3, lpddr2	ddr3
-rr	rise-rise, rise-fall	rr, rf	rf
bl	Burst-length	bl4, bl8, bc4, bc8	bl8

Note: The verification as part of the release package is done only with Synopsys simulator VCS. For issues related to IP model usage with other simulators, please contact Synopsys support.

4 Synthesis, Static Timing Analysis, and Formal Verification

The phy_top design kit contains a set of script and environment setup for Synthesis and Static Timing Analysis located under the “phy_top” component.

4.1 Directory structure

Within your work area, create a directory for synthesizing the IP:

```
% mkdir <IP_dir>/syn
% mkdir <IP_dir>/syn/scr
% cd    <IP_dir>/syn/scr

% cp <IP_dir>/synopsys/ddr4_multiphy<process>/Latest/macro/Latest/example/syn/* .

% mkdir <IP_dir>/constraints
% cd    <IP_dir>/constraints

% cp <IP_dir>/synopsys/ddr4_multiphy<process>/Latest/macro/Latest/constraints/* .

% mkdir <IP_dir>/syn/sta
% cd    <IP_dir>/syn/sta

% cp <IP_dir>/synopsys/ddr4_multiphy<process>/Latest/macro/Latest/example/sta/* .

% mkdir <IP_dir>/syn/fm
% cd    <IP_dir>/syn/fm

% cp <IP_dir>/synopsys/ddr4_multiphy<process>/Latest/macro/Latest/example/formality/* .
```

4.2 Library Setup

Update the "lib_info.customer.txt" file in the constraints directory just created to point to the customer-supplied standard cell library to be used for synthesis. Lines that will need to be updated include:

set std_cell_library_dir ../../std_cells_libs/synopsys # change to local path to standard cell library .lib/.db files

set STD_CELL_VERILOG_LIB ../../std_cells_libs/verilog/generic.v # change to local path to standard cell library verilog model file

set STD_CELL_LIBNAME_WC generic_wc # change to SS/WC liberty file base name

set STD_CELL_LIBNAME_TC generic_wc # change to TT/TC liberty file base name

set STD_CELL_LIBNAME_BC generic_wc # change to FF/BC liberty file base name

set DRIVING_CELL GEN_BUF_12 # change to name of a suitable buffer cell

set WIRELOAD_MIN_NAME ZeroWLM # change to an appropriate wireload model name for the library.

set WIRELOAD_MAX_NAME ZeroWLM # change to an appropriate wireload model name for the library.

Do not add or remove variables from this file.

For at least the first synthesis attempt, the "macro_lib_info.customer.txt" file should be used as supplied. If desired, it is possible to edit the specific PVT corners used for "WC", "TC" and "BC" provided that one is cautious to follow the sytax and variable conventions used in these strings.

In many cases, the set up file DWC_DDRPHY_setup.tcl in the constraints directory just created can be used as supplied (once the lib_info.customer.txt file has been updated with the particulars of the customer-supplied standard cell library for synthesis). In the event that default settings such as HDR/SDR mode, rise-to-fall/rise-to-rise mode, clock frequency, and so on are not suited to the customer's design, please refer to the DWC_DDRPHY_setup.tcl file for additional self-documenting minor variables that may need to be set.

4.3 PHY Configuration

Edit <IP_dir>/synopsys/ddr4_sdram/Latest/phy_top/Latest/rtl/DWC_DDRPHY_define.v to reflect the specific configuration of the customer application."

"This is most easily accomplished using the DWC_DDRPHY_define.v generated by PHY Compiler

4.4 Running Synthesis

Run Synopsys design compiler from *<IP_dir>*/syn/scr directory

```
% cd    <IP_dir>/syn/scr
% dc_shell-t -f DWC_DDRPHY_chip_synth.scr
```

The synthesis results are directed to following directories:

- Verilog gate-level netlist ../netlist/DWC_DDRPHY_chip.vg
- Synthesis reports ../reports/syn/*.rpt

Example of report files generated by synthesis script:

```
DWC_DDRPHY_chip_area_hier.rpt
DWC_DDRPHY_chip_area.rpt
DWC_DDRPHY_chip_cell.rpt
DWC_DDRPHY_chip_chk_netlist.rpt
DWC_DDRPHY_chip_chk_timing.rpt
DWC_DDRPHY_chip_clock.rpt
DWC_DDRPHY_chip_clocks.rpt
DWC_DDRPHY_chip_constraint.rpt
DWC_DDRPHY_chip_constr.rpt
DWC_DDRPHY_chip_fanout_check.rpt
DWC_DDRPHY_chip_feedback_loop.rpt
DWC_DDRPHY_chip_hierarchy.rpt
DWC_DDRPHY_chip_hold_end.rpt
DWC_DDRPHY_chip_hold_full.rpt
DWC_DDRPHY_chip_isolate.rpt
DWC_DDRPHY_chip_link.log
DWC_DDRPHY_chip_path_end.rpt
DWC_DDRPHY_chip_path_full_lb.rpt
DWC_DDRPHY_chip_path_full.rpt
DWC_DDRPHY_chip_port.rpt
DWC_DDRPHY_chip_reference.rpt
```

4.5 Running Static Timing Analysis

Run Synopsys Prime-Time using DWC_DDRPHY_chip_sta.scr

```
% cd    <IP_dir>/syn/sta
% pt_shell -f DWC_DDRPHY_chip_sta.scr
```

The Static Timing analysis will create reports in the following directory: ../reports/sta/*.rpt

Example of report files created by STA script:

```
DWC_DDRPHY_chip_clock.rpt
DWC_DDRPHY_chip_clocks.rpt
DWC_DDRPHY_chip_constr.rpt
DWC_DDRPHY_chip_hold_end.rpt
DWC_DDRPHY_chip_hold_full.rpt
DWC_DDRPHY_chip_lb_timing.rpt
DWC_DDRPHY_chip_path_end.rpt
DWC_DDRPHY_chip_path_full.rpt
DWC_DDRPHY_ctl_a_timing.rpt
DWC_DDRPHY_ctl_d_timing.rpt
DWC_DDRPHY_do_a_timing.rpt
DWC_DDRPHY_do_i_timing.rpt
DWC_DDRPHY_d_timing.rpt
DWC_DDRPHY_phy_a_timing.rpt
DWC_DDRPHY_phy_q_timing.rpt
DWC_DDRPHY_q_timing.rpt
```

4.6 Running Formal Verification

Run Synopsys Formality using DWC_DDRPHY_rtl_vs_gate.tcl

```
% cd    <IP_dir>/syn/fm
% fm_shell -f DWC_DDRPHY_rtl_vs_gate.tcl
```

The Static Timing analysis will create a report in the following directory: ../reports/syn/*.rpt

Example of report files created by Formality script:

DWC_DDRPHY_chip_fm_failing_points.rpt