



LPDDR4/3 + DDR4/3 COMBO PHY

FXLPDDR4CMSA100NSHJOP | FXLPDDR4CMSA100EWHJOP

FXLPDDR4CMSD100NSHJOP | FXLPDDR4CMSD100EWHJOP

FXLPDDR4CMSC100NSHJOP | FXLPDDR4CMSC100EWHJOP

FXLPDDR4CMSC100HJOP

UMC 28 nm HPC Plus Process

Application Note

Rev.: 0.1

Issue Date: February 2025

REVISION HISTORY

FXLPDDR4CMSC100HJOP | FXLPDDR4CMSC100EWHJOP | FXLPDDR4CMSC100NSHJOP |

FXLPDDR4CMSC100EWHJOP | FXLPDDR4CMSC100NSHJOP | FXLPDDR4CMSC100EWHJOP |

FXLPDDR4CMSC100HJOP Application Note

Date	Rev.	From	To
Feb. 2025	0.1	-	Original

©Copyright Faraday Technology, 2025
All Rights Reserved.

Printed in Taiwan 2025

Faraday and the Faraday Logo are trademarks of Faraday Technology Corporation in Taiwan and/or other countries. Other company, product and service names may be trademarks or service marks of others.

All information contained in this document is subject to change without notice. The products described in this document are NOT intended for use in implantation or other life support application where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Faraday's product specification or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Faraday or third parties. All information contained in this document was obtained in specific environments, and is presented as an illustration. The results obtained in other operating environments may vary.

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED ON AN "AS IS" BASIS. In no event will Faraday be liable for damages arising directly or indirectly from any use of the information contained in this document.

Faraday Technology Corporation
No. 5, Li-Hsin Road III, Hsinchu Science Park, Hsinchu City, Taiwan 300, R.O.C.

Faraday's home page can be found at:
<http://www.faraday-tech.com>

TABLE OF CONTENTS

Chapter 1	Introduction	1
1.1	Versions of the IPs	2
1.2	Features	2
1.3	Overview.....	3
1.4	Block Integration Diagram	4
Chapter 2	Signal Mapping.....	5
2.1	LPDDR4 Command/Address Pin Map	7
2.1.1	Signal Channel Mode (CHA)	7
2.1.2	Dual Channel Mode (CHA/CHB)	8
2.2	LPDDR3 Command/Address Pin Map	10
2.3	DDR4 Command/Address Pin Map.....	11
2.4	DDR3 Command/Address Pin Map.....	13
2.5	Data Byte Map	15
2.5.1	LPDDR4 Dual Channel Mode	15
2.5.2	Other Modes	16
Chapter 3	PHY Initialization	17
3.1	Overview.....	18
3.2	PLL Description	18
3.2.1	PLL Setting	19
3.2.2	PLL Example.....	20
3.3	ZQ Calibration.....	21
3.3.1	Restart ZQ calibration.....	22
3.3.2	Bypass ZQ Calibration.....	23
3.4	PHY Initialization Sequence	23
3.4.1	PHY Initialization Example	30
Chapter 4	PHY Testing.....	35
4.1	Overview.....	36
4.2	Scan Test	36
4.3	Loopback Test	37
4.3.1	Introduction	37
4.3.2	Loopback Test Sequence	38
4.4	I/O DC Test Mode	44
4.4.1	IDDDQ Test Mode	47
4.4.2	Input Pin Leakage Mode.....	47
4.4.3	I/O Max. Driving Test of Output Buffer	48
4.4.4	I/O DC Parameter Test of Input Buffer	49
4.4.5	I/O DC Parameter Test of Output Buffer	52
4.5	Mass Production Test Pattern.....	53

Chapter 5 Integration Instruction.....	55
5.1 refclk Requirement	56
5.2 Clock Tree Implementation.....	56
5.3 Floor Plan, Abutment, and Special Connections	58
5.3.1 Special Nets	58
5.3.2 Special Connection.....	59
5.3.3 Floor Plan and Abutment	71
5.3.4 Clock and Abutted Signal Connections in RTL	72
5.4 Retention Pin Connection of LPDDR4/LPDDR3-DDR4/DDR3 Combo PHY	76
5.5 Routing of Compensation Bits from FXLPDDR4MCFD100*HJOP to Other Blocks	77
5.6 Pad Sequence in Each Block	79
5.7 Bit Swapping in Data Block and Address/Command Block.....	85
5.8 Board-level Connection for Compensation Block in FXLPDDR4CMSC100*HJOP	86
5.9 Power Consumption Estimation	86
5.9.1 Power Consumption in DDR4	87
5.9.2 Power Consumption in DDR3	87
5.9.3 Power Consumption in LPDDR4	88
5.9.4 Power Consumption in LPDDR3	89
5.9.5 Total Power Estimation Example	89
Chapter 6 System Application Circuit	91
6.1 Termination Connections.....	92
6.2 Recommended Connection for ODT_CA Pins in LPDDR4	94
6.3 Built-in VREF Generator Usage	94
6.4 Package Requirement for I/O Power/Ground.....	94
6.5 Package Requirement for Core Power	96
6.6 PCB Layout Guideline	96
6.7 Power Sequencing Requirement for PHY	96
Chapter 7 FAQ.....	97
Q1: Do I need “start-up sequence” for all FXLPDDR4CMSC100*HJOP, FXLPDDR4CMSC100*HJOP, and FXLPDDR4CMSC100*HJOP?	98
Q2: How to run the zero delay simulation?.....	98
Q3: How to run the post simulation with SDF annotate?	98
Q4: Does user have the implementation check list of DDR-PHY?	99

LIST OF TABLES

Table 1-1.	Versions of IPs	2
Table 2-1.	Number of APHY Used in Different Applications.....	6
Table 2-2.	LPDDR4 Single Channel DFIW-to-DRAM Command/Address Pin Map.....	7
Table 2-3.	LPDDR4 Dual Channel DFIW-to-DRAM Command/Address Pin Map.....	8
Table 2-4.	LPDDR3 DFIW-to-DRAM Command/Address Pin Map	10
Table 2-5.	DDR4 DFIW-to-DRAM Command/Address Pin Map	11
Table 2-6.	DDR3 DFIW-to-DRAM Command/Address Pin Map	13
Table 2-7.	LPDDR4 Dual Channel DFIW-to-DRAM Data Pin Map.....	15
Table 2-8.	LPDDR4 Single Channel/LPDDR3/DDR4/DDR3 DFIW-to-DRAM Data Pin Map	16
Table 3-1.	PLL Specification.....	19
Table 3-2.	PLL Setting for Generating 800 MHz without SSC Enable	20
Table 3-3.	CPHY Register Setting of ZQ Calibration for Different DRAM Applications	22
Table 3-4.	Resistance of RZQ in Different DRAM Applications	22
Table 3-5.	Pull-Up/Pull-Down Resistance Control Setting.....	23
Table 3-6.	APHY wby2ck Setting.....	24
Table 3-7.	DPHY wby2ck Setting	24
Table 3-8.	CPHY wby2ck Setting.....	24
Table 3-9.	APHY EFIFO Setting.....	25
Table 3-10.	DPHY EFIFO Setting.....	25
Table 3-11.	APHY phyck_skew Setting.....	25
Table 3-12.	DPHY phyck_skew Setting	25
Table 3-13.	CPHY phyck_skew Setting.....	26
Table 3-14.	CPHY ZQ Calibration Setting	26
Table 3-15.	CPHY PLL Setting.....	27
Table 3-16.	APHY DCD Setting.....	28
Table 3-17.	DPHY DCD Setting.....	28
Table 3-18.	APHY ADDLL Wait Time Setting	28
Table 3-19.	DPHY ADDLL Wait Time Setting	28
Table 3-20.	APHY soft_resetn Setting.....	29
Table 3-21.	DPHY soft_resetn Setting.....	29
Table 3-22.	CPHY soft_resetn Setting	29
Table 3-23.	PHY Initialization Sequence	31
Table 4-1.	Scan Patterns.....	36
Table 4-2.	Coverage Rate	37
Table 4-3.	APHY Loopback Training Done Observation.....	39
Table 4-4.	APHY Loopback Training Phase Observation.....	39
Table 4-5.	DPHY Loopback Training Done Observation.....	39
Table 4-6.	DPHY Loopback Training Phase Observation.....	39

Table 4-7.	APHY per bit PRBS Test Result Observation	40
Table 4-8.	DPHY per bit PRBS Test Result Observation	43
Table 4-9.	Pin Descriptions of Output Buffer when FXLPDDR4CMSC100*HJOP in Test Mode	46
Table 4-10.	Pin Descriptions of Output Buffer when FXLPDDR4CMSC100*HJOP in Test Mode	46
Table 4-11.	Pin Descriptions of input put Buffer when FXLPDDR4CMSC100*HJOP in Test Mode	46
Table 4-12.	Mass Production Test Pattern List.....	53
Table 5-1.	refclk Requirement.....	56
Table 5-2.	Registers for Balancing Phyck among all PHYs, phyck_skewcode	57
Table 5-3.	Parameter in Worst Cold Case for Calculating Value of phyck_skewcode	57
Table 5-4.	Wby2ck Related Setting when Set APHY1 as Wby2ck Source (Master)	60
Table 5-5.	Wby2ck Related Setting when Set DPHY as Wby2ck Source (Master)	62
Table 5-6.	rptr_*_rstn_* Related Setting when Setting APHY0 as rptr_rstn Source (Master).....	68
Table 5-7.	rptr_*_rstn_* Related Setting when Setting DPHY0 as rptr_rstn Source (Master).....	70
Table 5-8.	PHYCK Connections between APHY1 and CPHY	72
Table 5-9.	PHYCK Connections between APHY0 and APHY1	73
Table 5-10.	PHYCK Connections between DPHY1 and APHY0	73
Table 5-11.	PHYCK Connections between DPHY0 and DPHY1.....	73
Table 5-12.	Special Signals Connection between APHY1 and CPHY	74
Table 5-13.	Special Signals Connection between APHY0 and APHY1	74
Table 5-14.	Special Signals Connection between DPHY1 and APHY0.....	74
Table 5-15.	Special Signals Connection between DPHY0 and DPHY1.....	75
Table 5-16.	Power/Ground list	75
Table 5-17.	Power Consumption in DDR4 at Different Data Rates.....	87
Table 5-18.	Power Consumption in DDR3 at Different Data Rates	87
Table 5-19.	Power Consumption in LPDDR4 at Different Data Rates	88
Table 5-20.	Power Consumption in LPDDR3 at Different Data Rates	89
Table 7-1.	Implementation Check List of DDR-PHY.....	99

LIST OF FIGURES

Figure 1-1. Integration Diagram of Controller and PHY Blocks	4
Figure 3-1. Connecton Relationship between PLL361 and PLL365	18
Figure 3-2. Floorplan Example	30
Figure 3-3. wby2ck Signal Connection Example	30
Figure 3-4. FIFO Signal Connection Example	31
Figure 4-1. Loopback Path with Loopback Module	38
Figure 4-2. APHY I/O Configuration in Test Mode	45
Figure 4-3. DPHY I/O Configuration in Test Mode	45
Figure 4-4. NAND Tree Function of FXLPDDR4CMSC100*HJOP Block	50
Figure 5-1. Conceptual Block Diagram of 16-bit Single-channel LPDDR4 PHY for Clock Tree Connection	56
Figure 5-2. Conceptual Block Diagram of 16-bit Single-channel LPDDR4 PHY for Abutted Signals Connection	58
Figure 5-3. Conceptual Block Diagram of 16-bit DDR4 PHY in NS Orientation for Abutted Signals Connection	58
Figure 5-4. Wby2ck Connection Example when set APHY1 as Wby2ck Source (Master)	60
Figure 5-5. Wby2ck Connection Example when Set DPHY0 as Wby2ck Source (Master)	61
Figure 5-6. Choose rptr_*_rstn_* Source in Odd Case	63
Figure 5-7. Choose rptr_*_rstn_* Source in Even Case	63
Figure 5-8. Guidance to Select rptr_*_rstn_* or rptr_*_rstn_fwd_* as Connection	64
Figure 5-9. Guidance to Count Total Number of Connection to Master	64
Figure 5-10. Guidance to Count Total Number of Connection to Block Abut	65
Figure 5-11. Guidance to Count Total Number of Connection to Block Not Abut Master	65
Figure 5-12. Guidance to Find the Synchronous Target Case1	66
Figure 5-13. Guidance to Find Synchronous Target Case2	67
Figure 5-14. rptr_*_rstn_* Connection Example when Setting APHY0 as rptr_rstn Source (Master)	68
Figure 5-15. rptr_*_rstn_* Connection Example when Setting DPHY0 as rptr_rstn Source (Master)	69
Figure 5-16. rptr_*_rstn_* Connection Prohibited Example	71
Figure 5-17. Abutment of Different Blocks for a 16-bit Single-channel LPDDR4 PHY	71
Figure 5-18. Abutment of Different Blocks for a 16-bit DDR4 PHY in NS Orientation	72
Figure 5-19. Floorplan Example of Clock and Abutted Signals Connections	72
Figure 5-20. Timing Sequence to Go into PHY Retention (Not Drawn to Scale)	77
Figure 5-21. Compensation Bits Routing from FXLPDDR4CMSC100*HJOP to FXLPDDR4CMSC100* HJOP /FXLPDDR4CMSC100* HJOP for a 1-channel 16-bit LPDDR4 PHY	78
Figure 5-22. Pad Sequence of FXLPDDR4CMSC100NSHJOP	79
Figure 5-23. Pad Sequence of FXLPDDR4CMSC100NSHJOP	80
Figure 5-24. Pad Sequence of FXLPDDR4CMSC100NSHJOP	81
Figure 5-25. Pad Sequence of FXLPDDR4CMSC100EWHJOP	82
Figure 5-26. Pad Sequence of FXLPDDR4CMSC100EWHJOP	83
Figure 5-27. Pad Sequence of FXLPDDR4CMSC100EWHJOP	84
Figure 5-28. Pad Sequence of FXLPDDR4CMSC100HJOP (Placed on the Corner)	85

Figure 5-29. Required Connection to PAD (“comp-pad_o”) on the Board	86
Figure 6-1. Connection of Termination to CK_T and CK_C on PCB.....	92
Figure 6-2. First Example of Connection of Termination to Other Pins in Address Block on PCB	93
Figure 6-3. Second Example of Connection of Termination to Other Pins in Address Block on PCB	93
Figure 6-4. Connection of VCCIO_DDR and VCCIO_DDRCK of PHY in Package.....	94
Figure 6-5. Connection of VCC18A_REG of PHY in Package.....	95
Figure 6-6. Connection of GND, GNDIO_DDR and GNDIO_DDRCK of PHY in Package	95
Figure 6-7. Connection of GNDA of PHY in Package	95

Chapter 1

Introduction

This chapter contains the following sections:

- 1.1 Versions of the IPs
- 1.2 Features
- 1.3 Overview
- 1.4 Block Integration Diagram

1.1 Versions of the IPs

The version of each IP is listed in **Table 1-1**.

Table 1-1. Versions of IPs

IP Name	Poly Direction	IP Version
FXLPDDR4CMSC100HJOP	NS	0.1.0
FXLPDDR4CMSC100EWHJOP	EW	0.1.0
FXLPDDR4CMSC100NSHJOP	NS	0.1.0
FXLPDDR4CMSC100EWHJOP	EW	0.1.0
FXLPDDR4CMSC100NSHJOP	NS	0.1.0
FXLPDDR4CMSC100EWHJOP	EW	0.1.0
FXLPDDR4CMSC100HJOP	On the Corner	0.1.0

Note: (0.1.0) for metal option: 1P8M02006, 1P8M11006

1.2 Features

- UMC 28 nm HPC Plus process
- Targeted for Wire Bond application
- Compliant with JEDEC LPDDR4/LPDDR3/DDR4/DDR3 standard
- Supports data rates up to 1600 Mbps for DDR4
- Supports data rates up to 1600 Mbps for LPDDR4
- Supports data rates up to 1600 Mbps for DDR3/LPDDR3
- Supports 1:1 and 1:2 and 1:4 (Controller clock to DRAM bus clock) modes
- Configurable channel width in 8-bit increments
- Supports chip-to-chip configurations
- Built-in Delay-Locked Loop (DLL) and COMBO PHY combo I/O
- Synchronizes read data from SDRAM domain to DFI clock domain of ASIC
- Programmable output driving strength
- Programmable On-Die Termination (ODT) resistor
- Supports IDQ test mode
- ESD (HBM2KV/CDM6A) robustness and latch-up immunity proven by silicon verification
- Supports Read/Write Data eye training
- Supports Read/Write Vref training
- Supports Read/Write leveling training
- Supports Second Write leveling training
- Supports Command/Address training
- Supports Auto duty cycle adjustment
- Supports skew correction among signal groups
- Digital core/Analog supply voltage range: 0.81 V ~ 0.99 V
- Operating junction temperature range: -40 °C ~ 125 °C
- Minimum metal requirement: 8 metal layers
- Pad opening size: 55.46 µm * 74.08 µm (before shrink)

1.3 Overview

Faraday offers a complete silicon-verified solution for IC designers to integrate the physical interface (PHY) into a real design or to use it as a reference design. The complete PHY solution can be used to set up a standard integration procedure that guarantees a successful system power-up.

In general, memory interface designers separately choose an I/O library with the COMBO PHY specification, a DLL, and a PLL from different IP vendors to cut down the cost. At the same time, designers may spend a great deal of efforts and time in solving integration issues. As an alternative, Faraday offers a complete PHY solution that can effortlessly deliver designs and reduce time-to-market for end products.

This complete PHY solution comprises of three hard macros: the data block, FXLPDDR4CMSC100*HJOP, the command and address block, FXLPDDR4CMSC100*HJOP, and the clock generation block, FXLPDDR4CMSC100*HJOP. Every complete PHY solution must contain a minimum of one data block, one command and address block, and one clock generation block. Memory interface designers can configure the width of an external data bus in 8-bit increments by adding additional data blocks to the design. In some applications, it is a feasible solution of using only the data block and the command and address block.

1.4 Block Integration Diagram

Figure 1-1 shows the typical architecture of the DDR memory system that utilizes the Combo PHY blocks. This system includes a memory controller which controls the DDR read/write operations, a combo PHY (Wrapper and hardened PHY) and an off-chip DRAM. The Wrapper contains the training logic to calibrate the hardened PHY timing settings.

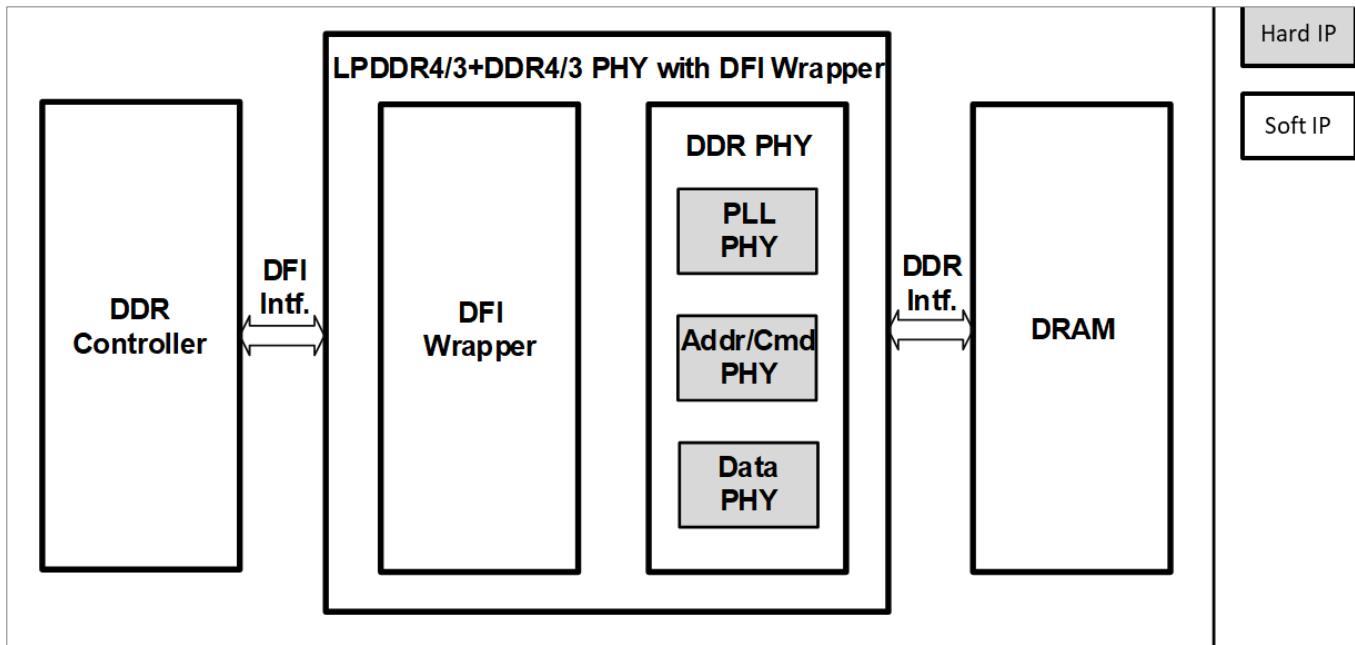


Figure 1-1. Integration Diagram of Controller and PHY Blocks

Chapter 2

Signal Mapping

This chapter contains the following sections:

- 2.1 LPDDR4 Command/Address Pin Map
- 2.2 LPDDR3 Command/Address Pin Map
- 2.3 DDR4 Command/Address Pin Map
- 2.4 DDR3 Command/Address Pin Map
- 2.5 Data Byte Map

This chapter provides the default connection mapping table from DFIW to DRAM interface when DDR PHY operates in different DRAM modes.

This LPDDR4 combo PHY supports LPDDR4/LPDDR3/DDR4/DDR3 SDRAM and the number of command/address pad of these SDRAMs are quite different, so the number of APHY will be different in different SDRAMs. In order to save area for LPDDR4/LPDDR3 applications, there are only 18 I/O pads as CA[9:0], CS[1:0], CTRL[3:0], and CK_t/CK_c in APHY. Therefore, in DDR4/DDR3 applications, please use two APHYs to meet the command/address pad count for SDRAM specification. For summary information of APHY application, please refer to **Table 2-1** for more details.

Table 2-1. Number of APHY Used in Different Applications

Application	Number of APHY
LPDDR4, 1 channel	1
LPDDR4, 2 channels	2
LPDDR3	1
DDR4	2
DDR3	2

This IP supports the following types of pin swap:

1. In DPHY, it supports DQ0 ~ DQ7, and DM/DBI swap.
 - Not support pin swap between bytes
For example, DQ1 in DPHY_0 swap with DQ2 in DPHY_1 (**Prohibited**)
2. In APHY,
 - CA_a0[0] ~ CA_a0[9] and CA_a1[0] ~ CA_a1[9] can be swapped between APHY0 and APHY1.
 - CS_a0[0] ~ CS_a0[1] and CS_a1[0] ~ CS_a1[1] can be swapped between APHY0 and APHY1.
 - CTRL_a0[0] ~ CTRL_a0[3] and CTRL_a1[0] ~ CTRL_a1[3] can be swapped between APHY0 and APHY1.
3. CK and CKB do not support pin swap.
4. DQS_T and DQS_C do not support pin swap.

The swap logic is implemented in DFIW. Please refer to the sections for “pin swap” in DFIW500 series document for more details.

2.1 LPDDR4 Command/Address Pin Map

2.1.1 Signal Channel Mode (CHA)

Table 2-2. LPDDR4 Single Channel DFIW-to-DRAM Command/Address Pin Map

DFIW Signal	Bump Name	LPDDR4 CHA
wca_{even/odd}_bit_{p0~p3}_a0_i[0]	CA_a0[0]	CA0_A
wca_{even/odd}_bit_{p0~p3}_a0_i[1]	CA_a0[1]	CA1_A
wca_{even/odd}_bit_{p0~p3}_a0_i[2]	CA_a0[2]	CA2_A
wca_{even/odd}_bit_{p0~p3}_a0_i[3]	CA_a0[3]	CA3_A
wca_{even/odd}_bit_{p0~p3}_a0_i[4]	CA_a0[4]	CA4_A
wca_{even/odd}_bit_{p0~p3}_a0_i[5]	CA_a0[5]	CA5_A
wca_{even/odd}_bit_{p0~p3}_a0_i[6]	CA_a0[6]	-
wca_{even/odd}_bit_{p0~p3}_a0_i[7]	CA_a0[7]	-
wca_{even/odd}_bit_{p0~p3}_a0_i[8]	CA_a0[8]	-
wca_{even/odd}_bit_{p0~p3}_a0_i[9]	CA_a0[9]	-
wcs_{even/odd}_bit_{p0~p3}_a0_i[0]	CS_a0[0]	CS0_A
wcs_{even/odd}_bit_{p0~p3}_a0_i[1]	CS_a0[1]	CS1_A
wctrl_{even/odd}_bit_{p0~p3}_a0_i[0]	CTRL_a0[0]	CKE0_A
wctrl_{even/odd}_bit_{p0~p3}_a0_i[1]	CTRL_a0[1]	CKE1_A
wctrl_{even/odd}_bit_{p0~p3}_a0_i[2]	CTRL_a0[2]	RESET_n
wctrl_{even/odd}_bit_{p0~p3}_a0_i[3]	CTRL_a0[3]	-
wck_{even/odd}_bit_{p0~p3}_a0_i	CK_t_a0	CK_t_A
	CK_c_a0	CK_c_A

2.1.2 Dual Channel Mode (CHA/CHB)

Table 2-3. LPDDR4 Dual Channel DFIW-to-DRAM Command/Address Pin Map

DFIW Signal	Bump Name	LPDDR4 CHA
wca_{even/odd}_bit_{p0~p3}_a0_i[0]	CA_a0[0]	CA0_A
wca_{even/odd}_bit_{p0~p3}_a0_i[1]	CA_a0[1]	CA1_A
wca_{even/odd}_bit_{p0~p3}_a0_i[2]	CA_a0[2]	CA2_A
wca_{even/odd}_bit_{p0~p3}_a0_i[3]	CA_a0[3]	CA3_A
wca_{even/odd}_bit_{p0~p3}_a0_i[4]	CA_a0[4]	CA4_A
wca_{even/odd}_bit_{p0~p3}_a0_i[5]	CA_a0[5]	CA5_A
wca_{even/odd}_bit_{p0~p3}_a0_i[6]	CA_a0[6]	-
wca_{even/odd}_bit_{p0~p3}_a0_i[7]	CA_a0[7]	-
wca_{even/odd}_bit_{p0~p3}_a0_i[8]	CA_a0[8]	-
wca_{even/odd}_bit_{p0~p3}_a0_i[9]	CA_a0[9]	-
wcs_{even/odd}_bit_{p0~p3}_a0_i[0]	CS_a0[0]	CS0_A
wcs_{even/odd}_bit_{p0~p3}_a0_i[1]	CS_a0[1]	CS1_A
wctrl_{even/odd}_bit_{p0~p3}_a0_i[0]	CTRL_a0[0]	CKEO_A
wctrl_{even/odd}_bit_{p0~p3}_a0_i[1]	CTRL_a0[1]	CKE1_A
wctrl_{even/odd}_bit_{p0~p3}_a0_i[2]	CTRL_a0[2]	RESET_n
wctrl_{even/odd}_bit_{p0~p3}_a0_i[3]	CTRL_a0[3]	-
wck_{even/odd}_bit_{p0~p3}_a0_i	CK_t_a0	CK_t_A
	CK_c_a0	CK_c_A

DFIW Signal	Bump Name	LPDDR4 CHB
wca_{even/odd}_bit_{p0~p3}_a1_i[0]	CA_a1[0]	CA0_B
wca_{even/odd}_bit_{p0~p3}_a1_i[1]	CA_a1[1]	CA1_B
wca_{even/odd}_bit_{p0~p3}_a1_i[2]	CA_a1[2]	CA2_B
wca_{even/odd}_bit_{p0~p3}_a1_i[3]	CA_a1[3]	CA3_B
wca_{even/odd}_bit_{p0~p3}_a1_i[4]	CA_a1[4]	CA4_B
wca_{even/odd}_bit_{p0~p3}_a1_i[5]	CA_a1[5]	CA5_B
wca_{even/odd}_bit_{p0~p3}_a1_i[6]	CA_a1[6]	-
wca_{even/odd}_bit_{p0~p3}_a1_i[7]	CA_a1[7]	-
wca_{even/odd}_bit_{p0~p3}_a1_i[8]	CA_a1[8]	-
wca_{even/odd}_bit_{p0~p3}_a1_i[9]	CA_a1[9]	-
wcs_{even/odd}_bit_{p0~p3}_a1_i[0]	CS_a1[0]	CS0_B
wcs_{even/odd}_bit_{p0~p3}_a1_i[1]	CS_a1[1]	CS1_B
wctrl_{even/odd}_bit_{p0~p3}_a1_i[0]	CTRL_a1[0]	CKEO_B
wctrl_{even/odd}_bit_{p0~p3}_a1_i[1]	CTRL_a1[1]	CKE1_B
wctrl_{even/odd}_bit_{p0~p3}_a1_i[2]	CTRL_a1[2]	-
wctrl_{even/odd}_bit_{p0~p3}_a1_i[3]	CTRL_a1[3]	-
wck_{even/odd}_bit_{p0~p3}_a1_i	CK_t_a1	CK_t_B
	CK_c_a1	CK_c_B

2.2 LPDDR3 Command/Address Pin Map

Table 2-4. LPDDR3 DFIW-to-DRAM Command/Address Pin Map

DFIW Signal	Bump Name	LPDDR3
wca_{even/odd}_bit_{p0~p3}_a0_i[0]	CA_a0[0]	CA0
wca_{even/odd}_bit_{p0~p3}_a0_i[1]	CA_a0[1]	CA1
wca_{even/odd}_bit_{p0~p3}_a0_i[2]	CA_a0[2]	CA2
wca_{even/odd}_bit_{p0~p3}_a0_i[3]	CA_a0[3]	CA3
wca_{even/odd}_bit_{p0~p3}_a0_i[4]	CA_a0[4]	CA4
wca_{even/odd}_bit_{p0~p3}_a0_i[5]	CA_a0[5]	CA5
wca_{even/odd}_bit_{p0~p3}_a0_i[6]	CA_a0[6]	CA6
wca_{even/odd}_bit_{p0~p3}_a0_i[7]	CA_a0[7]	CA7
wca_{even/odd}_bit_{p0~p3}_a0_i[8]	CA_a0[8]	CA8
wca_{even/odd}_bit_{p0~p3}_a0_i[9]	CA_a0[9]	CA9
wcs_{even/odd}_bit_{p0~p3}_a0_i[0]	CS_a0[0]	CS0_n
wcs_{even/odd}_bit_{p0~p3}_a0_i[1]	CS_a0[1]	CS1_n
wctrl_{even/odd}_bit_{p0~p3}_a0_i[0]	CTRL_a0[0]	CKE0
wctrl_{even/odd}_bit_{p0~p3}_a0_i[1]	CTRL_a0[1]	CKE1
wctrl_{even/odd}_bit_{p0~p3}_a0_i[2]	CTRL_a0[2]	ODT0
wctrl_{even/odd}_bit_{p0~p3}_a0_i[3]	CTRL_a0[3]	ODT1
wck_{even/odd}_bit_{p0~p3}_a0_i	CK_t_a0	CK_t
	CK_c_a0	CK_c

2.3 DDR4 Command/Address Pin Map

Table 2-5. DDR4 DFIW-to-DRAM Command/Address Pin Map

DFIW Signal	Bump Name	DDR4
wca_{even/odd}_bit_{p0~p3}_a0_i[0]	CA_a0[0]	A0
wca_{even/odd}_bit_{p0~p3}_a0_i[1]	CA_a0[1]	A1
wca_{even/odd}_bit_{p0~p3}_a0_i[2]	CA_a0[2]	A2
wca_{even/odd}_bit_{p0~p3}_a0_i[3]	CA_a0[3]	A3
wca_{even/odd}_bit_{p0~p3}_a0_i[4]	CA_a0[4]	A4
wca_{even/odd}_bit_{p0~p3}_a0_i[5]	CA_a0[5]	A5
wca_{even/odd}_bit_{p0~p3}_a0_i[6]	CA_a0[6]	A6
wca_{even/odd}_bit_{p0~p3}_a0_i[7]	CA_a0[7]	A7
wca_{even/odd}_bit_{p0~p3}_a0_i[8]	CA_a0[8]	A8
wca_{even/odd}_bit_{p0~p3}_a0_i[9]	CA_a0[9]	A9
wcs_{even/odd}_bit_{p0~p3}_a0_i[0]	CS_a0[0]	CS0
wcs_{even/odd}_bit_{p0~p3}_a0_i[1]	CS_a0[1]	CS1
wctrl_{even/odd}_bit_{p0~p3}_a0_i[0]	CTRL_a0[0]	CKEO
wctrl_{even/odd}_bit_{p0~p3}_a0_i[1]	CTRL_a0[1]	CKE1
wctrl_{even/odd}_bit_{p0~p3}_a0_i[2]	CTRL_a0[2]	ODT0
wctrl_{even/odd}_bit_{p0~p3}_a0_i[3]	CTRL_a0[3]	ODT1
wck_{even/odd}_bit_{p0~p3}_a0_i	CK_t_a0	CK_t
	CK_c_a0	CK_c
wca_{even/odd}_bit_{p0~p3}_a1_i [0]	CA_a1[0]	A10/AP
wca_{even/odd}_bit_{p0~p3}_a1_i [1]	CA_a1[1]	A11
wca_{even/odd}_bit_{p0~p3}_a1_i [2]	CA_a1[2]	A12/BC_n

DFIW Signal	Bump Name	DDR4
wca_{even/odd}_bit_{p0~p3}_a1_i[3]	CA_a1[3]	A13
wca_{even/odd}_bit_{p0~p3}_a1_i[4]	CA_a1[4]	A14/WE_n
wca_{even/odd}_bit_{p0~p3}_a1_i[5]	CA_a1[5]	A15/CAS_n
wca_{even/odd}_bit_{p0~p3}_a1_i[6]	CA_a1[6]	A16/RAS_n
wca_{even/odd}_bit_{p0~p3}_a1_i[7]	CA_a1[7]	BA0
wca_{even/odd}_bit_{p0~p3}_a1_i[8]	CA_a1[8]	BA1
wca_{even/odd}_bit_{p0~p3}_a1_i[9]	CA_a1[9]	TEN
wcs_{even/odd}_bit_{p0~p3}_a1_i[0]	CS_a1[0]	BG0
wcs_{even/odd}_bit_{p0~p3}_a1_i[1]	CS_a1[1]	BG1
wctrl_{even/odd}_bit_{p0~p3}_a1_i[0]	CTRL_a1[0]	RESET_n
wctrl_{even/odd}_bit_{p0~p3}_a1_i[1]	CTRL_a1[1]	ALERT_n
wctrl_{even/odd}_bit_{p0~p3}_a1_i[2]	CTRL_a1[2]	ACT_n
wctrl_{even/odd}_bit_{p0~p3}_a1_i[3]	CTRL_a1[3]	PAR
wck_{even/odd}_bit_{p0~p3}_a1_i	CK_t_a1	-
	CK_c_a1	-

2.4 DDR3 Command/Address Pin Map

Table 2-6. DDR3 DFIW-to-DRAM Command/Address Pin Map

DFIW Signal	Bump Name	DDR3
wca_{even/odd}_bit_{p0~p3}_a0_i[0]	CA_a0[0]	A0
wca_{even/odd}_bit_{p0~p3}_a0_i[1]	CA_a0[1]	A1
wca_{even/odd}_bit_{p0~p3}_a0_i[2]	CA_a0[2]	A2
wca_{even/odd}_bit_{p0~p3}_a0_i[3]	CA_a0[3]	A3
wca_{even/odd}_bit_{p0~p3}_a0_i[4]	CA_a0[4]	A4
wca_{even/odd}_bit_{p0~p3}_a0_i[5]	CA_a0[5]	A5
wca_{even/odd}_bit_{p0~p3}_a0_i[6]	CA_a0[6]	A6
wca_{even/odd}_bit_{p0~p3}_a0_i[7]	CA_a0[7]	A7
wca_{even/odd}_bit_{p0~p3}_a0_i[8]	CA_a0[8]	A8
wca_{even/odd}_bit_{p0~p3}_a0_i[9]	CA_a0[9]	A9
wcs_{even/odd}_bit_{p0~p3}_a0_i[0]	CS_a0[0]	CS0#
wcs_{even/odd}_bit_{p0~p3}_a0_i[1]	CS_a0[1]	CS1#
wctrl_{even/odd}_bit_{p0~p3}_a0_i[0]	CTRL_a0[0]	CKEO
wctrl_{even/odd}_bit_{p0~p3}_a0_i[1]	CTRL_a0[1]	CKE1
wctrl_{even/odd}_bit_{p0~p3}_a0_i[2]	CTRL_a0[2]	ODT0
wctrl_{even/odd}_bit_{p0~p3}_a0_i[3]	CTRL_a0[3]	ODT1
wck_{even/odd}_bit_{p0~p3}_a0_i	CK_t_a0	CK
	CK_c_a0	CK#
wca_{even/odd}_bit_{p0~p3}_a1_i[0]	CA_a1[0]	A10/AP
wca_{even/odd}_bit_{p0~p3}_a1_i[1]	CA_a1[1]	A11
wca_{even/odd}_bit_{p0~p3}_a1_i[2]	CA_a1[2]	A12/BC#

DFIW Signal	Bump Name	DDR3
wca_{even/odd}_bit_{p0~p3}_a1_i[3]	CA_a1[3]	A13
wca_{even/odd}_bit_{p0~p3}_a1_i[4]	CA_a1[4]	A14
wca_{even/odd}_bit_{p0~p3}_a1_i[5]	CA_a1[5]	A15
wca_{even/odd}_bit_{p0~p3}_a1_i[6]	CA_a1[6]	BA0
wca_{even/odd}_bit_{p0~p3}_a1_i[7]	CA_a1[7]	BA1
wca_{even/odd}_bit_{p0~p3}_a1_i[8]	CA_a1[8]	BA2
wca_{even/odd}_bit_{p0~p3}_a1_i[9]	CA_a1[9]	-
wcs_{even/odd}_bit_{p0~p3}_a1_i[0]	CS_a1[0]	-
wcs_{even/odd}_bit_{p0~p3}_a1_i[1]	CS_a1[1]	-
wctrl_{even/odd}_bit_{p0~p3}_a1_i[0]	CTRL_a1[0]	RESET#
wctrl_{even/odd}_bit_{p0~p3}_a1_i[1]	CTRL_a1[1]	RAS#
wctrl_{even/odd}_bit_{p0~p3}_a1_i[2]	CTRL_a1[2]	CAS#
wctrl_{even/odd}_bit_{p0~p3}_a1_i[3]	CTRL_a1[3]	WE#
wck_{even/odd}_bit_{p0~p3}_a1_i	CK_t_a1	-
	CK_c_a1	-

2.5 Data Byte Map

2.5.1 LPDDR4 Dual Channel Mode

Table 2-7. LPDDR4 Dual Channel DFIW-to-DRAM Data Pin Map

DFIW Signal	Bump Name	LPDDR4 CHA
wdq_{even/odd}_bit_{p0~p3}_b{n}_i[0]	DQ_b{n}[0]	DQ [{n*8}+0]_A
wdq_{even/odd}_bit_{p0~p3}_b{n}_i[1]	DQ_b{n}[1]	DQ [{n*8}+1]_A
wdq_{even/odd}_bit_{p0~p3}_b{n}_i[2]	DQ_b{n}[2]	DQ [{n*8}+2]_A
wdq_{even/odd}_bit_{p0~p3}_b{n}_i[3]	DQ_b{n}[3]	DQ [{n*8}+3]_A
wdq_{even/odd}_bit_{p0~p3}_b{n}_i[4]	DQ_b{n}[4]	DQ [{n*8}+4]_A
wdq_{even/odd}_bit_{p0~p3}_b{n}_i[5]	DQ_b{n}[5]	DQ [{n*8}+5]_A
wdq_{even/odd}_bit_{p0~p3}_b{n}_i[6]	DQ_b{n}[6]	DQ [{n*8}+6]_A
wdq_{even/odd}_bit_{p0~p3}_b{n}_i[7]	DQ_b{n}[7]	DQ [{n*8}+7]_A
wdm_{even/odd}_bit_{p0~p3}_b{n}_i[8]	DMI_b{n}	DMI[{n*8}+0]_A
wdqs_{even/odd}_bit_{p0~p3}_b{n}_i	DQS_t_b{n}	DQS[n]_t_A
	DQS_c_b{n}	DQS[n]_c_A
DFIW Signal	Bump Name	LPDDR4 CHB
wdq_{even/odd}_bit_{p0~p3}_b{y}_i[0]	DQ_b{y}[0]	DQ [{y*8}+0]_B
wdq_{even/odd}_bit_{p0~p3}_b{y}_i[1]	DQ_b{y}[1]	DQ [{y*8}+1]_B
wdq_{even/odd}_bit_{p0~p3}_b{y}_i[2]	DQ_b{y}[2]	DQ [{y*8}+2]_B
wdq_{even/odd}_bit_{p0~p3}_b{y}_i[3]	DQ_b{y}[3]	DQ [{y*8}+3]_B
wdq_{even/odd}_bit_{p0~p3}_b{y}_i[4]	DQ_b{y}[4]	DQ [{y*8}+4]_B
wdq_{even/odd}_bit_{p0~p3}_b{y}_i[5]	DQ_b{y}[5]	DQ [{y*8}+5]_B
wdq_{even/odd}_bit_{p0~p3}_b{y}_i[6]	DQ_b{y}[6]	DQ [{y*8}+6]_B
wdq_{even/odd}_bit_{p0~p3}_b{y}_i[7]	DQ_b{y}[7]	DQ [{y*8}+7]_B

DFIW Signal	Bump Name	LPDDR4 CHB
wdm_{even/odd}_bit_{p0~p3}_b{y}_i[8]	DMI_b{y}	DMI[{y}*8]+0_B
wdqs_{even/odd}_bit_{p0~p3}_b{y}_i	DQS_t_b{y}	DQS[y]_t_B
	DQS_c_b{y}	DQS[y]_c_B

Notes:

1. N means the number of Dbyte used in DDR; n = 0 ~ (N-Y-1).
2. Y means the number of Dbyte used in CHA; y = (N-Y) ~ (N-1).

2.5.2 Other Modes

Table 2-8. LPDDR4 Single Channel/LPDDR3/DDR4/DDR3 DFIW-to-DRAM Data Pin Map

DFIW Signal	Bump Name	DDR*/LPDDR*
wdq_{even/odd}_bit_{p0~p3}_b{n}_i[0]	DQ_b{n} [0]	DQ [{n*8}+0]
wdq_{even/odd}_bit_{p0~p3}_b{n}_i[1]	DQ_b{n} [1]	DQ [{n*8}+1]
wdq_{even/odd}_bit_{p0~p3}_b{n}_i [2]	DQ_b{n} [2]	DQ [{n*8}+2]
wdq_{even/odd}_bit_{p0~p3}_b{n}_i [3]	DQ_b{n} [3]	DQ [{n*8}+3]
wdq_{even/odd}_bit_{p0~p3}_b{n}_i [4]	DQ_b{n} [4]	DQ [{n*8}+4]
wdq_{even/odd}_bit_{p0~p3}_b{n}_i [5]	DQ_b{n} [5]	DQ [{n*8}+5]
wdq_{even/odd}_bit_{p0~p3}_b{n}_i [6]	DQ_b{n} [6]	DQ [{n*8}+6]
wdq_{even/odd}_bit_{p0~p3}_b{n}_i [7]	DQ_b{n} [7]	DQ [{n*8}+7]
wdm_{even/odd}_bit_{p0~p3}_b{n}_i [8]	DMI_b{n}	DM {{n*8}+0}
wdqs_{even/odd}_bit_{p0~p3}_b{n}_i	DQS_t_b{n}	DQS[n]_t
	DQS_c_b{n}	DQS[n]_c

Note: N means the number of Dbyte used in DDR; n = 0 ~ (N-1).

Chapter 3

PHY Initialization

This chapter contains the following sections:

- 3.1 Overview
- 3.2 PLL Description
- 3.3 ZQ Calibration
- 3.4 PHY Initialization Sequence

3.1 Overview

Prior to entering any mission modes, all of the PHY blocks must be at Normal state. This chapter details all the settings of PHY initialization period including PLL setting and ZQ calibration.

3.2 PLL Description

In PHY, it should generate the clock for DRAM interface and PHY core logic (i.e. PHYCLK and DfiClk). The PLLs are embedded in CPHY by using two Faraday's PLL IPs, "PLL361" and "PLL365". The clk output of PLL361 connects to PLL365 and the highest frequency comes from the clk output of PLL365.

Figure 3-1 shows the connection relationship between PLL361 and PLL365.

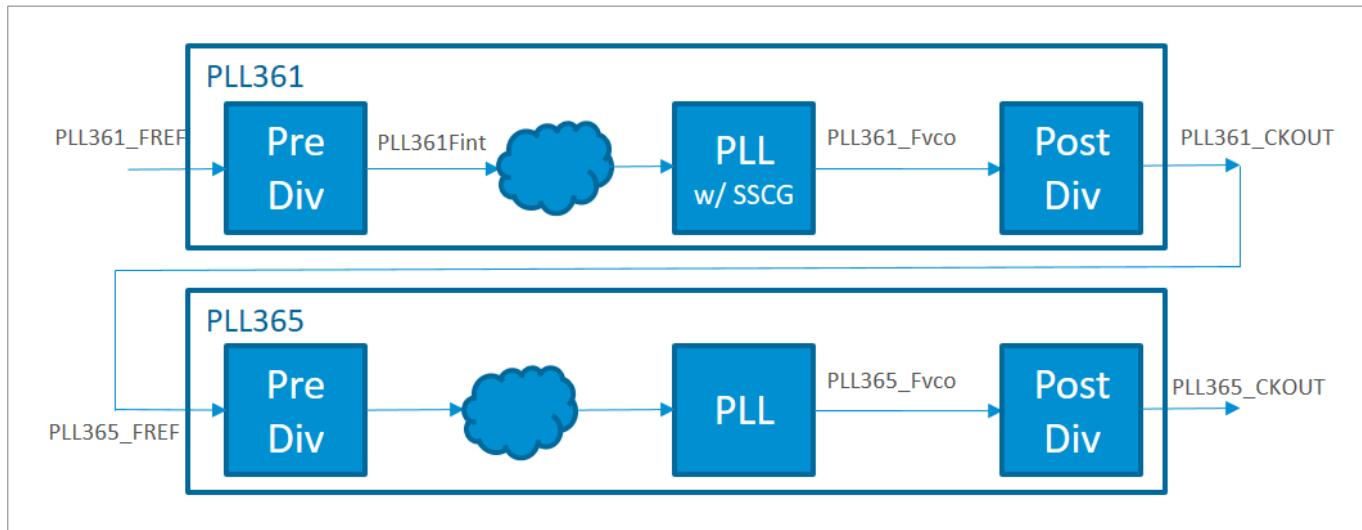


Figure 3-1. Connection Relationship between PLL361 and PLL365

3.2.1 PLL Setting

Table 3-1 lists the specification of each PLL parameter. Each parameter is limited within the range provided in table.

Table 3-1. PLL Specification

Signal	Specification (MHz)
PLL361	
PLL361_FREF	10 ~ 50
PLL361_Fint	10 ~ 20
PLL361_Fvco	900 ~ 1800
PLL361_CKOUT	37.5 ~ 1800
PLL365	
PLL365_FREF	75 ~ 150
PLL365_Fvco	2400 ~ 4800
PLL365_CKOUT	9.375 ~ 2400

Formula:

- $\text{PLL361_Fint} = \text{PLL361_FREF}/\text{PLL361_MS}$
- $\text{PLL361_Fvco} = \text{PLL361_Fint} * \text{PLL361_NF} / 2^{15}$
- $\text{PLL361_CKOUT} = \text{PLL361_Fvco}/\text{PLL361_PS}$
- When spreading frequency function enable:
 - Spreading frequency: $\text{Fssc} = \text{Fint}/2/\text{PLL361_NS}$
 - Modulation depth: $2D = \text{PLL361_NV} * \text{PLL361_NS}/\text{PLL361_NF}$
- $\text{PLL365_FREF} = \text{PLL361_CKOUT}$
- $\text{PLL365_Fvco} = \text{PLL365_FREF} * \text{PLL365_NS} * 2$
- $\text{PLL365_CKOUT} = \text{PLL365_Fvco}/\text{PLL365_PS}$

3.2.2 PLL Example

For DDR4-1600, 800 MHz needs to be generated from PLL365. The following examples are used as guiding for how to generate the target frequency. PLL361_FREF is 50 MHz and the settings of PLL in CPHY are as shown in **Table 3-2** for generating the 800 MHz on PLL365_CKOUT:

Table 3-2. PLL Setting for Generating 800 MHz without SSC Enable

Block	Offset	Bit	Register Name	Value
CPHY	0x0050	17:16	PLL361_MS 'h0 = Invalid 'h1 = /1 'h2 = /2 'h3 = /3	0x3
		15:12	PLL361_PS 'h0 = Invalid 'h1 = /1 'h2 = /2 'h3 = /3 'h4 = Invalid 'h5 = /2 'h6 = /4 'h7 = /6 'h8 = Invalid 'h9 = /4 'ha = /8 'hb = /12 'hc = Invalid 'hd = /8 'he = /16 'hf = /24	0xb
		11:0	PLL361_NS	0x48
0x0070		29:28	PLL361_Type	0x0
		27	PLL361_FIR	0x0
		26	PLL361_IS	0x0
		25	PLL361_MP	0x0
		24	PLL361_SSEN	0x0
		23:0	PLL361_NF	0x240000
0x0074		23:0	PLL361_NV	0x0

Block	Offset	Bit	Register Name	Value
	0x0054	25:24	PLL365_IS	0x0
		20:16	PLL365_NS	0x10
		14:12	(FSP3) PLL365_PS 'h0 = /2 'h1 = /4 'h2 = /8 'h3 = /16 'h4 = /32 'h5 = /64 'h6 = /128 'h7 = /256	0x3
		10:8	(FSP2) PLL365_PS	0x2
		6:4	(FSP1) PLL365_PS	0x1
		2:0	(FSP0) PLL365_PS	0x1

- For FSPO's PLL365_CKOUT is 1600.32 MHz:
 - (Meet the specification) PLL361_Fint = 50/3 MHz
 - (Meet the specification) PLL361_Fvco = $(50/3) * 2359296/2^{15} = 1200$ MHz
- PLL361_NF= 0x240000 (hex) = 23592696 (dec)
 - (Meet the specification) PLL361_CKOUT = $1200.24/12 = 100$ MHz = PLL365_FREF
 - (Meet the specification) PLL365_Fvco = $100 * 16 * 2 = 3200$ MHz
- PLL365_NS = 0x10 (hex) = 16 (dec)
 - (Meet the specification) For FSPO's PLL365_CKOUT = $3200/4 = 800$ MHz
 - (Meet the specification) For FSP1's PLL365_CKOUT is 800 MHz.
 - (Meet the specification) For FSP2's PLL365_CKOUT is 400 MHz.
 - (Meet the specification) For FSP3's PLL365_CKOUT is 200 MHz.

3.3 ZQ Calibration

ZQ calibration is used to calibrate DDR-PHY Ron & ODT values and will be performed at initialization automatically. Users can restart the ZQ calibration after observing the changes of the dramatic environment condition. For more details of restarting ZQ calibration, please refer to Section 3.3.1.

Since the DDR-PHY supports 4 kinds of DRAM types, please follow **Table 3-3** and **Table 3-4** to set related registers and choose the correct Resistance of RZQ for different applications. **RZQ should be connected between "VCC12IO_DDR" and "comp_pad_o"** which is a pad in CPHY, and the resistance should be within $\pm 1\%$.

Table 3-3. CPHY Register Setting of ZQ Calibration for Different DRAM Applications

Register Offset	Register Bit	Register Bit Name	DDR4	LPDDR4 (VOH = VDDQ/3)	LPDDR4 (VOH = VDDQ/2.5)	DDR3	LPDDR3
0x0040	0	sel	0x1	0x1	0x1	0x1	0x1
	5	ddr4md1	0x0	0x0	0x0	0x0	0x0
	4	ddr4md0	0x1	0x0	0x0	0x0	0x0
	3	two_run	0x1	0x0	0x0	0x0	0x0
	2:1	fs	2'b00: 0 Mbps < Datarate ≤ 400 Mbps 2'b01: 400 Mbps < Datarate ≤ 800 Mbps 2'b10: 800 Mbps < Datarate ≤ 1600 Mbps 2'b11: 1600 Mbps < Datarate ≤ 3200 Mbps				
	22:20	pronmd1	0x3	0x3	0x3	0x3	0x3
	18:16	nronmd1	0x3	0x3	0x3	0x3	0x3
	14:12	pronmd0	0x3	0x1	0x1	0x1	0x1
	10:8	nronmd0	0x0	0x3	0x2	0x1	0x1
	24:16	vref_code1	0x100	0x100	0x100	0x100	0x100
	8:0	vref_code0	0x199	0x0AA	0x0CC	0x100	0x100

Table 3-4. Resistance of RZQ in Different DRAM Applications

Symbol	DDR4	LPDDR4	DDR3	LPDDR3
RZQ	60 Ω	120 Ω	120 Ω	120 Ω

3.3.1 Restart ZQ calibration

Before triggering Restart ZQ calibration, please make sure that DDRPHY subsystem is at idle state which means that there is no any read/write quote.

Below sequences show users how to restart ZQ calibration:

1. Make sure that DDRPHY subsystem is at idle state.
2. Set CPHY's PHY_OVR3::zq_cal_req = 'b11 to trigger ZQ calibration.
3. Wait at least 30 μs
4. Set CPHY's PHY_OVR3::zq_cal_req = 'b0 to disable override control.
5. ZQ re-calibration has been done

3.3.2 Bypass ZQ Calibration

Below sequences show user how to bypass ZQ calibration. User can set PLEG and NLEG manually:

1. Set CPHY's PLEG[5:0] (pull-up resistance control) and NLEG[5:0] (pull-down resistance control)

Table 3-5. Pull-Up/Pull-Down Resistance Control Setting

Register Bit	Register Bit	Register Bit Name
0x48	29:24	dip_aphy
	21:16	din_aphy
	13:8	dip_dphy
	5:0	din_dphy

2. Set CPHY's ZQ_CAL_CTRL0: sel = 'h0 to force ZQ calibrator use user-defined value.
3. Set CPHY's PHY_OVR3: zq_cal_req = 'b11 to trigger ZQ calibration with bypass.
4. Wait at least 10 refclk cycles
5. Set CPHY's PHY_OVR3: zq_cal_req = 'b0 to disable override control.
6. Bypass ZQ calibration has been done.

3.4 PHY Initialization Sequence

The sequence in this section guides user to initialize PHY during Pre-layout and Post-layout simulation and silicon test. Before PHY initialization is triggered, there are several types of register groups to be set up, such as ZQ calibration, PLL, wckby2 logic, FIFO reset, and ADDLL logic.

Step 1: Set “wby2ck setting” of the PHY block depending on the floorplan used by user; the settings of wby2ck_sync_phase_sel, wby2ck_sync_auto, and wby2ck_sync_source_sel must be set correctly. Please refer to Section 5.3.2.1 for more details on setting these registers.

In APHY:

Table 3-6. APHY wby2ck Setting

Offset	Bit	Register Name
0x0060	[7]	wby2ck_sync_phase_sel
	[6]	wby2ck_sync_auto
	[5]	wby2ck_sync_source_sel

In DPHY:

Table 3-7. DPHY wby2ck Setting

Offset	Bit	Register Name
0x0064	[2]	wby2ck_sync_phase_sel
	[1]	wby2ck_sync_auto
	[0]	wby2ck_sync_source_sel

In CPHY:

Table 3-8. CPHY wby2ck Setting

Offset	Bit	Register Name
0x0060	[6]	wby2ck_sync_phase_sel
	[5]	wby2ck_sync_auto
	[4]	wby2ck_sync_source_sel

Step 2: Set “EFIFO setting” of the PHY block depending on the floorplan used by user; the settings of wptra_rst_sel, rptr_rst_sel, and sync_pos must be set correctly. Please refer to Section 5.3.2.2 for more details on setting these registers.

In APHY:

Table 3-9. APHY EFIFO Setting

Offset	Bit	Register Name
0x0070	[12:8]	wptr_RST_SEL
	[7:4]	rptr_RST_SEL
	[1:0]	Sync_pos

In DPHY:

Table 3-10. DPHY EFIFO Setting

Offset	Bit	Register Name
0x0074	[12:8]	wptr_RST_SEL
	[7:4]	rptr_RST_SEL
	[1:0]	Sync_pos

Step 3: Set “PHYCK SKEW setting” of the PHY block depending on the floorplan used by user. The setting of

`phyck_skewcode` must be set correctly. Please refer to Section 5.1 for more details on setting these registers.

Table 3-11. APHY phyck_skew Setting

Offset	Bit	Register Name
0x0064	[23:20]	phyck_skewcode

In DPHY:

Table 3-12. DPHY phyck_skew Setting

Offset	Bit	Register Name
0x0064	[11:8]	phyck_skewcode

In CPHY:

Table 3-13. CPHY phyck_skew Setting

Offset	Bit	Register Name
0x0060	[15:12]	phyck_skewcode

Step 4: Set CPHY block's registers related to ZQ calibration, the settings in ZQ_CAL_CTRL0 and ZQ_CAL_CTRL1 must be set as user's application scenario (e.g. DDR4 mode and DDR3 mode, etc.) Please refer to Section 3.3 for more details on setting these registers.

In CPHY:

Table 3-14. CPHY ZQ Calibration Setting

Offset	Bit	Register Name
0x0040	[22:20]	pronmd1
	[18:16]	nronmd1
	[14:12]	pronmd0
	[10:8]	nronmd0
	[5]	ddr4md1
	[4]	ddr4md0
	[3]	two_run
	[2:1]	fs
	0	sel
0x0044	[24:16]	vref_code1
	[8:0]	vref_code0

Step 5: In CPHY: Set CPHY block's registers related to PLL, the settings in PLL_CTRL0, PLL_CTRL1, PLL_CTRL2, and PLL_CTRL3 must be set as user's target frequency. Please refer to Section 3.2 for more details on setting these registers.

Table 3-15. CPHY PLL Setting

Offset	Bit	Register Name
0x0050	17:16	PLL361_MS
	15:12	PLL361_PS
	11:0	PLL361_NS
0x0070	29:28	PLL361_Type
	27	PLL361_FIR
	26	PLL361_IS
	25	PLL361_MP
	24	PLL361_SSEN
	23:0	PLL361_NF
	23:0	PLL361_NV
0x0074	25:24	PLL365_IS
	20:16	PLL365_NS
	14:12	(FSP3) PLL365_PS
	10:8	(FSP2) PLL365_PS
	6:4	(FSP1) PLL365_PS
	2:0	(FSPO) PLL365_PS

Step 6: (It is set only in simulation, do not set in silicon test.) Set PHY block's registers related to DCD for reducing simulation time. The setting of master_bypass can be set as 1'h1 during simulation to bypass DCD function. In silicon test, please use the default setting.

In APHY:

Table 3-16. APHY DCD Setting

Offset	Bit	Register Name
0x00AC	24	Master_bypass

In DPHY:

Table 3-17. DPHY DCD Setting

Offset	Bit	Register Name
0x00AC	24	Master_bypass

Step 7: (It is set only in simulation, do not set in silicon test.) Set PHY block's registers related to ADDLL for reducing simulation time. The setting of registers can be set 'h10 during simulation to reduce ADDLL locking time. In silicon test, please use the default setting.

In APHY:

Table 3-18. APHY ADDLL Wait Time Setting

Offset	Bit	Register Name
0x0030	[15:0]	reset_to_norm_wait1_val
0x0050	[31:16]	lp3_to_norm_wait2_val
0x0054	[31:16]	freq_change_wait1_val

In DPHY:

Table 3-19. DPHY ADDLL Wait Time Setting

Offset	Bit	Register Name
0x0024	[15:0]	reset_to_norm_wait3_val
0x0040	[15:0]	lp3_to_norm_wait2_val
0x0048	[15:0]	freq_change_wait1_val

Step 8: Set “soft_resetn” of the PHY block’s to ‘h1 to initialize each PHY block.

In APHY:

Table 3-20. APHY soft_resetn Setting

Offset	Bit	Register Name
0x0084	0	soft_resetn

In DPHY:

Table 3-21. DPHY soft_resetn Setting

Offset	Bit	Register Name
0x0084	0	soft_resetn

In CPHY:

Table 3-22. CPHY soft_resetn Setting

Offset	Bit	Register Name
0x0084	0	soft_resetn

Step 9: Wait about 7000 refclk cycle (at 50 MHz) for all the initialization of PHY blocks are done. When PHY initialization has been done and all the PHY blocks are at normal state, then it means that this PHY is ready to use.

3.4.1 PHY Initialization Example

The section shows examples of full initialization sequences of PHY under the condition of DDR4-1600, 16-bit.

- Floorplan



Figure 3-2. Floorplan Example

- Wby2ck Connection
 - Please refer to Section 5.3.2.1 for more details.

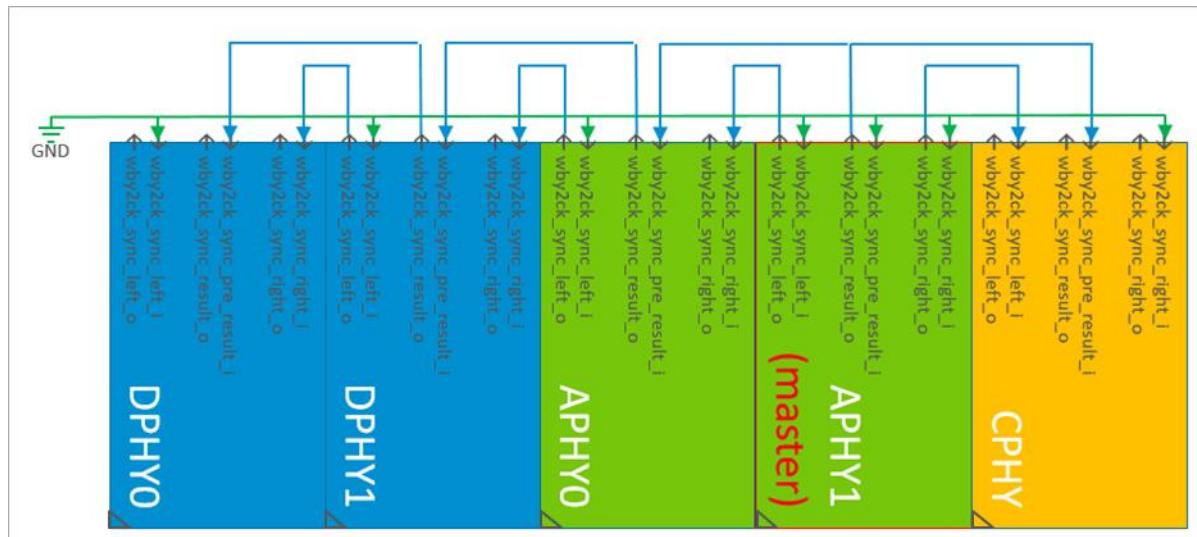


Figure 3-3. wby2ck Signal Connection Example

- Rptr_rstn Connection
 - Please refer to Section 5.3.2.2 for more details.

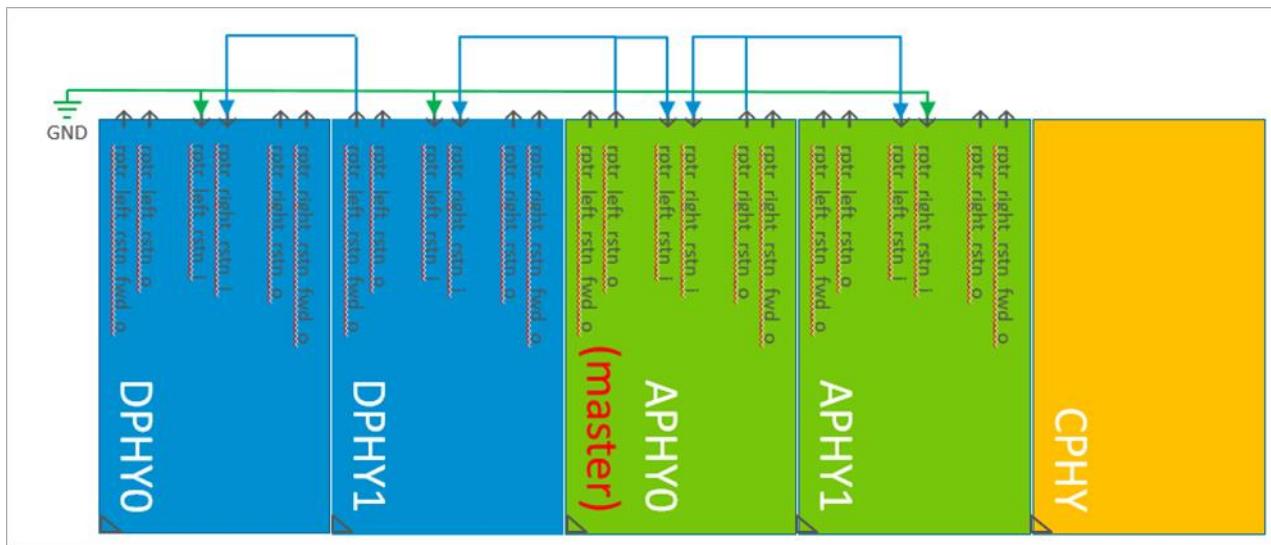


Figure 3-4. EFIFO Signal Connection Example

- Full Initialization Sequence

Table 3-23. PHY Initialization Sequence

Step	PHY Block	Offset	Value
Step 1	CPHY	0x0100	0x0000_0220
	APHY0	0x0100	0x0080_77c0
	APHY1	0x0100	0x0080_7740
	DPHY0	0x0064	0x0000_0003
	DPHY1	0x0064	0x0000_0003
Step 2	APHY0	0x0070	0x0000_0501
	APHY1	0x0070	0x0000_0501
	DPHY0	0x0074	0x0000_0501
	DPHY1	0x0074	0x0000_0531
Step 3	CPHY	0x0040	0x0033_301f
	CPHY	0x0044	0x0100_0199

Step	PHY Block	Offset	Value
Step 4	CPHY	0x0050	0x0003_b048
	CPHY	0x0070	0x0024_0000
	CPHY	0x0074	0x0000_0000
	CPHY	0x0054	0x0010_3211
Step 5	APHY0	0x017c	0x0100_ff37
	APHY1	0x017c	0x0100_ff37
	DPHY0	0x017c	0x0100_ff37
	DPHY1	0x017c	0x0100_ff37
Step 6	APHY0	0x0030	0x0003_0010
	APHY0	0x0050	0x0010_0003
	APHY0	0x0054	0x0010_0003
	APHY1	0x0030	0x0003_0010
	APHY1	0x0050	0x0010_0003
	APHY1	0x0054	0x0010_0003
	DPHY0	0x0024	0x0003_0010
	DPHY0	0x0040	0x0005_0010
	DPHY0	0x0048	0x0000_0010
	DPHY1	0x0024	0x0003_0010
	DPHY1	0x0040	0x0005_0010
	DPHY1	0x0048	0x0000_0010
	CPHY	0x0098	0x0003_4000
	CPHY	0x0098	0x0000_0000

Step	PHY Block	Offset	Value
Step 8	CPHY	0x0084	0x0000_0003
	APHY0	0x0084	0x0000_0003
	APHY1	0x0084	0x0000_0003
	DPHY0	0x0084	0x0000_0003
	DPHY1	0x0084	0x0000_0003
Step 9	Wait 7000 refclk cycles (At 50 MHz)		
	PHY initialization has been done.		

Chapter 4

PHY Testing

This chapter contains the following sections:

- 4.1 Overview
- 4.2 Scan Test
- 4.3 Loopback Test
- 4.4 I/O DC Test Mode
- 4.5 Mass Production Test Pattern

4.1 Overview

The DDR PHY is equipped with a comprehensive suite of on-chip test features designed to ensure thorough coverage of the entire macro. These features are well-suited for production testing. The following features are provided:

- Loop back mode with IO and without IO
 - Default training 16-bit pattern generator and receiver for each of APHY and DPHY DDR I/O bits
 - 7-bit pseudorandom (i.e. PRBS7) pattern generator and receiver for each of APHY and DPHY DDR I/O bits
 - Built-in-self-test (BIST) utilizing the internal loopback mode, transmit pattern generator and receive error detector
- Supports for DFT scan testing in both stuck-at mode and at-speed mode
- I/O DC tests such as IDDQ Test, Leak Test, Tristate Test, Ron Test, and so on
- Devoted analog test pad for DC voltage
- 12-bit wide digital test bus for digital layer characterization and debug
- APB/I²C slave interface inside each PHY block for programming the CSRs (control and status registers)

4.2 Scan Test

Each PHY block has been embedded with scan chain for improving the quality of die during the CP/FT test. There are five patterns in each PHY block as shown in **Table 4-1**. The coverage rate of each PHY block is shown in

Table 4-2.

The pattern of scan_chain is used to test the function of scan chain:

- The pattern of SSA is low-frequency test for testing if there exists the fault of “stuck at fault” in function logics.
- The pattern of OCC is at speed test for testing the transitional fault in function logics.

All patterns in each PHY block must be planned in user’s DDR subsystem and be tested during CP/FT to pick out the bad die for the improvement of die quality.

Table 4-1. Scan Patterns

Pattern Name	Provider	Comment	Pass Condition
SSA.ftl	Faraday	Test the digital circuits	Compare with VSIA scan in and scan out value
SSA_sample.ftl	Faraday	Test the digital circuits	Compare with VSIA scan in and scan out value
scan_chain.ftl	Faraday	For the scan chain test	Compare with VSIA scan in and scan out value
OCC_tran.ftl	Faraday	Test the digital circuits	Compare with VSIA scan in and scan out value

Pattern Name	Provider	Comment	Pass Condition
occ_sample.ftl	Faraday	Test the digital circuits	Compare with VSIA scan in and scan out value

Table 4-2. Coverage Rate

Coverage Rate	SSA	OCC
FXLPDDR4CMSC100HJOP	99.54%	92.81%
FXLPDDR4CMSC100EWHJOP	99.60%	91.82%
FXLPDDR4CMSC100NSHJOP	99.06%	93.78%
FXLPDDR4CMSC100EWHJOP	99.06%	92.31%
FXLPDDR4CMSC100NSHJOP	99.20%	93.22%
FXLPDDR4CMSC100EWHJOP	99.18%	93.02%
FXLPDDR4CMSC100HJOP	99.20%	93.22%

4.3 Loopback Test

4.3.1 Introduction

Each APHY and DPHY has built-in internal loopback path. This loopback mode is used to test the DPHY block and the APHY block of the LPDDR4/LPDDR3/DDR4/DDR3 Combo PHY. When enabled, the loopback module automatically generates the test patterns for the loopback test. The test data is fed to the transmit path in the DPHY and APHY. This data traverses through the entire transmit path and finally loops back to the receive path in the DDR-IOs. This data then travels through the entire receive path reaches the loopback module. The loopback module will then compare the looped back data with the original test data. The compared results are stored in registers when all test patterns are compared.

Figure 4-1 shows the internal loopback mode path along with the loopback module. The Loopback Test can be performed in the APHY and DPHY in parallel.

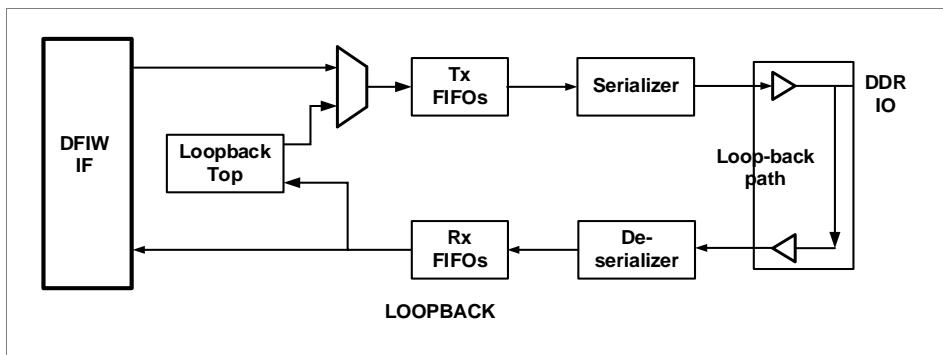


Figure 4-1. Loopback Path with Loopback Module

4.3.2 Loopback Test Sequence

Before starting the Loopback Test, all of the PHYs have to be brought to Normal/Ready state. Once all the PHYs are at Ready state, the test can be started. The Loopback Test is performed at two phases. The first phase is called the training phase. At this phase, default training patterns are sent out from the Loopback Top module. The same patterns are also expected by the receiver in the Loopback Top. For accurate loopback testing, the sampler within the analog circuitry must correctly capture the looped-back data. This ensures that the transmitted data and the data received by the Loopback Top module match. To achieve this precise sampling, the clock phase delay at the sampler input is adjusted until the optimal sampling point is determined.

The second phase is the PRBS phase. After the training is done, at this phase, PRBS-7 patterns are generated by the PRBS generator modules inside the Loopback Top for each of the APHY and DPHY DDR I/O bits. These PRBS patterns travel through the entire path and are looped back to the PRBS checker modules inside the Loopback Top and are compared. After the comparison is done, error status is stored in the register.

4.3.2.1 Loopback Training Sequence

Users are suggested to follow the below sequences for performing loopback training:

1. Bring all the PHYs to Normal/Ready state by following the initialization sequences. Please refer to Section 3.4.1 for more details.
2. Enable the short loopback path inside analog PHY by setting APHY::LB_CTRL19::sloop_en and DPHY::LB_CTRL12::sloop_en to 1'b1 to enable short path loopback. Meanwhile, setting APHY::LB_CTRL19::long_loopback_en and DPHY::LB_CTRL12::long_loopback_en to 1'b0

3. Enable the PRBS pattern test by setting APHY::LB_CTRL19::prbs_en and DPHY::LB_CTRL12::prbs_en to 1'b1. PRBS pattern test will start after loopback training is done.
4. Enable the loopback test by setting APHY::LB_CTRL19::loopback_en and DPHY::LB_CTRL12::loopback_en to 1'b1
5. Monitor APHY::PHY_DEBUG_STS5::lb_training_done/ DPHY::PHY_DEBUG_STS12::lb_training_done signal. When lb_training_done goes high, user can monitor APHY::PHY_DEBUG_STS5::lb_training_err_status and DPHY::PHY_DEBUG_STS12::lb_training_err_status to check the loopback phase training result.

Table 4-3 ~ Table 4-8 are the details of registers.

In APHY:

Table 4-3. APHY Loopback Training Done Observation

Offset	Bit	Register Name
0x0218	[23]	lb_training_done

Table 4-4. APHY Loopback Training Phase Observation

Offset	Bit	Register Name
0x0218	[27:24]	lb_training_err_status
bit[0] = Left boundary even bit is not found. bit[1] = Left alignment is failed. bit[2] = Left boundary odd bit is not found. bit[3] = Right boundary is not found.		

In DPHY:

Table 4-5. DPHY Loopback Training Done Observation

Offset	Bit	Register Name
0x0234	[31]	lb_training_done

Table 4-6. DPHY Loopback Training Phase Observation

Offset	Bit	Register Name
0x0114	[22:18]	lb_training_err_status
LB Training FSM error status bit[0] = Left boundary even bit is not found. bit[1] = Left alignment even bit is failed. bit[2] = Left boundary odd bit is not found. bit[3] = Right boundary even bit is not found. bit[4] = Right boundary odd bit is not found.		

6. After loopback training is done, PRBS pattern test will release automatic execution. PRBS pattern length depends on the setting in APHY::LB_CTRL20::run_prbs_len and DPHY::LB_CTRL13::run_prbs_len.
7. After PRBS pattern test is done, disable the loopback test by setting APHY::LB_CTRL19::loopback_en and DPHY::LB_CTRL12::loopback_en to 1'b0
8. User can monitor the result of PRBS test on each test bit by setting APHY::LB_CTRL21_prbs_err_count_sel and DPHY::LB_CTRL14_prbs_err_count_sel.
9. Read error count from APHY::LB_CTRL21::prbs_err_count and DPHY::LB_CTRL14::prbs_err_count
10. Repeat Step 9 and Step 10 until all the points to-be-observed are read out. **Table 4-7** and **Table 4-8** are the mapping tables of err_count_sel:

In APHY:

Table 4-7. APHY per bit PRBS Test Result Observation

Sel[6:0]	Register Name
7'd0	ca0 even bit at even pstate 0
7'd1	ca1 even bit at even pstate 0
7'd2	ca2 even bit at even pstate 0
7'd3	ca3 even bit at even pstate 0
7'd4	ca4 even bit at even pstate 0
7'd5	ca5 even bit at even pstate 0
7'd6	ca6 even bit at even pstate 0
7'd7	ca7 even bit at even pstate 0
7'd8	ca8 even bit at even pstate 0
7'd9	ca9 even bit at even pstate 0
7'd10	cs0 even bit at even pstate 0
7'd11	cs1 even bit at even pstate 0
7'd12	ctrl0 even bit at even pstate 0
7'd13	ctrl1 even bit at even pstate 0
7'd14	ctrl2 even bit at even pstate 0
7'd15	ctrl3 even bit at even pstate 0
7'd16	ca0 even bit at even pstate 1
7'd17	ca1 even bit at even pstate 1
7'd18	ca2 even bit at even pstate 1
7'd19	ca3 even bit at even pstate 1
7'd20	ca4 even bit at even pstate 1
7'd21	ca5 even bit at even pstate 1
7'd22	ca6 even bit at even pstate 1
7'd23	ca7 even bit at even pstate 1
7'd24	ca8 even bit at even pstate 1
7'd25	ca9 even bit at even pstate 1
7'd26	cs0 even bit at even pstate 1
7'd27	cs1 even bit at even pstate 1
7'd28	ctrl0 even bit at even pstate 1
7'd29	ctrl1 even bit at even pstate 1
7'd30	ctrl2 even bit at even pstate 1

Sel[6:0]	Register Name
7'd31	ctrl3 even bit at even pstate 1
7'd32	ca0 even bit at even pstate 2
7'd33	ca1 even bit at even pstate 2
7'd34	ca2 even bit at even pstate 2
7'd35	ca3 even bit at even pstate 2
7'd36	ca4 even bit at even pstate 2
7'd37	ca5 even bit at even pstate 2
7'd38	ca6 even bit at even pstate 2
7'd39	ca7 even bit at even pstate 2
7'd40	ca8 even bit at even pstate 2
7'd41	ca9 even bit at even pstate 2
7'd42	cs0 even bit at even pstate 2
7'd43	cs1 even bit at even pstate 2
7'd44	ctrl0 even bit at even pstate 2
7'd45	ctrl1 even bit at even pstate 2
7'd46	ctrl2 even bit at even pstate 2
7'd47	ctrl3 even bit at even pstate 2
7'd48	ca0 even bit at even pstate 3
7'd49	ca1 even bit at even pstate 3
7'd50	ca2 even bit at even pstate 3
7'd51	ca3 even bit at even pstate 3
7'd52	ca4 even bit at even pstate 3
7'd53	ca5 even bit at even pstate 3
7'd54	ca6 even bit at even pstate 3
7'd55	ca7 even bit at even pstate 3
7'd56	ca8 even bit at even pstate 3
7'd57	ca9 even bit at even pstate 3
7'd58	cs0 even bit at even pstate 3
7'd59	cs1 even bit at even pstate 3
7'd60	ctrl0 even bit at even pstate 3
7'd61	ctrl1 even bit at even pstate 3
7'd62	ctrl2 even bit at even pstate 3
7'd63	ctrl3 even bit at even pstate 3
7'd64	ca0 odd bit at odd pstate 0
7'd65	ca1 odd bit at odd pstate 0
7'd66	ca2 odd bit at odd pstate 0
7'd67	ca3 odd bit at odd pstate 0
7'd68	ca4 odd bit at odd pstate 0
7'd69	ca5 odd bit at odd pstate 0
7'd70	ca6 odd bit at odd pstate 0
7'd71	ca7 odd bit at odd pstate 0
7'd72	ca8 odd bit at odd pstate 0
7'd73	ca9 odd bit at odd pstate 0
7'd74	cs0 odd bit at odd pstate 0
7'd75	cs1 odd bit at odd pstate 0
7'd76	ctrl0 odd bit at odd pstate 0
7'd77	ctrl1 odd bit at odd pstate 0
7'd78	ctrl2 odd bit at odd pstate 0
7'd79	ctrl3 odd bit at odd pstate 0
7'd80	ca0 odd bit at odd pstate 1

Sel[6:0]	Register Name
7'd81	ca1 odd bit at odd pstate 1
7'd82	ca2 odd bit at odd pstate 1
7'd83	ca3 odd bit at odd pstate 1
7'd84	ca4 odd bit at odd pstate 1
7'd85	ca5 odd bit at odd pstate 1
7'd86	ca6 odd bit at odd pstate 1
7'd87	ca7 odd bit at odd pstate 1
7'd88	ca8 odd bit at odd pstate 1
7'd89	ca9 odd bit at odd pstate 1
7'd90	cs0 odd bit at odd pstate 1
7'd91	cs1 odd bit at odd pstate 1
7'd92	ctrl0 odd bit at odd pstate 1
7'd93	ctrl1 odd bit at odd pstate 1
7'd94	ctrl2 odd bit at odd pstate 1
7'd95	ctrl3 odd bit at odd pstate 1
7'd96	ca0 odd bit at odd pstate 2
7'd97	ca1 odd bit at odd pstate 2
7'd98	ca2 odd bit at odd pstate 2
7'd99	ca3 odd bit at odd pstate 2
7'd100	ca4 odd bit at odd pstate 2
7'd101	ca5 odd bit at odd pstate 2
7'd102	ca6 odd bit at odd pstate 2
7'd103	ca7 odd bit at odd pstate 2
7'd104	ca8 odd bit at odd pstate 2
7'd105	ca9 odd bit at odd pstate 2
7'd106	cs0 odd bit at odd pstate 2
7'd107	cs1 odd bit at odd pstate 2
7'd108	ctrl0 odd bit at odd pstate 2
7'd109	ctrl1 odd bit at odd pstate 2
7'd110	ctrl2 odd bit at odd pstate 2
7'd111	ctrl3 odd bit at odd pstate 2
7'd112	ca0 odd bit at odd pstate 3
7'd113	ca1 odd bit at odd pstate 3
7'd114	ca2 odd bit at odd pstate 3
7'd115	ca3 odd bit at odd pstate 3
7'd116	ca4 odd bit at odd pstate 3
7'd117	ca5 odd bit at odd pstate 3
7'd118	ca6 odd bit at odd pstate 3
7'd119	ca7 odd bit at odd pstate 3
7'd120	ca8 odd bit at odd pstate 3
7'd121	ca9 odd bit at odd pstate 3
7'd122	cs0 odd bit at odd pstate 3
7'd123	cs1 odd bit at odd pstate 3
7'd124	ctrl0 odd bit at odd pstate 3
7'd125	ctrl1 odd bit at odd pstate 3
7'd126	ctrl2 odd bit at odd pstate 3
7'd127	ctrl3 odd bit at odd pstate 3

In DPHY:

Table 4-8. DPHY per bit PRBS Test Result Observation

Sel[6:0]	Register Name
7'd1	rdq0 even bit at pstate 0
7'd2	rdq1 even bit at pstate 0
7'd3	rdq2 even bit at pstate 0
7'd4	rdq3 even bit at pstate 0
7'd5	rdq4 even bit at pstate 0
7'd6	rdq5 even bit at pstate 0
7'd7	rdq6 even bit at pstate 0
7'd8	rdq7 even bit at pstate 0
7'd9	rdq0 odd bit at pstate 0
7'd10	rdq1 odd bit at pstate 0
7'd11	rdq2 odd bit at pstate 0
7'd12	rdq3 odd bit at pstate 0
7'd13	rdq4 odd bit at pstate 0
7'd14	rdq5 odd bit at pstate 0
7'd15	rdq6 odd bit at pstate 0
7'd16	rdq7 odd bit at pstate 0
7'd17	rdm even bit at pstate 0
7'd18	rdm odd bit at pstate 0
7'd19	rdq0 even bit at pstate 1
7'd20	rdq1 even bit at pstate 1
7'd21	rdq2 even bit at pstate 1
7'd22	rdq3 even bit at pstate 1
7'd23	rdq4 even bit at pstate 1
7'd24	rdq5 even bit at pstate 1
7'd25	rdq6 even bit at pstate 1
7'd26	rdq7 even bit at pstate 1
7'd27	rdq0 odd bit at pstate 1
7'd28	rdq1 odd bit at pstate 1
7'd29	rdq2 odd bit at pstate 1
7'd30	rdq3 odd bit at pstate 1
7'd31	rdq4 odd bit at pstate 1
7'd32	rdq5 odd bit at pstate 1
7'd33	rdq6 odd bit at pstate 1
7'd34	rdq7 odd bit at pstate 1
7'd35	rdm even bit at pstate 1
7'd36	rdm odd bit at pstate 1
7'd37	rdq0 even bit at pstate 2
7'd38	rdq1 even bit at pstate 2
7'd39	rdq2 even bit at pstate 2
7'd40	rdq3 even bit at pstate 2
7'd41	rdq4 even bit at pstate 2
7'd42	rdq5 even bit at pstate 2
7'd43	rdq6 even bit at pstate 2
7'd44	rdq7 even bit at pstate 2
7'd45	rdq0 odd bit at pstate 2
7'd46	rdq1 odd bit at pstate 2
7'd47	rdq2 odd bit at pstate 2
7'd48	rdq3 odd bit at pstate 2
7'd49	rdq4 odd bit at pstate 2

Sel[6:0]	Register Name
7'd50	rdq5 odd bit at pstate 2
7'd51	rdq6 odd bit at pstate 2
7'd52	rdq7 odd bit at pstate 2
7'd53	rdm even bit at pstate 2
7'd54	rdm odd bit at pstate 2
7'd55	rdq0 even bit at pstate3
7'd56	rdq1 even bit at pstate3
7'd57	rdq2 even bit at pstate3
7'd58	rdq3 even bit at pstate3
7'd59	rdq4 even bit at pstate3
7'd60	rdq5 even bit at pstate3
7'd61	rdq6 even bit at pstate3
7'd62	rdq7 even bit at pstate3
7'd63	rdq0 odd bit at pstate3
7'd64	rdq1 odd bit at pstate3
7'd65	rdq2 odd bit at pstate3
7'd66	rdq3 odd bit at pstate3
7'd67	rdq4 odd bit at pstate3
7'd68	rdq5 odd bit at pstate3
7'd69	rdq6 odd bit at pstate3
7'd70	rdq7 odd bit at pstate3
7'd71	rdm even bit at pstate3
7'd72	rdm odd bit at pstate3

4.4 I/O DC Test Mode

IO DC test mode is provided to check the static behavior of analog PHY through primary inputs. This helps to debug for any existed functional issues with IO operation; moreover, it is also used to qualify IP's in mass production test environments.

- IDDQ test
 - This is used to check static leakage currents through all supplies.
- I/O Tristate test
 - This is used to check input pin leakage by tristate IO pads.
- I/O DC parameter test
 - Driver Strength Ron check without ZQ calibration
 - I/O Driver Ron check with ZQ calibration
 - I/O ODT Rtt check with ZQ calibration and I/O Receiver V_{il}/V_{ih} check

It is recommended enabling this mode after CPHY initialization done if ZQ calibration is needed. There are separated and independent control pins for each PHY to control DC test mode. There is an option to bypass ZQ calibration and feed driver manual code to IOs by internal register. Please refer to Section 3.3 for more details.

To reduce pin count at ASIC level, Faraday has implemented NAND structure to measure V_{il}/V_{ih} using

minimum set of internal signals. It is recommended that users following steps as mentioned in Section 4.4.1 ~ Section 4.4.5 to perform I/O DC test.

Figure 4-2 and Figure 4-3 depict the wiring relationship of IO control in APHY and DPHY.

Table 4-9 and **Table 4-11** list the pin descriptions of the I/O buffer when the FXLPDDR4CMSC100*HJOP block and FXLPDDR4CMSC100*HJOP block are in the test mode.

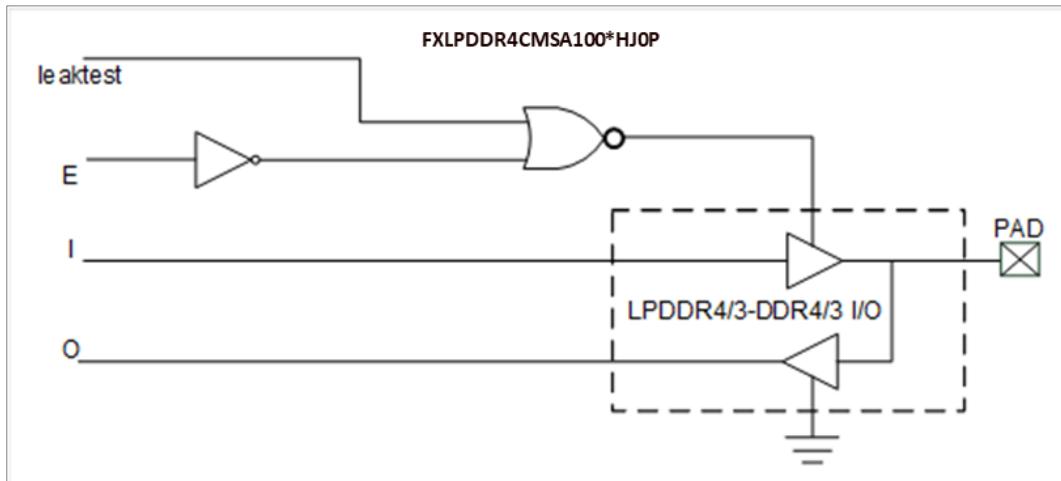


Figure 4-2. APHY I/O Configuration in Test Mode

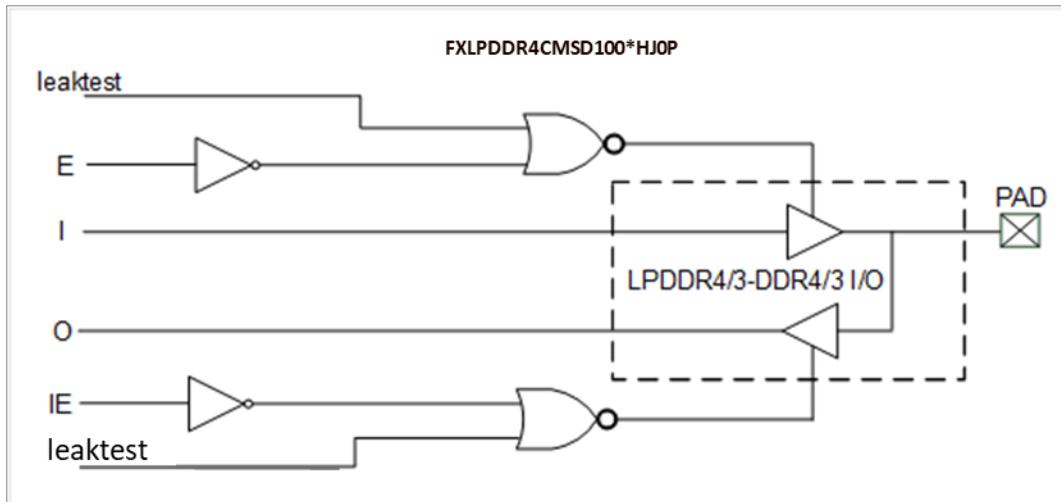


Figure 4-3. DPHY I/O Configuration in Test Mode

*Table 4-9. Pin Descriptions of Output Buffer when FXLPDDR4CMSA100*HJOP in Test Mode*

IP	I/O Pad	I	E	O	Note
FXLPDDR4CMSA100*HJOP (dctest_mode_i = '1')	CA[9:0]	dctest_tx_i	dctest_tx_en_i	Don't care	When leaktest_i = 0 and dctest_tx_en_a*_i = 1, all IO Pads will follow the input dctest_tx_a*_i.
	CK_T/CK_C				
	CS[1:0]				This configuration is used to test "Ron" and "Static behaviors of IO Driver".
	CTRL[3:0]				

*Table 4-10. Pin Descriptions of Output Buffer when FXLPDDR4CMSC100*HJOP in Test Mode*

IP	I/O Pad	I	E	IE	O	Note
FXLPDDR4CMSC100*HJOP (dctest_mode_i = '1')	DQ[7:0]	dctest_tx_i	dctest_tx_en_i	1'b0	Don't care	When leaktest_i = 0 and dctest_tx_en_b*_i = 1, all IO Pads will follow the input dctest_tx_b*_i.
	DQS_T					
	DQS_C					This configuration is used to test "Ron" and "Static behaviors of IO Driver".
	DMI					

*Table 4-11. Pin Descriptions of Input Buffer when FXLPDDR4CMSC100*HJOP in Test Mode*

IP	I/O Pad	I	E	IE	O	Note
FXLPDDR4CMSC100*HJOP (dctest_mode_i = '1')	DQ[7:0]	Don't care	1'b0	dctest_rx_ie_i	dq*_out	When leaktest_i = 0, dctest_rx_ie = 1 and dctest_rx_odten_i = 1, all the O pins will follow the pattern from I/O pad.
	DQS_T					
	DQS_C					This configuration is used to test ODT and static behavior of IO receiver.
	DMI					All the O pins will be connected to a NAND tree, please refer to Figure 4-4 for more details.

4.4.1 IDDQ Test Mode

The purpose of this test is to measure DC current when IP is at power-on ready state, it is used during production testing. It helps to find out if any abnormal currents from power supplies. The state of all static current consumption blocks is controlled through a dedicated primary input for each IP. It is recommended stopping all clocks to avoid dynamic power from supply. In this mode, only leakage currents are expected from supplies.

Setting of primary input pin related to the I/O IDDQ test mode:

- FXLPDDR4CMSC100*HJOP::retention_en_i = '0'
- FXLPDDR4CMSC100*HJOP::leaktest_i = '1'
- FXLPDDR4CMSC100*HJOP::leaktest_i = '1'
- FXLPDDR4CMSC100*HJOP::leaktest_i = '1'

Set PHY_OVR2::weak_pull_en = 'h1 to disable I/O weak pull-down.

- Program FXLPDDR4CMSC100*HJOP::0x0168 = 'h0000_0010

For the I/O IDDQ test, all gates should be pulled up to power or pulled down to ground. User should set leaktest of FXLPDDR4CMSC100*HJOP, leaktest of FXLPDDR4CMSC100*HJOP and leaktest of FXLPDDR4CMSC100*HJOP to '1' to disable the LPDDR4/3-DDR4/3 Combo PHY I/O and set retention_en_i of FXLPDDR4CMSC100*HJOP to '0'. Those paths must be implemented by pure combinational logic from the input pin.

For the I/O IDDQ test, all PHYs (Including the APHY, DPHY, and CPHY) should be able to enter the IDDQ test mode at the same time. Because the I/O powers of these PHYs are connected together, if users cannot enter the IDDQ test mode at the same time for these PHYs, it is unavailable for users to measure the IDDQ test for these PHY blocks.

4.4.2 Input Pin Leakage Mode

Setting of primary input pin:

- FXLPDDR4CMSC100*HJOP::retention_en_i = '0'
- FXLPDDR4CMSC100*HJOP::leaktest = '0'
- FXLPDDR4CMSC100*HJOP::leaktest = '0'
- FXLPDDR4CMSC100*HJOP::dctest_mode_i = '1'
- FXLPDDR4CMSC100*HJOP::dctest_tx_en_i = '0'
- FXLPDDR4CMSC100*HJOP::dctest_rx_ie_i = '0'
- FXLPDDR4CMSC100*HJOP::dctest_rx_odten_i = '0'
- FXLPDDR4CMSC100*HJOP::leaktest = '0'
- FXLPDDR4CMSC100*HJOP::dctest_mode_i = '1'
- FXLPDDR4CMSC100*HJOP::dctest_tx_en_i = '0'

The input pin leakage mode is for testing the leakage on I/O pins when I/O pins are operating in tristate mode. Before measuring all gates, it should be pulled up to power or pulled down to ground. Please complete below sequences:

1. Set CPHY's PLEG[5:0] = 'h3f (pull-up resistance control) and NLEG[5:0] = 'h3f (pull-down resistance control) to open all of resistance control pin
 - Program FXLPDDR4CMSC100*HJOP::0x0048 = 'h3f3f_3f3f'
2. Set CPHY's ZQ_CAL_CTRL0::sel = 'h0 to force ZQ calibrator use user-defined value
 - Program FXLPDDR4CMSC100*HJOP::0x0040 = 'h0033_1306'
3. Set CPHY's PHY_OVR3::zq_cal_req = 'h3 to bypass ZQ calibration
 - Program FXLPDDR4CMSC100*HJOP::0x009C = 'h0000_0003'
4. Trigger CPHY's initialization
 - Program FXLPDDR4CMSC100*HJOP::0x0084 = 'h0000_0003'
5. Wait for CPHY is ready.
6. Set MISC_ANA_CNTRL0::pronmd = 'h7 and MISC_ANA_CNTRL0::nronmd = 'h7 to the strongest impedance
 - Program FXLPDDR4CMSC100*HJOP::0x0060 = 'h000A_0077'
7. Set MISC_ANA_CNTRL0::pronmd = 'h7 and MISC_ANA_CNTRL0::nronmd = 'h7 to the strongest impedance
 - Program FXLPDDR4CMSC100*HJOP::0x0060 = 'h0010_7721'
8. Force all I/O pads to VDDQ; measure the pin current
9. Force all I/O pads to VSS; measure the pin current

4.4.3 I/O Max. Driving Test of Output Buffer

Setting of primary input pin:

- FXLPDDR4CMSC100*HJOP::retention_en_i = '0'
- FXLPDDR4CMSC100*HJOP::leaktest = '0'
- FXLPDDR4CMSC100*HJOP::leaktest = '0'
- FXLPDDR4CMSC100*HJOP::dctest_mode_i = '1'
- FXLPDDR4CMSC100*HJOP::dctest_tx_en_i = '1'
- FXLPDDR4CMSC100*HJOP::dctest_rx_ie_i = '0'
- FXLPDDR4CMSC100*HJOP::dctest_rx_odten_i = '0'
- FXLPDDR4CMSC100*HJOP::leaktest = '0'
- FXLPDDR4CMSC100*HJOP::dctest_mode_i = '1'
- FXLPDDR4CMSC100*HJOP::dctest_tx_en_i = '1'

The I/O Max. driving test of output buffer is to test if the IO resistance control can be all opened or not. Before measuring all gates, it should be pulled up to power or pulled down to ground. Please complete sequences below:

1. Set CPHY's PLEG[5:0] = 'h3f (pull-up resistance control) and NLEG[5:0] = 'h3f (pull-down resistance control) to open all of the resistance control pins
 - Program FXLPDDR4CMSC100*HJOP::0x0048 = 'h3f3f_3f3f'
2. Set CPHY's ZQ_CAL_CTRL0::sel = 'h0 to force ZQ calibrator use user-defined value
 - Program FXLPDDR4CMSC100*HJOP::0x0040 = 'h0033_1306'

3. Set CPHY's PHY_OVR3::zq_cal_req = 'h3 to bypass ZQ calibration
 - Program FXLPDDR4CMSC100*HJOP::0x009C = 'h0000_0003
4. Trigger CPHY's initialization
 - Program FXLPDDR4CMSC100*HJOP::0x0084 = 'h0000_0003
5. Wait for CPHY is ready.
6. Set MISC_ANA_CNTRL0::pronmd = 'h7 and MISC_ANA_CNTRL0::nronmd = 'h7 to the strongest impedance
 - Program FXLPDDR4CMSC100*HJOP::0x0060 = 'h000A_0077
7. Set MISC_ANA_CNTRL0::pronmd = 'h7 and MISC_ANA_CNTRL0::nronmd = 'h7 to the strongest impedance
 - Program FXLPDDR4CMSC100*HJOP::0x0060 = 'h0010_7721
8. Set FXLPDDR4CMSC100*HJOP::dctest_tx_i = '1' and measure pull-up current at VOM* for all IO PADs
9. Set FXLPDDR4CMSC100*HJOP::dctest_tx_i = '0' and measure pull-down current at VOM* for all IO PADs

*: VOM is defined in the JEDEC, please refer to JEDEC standard for different DRAM applications.

4.4.4 I/O DC Parameter Test of Input Buffer

Setting of primary input pin:

- FXLPDDR4CMSC100*HJOP::retention_en_i = '0'
- FXLPDDR4CMSC100*HJOP::leaktest = '0'
- FXLPDDR4CMSC100*HJOP::leaktest = '0'
- FXLPDDR4CMSC100*HJOP::dctest_mode_i = '1'
- FXLPDDR4CMSC100*HJOP::dctest_tx_en_i = '0'
- FXLPDDR4CMSC100*HJOP::dctest_rx_ie_i = '1'
- FXLPDDR4CMSC100*HJOP::dctest_rx_odten_i = '1'

The I/O DC parameter test of the input buffer is for testing VIH and VIL and termination impedance (Rtt). There is a “NAND Tree” implemented by Faraday for the DPHY block to reduce mux-out logics. The “NAND Tree” is used to measure V_{ih} and V_{il} of the input buffer. **Figure 4-4** shows the NAND tree function of the one byte. dqs_diff_out is an output signal of differential receiver of DQS/DQSB while dqs_se_out is an output signal of single-end receiver of DQS. They can be selected by $dctest_rx_nt_se_sel$. dq^*_out and dmi_out are output signals of single-end receiver for DQ[7:0] and DMI.

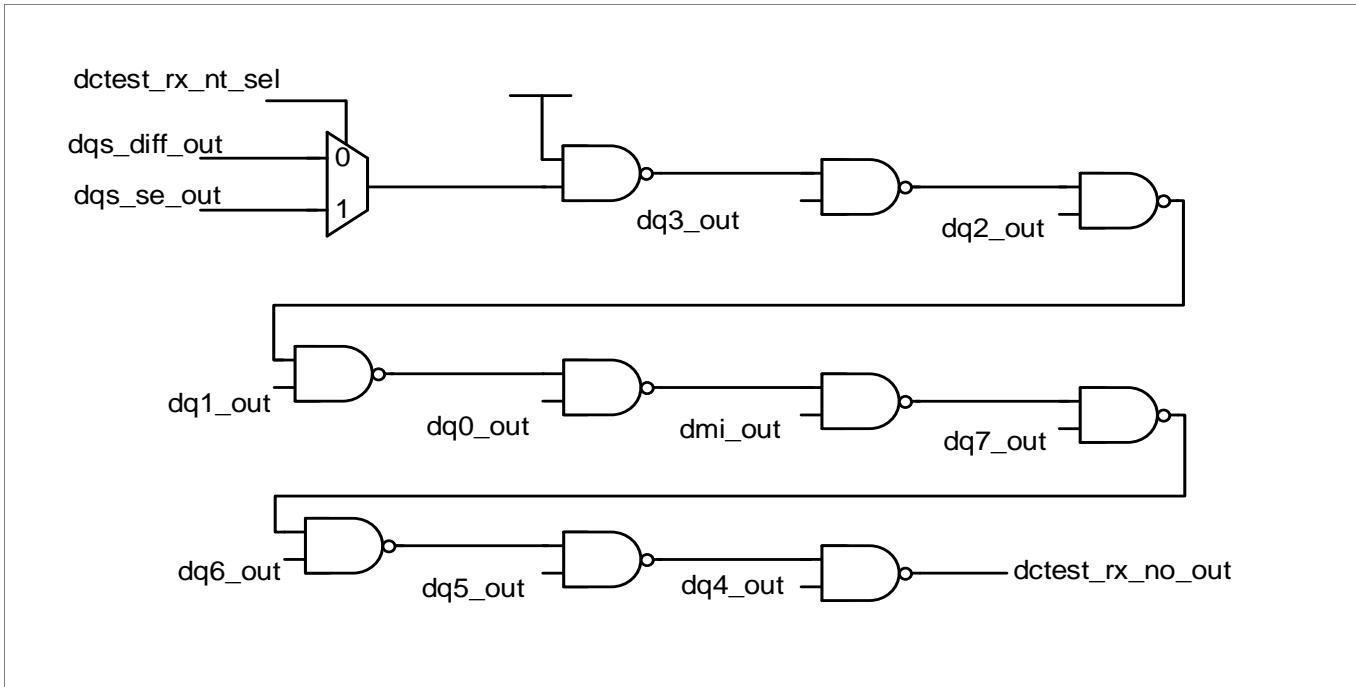


Figure 4-4. NAND Tree Function of FXLPDDR4CMSC100*HJOP Block

1. Set CPHY's MSM_CTRL4 = 'h00fa_00fa and MSM_CTRL5 = 'h0fa0_0fa0 to have enough time for PLL locking
 - Program FXLPDDR4CMSC100*HJOP::0x0010 = 'h00fa_00fa
 - Program FXLPDDR4CMSC100*HJOP::0x0014 = 'h0fa0_0fa0
2. Set CPHY's PLL related register to target frequency based on user's refclk
 - Program FXLPDDR4CMSC100*HJOP::0x0050
 - Program FXLPDDR4CMSC100*HJOP::0x0070
 - Program FXLPDDR4CMSC100*HJOP::0x0054
 - For more details, please refer to **Section 3.2**.
3. Set CPHY's ZQ calibration related register and DPHY's IO related register based on user's application
 - DDR4 mode
 - Program FXLPDDR4CMSC100*HJOP::0x0040 = 'h0033_301f
 - Program FXLPDDR4CMSC100*HJOP::0x0044 = 'h0100_0199
 - Program FXLPDDR4CMSC100*HJOP::0x0068 = 'h0010_5130
 - DDR3/LPDDR3 mode
 - Program FXLPDDR4CMSC100*HJOP::0x0040 = 'h0000_1107
 - Program FXLPDDR4CMSC100*HJOP::0x0044 = 'h0100_0100
 - Program FXLPDDR4CMSC100*HJOP::0x0068 = 'h0010_5030
 - LPDDR4 mode
 - Program FXLPDDR4CMSC100*HJOP::0x0040 = 'h0000_1307
 - Program FXLPDDR4CMSC100*HJOP::0x0044 = 'h0100_00aa
 - Program FXLPDDR4CMSC100*HJOP::0x0068 = 'h0010_5230
4. Set PHY_OVR2::weak_pull_en = 'h1 to disable I/O weak pull-down.
 - Program FXLPDDR4CMSC100*HJOP::0x0168 = 'h0000_0010
5. Set PHY_OVR1::pdb_rx_bias = 'h3 to enable I/O rxbias
 - Program FXLPDDR4CMSC100*HJOP::0x0164 = 'h0000_c000

6. Set DPHY*::MISC_ANA_CNTRL2::rx_mode to appropriate value depending on user's application
Please refer to datasheet in details.
7. Set DPHY*::PHY_OVR23::vref_code to appropriate value depending on user's DC setting. Please refer to datasheet in details.
8. Trigger CPHY's initialization by
 - Program FXLPDDR4CMSC100*HJOP::0x0084 = 'h0000_0003'
9. Wait for CPHY to be ready
10. Set FXLPDDR4CMSC100*HJOP:: dctest_rx_nt_se_sel = '0' to test differential receiver
11. Set {DQS_t, DQS_c, DQ[7:0], DMI} = {1'b1, 1'b0, 8'b1111_1111, 1'b1} keep one refclk cycle
12. Set {DQS_t, DQS_c, DQ[7:0], DMI} = {1'b0, 1'b1, 8'b1111_1111, 1'b1} keep one refclk cycle
13. Set {DQS_t, DQS_c, DQ[7:0], DMI} = {1'b0, 1'b1, 8'b1111_0111, 1'b1} keep one refclk cycle
14. Set {DQS_t, DQS_c, DQ[7:0], DMI} = {1'b0, 1'b1, 8'b1111_0011, 1'b1} keep one refclk cycle
15. Set {DQS_t, DQS_c, DQ[7:0], DMI} = {1'b0, 1'b1, 8'b1111_0001, 1'b1} keep one refclk cycle
16. Set {DQS_t, DQS_c, DQ[7:0], DMI} = {1'b0, 1'b1, 8'b1111_0000, 1'b1} keep one refclk cycle
17. Set {DQS_t, DQS_c, DQ[7:0], DMI} = {1'b0, 1'b1, 8'b1111_0000, 1'b0} keep one refclk cycle
18. Set {DQS_t, DQS_c, DQ[7:0], DMI} = {1'b0, 1'b1, 8'b0111_0000, 1'b0} keep one refclk cycle
19. Set {DQS_t, DQS_c, DQ[7:0], DMI} = {1'b0, 1'b1, 8'b0011_0000, 1'b0} keep one refclk cycle
20. Set {DQS_t, DQS_c, DQ[7:0], DMI} = {1'b0, 1'b1, 8'b0001_0000, 1'b0} keep one refclk cycle
21. Set {DQS_t, DQS_c, DQ[7:0], DMI} = {1'b0, 1'b1, 8'b0000_0000, 1'b0} keep one refclk cycle
22. Set {DQS_t, DQS_c, DQ[7:0], DMI} = {1'b0, 1'b1, 8'b0001_0000, 1'b0} keep one refclk cycle
23. Set {DQS_t, DQS_c, DQ[7:0], DMI} = {1'b0, 1'b1, 8'b0011_0000, 1'b0} keep one refclk cycle
24. Set {DQS_t, DQS_c, DQ[7:0], DMI} = {1'b0, 1'b1, 8'b0111_0000, 1'b0} keep one refclk cycle
25. Set {DQS_t, DQS_c, DQ[7:0], DMI} = {1'b0, 1'b1, 8'b1111_0000, 1'b0} keep one refclk cycle
26. Set {DQS_t, DQS_c, DQ[7:0], DMI} = {1'b0, 1'b1, 8'b1111_0000, 1'b1} keep one refclk cycle
27. Set {DQS_t, DQS_c, DQ[7:0], DMI} = {1'b0, 1'b1, 8'b1111_0001, 1'b1} keep one refclk cycle
28. Set {DQS_t, DQS_c, DQ[7:0], DMI} = {1'b0, 1'b1, 8'b1111_0011, 1'b1} keep one refclk cycle
29. Set {DQS_t, DQS_c, DQ[7:0], DMI} = {1'b0, 1'b1, 8'b1111_0111, 1'b1} keep one refclk cycle
30. Set {DQS_t, DQS_c, DQ[7:0], DMI} = {1'b0, 1'b1, 8'b1111_1111, 1'b1} keep one refclk cycle
31. Set {DQS_t, DQS_c, DQ[7:0], DMI} = {1'b1, 1'b0, 8'b1111_1111, 1'b1} keep one refclk cycle
32. Set FXLPDDR4CMSC100*HJOP:: dctest_rx_nt_se_sel = '1' to test single-end receiver
33. Repeat Step 11 ~ Step 31

4.4.5 I/O DC Parameter Test of Output Buffer

Setting of primary input pin:

- FXLPDDR4CMSC100*HJOP::retention_en_i = '0'
- FXLPDDR4CMSC100*HJOP::leaktest = '0'
- FXLPDDR4CMSC100*HJOP::leaktest = '0'
- FXLPDDR4CMSC100*HJOP::dctest_mode_i = '1'
- FXLPDDR4CMSC100*HJOP::dctest_tx_en_i = '1'
- FXLPDDR4CMSC100*HJOP::dctest_rx_ie_i = '0'
- FXLPDDR4CMSC100*HJOP::dctest_rx_odten_i = '0'
- FXLPDDR4CMSC100*HJOP::leaktest = '0'
- FXLPDDR4CMSC100*HJOP::dctest_mode_i = '1'
- FXLPDDR4CMSC100*HJOP::dctest_tx_en_i = '1'

The I/O DC parameter test of output buffer is used to test the behavior of output buffer and the driving strength which is calibrated by ZQ calibration sequence. The calibrated driving strength is used to verify the calibration circuit and the passing criteria is followed by JEDEC specification.

Please complete the sequences below:

1. Set CPHY's MSM_CTRL4 = 'h00fa_00fa and MSM_CTRL5 = 'h0fa0_0fa0 to have enough time for PLL locking
 - Program FXLPDDR4CMSC100*HJOP::0x0010 = 'h00fa_00fa
 - Program FXLPDDR4CMSC100*HJOP::0x0014 = 'h0fa0_0fa0
2. Set CPHY's PLL related register to target frequency based on user's refclk.
 - Program FXLPDDR4CMSC100*HJOP::0x0050
 - Program FXLPDDR4CMSC100*HJOP::0x0070
 - Program FXLPDDR4CMSC100*HJOP::0x0054
 - For more details, please refer to Section 3.2.
3. Set CPHY's ZQ calibration related register based on user's application
 - DDR4 mode
 - Program FXLPDDR4CMSC100*HJOP::0x0040 = 'h0033_301f
 - Program FXLPDDR4CMSC100*HJOP::0x0044 = 'h0100_0199
 - Program FXLPDDR4CMSC100*HJOP::0x0060 = 'h000a_0066
 - Program FXLPDDR4CMSC100*HJOP::0x0060 = 'h0010_6621
 - DDR3/LPDDR3 mode
 - Program FXLPDDR4CMSC100*HJOP::0x0040 = 'h0000_1107
 - Program FXLPDDR4CMSC100*HJOP::0x0044 = 'h0100_0100
 - Program FXLPDDR4CMSC100*HJOP::0x0060 = 'h000a_0066
 - Program FXLPDDR4CMSC100*HJOP::0x0060 = 'h0010_6621
 - LPDDR4 mode
 - Program FXLPDDR4CMSC100*HJOP::0x0040 = 'h0000_1307
 - Program FXLPDDR4CMSC100*HJOP::0x0044 = 'h0100_00aa
 - Program FXLPDDR4CMSC100*HJOP::0x0060 = 'h000a_0026
 - Program FXLPDDR4CMSC100*HJOP::0x0060 = 'h0010_2621
4. Trigger CPHY's initialization by
 - Program FXLPDDR4CMSC100*HJOP::0x0084 = 'h0000_0003
5. Wait for CPHY to be ready

6. Set FXLPDDR4CMSA100*HJOP::dctest_tx_i = '1' and measure pull-up current at VOM* for all IO PADs
7. Set FXLPDDR4CMSA100*HJOP::dctest_tx_i = '0' and measure pull-down current at VOM* for all IO PADs

*: VOM is defined in the JEDEC, please refer to JEDEC standard for different DRAM applications.

4.5 Mass Production Test Pattern

Table 4-12. Mass Production Test Pattern List

Pattern Name	Provider	Comment
SSA.ftl	Faraday	Test the digital circuits
SSA_sample.ftl	Faraday	Test the digital circuits
scan_chain.ftl	Faraday	For the scan chain test
OCC_tran.ftl	Faraday	Test the digital circuits
occ_sample.ftl	Faraday	Test the digital circuits
PHY_TOP_aphy_lb_compress.ftl	Faraday	Short loopback test mode of APHY Please refer to Section 4.3 for more details.
PHY_TOP_dphy_lb_compress.ftl	Faraday	Short loopback test mode of DPHY Please refer to Section 4.3 for more details.
PHY_TOP_io_hiz_compress.ftl	Faraday	IO Tristate test mode for checking input leakage and Hi-Z. Please refer to Section 4.4.2 for more details.
PHY_TOP_io_max_compress.ftl	Faraday	IO output driving test mode with the maximum driving setting Please refer to Section 4.4.3 for more details.
PHY_TOP_ron_d3_1600_compress.ftl	Faraday	Calibrated IO output driving test mode with the calibration setting of DDR3 Please refer to Section 4.4.5 for more details.
PHY_TOP_ron_d4_1600_compress.ftl	Faraday	Calibrated IO output driving test mode with the calibration setting of DDR4 Please refer to Section 4.4.5 for more details.
PHY_TOP_ron_lp4_1600_compress.ftl	Faraday	Calibrated IO output driving test mode with the calibration setting of LPDDR4 Please refer to Section 4.4.5 for more details.
PHY_TOP_rtt_d3_1600_compress.ftl	Faraday	Rx VIH/VIL and calibrated IO termination resistance test mode with the calibration setting of DDR3 Please refer to Section 4.4.4 for more details.
PHY_TOP_rtt_d4_1600_compress.ftl	Faraday	Rx VIH/VIL and calibrated IO termination resistance test mode with the calibration setting of DDR4 Please refer to Section 4.4.4 for more details.
PHY_TOP_rtt_lp4_1600_compress.ftl	Faraday	Rx VIH/VIL and calibrated IO termination resistance test mode with the calibration setting of LPDDR4 Please refer to Section 4.4.4 for more details.

Pattern Name	Provider	Comment
PHY_TOP_leakage_compress.ftl	Faraday	Power leakage test mode Please refer to Section 4.4.1 for more details.

Chapter 5

Integration Instruction

This chapter contains the following sections:

- 5.1 refclk Requirement
- 5.2 Clock Tree Implementation
- 5.3 Floor Plan, Abutment, and Special Connections
- 5.4 Retention Pin Connection of LPDDR4/LPDDR3-DDR4/DDR3 Combo PHY
- 5.5 Routing of Compensation Bits from FXLPDDR4MCFD100*HJOP to Other Blocks
- 5.6 Pad Sequence in Each Block
- 5.7 Bit Swapping in Data Block and Address/Command Block
- 5.8 Board-level Connection for Compensation Block in FXLPDDR4CMSC100*HJOP
- 5.9 Power Consumption Estimation

5.1 refclk Requirement

This section depicts the implementation requirement of source clock. refclk is a source clock used in FXLPDDR4CMSC100*HJOP, FXLPDDR4CMSC100*HJOP, and FXLPDDR4CMSD100*HJOP. refclk is also used in FXLPDDR4CMSC100*HJOP as embedded PLL's reference clock. refclk source is chosen depending on application; for example, which can be from OSC on the board, ASIC PLL in ASIC, etc.

To meet the embedded PLL's input clock specification, **Table 5-1** shows the duty cycle and clock peak-to-peak jitter. User must meet the specification to ensure that the embedded PLL works normally in FXLPDDR4CMSC100*HJOP.

Table 5-1. refclk Requirement

item	Description	Min	Typ.	Max.	Unit
Duty	Input clock duty ratio	30	50	70	%
FREF period jitter	$10 \text{ MHz} \leq FREF \leq 50 \text{ MHz}$	-	-	$\pm 1\%$	Period

Note: If refclk is changed, user shall change the setting base by the descriptions in Section 3.2.

5.2 Clock Tree Implementation

The clock tree from the clock generation block (FXLPDDR4CMSC100*HJOP) to all data blocks and address/command blocks are already physically presented in each block. Relevant pin connections need to be made in each data block/address block so that the clock tree reaches each data block/address block correctly. No extra buffers/inverters need to be inserted during integration. As users do for clock routing, relevant pin connections need to be done for abutted signals routing as well.

A conceptual diagram for clock tree connection during integration of clock-generation block with data and address/command blocks for a 16-bit single-channel LPDDR4 PHY in NS orientation is as shown in **Figure 5-1**.

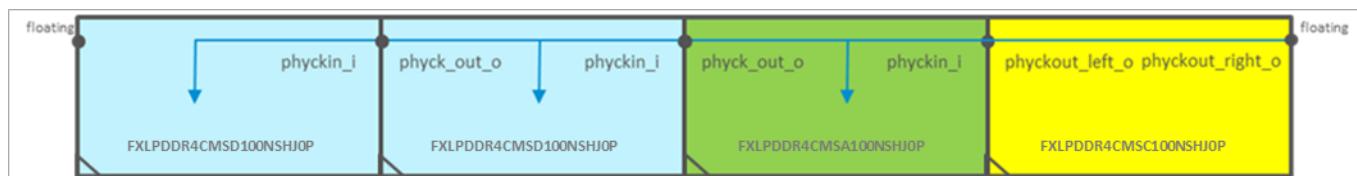


Figure 5-1. Conceptual Block Diagram of 16-bit Single-channel LPDDR4 PHY for Clock Tree Connection

The highest clock named phyck will be propagated by the embedded buffers in APHY/DPHY/CPHY, but there will be skew difference among all macros. If there are 1 CPHY, 2 APHYS, and 8 DPHYS in the whole

DDR application, the clock skew between first APHY and the latest DPHY might be huge. In order to make the skew be balanced among all macros, there is an embedded delay line in each APHY/DPHY/CPHY to adjust the total delay of the highest clock. The embedded delay line of each PHY is controlled by the register shown in **Table 5-2**. Those registers must set a proper value at the beginning.

Table 5-2. Registers for Balancing Phyck among all PHYs, phyck_skewcode

Block	Offset	Bit	Register Name
APHY	0x0064	22:20	phyck_skewcode
DPHY	0x0064	10:8	phyck_skewcode
CPHY	0x0060	14:12	phyck_skewcode

The delay incremental of phyck skew code is 42 ps in Worst Cold case. **Table 5-3** shows the delay of the embedded buffers in different connections. When calculating the total phyck delays of each macro, please make sure that it meets the following requirements. The maximum skew among all macros is < 100 ps.

Table 5-3. Parameter in Worst Cold Case for Calculating Value of phyck_skewcode

From	To	Delay of the Embedded Buffers	Note
CPHY	CPHY	0 ps	CPHY will be the clock source and the delay of embedded buffers will be canceled by internal dummy buffers, so it can be disregarded.
CPHY	APHY		
CPHY	DPHY		
APHY	APHY	79 ps	-
APHY	DPHY	70 ps	-
DPHY	DPHY	62 ps	-

Example: (Take **Figure 5-1** as example)

The CPHY abutted with APHY:

Total Delay of phyck is 0 ps + 40 ps *3 = 120 ps
 Please set phyck_skewcode = 3'b011.

The APHY abutted with DPHY:

Total Delay of phyck is 0 ps + 40 ps * 3 = 120 ps
 Please set phyck_skewcode = 3'b011.

The DPHY abutted with APHY and DPHY:

Total Delay of phyck is 70 ps + 40 ps * 1 = 110 ps

Please set phyck_skewcode = 3'b001.

The DPHY abutted with DPHY:

Total Delay of phyck is 70 ps + 62 ps + 40 ps * 0 = 132 ps

Please set phyck_skewcode = 3'b000.

5.3 Floor Plan, Abutment, and Special Connections

Similarly, a conceptual diagram for abutted signals routing during integration of clock-generation block with data and address/command blocks for a 16-bit single-channel LPDDR4 PHY in NS orientation is as shown in **Figure 5-2**.

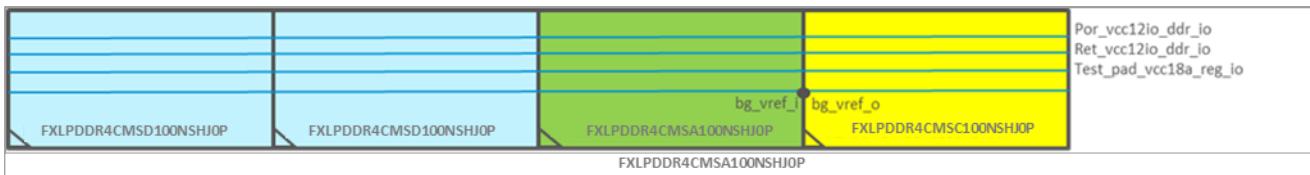


Figure 5-2. Conceptual Block Diagram of 16-bit Single-channel LPDDR4 PHY for Abutted Signals Connection

A conceptual diagram for abutted signals routing during integration of clock-generation block with data and address/command blocks for a 16-bit DDR4 PHY in NS orientation is as shown in **Figure 5-3**.

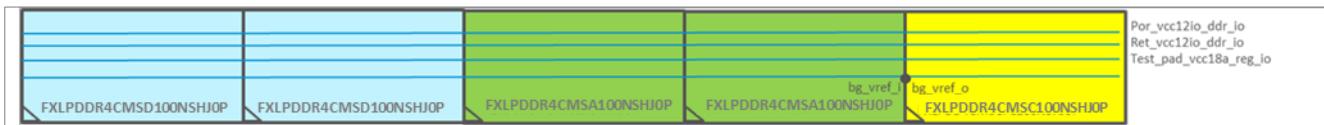


Figure 5-3. Conceptual Block Diagram of 16-bit DDR4 PHY in NS Orientation for Abutted Signals Connection

5.3.1 Special Nets

There are some special nets which must not be buffered during implementation. They are defined as “Don’t Touch” at synthesis stage, and the same at place and route stage. **Please make sure that there is no logic on the following nets.**

- APHY
 - bg_vref_i
 - phyck_in_i
 - phyck_out_o
 - wby2ck_sync_left_i
 - wby2ck_sync_right_i

- wby2ck_sync_left_o
- wby2ck_sync_right_o
- por_vcc12io_ddr_io
- ret_vcc12io_ddr_io
- test_pad_vcc18a_reg_io
- DPHY
 - bg_vref_i
 - phyck_in_i
 - phyck_out_o
 - wby2ck_sync_left_i
 - wby2ck_sync_right_i
 - wby2ck_sync_left_o
 - wby2ck_sync_right_o
 - por_vcc12io_ddr_io
 - ret_vcc12io_ddr_io
 - test_pad_vcc18a_reg_io
- CPHY
 - bg_vref_o
 - phyckout_left_o
 - phyckout_right_o
 - wby2ck_sync_left_i
 - wby2ck_sync_right_i
 - wby2ck_sync_left_o
 - wby2ck_sync_right_o
 - por_vcc12io_ddr_o
 - ret_vcc12io_ddr_o
 - test_probe_i
 - retention_en_i

5.3.2 Special Connection

5.3.2.1 Wby2CK

In UMC28HPCP_LPDDR43_DDR43_PHY, a special synchronize method is used to propagate Wby2CK clocks between PHY blocks to make sure that the phases of Wby2ck are all the same in each PHY block. In different floorplans, the connection and setting would be different.

The following steps are the guideline for setting Wby2ck.

- Step 1: Decide DDR floorplan
- Step 2: Choose Wby2ck source (Master)
- Step 3: Connect necessary Wby2ck* signal
- Step 4: Set appropriate settings in each PHY block

Please make sure that the connection and setting are correct or consult **Faraday DDR AE team** for more details. The connection method and corresponding settings are shown in the following examples:

- Example 1:
 - Floorplan and connection:

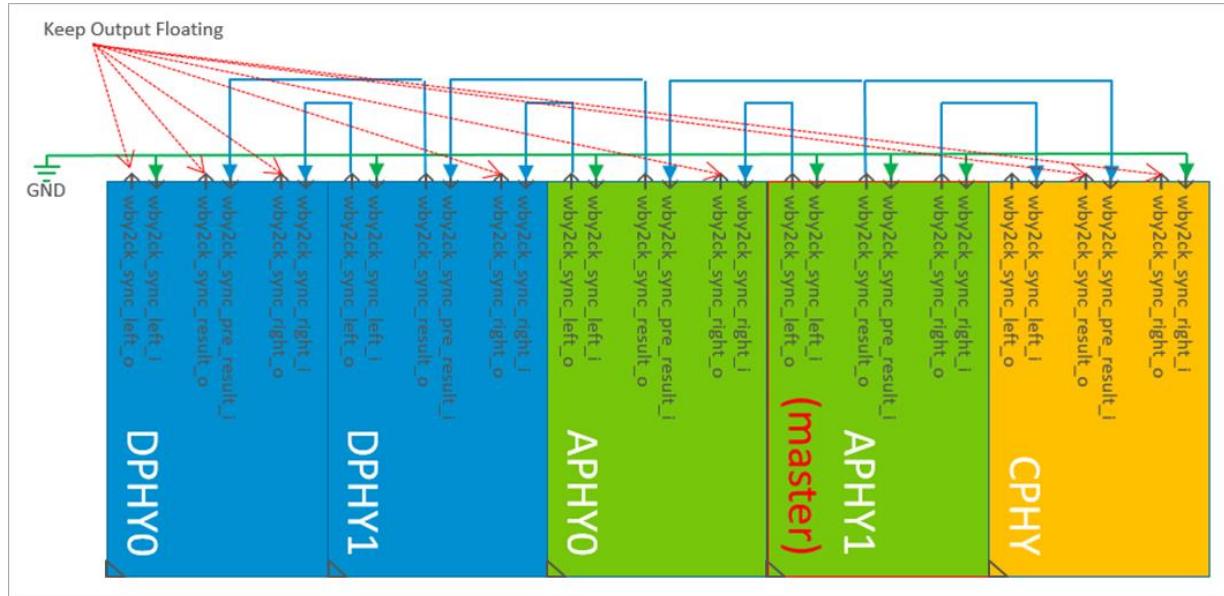


Figure 5-4. Wby2ck Connection Example when set APHY1 as Wby2ck Source (Master)

- Setting:

Table 5-4. Wby2ck Related Setting when Set APHY1 as Wby2ck Source (Master)

Block	Offset	Bit	Register Name	Value
CPHY	0x0060	6	wby2ck_sync_phase_sel	0x0
		5	wby2ck_sync_auto	0x1
		4	wby2ck_sync_source_sel	0x0
APHY1	0x0060	7	wby2ck_sync_phase_sel	0x0
		6	wby2ck_sync_auto	0x0
		5	wby2ck_sync_source_sel	0x0
DPHY0	0x0060	7	wby2ck_sync_phase_sel	0x0
		6	wby2ck_sync_auto	0x1
		5	wby2ck_sync_source_sel	0x1

Block	Offset	Bit	Register Name	Value
DPHY1	0x0064	2	wby2ck_sync_phase_sel	0x0
		1	wby2ck_sync_auto	0x1
		0	wby2ck_sync_source_sel	0x1
DPHY0	0x0064	2	wby2ck_sync_phase_sel	0x0
		1	wby2ck_sync_auto	0x1
		0	wby2ck_sync_source_sel	0x1

- Example 2:
 - Floorplan and connection:

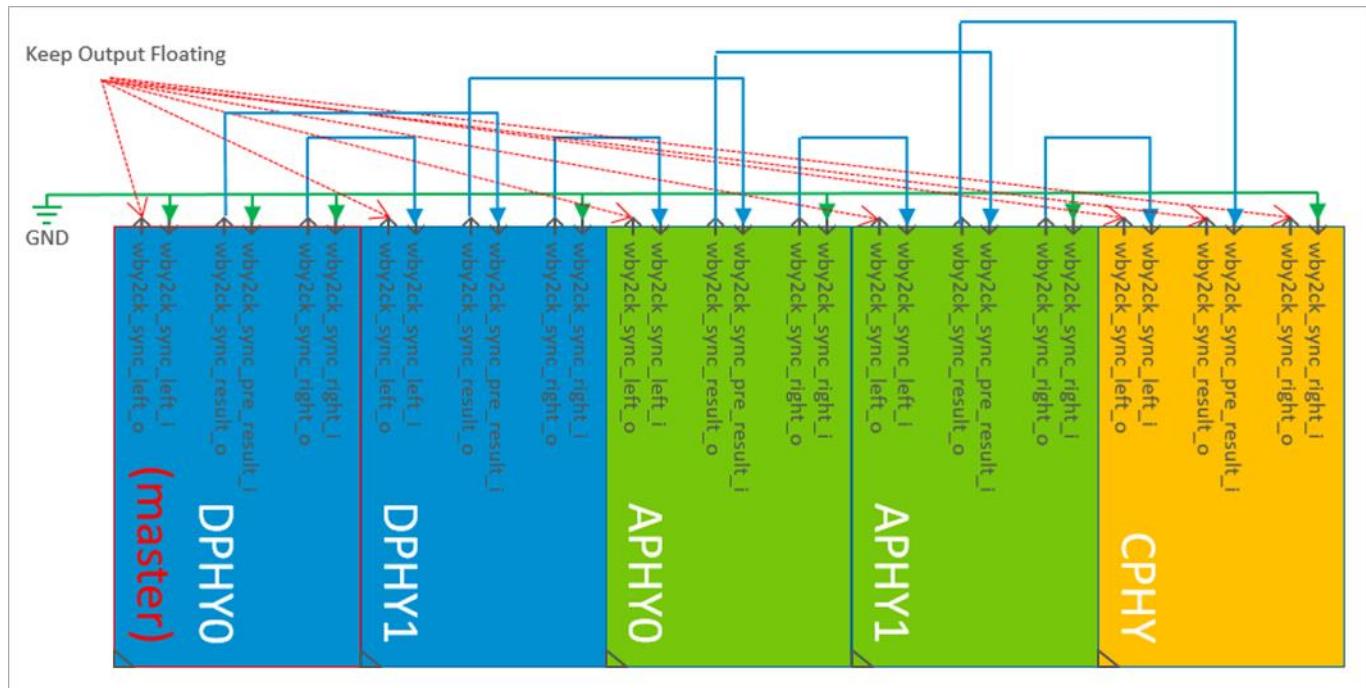


Figure 5-5. Wby2ck Connection Example when Set DPHY0 as Wby2ck Source (Master)

- Setting:

Table 5-5. Wby2ck Related Setting when Set DPHY as Wby2ck Source (Master)

Block	Offset	Bit	Register Name	Value
CPHY	0x0060	6	wby2ck_sync_phase_sel	0x0
		5	wby2ck_sync_auto	0x1
		4	wby2ck_sync_source_sel	0x0
APHY1	0x0060	7	wby2ck_sync_phase_sel	0x0
		6	wby2ck_sync_auto	0x1
		5	wby2ck_sync_source_sel	0x0
APHY0	0x0060	7	wby2ck_sync_phase_sel	0x0
		6	wby2ck_sync_auto	0x1
		5	wby2ck_sync_source_sel	0x0
DPHY1	0x0064	2	wby2ck_sync_phase_sel	0x0
		1	wby2ck_sync_auto	0x1
		0	wby2ck_sync_source_sel	0x0
DPHY0	0x0064	2	wby2ck_sync_phase_sel	0x0
		1	wby2ck_sync_auto	0x0
		0	wby2ck_sync_source_sel	0x0

5.3.2.2 EFIFO Read Pointer Reset

In UMC28HPCP_LPDDR43_DDR43_PHY, the Write FIFO's write/read pointer operation is free-run after PHY initialization is done. To keep the propagate command/address signal and data signal from DFIW to DRAM interface, users must synchronize all of PHY blocks de-assert read pointer reset signal at the same cycle. In this way, users can ensure that the signal phase from DFIW to DRAM interface propagates correctly.

The following steps are the guideline for setting rptr_*_rstn_*.

- Step 1: Decide DDR floorplan
- Step 2: Choose rptr_*_rstn_* source(Master)
 - Suggestion: Please choose the central block in the whole floorplan as the master.
- Step 3: Connect necessary rptr_*_rstn_* signal

- Step 4: Set appropriate setting in each PHY blocks
 - In APHY and DPHY's registers, it supports 0T ~ 15T cycle delays on Write FIFO's read domain. (= Tsetting)
 - In APHY, the intrinsic delay from the input reset signal to the Write FIFO's read pointer is 9T cycle delays (= Tintrinsic_dphy).
 - In DPHY, the intrinsic delay from the input reset signal to the Write FIFO's read pointer is 6T cycle delays (= Tintrinsic_aphy).
 - The propagate delay from master to the target PHY block depends on the number of connection rptr_*_rstn_* and rptr_*_rstn_fwd_. Every connection represents 3T cycle delays (= Tpropagate = 3* "No. of propagate connection").
 - Find the largest delay of PHY block as synchronous target (= Tlargest)
 - The setting of each target PHY block => Tsetting = Tlargest – Tintrinsic – Tpropagate

Please make sure that the connection and setting are correct or please consult [Faraday DDR AE team](#) for more details.

The connection method and corresponding setting are shown in the following examples:

- How to decide rptr_*_rstn_* source (Master):
 - Total No. of A/DPHY is odd case:

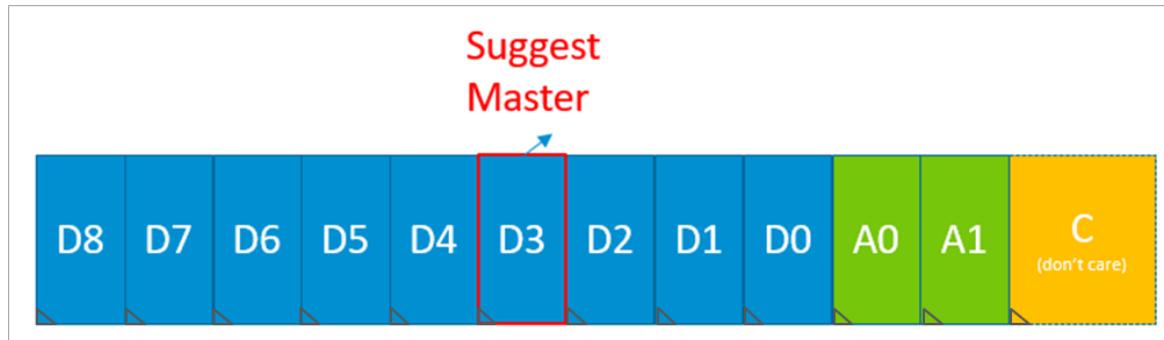


Figure 5-6. Choose rptr_*_rstn_* Source in Odd Case

- Total No. of A/DPHY is even case:

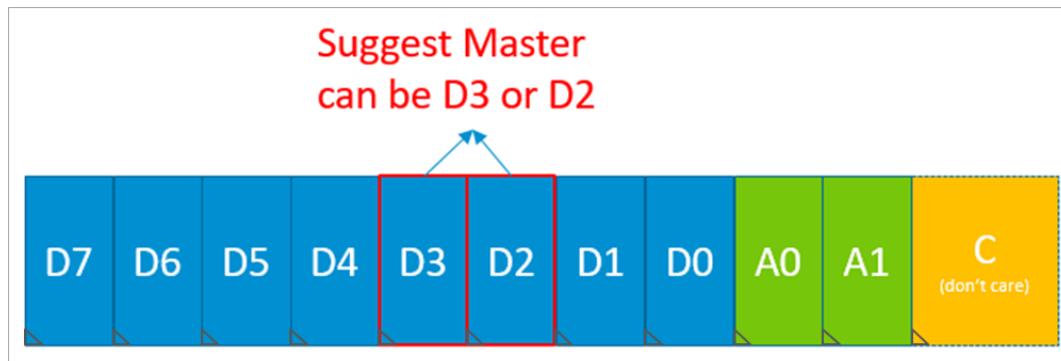


Figure 5-7. Choose rptr_*_rstn_* Source in Even Case

- How to choose `rptr_*_rstn_*` and `rptr_*_rstn_fwd_*` as connection:
 - The block abutted with Master use `rptr_*_rstn_*`
 - The block not abutted with Master use `rptr_*_rstn_fwd_*`

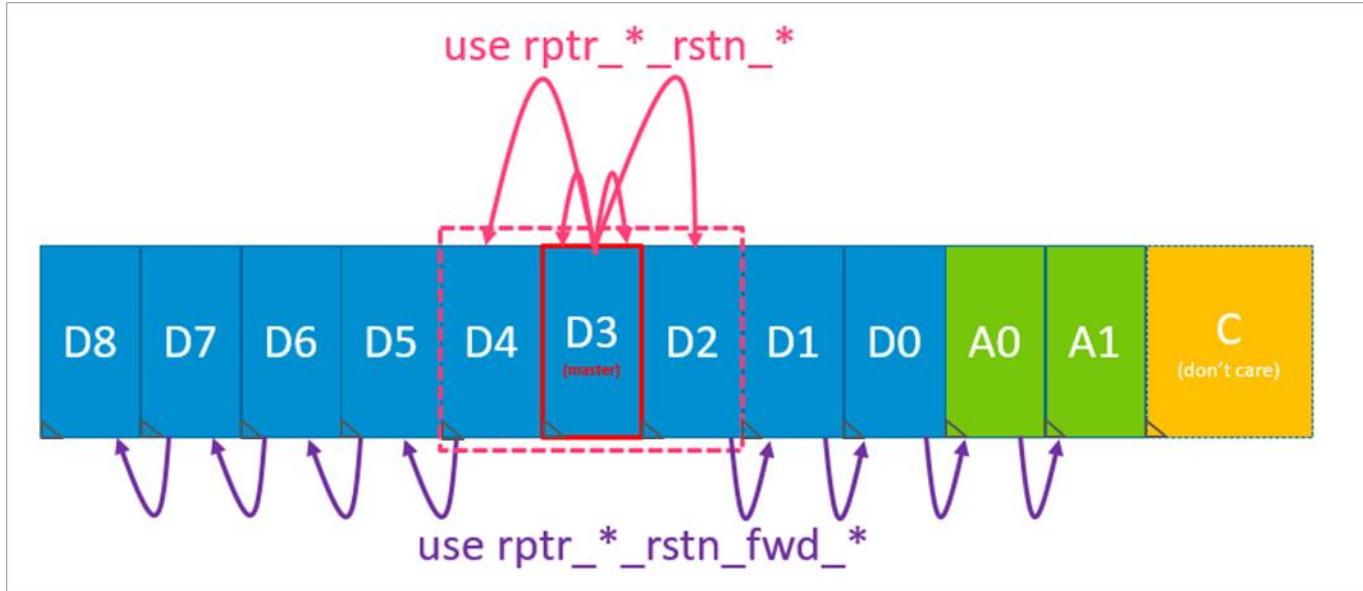


Figure 5-8. Guidance to Select `rptr_*_rstn_*` or `rptr_*_rstn_fwd_*` as Connection

- How to count the “No. of propagate connection”:
 - Master itself
 - For APHYO, the number of propagate connection is ‘1’.
 - If user chooses DPHY as Master, please follow the same rules.

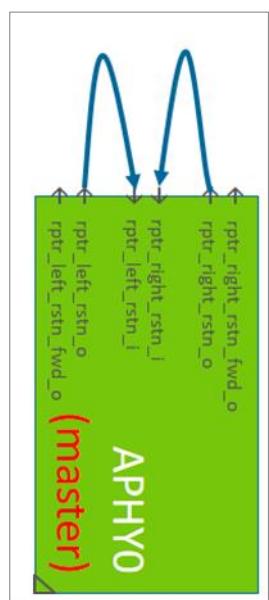


Figure 5-9. Guidance to Count Total Number of Connection to Master

- Abut Master left or right side
 - For DPHY1 and APHY1, the number of propagate connection is '1'.



Figure 5-10. Guidance to Count Total Number of Connection to Block Abut

- Others
 - For DPHY0, the number of propagate connection is '2'.

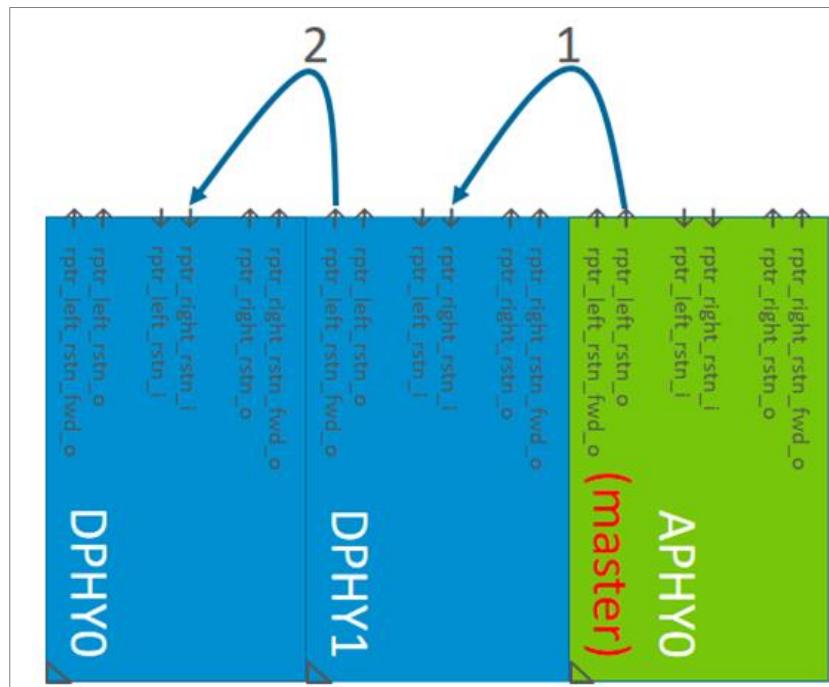


Figure 5-11. Guidance to Count Total Number of Connection to Block Not Abut Master

- How to find synchronous target
 - Case1:

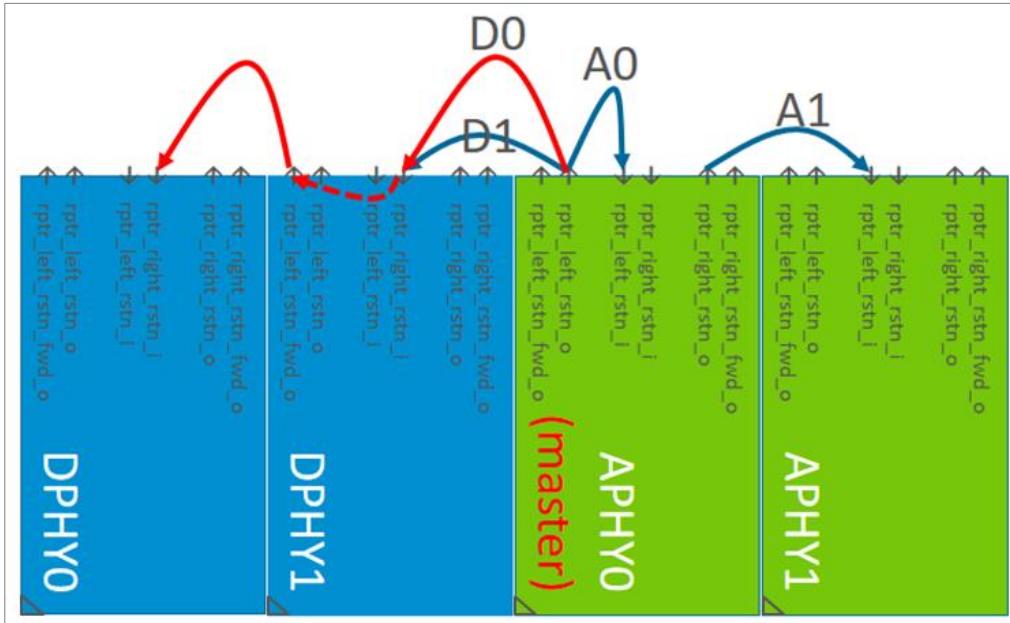


Figure 5-12. Guidance to Find the Synchronous Target Case1

- T_APHY0 is 12T ($9T + 3T * 1$).
- T_APHY1 is 12T ($9T + 3T * 1$).
- T_DPHY0 is 12T ($6T + 3T * 2$).
- T_DPHY1 is 9T ($6T + 3T * 1$).
 - ➔ $T_{largest} = 12T$, (DPHY0, APHY0, and APHY1)
 - ➔ T_DPHY1 is smaller than $T_{largest}$, so DPHY1 should set DPHY: MISC_DIG_CNTRL1::rptr_RST_SEL = 4'h3 for compensating the difference.

- Case2:

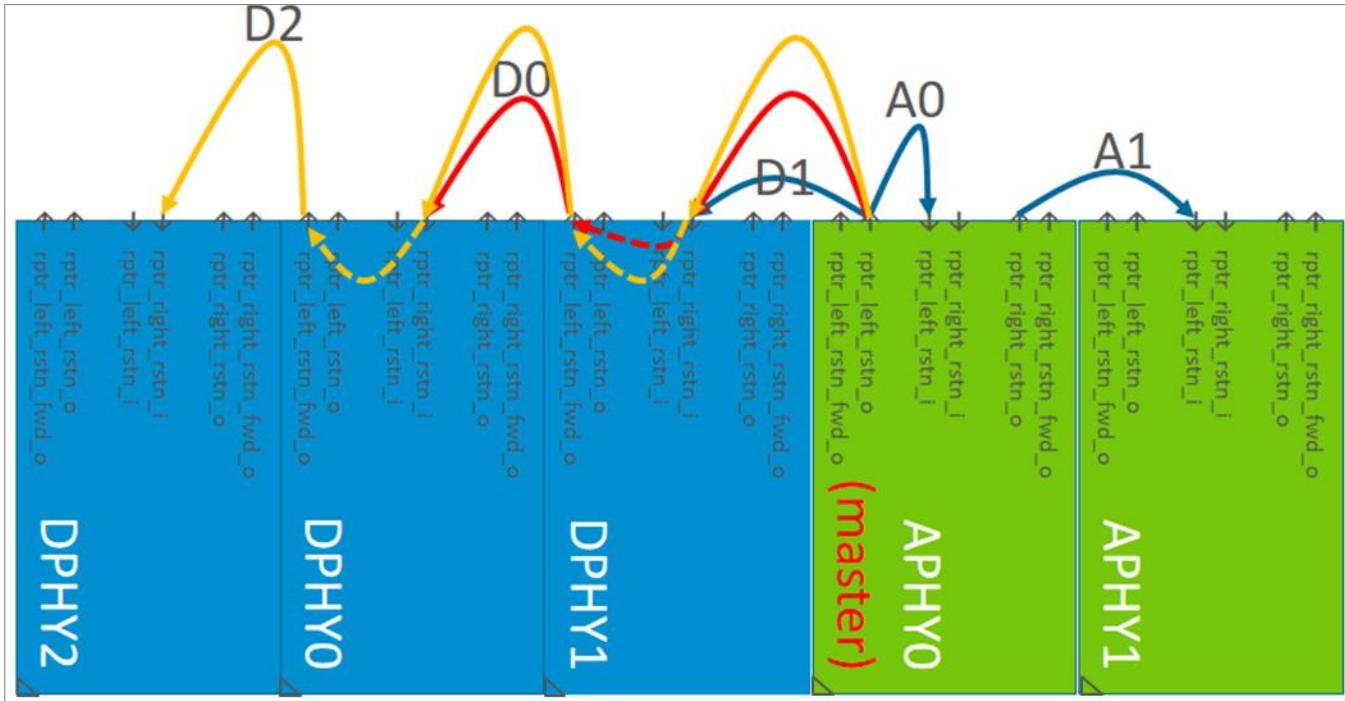


Figure 5-13. Guidance to Find Synchronous Target Case2

- T_APY0 is 12T (9T + 3T * 1)
- T_APY1 is 12T (9T + 3T * 1)
- T_DPHY0 is 12T (6T + 3T * 2)
- T_DPHY1 is 9T (6T + 3T * 1)
- T_DPHY2 is 15T (6T + 3T * 3)
 - $T_{largest} = 15T$, (DPHY2)
 - T_APY0 and T_APY1 are smaller than $T_{largest}$, so APY0 and APY1 should set APY::MISC_DIG_CNTRL0::rptr_RST_SEL = 4'h3 for compensating the difference.
 - T_DPHY0 is smaller than $T_{largest}$, so DPHY1 should set DPHY::MISC_DIG_CNTRL1::rptr_RST_SEL = 4'h3 for compensating the difference.
 - T_DPHY1 is smaller than $T_{largest}$, so DPHY1 should set DPHY::MISC_DIG_CNTRL1::rptr_RST_SEL = 4'h6 for compensating the difference.

- `rptr_*_rstn_*` Example1
 - Floorplan and Connection:

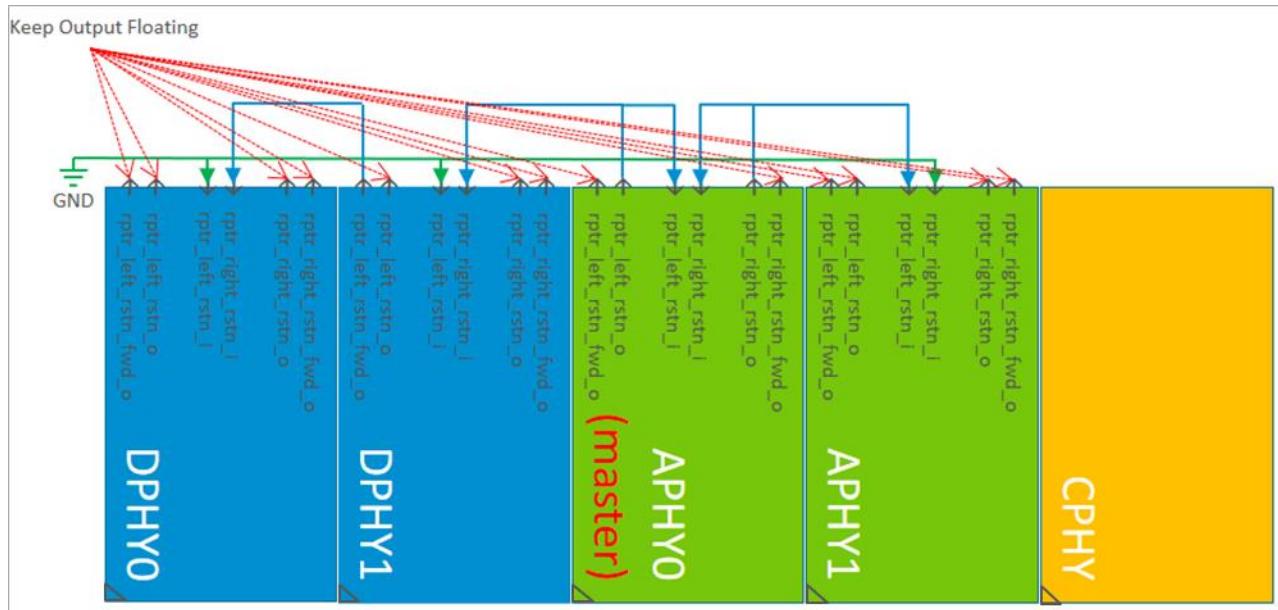


Figure 5-14. `rptr_*_rstn_*` Connection Example when Setting APHY0 as `rptr_rstn` Source (Master)

- Setting calculation:
 - DPHY0 is synchronous target. $T_{largest}$ is 12T ($T_{intrinsic_dphy} + T_{propagate} * 2$)
 - $T_{setting_DPHY0} = 12T - 6T - 3T * 2 = 0T$
 - $T_{setting_DPHY1} = 12T - 6T - 3T * 1 = 3T$
 - $T_{setting_APHY0} = 12T - 9T - 3T * 1 = 0T$
 - $T_{setting_APHY1} = 12T - 9T - 3T * 1 = 0T$
- Setting

Table 5-6. `rptr_*_rstn_*` Related Setting when Setting APHY0 as `rptr_rstn` Source (Master)

Block	Offset	Bit	Register name	Value	Note
APHY1	0x0070	12:8	wptr_rst_sel	Set to 0x5	1, 2
		7:4	rptr_rst_sel	0x0	-
		1:0	sync_pos	Don't care	3

Block	Offset	Bit	Register name	Value	Note
APHY0 (Master)	0x0070	12:8	wptr_RST_SEL	Set to 0x5	1, 2
		7:4	rptr_RST_SEL	0x0	-
		1:0	sync_pos	Keep default	4
DPHY1	0x0074	12:8	wptr_RST_SEL	Set to 0x5	1, 2
		7:4	rptr_RST_SEL	0x3	-
		1:0	sync_pos	Don't care	3
DPHY0	0x0074	12:8	wptr_RST_SEL	Set to 0x5	1, 2
		7:4	rptr_RST_SEL	0x0	-
		1:0	sync_pos	Don't care	3

Note 1: "wptr_RST_SEL" of all the PHY blocks must set the same value.

Note 2: **Caution!** In this PHY IP, after PHY initialization has been done, please trigger efifo_RST_N by DFIW.

Note 3: "sync_pos" only works on the source (Master) of rptr_RST_N which is not the master block to affect "rptr_RST_N".

Note 4: "sync_pos" default is set to '0x1'. If users want to tune this value, please consult [Faraday DDR AE team](#) for more details.

- **rptr_*_rstn_* Example 2**
 - Floorplan and connection:

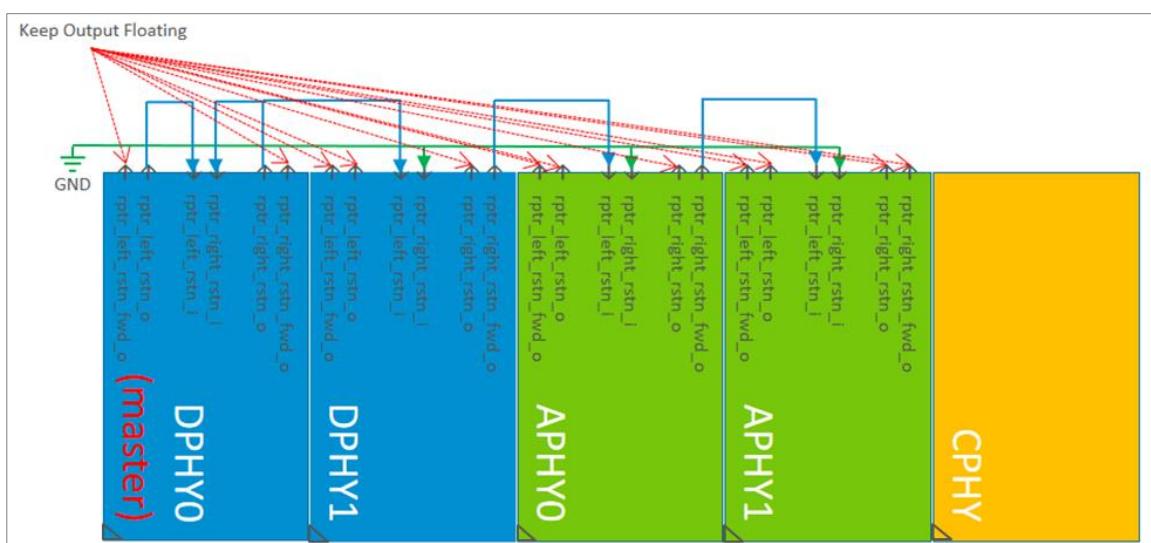


Figure 5-15. *rptr_*_rstn_* Connection Example when Setting DPHY0 as rptr_rstn Source (Master)*

- Setting calculation:
 - APHY1 is synchronous target. Tlargest is 18T (Tintrinsic_aphy + Tpropagate*3).
 - Tsetting_DPHY0 = 18T - 6T - 3T*1 = 9T
 - Tsetting_DPHY1 = 18T - 6T - 3T*1 = 9T
 - Tsetting_APHY0 = 18T - 9T - 3T*2 = 3T
 - Tsetting_APHY1 = 18T - 9T - 3T*3 = 0T

*Table 5-7. rptr_*_rstn_* Related Setting when Setting DPHY0 as rptr_rstn Source (Master)*

Block	Offset	Bit	Register name	Value	Note
APHY1	0x0070	12:8	wptr_rst_sel	Set to 0x5	1, 2
		7:4	rptr_rst_sel	0x0	-
		1:0	sync_pos	Don't care	3
APHY0	0x0070	12:8	wptr_rst_sel	Set to 0x5	1, 2
		7:4	rptr_rst_sel	0x3	-
		1:0	sync_pos	Don't care	3
DPHY1	0x0074	12:8	wptr_rst_sel	Set to 0x5	1, 2
		7:4	rptr_rst_sel	0x9	-
		1:0	sync_pos	Don't care	3
DPHY0 (Master)	0x0074	12:8	wptr_rst_sel	Set to 0x5	1, 2
		7:4	rptr_rst_sel	0x9	-
		1:0	sync_pos	Keep default	4

Note 1: "wptr_rst_sel" of all the PHY blocks must set the same value.

Note 2: Caution! In this PHY IP, after PHY initialization has been done, please trigger efifo_rst_n by DFIW.

Note 3: "sync_pos" only works on the source (Master) of rptr_rst_n which is not master block to affect "rptr_rst_n".

Note 4: "sync_pos" default is set to '0x1'. If users want to tune this value, please consult [Faraday DDR AE team](#) for more details.

- `rptr_*_rstn_*` Example3 (**Prohibited cases**)
 - floorplan:



Figure 5-16. `rptr_*_rstn_*` Connection Prohibited Example

- Setting calculation:
 - DPHY8 is synchronous target. Tlargest is 36T ($T_{intrinsic_dphy} + T_{propagate} * 10$).
 - $T_{setting_APHY0} = 36T - 9T - 3T*1 = 24T$ (Not supported)
 - $T_{setting_APHY1} = 36T - 9T - 3T*1 = 24T$ (Not supported)
 - $T_{setting_DPHY0} = 36T - 6T - 3T*2 = 24T$ (Not supported)
 - $T_{setting_DPHY1} = 36T - 6T - 3T*3 = 21T$ (Not supported)
 - $T_{setting_DPHY2} = 36T - 6T - 3T*4 = 18T$ (Not supported)
 - $T_{setting_DPHY3} = 36T - 6T - 3T*5 = 15T$
 - $T_{setting_DPHY4} = 36T - 6T - 3T*6 = 12T$
 - $T_{setting_DPHY5} = 36T - 6T - 3T*7 = 9T$
 - $T_{setting_DPHY6} = 36T - 6T - 3T*8 = 6T$
 - $T_{setting_DPHY7} = 36T - 6T - 3T*9 = 3T$
 - $T_{setting_DPHY8} = 36T - 6T - 3T*10 = 0T$

5.3.3 Floor Plan and Abutment

For the complete PHY solution, it must contain at least one data block, one (or two for DDR4/3) address/command block(s), and one clock-generation block. If users want to come up with any other combinations other than what have been mentioned in this section, please consult [Faraday DDR AE team](#) for more details. **Figure 5-17** shows one placement for 16-bit LPDDR4 PHY in NS orientation. In this placement, all blocks need to be abutted to each other, and make sure that power/ground and other internal signals are automatically shared among blocks.



Figure 5-17. Abutment of Different Blocks for a 16-bit Single-channel LPDDR4 PHY

Figure 5-18 shows one placement for 16-bit DDR4 PHY if the entire PHY is placed in NS orientation.

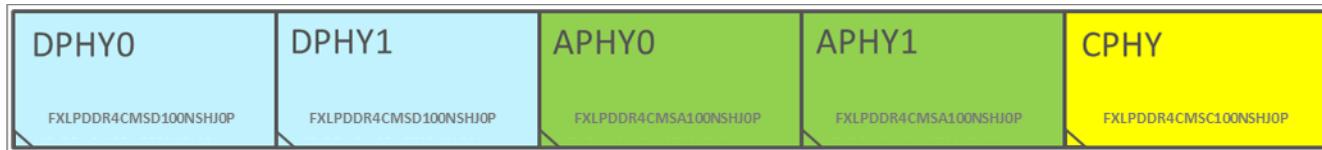


Figure 5-18. Abutment of Different Blocks for a 16-bit DDR4 PHY in NS Orientation

5.3.4 Clock and Abutted Signal Connections in RTL

In this section, use **Figure 5-19** as an example to guide user how to connect PHYCK and special abutted signals between PHY blocks. The connections must be done correctly in RTL during integration.

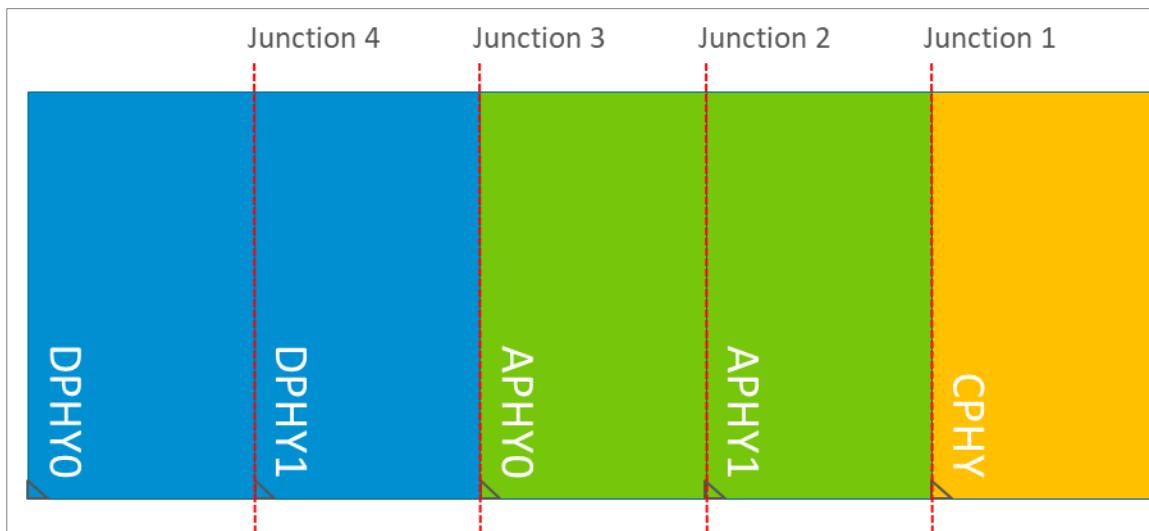


Figure 5-19. Floorplan Example of Clock and Abutted Signals Connections

5.3.4.1 PHYCK Connection

Table 5-8 ~ Table 5-11 show the connections that need to be done in RTL during integration of different blocks and list the connections that need to be done for clock routing among all blocks.

Table 5-8. PHYCK Connections between APHY1 and CPHY

Junction 1	APHY1	CPHY
Pin	phyckin_i	phyckout_left_o

Note: CPHY's phyckout_right_o did not abut any blocks, keep it floating.

Table 5-9. PHYCK Connections between APHY0 and APHY1

Junction 2	APHY0	APHY1
Pin	phyckin_i	phyckout_o

Table 5-10. PHYCK Connections between DPHY1 and APHY0

Junction 3	DPHY1	APHY0
Pin	phyckin_i	phyckout_o

Table 5-11. PHYCK Connections between DPHY0 and DPHY1

Junction 4	DPHY0	DPHY1
Pin	phyckin_i	phyckout_o

Note: DPHY0's phyckout_o did not abut any blocks, keep it floating.

5.3.4.2 Special Signals Abutment

The pins listed below must be abutted at P&R stage. The connection that need to be done in RTL during integration. Please follow the guidance from **Table 5-12 ~ Table 5-15**.

- CPHY:
 - bg_vref_o
 - por_vcc12io_ddr_o
 - ret_vcc12io_ddr_o
 - test_probe_i
- APHY:
 - bg_vref_i
 - por_vcc12io_ddr_io
 - ret_vcc12io_ddr_io
 - test_pad_vcc18a_reg_io
- DPHY:
 - bg_vref_i
 - por_vcc12io_ddr_io
 - ret_vcc12io_ddr_io
 - test_pad_vcc18a_reg_io

Table 5-12. Special Signals Connection between APHY1 and CPHY

Junction 1	APHY1	CPHY
Pin		
	bg_vref_i	bg_vref_o
	por_vcc12io_ddr_io	por_vcc12io_ddr_o
	ret_vcc12io_ddr_io	ret_vcc12io_ddr_o
	test_pad_vcc18a_reg_io	test_probe_i

Table 5-13. Special Signals Connection between APHY0 and APHY1

Junction 2	APHY0	APHY1
Pin		
	bg_vref_i	bg_vref_i
	por_vcc12io_ddr_io	por_vcc12io_ddr_io
	ret_vcc12io_ddr_io	ret_vcc12io_ddr_io
	test_pad_vcc18a_reg_io	test_pad_vcc18a_reg_io

Table 5-14. Special Signals Connection between DPHY1 and APHY0

Junction 3	DPHY1	APHY0
Pin		
	bg_vref_i	bg_vref_i
	por_vcc12io_ddr_io	por_vcc12io_ddr_io
	ret_vcc12io_ddr_io	ret_vcc12io_ddr_io
	test_pad_vcc18a_reg_io	test_pad_vcc18a_reg_io

Table 5-15. Special Signals Connection between DPHY0 and DPHY1

Junction 4	DPHY0	DPHY1
Pin	bg_vref_i	bg_vref_i
	por_vcc12io_ddr_io	por_vcc12io_ddr_io
	ret_vcc12io_ddr_io	ret_vcc12io_ddr_io
	test_pad_vcc18a_reg_io	test_pad_vcc18a_reg_io

5.3.4.3 Power/Ground Connection

Table 5-16 lists the power and ground in each PHY block, the same naming of power/ground must be shorted and cannot be separated in RTL.

Table 5-16. Power/Ground list

-	CPHY	APHY	DPHY	Note
Power	VCC09K	VCC09K	VCC09K	It must be shorted to each other.
	VCC12IO_DDR	VCC12IO_DDR	VCC12IO_DDR	Please make a virtual shot connection in layout/GDS and have they are merged in package. For more details of pkg requirement on I/O power/ground, please refer to Section 6.4.
	-	VCC12IO_DDRCK	-	Only APHY has VCC12IO_DDRCK power pin
	VCC18A_REG	VCC18A_REG	VCC18A_REG	It must be shorted to each other.
Ground	GND09K	GND09K	GND09K	It must be shorted to each other.
	GNDA	-	-	Only CPHY has GNDA ground pin.
	GNDIO_DDR	GNDIO_DDR	GNDIO_DDR	It must be shorted to each other. Please make a virtual shot connection in layout/GDS and have them merged in package. For more details of pkg requirement on I/O power/ground, please refer to Section 6.4
	-	GNDIO_DDRCK	-	Only APHY has GNDIO_DDRCK ground pin

5.4 Retention Pin Connection of LPDDR4/LPDDR3-DDR4/DDR3 Combo PHY

As discussed in the data sheet, this LPDDR4/LPDDR3-DDR4/DDR3 PHY supports retention of I/O pads when the core voltage is turned off. This feature is enabled through the core-side input pin “*retention_en_i*” in FXLPDDR4CMSC100*HJOP. During retention, the PHY requires this pin to be logic high. Hence, it is recommended connecting this signal and control it through Live-core domain (core domain in which core voltage is still available even though core voltage of LPDDR4/LPDDR3-DDR4/DDR3 PHY is turned off) during integration of our PHY in SoC.

Another way to control this pin (if live-core domain is not available in SoC) is to connect this pin through a BAREWIRE (I/O cell that connects PAD to core input through a resistor) and control the PAD on-board. In this case, it is important not to connect any buffer in the path from BAREWIRE to the input pin, “*retention_en_i*”. If any buffers are inserted in the path, its logic value will be lost when core voltage of the buffer is turned off and making retention feature meaningless.

Below are the sequences that can be followed to enter into (and) exit from retention:

1. Enter into Self-refresh mode for DRAM
2. Pull CKE to logic LOW as per DRAM specification
3. Enable retention (Pull “*retention_en_i*” to logic HIGH driven from Live-core domain)
4. Values for CKE and RESET_N will be retained at their respective I/O PAD.
5. Power for both VCC09K can be turned off for the PHY.
6. Power-up after retention period
7. Follow PHY reset sequences
8. Write all configuration registers with trained values to continue with normal operation as the values in the registers would be lost during retention (as VCC09K power was turned off).
9. Disable retention (Pull “*retention_en_i*” to logic low)
10. Drive CKE with logic high
11. DRAM and PHY are back to normal operation.

Figure 5-20 depicts the timing diagram for retention sequence for LPDDR4 mode. Please note that this is not drawn to scale and proper timing needs to be followed between each state.

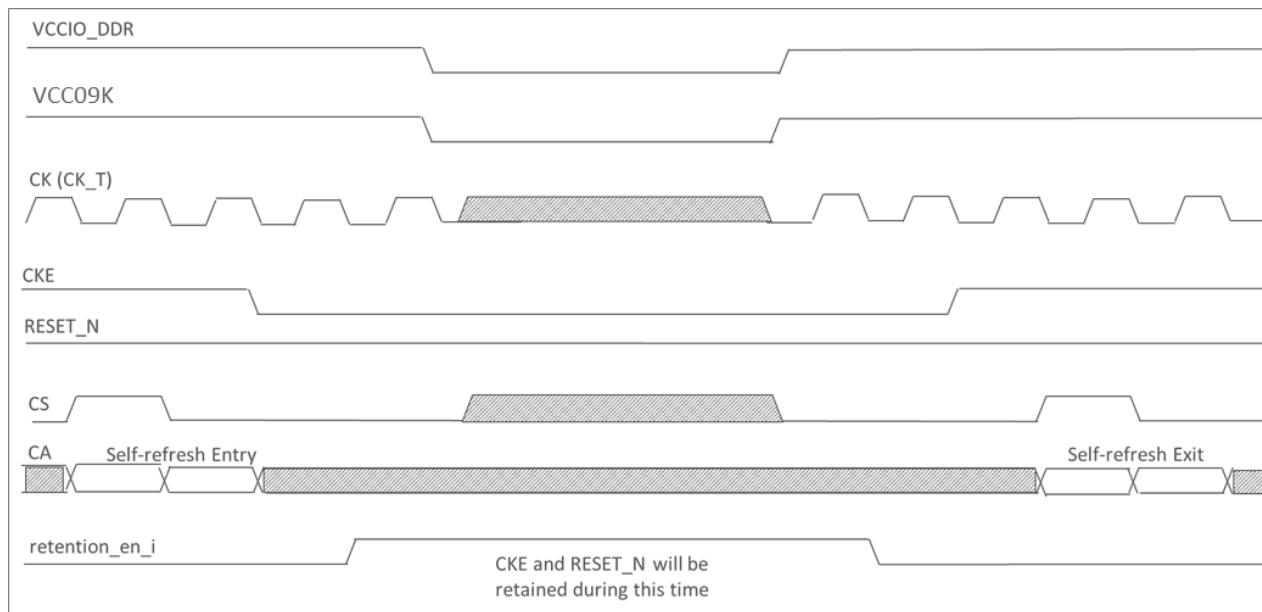


Figure 5-20. Timing Sequence to Go into PHY Retention (Not Drawn to Scale)

5.5 Routing of Compensation Bits from FXLPDDR4MSCFD100*HJOP to Other Blocks

Since the compensation block is presented in FXLPDDR4CMSC100*HJOP, the compensation bits at the end of compensation are available as outputs from the same block. These bits need to be connected to data block (FXLPDDR4CMSC100*HJOP) and address/command block (FXLPDDR4CMSC100*HJOP), so that the I/O drivers presented inside the data block and address/command block use the correct compensation bits across PVT.

Compensation output bits from FXLPDDR4CMSC100*HJOP are `comp_dop_aphy_o[5:0]`, `comp_don_aphy_o[5:0]`, `comp_dop_dphy_o[5:0]`, and `comp_don_dphy_o[5:0]`. Compensation input bits in FXLPDDR4CMSC100*HJOP and FXLPDDR4CMSC100*HJOP are `comp_dop_i[5:0]` and `comp_don_i[5:0]`.

During integration, customer is required to

- connect “`comp_dop_aphy_o[5:0]`” from FXLPDDR4CMSC100*HJOP to “`comp_dop_i[5:0]`” in FXLPDDR4CMSC100*HJOP.
- connect “`comp_don_aphy_o[5:0]`” from FXLPDDR4CMSC100*HJOP to “`comp_don_i[5:0]`” in FXLPDDR4CMSC100*HJOP.
- connect “`comp_dop_dphy_o[5:0]`” from FXLPDDR4CMSC100*HJOP to “`comp_dop_i[5:0]`” in FXLPDDR4CMSC100*HJOP.
- connect “`comp_don_dphy_o[5:0]`” from FXLPDDR4CMSC100*HJOP to “`comp_don_i[5:0]`” in FXLPDDR4CMSC100*HJOP.

FXLPDDR4CMSC100*HJOP as shown in **Figure 5-21**. It is recommended that user adds buffers in the connection between FXLPDDR4CMSC100*HJOP and FXLPDDR4CMSD100*HJOP/
FXLPDDR4CMSC100*HJOP as it is a very long routing.

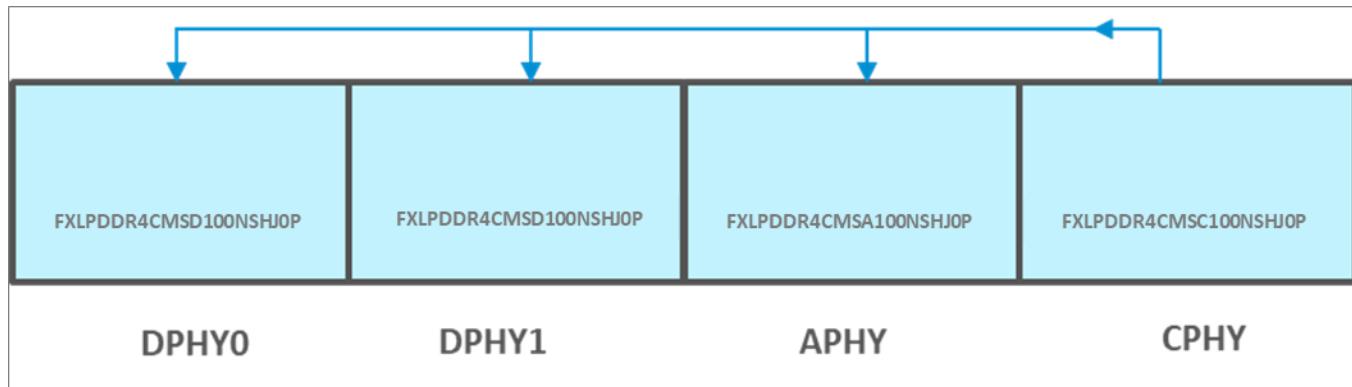


Figure 5-21. Compensation Bits Routing from FXLPDDR4CMSC100*HJOP to FXLPDDR4CMSD100* HJOP /FXLPDDR4CMSC100* HJOP for a 1-channel 16-bit LPDDR4 PHY

5.6 Pad Sequence in Each Block

This section mentions the pad sequence for clock-generation, data, and address/command blocks in both NS and EW orientations.

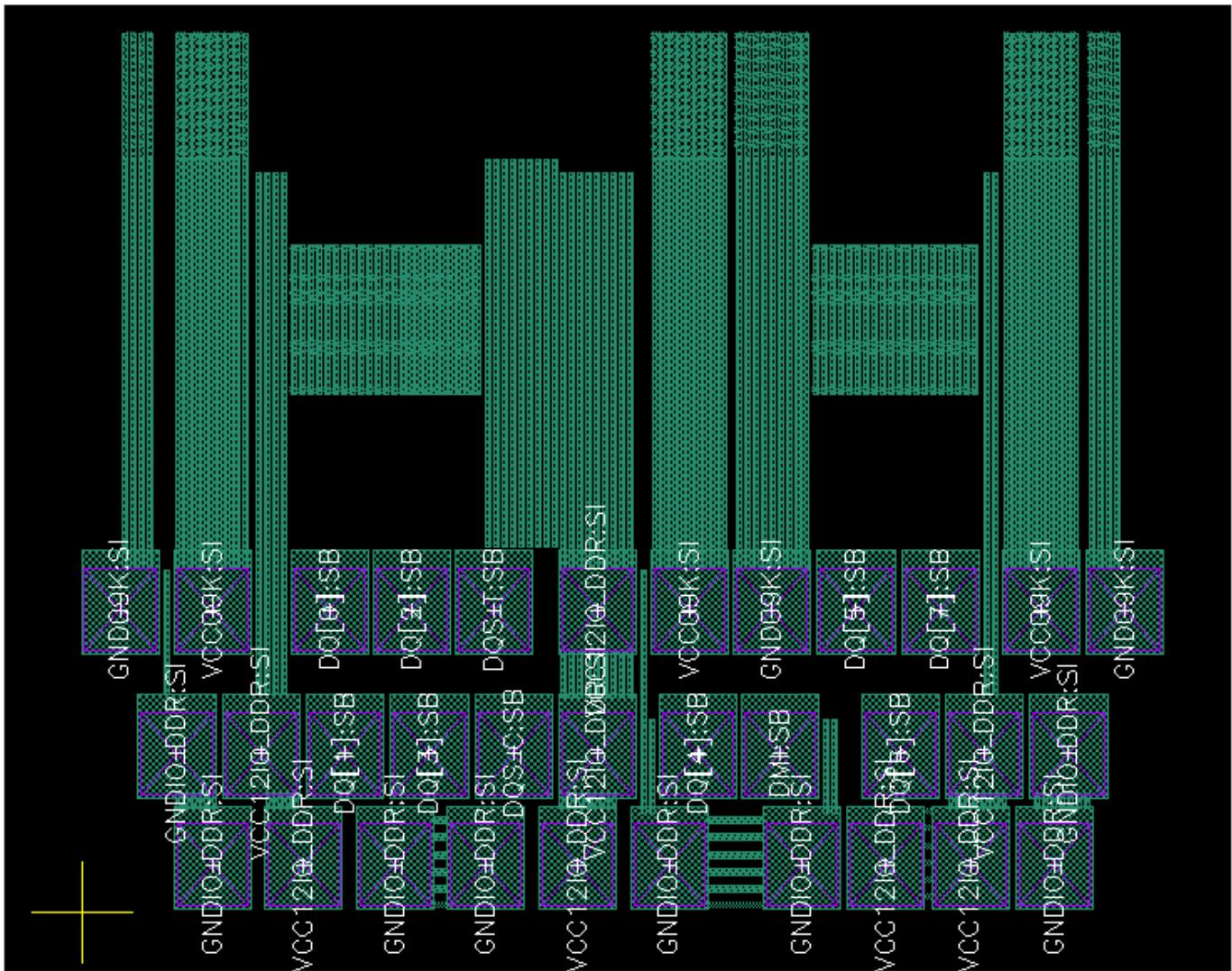


Figure 5-22. Pad Sequence of FXLPDDR4CMSD100NSHJOP

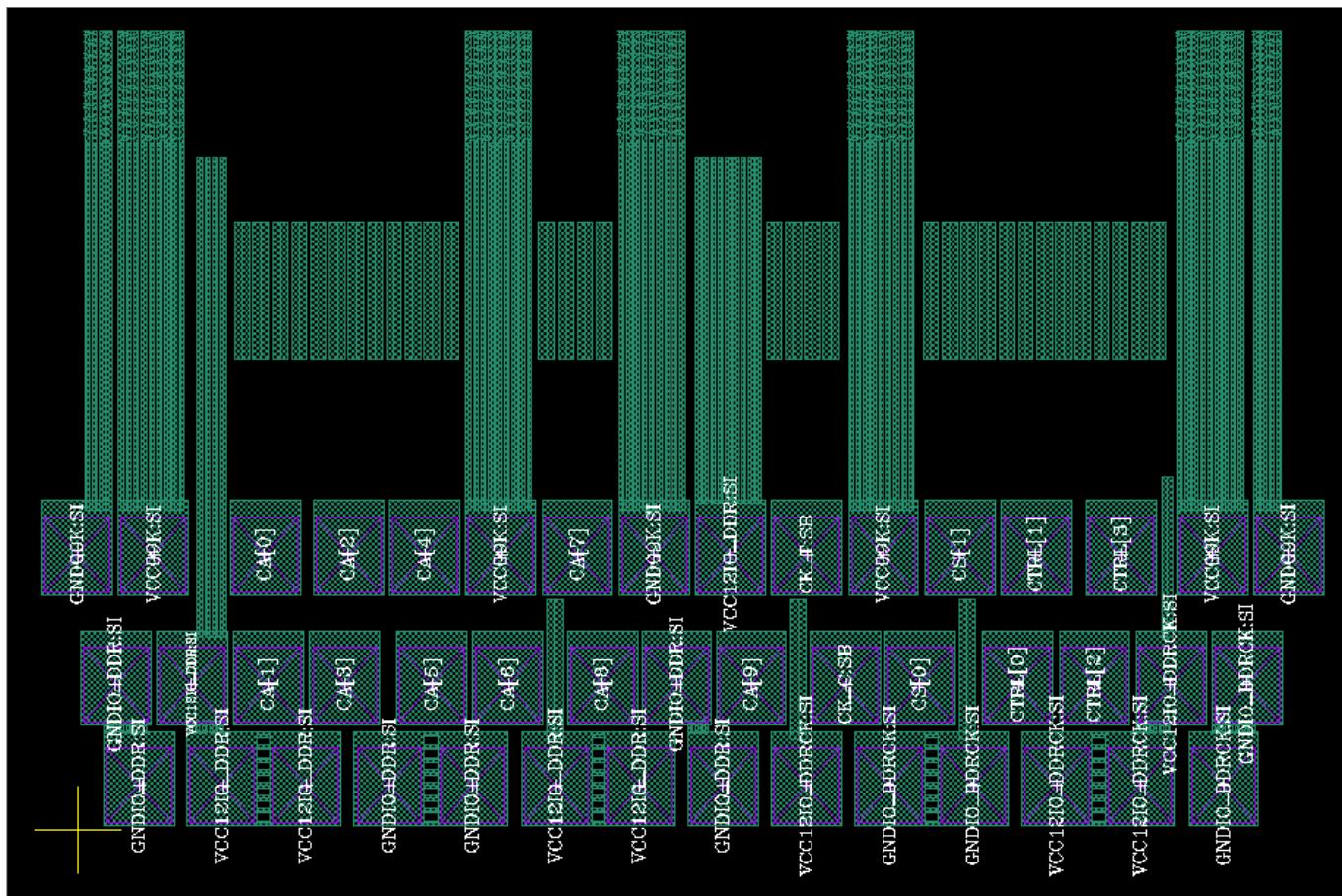


Figure 5-23. Pad Sequence of FXLPDDR4CMSC100NSHJOP

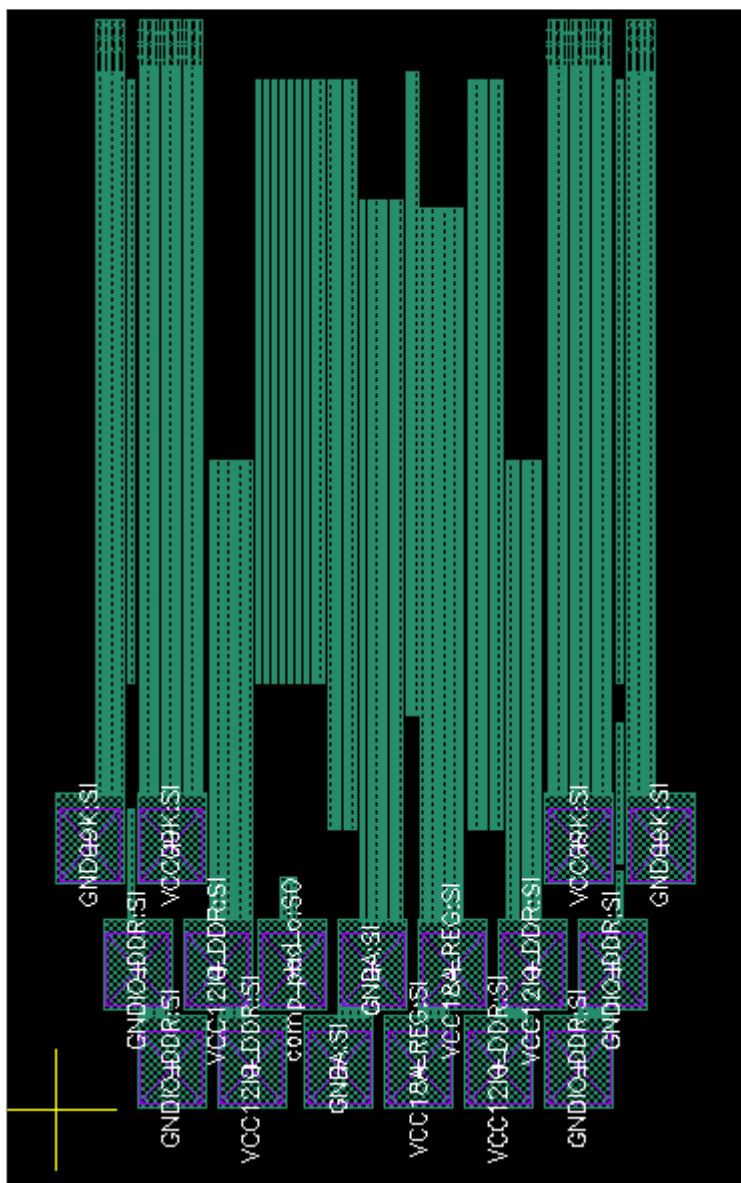


Figure 5-24. Pad Sequence of FXLPDDR4CMSC100NSHJOP

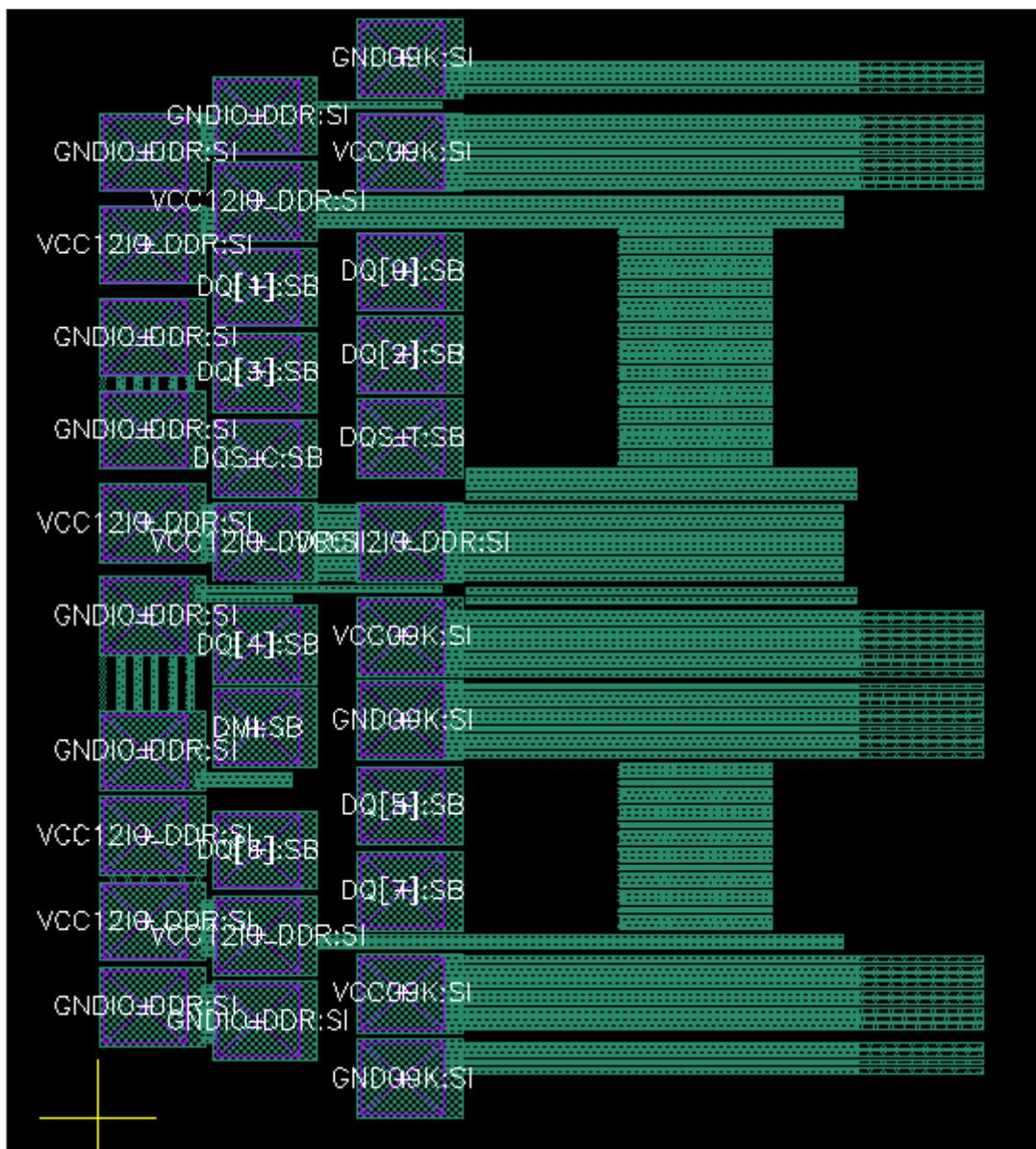


Figure 5-25. Pad Sequence of FXLPDDR4CMSC100EWHJOP

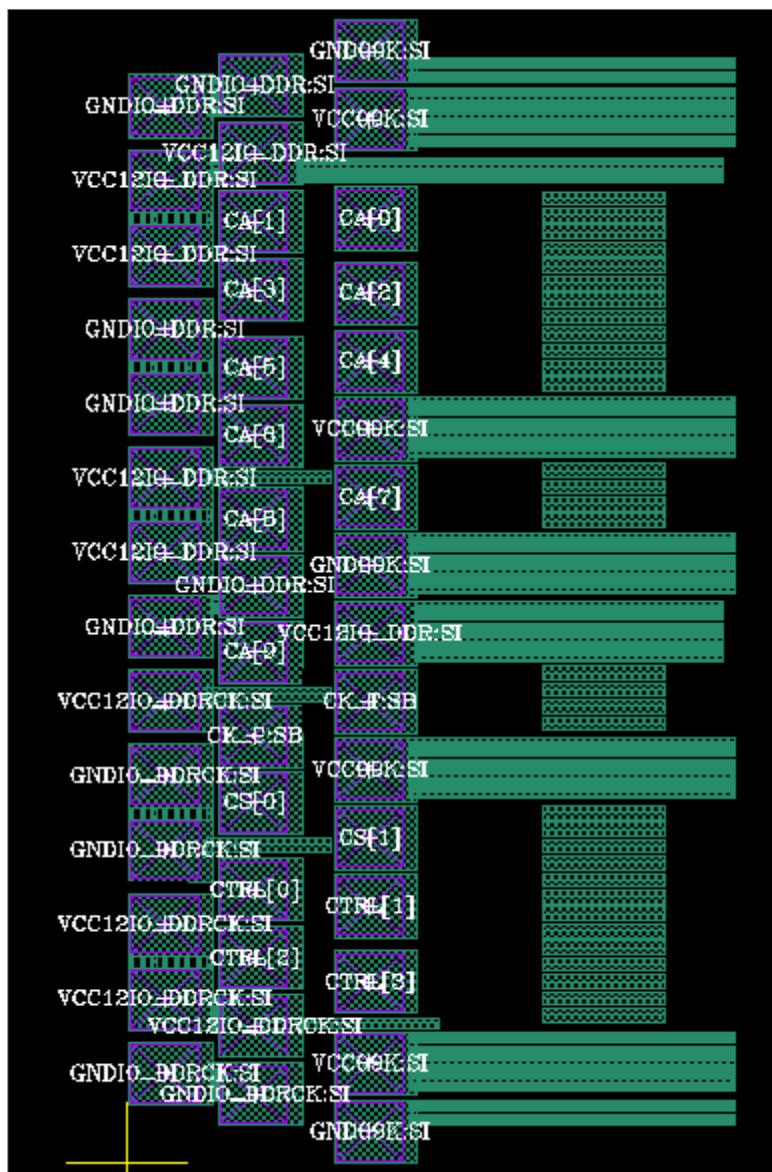


Figure 5-26. Pad Sequence of FXLPDDR4CMSA100EWHJOP

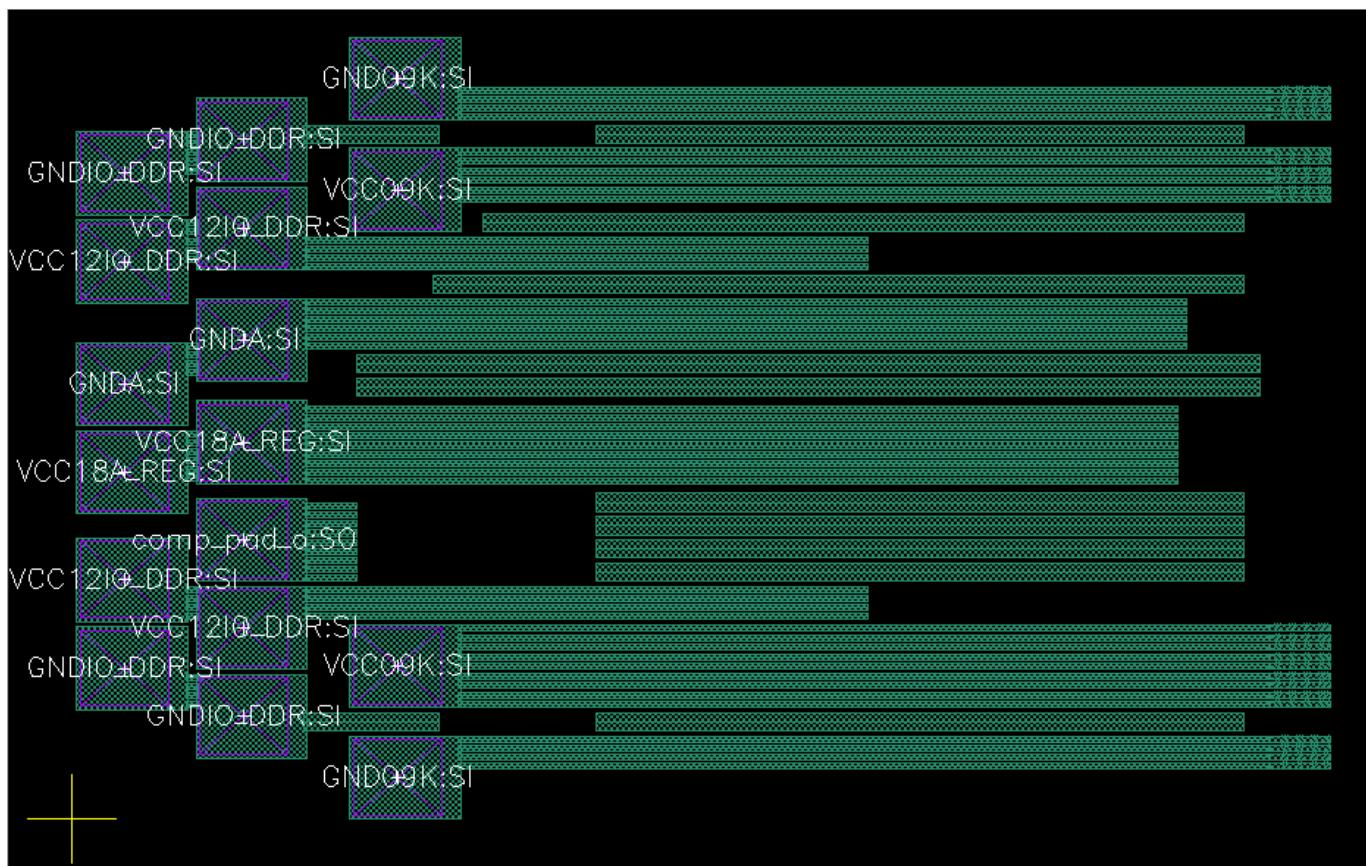


Figure 5-27. Pad Sequence of FXLPDDR4CMSC100EWHJOP

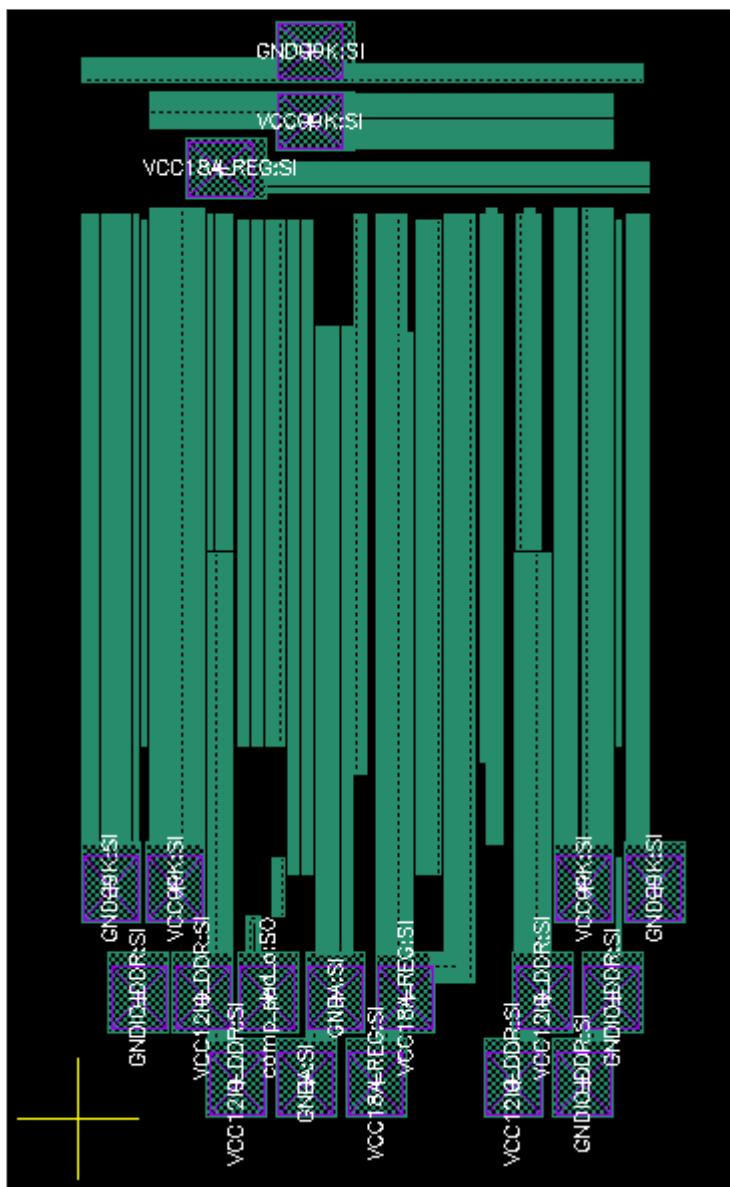


Figure 5-28. Pad Sequence of FXLPDDR4CMSC100HJOP (Placed on the Corner)

5.7 Bit Swapping in Data Block and Address/Command Block

LPDDR4/LPDDR3-DDR4/DDR3 Combo-PHY, by default, does not support bit-swapping as it is. Instead, the bit-swapping is supported in the DFIW block. If user wants to swap the bits on package/board due to routing constraints, please contact Faraday service team for more detailed information.

5.8 Board-level Connection for Compensation Block in FXLPDDR4CMSC100*HJOP

The compensation block is presented in FXLPDDR4CMSC100*HJOP and it is mandatory to follow the below guidelines for the PHY to perform internal calibration of driver impedances. It would be difficult to perform internal calibration of driver output impedances without meeting the requirements in the following guidelines.

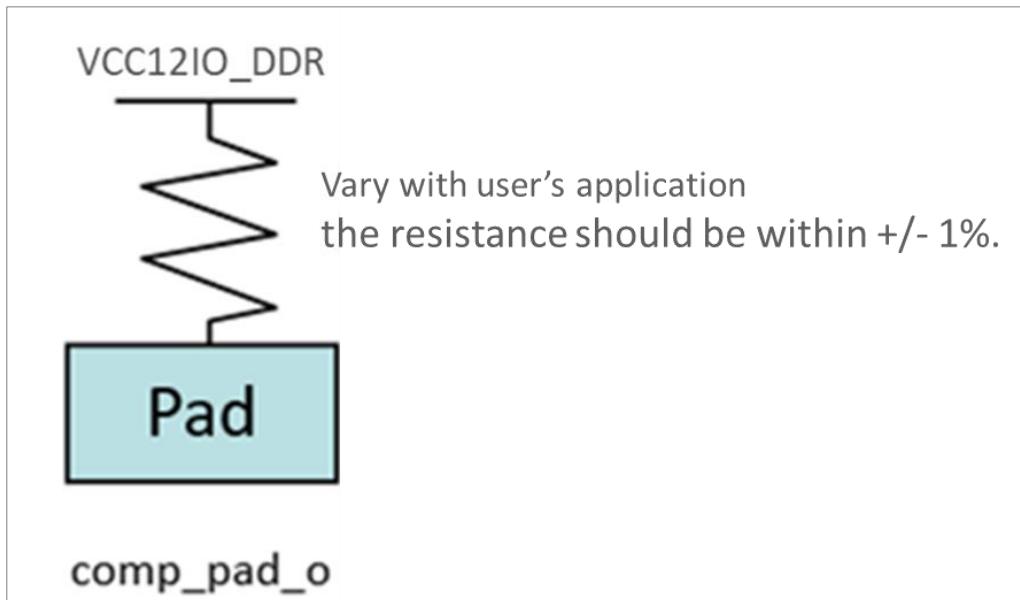


Figure 5-29. Required Connection to PAD ("comp-pad_o") on the Board

At the system/board level, the compensation block needs a resistor to be connected as described below:

An external resistor should be connected between "comp_pad_o" and "VCC12IO_DDR" to calibrate the NMOS driving impedance. The external resistor value is based on user's application. For more details on value, please refer to **Table 3-4** for more details. Once the NMOS is calibrated with this external resistor, PMOS is calibrated with the already calibrated NMOS within the compensation block.

5.9 Power Consumption Estimation

Table 5-17 ~ Table 5-19 illustrate the power consumption estimation in different DDR type at different data rates, which are based on the reasonable activity factors and toggle rate assumption of the command, address clock, and data block.

- Activity factor for DQ: 0.5
- Activity factor for CA: 0.2
- Percentage time for Write: 0.3
- Percentage time for Read: 0.3
- Percentage time for Idle: 0.4

5.9.1 Power Consumption in DDR4

Table 5-17 shows the power consumption estimations of each block in DDR4 mode at different data rates. To measure VCCIO_DDR power, drive strength of $34\ \Omega$ and termination resistance of $40\ \Omega$ are considered. Actual silicon results may vary based on read/write transactions, activity factor, drive strengths, and termination resistances.

Table 5-17. Power Consumption in DDR4 at Different Data Rates

Application Mode	Data Rate (Mbps)	Power Supply Name	DPHY* Power Consumption (mW)	APHY* Power Consumption (mW)	CPHY* Power Consumption (mW)
DDR4	1600	VCC18A_REG	1.11	1.11	14.61
		VCC12IO_DDR + VCC12IO_DDRCK	44.22	18.02	3.78
		VCC09K	23.56	21.65	1.01

Notes:

1. DPHY means “FXLPDDR4CMSC100*HJOP”.
2. APHY means “FXLPDDR4CMSC100*HJOP”.
3. CPHY means “FXLPDDR4CMSC100*HJOP”.

5.9.2 Power Consumption in DDR3

Table 5-18 shows the power consumption estimations of each block in DDR3 mode at different data rates. To measure VCCIO_DDR power, drive strength of $34\ \Omega$ and termination resistance of $40\ \Omega$ are considered. Actual silicon results may vary based on read/write transactions, activity factor, drive strengths, and termination resistances.

Table 5-18. Power Consumption in DDR3 at Different Data Rates

Application Mode	Data Rate (Mbps)	Power Supply Name	DPHY* Power Consumption (mW)	APHY* Power Consumption (mW)	CPHY* Power Consumption (mW)
DDR3	1600	VCC18A_REG	1.11	1.11	14.61
		VCC12IO_DDR + VCC12IO_DDRCK	84.39	42.11	4.72
		VCC09K	22.68	21.64	1.01
	800	VCC18A_REG	1.11	1.11	14.61
		VCC12IO_DDR + VCC12IO_DDRCK	72.62	28.14	2.63

Application Mode	Data Rate (Mbps)	Power Supply Name	DPHY* Power Consumption (mW)	APHY* Power Consumption (mW)	CPHY* Power Consumption (mW)
	VCC09K		15.28	11.82	0.51

Notes:

1. DPHY means “FXLPDDR4CMSC100*HJOP”.
2. APHY means “FXLPDDR4CMSA100*HJOP”.
3. CPHY means “FXLPDDR4CMSC100*HJOP”.

5.9.3 Power Consumption in LPDDR4

Table 5-19 shows the power consumption estimations of each block in LPDDR4 mode at different data rates. To measure VCCIO_DDR power, drive strength of $40\ \Omega$ and termination resistance of $40\ \Omega$ are considered. Actual silicon results may vary based on read/write transactions, activity factor, drive strengths, and termination resistances

Table 5-19. Power Consumption in LPDDR4 at Different Data Rates

Application Mode	Data Rate (Mbps)	Power Supply Name	DPHY* Power Consumption (mW)	APHY* Power Consumption (mW)	CPHY* Power Consumption (mW)
LPDDR4	1600	VCC18A_REG	1.11	1.11	14.61
		VCC12IO_DDR + VCC12IO_DDRCK	20.62	22.40	3.46
		VCC09K	23.37	21.65	1.01

Notes:

1. DPHY means “FXLPDDR4CMSC100*HJOP”.
2. APHY means “FXLPDDR4CMSA100*HJOP”.
3. CPHY means “FXLPDDR4CMSC100*HJOP”.

5.9.4 Power Consumption in LPDDR3

Table 5-20 shows the power consumption estimations of each block in LPDDR3 mode at different data rates. To measure VCCIO_DDR power, drive strength of $34\ \Omega$ and termination resistance of $40\ \Omega$ are considered. Actual silicon results may vary based on read/write transactions, activity factor, drive strengths, and termination resistances.

Table 5-20. Power Consumption in LPDDR3 at Different Data Rates

Application Mode	Data Rate (Mbps)	Power Supply Name	DPHY* Power Consumption (mW)	APHY* Power Consumption (mW)	CPHY* Power Consumption (mW)
LPDDR3	1600	VCC18A_REG	1.11	1.11	14.61
		VCC12IO_DDR + VCC12IO_DDRCK	44.22	18.02	3.78
		VCC09K	23.56	21.65	1.01
	800	VCC18A_REG	1.11	1.11	14.61
		VCC12IO_DDR + VCC12IO_DDRCK	36.93	9.36	2.11
		VCC09K	16.13	11.82	0.51

Notes:

1. DPHY means “FXLPDDR4CMSC100*HJOP”.
2. APHY means “FXLPDDR4CMSC100*HJOP”.
3. CPHY means “FXLPDDR4CMSC100*HJOP”.

5.9.5 Total Power Estimation Example

The total power consumption also depends on the number of used blocks.

Total power consumption

= Number of data blocks * Power consumption of data block + Number of command and address blocks * Power consumption of command and address block + Power consumption of clock-generation block

Examples:

- Application: Single-channel, 16-bit, DDR4-1600
- There are two command/address blocks, two data blocks, and one clock-generation block along in this application.
- Drive strength is $40\ \Omega$.
- The termination resistance is $60\ \Omega$.

$$\text{Total power consumption} = 2 * (1.11 + 44.22 + 23.56) + 2 * (1.11 + 18.02 + 21.65) + 1 * (14.61 + 3.78 + 1.01) = 243.08 \text{ (mW)}$$

Chapter 6

System Application Circuit

This chapter contains the following sections:

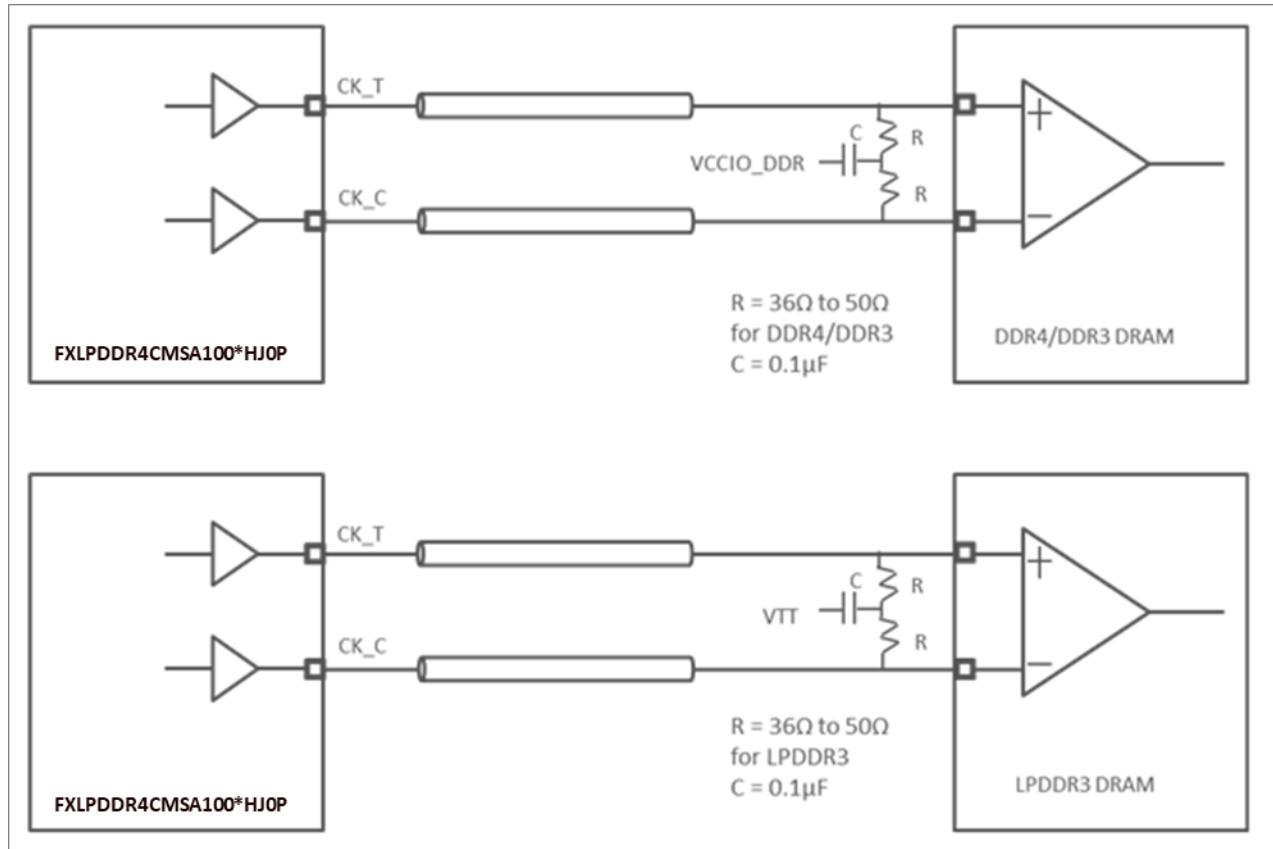
- 6.1 Termination Connections
- 6.2 Recommended Connection for ODT_CA Pins in LPDDR4
- 6.3 Built-in VREF Generator Usage
- 6.4 Package Requirement for I/O Power/Ground
- 6.5 Package Requirement for Core Power
- 6.6 PCB Layout Guideline
- 6.7 Power Sequencing Requirement for PHY

6.1 Termination Connections

Since LPDDR4 supports On-die termination for data/address/clock, there is no need for any external termination on-board.

- a) Connections for CK_T and CK_C in FXLPDDR4CMSC100*HJOP on PCB:

It is recommended placing a termination resistor between the CK_T and CK_C pins, as shown in **Figure 6-1**. Based on the Faraday experiences, an estimated value between 36Ω and 50Ω should be used. As for LPDDR3, the resistor between the CK_T and CK_C pins is not necessary. Please contact the DRAM vendors for the exact value of the applied termination resistor.



- b) Connections for other pins in FXLPDDR4CMSC100*HJOP on PCB:
 Two examples are connecting termination resistors as shown in **Figure 6-2** and **Figure 6-3**.

Figure 6-2 shows the first example.

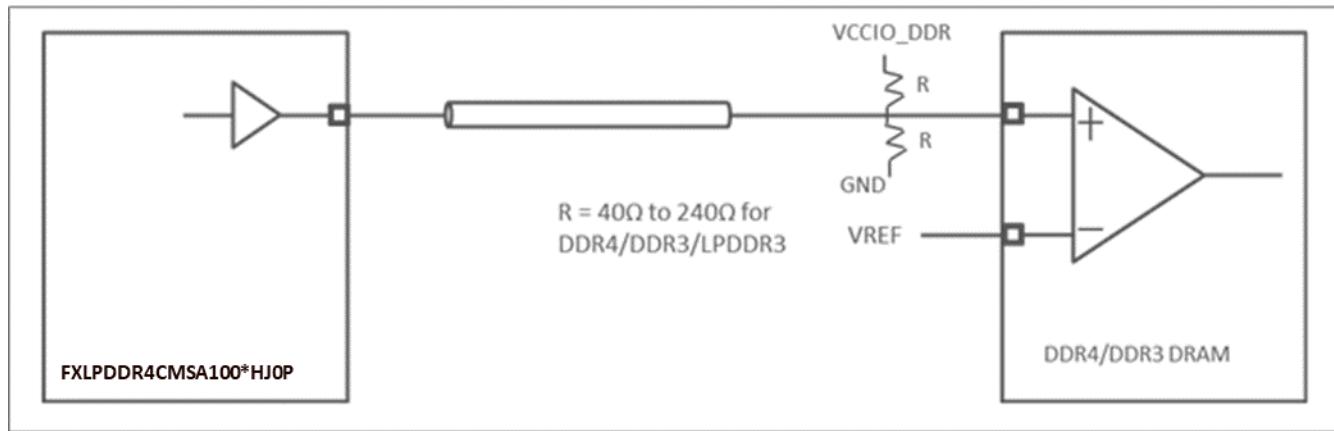


Figure 6-2. First Example of Connection of Termination to Other Pins in Address Block on PCB

Figure 6-3 shows the second example.

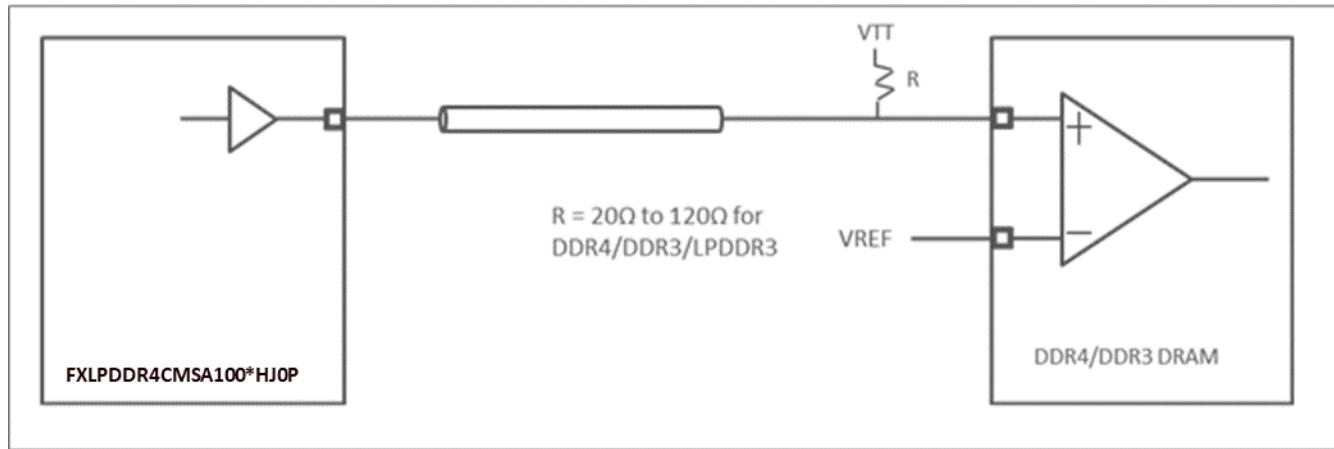


Figure 6-3. Second Example of Connection of Termination to Other Pins in Address Block on PCB

- c) Connections for pins in FXLPDDR4CMSC100*HJOP on PCB:
 Since all pins like DQ/DMI/DQS in Data block use On-die termination resistors within DRAM for DDR4/DDR3/LPDDR3, there is no explicit connection needed on PCB. To use On-die termination resistors within the DRAM, they must be enabled through appropriate mode registers for that particular standard.

6.2 Recommended Connection for ODT_CA Pins in LPDDR4

Since the controller/PHY does not control On-die termination for address bus within LPDDR4 DRAM, it is advisable to connect “ODT_CA_A” and “ODT_CA_B” pins of LPDDR4 DRAM to VDD2 (logic high) on the PCB. Please refer to LPDDR4 DRAM data sheet for VDD2 voltage level.

6.3 Built-in VREF Generator Usage

This LPDDR4/3-DDR4/3 Combo PHY supports only internal VREF generation for the data block and hence there is no provision to supply VREF on-board from an external pin.

6.4 Package Requirement for I/O Power/Ground

- Requirement 1: VCCIO_DDRCK from different address/command blocks can be shorted on package.
- Requirement 2: VCCIO_DDR from clock-generation block, data block and address/command block can be shorted on package.
- Requirement 3: GND , GNDIO_DDR and GNDIO_DDRCK from clock-generation block, data block, and address/command block can be shorted on package.
- Requirement 4: GNDA and VCC18A_REG needs to be separate on package for better noise immunity for PLL.

Figure 6-4 ~ Figure 6-7 show the conceptual diagrams that show the connections from die-nodes to package.

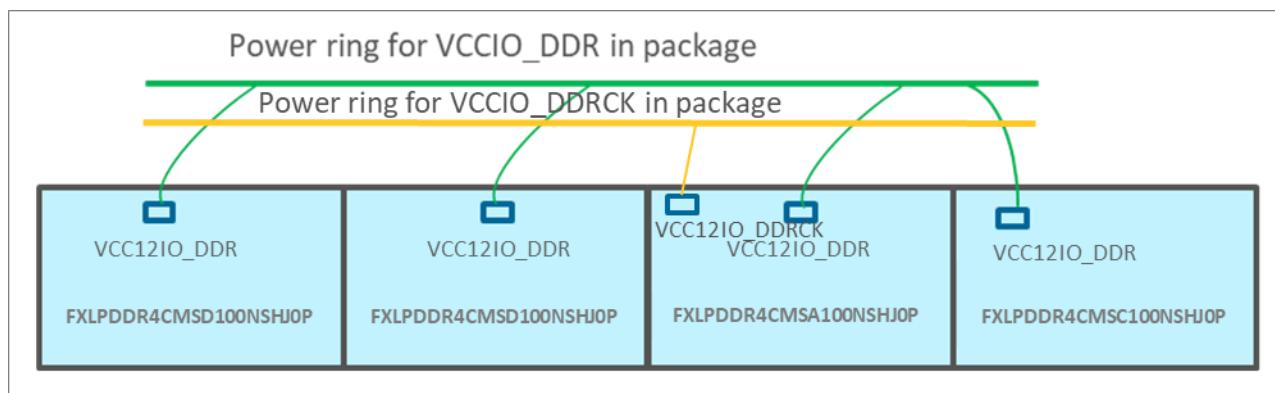


Figure 6-4. Connection of VCCIO_DDR and VCCIO_DDRCK of PHY in Package

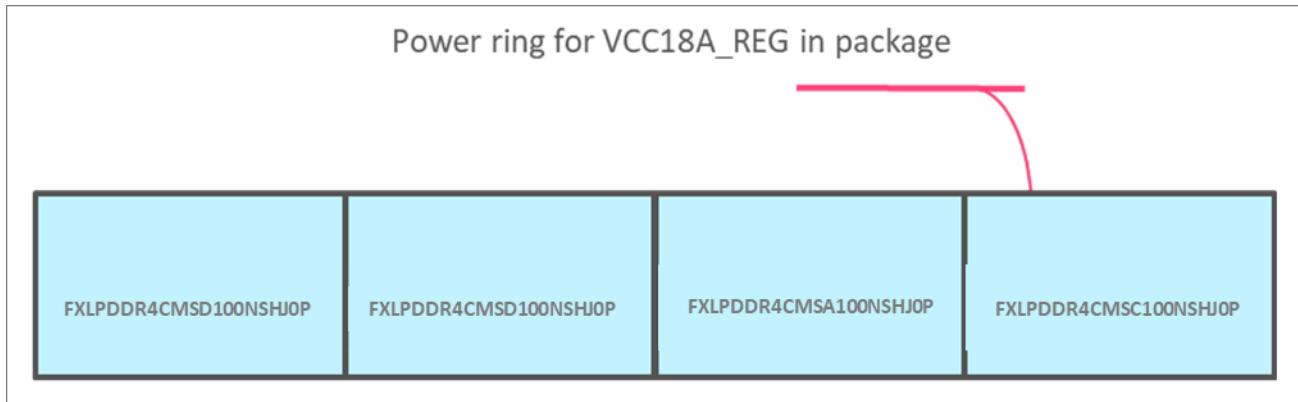


Figure 6-5. Connection of VCC18A_REG of PHY in Package

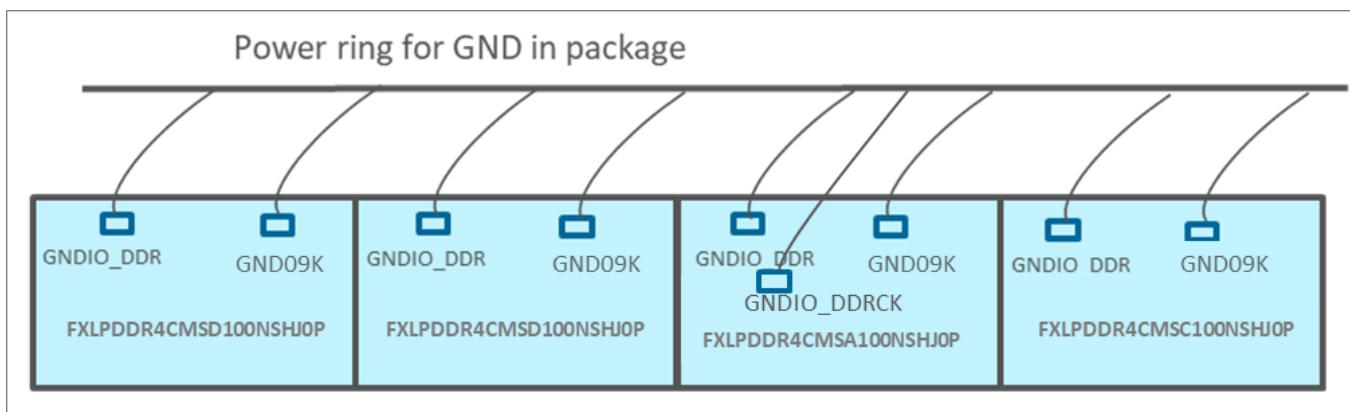


Figure 6-6. Connection of GND, GNDIO_DDR and GNDIO_DDRCK of PHY in Package

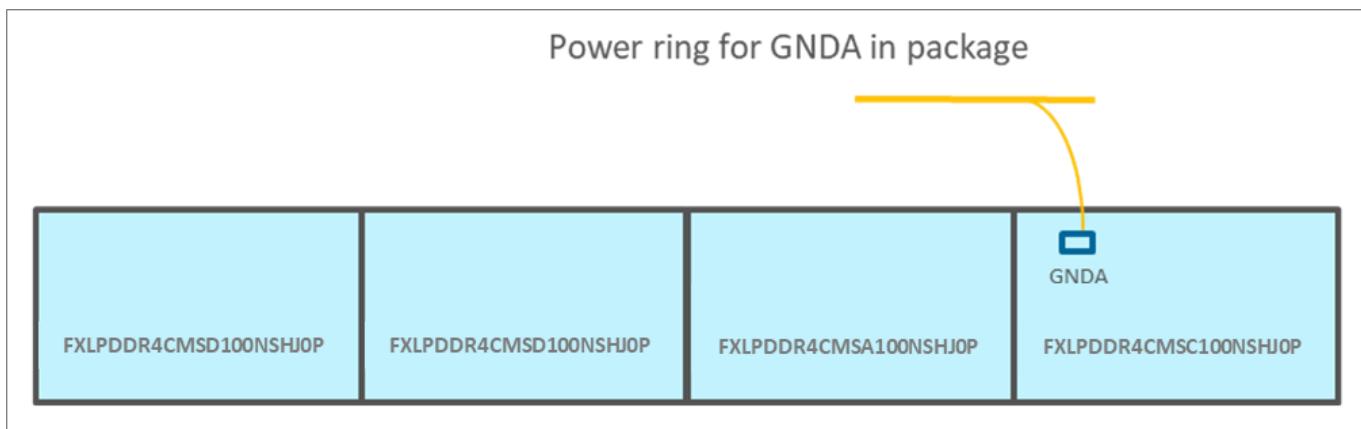


Figure 6-7. Connection of GNDA of PHY in Package

6.5 Package Requirement for Core Power

Requirement 1: The VCC09K of PHY's core power can be merged with SoC's core power VCC09K in Layout/GDS.

6.6 PCB Layout Guideline

For PCB design guideline, please refer to Faraday "DDRn/LPDDRn PCB design guideline", which has specified the PCB design guideline with the following check items:

1. Trace Characteristics Impedance
2. Power/GND domain
3. General Signal Routing (DQ, DM, and ADD/CMD)
4. Differential Signal Routing (Clock and DQS)

6.7 Power Sequencing Requirement for PHY

LPDDR4/LPDDR3-DDR4/DDR3 PHY needs a power sequencing requirement for VCC18A_REG only.

While powering up, VCC18A_REG needs to be ramped up after all other power supplies like VCC12IO_DDR, VCC12IO_DDRCK, and VCC09K. There is no sequencing requirement among the supplies VCC12IO_DDR, VCC12IO_DDRCK, and VCC09K while powering up.

While powering down, VCC18A_REG needs to be ramped down before all other power supplies like VCC12IO_DDR, VCC12IO_DDRCK, and VCC09K. There is no sequencing requirement among the supplies VCC12IO_DDR, VCC12IO_DDRCK, and VCC09K while powering down.

The above-mentioned sequencing requirement for VCC18A_REG needs to be followed to avoid unnecessary DC leakage in VCC18A_REG.

Chapter 7

FAQ

Faraday provides the most frequently asked questions and answers in the following pages for your reference.

- Q1: Do I need “start-up sequence” for all FXLPDDR4CMSC100*HJOP, FXLPDDR4CMSC100*HJOP, and FXLPDDR4CMSC100*HJOP?
- Q2: How to run the zero delay simulation?
- Q3: How to run the post simulation with SDF annotate?
- Q4: Does user have the implementation check list of DDR-PHY?

Q1: Do I need “start-up sequence” for all **FXLPDDR4CMSA100*HJOP, **FXLPDDR4CMSC100*HJOP**, and **FXLPDDR4CMSC100*HJOP**?**

A1:

Yes, all the IPs need “start-up sequence”. After the power-on reset, IP will start initialization sequence and generate the phy_ready signal. Once IP generates phy_ready, it means that it is ready for operation.

Q2: How to run the zero delay simulation?

A2:

Please define “ddr_allzero” and “FXLPDDR4CMSACD_PRE_NETLIST_SIM” in the run command.

Q3: How to run the post simulation with SDF annotate?

A3:

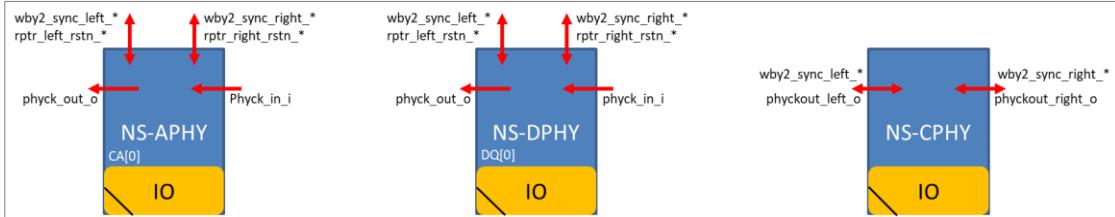
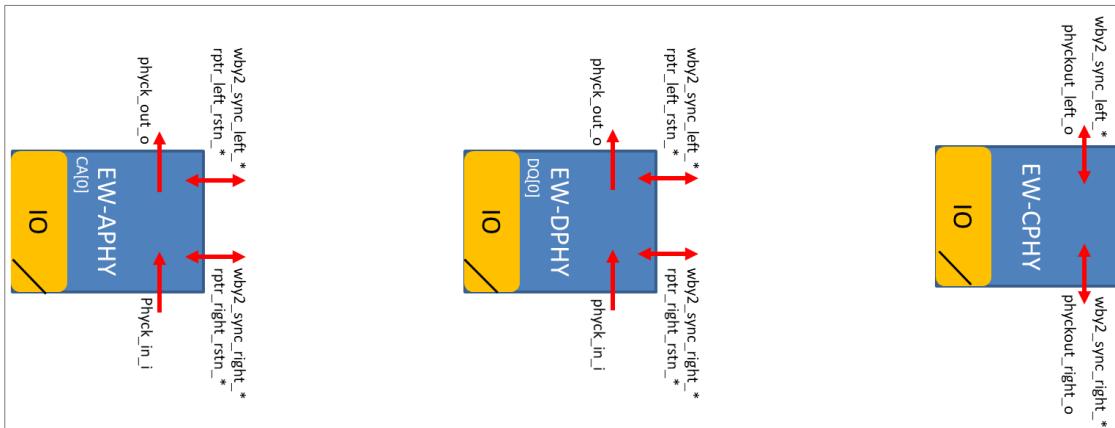
Please define “ddr_allzero”. For TC, WC, and WCC, please define “maxdelays” in the run command. For BC and BCH, please define “mindeals” in the run command. Adding timing disable list in the run command for some registers would face timing violation on CDC region. Please use the *.tfile in database.

Q4: Does user have the implementation check list of DDR-PHY?

A4:

Table 7-1 lists the implementation check items of DDR-PHY.

Table 7-1. Implementation Check List of DDR-PHY

Check Item	Pass/Fail
Specific direction pin information (Basic structure information only, check item is in row 10.)	
Users have to know that all the IPs have specific direction pin.	
Once the floor plan and instance sequence are reviewed, it cannot be changed.	
Keyword to recognize each PHY:	
"**MSA*NS**" --> NS_APHY	
"**MSD*NS**" --> NS_DPHY	
"**MSC*NS**" --> NS_CPHY	
	
Specific direction pin information (Basic structure information only, check item is in row 10.)	
Users have to know that all the IPs have specific direction pin.	
Once the floor plan and instance sequence are reviewed, it cannot be changed.	
Keyword to recognize each PHY:	
"**MSA*EW**" --> EW_APHY	
"**MSD*EW**" --> EW_DPHY	
"**MSC*EW**" --> EW_CPHY	
	

Floor plan of all PHYs

Since all the IPs have specific direction pins, the instance sequence and rotation degree cannot be placed and changed carelessly.

In the beginning of floor plan stage, please provide the following information and ask manager of ITD/DDDRPHY2 to review if the following item is consistent.

1. PHY_TOP.v --> **Provide connection of "wby2_sync*" and "phyck_***" and "rptr_*_rstn_*" of every single PHY
2. Instance sequence and Rotation degree of each IP --> **Provide snapshot of current layout floor plan**

Check Item

Pass/Fail

OSC location

There is a 10 MHz ~ 50 MHz source clock as reference clock for the embedded PLL in CPHY.

Please make the OSC as close to CPHY as possible. --> **Provide snapshot of current layout floor plan**

OSC output and clock tree electrical Specifications and placement requirement

Clock tree from OSC to CPHY, please place on the ASIC core boundary (Through the low-noise area) --> **Provide snapshot of current layout floor plan**

Item	Description	Min.	Typ.	Max.	Unit
Duty	Input clock duty ratio	30	50	70	%
PN	Input clock phase noise At 100 kHz and over offset frequency	-	-	135	dBc/Hz
FREF period jitter (Peak-to-peak)	10 MHz ≤ FREF < 50 MHz	-	-	±1%	Period
	50 MHz ≤ FREF < 100 MHz	-	-	±2%	

Abutted signals

Here is an example of NS version, please check the following figure and table. (EW version is the same as NS version) All the abutted signals should be set as “don't touch” and shouldn't be inserted any logic cell-->

Provide connection information

PHY block	Abutted signals
NS_APHY EW_APHY	bg_vref_i, por_vcc12io_ddr_io, ret_vcc12io_ddr_io test_pad_vcc18a_reg_io, phyck_in_i, phyck_out_o
NS_DPHY EW_DPHY	bg_vref_i, por_vcc12io_ddr_io, ret_vcc12io_ddr_io test_pad_vcc18a_reg_io, phyck_in_i, phyck_out_o
NS_CPHY EW_CPHY	bg_vref_o, por_vcc12io_ddr_o, ret_vcc12io_ddr_o test_probe_i, phyckout_left_o, phyckout_right_o, phyck_in_dmy_o (floating dummy pin)

Metal routing signals (wby2ck sync right i, wby2ck sync right o, wby2ck sync left i, wby2ck sync left o)

Here is an example of NS version, please check the following figure and table. (EW version is the same as NS version.) All the metal routing signals should be set as “don't touch” and shouldn't be inserted any logic cell -->

Provide connection information

From the points of view on layout, users have to make sure that these pins should be very close.

PHY block	Metal routing signals
NS_APHY EW_APHY	wby2ck_sync_right_i, wby2ck_sync_right_o, wby2ck_sync_left_i, wby2ck_sync_left_o
NS_DPHY EW_DPHY	wby2ck_sync_right_i, wby2ck_sync_right_o, wby2ck_sync_left_i, wby2ck_sync_left_o
NS_CPHY EW_CPHY	wby2ck_sync_right_i, wby2ck_sync_right_o, wby2ck_sync_left_i, wby2ck_sync_left_o

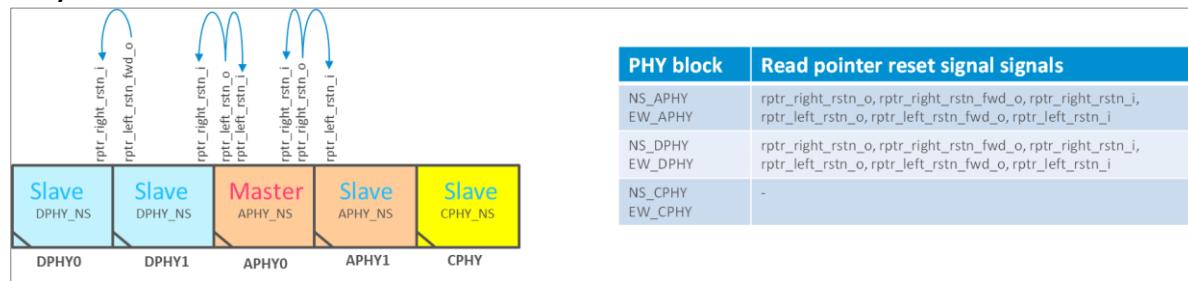
Check Item

Pass/Fail

Read pointer reset signal (rptr_right_rstn_o, rptr_right_rstn_fwd_o, rptr_right_rstn_i, rptr_left_rstn_o, rptr_left_rstn_fwd_o, rptr_left_rstn_i)

Here is an example of NS version, please check the following figure and table. (EW version is the same as NS version.)

From the points of view on layout, all the read pointer reset signals should be very close. --> **Provide snapshot of layout**



DDR PHY Power domain requirement for die level and package level

Please follow the below table to plan ASIC power domain --> **Provide netlist and snapshot of layout**

Power/Ground pin	Die level	Package level
VCC18A_REG	Independent analog power for embedded PLL	Independent analog power for embedded PLL
GND_A	Independent analog Ground for embedded PLL	Independent analog Ground for embedded PLL
VCC12IQ_DDR	Independent IO Power for DDR-PHY	Independent IO Power for DDR-PHY
GNDIO_DDR	Independent IO Ground for DDR-PHY	Merge with SOC GND
VCC09K	Merge with SOC VCCK	Merge with SOC VCCK
GND09K	Merge with SOC GNDK	Merge with SOC GND

Requirement of retention_en_i (CPHY)

This pin is used to retention the IO pad signal of DDR I/O when the VCCK of PHY is at power-down stage.

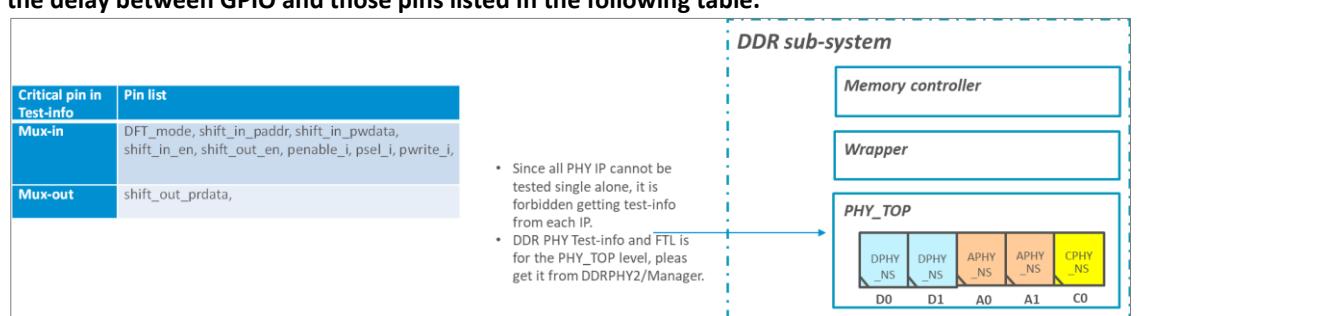
The input voltage level can be 0.9 V or 1.8 V. --> **Provide the input signal voltage level**

If user needs the VCCK power-down mode, please make sure that all the logic cells of the signals are set on always alive power domain. --> **Provide the snapshot of connection and power domain information**

test-info and fti

Here is an example of NS version, please check the following figure and description before planning the test mode. (EW version is the same as NS version.) test-info and fti are for the PHY_TOP level and they will be different by different configurations. Please get the proper test-info and fti from DDRPHY2/Manager --> **Get fti & test-info and provide ASIC test-info**

There are some critical pins that need to be taken care of their latency, please use the nearby IO. --> **Provide the delay between GPIO and those pins listed in the following table.**



Check Item	Pass/Fail
<u>Register setting review</u>	
1. wby2ck auto sync	
2. Read pointer rest	
3. cktree skew	
4. Pin swap setting	