DesignWare® Cores PHY – External ROM and External SRAM Interfaces and Firmware Loading

Application Note

Version 1.20
January 2020

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

Date	Version	Description
January 2020	1.20	The changes since the previous version include the following: Updated: ■ Links to https://solvnetplus.synopsys.com throughout the book ■ Links to web resources in "Preface" ■ "Customer Support"
December 2019	1.10	The changes since the previous version include the following: Updated: ■ A.4 ■ A.10 Added: ■ Note in A.5
July 2019	1.00	Initial version

Preface

This document describes the firmware feature.

⚠ **Attention** The PHY databook contains the latest information. Therefore, for all late-breaking changes, refer to the PHY databook.

Application Note Organization

This document is organized as follows:

- Chapter 1, "ROM Interface, SRAM Interface, and Firmware Loading FAQs", provides information about firmware, and how to load and use the latest version of the firmware.

Web Resources

- DesignWare IP product information: https://www.synopsys.com/designware-ip.html
- Your custom DesignWare IP page: https://www.synopsys.com/dw/mydesignware.php
- Documentation through SolvNet: https://solvnetplus.synopsys.com (Synopsys password required)
- Synopsys Common Licensing (SCL): https://www.synopsys.com/keys

Customer Support

To obtain support for your product, contact Support Center using one of the following methods:

- *For the fastest response*, enter a case through SolvNetPlus:

  a. https://solvnetplus.synopsys.com

  b. Click the **Cases** menu and then click **Create a New Case** (below the list of cases).

  c. Complete the mandatory fields that are marked with an asterisk and click **Save**.

  For more information about general usage information, refer to the following arcticle in SolvNetPlus:

  https://solvnetplus.synopsys.com/s/article/SolvNetPlus-Usage-Help-Resources

- Or, send an e-mail message to support_center@synopsys.com (your email will be queued and then, on a first-come, first-served basis, manually routed to the correct support engineer):
  - Include the Product L1 and Product L2 names, and Version number in your e-mail so it can be routed correctly.

  - For simulation issues, include the timestamp of any signals or locations in waveforms that are not understood
  - Attach any debug files you created in the previous step.
- Or, telephone your local support center:
  - North America:
    Call 1-800-245-8005 from 7 AM to 5:30 PM Pacific time, Monday through Friday.
  - All other countries:
    https://www.synopsys.com/support/global-support-centers.html

1 ROM Interface, SRAM Interface, and Firmware Loading FAQs

This chapter includes the frequently asked questions (FAQs) about the Synopsys SerDes PHY external ROM interface, external SRAM interface, and firmware loading:

- "Q.1: What is a Firmware and What is the Purpose of the Firmware?"
- "Q.2: Is it Needed to Write Your Own Firmware for PHY Operation or Calibration?"
- "Q.3: Where is the Firmware Located in the PHY package and What is the Format of the Firmware?"
- "Q.4: If an Updated Firmware is Available, Should I Upgrade to the Newer Version? How Do I Load the Updated Firmware?"
- "Q.5: What are the Interfaces Available for Loading the Firmware into SRAM?"
- "Q.6: What is the Time Taken to Load the Firmware into SRAM?"
- "Q.7: What is the Size of the SRAM that is Required?"
- "Q.8: What is the SRAM Address Space?"
- "Q.9: If a Firmware Update in SRAM is Not Required, can the PHY sram_bypass input be used to bypass the SRAM interface, where the Adaptation and Calibration Algorithms are Executed from the Hard Wired Values (in LUT) within the Raw PCS?"
- "Q.10: Does PHY Implement Parity or ECC Protection on SRAM Configuration?"

Q.1 What is a Firmware and What is the Purpose of the Firmware?

A.1 Firmware is the instruction code that helps to execute the calibration and adaptation algorithms, and comes packaged with the PHY. Raw PCS stores this pre-optimized and pre-validated code in the form of hard-coded lookup table (LUT).

When synthesis is performed, synthesis scripts provided by Synopsys synthesize the hard-coded lookup table (LUT) with the firmware that comes with the PHY package.

To have the flexibility of loading the updated firmware in silicon, Synopsys PHY comes with an external SRAM interface. For more information about this interface, refer to the “External SRAM Support” section in the PHY databook.

Note Firmware is protocol-independent.

Q.2 Is it Needed to Write Your Own Firmware for PHY Operation or Calibration?

A.2 Synopsys supplied Firmware includes all the necessary “calibration and adaptation algorithms.” It is not expected to create, interpret, or modify the firmware.

Q.3 Where is the Firmware Located in the PHY package and What is the Format of the Firmware?

A.3 Firmware location: `dwc_*_phy_<process_technology>\<phy_version>\phy\include\`

Firmware filename: `*_pcs_raw_mem_rst.v`

As an example, in the Firmware file *_pcs_raw_mem_rst.v, SRAM firmware content starts from below line:

```
//START RAM CONTENT

//%%CREG RAM_CMN0_B0_R0 - Common mem #0, Bank #0, Reg #0
// OFFSET: 0x8000
// ATTR: READ,WRITE
// FIELDS:
// [15:0] DATA - Memory data
// RESET: 0x081D

//%%CREG RAM_CMN0_B0_R1 - Common mem #0, Bank #0, Reg #1
// OFFSET: 0x8001

// ATTR: READ,WRITE
// FIELDS:
// [15:0] DATA - Memory data
// RESET: 0xFFFF

//%%CREG RAM_CMN0_B0_R2 - Common mem #0, Bank #0, Reg #2
// OFFSET: 0x8002
// ATTR: READ,WRITE
// FIELDS:
// [15:0] DATA - Memory data
// RESET: 0x33AF
....
```

There is another format in which the firmware is supplied in the package. This format is intended for more readability and is not compiled into the RTL code. This format shows the content of the SRAM address locations as mapped into the PHY address space. If you want to access the SRAM directly, that is, not using PHY CREG interface, the SRAM addresses start from 0x0000.

```
//=====================
// ADDR            DATA
//=====================
0x6000  0x0207

0x6001  0x0001

0x6002  0x080E

0x6003  0xFFFF

...
```

Note Some PHY packages contain `*.fw` files, which is readily available in the format shown previously.

Q.4 If an Updated Firmware is Available, Should I Upgrade to the Newer Version? How Do I Load the Updated Firmware?

A.4 Synopsys periodically supplies updated Firmware code in the form of `/phy/include/*_pcs_raw_mem_rst.v` file, `/phy/include/* mem_sram_cr_para.fw` file (which is the External SRAM firmware file), or `/phy/include/* mem_sram_cr_para_fastsim.fw` file (which is the SRAM firmware file for fast-simulation mode file).

Note Synopsys recommends to use the newer firmware when available.

Loading Firmware (For Example, Version A) that Comes Packaged with the PHY

- During pre-silicon stage, performing synthesis with firmware (for example, Version A) uses the firmware code to be part of internal look up table (LUT).
- During PHY operation, after de-assertion of the phy_reset input signal, the boot loader in the Raw PCS loads the firmware code from the internal look up table (LUT) to the external SRAM and asserts the output signal, sram_init_done, indicating that SRAM initialization is done.
- After initialization of SRAM, you can change the SRAM contents (or access any PHY register) using either the JTAG or the CR Parallel Interface by addressing the SRAM address space in the register map. After external access to the SRAM (or any other PHY register) is complete, input sram_ext_ld_done (SRAM External Load Done) is set high, allowing the FSMs in the Raw PCS to start executing the code from SRAM.

Note If no update to SRAM content is required, assert sram_ext_ld_done soon after sram_init_done is asserted. There is no timing delay required between assertion of sram_init_done and assertion of sram_ext_ld_done.

Loading Newer Firmware (For Example, Version B) that You have Received from Synopsys at a Later Stage

- If your design is in pre-silicon stage, performing re-synthesis with newer firmware (for example, Version B) uses the firmware code to be part of internal look up table (LUT).
- During PHY operation, after de-assertion of the phy_reset input signal, the boot loader in the Raw PCS loads the newer firmware code from the internal look up table to the external SRAM and asserts output signal, sram_init_done (SRAM Initialization done, an output signal).
- After silicon is taped-out, to load the newer firmware (for example, Version B) supplied by Synopsys, load the firmware after the PHY output signal, sram_init_done, is asserted. After loading of the newer firmware is complete, set the PHY input signal, sram_ext_ld_done, to high— allowing the FSMs in the Raw PCS to start executing the code from SRAM.

Note Even if the updated firmware (for example, Version B) is not available, you can still choose to load the older firmware (Version A) on to the SRAM after LUT content (Version A) is loaded into SRAM by the boot-loader and sram_init_done is asserted. However, this exercise comes at the cost of additional SRAM loading time. After the loading of the SRAM is done, set the PHY input signal, sram_ext_ld_done, to high—allowing the FSMs in the Raw PCS to start executing the code (Version A) from SRAM.

Q.5 What are the Interfaces Available for Loading the Firmware into SRAM?

A.5 After the initialization of SRAM (that is, after the PHY output signal, sram_init_done, is asserted), use either the JTAG or CR parallel interface to change the SRAM contents or access any PHY register by addressing the SRAM address space in the register map. You can also implement an external SRAM interface and multiplex with the SRAM interface from the PHY. This can be beneficial in reducing the SRAM load time if user interface is significantly faster than the PHY CR para interface. Select such external interface only after sram_init_done is asserted and before sram_ext_ld_done is asserted.

Note If SRAM ECC is enabled, use mem_sram_sie_load_ecc.bin for previously mentioned side loading SRAM with MUX.

Q.6 What is the Time Taken to Load the Firmware into SRAM?

A.6 If you use cr_para_clk (Control Register Clock) at 100 MHz, it takes approximately 0.5 ms to load the firmware into SRAM after the sram_init_done signal is asserted.

Q.7 What is the Size of the SRAM that is Required?

A.7 Size of the SRAM that is required varies from one PHY to another. For more information about the SRAM size requirement, refer to the “External SRAM Support” section of the PHY databook.

Q.8 What is the SRAM Address Space?

A.8 For more information about the SRAM address space, refer to the “External SRAM Support” section of the PHY databook.

Q.9 If a Firmware Update in SRAM is Not Required, can the PHY sram_bypass input be used to bypass the SRAM interface, where the Adaptation and Calibration Algorithms are Executed from the Hard Wired Values (in LUT) within the Raw PCS?

A.9 When the PHY sram_bypass input signal is asserted, the adaptation and calibration algorithms (firmware) are executed from the hard wired values within the Raw PCS.

Note

- PHY input signal sram_bypass signal is meant to be used only for debugging purposes and must not change after phy_reset is negated.
- Even if there is no updated firmware available, Synopsys still recommends to use the SRAM interface so that the firmware is first loaded by Raw PCS into the SRAM at which point you can assert sram_ext_ld_done soon after sram_init_done is asserted.

Q.10 Does PHY Implement Parity or ECC Protection on SRAM Configuration?

A.10 ECC is implemented at Synopsys SRAM interface inside RAW PCS. It can be enabled by asserting sram_ecc_en before phy_reset deasserted.

For more information about the SRAM address space, refer to the "External SRAM Support" section of the PHY databook.