SYNOPSYS®

DesignWare® Cores LPDDR54 PHY Quickboot Firmware

Application Note

DWC LPDDR54 PHY
Firmware Version: C-2021-10

Firmware Version: C-2021-10
Document Version: 1.70a
October 13, 2021

Copyright Notice and Proprietary Information

© 2021 Synopsys, Inc. All rights reserved. This Synopsys software and all associated documentation are proprietary to Synopsys, Inc. and may only be used pursuant to the terms and conditions of a written license agreement with Synopsys, Inc. All other use, reproduction, modification, or distribution of the Synopsys software or the associated documentation is strictly prohibited.

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

The following tables lists the revision history of the PHY from release to release. Refer to the PHY release notes for a detailed description of PHY component updates.

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
- In some instances, documentation-only updates occur. The DesignWare IP product information (http://www.designware.com) has the latest documentation

Document Version	Date	Description
1.70a	October 13, 2021	Updated: -Structure reformatting Added: - Chapter "Preface" on page 7
1.60a	August 20, 2021	Updated: - Section 5.1.2 MemReset Toggle
1.50a	June 16, 2021	Updated: - Date and Version - Section 5.1.2 MemReset Toggle
1.40a	April 19, 2021	Updated: - Section 4.3.2 Step (I-2) Save PHY registers and ACSM SRAM - Section 5.1 QuickBoot flow
1.30a	December 15, 2020	Added: -Table 2 Required QuickBoot Message Block Settings for LPDDR4 on page 17 -Table 3 Required QuickBoot Message Block Settings for LPDDR5 on page 18
1.20a	July 29, 2020	Minor CSR update

Document Version	Date	Description
1.10a	May 29, 2020	Minor updates
1.00a	April 14, 2020	Initial release

Preface

This Application Note describes the LPDDR54 PHY Quickboot Firmware.

Recommended Reading

The following documentation provides essential information about the IP to create a complete solution:

- Designware Cores LPDDR5/4/4X PHY Implementation Guide
- DesignWare Cores LPDDR5/4/4X PHY Databook

Web Resources

- DesignWare IP product information: http://www.designware.com
- Your custom DesignWare IP page: http://www.mydesignware.com
- Documentation through SolvNetPlus: http://solvnetplus.synopsys.com (Synopsys password required)
- Synopsys Common Licensing (SCL): http://www.synopsys.com/keys

Synopsys Statement on Inclusivity and Diversity

Synopsys is committed to creating an inclusive environment where every employee, customer, and partner feels welcomed. We are reviewing and removing exclusionary language from our products and supporting customer-facing collateral. Our effort also includes internal initiatives to remove biased language from our engineering and working environment, including terms that are embedded in our software and IPs. At the same time, we are working to ensure that our web content and software applications are usable to people of varying abilities. You may still find examples of non-inclusive language in our software or documentation as our IPs implement industry-standard specifications that are currently under review to remove exclusionary language.

Customer Support

To obtain support for your product:

- First, prepare the following debug information, if applicable:
- For environment setup problems or failures with configuration, simulation, or synthesis that occur within coreConsultant or coreAssembler, use the following menu entry:

File > Build Debug Tar-file

Check all the boxes in the dialog box that apply to your issue. This menu entry gathers all the Synopsys product data needed to begin debugging an issue and writes it to the file <core tool startup directory>/debug.tar.gz.

- For simulation issues outside of coreConsultant or coreAssembler:
  - Create a waveforms file (such as VPD or VCD)
  - Identify the hierarchy path to the DesignWare instance
  - Identify the timestamp of any signals or locations in the waveforms that are not understood
- Then, contact Support Center, with a description of your question and supplying the above information, using one of the following methods:
  - For fastest response, use the SolvNet website. If you fill in your information as explained below, your issue is automatically routed to a support engineer who is experienced with your product. The Sub Product entry is critical for correct routing.

    Go to http://solvnet.synopsys.com/EnterACall and click on the link to enter a call. Provide the requested information, including:
- Product: Designware Cores
- Sub Product: LPDDR54_PHY
- Tool Version:
- Problem Type:
- Priority:
- Title: Provide a brief summary of the issue or list the error message you have encountered
- Description: For simulation issues, include the timestamp of any signals or locations in waveforms that are not understood

After creating the case, attach any debug files you created in the previous step.

- 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 name, Sub Product name, and Tool Version number in your e-mail (as identified above) 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:
  http://www.synopsys.com/Support/GlobalSupportCenters

1 Introduction

The following sections are included in this chapter:

1.1 Purpose

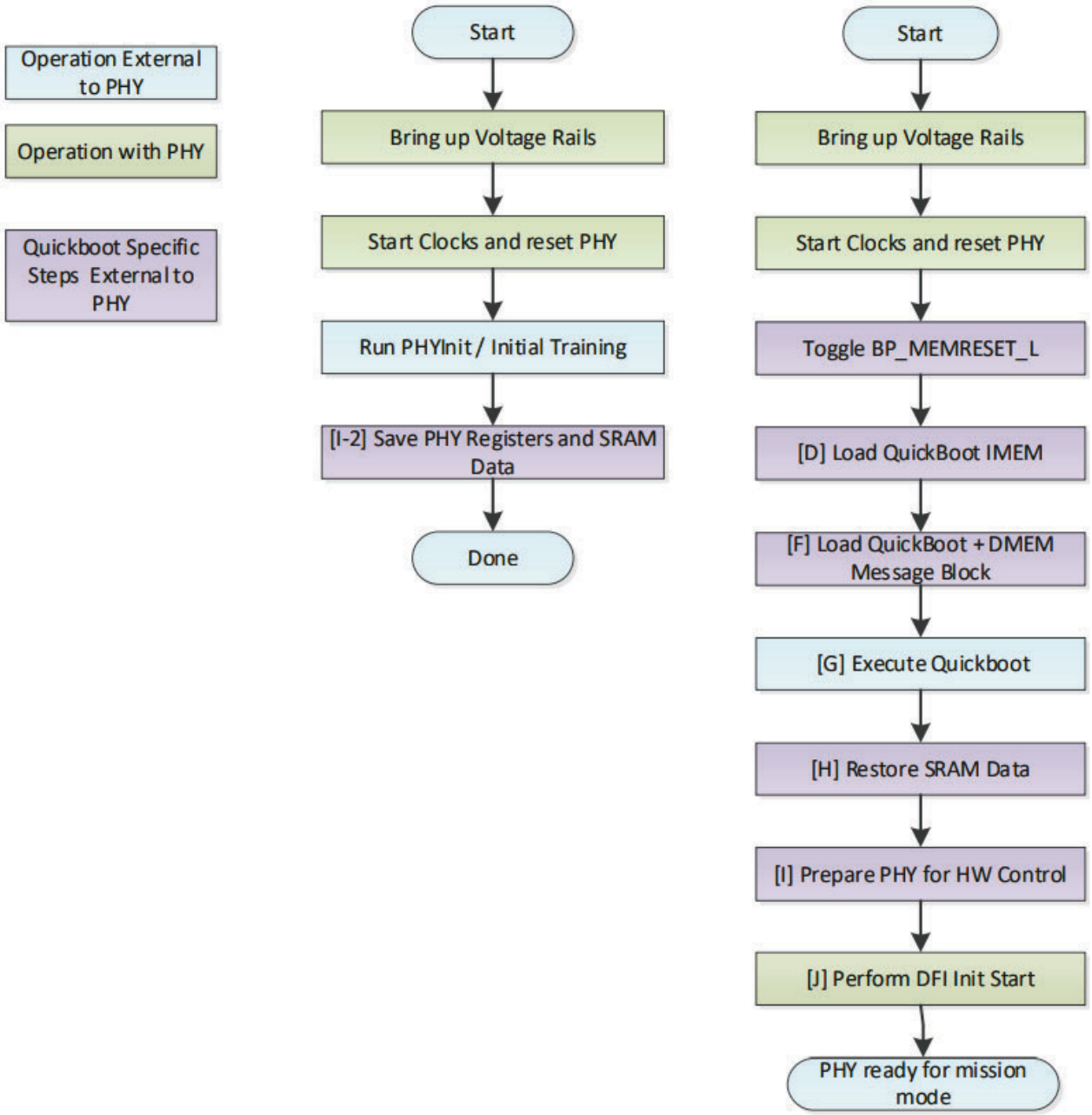
The DWC DDRPHY QuickBoot Firmware is designed to restore saved trained settings to allow the fastest boot time possible for LPDDR4 and LPDDR5. It restores PHY training registers and performs DRAM initialization.

Note that this specification document is preliminary and subject to change.

1.2 Overview

QuickBoot uses previously saved information from an initial boot to quickly restore the PHY state and initialize the DRAM. The initial training is done once in a stage called “First Boot” where the message block outputs and CSRs are saved. QuickBoot uses this information to initialize the PHY and the DRAM. The PLL, DLLs, and drivers will be re-calibrated, and the PIE will execute VT-drift compensation for initial Write, Read, and Read gate training results.

Figure 1-1 First Boot and QuickBoot comparison 1



2

First Boot Requirements

The following sections are included in this chapter:

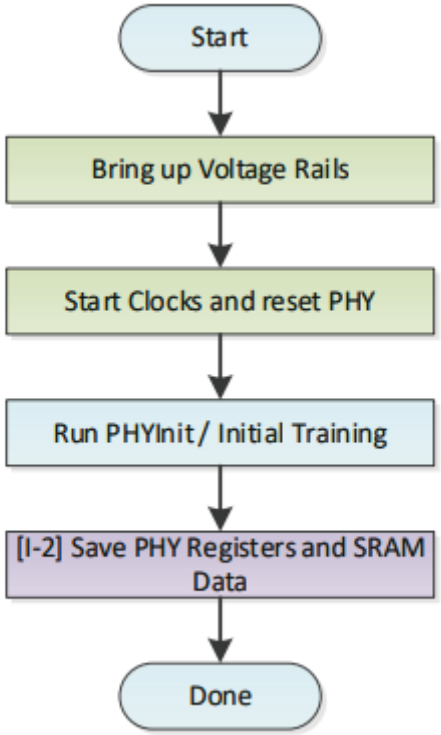
2.1 Initial Training

The setup and configuration of the Initial Training is the same as normal 1D and 2D training as described in the Firmware Application Note. If more than one Pstate is required, then the training needs to be completed for each Pstate.

2.2 First Boot Register Save

Directly after PHYInit step I (Load PHYInit Engine image), a list of required CSRs needs to be read and saved in nonvolatile storage. These values will be written to the PHY microcontroller DMEM in a special area that will allow the PHY to read them when needed. The list of registers is contained in a file included with the QuickBoot distribution. The configuration file depends on the number of configured DBYTES. For example, a 4 Dbyte configuration the file is called REG_LIST_DBYTE4.txt. It is important to note that the order of the values is important. The order needs to be preserved when they are written to the DMEM area during QuickBoot.

Figure 2-1 First Boot Register Save Diagram



2.3 First Boot Flow

The First Boot flow is shown in Figure 2-1 on page 19. Important steps are described in some detail in the following sections. First Boot is where the standard training algorithms are run in order to get the CSR settings needed for proper mission mode operation. After running training, the results need to be saved.

2.3.1 PHYInit / Initial Training

Training should be run for all required PStates and training stages as described in the PHYInit and Firmware application notes.

2.3.2 Step (I-2) Save PHY registers and ACSM SRAM

PHY registers listed in REG_LIST must be saved in nonvolatile memory to be used when initializing the QuickBoot data memory. See section 3.1.6 on page 27 for the details on how this data needs to be saved so that it may be restored properly.

At this point, the PHY is not configured for APB reads since the PHY Microcontroller and Training Hardware are clock gated and the APB Mux select does not enable APB reads. The following steps must be taken to fully enable APB reads for "PHY Register SAVE":

The procedure should be as follows:

1. Complete all standard initial training and finish step I
2. Issue the following APB writes to allow the host to access the PHY CSRs:
   - apb_wr(32'h000D0000, 16'h0000); //DWC_DDRPHYA_APBONLY0__MicroContMuxSel
   - apb_wr(32'h000C0080, 16'h0003); //DWC_DDRPHYA_DRTUB0_UcclkHclkEnables
3. Read CSRs in the register list and save them to non-volatile storage to be used for QuickBoot initialization.
4. Read ACSM SRAM area and save it to non-volatile storage.Psuedo code provided to save ACSM SRAM as follows:

   for (int sram_addr=0; sram_addr<1024; sram_addr++) begin for (int half_word=0; half_word<4; half_word++) begin apb_read16((ACSM_SRAM_BASE_ADDR + (sram_addr*4) + half_word), data); quickboot_acsm_sram[sram_addr] = quickboot_acsm_sram[sram_addr] | (data<<(16*half_word)); end end where ACSM_SRAM_BASE_ADDR is 0x41000
5. Read PState SRAM area and save it to non-volatile storage only if there is more than 2 PState. Psuedo code provide to save PState SRAM as follows:

   ```
   foreach(qb_ps_sram[adr]) begin     qb_ps_sram[adr] = apb_read16
   (PSTATE_SRAM_BASE_ADDR  | adr,); end where PSTATE_SRAM_BASE_ADDR  is 0x A0000
   ```
6. After the register save operation is complete, the user should restore these settings for PHY mission mode operation.

apb_wr(32'h000C0080, 16'h0000); //DWC_DDRPHYA_DRTUB0_UcclkHclkEnables

apb_wr(32'h000D0000, 16'h0001); //DWC_DDRPHYA_APBONLY0__MicroContMuxSel

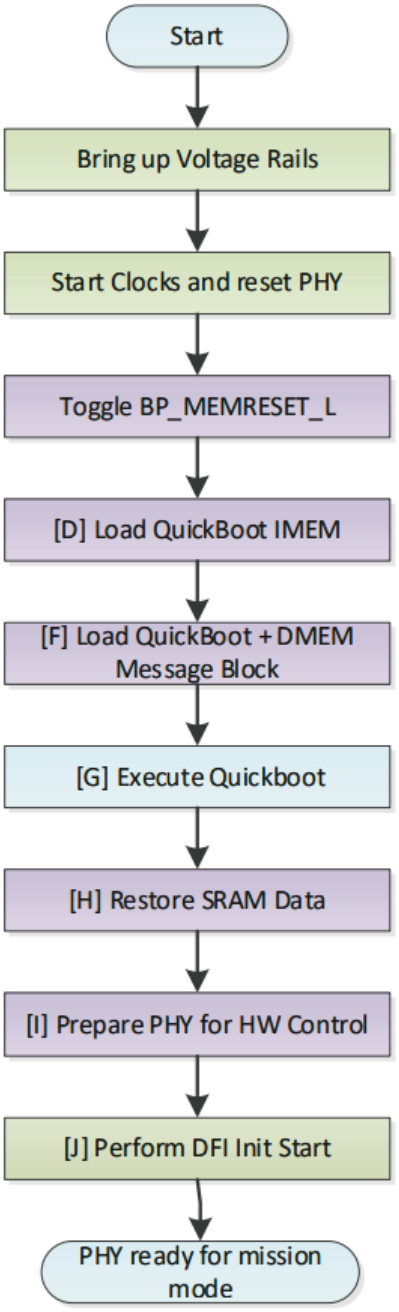
After following these steps, the PHY is ready to proceed to run mission mode traffic.

3 QuickBoot

The loading and running of QuickBoot is similar to loading and running training firmware with some key differences.

1. Unlike initial training firmware, PHYInit does not support QuickBoot. The correct APB writes need to be performed by the host.
2. QuickBoot does not assert the memory device reset. Memory reset is toggled by the host before loading the QuickBoot firmware.
3. In the time that TINIT3 is being satisfied, the IMEM and DMEM images are loaded
4. The DMEM image consists of three parts
    a. The firmware DMEM incv supplied by the QuickBoot firmware
    b. The message block to configure the QuickBoot
    c. The DMEM restore area

Figure 3-1 QuickBoot flow



3.1 QuickBoot flow

The QuickBoot flow is shown in Figure 3-1 on page 22. The First Boot flow is shown next to it for reference. The following sections explain the important QuickBoot steps in detail.

Note In the scenario where multiple PStates have completed initial training, Quickboot must be executed with the MessageBlock of the last trained PState

3.1.1 Initialization

Steps A and B are performed as described in the PUB databook. This procedure is the same as First Boot.

3.1.2 MemReset Toggle

The host must toggle memory device reset before running the QuickBoot image. It is up to the host to ensure that the toggle duration and TINIT3 is satisfied. The intention is for the host to toggle reset first and then to load the SRAM images while waiting for TINIT3 to expire.

After following the initialization procedure outlined in the PUB databook, memory reset will be asserted (BP_MEMRESET_L=0).

Note dwc_ddrphy_phyinit_userCustom_A_bringupPower() and dwc_ddrphy_phyinit_userCustom_B_startClockResetPhy() are to be executed by the host at this step.

The following APB commands should be sent by the host to de-assert reset:

apb_wr(32’h000D0000, 16’h0000); //DWC_DDRPHYA_APBONLY0__MicroContMuxSel

apb_wr(32'h0003F0A2 , 16’h0F00); // DWC_DDRPHYA_ACx__AsyncAcTxEn

apb_wr(32'h0003F042 , 16’h0F0F); // DWC_DDRPHYA_ACx__TxImpedanceDIFF0T

apb_wr(32'h0003F043 , 16’h0F0F); // DWC_DDRPHYA_ACx__TxImpedanceDIFF0C

if(PUB_Version == PUB1.02, PUB1.04, PUB1.05 or PUB2.00){

    apb_wr(32'h0003F0AC , 16’h07FF); // DWC_DDRPHYA_ACx__AcLnDisable

    apb_wr(32'h0001F00F , 16’h0006); // DWC_DDRPHYA_DBYTEx_ RxReplicaCtl04_p0

    apb_wr(32'h000200F8 , 16’h0000); // DWC_DDRPHYA_MASTER0__TxRdPtrInit

}apb_wr(32'h00020090 , 16’h0001); //DWC_DDRPHYA_MASTER0__PorControl

apb_wr(32’h00020060, 16'h0003); //DWC_DDRPHYA_MASTER0__MemResetL

After reset is de-asserted the host can proceed to the next steps, but must not start the microcontroller until TINIT3 is satisfied.

3.1.2.1 Full Speed PMU Clock Setting

The LPDDR54 PHY has a PMU clock divider option that allows for a half rate microcontroller clock. This can be used in case that the DFI clock frequency needed to initialize the DRAM is higher than the frequency

that the PMU supports. Note that using the divided clock will increase QuickBoot runtime and increase PMU SRAM access time from the APB bus.

In case the PMU needs to run at a slower speed the following setting should be used:

//Low Speed PMU clock

apb_wr(32'h000D0000, 16'h0000); //DWC_DDRPHYA_APBONLY0__MicroContMuxSel

apb_wr(32'h000C0080, 16'h0003); //DWC_DDRPHYA_DRTUB0_UcclkHclkEnables

In case the customers PMU can run at full rate, the following setting can be used to decrease QuickBoot runtime:

//High Speed PMU clock

apb_wr(32'h000D0000, 16'h0000); //DWC_DDRPHYA_APBONLY0__MicroContMuxSel

apb_wr(32'h000C0080, 16'h0007); //DWC_DDRPHYA_DRTUB0_UcclkHclkEnables

Once the QuickBoot firmware is complete and all SRAM accesses are complete, the PMU clock is not used, and the clock divider setting is no longer relevant.

3.1.2.2 Other APB Only register setting

NeverGateAcCsrClock and DfiCfgRdDataValidTicks are APB-only registers that user has to restore manually prior quickboot firmware runs. The value is same as what PHYINIT programed for training firmware. User can find out their respective value in step dwc_ddrphy_phyinit_C_initPhyConfig() of PHYINIT's output log.

Sample code to program these registers are: apb_wr(32'h000D0000, 16'h0000); //DWC_DDRPHYA_APBONLY0__MicroContMuxSel

dwc_ddrphy_apb_wr(32'hd0036,16'hxxxx); // DWC_DDRPHYA_APBONLY0_NeverGateAcCsrClock

dwc_ddrphy_apb_wr(32'hd0037,16'hxxxx); // DWC_DDRPHYA_APBONLY0_ DfiCfgRdDataValidTicks

Note NeverGateAcCsrClock does not exist prior PUB version v1.02

Note PHYINIT may skip the programming of DfiCfgRdDataValidTicks during dwc_ddrphy_phyinit_C_initPhyConfig(). If user can't find it being programmed by PHYINIT, then there is no need to restore it either prior starting quickboot firmware.

3.1.3 SRAM ECC Initialization

The LPDDR54 has an ECC. To avoid having to initialize the entire SRAM, it is recommended that this feature is disabled, by setting the following csr:

apb_wr(32'h000C0086, 16'h0001); // DWC_DDRPHYA_DRTUB0__ArcPmuEccCtl

If the csr is not set, then the entire ICC and DCC SRAM must be initialized with zero.

3.1.4 Step D Load QuickBoot IMEM

Load entire contents of QuickBoot instruction memory into the PHY IMEM. This operation is the same as the one performed by PHYInit.

3.1.5 Step F Load QuickBoot DMEM

3.1.5.1 Step F.1: Load QuickBoot DMEM image

Load the QuickBoot DMEM image (lpddr4_QuickBoot_dmem.incv) into the PHY data memory starting at locations 32'h58200 and ending at the last location in the DMEM image.

3.1.5.2 Step F.2: Load Correct Message Block

Load message block region starting at 32'h58000 to the end of the message block as described in the message block header (32'h58000). If there is more than 1 PState trained during the Frist Boot, load the message block of the last PState trained during First Boot. The Message Block has the same data as with First Boot with the exceptions in Table 3-1 on page 25

Note: Table 3-2 and Table 3-3 need to be updated accordingly with the DDR protocol chosen.

Table 3-1 Required QuickBoot Message Block Common Settings

Message Block Field	Required Value
SequenceCtrl	0x1 (DevInit Only)
QuickBoot	0x1

Table 3-2 Required QuickBoot Message Block Settings for LPDDR4

Message Block Field	Required Value
MR12_A0	TrainedVREFCA_A0
MR12_A1	TrainedVREFCA_A1
MR12_B0	TrainedVREFCA_B0
MR12_B1	TrainedVREFCA_B1
MR14_A0	TrainedVREFDQ_A0
MR14_A1	TrainedVREFDQ_A1
MR14_B0	TrainedVREFDQ_B0
MR14_B1	TrainedVREFDQ_B1

Table 3-3 Required QuickBoot Message Block Settings for LPDDR5

Message Block Field	Required Value
MR12_A0	TrainedVREFCA_A0

Message Block Field	Required Value
MR12_A1	TrainedVREFCA_A1
MR12_B0	TrainedVREFCA_B0
MR12_B1	TrainedVREFCA_B1
MR14_A0	TrainedVREFDQ_A0
MR14_A1	TrainedVREFDQ_A1
MR14_B0	TrainedVREFDQ_B0
MR14_B1	TrainedVREFDQ_B1
MR15_A0	TrainedVREFDQU_A0
MR15_A1	TrainedVREFDQU_A1
MR15_B0	TrainedVREFDQU_B0
MR15_B1	TrainedVREFDQU_B1
MR24_A0	TrainedDRAMDFE_A0
MR24_A1	TrainedDRAMDFE_A1
MR24_B0	TrainedDRAMDFE_B0
MR24_B1	TrainedDRAMDFE_B1
MR30_A0	TrainedDRAMDCA_A0
MR30_A1	TrainedDRAMDCA_A1
MR30_B0	TrainedDRAMDCA_B0
MR30_B1	TrainedDRAMDCA_B1

Note For Table 3-2 on page 25 or Table 3-3 on page 25, the Trained* value need to be read at the end of training, and then written to it respective MR location in Message Block prior Quickboot firmware starts to run.

Note If run in LPDDR4 protocol, update message block's filed as listed in Table 3-2 on page 25 and Table 3-3 on page 25.

If run in LPDDR5 protocol, update message block's filed as listed in Table Table 3-2 on page 25 and Table 3-3 on page 25

3.1.5.3 Step F.3: Load PHY SAVE Register into PHY data-memory SAVE AREA

The data saved in step I-2 in First Boot must be written to the save area starting at location 32'h58200.

Note that only the data read during step I-2 of First Boot should be written to this area. The address values are not written. Therefore, it is required that the CSR save data is written in 16-bit values in the order specified in the REG_LIST file. The QuickBoot firmware calculates the CSR address based on the order it is written to the save area.

Note It is required that the CSR save data is written to the QuickBoot save area in the exact order specified in the REG_LIST file.

3.1.5.4 QuickBoot Save format

Save data must be written in a very specific order. REG_LIST_DBYTE<n>.txt contains the exact list of PHY registers that need to read after First Boot and written to the SAVE area during QuickBoot.

When saving the register on the host side during First Boot, the host must read all CSRs in the REG_LIST_DBYTE and save the data in 16-bit values, regardless of the underlying size of the CSR. The address should not be written to the QuickBoot SAVE area.

For First Boot:

CSR read address from REG_LIST_DBYTE<N>.txt line 1 -> Host Non-volatile storage index 0

CSR read address from REG_LIST_DBYTE<N>.txt line 2 -> Host Non-volatile storage index 1

[…]

CSR read address from REG_LIST_DBYTE<N>.txt line N+1 -> Host nonvolatile storage index N

For Quick boot SAVE area writes:

Data from Host nonvolatile storage index 0 -> PHY DMEM address 0x58200

Data from Host nonvolatile storage index 1 -> PHY DMEM address 0x58201

[…]

Data from Host nonvolatile storage index N -> PHY DMEM address 0x58200+N

Where N is the number of CSR addresses in REG_LIST_DBYTE.

3.1.6 Step G Execute QuickBoot Firmware

3.1.6.1 Start Firmware

Once the IMEM and DMEM are loaded in the previous steps, the firmware should be executed with the following sequence:

1. apb_wr(32’h000D0000, 16’h0000); //DWC_DDRPHYA_APBONLY0__MicroContMuxSel
2. apb_wr(32’h000D0031, 16’h0001); //DWC_DDRPHYA_APBONLY0_DctWriteProt
3. apb_wr(32’h000D0000, 16’h0001); //DWC_DDRPHYA_APBONLY0__MicroContMuxSel
4. apb_wr(32'h000D0099, 16'h0009); //DWC_DDRPHYA_APBONLY0_MicroReset
5. apb_wr(32'h000D0099, 16'h0001); //DWC_DDRPHYA_APBONLY0_MicroReset
6. apb_wr(32'h000D0099, 16'h0000); //DWC_DDRPHYA_APBONLY0_MicroReset

7. Wait for the Quickboot firmware run to finish
8. apb_wr(32'h000D0099, 16'h0001); //DWC_DDRPHYA_APBONLY0_MicroReset
9. apb_wr(32'h000D0000, 16'h0000); //DWC_DDRPHYA_APBONLY0__MicroContMuxSel

Table 3-4 Mailbox Message Values

UctWriteOnlyShadow Value	Message meaning
0x00	Reset Value
0x07	Quickboot firmware finished successfully (firmware complete)
0xFF	Abnormal exit (firmware complete)

3.1.6.2 Firmware Runtime

Firmware runtime depends on a number of factors, including Dfi frequency, data rate and ARC clock divider. The customer can poll for the completion message to know when the firmware is complete, or they can wait a pre-determined time that is right for their system. The pre-determined time can be found by measuring a typical system running at a given configuration and waiting 10% longer than that value.

3.1.7 Step H Restore SRAM data

Step-H is required only if NumPStates is more than 2. The starting address of SRAM is at location 32'hA0000. The total amount of data needed to be restored is NumPStates*4*1024.

For Quickboot SAVE area writes:

Data from Host nonvolatile storage index 0 -> PHY DMEM address 0xA0000

Data from Host nonvolatile storage index 1 -> PHY DMEM address 0xA0001

[…]

Data from Host nonvolatile storage index N -> PHY DMEM address 0xA0000+N

Where N is the size of Pstate SRAM: (NumPStates*4*1024) – 1

Note NumPStates is defined by PHYINIT. Please see the PHYINIT application note for further details.

3.1.8 Step I Configure PHY for Hardware

After restoring the ACSM SRAM data, the PHY must be configured for hardware control:

apb_wr(32'h000D00E7, 16'h0400); //DWC_DDRPHYA_APBONLY0_SequencerOverride (PUB1.xx)

or

apb_wr(32'h000D00E7, 16'h0600); //DWC_DDRPHYA_APBONLY0_SequencerOverride (PUB2.xx)

apb_wr(32'h000C0080, 16'h0002); //DWC_DDRPHYA_DRTUB0_UcclkHclkEnables

apb_wr(32'h000D0000, 16'h0001); //DWC_DDRPHYA_APBONLY0__MicroContMuxSel

<ENABLE Memory Controller>

<Memory Controller issues dfi_init_start for the target Pstate>

3.1.9 Finish

When the dfi_init_start signal is asserted, DRAM drift compensation runs and will correct for variation in the DRAM device. After the completing the previous steps in the previous section, the PHY is in mission mode.