SYNOPSYS®

DesignWare® Cores LPDDR5/4/4X PHY Initialization (PHYInit) Software Overview

Application Note

DWC LPDDR5/4/4X PHY

Version 1.00
December 12, 2019

Copyright Notice and Proprietary Information

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 https://www.synopsys.com/company/legal/trademarks-brands.html

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

Free and Open-Source Software Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.
www.synopsys.com

Contents

Revision History

Note Links and references to section, table, figure, and page numbers in this table are only assured to be valid for the version in which the change is made.
This application note supports current and previous PHYInit version.

Date	Document Version	Description
December 12, 2019	1.00	Generic Document Initial release

1 Introduction

1.1 What is PHYInit?

PhyInit is a program to generate txt files corresponding to sequence of registers writes/reads required to initialize the PHY for mission mode. The output txt file is generated based on user programing of PhyInit inputs.

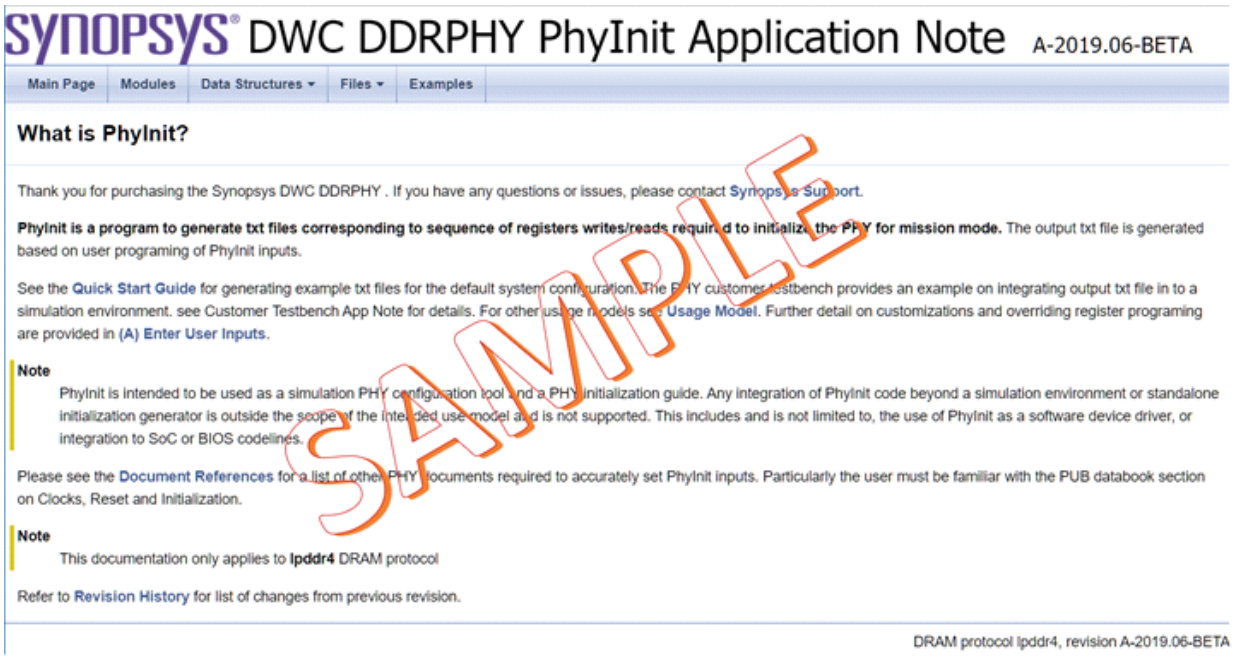
A quick start guide and a README is provided in PhyInit folder in order to speed up initial integration. Final sequence can be obtained only when all PhyInit configuration is fully considered. This document only covers a high level overview of the initialization process.

The PhyInit software and corresponding PhyInit Aplication Note (HTML version) are available in the following path in the deliverables directory:

```
<PHY Install
Directory>/synopsys/<technology>_<process>/Latest/phyinit/Latest/software/<protocol>/doc/
```

For ease of use, a Microsoft Compiled HTML (.CHM) as well as HTML tar file of the documentation is provided. This document will provide an introductory overview of PhyInit with references to HTML documentation. Figure 1-1 on page 8 shows the opening page of HTML documentation.

Figure 1-1 PhyInit HTML Documentation



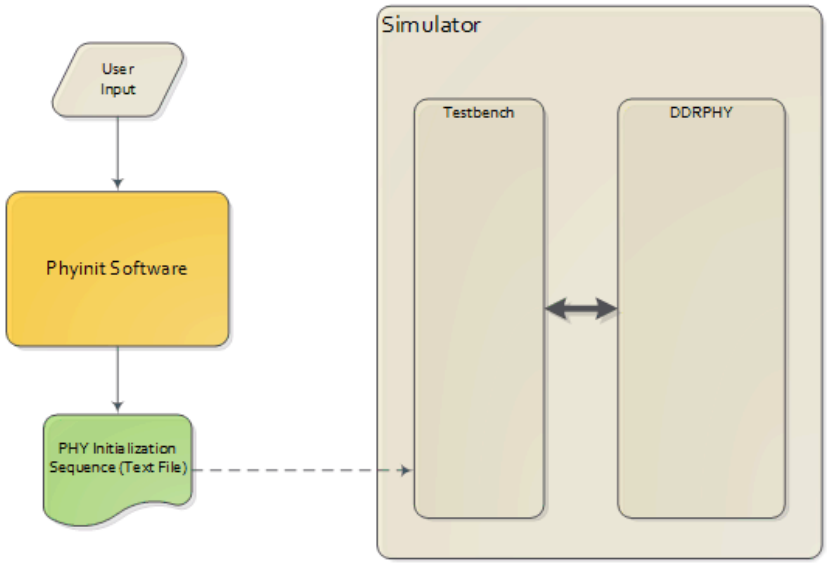
Note PhyInit is intended to be used as a simulation PHY configuration tool and a PHY initialization guide. Any integration of PhyInit code beyond a simulation environment or standalone initialization generator is outside the scope of the intended use model and is not supported. This includes and is not limited to, the use of PhyInit as a software device driver, or integration to SoC or BIOS codelines.

1.2 Purpose

The purpose of the PHY initialization software (PhyInit) is to allow the user to generate a customized PHY initialization sequence specific to the user's system configuration. The user will provide a set of configuration inputs to the program and it will output a PHY initialization sequence text file specific to the inputs provided. The user can choose to execute or skip the training firmware in the initialization sequence.

The PHYInit software is written in C and outputs a text file for each DRAM protocol specifying the exact registers that need to be set at each of the PHY initialization steps describes in the section “Phase Bits and Chip Selects” of the *PUB Databook*. Figure 1-2 shows the role of PHYInit software in a simulation of the PHY.

Figure 1-2 Overview



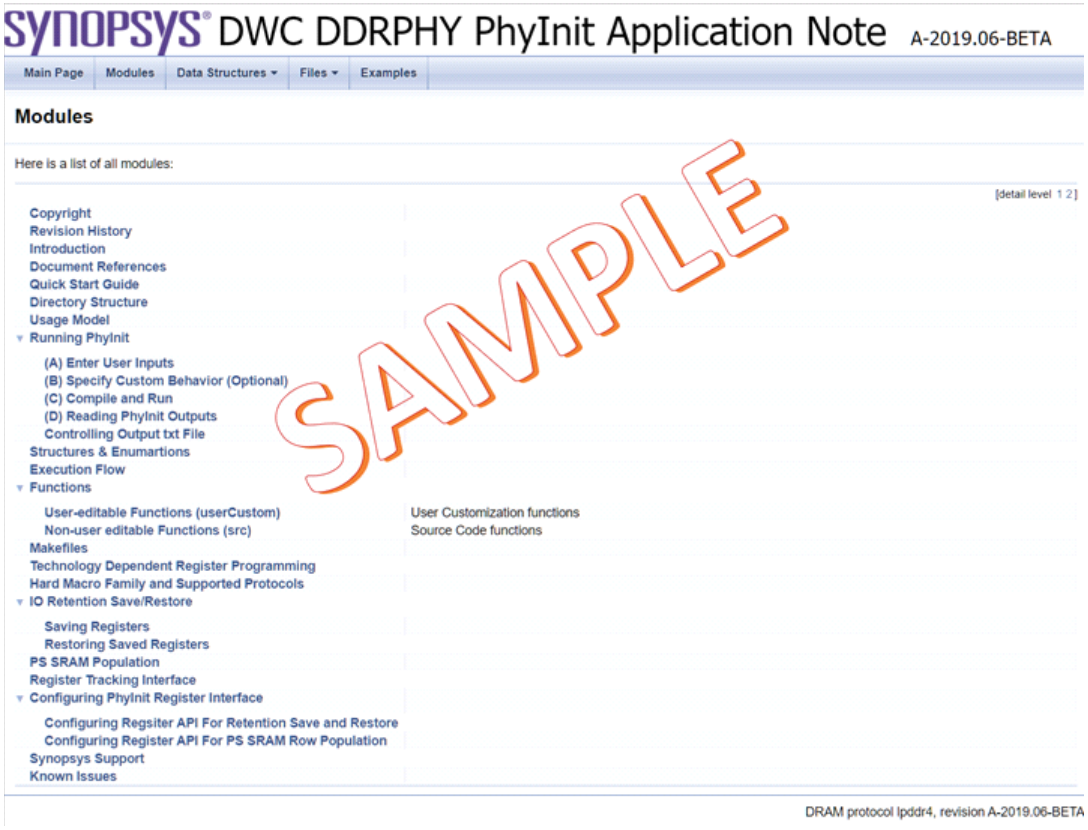
The following sections of this document will describe organization and use of the PhyInit software.

2 About This Document

This document provides only an overview of topics on PhyInit. For detailed documentation, refer to the “Modules” page in the HTML document pictured below. Specific details for each DRAM protocol differ. A dedicated version of HTML documentation for each DRAM protocol is provided in the release package.

Note The image captures of the PhyInit HTML documentation in this document are provided for reference and can be out of date. Always refer the HTML documentation directly for accurate up-to-date documentation.

Figure 2-1 Modules page in HTML Documentation



3 Document References

Following documents are referenced in the application note.

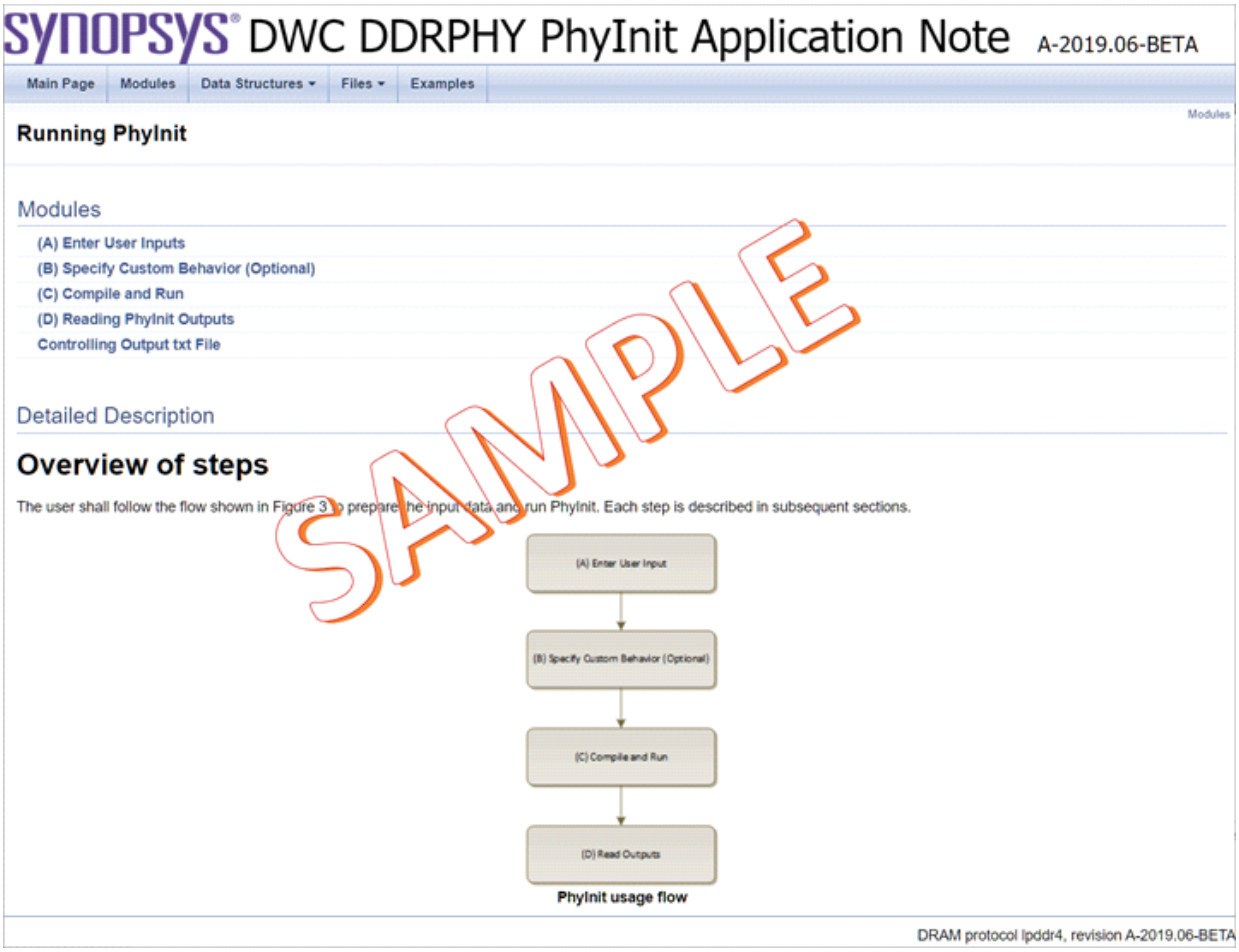
Table 3-1 PHY Documents Required to Set PHYInit Inputs

Document Reference	Document Name
Training firmware Application Note	DesignWare® Cores Firmware Training Application Notes
PUB Databook	DesignWare® Cores PHY Utility Block (PUB) Databook
PHY Databook	DesignWare® Cores PHY Databook
DQ HSPICE Application Note	DesignWare® Cores TXRXDQ Slice HSPICE Driver and Receiver Model for your process technology
CA HSPICE Application Note	DesignWare® Cores TXRXCA Slice HSPICE Driver and Receiver Model for your process technology
Customer Testbench App Note	DesignWare® Cores DDRn PHY Customer Testbench Application Note

4 Running PHYInit

The user shall follow the flow shown in Figure 4-1 to prepare the input data and run PhyInit. Each step is described further in details in the HTML documentation under the section “Running PhyInit”. For details of each step from A through D refer to HTML documentation.

Figure 4-1 Running PhyInit Page



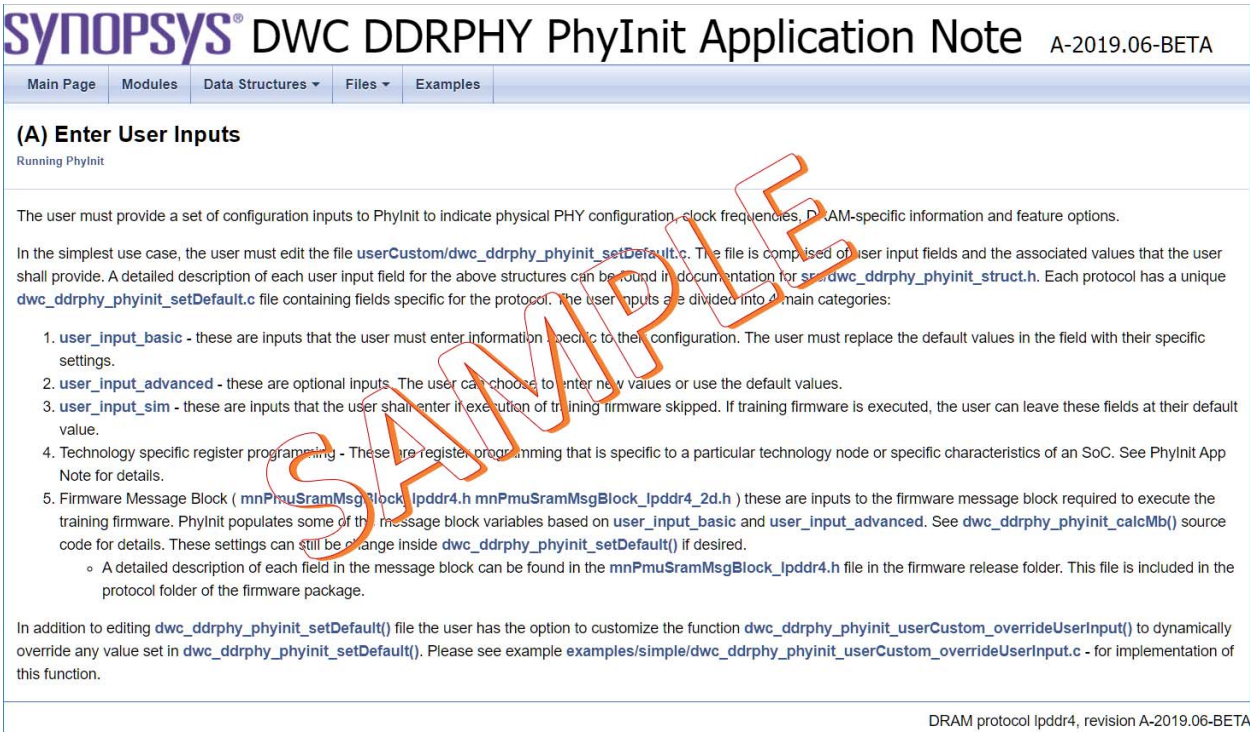
4.1 (A) Enter User Inputs

There are 5 keys inputs that need to be specified:

1. user_input_basic
2. user_input_advanced
3. user_input_sim (simulation only)
4. Technology specific register programming
5. Firmware Message Block

These inputs are discussed in the details in various sections in the HTML documentation. The user must obtain the required data to program and configure them for their specific application. More detail guidance is provided in the HTML documentation. A capture of the specific page in HTML documentation is provided below for reference.

Figure 4-2 Enter User Inputs Page



4.2 (B) Specify Custom Behavior (Optional)

Behavior of PhyInit can be overridden using the userCustom() functions described in the documentation. These overrides can be either before or after training has been completed:

- **dwc_ddrphy_phyinit_userCustom_customPreTrain():** This function is called before training firmware is executed. Any register override in this function might affect the firmware training results. Technology specific register programming can be done here.
- **dwc_ddrphy_phyinit_userCustom_customPreTrainPsLoop():** This function is called for each PState, before training firmware is executed. Any per-PState register override in this function might affect the firmware training results.
- **dwc_ddrphy_phyinit_userCustom_customPostTrain():** This function is called after the execution of training firmware is complete. The user can use this function to override any CSR value programmed by PhyInit or training firmware. For further examples see related document section below.
- **dwc_ddrphy_phyinit_userCustom_customPreTrainPsLoop():** This function is called for each PState, after training firmware is executed. Any per-PState register override meant to change a CSR value programmed by PhyInit or training firmware can be done at this moment.

For details and examples see HTML documentation section.

4.3 (C) Compile and Run

A makefile is provided to compile PhyInit code and execute the object to generate the initialization sequence. Further instructions are provided in the pages referred in the HTML documentation in the release package.

Figure 4-3 Compile and Run Page

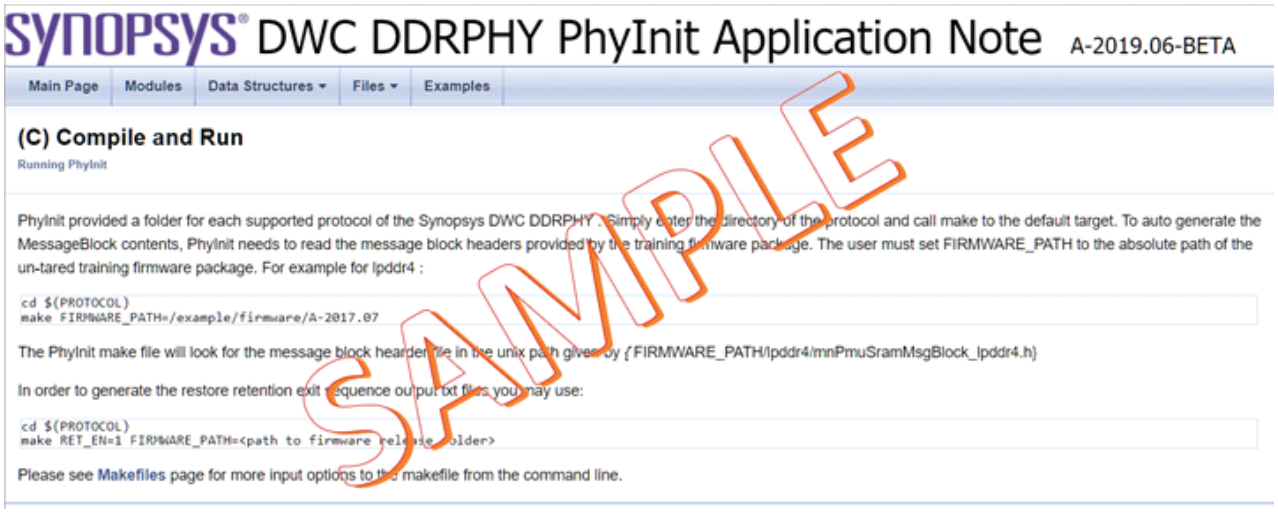
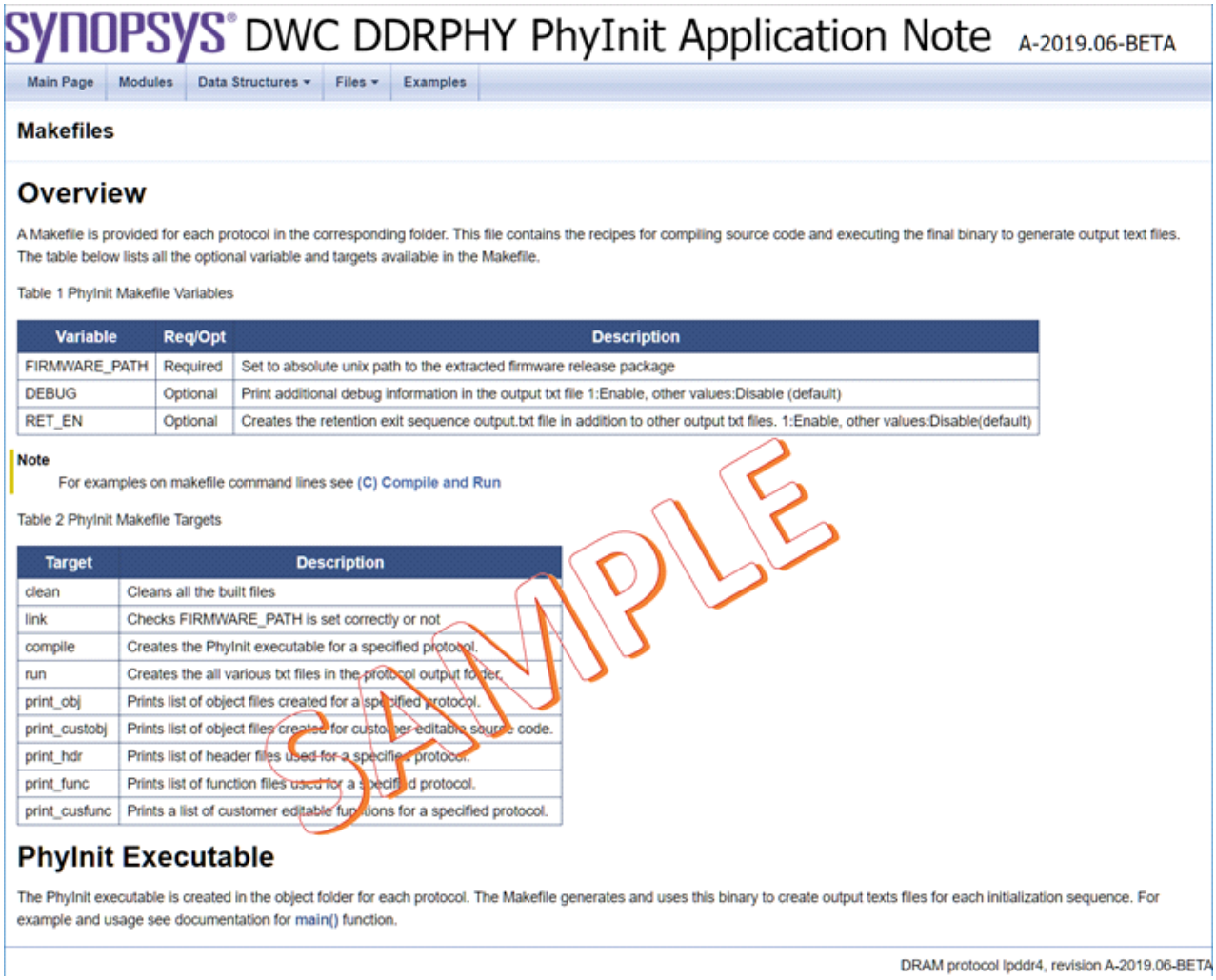


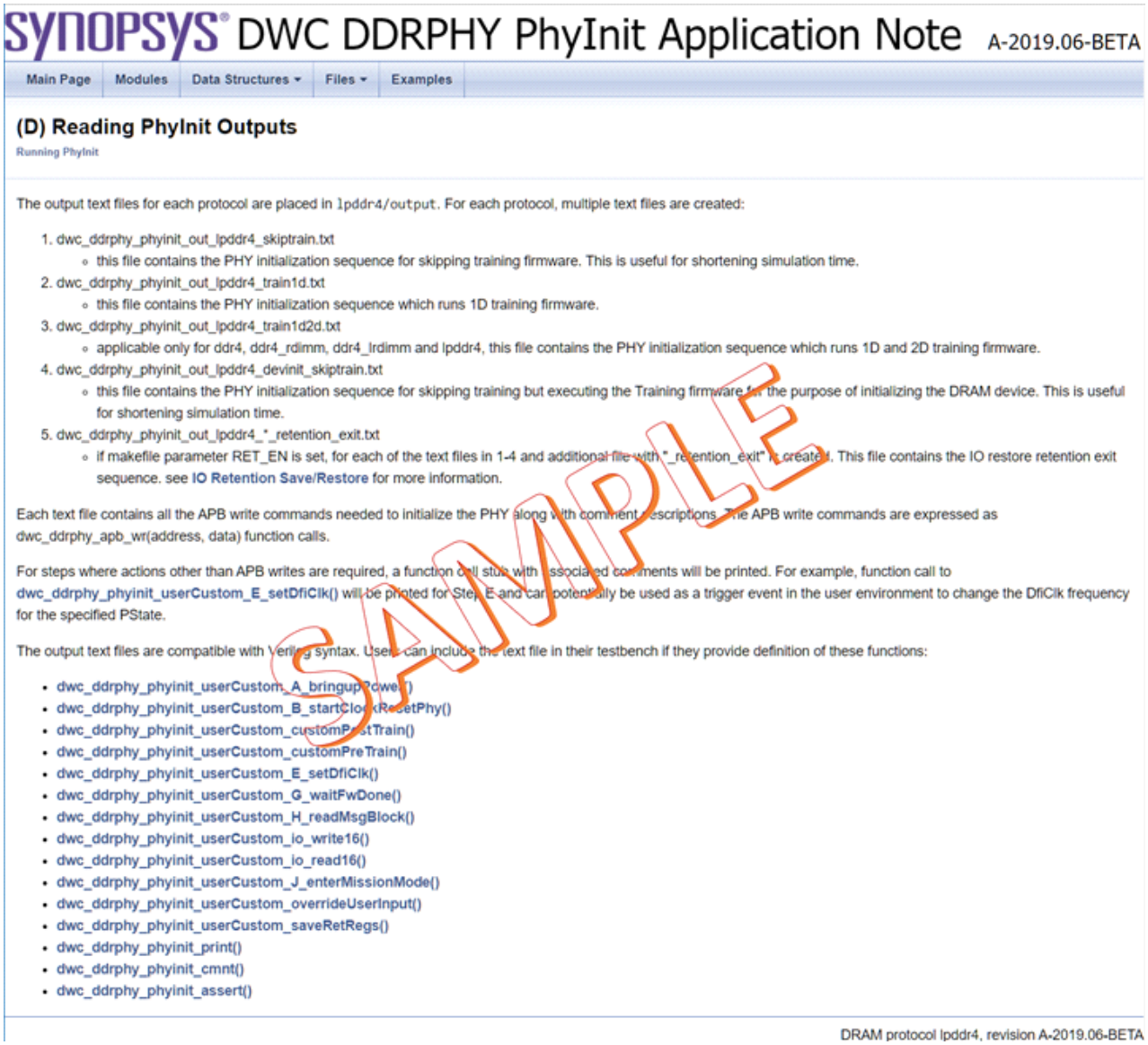
Figure 4-4 PhyInit Makefile Documentation Page



4.4 (D) Reading PhyInit Outputs

The output text files for each protocol are placed in an output folder. Depending on the Makefile parameters, a number of output sequences are generated. Details on the usage and intention is provided further in the HTML documentation as shown below.

Figure 4-5 Reading Outputs Page



SYNOPSYS® DWC DDRPHY PhyInit Application Note A-2019.06-BETA

Main Page | Modules | Data Structures | Files | Examples

(D) Reading PhyInit Outputs

Running PhyInit

The output text files for each protocol are placed in `lpddr4/output`. For each protocol, multiple text files are created:

1. dwc_ddrphy_phyinit_out_lpddr4_skiptrain.txt
   - this file contains the PHY initialization sequence for skipping training firmware. This is useful for shortening simulation time.
2. dwc_ddrphy_phyinit_out_lpddr4_train1d.txt
   - this file contains the PHY initialization sequence which runs 1D training firmware.
3. dwc_ddrphy_phyinit_out_lpddr4_train1d2d.txt
   - applicable only for ddr4, ddr4_rdimm, ddr4_lrdimm and lpddr4, this file contains the PHY initialization sequence which runs 1D and 2D training firmware.
4. dwc_ddrphy_phyinit_out_lpddr4_devinit_skiptrain.txt
   - this file contains the PHY initialization sequence for skipping training but executing the Training firmware for the purpose of initializing the DRAM device. This is useful for shortening simulation time.
5. dwc_ddrphy_phyinit_out_lpddr4_*_retention_exit.txt
   - if makefile parameter RET_EN is set, for each of the text files in 1-4 and additional file with "_retention_exit" is created. This file contains the IO restore retention exit sequence. see IO Retention Save/Restore for more information.

Each text file contains all the APB write commands needed to initialize the PHY along with comment descriptions. The APB write commands are expressed as dwc_ddrphy_apb_wr(address, data) function calls.

For steps where actions other than APB writes are required, a function call stub with associated comments will be printed. For example, function call to dwc_ddrphy_phyinit_userCustom_E_setDfiClk() will be printed for Step E and can potentially be used as a trigger event in the user environment to change the DfiClk frequency for the specified PState.

The output text files are compatible with Verilog syntax. Users can include the text file in their testbench if they provide definition of these functions:

- dwc_ddrphy_phyinit_userCustom_A_bringupPowered()
- dwc_ddrphy_phyinit_userCustom_B_startClockResetPhy()
- dwc_ddrphy_phyinit_userCustom_customPostTrain()
- dwc_ddrphy_phyinit_userCustom_customPreTrain()
- dwc_ddrphy_phyinit_userCustom_E_setDfiClk()
- dwc_ddrphy_phyinit_userCustom_G_waitFwDone()
- dwc_ddrphy_phyinit_userCustom_H_readMsgBlock()
- dwc_ddrphy_phyinit_userCustom_io_write16()
- dwc_ddrphy_phyinit_userCustom_io_read16()
- dwc_ddrphy_phyinit_userCustom_J_enterMissionMode()
- dwc_ddrphy_phyinit_userCustom_overrideUserInput()
- dwc_ddrphy_phyinit_userCustom_saveRetRegs()
- dwc_ddrphy_phyinit_print()
- dwc_ddrphy_phyinit_cmnt()
- dwc_ddrphy_phyinit_assert()

DRAM protocol lpddr4, revision A-2019.06-BETA

5 Synopsys Support

5.1 PhyInit Support and Debugging

A default configuration is provided to help run PHYInit for the first time and generate the desired txt file.

While PHYInit and the training firmware are tested in our environment, errors may occur due to unforeseen configurations and conditions. If you are experiencing issues, you may wish to contact Customer Support. The quickest and most efficient way is to open a SolvNet ticket, however, you may also email support_center@synopsys.com.

When requesting support, you will need to provide the following logs:

- PhyInit output.txt
- PhyInit release revision
- Training Firmware release revision
- Training Firmware log